z/OS Communications Server

SNA Network Implementation Guide

Version 1 Release 13

SC31-8777-11

Note:
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 789.

Twelfth Edition (September 2011)

This edition applies to Version 1 Release 13 of z/OS (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You may send your comments to the following address.

International Business Machines Corporation
Attn: z/OS Communications Server Information Development
Department AKCA, Building 501
P.O. Box 12195, 3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195

You can send us comments electronically by using one of the following methods:

Fax (USA and Canada):

1+919-254-1258

Send the fax to "Attn: z/OS Communications Server Information Development"

Internet email:

comsvrcf@us.ibm.com

World Wide Web:

http://www.ibm.com/systems/z/os/zos/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number. Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corporation 1984, 2011.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures . . . . . **xv**

Tables . . . . . **xix**

About this document . . . . . **xxi**
Who should read this document xxi
How this document is organized xxi
How to use this document xxii
Determining whether a publication is current xxii
How to contact IBM service xxiii
Conventions and terminology that are used in this document xxiii
How to read a syntax diagram xxiv
Prerequisite and related information xxvi
How to send your comments xxx

Summary of changes . . . . . **xxxi**
Changes made in z/OS Communications Server Version 1 Release 13 xxxi
Changes made in z/OS Communications Server Version 1 Release 12 xxxi
Changes made in z/OS Communications Server Version 1 Release 11 xxxii

Chapter 1. Post-installation considerations for z/OS Communications Server **1**
Defining Communications Server SNA to z/OS. 1
Using automatic restart manager 1
Starting z/OS Communications Server SNA 2

Chapter 2. VTAM networking concepts **5**
VTAM functions 5
VTAM nodes 5
Nodes with APPN function only. 5
Nodes with subarea function only 9
Nodes with both subarea and APPN function 10
Network accessible units 12
Physical unit 12
Logical unit 12
Network control sessions 14
SSCP-SSCP sessions 14
CP-CP sessions 14
End-user sessions 15
How VTAM locates resources 16
Locating resources in a subarea network. 16
Locating resources in an APPN network. 17
Route selection 17
Routing in a subarea network 17
Routing in an APPN network 18
Addressing 20
Subarea addressing 21
APPN addressing 21
Controlling network data flow using pacing 21
Pacing concepts 22
Pacing window 22

Chapter 3. Implementing a VTAM network. **25**
Using start options and configuration lists 27
Start options 28

© Copyright IBM Corp. 1984, 2011

Configuration lists . . . 37
Identifying resources to VTAM . . . 38
Coding concepts . . . 38
Sift-down effect . . . 39
Using MVS system symbols . . . 39
Verifying a VTAM network . . . 46
Verifying a multiple-domain subarea network . . . 46
Verifying a multiple-network environment . . . 47
Verifying a VTAM APPN network . . . 47

Chapter 4. Connecting an APPN node to VTAM . . . 49
Connections through boundary function-based transmission groups . . . 50
Multiple connections with parallel transmission groups . . . 50
Channel connections between APPN nodes . . . 51
Multipath channel connections . . . 51
Composite network node channel connections . . . 58
Leased connections between APPN nodes . . . 59
IBM 3172 Nways Interconnect Controller connections between APPN nodes . . . 60
Example . . . 61
Using a connection network . . . 62
IBM Open Systems Adapter connections between APPN nodes . . . 68
APPN multiple network connectivity . . . 89
Peripheral subnetwork boundaries . . . 91
Extended subnetwork boundaries . . . 91
APPN multiple network connectivity support . . . 92
Virtual-route-based transmission groups . . . 95
Defining a VR-based TG . . . 97
VR-based TG recommendations . . . 98
Selecting the network node server for end nodes . . . 100
Creating a network node server list . . . 100
Activating, replacing, and displaying a network node server list . . . 101
Using the NNSPREF start option . . . 101

Chapter 5. Connecting a subarea node to VTAM . . . 103
Connecting two VTAMs using channels . . . 103
Channel-to-channel adapter connection . . . 103
Multipath channel connections . . . 105
Connecting two VTAMs through communication controllers . . . 106
Nonswitched SDLC subarea connection . . . 106
Switched SDLC subarea connection . . . 108
X.25 subarea connection (NPSI support) . . . 110
Token-ring connection (NTRI support) . . . 111
Sample NCP definition statements for token-ring connection . . . 111
Connecting two VTAMs using an external communication adapter . . . 112
Sample configuration with Ethernet or Ethernet-type LAN . . . 112
Sample configuration with a token-ring local area network . . . 114
Connecting an NCP to VTAM . . . 115
Channel-attached NCP . . . 116
Channel-attached NCP to a VTAM data host . . . 120
NCP definition statements . . . 122
Connecting an NCP to another NCP . . . 127
Recovery operations . . . 127
Defining a link-attached NCP . . . 128
Enhanced remote controller support . . . 129
NCP switched SDLC connections . . . 130

Chapter 6. Using Enterprise Extender (EE) . . . 133
Overview . . . 134
Benefits of Enterprise Extender . . . 134
Availability of Enterprise Extender . . . 134

Hardware requirements . . . 135
EE reliability and strategy . . . 135
Using EE and extended border node (EBN) as a replacement for SNI . . . 136
EE implementation considerations . . . 137
Designing the EE network . . . 137
Distinctions between an EE network and an EE connection network . . . 138
Characteristics of EE connections . . . 138
Static VIPA considerations . . . 139
IP multipath considerations . . . 140
Class of Service preservation dependencies . . . 140
Comparing host name and IP address definitions . . . 141
Configuring the EE network . . . 142
Preservation of SNA transmission priority . . . 144
Network address translation (NAT) considerations . . . 144
Steps for configuring and activating an EE network . . . 144
Configuring the EE connection network . . . 161
Connection network concepts . . . 165
EE connection network rules . . . 166
Contrasting local and global networks . . . 166
Benefits of defining multiple Enterprise Extender virtual routing nodes . . . 167
Defining an EE connection network in the EE XCA major node . . . 170
EE security considerations . . . 172
SNA session level encryption (SLE) . . . 172
IP security (IPSec) . . . 172
Using EE with network address translation (NAT) . . . 173
IP filtering . . . 174
IDS for Enterprise Extender . . . 174
OEM security products — EE proxy solutions . . . 175
Tuning the EE network . . . 176
Tuning Enterprise Extender-specific buffer pools . . . 176
Timers . . . 177
HPR ALIVE timer optimization for Enterprise Extender . . . 180
Enterprise Extender LDLC keep-alive reduction . . . 181
Enterprise Extender improved packet loss tolerance . . . 182
Disconnect and inactivity summary . . . 182
Customizing IP type of service . . . 183
Advanced coding considerations for EE . . . 184
EE connection network reachability awareness . . . 184
TCP/IP MTU size for EE . . . 193
Running EE in constrained or virtualized environments . . . 194
RTP transmission stall operator awareness and recovery support . . . 195
Load balancing . . . 196
Transmission group profiles . . . 197
Dynamic reconfiguration . . . 197
Dial usability - DWACT, DWINOP, KEEPACT, REDIAL, and REDDELAY . . . 197
Customization for EE connection network PUs . . . 198
Cross-subnet routing with global VRNs . . . 199
Troubleshooting EE problems . . . 202

Chapter 7. OSA-Express . . . 209
OSA-Express overview . . . 209
Defining an OSA-Express device to z/OS Communications Server using QDIO . . . 213
OSA routing . . . 215
OSA-Express virtual MAC (VMAC) routing . . . 215
Primary and secondary routing . . . 217
Outbound priorities . . . 219
MTU . . . 219

Chapter 8. Sharing NCP peripheral nodes (CMC configurations) . . . 221
Channel-attached NCPs in a CMC configuration . . . 222

Link-attached NCPs in a CMC configuration . . . 224
Controlling resource ownership in a CMC configuration . . . 224
Problem determination in a CMC configuration . . . 225

Chapter 9. Defining resources dynamically. . . . 227
Defining switched resources dynamically . . . 227
Dynamic PU definition (DYNPU operand) . . . 227
Dynamic switched definitions . . . 227
Dynamic configuration of channel-attached devices. . . . 230
Installation and preparation . . . 232
Defining your configuration . . . 232
Building resource definitions . . . 232
Using the default naming convention . . . 233
Customizing the command lists . . . 234
Dynamic reconfiguration and change of operands . . . 235
Dynamic reconfiguration and dynamic change requirements . . . 237
Using the VARY ACT,UPDATE technique . . . 238
Using the VARY DRDS technique . . . 241
Using the MODIFY DR technique . . . 243
Dynamic reconfiguration of independent LUs . . . 244
Dynamic reconfiguration of frame-relay resources . . . 244
Implicit dynamic reconfiguration . . . 246
Dynamic reconfiguration examples . . . 246
Line backup through dynamic reconfiguration . . . 250

Chapter 10. Defining peripheral nodes. . . . 253
Defining type 2.1 peripheral nodes . . . 253
Nonnative network type 2.1 connections . . . 254
Attaching peripheral nodes to VTAM . . . 255
Local non-SNA connection . . . 255
Local SNA connection . . . 256
Loop-adapter-attached connection . . . 258
External communication adapter (XCA) connections . . . 258
Attaching peripheral nodes to NCP . . . 261
Defining type 2.1 peripheral nodes . . . 261
Nonswitched SDLC connection . . . 262
Switched SDLC connection . . . 262
Defining switched resources used by multiple VTAMs . . . 267
NCP/Token Ring interconnection (NTRI) . . . 268
BSC connection. . . . 270

Chapter 11. Defining LUs . . . 273
Independent LUs . . . 273
Characteristics of independent LUs . . . 273
Defining independent LUs . . . 275
Multiple connections between a type 2.1 node and a subarea node . . . 279
Dynamic selection of session connections . . . 281
Connecting independent LUs through NCPs . . . 281
Authorized transmission priority for LEN connections. . . . 282
Restrictions on using independent LUs . . . 282
Dependent logical units . . . 283

Chapter 12. Establishing and controlling SNA sessions . . . 287
Multicultural support. . . . 288
Multicultural support for end-user USS messages and commands . . . 288
Defining USS tables for use with the LANGTAB operand . . . 289
Defining USS messages to the MVS message service . . . 289
Multicultural support for the language passed to application programs . . . 289
Model name table . . . 290
Associated LU table . . . 291

Selecting session parameters for the logon mode table . . . 292
Automatic logons . . . 300
Coding for automatic logon . . . 300
Operator commands for automatic logon . . . 301
Reallocation of autologon sessions . . . 301
Session management exits . . . 305
Session authorization . . . 306
Session accounting . . . 307
Session-level pacing . . . 307
Fixed session-level pacing . . . 308
Adaptive session-level pacing . . . 308
Setting initial pacing values . . . 310
Sample configurations . . . 314
Logon and logoff requests from dependent logical units . . . 321
Unformatted logon requests using mixed-case passwords . . . 322

Chapter 13. Network routing . . . 325
Network routing and resource location for APPN nodes . . . 325
Types of searches . . . 326
Minimizing broadcast searches . . . 329
Network search overhead caused by duplicate resource definitions . . . 332
Avoiding congestion . . . 332
Checkpointing of the TRS database and the directory database . . . 333
APPN Class of Service . . . 334
APPN network routing through a composite network node (CNN) . . . 341
Using the SAMAP table . . . 344
Network routing for subarea nodes . . . 348
Physical paths . . . 348
Logical paths . . . 350
How session traffic is assigned to a specific route . . . 352
How to plan routes in your network . . . 357
How VTAM handles network and subarea addressing . . . 360
Virtual route pacing . . . 365
Parallel sessions using parallel transmission groups . . . 368
Dynamic path update . . . 369

Chapter 14. Application programs . . . 373
Naming an application program . . . 373
Model application program definitions . . . 374
Overview . . . 375
Coding guidelines . . . 375
How VTAM finds the best match . . . 377
Example of using model application program definitions . . . 378
Resource state requirements . . . 380
Authorizing application facilities . . . 380
Passing and validating logon requests . . . 380
Overriding dial number digits for dial or token-ring connections . . . 380
Acquiring LU sessions . . . 381
Enabling parallel sessions . . . 381
Authorizing privileged paths . . . 381
Data compression . . . 381
Types of compression . . . 382
Implementing data compression . . . 382
Compression level negotiation . . . 383
Summary of data compression . . . 385
Security features . . . 386
Cryptography facility . . . 386
Message authentication . . . 396
SLU authentication . . . 397
VTAM application security . . . 397

Confidential data . . . 398
Logon mode parameters . . . 399
Using user variables (USERVAR) . . . 399
Application workload balancing with USERVAR . . . 400
Classes of USERVARs . . . 400
Types of USERVARs . . . 401
Processing USERVARs . . . 402
USERVAR propagation and routing . . . 403
Defining your network with USERVARs . . . 404
Dynamic USERVAR update session failure . . . 405
Generic resources function . . . 406
High availability using extended recovery facility . . . 407
Security features in an XRF environment . . . 407
LU 6.2 in an XRF Environment . . . 408
Persistent LU-LU sessions . . . 409
Single node persistent sessions . . . 410
VTAM common network services . . . 411
Cross-memory application programming interface (API) . . . 412
Allocating private storage . . . 412
Communicating with start-stop devices . . . 412
LU 6.2 application programs . . . 412
Enabling LU 6.2 support . . . 412
LU 6.2 sessions . . . 413
LU 6.2 session limits . . . 415
Managing LU 6.2 sessions with operator commands . . . 417
LU 6.2 security . . . 417
LU 6.2 sync point services . . . 419
Selective termination of idle LU 6.2 sessions . . . 419
Selective termination of idle network management sessions . . . 420
APING support . . . 421
High-performance data transfer (HPDT) . . . 424
Communications storage manager (CSM) . . . 425

Chapter 15. CMIP application programs . . . 427
VTAM topology agent CMIP application program . . . 427
Implementing CMIP services . . . 427
What the topology agent does . . . 428
How data flows between the topology manager and the topology agent . . . 429
Associations and using the directory definition file for CMIP services . . . 429
Controlling associations . . . 430
Determining security for associations . . . 430
Updating the directory definition file . . . 432

Chapter 16. Functions provided by VTAM in a sysplex . . . 435
Setting up the sysplex environment for VTAM and TCP/IP functions . . . 436
Sysplex subplexing . . . 436
Coupling facility structure attributes . . . 438
Determining the size of the coupling facility structure . . . 441
Sample CFRM coding . . . 441
Connecting to and allocating storage for coupling facility structures . . . 443
Structure rebuild . . . 443
Coupling facility duplexing . . . 444
Coupling facility storage shortages . . . 444
Dynamic altering of structures . . . 445
Dynamic definition of VTAM-to-VTAM connections . . . 445
Generic resources . . . 448
Generic resources requirements . . . 448
Generic resource mapping . . . 450
Partner LU mapping . . . 450
Initiating sessions with generic resource members . . . 454

Implementation considerations . . . 456
Coupling facility failures for generic resource configuration . . . 457
Removing a generic resource . . . 458
Routine maintenance for VTAM nodes . . . 464
Multinode persistent sessions . . . 464
Multinode persistent session configuration requirements . . . 467
Using multiple coupling facility structures for multinode persistent sessions . . . 470
Establishing multinode persistent sessions . . . 470
Coupling facility failures for multinode persistent session configuration . . . 474
Failure recovery processing . . . 475
MNPS planned and forced takeover processing . . . 477
What to do if recovery does not occur or complete . . . 481
TSO generic resources . . . 483
Sysplex-wide security associations . . . 484
Coupling facility failures for sysplex-wide security associations . . . 484
Failure of a TCP/IP stack . . . 484
Failure of a VTAM node . . . 484
Rebuild of the sysplex-wide security associations structure (EZBDVIPA) . . . 484
Disconnect from the EZBDVIPA structure . . . 485
Sysplexports . . . 485
Coupling facility failures for Sysplexports . . . 485
Failure of a TCP/IP stack . . . 485
Failure of a VTAM node . . . 485
Rebuild of the Sysplexports structure (EZBEPORT) . . . 486
Disconnect from the EZBEPORT structure . . . 486

Chapter 17. Implementing an APPN network . . . 487
Coding considerations for APPN resources . . . 487
Maximum APPN Locate size considerations . . . 489
Enabling control sessions . . . 489
CP-CP sessions between two VTAM nodes . . . 490
Defining adjacent APPN nodes . . . 490
Defining the logon mode for CP-CP sessions . . . 490
High-Performance Routing (HPR) . . . 491
What is High-Performance Routing? . . . 491
What is Rapid Transport Protocol? . . . 492
What is automatic network routing? . . . 493
How does HPR switch paths? . . . 494
HPR implementation overview . . . 495
Setting session paths using HPRNCPBF . . . 503

Chapter 18. Implementing a combined APPN and subarea network . . . 505
Start options defining a combined subarea and APPN environment . . . 508
SORDER start option . . . 509
SSEARCH start option . . . 509
CDRSCTI start option . . . 509
IOPURGE start option . . . 509
Dependent LUs . . . 510
Dependent LU server . . . 510
Dependent LU server support across subnetwork boundaries . . . 514
Defining CDRSCs . . . 515
Shared NCPs . . . 515
SSCP takeover . . . 516
Establishing and controlling sessions . . . 519
Controlling searches . . . 519
Using SORDER to control network search order . . . 519
Using SSEARCH to limit subarea network searches . . . 520
Using the CDRSCTI start option to reduce broadcast searches of APPN . . . 521
Using the DISJOINT operand to define disjoint subarea networks . . . 521
APPN and subarea Class of Service resolution . . . 523

Resolving logon mode names to subarea and APPN Classes of Service . . . 525
Defining APPN and subarea Classes of Service in logon mode tables . . . 527
Defining APPNTOSA and SATOAPPN class of service mapping tables . . . 527
Adding and moving nodes . . . 527

Chapter 19. Implementing a subarea network. . . . 529
Start options defining other domains . . . 530
Defining the location of cross-domain resource managers by coding adjacent SSCP tables . . . 530
Defining the location of cross-domain resource managers dynamically . . . 531
Specifying timeout values for locating cross-domain resources . . . 531
Identifying VTAMs in other domains (CDRMs) . . . 532
Identifying resources in other domains . . . 534
Dynamic definition of cross-domain resources . . . 534
Static definition of cross-domain resources. . . . 536
Model definition of cross-domain resources . . . 538
Adjacent SSCPs. . . . 541
CDRM owner verification for cross-domain resources . . . 548
Changing ownership of cross-domain resources . . . 549
Shadow resources . . . 549

Chapter 20. Connecting multiple subarea networks . . . 551
Defining a multiple-network environment . . . 551
SNI configurations . . . 554
Start options defining other networks . . . 555
Start options for gateway VTAMs . . . 555
GWSSCP start option for nongateway VTAMs . . . 555
Configuration lists for gateway VTAMs . . . 556
Connecting networks . . . 556
Defining a gateway VTAM . . . 557
Defining a gateway NCP . . . 558
Converting a non-SNI NCP to a gateway NCP . . . 559
Defining native networks . . . 560
Defining nonnative networks . . . 562
Dynamic network identification . . . 563
Defining cross-domain resource managers . . . 567
Defining cross-domain resources . . . 570
Session initiation request . . . 571
Name assumption . . . 572
Predefined cross-domain resources without network specification . . . 574
Predefined cross-domain resources with network specification . . . 575
Dynamic cross-domain resources . . . 576
Defining adjacent SSCPs. . . . 579
Types of adjacent SSCP tables . . . 579
Deciding whether to code adjacent SSCP tables . . . 579
Sample of adjacent SSCP tables for a multiple-network environment . . . 580
Request routing . . . 583
Dynamically defined CDRSCs and adjacent SSCP tables . . . 584
Alias name translation and adjacent SSCP tables . . . 584
Cross-network routing . . . 584
Network address structures . . . 584
Network naming conventions . . . 585
Controlling paths for interconnected networks . . . 586
Handling class of service tables . . . 588
Address translation . . . 590
Resource name translation . . . 592
Alias selection function of the session management exit routine . . . 593
NetView alias name translation facility . . . 593
Establishing and controlling SNA sessions . . . 600
Nonnative network type 2.1 connections . . . 600
Automatic logon . . . 601

Operating VTAM 602
Gateway NCP and VTAM interaction 602
Using the NetView program for network management 602
Application programs 603
Defining the NetView program 603

Chapter 21. Operating VTAM 605
Starting the domain 605
Configuration restart 606
Information recorded by configuration restart 609
Activating resources 610
Order of activation 611
Resources automatically activated by VTAM 612
Activating application programs 614
Monitoring the domain 615
Using the DISPLAY command 615
Monitoring I/O problems 615
Suppressing messages 615
Message flooding prevention 616
Other methods of controlling messages 616
Displaying and testing routes 617
Defining operator messages and commands 617
Multiple console support (MCS) in VTAM 618
Controlling the domain 618
Establishing and terminating sessions with operator commands 619
Dynamic table replacement 619
Deactivating resources 619
Order of deactivation 619
Automatic deactivation 620
Normal deactivation 620
Immediate deactivation 621
Forced deactivation 621
Forced reactivation 621
Halting VTAM 621
Canceling VTAM 623
Automatic operations 623
Program operators 623
CNM application programs 624
Collecting session awareness (SAW) data 625
Operating VTAM in a multiple-domain subarea network 626
Links and link stations 626
Discontiguous domains 630
Backing up resource owners 631
Sharing an NCP 632

Chapter 22. Operating with an NCP 635
Loading an NCP 635
NCP load process 635
Identifying the communication controller 636
Selecting a load station 637
Loading from the host or from external storage 638
Reloading an NCP 638
Scheduled automatic reloading of a communication controller 638
Renaming NCP load modules 641
Loading or purging a load module on a communication controller disk 643
Activating an NCP 643
Automatic activation of adjacent link stations 644
Reactivating a failed NCP 647
Activating links 647
Loading and activating an NCP over a switched link 649

Enabling communication controller channels for activation . . . 650
Monitoring an NCP . . . 651
Sending line errors to VTAM . . . 651
Displaying NCP storage . . . 652
Testing a nonswitched SDLC line . . . 652
Deactivating an NCP . . . 653
Deactivating links . . . 653
Dumping an NCP . . . 654
Selecting a dump station for a static dump of an NCP . . . 655
Selecting a dump file for an NCP dump . . . 656
Selecting the disposition of an NCP dump . . . 656
Other communication controller dumps . . . 657
Selecting a dump file for a MOSS or CSP dump . . . 657
IBM Communication Controller for Linux on System z considerations . . . 658
Defining and activating XCA-attached CCL NCPs . . . 659
Loading XCA-attached CCL NCPs . . . 659
Monitoring and managing XCA-attached CCL NCPs . . . 661
Dumping XCA-attached CCL NCPs . . . 661

Chapter 23. Tuning VTAM for your environment . . . 663
Introduction to tuning . . . 663
Tuning tools . . . 663
Estimating active sessions . . . 664
Common storage areas . . . 664
Buffer pools . . . 664
NCP slowdown . . . 664
Coattailing . . . 664
Tuning tools . . . 665
Monitoring common storage areas . . . 665
Analyzing slowdown . . . 667
Gathering tuning information with the performance monitor interface . . . 667
Gathering tuning statistics . . . 668
Analyzing tuning statistics . . . 681
Determining the amount of coattailing in your system . . . 682
Migrating from user-replaceable constants . . . 682
Estimating the number of active sessions . . . 682
EAS operand for application programs . . . 682
EAS operand for independent logical units . . . 683
Common storage areas . . . 683
Common service area limit . . . 683
Common service area 24-bit . . . 683
DISPLAY STORUSE pools . . . 684
Buffer pools . . . 691
Types of buffer pools . . . 691
Buffer pool allocation . . . 693
HOT I/O detection/termination . . . 700
Controlling NCP slowdown . . . 702
Maximizing coattailing . . . 702
Controlling outbound coattailing . . . 703
Controlling inbound coattailing . . . 704
Coattailing for SNA controllers . . . 705
Coattailing for channel-to-channel operations . . . 711
Session-level pacing tuning considerations . . . 713
Input/output buffers . . . 713
Application program data space . . . 713
CSM storage . . . 713

Appendix A. TSO/VTAM . . . 715
Defining TSO to VTAM . . . 715
Defining the TCAS application to VTAM . . . 716

Defining TSO/VTAM session parameters . . . 720
Defining an interpret table for compatible logons . . . 725
Defining TSO to MVS . . . 725
Writing a procedure to start TSO/VTAM time sharing . . . 725
Creating a TSOKEY00 PARMLIB member . . . 725
Defining TCAS program properties . . . 726
Implementing TSO/VTAM . . . 726
Translation tables . . . 726
Coding TSO/VTAM exit routines . . . 726
Security . . . 726
Performance . . . 727
3270 large screen considerations . . . 727
TSO considerations . . . 728
Multicultural support for TSO/VTAM end-user messages . . . 728
Operating VTAM under TSO . . . 729

Appendix B. Storage estimate worksheets . . . 731
General information . . . 732
APPN interchange node or network node . . . 733
APPN migration data host and end node . . . 735
Subarea data host . . . 736
Subarea communication management configuration . . . 736
APPL EAS storage estimates . . . 738

Appendix C. Communications storage manager . . . 741
CSM installation and definition . . . 742
Initializing CSM . . . 742
Monitoring CSM . . . 742
CSM problem diagnosis . . . 743

Appendix D. Logon manager . . . 745
How the logon manager operates . . . 745
Installing the logon manager . . . 746
Starting the logon manager . . . 746
Defining the logon manager . . . 746
Sample logon manager configuration . . . 747
Defining the logon manager and TPF applications to VTAM . . . 748
Defining the logon manager configuration . . . 749
Monitoring logon manager resources . . . 752
Halting the logon manager . . . 753

Appendix E. Cryptographic keys . . . 755
Filing SLU keys for single-domain cryptographic sessions . . . 755
Single-domain cryptographic sessions that use PCF/CUSP . . . 756
Single-domain cryptographic sessions that use ICSF/MVS . . . 756
Filing CDRM keys for cross-domain cryptographic sessions . . . 757
Cross-domain cryptographic sessions in which both hosts use PCF/CUSP . . . 757
Cross-domain cryptographic sessions in which both hosts use ICSF/MVS . . . 758
Cross-domain cryptographic sessions in which the hosts use different cryptographic services . . . 760
Changing the cryptographic capability of a logical unit . . . 762

Appendix F. Command lists: Dynamic configuration of channel-attached devices . . . 763
ISTDINFO: VTAM device information services . . . 763
Dependencies and restrictions . . . 763
Output variable . . . 763
Output tokens . . . 764
ISTDEFIN: VTAM device information services . . . 766
Dependencies and restrictions . . . 767
Output variable . . . 767
Output tokens . . . 767

Appendix G. Message translation using the MVS Message Service 769
Overview of MMS support 769
Internal translation 769
Selecting internal translation 769
Defining messages for internal translation 770
External translation 770
Selecting external translation 770
Defining messages for external translation 771
Skeleton file use 771

Appendix H. Forcing an APPN route in a VTAM network 773

Appendix I. Border node connection types 777

Appendix J. VTAM restricted materials 781

Appendix K. Architectural specifications 783

Appendix L. Accessibility 785

Notices 789
Policy for unsupported hardware 796
Trademarks 797

Bibliography 799

Index 803

Communicating your comments to IBM 825

Figures

1. Pacing flow – outbound pacing . . . 22
2. Pacing flow – inbound pacing . . . 22
3. Pacing flow – receiving early pacing response . . . 23
4. VTAM network . . . 27
5. Configuration list . . . 37
6. MPC connection between two VTAM network nodes . . . 54
7. Type 2.1 channel connection between a composite network node and a network node . . . 59
8. Leased connection between two composite network nodes . . . 60
9. Two network nodes connected using an IBM 3172 Nways Interconnect Controller . . . 61
10. VTAM attachment to a LAN—No meshed connection definitions . . . 63
11. VTAM attachment to a LAN—Meshed connection definitions provide optimal route calculation . . . 64
12. VTAM attachment to a connection network reduces required connection definitions (token ring) . . . 65
13. VTAM attachment to a connection network also enables optimal route calculation (token ring). . . . 66
14. Basic ATM configuration . . . 69
15. VTAM connection to the IBM Open Systems Adapter . . . 70
16. Definition of VTAM connection to the IBM Open Systems Adapter . . . 71
17. Port on the IBM Open Systems Adapter through which the ATM network is accessed. . . . 72
18. Definition of port on the IBM Open Systems Adapter through which the ATM network is accessed . . . 73
19. TGs that route data across the ATM network . . . 73
20. Definition of a TG over a PVC . . . 74
21. TG over an SVC . . . 75
22. Definition of a TG over an SVC . . . 77
23. Multiple nodes communicating across an ATM network. . . . 80
24. ATM configuration with a connection network . . . 81
25. Definitions in VTAMLST for the VTAM in HOST1 . . . 83
26. Definitions in VTAMLST for the VTAM in HOST2 . . . 84
27. Definitions in VTAMLST for the VTAM in HOST3 . . . 85
28. APPN subnetworks through APPN multiple network connectivity support . . . 90
29. VR-based TG between composite network nodes . . . 96
30. Multiple contiguous VR-based TGs . . . 97
31. VR-based TGs in a communication management configuration . . . 99
32. Parallel transmission groups in multiple domain environment with NCP . . . 105
33. Attaching domains using NCP-to-NCP token-ring connection . . . 111
34. XCA multiple domain configuration with Ethernet or Ethernet-type LAN . . . 113
35. XCA multiple domain configuration . . . 114
36. Channel-attached 3720 Communication Controller . . . 117
37. Parallel transmission groups between VTAM and NCP. . . . 119
38. Example NCP to VTAM data host connection . . . 121
39. Defining, loading, and activating a link-attached NCP . . . 128
40. Comparison between an EE network and a SNA network. . . . 136
41. How EE and EBN work together . . . 137
42. How ToS settings affect IP traffic . . . 141
43. Four types of EE connectivity . . . 143
44. VRN connectivity . . . 161
45. VTAM routing with an SATF . . . 162
46. VTAM routing meshed connections . . . 163
47. VTAM attachment to a connection network . . . 164
48. VTAM optimal route calculations . . . 165
49. Defining multiple EE VRNs. . . . 167
50. EE connection network reachability awareness in a mixed-release environment . . . 188
51. Connection network reachability example 1 . . . 191
52. Connection network reachability example 2 . . . 192
53. Global VRN with extended border nodes . . . 200
54. OSA-Express virtual MAC routing . . . 216
55. QDIO primary and secondary routing . . . 218

© Copyright IBM Corp. 1984, 2011

56. Example of communication management configuration . . . 222
57. Creating resources in dynamic switched major node . . . 228
58. Dynamic configuration of channel-attached device . . . 231
59. DR ADD operation . . . 247
60. DR DELETE operation . . . 248
61. DR MOVE operation . . . 249
62. Sample implicit dynamic reconfiguration . . . 250
63. Line backup through dynamic reconfiguration . . . 251
64. Nonnative network type 2.1 connection . . . 255
65. Local SNA devices . . . 257
66. XCA connection in a single-domain environment . . . 259
67. Major elements of PATH definition statements for switched major node . . . 265
68. Independent LU with multiple connections to VTAM . . . 280
69. Definition building for dynamically defined dependent LUs . . . 285
70. Macroinstructions for logon mode table . . . 294
71. How session parameters are obtained from a logon mode table . . . 299
72. Automatic logon to A50ACCTS application program . . . 300
73. Automatic logon reallocation . . . 303
74. Fixed-session pacing (one- and two-stage) . . . 308
75. Adaptive session pacing . . . 309
76. Pacing windows for SNA LUs . . . 312
77. Pacing windows for non-SNA LUs . . . 312
78. Same domain application program-to-application program session . . . 315
79. Same domain application program-to-local device session . . . 316
80. Application program-to-application program over APPN host-to-host connection . . . 317
81. Application program-to-application program over CTCA connection . . . 318
82. Application program-to-local SNA device over CTCA connection . . . 319
83. Application program-to-local SNA device over AHHC connection . . . 320
84. Application program-to-application program with VR from intermediate host-to-SLU host . . . 321
85. Routing example through a CNN node . . . 342
86. CNN route calculation example . . . 343
87. Composite network node route calculation example . . . 343
88. Composite network node route BIND reroute example . . . 344
89. Typical CNN session path . . . 345
90. APPN view of CNN session path . . . 345
91. SAMAP session routing concept . . . 346
92. SAMAP example . . . 347
93. One explicit route in each direction . . . 350
94. Two explicit routes in each direction . . . 350
95. Virtual route and explicit route associations and transmission priority . . . 352
96. Class of Service substitution . . . 355
97. Class of Service hierarchy . . . 356
98. Element and subarea address incompatibility in multiple-domain environment . . . 362
99. Local flow control . . . 366
100. Parallel sessions using parallel transmission groups . . . 368
101. Sample single-domain network . . . 370
102. Data compression yield . . . 385
103. Encryption facility specifications . . . 389
104. Encryption facility in an APPN environment . . . 393
105. Encryption facility in multiple-network environment . . . 394
106. Sample extended recovery facility network . . . 402
107. Application program backup using persistent LU-LU sessions . . . 410
108. Example of flows between client and server for DISPLAY APING command . . . 423
109. Generic resource mapping . . . 450
110. Partner LU mapping . . . 452
111. Session establishment with generic resource members . . . 453
112. Application program backup using multinode persistent sessions . . . 466
113. Multinode persistent session network example . . . 469
114. Session routes . . . 472
115. Path switch processing . . . 473
116. HPR=(RTP,ANR) and TG capabilities . . . 496

117. HPR over composite network nodes 497
118. HPR Over APPN host-to-host channel connections 498
119. Multiple HPR routes between HPR session endpoints 499
120. Using VR-based TGs for non-HPR endpoints 500
121. Session involving HPR and APPN routes 501
122. Interchange node utilizing HPR routing between subarea and APPN 502
123. Sessions traversing APPN and subarea networks. 503
124. Example of communication management configuration 505
125. Communication management configuration after conversion. 506
126. VTAM functioning as a dependent LU server 511
127. Switched major node for a dependent LU server. 512
128. DLUS located in different APPN subnetwork than DLUR or PLU 514
129. PLU exists in or through a subarea network 515
130. SSCP takeover when adjacent CP does not support CP name change 516
131. SSCP takeover when adjacent CP is another composite network node. 518
132. Disjoint hosts 522
133. Hosts with subarea connection. 523
134. LOGMODE resolution example 523
135. Class of Service resolution at multiple nodes 526
136. Multiple-domain network 529
137. Major and minor nodes in multiple-domain environment 533
138. Example of adjacent SSCP table connection 544
139. Multiple-network environment. 553
140. Backup of a gateway NCP 566
141. Multiple-network configuration: CDRM major nodes 569
142. Example of three interconnected networks 572
143. Example of two interconnected networks 574
144. Multiple-network configuration 580
145. Multiple-network paths 586
146. COS resolution in a multiple-network environment 589
147. COS tables and routing in an SNI back-to-back configuration 590
148. Address translation 591
149. Example of name translation 597
150. Nonnative network type 2.1 connections 601
151. Restoring resource definitions with configuration restart and NODELST files 607
152. Effects of NCP deactivation on cross-subarea links and link stations 627
153. NCP generation. 636
154. Activating NCPs and link stations: Through another NCP 645
155. Activating NCPs and link stations: Through host link station 645
156. Automatic activation of link stations 646
157. Loading and activating NCP using switched link 649
158. Loading over a channel 651
159. Example of an MPC channel-to-channel, QDIO, and Hipersockets tuning statistics report 674
160. Buffer pool after initial allocation and after one expansion 695
161. How VTAM uses input buffers. 699
162. How VTAM uses output buffers 700
163. Effect of DELAY time on coattailing 706
164. General I/O buffer format 709
165. Using multiple I/O buffers to transfer single message 710
166. Multiple-buffer considerations 710
167. Sample Logon Manager Configuration 747
168. Cryptography in multiple-domain environment (Both hosts use PCF/CUSP) 758
169. Cryptography in multiple-domain environment (Both hosts use ICSF/MVS) 760
170. Cryptography in multiple-domain environment (Hosts use different cryptographic services) 762
171. Sample network showing default route 773
172. Sample network using default for UPARM1 774
173. Sample network using TG profile on some links 775

Tables

1. LU types . . . 16
2. Pacing types . . . 21
3. MVS static system symbols that can be used in VTAM . . . 41
4. HPDT Packing - packing buffer size, CSM pools size, and waste per packing buffer . . . 58
5. Comparison between static and dynamic definitions . . . 142
6. SNA priorities and corresponding port numbers and default ToS values . . . 183
7. Connection conditions and results . . . 193
8. Troubleshooting EE problems . . . 202
9. OSA-Express support . . . 211
10. Definitions for dynamically configured devices . . . 233
11. Dynamic reconfiguration operations for valid major nodes . . . 237
12. Rules for multiple definition of resources . . . 244
13. Example of model name table . . . 290
14. Example of associated LU table . . . 292
15. How session parameters are identified . . . 294
16. Correspondence of methods to letters . . . 311
17. Same domain application program-to-application program session—PLU to SLU flow . . . 315
18. Same domain application program-to-application program Session—SLU to PLU Flow . . . 315
19. Same domain application program-to-local device session—PLU to SLU flow . . . 316
20. Same domain application program-to-local device session—SLU to PLU flow . . . 316
21. Application program-to-application program over APPN host-to-host connection—PLU to SLU flow 317
22. Application program-to-application program over APPN host-to-host connection—SLU to PLU flow 317
23. Application program-to-application program over CTCA connection—PLU to SLU flow . . . 318
24. Application program-to-application program over CTCA connection—SLU to PLU flow . . . 318
25. Application program to local SNA device over CTCA connection—PLU to SLU flow . . . 319
26. Application program-to-local SNA device over CTCA connection—SLU to PLU flow . . . 319
27. Application program-to-local SNA device over AHHC connection—PLU to the SLU . . . 320
28. Application program-to-local SNA device over AHHC connection—SLU to PLU flow . . . 320
29. Application program-to-application program with VR from intermediate host-to-SLU host—PLU to SLU flow . . . 321
30. Application program-to-application program with VR from intermediate host-to-SLU host—SLU to PLU flow . . . 321
31. Default registration values for resources . . . 330
32. COSAPPN #CONNECT class of service LINEROW values . . . 335
33. ISTACST2 #CONNECT class of service LINEROW values . . . 336
34. ISTACST3 #CONNECT class of service LINEROW values . . . 336
35. ISTINCLM APPNCOS values based on LOGMODE operand values . . . 338
36. Default TG characteristics . . . 340
37. #CONNECT Class of Service LINEROW values . . . 340
38. Adjacent node subarea requirements for multiple-domain environment . . . 363
39. Endpoint node subarea requirements for multiple-domain environment . . . 363
40. Sample model application program names . . . 376
41. Compression values for example of data compression yield . . . 385
42. DES-TDES24 encryption options . . . 389
43. Resource verification reduction matrix . . . 415
44. Structure attributes used to compute structure size . . . 439
45. Structure attributes defined by VTAM . . . 440
46. VTAM to VTAM connection example . . . 447
47. Start options and node type relationship . . . 487
48. Network resource list example . . . 599
49. Loading a 3745 from VTAM . . . 660
50. Loading a CCL or 3745 from VTAM . . . 661
51. Record format for SNA controller . . . 669
52. Record format for channel-to-channel adapters . . . 670
53. Record format for multipath channel connections (XCF) . . . 672

© Copyright IBM Corp. 1984, 2011

54. Record format for multipath channel connections (channel) 675
55. Record format for TCP connections 679
56. DISPLAY STORUSE pools 684
57. VTAM buffer pools 691
58. Number of buffers per page 697
59. I/O buffer size and number of buffers per page 698
60. Coding the device-specific hexadecimal data of PSERVIC 722
61. Worksheet for APPN interchange node or network node storage 733
62. Summary of worksheet, APPN interchange node or network node storage 734
63. Worksheet for APPN migration data host and end node 735
64. Summary of APPN migration data host and end node 735
65. Worksheet for subarea data host 736
66. Summary of subarea data host 736
67. Worksheet for subarea communication management configuration 736
68. Summary of subarea communication management configuration 738
69. Buffer pools in CSM 741
70. Connection type for selected VTAM and partner node combinations 777
71. User replaceable or modifiable modules. 781
72. VTAM message modules. 781

About this document

This document provides an understanding of the major concepts involved in implementing a VTAM® network and describes how to:

- Install VTAM
- Define network resources to VTAM
- Replace user tables and exit routines
- Tune VTAM for a specific environment

It also provides guidance for using VTAM resource definition statements and macroinstructions. Use this document when installing, upgrading, or otherwise changing a VTAM network.

The information in this document includes descriptions of support for both IPv4 and IPv6 networking protocols. Unless explicitly noted, descriptions of IP protocol support concern IPv4. IPv6 support is qualified within the text.

The *z/OS Communications Server: SNA Resource Definition Reference* contains a brief description and shows the exact coding format of each VTAM definition statement, start option, and user table. It serves as a reference document for system programmers who are already familiar with the major concepts involved in implementing a VTAM network.

Who should read this document

System programmers who are familiar with the basic concepts of telecommunication, Systems Network Architecture (SNA), and VTAM.

For an overview of the new functions in VTAM, refer to the *z/OS Communications Server: New Function Summary*.

How this document is organized

The *z/OS Communications Server: SNA Network Implementation Guide* is divided into the following sections:

The first section describes subarea and Advanced Peer-to-Peer Networking (APPN) concepts for VTAM.

The second section presents an overview of VTAM implementation issues common to all VTAM networks.

The third section describes the following:

- Various ways of connecting VTAM as an APPN node in the network
- Enterprise Extender connections
- Types of subarea connections to VTAM
- VTAM as a data host
- Methods for VTAM resources to be defined dynamically
- Details about how to define PUs to VTAM
- Details about how to define independent and dependent LUs to VTAM

© Copyright IBM Corp. 1984, 2011

The fourth section describes the following:

- Major concepts involved in establishing sessions.
- Major concepts involved in routing data through a VTAM network; both routing considerations for APPN and subarea nodes are covered.

The fifth section describes the following:

- Concepts and functions available for VTAM application programs. For detailed information on implementation of VTAM application programs, refer to *z/OS Communications Server: SNA Programming* and *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.
- Concepts involved in implementing a VTAM application using the Common Management Information Protocol (CMIP). Refer to the *z/OS Communications Server: CMIP Services and Topology Agent Guide* for detailed information on implementing a CMIP application program.

The sixth section describes how to incorporate VTAM functions and enhancements used in a sysplex environment.

The seventh section presents the major concepts in implementing an APPN network and describes the following:

- How to implement VTAM as an APPN node in the network
- Considerations for implementing a combined APPN and subarea network

The eight section presents the major concepts involved in implementing a subarea network and describes the following:

- Major concepts in implementing a basic subarea network environment
- Major concepts in implementing a multiple-network environment using the SNA network interconnection (SNI)

The ninth section describes VTAM operator control functions including the basic tasks available to the VTAM operator for controlling the VTAM domain and how to control a domain that includes NCP subarea nodes.

The tenth section describes how you can adjust VTAM to provide optimal service for your environment.

The appendixes provide additional information for this document.

How to use this document

To use this document, you should be familiar with the basic concepts of telecommunication, SNA, and VTAM.

Determining whether a publication is current

As needed, IBM updates its publications with new and changed information. For a given publication, updates to the hardcopy and associated BookManager® softcopy are usually available at the same time. Sometimes, however, the updates to hardcopy and softcopy are available at different times. The following information describes how to determine if you are looking at the most current copy of a publication:

- At the end of a publication's order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level

is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.

- If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.
- To compare softcopy publications, you can check the last two characters of the publication's file name (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.

How to contact IBM service

For immediate assistance, visit this website: http://www.software.ibm.com/network/commserver/support/

Most problems can be resolved at this website, where you can submit questions and problem reports electronically, as well as access a variety of diagnosis information.

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-IBM-SERV). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

If you would like to provide feedback on this publication, see "Communicating your comments to IBM" on page 825.

Conventions and terminology that are used in this document

Commands in this book that can be used in both TSO and z/OS® UNIX environments use the following conventions:

- When describing how to use the command in a TSO environment, the command is presented in uppercase (for example, NETSTAT).
- When describing how to use the command in a z/OS UNIX environment, the command is presented in bold lowercase (for example, **netstat**).
- When referring to the command in a general way in text, the command is presented with an initial capital letter (for example, Netstat).

All the exit routines described in this document are *installation-wide exit routines*. The installation-wide exit routines also called installation-wide exits, exit routines, and exits throughout this document.

The TPF logon manager, although included with VTAM, is an application program; therefore, the logon manager is documented separately from VTAM.

Samples used in this book might not be updated for each release. Evaluate a sample carefully before applying it to your system.

For definitions of the terms and abbreviations that are used in this document, you can view the latest IBM terminology at the IBM Terminology website.

Clarification of notes

Information traditionally qualified as Notes is further qualified as follows:

Note Supplemental detail

Tip Offers shortcuts or alternative ways of performing an action; a hint

Guideline
Customary way to perform a procedure

Rule Something you must do; limitations on your actions

Restriction
Indicates certain conditions are not supported; limitations on a product or facility

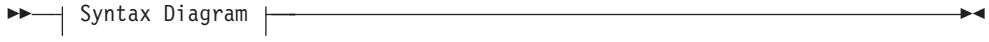
Requirement
Dependencies, prerequisites

Result Indicates the outcome

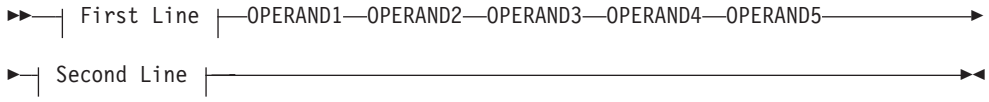
How to read a syntax diagram

This section describes how to read the syntax diagrams used in this book.

- Read the diagrams from left-to-right, top-to-bottom, following the main path line. Each diagram begins on the left with double arrowheads (►►) and ends on the right with two arrowheads facing each other (►◄).



- If a diagram is longer than one line, the first line ends with a single arrowhead (►) and the second line begins with a single arrowhead.

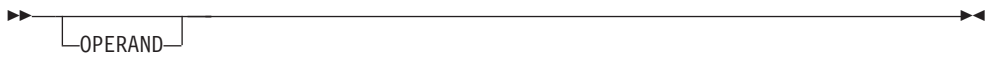


- Required operands and values appear on the main path line.

  You must code required operands and values.

  If there is more than one mutually exclusive required operand or value to choose from, they are stacked vertically in alphanumeric order.

- Optional operands and values appear below the main path line.

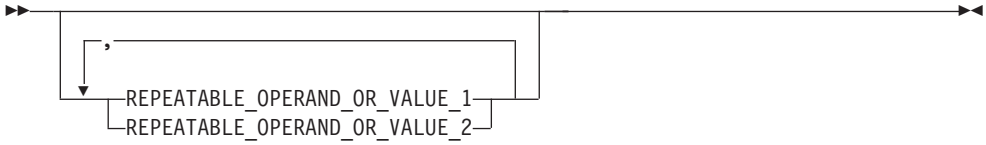


You can choose not to code optional operands and values.

If there is more than one mutually exclusive optional operand or value to choose from, they are stacked vertically in alphanumeric order below the main path line.

- An arrow returning to the left above an operand or value on the main path line means that the operand or value can be repeated. The comma means that each operand or value must be separated from the next by a comma.

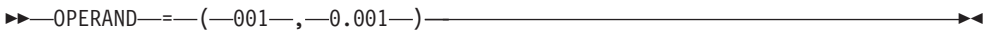
- An arrow returning to the left above a group of operands or values means more than one can be selected, or a single one can be repeated.



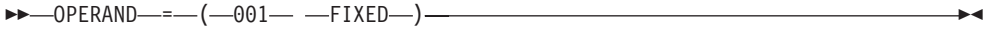
- A word in all uppercase is an operand or value you must spell exactly as shown. In this example, you must code ***OPERAND***.

  Note: VTAM and IP commands are not case sensitive. You can code them in uppercase or lowercase. If the operand is shown in both uppercase and lowercase, the uppercase portion is the abbreviation (for example, OPERand).

  If an operand or value can be abbreviated, the abbreviation is discussed in the text associated with the syntax diagram.
- If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code ***OPERAND=(001,0.001)***.



- If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code ***OPERAND=(001 FIXED)***.



- Default operands and values appear above the main path line. VTAM uses the default if you omit the operand entirely.

- A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

- References to syntax notes appear as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

Notes:

1 An example of a syntax note.

- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

Syntax Fragment:

Prerequisite and related information

z/OS Communications Server function is described in the z/OS Communications Server library. Descriptions of those documents are listed in "Bibliography" on page 799, in the back of this document.

Required information

Before using this product, you should be familiar with TCP/IP, VTAM, MVS™, and UNIX System Services.

Softcopy information

Softcopy publications are available in the following collections.

Titles	Order Number	Description
z/OS V1R13 and Software Products DVD Collection	SK3T-4271	This collection includes the libraries of z/OS (the element and feature libraries) and the libraries for z/OS software products in both BookManager format and PDF files. This collection combines SK3T-4269 and SK3T-4270.

Titles	Order Number	Description
IBM System z Redbooks Collection	SK3T-7876	The Redbooks® selected for this CD series are taken from the IBM® Redbooks inventory of over 800 books. All the Redbooks that are of interest to the zSeries® platform professional are identified by their authors and are included in this collection. The zSeries subject areas range from e-business application development and enablement to hardware, networking, Linux, solutions, security, parallel sysplex, and many others.

Other documents

For information about z/OS products, refer to *z/OS Information Roadmap* (SA22-7500). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, as well as describing each z/OS publication.

Relevant RFCs are listed in an appendix of the IP documents. Architectural specifications for the SNA protocol are listed in an appendix of the SNA documents.

The following table lists documents that might be helpful to readers.

Title	Number
DNS and BIND, Fifth Edition, O'Reilly Media, 2006	ISBN 13: 978-0596100575
Routing in the Internet , Second Edition, Christian Huitema (Prentice Hall 1999)	ISBN 13: 978-0130226471
sendmail, Fourth Edition, Bryan Costales, Claus Assmann, George Jansen, and Gregory Shapiro, O'Reilly Media, 2007	ISBN 13: 978-0596510299
SNA Formats	GA27-3136
TCP/IP Illustrated, Volume 1: The Protocols, W. Richard Stevens, Addison-Wesley Professional, 1994	ISBN 13: 978-0201633467
TCP/IP Illustrated, Volume 2: The Implementation, Gary R. Wright and W. Richard Stevens, Addison-Wesley Professional, 1995	ISBN 13: 978-0201633542
TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols, W. Richard Stevens, Addison-Wesley Professional, 1996	ISBN 13: 978-0201634952
TCP/IP Tutorial and Technical Overview	GG24-3376
Understanding LDAP	SG24-4986
z/OS Cryptographic Services System SSL Programming	*SC24-5901*
z/OS Integrated Security Services LDAP Server Administration and Use	*SC24-5923*
z/OS JES2 Initialization and Tuning Guide	*SA22-7532*
z/OS Problem Management	*G325-2564*
z/OS MVS Diagnosis: Reference	*GA22-7588*
z/OS MVS Diagnosis: Tools and Service Aids	*GA22-7589*
z/OS MVS Using the Subsystem Interface	*SA22-7642*
z/OS Program Directory	*GI10-0670*
z/OS UNIX System Services Command Reference	*SA22-7802*
z/OS UNIX System Services Planning	*GA22-7800*
z/OS UNIX System Services Programming: Assembler Callable Services Reference	*SA22-7803*

Title	Number
z/OS UNIX System Services User's Guide	*SA22-7801*
z/OS XL C/C++ Run-Time Library Reference	*SA22-7821*
zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference	*SA22-7935*

Redbooks

The following Redbooks might help you as you implement z/OS Communications Server.

Title	Number
IBM z/OS V1R12 Communications Server TCP/IP Implementation, Volume 1: Base Functions, Connectivity, and Routing	SG24-7896
IBM z/OS V1R12 Communications Server TCP/IP Implementation, Volume 2: Standard Applications	SG24-7897
IBM z/OS V1R12 Communications Server TCP/IP Implementation, Volume 3: High Availability, Scalability, and Performance	SG24-7898
IBM z/OS V1R12 Communications Server TCP/IP Implementation, Volume 4: Security and Policy-Based Networking	SG24-7899
IBM Communication Controller Migration Guide	SG24-6298
IP Network Design Guide	SG24-2580
Managing OS/390 TCP/IP with SNMP	SG24-5866
Migrating Subarea Networks to an IP Infrastructure Using Enterprise Extender	SG24-5957
SecureWay™ Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements	SG24–5631
SNA and TCP/IP Integration	SG24-5291
TCP/IP in a Sysplex	SG24-5235
TCP/IP Tutorial and Technical Overview	GG24-3376
Threadsafe Considerations for CICS	SG24-6351

Where to find related information on the Internet

z/OS

This site provides information about z/OS Communications Server release availability, migration information, downloads, and links to information about z/OS technology

http://www.ibm.com/systems/z/os/zos/

z/OS Internet Library

Use this site to view and download z/OS Communications Server documentation

www.ibm.com/systems/z/os/zos/bkserv/

IBM Communications Server product

The primary home page for information about z/OS Communications Server

http://www.software.ibm.com/network/commserver/

IBM Communications Server product support

Use this site to submit and track problems and search the z/OS Communications Server knowledge base for Technotes, FAQs, white papers, and other z/OS Communications Server information

http://www.software.ibm.com/network/commserver/support/

IBM Communications Server performance information

This site contains links to the most recent Communications Server performance reports.

http://www.ibm.com/support/docview.wss?uid=swg27005524

IBM Systems Center publications

Use this site to view and order Redbooks, Redpapers, and Technotes

http://www.redbooks.ibm.com/

IBM Systems Center flashes

Search the Technical Sales Library for Techdocs (including Flashes, presentations, Technotes, FAQs, white papers, Customer Support Plans, and Skills Transfer information)

http://www.ibm.com/support/techdocs/atsmastr.nsf

RFCs

Search for and view Request for Comments documents in this section of the Internet Engineering Task Force website, with links to the RFC repository and the IETF Working Groups web page

http://www.ietf.org/rfc.html

Internet drafts

View Internet-Drafts, which are working documents of the Internet Engineering Task Force (IETF) and other groups, in this section of the Internet Engineering Task Force website

http://www.ietf.org/ID.html

Information about web addresses can also be found in information APAR II11334.

Note: Any pointers in this publication to websites are provided for convenience only and do not in any manner serve as an endorsement of these websites.

DNS websites

For more information about DNS, see the following USENET news groups and mailing addresses:

USENET news groups
comp.protocols.dns.bind

BIND mailing lists
https://lists.isc.org/mailman/listinfo

BIND Users

- Subscribe by sending mail to bind-users-request@isc.org.
- Submit questions or answers to this forum by sending mail to bind-users@isc.org.

BIND 9 Users (This list might not be maintained indefinitely.)

- Subscribe by sending mail to bind9-users-request@isc.org.
- Submit questions or answers to this forum by sending mail to bind9-users@isc.org.

The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS

To access the z/OS Basic Skills Information Center, open your web browser to the following website, which is available to all users (no login required): http://publib.boulder.ibm.com/infocenter/zoslnctr/v1r7/index.jsp

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this document or any other z/OS Communications Server documentation, do one of the following:

- Go to the z/OS contact page at http://www.ibm.com/systems/z/os/zos/webqs.html. You can enter and submit your comments in the form provided at this website.
- Send your comments by email to comsvrcf@us.ibm.com. Be sure to include the name of the document, the part number of the document, the version of z/OS Communications Server, and, if applicable, the specific location of the text that you are commenting on (for example, a section number, a page number or a table number).

Summary of changes

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Changes made in z/OS Communications Server Version 1 Release 13

This document contains information previously presented in *z/OS Communications Server: SNA Network Implementation Guide*, SC31-8777-10, which supports z/OS Version 1 Release 12.

New information:

- Performance improvements for Enterprise Extender traffic, see "Hardware requirements" on page 135.
- Intrusion detection services support for Enterprise Extender, see the following topics:
  - "IP filtering" on page 174
  - "IDS for Enterprise Extender" on page 174

Changed information:

- HiperSockets™ optimization for intraensemble data networks, see the following topics:
  - "Defining an OSA-Express device to z/OS Communications Server using QDIO" on page 213
  - "Resources automatically activated by VTAM" on page 612

Deleted information:

- Information describing how TSO/VTAM differs from TSO through Telecommunications Access Method (TCAM) has been deleted.

Changes made in z/OS Communications Server Version 1 Release 12

This document contains information previously presented in *z/OS Communications Server: SNA Network Implementation Guide*, SC31-8777-09, which supports z/OS Version 1 Release 11.

New information:

- Enterprise Extender connection health verification, see "Step 3: Prepare your VTAM definitions" on page 148.
- Support for z/OS Communications Server for intraensemble networks, see the following topics:
  - "OSA-Express overview" on page 209
  - "Defining an OSA-Express device to z/OS Communications Server using QDIO" on page 213

Changed information:

- Multipath control for Enterprise Extender, see the following topics:
  - "IP multipath considerations" on page 140

© Copyright IBM Corp. 1984, 2011

- "Troubleshooting EE problems" on page 202
- Enhancements to topology database diagnostics, see "Verifying activation" on page 154.
- Performance improvements for Sysplex Distributor connection routing, see the following tables:
  - Table 51 on page 669
  - Table 52 on page 670
  - Table 53 on page 672
  - Table 54 on page 675
  - Table 55 on page 679

Changes made in z/OS Communications Server Version 1 Release 11

This document contains information previously presented in *z/OS Communications Server: SNA Network Implementation Guide*, SC31-8777-08, which supports z/OS Version 1 Release 10.

New information:

- APPN topology database update enhancements, see "Topology database" on page 18.
- HPR performance enhancements, see the following topics:
  - "Step 3: Prepare your VTAM definitions" on page 148
  - "When does the EE connection go away?" on page 177
  - "Running EE in constrained or virtualized environments" on page 194.
- Enterprise Extender IPSec performance improvement, see "IP security (IPSec)" on page 172.
- Display potential model application name, see "How VTAM finds the best match" on page 377.

Chapter 1. Post-installation considerations for z/OS Communications Server

This section describes some post-installation considerations for Communications Server under the z/OS operating system.

Defining Communications Server SNA to z/OS

To define z/OS Communications Server, do the following:

1. Change the message routing codes by coding a system user exit routine (if there is multiple console support).

   Message routing codes determine the console at which messages will appear. If the routing codes provided for z/OS Communications Server messages do not meet your needs, you can change the routing codes used on the messages by coding a system user exit routine (if there is multiple console support). The exit routine receives control before messages are routed so it can examine the messages' routing codes (and descriptor codes) and change them. The system uses the modified routing codes to route these messages. To change routing codes, do the following:

   a. Prepare the write-to-operator/write-to-operator-with-response (WTO/WTOR) exit routine, and add it to the control program. The WTO/WTOR exit routine can be inserted into the resident portion (communications task) of the control program either before or after system generation.
   b. Refer to *z/OS Communications Server: SNA Messages* for the message routing codes, and decide which new routing codes you want to assign to each message.

2. Define channel-attached devices.

   If you are adding channel-attached devices, you can define these devices using the Hardware Configuration Definition (HCD) to dynamically add the devices.

3. Determine the ECSA value.

   The ECSA value is defined by the second value on the CSA parameter in member IEASYS*xx*. It is recommended that you examine the ECSA value to ensure that it is adequate for z/OS Communications Server. You can determine the ECSA value by reviewing Appendix B, "Storage estimate worksheets," on page 731.

Using automatic restart manager

Automatic restart manager is a z/OS function that can automatically restart z/OS Communications Server after an abnormal end (abend).

During initialization, z/OS Communications Server automatically registers with the automatic restart manager, using the following options:

- REQUEST=REGISTER
- ELEMENT=NET@*cp_name*
- EVENTEXIT=NO_EVENTEXIT
- STARTTXT=NO_STARTTXT
- ELEMTYPE=SYSVTAM

© Copyright IBM Corp. 1984, 2011

- TERMTYPE=ELEMTERM

Note: The *cp_name* is the same name as that used on the SSCPNAME start option.

For more information about automatic restart manager, refer to *z/OS MVS Setting Up a Sysplex*.

Starting z/OS Communications Server SNA

You should code a z/OS Communications Server start procedure and save it in SYS1.PROCLIB. The system operator specifies the procedure when starting z/OS Communications Server.

The start procedure is called NET. The name NET is not required but is strongly recommended for consistency in entering the z/OS Communications Server operator commands and to reduce the operator's chances of making a syntax error. The procedure name you specify must be the first operand on the START, and MODIFY operator commands. For DISPLAY, HALT, and VARY the procedure name is always NET.

Following is an example of job control statements for a typical start procedure.

```
//NET      PROC
//VTAM     EXEC PGM=ISTINM01,TIME=1440,REGION=4096K,
//STEPLIB  DD   DSN=SYS1.SSPLIB,DISP=SHR
//FFSTLIB  DD   DSN=SYS1.VTAMLIB,DISP=SHR
//VTAMLST  DD   DSN=SYS1.VTAMLST,DISP=SHR
//         DD   DSN=USER1.AUTO.VTAMLST,DISP=SHR
//SISTCLIB DD   DSN=SYS1.SISTCLIB,DISP=SHR
//AUTOCKPT DD   DSN=VSAM.AUTOCKPT,AMP=AMORG,DISP=OLD
//VTAMLIB  DD   DSN=SYS1.VTAMLIB,DISP=SHR
//NCPDUMP  DD   DSN=SYS1.NCPDUMP,DISP=SHR
//* DATA SETS FOR APPN DATABASE CHECKPOINTING
//DSDB1    DD   DSN=SYS1.DSDB1,DISP=SHR
//DSDB2    DD   DSN=SYS1.DSDB2,DISP=SHR
//DSDBCTRL DD   DSN=SYS1.DSDBCTRL,DISP=SHR
//TRSDB    DD   DSN=SYS1.TRSDB,DISP=SHR
//* DATA SETS 3720, 3725, AND 3745 DUMPS
//LDRIOTAB DD   DSN=SYS1.LDRIOTAB,DISP=(SHR,PASS,KEEP)
//CSPDUMP  DD   DSN=SYS1.CDUMP,DISP=SHR
//MOSSDUMP DD   DSN=SYS1.MDUMP,DISP=SHR
//NCPLOAD  DD   DSN=SYS1.NCPLOAD,DISP=SHR
//* NODELST DATA SET
//NODLST1  DD   DSN=VSAM.NODLST1,AMP=AMORG,DISP=OLD
//* ALTERNATE NODELST DATA SET
//NODLST2  DD   DSN=VSAM.NODLST2,AMP=AMORG,DISP=OLD
//* RESTART DATA SET
//* CMIP services Data Sets
//ISTCMIP  DD   DSN=SYS1.SISTCMIP,DISP=SHR
//ACYGDMO  DD   DSN=SYS1.SISTGDMO(ACYGDMO),DISP=SHR
//ISTASN1  DD   DSN=SYS1.SISTASN1,DISP=SHR
```

Note: On the EXEC statement, ISTINM01 is the main z/OS Communications Server initialization module name. Code PGM=ISTINM01.

The previous example is based on the following assumptions:

- The node is APPN capable.
- A communication controller is in the network.
- The generated NCP and RRT modules for the communication controller reside in data set SYS1.NCPLOAD (NCPLOAD DD statement). The NCP source is in SYS1.VTAMLST.

- The SSP modules needed to load and dump the communication controllers are in SYS1.SSPLIB (STEPLIB DD statement).
- A dump data set is needed for the communication controller (NCPDUMP DD statement).
- The dynamic configuration of channel-attached devices facility (dynamic I/O) is being used (VTAMLST DD and AUTOCKPT DD statements).
- A dump data set is needed for the communication scanner processor (CSP) (CSPDUMP DD statement). For 3380 DASD, this data set should be allocated with at least 7 cylinders using a block size of 512.
- A dump data set is needed for the maintenance and operator subsystem (MOSS) (MOSSDUMP DD statement). For 3380 DASD, this data set should be allocated with at least 10 cylinders using a block size of 512.
- Two NODELST data sets have been defined and can be used by z/OS Communications Server (NODEDS1 DD and NODEDS2 DD statement).
- The following CMIP services have been defined (these data sets are required to enable CMIP services and the z/OS Communications Server topology agent):
  - SYS1.SISTCMIP
  - SYS1.SISTGDMO(ACYGDMO)
  - SYS1.SISTASN1

  The directory definition file in the SYS1.SISTCMIP data set can be updated while z/OS Communications Server is running, but CMIP services is aware of these changes only when:
  - The MODIFY TABLE command is issued.
  - CMIP services is restarted by one of the following methods:
    - If CMIP services is active, stop CMIP services by issuing the MODIFY VTAMOPTS command with the OSIMGMT=NO start option and then restart CMIP services by issuing the MODIFY VTAMOPTS command with the OSIMGMT=YES start option.
    - Restart z/OS Communications Server with the OSIMGMT=YES start option.

    It is recommended that you keep backup copies of both the original directory definition file (or the last directory definition file that loaded without error) and the edited version of the file. When you load the edited file, z/OS Communications Server writes over the existing version of the directory definition file. There is no way to display the contents of the file being used by CMIP services, because the file is read into an internal data structure.

    If the edited file has a syntax error, z/OS Communications Server does not use it. Message IST1444I is issued to indicate what is wrong with the file. z/OS Communications Server continues to use the last correct file that it read.

    With your backup copies, you can correct the syntax error in the edited file by comparing it to the previous file that loaded without error.

    Note: The member name, ACYDDF, must not be changed.
- Space has been allocated for all data sets, and they have been cataloged.

Chapter 2. VTAM networking concepts

This topic describes Virtual Telecommunications Access Method (VTAM) concepts for a subarea and Advanced Peer-to-Peer Networking (APPN) environments. VTAM is an application program that is based on System Network Architecture (SNA). This topic includes:

- VTAM functions
- VTAM nodes
- Network accessible units
- Network control sessions
- End-user sessions
- How VTAM locates resources
- Route selection
- Addressing
- Controlling network data flow using pacing

VTAM functions

VTAM performs a number of tasks in a network. For example, VTAM:

- Monitors and controls the activation and connection of resources
- Establishes connections and manages the flow and pacing of sessions
- Provides application programming interfaces (for example, an APPC API for LU 6.2 programming) that allow access to the network by user-written application programs and IBM-provided subsystems
- Provides interactive terminal support for Time Sharing Option (TSO) using Multiple Virtual Storage (MVS)
- Provides support for both locally and remotely attached resources

VTAM nodes

VTAM can be configured as one of the following types of nodes:

- APPN function only
- Subarea function only
- APPN and subarea function

The type you choose depends on the level of function required for the node and whether the node needs to communicate with other subarea nodes, other APPN nodes, or both.

Nodes with APPN function only

A VTAM APPN node is an SNA type 2.1 node that functions in a peer-to-peer environment. APPN nodes use APPN or low entry network (LEN) connections to communicate with other nodes in an APPN network.

A VTAM APPN node:

- Does not require subarea network routing definitions
- Uses control point-control point (CP-CP) sessions for network control data and logical unit-logical unit (LU-LU) sessions for end-user data

© Copyright IBM Corp. 1984, 2011

- Does not support system services control program-system services control program (SSCP-SSCP) sessions
- Has the NODETYPE start option defined
- Does not have a subarea number defined on the HOSTSA start option

  Note: If the start option, SACONNS, is set to NO, VTAM HOSTA is a pure APPN node even if HOSTA is coded.
- Cannot activate a network control point (NCP)
- Requires static definitions only for those resources located within the node
- Also provides dynamic definitions of some local connections

Control point

Each APPN node has a control point (CP) that manages the node and its resources. Control points provide function for the APPN network similar to the function the SSCP provides for the subarea network. The CP can implement end node or network node functions.

The CP:

- Activates and deactivates network resources
- Assists in session initiation and termination
- Routes cross-domain requests
- Maintains a dynamic representation of the network in the topology and directory services databases (network nodes only)

In an end node, the CP communicates only with the CP in the network node that provides it with network services. In a network node, the CP communicates with the CPs in adjacent network nodes and with the CPs in adjacent end nodes for which the CP is the network node server. The CP directs the activation and deactivation of resources and links and helps LUs start and stop sessions.

The control point name of an adjacent APPN node can be statically defined or determined during XID exchange. The control point name of a VTAM node is determined by the SSCPNAME start option.

Network node

A network node (NN) supports its own users and the end nodes it serves by providing directory and route selection services. It performs intermediate routing of data for sessions that cross it. The NN performs searches of the network to locate resources and calculates the best session route from the node of the primary LU to the node of the secondary LU, based on user-specified criteria.

A VTAM NN:

- Provides intermediate session routing
- Maintains CP-CP sessions with all adjacent network nodes and with all end nodes for which it provides network services
- Has the NODETYPE start option set to NN (NODETYPE=NN)
- Does not have a subarea number defined on the HOSTSA start option

  Note: If the start option, SACONNS, is set to NO, VTAM HOSTA is a pure APPN node even if HOSTA is coded.
- Participates in searches for network resources
- Cannot activate an NCP

- Provides dynamic route calculation

Network nodes and their interconnections form an intermediate routing network. Network nodes use this intermediate routing network to perform searches for partner LUs, and to calculate the best available route for a session for their own resources and the resources of end nodes for which they are acting as network node servers.

Network nodes are distinguished from end nodes by the functions they provide. An NN provides network services for its own resources and the resources of the end nodes for which it acts as a network node server. For more information on network node servers, see "Network node server" on page 8.

To perform NN-provided network services, network nodes maintain a directory services database and a topology database. The directory services database contains information on resources in the network, including network node local resources, and is updated dynamically as a result of searches for partner LUs. The topology database includes information on network nodes and transmission groups (TGs) and is updated dynamically whenever there is a change to the intermediate routing network.

All directory requests from resources (such as LUs on served end nodes, or the network node resources) use the directory services of the NN. Therefore, the network nodes are able to collect and control directory information about the APPN network. Network nodes can originate and participate in APPN broadcast searches.

End node

An end node (EN) is an APPN node that relies on the directory and route selection services of a network node to participate in an APPN network.

A VTAM EN:

- Interacts with a network node to request and receive directory and route selection services
- Supports CP-CP sessions between the end node and its network node server
- Can specify a preferred network node server by means of the NNSPREF start option
- Can define a network node server list
- Submits search requests to its network node server and obtains the session route from its server when initiating LU-LU sessions
- Dynamically registers its resources with its network node server so that the resources can be found during broadcast searches
- Has the NODETYPE start option set to EN (NODETYPE=EN)
- Cannot activate an NCP
- Does not have a subarea number defined on the HOSTSA start option

  Note: If the start option, SACONNS, is set to NO, VTAM HOSTA is a pure APPN node even if HOSTA is coded.

An EN provides communication services for those resources defined within the node. An EN provides directory and session services for its own LUs only.

For directory and routing services to other nodes, an end node uses the services of its network node server. An end node can have links to several network nodes for

connectivity and routing, but only one of these network nodes can be used as the network node server at any given time. A VTAM EN can register its resources in the directory services database of its network node server. This allows searches from other nodes to locate the resources on this EN because VTAM end nodes are not searched. An EN does not perform intermediate routing of sessions; it can only be the origin or destination of a session.

Both a data host and an EN primarily process applications and do not provide network services for the resources attached to them. If your data host has APPN connections and does not require subarea connections, you can migrate it to an EN. (If your data host requires subarea connections, you can migrate it to a migration data host, which is described in "Migration data host" on page 10.)

Network node server

A network node server (NNS) is a network node that provides resource location and route selection services to the LUs it serves. These LUs can be on the network node itself, or on end nodes that have established CP-CP sessions with this network node as their network node server.

To support the LUs on served end nodes, a network node server uses the CP-CP session to provide network information for session setup. Network node servers perform searches for partner LUs and calculate the best route to the partner LU.

Any network node can be a network node server for end nodes that are attached to it. The served end nodes are defined as being in that network node server domain.

Note: A domain in the subarea network is the set of resources controlled by a particular SSCP. A domain in the APPN network is the set of resources served by a particular network node or end node.

Central directory server

A central directory server (CDS) is a network node that builds and maintains a directory of resources throughout the network. The CDS server provides some similar function to that of a CMC host in a subarea network. While a CMC is a central place for network management information that operates and controls the network, a CDS manges only the directory functions, not the operational and network management functions that a CMC also provides. The purpose of a CDS is to reduce the number of network broadcast searches to no more than one per resource. The CDS is capable of maintaining its directory even when VTAM is halted and restarted.

VTAM network nodes and end nodes can register their resources with a central directory server, which acts as a focal point for resource location information. When a VTAM end node registers its resources, it can request that resources be registered either to its network node server only, or to both its network node server and the central directory server. Entries in a directory database can be registered, defined, or dynamic.

When a network node receives a search request for a resource for which it does not have location information, the network node first sends a directed search request to a central directory server, if there is one. The central directory server searches for information about the location of the resource in its directory. If it does not find the resource's location, the central directory server searches end nodes in its domain, other central directory servers, and, if necessary, the entire network.

If the resource is still not found, the central directory server notifies the node that originally requested the search that the search is unsuccessful.

Border node

A network node capable of APPN multiple network connectivity, also known as a border node, can maintain CP-CP connectivity with a network node with a different NETID. APPN topology information does not cross the border node connection or APPN subnetwork boundary, but search requests can, and an LU-LU session can be set up. An APPN subnetwork boundary is assumed by VTAM when a border node is connected to a network node (or border node) with a different network identifier.

An APPN subnetwork boundary can also be explicitly defined through system definition. This way, border nodes can be used to create subnetwork boundaries to adjacent network nodes (or border nodes) with the same network identifier.

For further information on border nodes, see "APPN multiple network connectivity" on page 89.

Nodes with subarea function only

A VTAM subarea node is an SNA type 5 node that functions in a hierarchical environment. Subarea nodes provide services for and control over peripheral nodes. Peripheral nodes require the services of a VTAM subarea node to communicate with other peripheral nodes and subarea nodes in the network. Peripheral nodes include SNA type 2.1, 2.0 or type 1 nodes and function as distributed processors, cluster controllers, or workstations. Type 2.1 nodes can use low-entry networking (LEN) or Advanced Peer-to-Peer Networking (APPN) connections to communicate with adjacent type 2.1 nodes; however, when attaching to a VTAM subarea node, they are restricted to using LEN connections.

A VTAM subarea node:

- Requires subarea network routing definitions, such as path, virtual route (VR), and explicit route (ER) definitions to communicate with other subarea nodes
- Uses SSCP-SSCP, SSCP-PU, SSCP-LU, and LU-LU sessions for network control and end-user data
- Does not support CP-CP sessions
- Has a subarea number defined on the HOSTSA start option
- Does not have the NODETYPE start option defined
- Can activate NCPs and have SSCP-SSCP sessions with other VTAM subarea, interchange, and migration data host nodes

System services control point

A system services control point (SSCP) provides resource management and other session services for users in a subarea network.

To control and provide services for its subordinate nodes, an SSCP establishes sessions with the physical units (PUs) and LUs in those nodes. For example, before an LU-LU session can be established, the SSCP must use a directory of network resources to locate the partner LU. When the partner LU is located, the SSCP establishes an SSCP-LU session with that partner LU.

An SSCP provides the following functions:

- Manages resources in a subarea network in accordance with the commands that network operators issue

- Coordinates the initiation and termination of sessions between LUs in separate nodes within its domain or across domains in cooperation with other SSCPs
- Coordinates the testing and status monitoring of resources within its domain
- Performs recovery when communication fails between network components

Communication management configuration and data hosts

A communication management configuration (CMC) simplifies network management. One VTAM in the network acts as the CMC host and owns all resources except applications, which are predominately owned by other hosts called data hosts. With this configuration the entire network can be controlled by a network operator at the CMC.

A CMC configuration can also be used to enhance routing. The CMC handles session establishment and termination. The data hosts then spend more time doing application processing and less time doing session establishment and routing.

Nodes with both subarea and APPN function

Nodes that need to be able to directly connect to both subarea and APPN networks require both subarea and APPN function.

Interchange node

An interchange node resides on the border of an APPN network and a subarea network. It provides protocol conversion between subarea and APPN networks to enable the integration of the two types of networks. Because an interchange node can convert session requests from one protocol to the other and can provide intermediate routing, it can establish sessions from one type of network to the other.

An interchange node combines the function of a subarea node and a network node. It controls resources and functions as a network node in the APPN network and as an SSCP and a cross-domain resource manager (CDRM) in the subarea network. All of the characteristics described for network nodes and subarea nodes apply to interchange nodes.

An interchange node:
- Uses subarea path definitions to determine routes within the subarea network
- Uses the topology database to determine routes within APPN networks
- Uses both SSCP-SSCP and CP-CP sessions to communicate with other nodes
- Has a subarea number and NODETYPE=NN defined
- Can own and activate NCPs

The interchange node communicates network control data by using SSCP-SSCP sessions with other subarea nodes and CP-CP sessions with other APPN nodes. To enable it to participate in the subarea network, it is defined with a unique subarea number and requires subarea path definition statements. It can be connected to other APPN nodes, LEN nodes, and subarea nodes.

Note: The SACONNS start option must be set correctly to allow VTAM to function as an interchange node.

Migration data host

A migration data host resides on the periphery of a combined APPN and subarea network. A migration data host combines the function of an end node with the function and role of a subarea data host.

A migration data host:
- Uses subarea network routing definitions
- Should not perform intermediate routing in APPN or subarea networks
- Uses CP-CP and SSCP-SSCP sessions to communicate with other nodes
- Has NODETYPE=EN defined
- Cannot activate an NCP
- Has a subarea number defined on the HOSTSA start option

Note: The SACONNS start option must be set correctly to allow VTAM to function as a migration data host.

Like a data host in a subarea network, a migration data host is dedicated to processing application programs and does not control network resources. It also participates as a cross-domain resource manager (CDRM) in the subarea network. The migration data host also functions as an end node in the APPN network. All of the characteristics previously described for end nodes apply to migration data hosts.

To enable the migration data host to participate in the subarea network, it is defined with a unique subarea number and supports subarea path definition statements.

The migration data host communicates network control data by using SSCP-SSCP sessions with other subarea nodes and CP-CP sessions with its network node server. It can be connected to other APPN nodes, LEN nodes, and subarea nodes. A migration data host can perform intermediate routing in either a subarea network or an APPN network, but it is not recommended. However, its locally attached LEN nodes and dependent LUs can establish sessions with resources in other APPN or subarea domains.

Composite network node

A composite network node (CNN) is composed of a VTAM interchange node and any NCPs that it owns. In an APPN network, it functions as a network node and appears to the APPN network as a single node. A composite network node is defined by coding the HOSTSA start option, the NODETYPE start option as NN, and by activating an NCP from that VTAM.

The composite network node can be attached to other APPN nodes and also to subarea nodes. In a composite network node where all external connections are APPN connections, the HOSTSA value is not used in communication outside the node. If the composite node is connected to other nodes by subarea connections, the HOSTSA value is used in communication with the subarea network.

Notes:

1. If the composite network node has APPN connections through its NCP, the NCP needs to be at Version 6 Release 2 or later.
2. The SACONNS start option must be set correctly to allow VTAM to function as a composite network node.

Low-entry networking node

A low-entry networking (LEN) node is a type 2.1 node that can function in both a subarea and an APPN environment. VTAM and any NCPs that it owns can present the image of a composite LEN node. It appears as a single node to all LEN or APPN nodes to which it is attached.

A LEN node provides peer-to-peer connectivity to subarea nodes, end nodes, or network nodes that are providing a LEN appearance. Unlike end nodes, the LEN node cannot establish CP-CP sessions with a network node. Therefore, a LEN node cannot automatically register its resources with a network node server.

A VTAM acting as a LEN node cannot establish SSCP-SSCP sessions across T2.1 connections with a VTAM subarea node. A LEN node can attach to a subarea network as a peripheral node. In a subarea network, LEN nodes can communicate with other LEN nodes without requiring the services of an SSCP.

In an APPN network, you can predefine the CP name of a network node adjacent to a LEN node, which enables the LEN node to send a BIND to the adjacent node. The adjacent node determines the actual location of the LU, calculates the route to it, and forwards the BIND.

Network accessible units

Network accessible units (NAUs) are elements in the network with which sessions can be established and queries can be directed. In subarea networks, these units have both a subarea number and an element address and are also called network addressable units. Although subarea addresses are not used in APPN networks, the term network accessible units is used to refer to these units in both types of networks.

Physical unit

Physical units (PUs) exist in type 4, type 5, type 2.0, and type 1 nodes. Combined type 2.0/2.1 nodes, such as a 3174 controller, have both a CP and PU.

A PU provides the following functions:

- Receives and acts upon requests from CPs, such as activating and deactivating links to adjacent nodes
- Manages links and link stations while accounting for the unique aspects of different link types
- Manages network addresses, virtual routes, and explicit routes in type 4 and type 5 nodes

The PU supports sessions with the SSCP in type 5 nodes.

Logical unit

A logical unit (LU) acts as the intermediary between the user and the network. Before two users can communicate, their LUs must be in session with one another. Communication occurs only between LUs of the same LU type. The LU-LU session manages the flow of data between the users.

VTAM supports LU types 0, 1, 2, 3, and 6.2. These types support the communication requirements of a variety of users. Each type supports a specific set of communication requirements. The number of the LU type is not related to the PU or node type.

LU types 0, 1, 2, and 3 support communication between application programs and peripheral equipment, such as workstations or printers. LU type 6.2 supports communication between two programs located at type 5 subarea nodes, type 2.1 peripheral nodes, or both, and between programs and devices. LU type 6.2 supports multiple concurrent sessions, called parallel sessions.

Dependent LU support

Dependent LUs are supported by VTAM in both subarea and APPN networks.

A dependent LU requires assistance from a system services control point (SSCP) to activate an LU-LU session; therefore, it requires an SSCP-LU session. Both subarea and APPN VTAM nodes provide SSCP support for dependent LUs in their domains. A VTAM APPN node provides SSCP support for those LUs in its domain even though it has no SSCP-SSCP sessions with other nodes.

Dependent LUs can use APPN and subarea connections to access VTAM application programs in other domains. For example, a dependent LU type 2 terminal session originating in one VTAM can be rerouted to CICS® on a VTAM end node.

To resources in the APPN network, a dependent LU appears to be located on the node that is providing SSCP services to the LU. Through the use of DLUS/DLUR, VTAM can establish SSCP-LU sessions across APPN links. Therefore, a VTAM node can be the owner of a dependent LU that is on an attached non-VTAM APPN node, such as an AS/400®, PS/2, or 3174 controller, regardless of whether the VTAM node has a CP-CP session with the other node.

How dependent LUs are defined

Dependent LUs are owned by a VTAM node and are defined to VTAM. They are considered to be in the domain of the VTAM node. If a type 2.1 node has dependent LUs attached to it, the dependent LU must also be defined to VTAM.

Dependent LUs such as terminals and printers appear as cross-domain resources to other VTAM hosts. If the LUs are registered, and if all nodes in the network are connected by CP-CP sessions, the cross-domain resource definition statements in other hosts might not be required. Some other hosts do support the dynamic creation of CDRSCs, even in a subarea environment. In an APPN environment, other hosts will still build CDRSCs when sessions are established.

How dependent LUs participate in sessions

Dependent LUs can start a session when the primary LU is owned by VTAM. Dependent LUs can be the secondary LU in sessions with partner LUs in the APPN or subarea network.

Sessions initiated by secondary LUs (such as LU type 2 displays) to a primary LU that is in a VTAM node in the APPN network are supported. Also supported are APPN LUs that initiate sessions with application programs in the subarea network.

Independent LU support

Independent LUs do not require the services of an SSCP to initiate an LU-LU session. Therefore, they do not have SSCP-LU sessions. Independent LUs can use the services of their local control point to locate and determine a session route to a desired session partner.

Independent LUs must be defined at the owning CP. At the owning CP, they are defined as applications or device-type LUs. Independent LUs can be statically defined at non-owning CPs as cross-domain resources (CDRSCs), but this is unnecessary because these resources can be defined dynamically as they are needed. The one exception is when the LU is owned by a LEN node. Because a

LEN node cannot register its LUs, these LUs cannot be found as destination LUs if they are not defined in the adjacent VTAM as CDRSCs with an ALSLST or as LOCADDR=0 LUs under a PU definition.

Network control sessions

Network control sessions allow two nodes in a network to communicate, enabling end-user sessions between logical units located at those nodes. In subarea networks, network control sessions are called SSCP-SSCP, SSCP-NCP, SSCP-PU, and SSCP-LU sessions. In APPN networks, network control sessions are called CP-CP sessions.

SSCP-SSCP sessions

SSCP-SSCP sessions provide control and coordinate session setup for cross-domain and cross-network sessions. These sessions are established between VTAMs only. SSCPs exchange capabilities during CDRM activation. SSCP-SSCP sessions are established between all hosts in a network. SSCP-SSCP sessions can only be established between two hosts with subarea function. Within one subarea network, for an LU-LU session to be established, the SSCPs that own the LUs must be in session with one another.

For dependent LUs, the SSCPs must also establish SSCP-PU and SSCP-LU sessions before an LU-LU session can be established. An SSCP establishes sessions with PUs and LUs defined to it during network activation, or, in switched links, during link activation. SSCP-LU sessions need to be established for dependent LUs only. Independent LUs can function without an SSCP-LU session.

CP-CP sessions

To perform directory services and topology and route selection services, adjacent APPN nodes throughout the APPN network use a set of two CP-CP sessions to exchange network information. CP-CP sessions are established between type 2.1 nodes. A network node or composite network node can establish CP-CP sessions with any network node or composite network node to which it has an APPN link that supports CP-CP sessions, with the exception that CP-CP session between network nodes having different Net IDs are permitted only if at least one of the network nodes supports the border node function. It also establishes CP-CP sessions with each of the end nodes that it serves. End nodes can be attached to several network nodes but can establish a CP-CP session pair with only one network node server at a time.

When two nodes have one or more CP-CP capable links between them, VTAM requires a CP-CP session to be active between the nodes, as long as none of the following conditions disallowing CP-CP sessions are true:

- The CP-CP capable links exist between two adjacent network nodes with different net IDs, and neither network node provides border node support.
- The CP-CP capable links exist between an end node and a network node, and the end node already has CP-CP sessions with a different network node.
- The CP-CP capable links exist between two adjacent end nodes.

If the CP-CP session is deactivated, you need to either reactivate the CP-CP session or deactivate all the CP-CP capable links and reactivate them with CPCP=NO.

In an APPN network you do not need meshed connectivity of CP-CP sessions with every node having a CP-CP session with every other node in the network. With APPN it is only necessary for each APPN node to have a CP-CP session into the

network. A CP-CP session is not required between the end points of an LU-LU session. Defining CP-CP capable links to every other node in your network might cause excessive topology update flows. It is recommended that you define a primary and backup link with CONNTYPE=APPN and CPCP=YES for each node in your network. It is also recommended that you define additional APPN-capable links with CPCP=NO to provide efficient routing for your LU-LU sessions.

CP-CP sessions between two network nodes are used to perform searches for resources, exchange topology information, and can be used to register resources with a central directory server.

After an APPN connection has been established, identification information is exchanged between the nodes, and CP-CP sessions can be started between the control points in the directly attached nodes.

After the CP-CP sessions are established, the two nodes exchange CP capabilities, which indicate the level of network services provided by the control point. When both nodes are network nodes, they exchange topology database update (TDU) messages. The TDU messages contain identifying information, node and link characteristics, and resource sequence numbers that identify the most recent updates for each of the resources described in the TDUs.

When an LU-LU session is requested, information is transported across the CP-CP sessions to request that an LU-LU session be established between the two session partners. The control points that own the two session partners do not have to have a direct CP-CP session between them in order to establish the LU-LU session.

End-user sessions

End-user sessions are called LU-LU sessions in both subarea and APPN networks. An LU-LU session between two logical units enables one user to communicate with another. An example of an LU-LU session is an application, such as CICS, communicating with a workstation. LU-LU sessions are used only for user data, not for network control data.

A VTAM node can both initiate sessions and respond to session initiation requests. An LU-LU session is established when a BIND is sent from the primary LU to the secondary LU. Dependent LUs act as secondary LUs only and have an LU-LU session limit of one. Independent LUs can have parallel sessions between the same pair of LUs and can have multiple sessions between one LU and several other LUs. Independent LUs exchange only UNBIND and RSP(UNBIND) for session termination. Independent LUs can be either the PLU or the SLU for any given session. An UNBIND can be sent by either LU.

A VTAM node can have both an SSCP and a CP. Therefore, VTAM nodes support the following types of session initiation over SSCP-SSCP or CP-CP sessions:
- Sessions initiated by the primary and secondary LU
- Queueing of session initiation requests
- Sessions initiated by a third party (CLSDST PASS)
- Orderly termination of existing sessions
- Automatic logon sessions
- Forwarding of session release requests

In an APPN network, the owning CP for an LU uses CP-CP sessions to obtain location and routing information needed to establish an LU-LU session. The

owning CPs of the session partners do not have to be adjacent to each other and do not require having a direct CP-CP session with each other if they are not adjacent.

Sessions are established using a specific mode. The mode identifies session parameters and the APPN Class of Service (COS) name to use for the session. In subarea networks, the Class of Service indicates the virtual route and transmission priority to be used for this session. In APPN networks, the APPN Class of Service tables specify weights to be applied to the various routes in the APPN network based on the characteristics of the links and nodes. These weights are calculated, taking into account the various characteristics such as acceptable level of security, cost per byte, cost per connect-time, propagation delay, and effective capacity. For more information about Class of Service in APPN, see "APPN class of service" on page 19.

VTAM allows dependent LUs, such as workstations and printers supporting LU types 0, 1, 2, and 3 in one VTAM domain to access VTAM application programs in a different domain by using APPN protocols between domains. All dependent LU types are supported through APPN networks with one exception: Sessions with binary synchronous 3270 devices are not supported over APPN links.

Examples of these dependent LU types are provided in Table 1.

Table 1. LU types

LU type	Example
Type 0	3650 and 4700 financial terminals
Type 1	3767 and 3770 remote job entry (RJE) stations
Type 2	3270 interactive displays
Type 3	3270 printers

How VTAM locates resources

For VTAM to locate a resource in the network, VTAM must either have a definition for the resource or be able to dynamically define the resource. Following are descriptions of how resources are located in a subarea or APPN network.

Locating resources in a subarea network

In a subarea network, when VTAM receives a session initiation or directory search request for a resource that is not located in that VTAM domain, it attempts to locate the resource by sending the request to its adjacent SSCPs.

If the resource is statically defined as a CDRSC (or matches an active model CDRSC) and has the CDRM operand defined, VTAM routes the request to the SSCP specified in the CDRM operand. VTAM uses an adjacent SSCP table if:

- The CDRM operand is not defined
- The resource is not found at the CDRM specified
- VTAM has no SSCP-SSCP session with the specified CDRM
- The CDRSC is dynamically defined

The adjacent SSCP table can be statically defined or, if the start option DYNASSCP is defined as YES (which is the default), it can be dynamic. VTAM searches each

SSCP in the table for the resource until the resource is found or the table is exhausted. See "Adjacent SSCPs" on page 541 for additional information about using adjacent SSCP tables.

For independent LUs located on or through adjacent LEN nodes, VTAM uses these definitions to determine connectivity to the resource when establishing a session with the independent LU.

Locating resources in an APPN network

In an APPN network, when VTAM receives a session initiation or directory search request for a resource, VTAM checks its directory services database for the location of the resource. The directory services database contains information on the location of resources obtained through resource definition, resource registration, and as the result of successful searches. If the resource is not located in that VTAM database, VTAM issues a broadcast search to all other APPN nodes or a search to a central directory server, if one is available. If there are interchange nodes in the network, at the end of a broadcast search, they are sequentially allowed to search the subareas to which they are attached. VTAM uses a cross-subnetwork search to search across a subnetwork boundary.

Route selection

VTAM selects routes differently in subarea and APPN networks.

Routing in a subarea network

Defining a route in a subarea network involves the following elements:

- Subarea links
- Transmission groups (TGs)
- Explicit routes (ERs)
- Virtual routes (VRs)
- Path definitions
- Logon mode tables
- Subarea class of service (COS)

Subarea links are the different types of connections that can be defined between subareas. For example, links include a channel connection between two VTAMs, a channel connection between VTAM and NCP, or an SDLC, frame-relay, or token-ring connection between two NCPs.

Transmission groups (TGs) allow links with similar characteristics to be logically associated. Several links between the same two subarea nodes might have the same TG number, or the TG number might represent only one link.

Explicit routes (ERs) consist of a list of TGs and subarea nodes in one direction between the two end points of the route. For each ER in one direction, there must be a corresponding ER in the opposite direction that follows the same path.

Virtual routes (VRs) represent the complete route path. Each VR has a specific number and the VR definition associates that VR number with the ER to be used for a session. The VR specifies two one-way ERs to create a round-trip path.

Explicit routes and VRs are defined within path definition statements. Each path definition statement gives the ultimate destination point of the path. The ER gives the transmission group and subarea number for the next hop on the path. The

association of the VR number with the ER within the path definition connects the individual hops together into a complete path from one end point to the other.

Each LU-LU session has an associated logon mode table entry. This entry specifies, among other things, the name of a subarea COS that contains a list of pairs of VR numbers and transmission priorities that can be used to establish the session. The subarea COS allows you to specify the VR that best fits the characteristics needed for that session such as capacity, security, and speed.

For more information on subarea routing, see "Network routing for subarea nodes" on page 348.

Routing in an APPN network

APPN nodes provide a set of topology and routing functions that are used with the directory functions to find resources and calculate routes for sessions. Links must be defined to the nodes that own them, but routes over other links in the network can be learned dynamically.

Topology and routing services

The following concepts describe the topology and routing functions:

- Topology database
  - Local
  - Network
- Class of Service
- Route selection

Topology database

To enable a network node to provide routing functions to and from itself and the end nodes it serves, every network node maintains a network topology database that has complete and current topology information about the network. This topology information consists of the characteristics of all network nodes in the network and of all transmission groups (TGs) between network nodes. The end nodes in the network and the TGs connected to them are not considered network topology information.

This section describes the two kinds of topology databases in an APPN network and explains how each is used. Local topology information and network topology information is maintained at each network node. The local topology information is unique to the node; the network topology information is replicated at all network nodes.

Local topology database

A local topology database contains information about a node and about the TGs connecting the node to other type 2.1 nodes. Both end nodes and network nodes have local topology databases.

In an end node, the local topology database is used to calculate a route to the network node server and to supply the endpoint TG vectors (vectors that describe the TGs connected to the end node) to the network node server. In a network node, the local topology database includes information about the attached end nodes. The local topology database is not saved when VTAM is stopped and restarted.

Network topology database

The network topology database is referred to as the *topology database*. The network topology database is kept only in network nodes and contains

complete and current information about all network nodes in the network, and about all transmission groups connecting them. It does not contain any information about LUs, end nodes, or LEN nodes.

The network topology database dynamically records changes in network topology and can be checkpointed to a file and reloaded when VTAM is restarted to reduce the amount of network traffic when a node is restarted.

To keep the topology database current, network nodes in an APPN network send each other topology database updates (TDUs) over CP-CP sessions whenever a resource (node or link) is activated, becomes inoperative, deactivated, or its characteristics change. Only the current changes are included in the TDU, not the complete topology. The TDUs contain information about the network nodes and the TGs between the network nodes. Every network node receives the TDUs, so all have the same view of the network.

A resource sequence number (RSN) is associated with each resource, and is incremented each time that resource changes status. The RSN is used to determine when all network nodes have received the most current information about the node or TG, at which time the TDU activity for that resource ceases. As long as all network nodes follow the same rules with regard to TDU processing, the RSN guarantees that a TDU war does not occur. A TDU war is an endless exchange of TDUs in contention over the same topology resource, resulting in continuous performance degradation of the APPN network. See the Display TDU information section in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for information about diagnosing TDU wars.

In an APPN network, several mechanisms are used to ensure that unnecessary or excessive TDUs are not propagated throughout the network. For example, multiple TDUs can be grouped and sent out together if there are updates to multiple local TGs.

The topology database is used by the network node to select routes for sessions that originate at the LUs in it and at the ENs that it serves.

APPN class of service

An APPN class of service (COS) defines the required or requested characteristics of a route for a session. A COS consists of a set of ranges of acceptable values for the characteristics of links and nodes to be used for a session specifying that particular COS. APPN Classes of Service are defined in a VTAMLST definition list.

Unlike the COS for the subarea network, where the COS is actually a list of VRs that are acceptable for a particular COS, APPN COS specifies the types of routes that are acceptable for a Class of Service.

Class of Service database

The COS database exists on all VTAM network nodes (NNs) and end nodes (ENs). It is the same on both ENs and NNs, and it is built when VTAM is initialized.

The COS database contains the following items:

- COS definitions from VTAMLST definition files

  IBM provides three sets of COS definitions: COSAPPN, ISTACST2. and ISTACST3. Each set contains the same seven default APPN COS definitions. However, differences exist in the way the seven Classes of Service are defined in each set. See “What are the IBM-supplied default classes of service?” on page 334 for information about the differences

between the three sets and how to determine which one is appropriate for your configuration. It is recommended that you use the defaults wherever possible and that the COS definitions be the same throughout the network. The default APPN COS definitions in z/OS Communications Server (COSAPPN) are the same default Classes of Service provided by other APPN products.

One or all of the IBM-supplied sets of COS definitions can be included in VTAMLST. User-defined Classes of Service can also be included. For information on how to define your own COS, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

You can have only one set of COS definitions in VTAMLST active at any time. The IBM-supplied COS definitions in APPNCOS are automatically activated during z/OS Communication Server initialization. COS definitions can also be activated with a VARY command or by specifying the name of the VTAMLST member in a CONFIG list. Entries can be added or modified by activating or reactivating the VTAMLST definitions.

- A list of logon mode names and the corresponding APPN COS

  A set of default mappings for the seven IBM-supplied classes of service is included in the default logon mode table. It is recommended that users also code one of the seven standard APPN COS names in any user defined mode tables. For information about defining user-specified modes and corresponding Classes of Service, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

- A list of calculated routes

  As an NN calculates routes for particular classes of service, the routes are stored in VTAM memory. Storing routes allows them to be reused, rather than being recalculated each time a route is requested.

  Routes are periodically recalculated to allow for distribution of sessions over equally-weighted paths. Start options control this process; refer to *z/OS Communications Server: SNA Resource Definition Reference*.

Route calculation and selection

VTAM provides automatic, preferred route selection based upon dynamic network characteristics and requested COS. During session establishment, the network node server of the origin LU refers to the topology database to calculate and select the current best route through the APPN network from the primary LU to the secondary LU for the requested COS.

An NN calculates routes for sessions that originate at the LUs in it and at the ENs it serves. When a route is calculated, it is stored and can be reused. The stored routes are unique to the node and are never broadcast. The cache of stored routes is maintained automatically. When the maximum cache size (set by the NUMTREES start option) is exceeded, the least recently used routes are discarded first.

Route selection is based on how the actual characteristics of each node and TG along the possible paths match the characteristics required by the requested COS. The route that an NN selects is the current least-weight (or best) route from the node containing the origin LU to the node containing the destination LU.

Addressing

VTAM uses different addressing schemes in subarea and APPN networks.

Subarea addressing

Subarea addressing uses an addressing scheme that consists of a subarea number representing the subarea a resource belongs to, and an element address, which uniquely identifies that resource within the subarea. The subarea number for VTAM is given by the HOSTSA start option. (The subarea number for NCP is determined by the SUBAREA operand on the BUILD statement.) The element address is assigned from a pool of addresses. Both VTAM and NCP maintain a pool of addresses for their attached resources.

APPN addressing

APPN nodes use local form session identifiers (LFSIDs) instead of subarea addresses for external communication in an APPN network. VTAM APPN nodes have subarea addresses that are completely internal to the nodes, used by the NCP in a composite network node to communicate with its owning VTAM. This subarea address is never used outside the composite network node.

When High-Performance Routing (HPR) is in use, LFSIDs are not used for the HPR portion of the sessions. Instead, automatic network routing (ANR) labels are used by each node to route the HPR network layer packets (NLPs). For further information on HPR, see "High-Performance Routing (HPR)" on page 491.

Controlling network data flow using pacing

Pacing is a means of controlling the flow of messages in the network to avoid congestion. Data congestion results whenever the rate at which data is going into a network exceeds the capacity of the network. Response times might lengthen and throughput might decrease. Severe or prolonged congestion in one part of a network can affect the other parts, causing overall network efficiency to suffer.

VTAM monitors traffic and limits congestion in the network with pacing. Pacing controls network flow by limiting the amount of data a transmitter can send before receiving an acknowledgment from the receiver. For example, if the agreed amount of data is three units, after sending three units of data the transmitter must wait for the receiver to send an acknowledgment before sending any more data.

Table 2 lists the two levels of pacing that are involved in the flow-control process.

Table 2. Pacing types

Pacing type	Description
Session	Session pacing prevents overrun of the buffers of the LUs at the session endpoints. There are two types of session pacing: fixed and adaptive. See "Virtual route pacing" on page 365 for more information.
Virtual Route	Virtual route pacing is used by subarea nodes to avoid buffer congestion at the virtual route endpoints. See "Virtual route pacing" on page 365 for more information.

Both pacing mechanisms allow pacing to occur inbound and outbound independently. Pacing values do not have to be the same for the two directions. The term *inbound pacing* is associated with the pacing window for message units being received by a node. *Outbound pacing* is associated with the pacing window for messages being sent by the node.

Pacing concepts

Before continuing with the specifics of the two pacing mechanisms, you should understand basic pacing concepts. Pacing is controlled using pacing window techniques. A *pacing window* is the number of message units that a pacing endpoint can send to another network component at one time. The first message unit contains a pacing request, which indicates that a pacing response must be sent by the receiver before another pacing window of message units can be sent.

Pacing window

For example, in Figure 1 the pacing window size is three. Once Node A sends three message units, it must wait for a pacing response (from Node B) before sending another message unit. The pacing response restores the pacing window size.

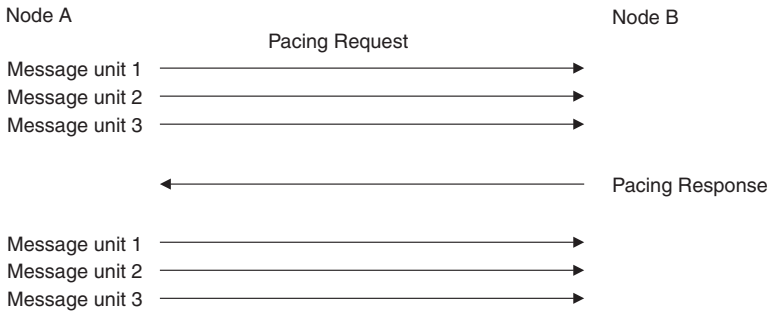


Figure 1. Pacing flow – outbound pacing

Figure 1 shows pacing from the sender's outbound pacing perspective. There could also be a pacing window size associated with the sender's receiving message units (inbound pacing). The pacing window size can be different from the outbound pacing window size.

In Figure 2, Node A inbound pacing window size is two. Node B, after sending two message units, will not send additional message units until Node A sends a pacing response to Node B as shown in Figure 2.

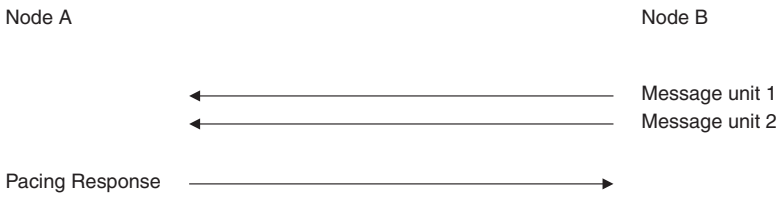


Figure 2. Pacing flow – inbound pacing

A pacing window response can be sent before the pacing window size is reached. If the receiver sends the pacing response early (before having received the maximum number of requests), the sender transmits the first group of requests and can then send another group of requests.

Figure 3 on page 23 shows the flow when a pacing response is sent before reaching the pacing window size. Node A inbound pacing window size is three. Node A could send the pacing response after receiving message unit 1. Node B could send up to three more message units before another pacing response is required to be

sent by Node A.

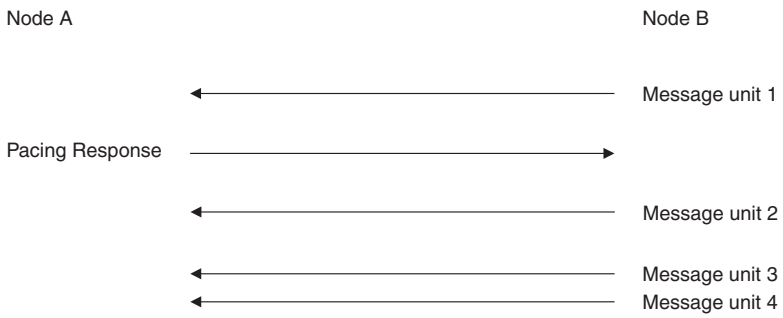


Figure 3. Pacing flow – receiving early pacing response

Chapter 3. Implementing a VTAM network

This topic explains how to implement a VTAM network and includes the following topics:

- Using start options and configuration lists
- Identifying resources to VTAM
- Verifying a VTAM network

Implementing a VTAM network involves the following:

- Use of start options and an optional configuration list
- Identification of particular resources in the network to VTAM
- Identification of any paths required for network routing
- Establishment of sessions
- Operation of VTAM itself

You will also need to understand the VTAM start procedure and the associated data sets. For information about this process, refer to *z/OS Communications Server: New Function Summary*.

This topic gives descriptions of VTAM network requirements and an example of implementing a VTAM network and includes the following:

Start options
Start options control the conditions under which VTAM runs. No matter what network configuration you are implementing, you need to code some start options. Start options are coded in files named ATCSTRxx, where xx specifies the identifier of a particular start option file.

Configuration list
A configuration list lets you specify which resources are activated automatically when you start VTAM. You are not required to define a configuration list, but it makes the VTAM operator's job easier because VTAM activates the resources in the configuration list automatically. Configuration lists are coded in files named ATCCONxx, where xx specifies the identifier of a particular configuration list.

Application programs
Each host application program you are running must be defined to VTAM as an application program minor node within an application program major node. A major node, such as an application program major node, is a set of minor nodes (in this case, the application programs) that can be activated and deactivated as a group. For more information on application programs, see Chapter 14, “Application programs,” on page 373.

Note: An application program may become a shadow resource if a CDRSC with the same name already exists when the major node containing the application program's definition is activated. For more information on shadow resources, see “Shadow resources” on page 549.

Subarea nodes
If you have one or more NCPs in your network, define them in NCP major nodes and channel-attachment major nodes.

© Copyright IBM Corp. 1984, 2011

If there are other VTAMs in your network, define connections to them in NCP major nodes, channel-attachment major nodes, or external communication adapter (XCA) major nodes. If your VTAM is to have SSCP-SSCP sessions with other VTAMs, also create a cross-domain resource manager major node and minor nodes for your VTAM and adjacent VTAMs with which your VTAM is to have such sessions.

You need to define paths for data flow between VTAM and any owned NCPs, and for data flow to and from other VTAMs and NCPs in your network over subarea connections. Paths are defined in PATH definition statements.

APPN nodes
If VTAM is going to use APPN functions, specify the NODETYPE start option.

Peripheral nodes
Other physical devices in your network must also be defined to VTAM, whether they are directly attached to the host or to an NCP. Peripheral nodes are dynamically defined or manually defined in channel-attachment, external communication adapter, local non-SNA, local SNA, LU group, model, packet, and switched major nodes. Logical units in or attached to peripheral nodes are defined with LU statements along with the peripheral nodes in the major node definitions, or if they are independent LUs, they can be defined in cross-domain resource major nodes. The major node you choose depends on the characteristics of the device you are defining.

Figure 4 on page 27 is a sample VTAM subarea network. The host is running MVS, VTAM, and two VTAM application programs. NCP11 is a channel-attached 3745 controller, and NCP12 is a link-attached 3725 controller. Peripheral nodes are connected using leased lines.

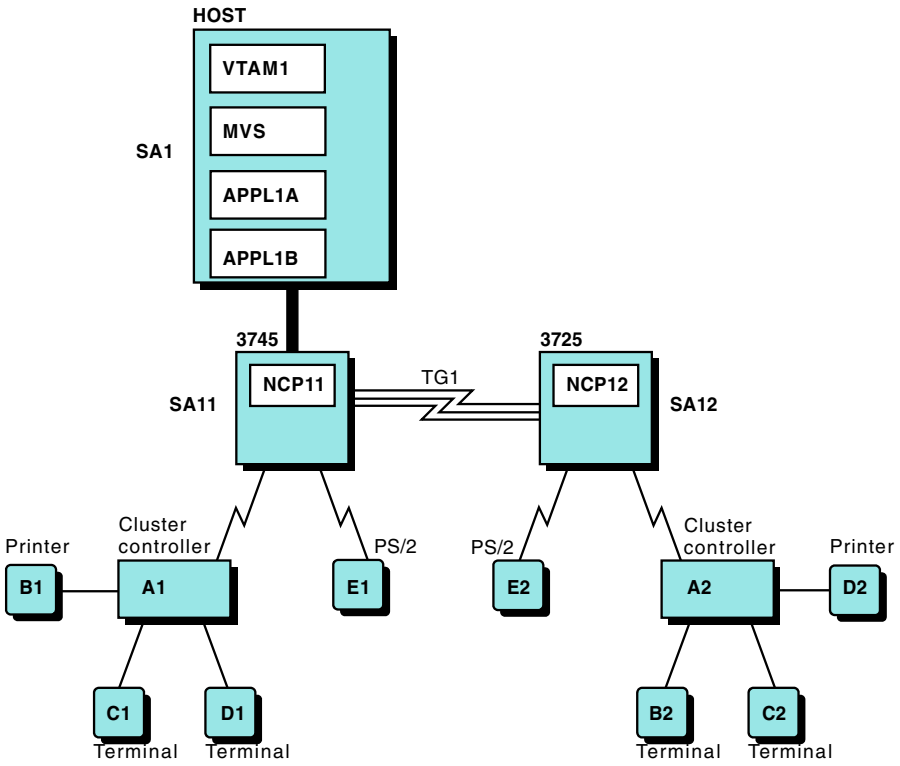


Figure 4. VTAM network

You need to code at least the following resources to define the network in Figure 4:

- Start options describing the host (NETID, SSCPID, and SSCPNAME).
- Application program minor nodes defining APPL1A and APPL1B.
- Two NCP major nodes defining NCP11 and NCP12. These include the NCP definitions. In the NCP major node, also code GROUP, LINE, PU, and LU definition statements to define the peripheral nodes attached to the NCPs.

  Note: APPN nodes, PUs, and LUs can be defined dynamically.
- PATH definition statements in each subarea to attach to the other subareas. For example, in NCP11, you would code two PATH definition statements to attach to subarea 1 (DESTSA=1) and subarea 12 (DESTSA=12).

Using start options and configuration lists

When you start VTAM, you can enter start options on the START command, or you can write one or more start lists to assist you in starting VTAM. You can also activate your resources either manually or automatically. To automatically activate some or all of your resources, you can write one or more configuration lists. Writing start lists and configuration lists:

- Reduces the amount of operator involvement and the chance of entering incorrect information
- Enables VTAM to initialize the domain faster

You can code multiple start options or configuration lists and use these lists according to your needs. After you specify start options and configuration lists, they remain in effect until you modify them or until VTAM is halted.

The syntax rules for start lists and configuration lists are less restrictive than those for other VTAM definition files. Full line comments must always begin with an asterisk (*) in column 1 and cannot be continued on the next line using a continuation character in column 72. No continuation characters are required in column 72, and commas are not required between statements. Continuation characters and commas are permitted, if desired, and will not cause a problem.

For more information on coding start options and configuration lists, refer to the Rules for coding start option lists and configuration lists section of the *z/OS Communications Server: SNA Resource Definition Reference*.

Start options

The following start options are discussed in this section:

- Required start options
- Recommended start options
- Sources of start options
- Start option processing
- Creating start option lists
- Improving VTAM performance using start options
- Increasing Host Subarea Element Addresses
- Enhanced addressing for session managers
- Diagnosing problems using start options

Start options provide information about the conditions under which VTAM runs. They also enable you to tailor VTAM to meet your needs each time VTAM is started. Many operands can have defaults specified as start options, thus reducing the amount of coding required. Many start options can be dynamically modified and also displayed. A complete list of start options is listed in the *z/OS Communications Server: SNA Resource Definition Reference*.

Required start options

The following three start options are required:

SSCPID

The SSCPID start option provides VTAM with a unique numeric identifier. The SSCPID value is used by some physical units to identify the VTAM with which it is in session.

If you plan to expand or incorporate a single-domain network into a larger network, be sure that the value of SSCPID is unique for each host. The SSCPID value you specify must be different from the SSCPIDs in other networks that can be in session with this host.

If you plan to incorporate a non-VTAM host, the values of the VTAM SSCPID and HOSTSA start options should be coded so that one of the following is true:

- The values for both SSCPID and HOSTSA are higher than the non-VTAM host.
- The values for both SSCPID and HOSTSA are lower than the non-VTAM host.

When activating the cross-domain resource manager-cross-domain resource manager (CDRM-CDRM) session, the host with the higher SSCPID value sends the activate cross-domain resource manager (ACTCDRM) and the host with the higher HOSTSA value activates the route. Unpredictable results occur if the route chosen for the ACTCDRM is different from the route activated.

SSCPNAME

The SSCPNAME start option provides a unique name for VTAM. This option is required for a single-domain network, but is primarily used in multiple-domain and multiple-network environments to identify a particular VTAM. The SSCPNAME option must be different from the HOSTPU start option that identifies the physical unit within VTAM.

The SSCPNAME option is also used as the CP name for a VTAM that implements APPN. Although the SSCPNAME and CP name are identical, note that the SSCP and CP are distinct resources and can be displayed individually.

Note: The SSCPNAME should match the name that is coded in the CDRM major node for this VTAM.

NETID

The NETID start option provides VTAM with the network identifier. If you connect your VTAM to another network, the network identifiers must be unique. The network name should conform to an overall naming standard, such as the following:

- The first two characters identify the country in which the network is managed and comply with the International Standards Organization (ISO) Standard 3166.
- The next four characters identify your enterprise.
- The last two characters identify a particular network within your enterprise.

For example, an IBM network within the United States could have the network identifier *USIBMCMK*, where *US* identifies the country, *IBMC* identifies the enterprise, and *MK* identifies a particular network within IBM.

Notes:

1. Using the # symbol for defining network identifiers (NETIDs) is not recommended.
2. To assist SNA network owners in controlling the uniqueness of their network IDs, IBM has established a worldwide SNA Network Registry. Registering a network ID in the SNA Network Registry confirms its uniqueness from all other network IDs within the registry, thus minimizing future connectivity problems. To register a network ID, contact your IBM branch office representative.

Recommended start option

The HOSTPU start option is recommended (but not required) for identifying VTAM to the network.

Use the HOSTPU start option to assign a user-defined name to the VTAM host physical unit. The default name for the host physical unit is ISTPUS. Because the NetView® program uses this name to associate network information to a specific VTAM, you should make sure that this name is meaningful and unique throughout the network.

For start options you are not required to code, VTAM uses the default values. You can override any start option value when VTAM is started or enter new start option values at the operator console.

Sources of start options

When VTAM is started, you can provide the options from any combination of the sources listed below (arranged from the lowest to highest priority):

- IBM-supplied values internal to VTAM (default values).
- Default start option list. When you start VTAM, a list of user-defined default values (ATCSTR00) is read from the VTAM definition library. If VTAM cannot find the ATCSTR00 file, it prompts you, giving you a choice of:
  - Using default values
  - Using an alternate start option file (ATCSTRxx, where xx specifies the identifier of the start option file you want to use)
  - Canceling the start attempt

  If you do not want to use the ATCSTR00 file, you can avoid the prompt by coding an ATCSTR00 file that contains only comments. VTAM issues a message that the file is empty, but does not interrupt start processing.
- Supplemental start option list. The VTAM operator can enter the LIST=xx start option on the START command to name another list to supplement ATCSTR00. The supplemental list (ATCSTRxx) can override options in the default ATCSTR00 list, as well as the IBM-supplied internal default values.
- Backup start option list. The LISTBKUP start option can be placed in a start list (ATCSTR00 or ATCSTRxx) to indicate the processing that occurs if a start option in the list is not valid. You should place the LISTBKUP start option first in a start list, so that it will have been processed if an error is found for another start option.
- Start options entered by the operator. The VTAM operator can enter additional start options on the START command, and also during VTAM startup if prompted. (VTAM prompts for start options if the operator does not enter any start options on the START command.) To prevent VTAM from prompting for start options during startup, code the NOPROMPT option in ATCSTR00.

  Start options entered by the operator can override options in a specified supplemental start option list, as well as start options in the default ATCSTR00 list and the IBM-supplied internal default values.

  To enter a list of options longer than the length of the console, place a comma after the last start option that will fit on the console and VTAM will continue to prompt for more start options.
- Start options reentered by the system operator to correct errors in previously processed start options.

Start option processing

In “Sources of start options” the list of start option sources is arranged from the lowest priority to the highest. VTAM processes the start options with the lowest priority first and those with the highest priority last. If conflicting specifications exist for a particular start option, the last valid specification entered always overrides any previous specification. For example, if ATCSTR00 contains CONFIG=01, but the operator enters LIST=ST and ATCSTRST contains the start option CONFIG=02, VTAM activates the configuration defined by ATCCON02.

If VTAM detects an error when reading in a start option list, it notifies the operator. The operator must then do one of the following:

- Respond to the error message by selecting the “continue processing” option and then reenter any start options that were specified incorrectly in the start option list. The operator can also enter additional options at this time.

- Specify a different start option list to use by entering xx (for the desired ATCSTRxx start option list). This cancels the start attempt and then begins start processing again with another start option list.
- Halt VTAM. This cancels the start attempt so that you can correct the start option list and try again.

After reading in a start option list, if VTAM detects an error while processing a specific start option, the following occurs:

- Backup processing occurs if the LISTBKUP start option was processed and the start list in error is not a backup start list (only one backup start list can be processed).
- If the LISTBKUP start option was not processed, the operator is provided with the same three choices available when VTAM encounters an error while reading in a start list.

Use the LISTBKUP start option to specify the following:

- Another start list should be used in place of the start list in error (LISTBKUP=xx).
- All start option values existing before the file in error was processed should be used (LISTBKUP=DEFAULTS).
- All valid start options in the file in error will be processed, and incorrectly specified start options should remain as previously set until VTAM reprompts for these options (LISTBKUP=PROMPT).

Creating start option lists

To use a start option list, create a list and put it in a partitioned data set member named ATCSTRyy in SYS1.VTAMLST. The *yy* value must be the same as the *yy* value in the LIST=*yy* start option entered by the operator.

When the VTAM operator enters LIST=*yy* as a start option, VTAM first attempts to locate ATCSTR00. If ATCSTR00 does not exist, VTAM sends a warning message to the operator. To avoid receiving this message, create an ATCSTR00 file that contains only comments or start options that are always used for that particular VTAM.

Following is a sample start list ATCSTR*yy*.

```
SSCPID=01
SSCPNAME=SSCP1A
NETID=NETA
DYNASSCP=YES
SSCPDYN=YES
HOSTSA=1
```

The operator can start VTAM with this list using LIST=*yy* on the start command.

One start list cannot refer to another using LIST=yy. Only the operator can enter LIST=yy to use a start list.

Improving VTAM performance using start options

You can use the following information to improve performance using start options:

- VTAM search reduction support can be used with the SRCHRED, SRTIMER, and SRCOUNT start options.
- The NCPBUFSZ start option can be used to improve performance when you load or dump a link-attached 3720 or 3745 communication controller by specifying a request limit size.

Your use of start options also affects VTAM use of host processor time and storage. Buffer pool specifications, the limit of buffer usage for session outage notification, and the number of subtasks used to perform dump, load, and restart processes impacts VTAM host resources. For more information about these start options and other factors that impact VTAM performance in your network, see Chapter 23, "Tuning VTAM for your environment," on page 663.

Reducing network search overhead using search reduction: When a resource is unreachable in a network, futile attempts to reach it can still occur. Excessive searching for unreachable resources can adversely affect network performance. Therefore, VTAM provides search reduction support, which limits requests for unreachable resources.

If search reduction support is enabled, an APPN or subarea search reduction entry is created when a resource discovery search (RDS) fails to locate a target resource. An APPN search reduction entry is saved as a directory entry in the directory services database. A subarea search reduction entry is saved as a CDRSC in the subarea resource definition table (RDT). VTAM then reduces searching for the unreachable resource in the appropriate part of the network during a designated amount of time or until a designated number of requests have been received for the resource.

If a subarea search reduction entry exists for a destination resource and VTAM is the SSCP of the originating LU (including instances where VTAM is an interchange node for requests entering subarea from APPN), search reduction optimizes adjacent SSCP table searching for that resource. VTAM limits attempts during the designated period to locate the unreachable resource in the subarea network.

If an APPN search reduction entry exists for a destination resource, the scope of optimization varies according to the node type of VTAM. This depends on the following:

- If VTAM is the central directory server or network node server for the originating LU, including instances where VTAM is an interchange node for requests entering APPN from a subarea, search reduction optimizes APPN searching in VTAM's subnetwork. VTAM limits attempts during the designated period to locate the unreachable resource in its APPN subnetwork. For a discussion of APPN subnetworks, see "APPN multiple network connectivity" on page 89.
- If VTAM is a border node for requests received across a subnetwork boundary, search reduction optimizes APPN searching in the VTAM subnetwork. VTAM limits attempts during the designated period to locate the unreachable resource in its APPN subnetwork.
- If VTAM is a border node for the originating LU, search reduction optimizes APPN searching throughout the APPN network. VTAM limits attempts during the designated period to locate the unreachable resource in the APPN network.

To use VTAM search reduction support, code the SRCHRED=ON start option or enter the MODIFY VTAMOPTS command with the SRCHRED=ON operand. Use the SRTIMER start option, or MODIFY VTAMOPTS with the SRTIMER operand, to designate the amount of time that VTAM should limit requests for a resource. Use the SRCOUNT start option, or MODIFY VTAMOPTS with the SRCOUNT operand, to designate the number of requests for a resource that VTAM should limit before attempting to locate the resource again.

In determining a proper value for SRTIMER or SRCOUNT, keep in mind that the value should be large enough to limit searching for an appropriate amount of time, but small enough so that a resource that becomes available again does not remain unreachable due to waiting for SRTIMER or SRCOUNT to expire. Also, keep in mind that the SRCLEAR=YES operand can be used on the MODIFY RESOURCE command to clear search reduction entries. That is, you can use this when a resource becomes available to clear search reduction entries, so that the specified resource can now be located.

Note that SRCLEAR=YES only clears search reduction entries; it does not turn off search reduction support. If a subsequent RDS fails to locate the target resource, VTAM creates a new search reduction entry and limits subsequent requests for the unreachable resource during the designated amount of time or until the designated number of requests have been made.

Notes:

1. The SRTIMER and SRCOUNT operands can also be specified for resources on the CDRSC definition statement or the GROUP definition statement in a CDRSC major node. These values can be dynamically modified using the MODIFY RESOURCE command.
2. The DISPLAY DIRECTRY command displays information about search reduction entries in the directory services database. The DISPLAY ID command displays information about search reduction entries for a particular CDRSC, as well as entries in the directory services database (if the IDTYPE=RESOURCE operand is used).

   When a resource has search reduction information associated with it, these commands display the defined values for SRTIMER and SRCOUNT, and the remaining amount of time and remaining number of searches before VTAM discards the search reduction information.
3. Use DISPLAY VTAMOPTS for information on the settings of the search reduction start options.
4. When using SRTIMER, SRCOUNT, or both, along with AUTOTI for autologon sessions, an unsuccessful autologon-session search for a controlling PLU can cause the creation of a search reduction entry for it. In this situation, be aware of the following:
   - If AUTOTI=x and SRTIMER=y, where y is greater than x, when the AUTOTI timer expires, subsequent requests for the PLU are limited by the search reduction entry until SRTIMER also expires.
   - If AUTOTI=x and SRCOUNT=z, when the AUTOTI timer expires, subsequent requests for the PLU are limited by the search reduction entry for (x * z) seconds. At that time, the search reduction entry expires.

   See “Controlling searches” on page 519 for additional information.

Improving VTAM performance using the NCPBUFSZ start option: To improve performance when you load or dump a link-attached 3720 or 3745 communication controller, you can specify the request unit (RU) size used to transfer either the load module to or the static dump from the communication controller. The default size is 512 bytes, but values of 1024 or 2048 can be specified on the NCPBUFSZ start option to increase buffer size and improve the performance of the load or dump operation.

VTAM determines the RU size to be used for a dynamic dump. VTAM always attempts to use the 2K transfer for a dynamic dump but uses a 256-byte transfer under any of the following conditions:

- The SSP does not support a 2K buffer for the transfer operation.
- The NCP does not support a 2K transfer operation.
- The lines between VTAM and NCP do not support the use of 2K path information units (PIUs).

Increasing host subarea element addresses

For APPN resources, VTAM maintains a table of network addresses used only by VTAM. To increase the number of APPN sessions VTAM can support, VTAM allows extended host element addressing using the ENHADDR=YES start option. If ENHADDR=YES is specified, high-order addresses are assigned for the primary LU (and often the secondary LU as well) when the addresses of both the primary LU and secondary LU are assigned by the same VTAM (not by an NCP). This includes only sessions where both session partners are either a VTAM application or an LU that is attached to VTAM (not to an NCP). The maximum number of extended address elements that can be assigned is 33 161 216. To monitor the number of extended addresses currently in use at any time, use the D NET,STATS,TYPE=VTAM command. Stat 165 displays such a count in decimal format.

To further extend the usage of enhanced host element addressing, high-order element addresses are assigned for EE PUs, EE lines, RTP PUs, and DLUR-served PUs (when possible). This is done regardless of the coding of the ENHADDR start option.

Note: The extended and base address in use counts (stats 164 and 165 from the D NET,STATS,TYPE=VTAM command) might be one greater than the respective highwater values (stats 161 and 162 from the D NET,STATS,TYPE=VTAM command). This is because the addresses specified by the highwater values are actual addresses from a pool beginning with value 0, while the address counts are actual counts of addresses in use from that pool of addresses.

Enhanced addressing for session managers

To further extend enhanced host element addressing, applications that are activated within an application program major node can be assigned a high-order base address. When an LU-LU session begins for the application, the addressing capabilities of the partner LUs are assessed. If either partner does not permit use of a high-order address, a low-order auxiliary address is used.

The extent to which high-order addresses are used is determined by the parallel session capability of the application and the type of host that the application belongs to.

If the host is not pure APPN (in other words, supports subarea), applications that are not parallel session-capable will be assigned a high-order address when the application major node is activated and a low-order address is acquired when an LU-LU session involving the application requires a low-order address.

Note: Session manager applications are not typically parallel session capable.

If the host is pure APPN, all application addresses will be high-order addresses regardless of its parallel session capability.

This enhancement increases the number of addresses available for sessions and reduces the instance of sessions failing because of insufficient network addresses.

Diagnosing problems using start options

The following start options provide help with problem diagnosis. You can implement these when you start VTAM.

ASIRFMSG Enables you to specify the SSCPs in which to issue messages IST890I and IST896I when an autologon session initiation request fails.

BSCMDRS Enables you to specify which BSC miscellaneous data records are to be recorded in LOGREC.

DSIRFMSG Enables you to specify the SSCPs in which to issue messages IST663I, IST664I, IST889I, and their message groups when searches that might not result in a session fail to locate the target resource.

Recommendation: Due to the volume of messages that can be generated, it is recommended that you disable this option during normal operation. It is recommended that you enable this option (using the MODIFY VTAMOPTS command) on all necessary hosts only when trying to diagnose specific problems. After the problem has been diagnosed or documentation has been collected, you should again disable this option (using the MODIFY VTAMOPTS command).

Tip: When SIRFMSG is coded on CDRSC definition statements, the DSIRFMSG function (for DLU resources) and SIRFMSG function (for OLU and DLU resources) are executed. This can be helpful in diagnosing problems involving those CDRSC resources.

ESIRFMSG Enables you to specify the SSCPs in which to issue messages IST890I, IST892I, and IST893I when a session initiation request fails and extended sense data exists.

FSIRFMSG Enables you to specify the SSCPs in which to issue additional search routing failure messages when sense data exists. When used in conjunction with the SIRFMSG or DSIRFMSG option, the FSIRFMSG option controls whether messages IST894I, IST895I, IST1704I, and IST1705I are issued. When used in conjunction with the LSIRFMSG option, the FSIRFMSG option controls whether messages IST2208I, IST2209I and IST1942I through IST1953I are issued.

INOPCODE Enables a specific inoperative condition, if experienced by any resource, to take a VTAM dump.

Restriction: Both the resource (INOPDUMP) and condition (INOPCODE) must be enabled in order for a dump to be taken.

INOPDUMP Enables a resource, if and when it experiences an inoperative condition, to take a VTAM dump.

Restriction: Both the resource (INOPDUMP) and condition (INOPCODE) must be enabled in order for a dump to be taken.

IOINT Enables you to request that VTAM inform the operator when a VTAM request is not answered after a specified interval of time. After the period of time specified by IOINT, VTAM issues a message to the operator. If VTAM issues the message multiple times for the same request or several different requests, initiate the appropriate diagnostic actions to resolve the problem.

LSIRFMSG Enables you to specify the network nodes in which to issue

messages IST663I, IST664I, IST889I, and their message subgroups when APPN locates fail to locate the target resource.

Recommendation: Due to the volume of messages that can be generated, it is recommended that you disable this option during normal operation. It is recommended that you enable this option (using the MODIFY VTAMOPTS command) on all necessary hosts only when trying to diagnose specific problems. After the problem has been diagnosed or documentation has been collected, you should again disable this option (using the MODIFY VTAMOPTS command).

Tip: When SIRFMSG and CPNAME are coded on CDRSC definition statements, the SIRFMSG setting will overide the LSIRFMSG start option (for DLU resources). This can be helpful in diagnosing problems involving those CDRSC resources.

MSGMOD Enables you to see the name of the module from which a message is issued. If you specify MSGMOD=YES, VTAM inserts the last five characters of the module name that issued the message, following the VTAM message identifier (for example, ISTnnnI). However, using MSGMOD causes the last five characters of the message to be truncated if the message exceeds the maximum console message length allowed by the operating system.

NMVTLOG Enables you to specify whether error records are written to the operating system's error data set (LOGREC). If NMVTLOG=NPDA (the default), VTAM records alerts in LOGREC only when the NetView hardware monitor, NPDA, is not active. If NMVTLOG=ALWAYS, VTAM always records alerts in LOGREC. If NMVTLOG=NEVER, VTAM never records them.

PPOLOG Enables you to specify whether to send operator commands and their resulting messages to the primary program operator for logging.

PSSTRACE Enables you to specify whether to record IRB and SRB entries in the VTAM internal trace table when the PSS trace option is in effect.

RSIRFMSG Enables you to specify the SSCPs in which to issue messages IST1460I, IST1461I, IST2102I, IST2103I, and IST2104I to display the RSCV when a session initiation request fails and RSCV data exists.

SDLCMDRS Enables you to specify whether SDLC statistical miscellaneous data records (MDRs) are written to LOGREC. If SDLCMDRS=YES (the default), VTAM records SDLC statistical MDRs in LOGREC.

SIRFMSG Enables you to specify the SSCPs in which to issue messages IST663I, IST664I, IST889I, and their message groups when a session initiation request fails.

Note: SIRFMSG can also be coded on the APPL and CDRSC definition statements to help diagnose problems involving those resources. Coding SIRFMSG on a CDRSC definition statement also enables the DSIRFMSG function and the LSIRFMSG function (if CPNAME is also coded) for that resource.

SNAPREQ Enables you to specify the number of requests for VTAM buffers between snapshot dumps.

TNSTAT Enables you to specify whether, where, and how frequently to record tuning statistics.

TRACE Enables you to collect information, such as data that flows through VTAM buffers, data transmitted on NCP links, and VTAM internal trace information. For more information about the trace facility, refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.

Refer to *z/OS Communications Server: SNA Operation* for a list of the start options and descriptions of the operator commands that you can use to start VTAM trace facilities. The ASIRFMSG, ESIRFMSG, FSIRFMSG, and RSIRFMSG start options produce messages only when the SIRFMSG, DSIRFMSG, or LSIRFMSG start options are also in use. For descriptions of the start options, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Configuration lists

A configuration list specifies the resources that are to be activated when you start VTAM. Place the member names of the resources you want to have activated when VTAM starts into an ATCCON*xx* member in the VTAM definition library, where *xx* is any two alphanumeric characters. The value *xx* can then be used on the CONFIG operand of the VTAM START command, or on the CONFIG start option in your start option list, to specify which definitions are to be activated at startup. In Figure 5, to use the configuration list defined in ATCCON01, you can specify CONFIG=01 on the VTAM START command.

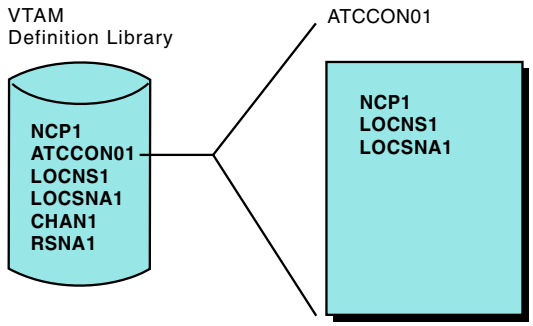


Figure 5. Configuration list

ATCCON01 contains the member names LOCSNA1, LOCNS1, NCP1, which will be activated when VTAM starts. CHAN1 and RSNA1 (found in the VTAM definition library) are not included in ATCCON01, and therefore must be activated manually by the VTAM operator using the VARY command, if they are to be used. If no configuration lists exist, the VTAM operator must use the VARY command to activate all resources.

Following is sample coding for configuration list ATCCON01.

```
LOCSNA1
LOCNS1
NCP1
```

Examples of resource definitions that can be included in a configuration list are:
- Paths
- Major nodes
- Minor nodes defined in previously listed major nodes
- Tables

- Dynamic reconfiguration files
- Network node server list

You can also use a configuration list to activate an NCP using a different PU name than the one specified in the NEWNAME operand on the BUILD definition statement. For example, if you want to load a communication controller with load module X3745A and want the PU name to be NCPA1, you would code the following in your configuration list:

```
NCPA1/X3745A
```

This would have the same result as using the following command:

```
VARY NET ACT,ID=NCPA1,LOADMOD=X3745A
```

Each of the names specified in the configuration list must be one to eight characters (one to seven characters for the load module name when specifying pu_name/load_module). The first character of each name must be alphabetical (A–Z) or one of the national characters (@, #, or $). The remaining characters of each name must be alphabetical (A–Z), numerical (0–9), or one of the national characters (@, #, or $). If a name is found that is not valid, VTAM stops processing the configuration list and is unable to initialize any of the subsequent major nodes.

Identifying resources to VTAM

In addition to specifying start options and coding configuration lists, you'll need to identify resources in the network to VTAM. Depending on your network, you might need to define a combination of the following resources:

- Application programs
- Adjacent control points
- NCPs
- Peripheral nodes
- LUs
- Other VTAMs

Any of these resources may be predefined to VTAM using a static definition statement. For example, applications are predefined using an application major node and application (APPL) statements; switched PUs are predefined using the NCP major node for the lines and switched major node for the PUs and LUs. In addition, many VTAM resources may be dynamically defined as VTAM learns of them. A switched PU may be defined when it dials in; an adjacent CP may be defined when the connection between the nodes is activated.

Coding concepts

This section describes basic coding concepts needed to define resources. For further information, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

To define your resources:

1. Code VTAM and NCP definition statements to define your major nodes.
   The following restrictions apply to coding your definition statements:
   - Names must be unique within a network.
   - Names cannot begin with “IST” or any of the names “VTAM”, or “VTAMSEG”, or “VTAMSG2”. Also, the name “VTAMTERM” cannot be used.
   - Code each minor node and major node name in column 1.

- Code each definition statement in column 10.
- Code each operand in column 16 or one space after the definition statement.
- If your operands are longer than one line, code a continuation character in column 72 and continue operands in column 16 of the next line.

Following is a sample definition file KIMAP VTAMLST (the scale is included to show coding restrictions, such as the continuation character in column 72):

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
KIMAP    VBUILD TYPE=APPL
AP1      APPL  AUTH=(PASS,ACQ),                                        X
               ACBNAME=ECHO
AP2      APPL  ACBNAME=ECHO
```

2. If you are defining NCP, code the NCP-only definition statements and operands as described in the NCP resource definition documents.
3. Generate the NCP as described in the *NCP, SSP, and EP Generation and Loading Guide*. If you are updating VTAM-only definition statements or operands, update the NCP definitions in VTAMLST; no NCP generation is required. If you are updating NCP-only definition statements or operands, update the NCP generation definition and generate NCP again, or use dynamic reconfiguration facilities.

   Note: If you need to make changes before you can regenerate the NCP, VTAM and NTuneNCP provide functions to allow some NCP definitions to be dynamically changed.
4. Store your definitions in the proper VTAM library. You can concatenate VTAMLST files, but either the block sizes must be the same for all files or the file with the largest block size must be the first file in the concatenated list.

Sift-down effect

The *sift-down* effect enables you to code an operand on a higher-level node so that you do not need to recode it on each lower-level node for which you want the same value. As a result, the sift-down effect greatly simplifies the coding process.

The *z/OS Communications Server: SNA Resource Definition Reference* identifies and describes the definition statements and operands to which sifting applies. For information about definition statement sequencing and the sift-down level for each NCP operand, refer to the NCP resource definition documents.

Using MVS system symbols

Multiple virtual storage (MVS) system symbols enable you to reduce the number of VTAM definitions you must code in VTAMLST. These definitions include:

- Start option lists
- Configuration lists
- Major nodes and minor nodes
- Routing and dynamic reconfiguration definitions
- The following user-definable tables:
  - APPN class of service (COS) definitions
  - APPN-to-subarea COS mapping table
  - Associated LU table
  - Message-flooding prevention table
  - Model name table
  - Subarea-to-APPN COS mapping table
  - SAMAP

Substitution text for the symbols is defined in MVS and is substituted by MVS in place of the symbols during VTAM startup, major node activation, and table activation.

By using MVS system symbols in VTAMLST, you can code a single start option list and a single configuration list that can be used to start VTAM on multiple systems. You can also code one set of major nodes, routing and dynamic reconfiguration definitions, and user-definable tables. You can define and maintain a single VTAM for use on multiple systems.

You can also use MVS system symbols in the TSOKEY00 parmlib member.

In addition, you can use MVS system symbols in VTAM network operator commands and Transaction Processing Facility (TPF) logon manager operator commands. Substitution text for the symbols is defined in MVS and is substituted by MVS in place of the symbols during command processing. This eliminates the need for the operator to know the exact name of a resource when issuing a command on a particular system.

You can use MVS system symbols in non-sysplex and sysplex environments.

Overview

Before coding MVS system symbols in VTAMLST, TSOKEY00, or in VTAM network operator commands or TPF logon manager operator commands, you might want to issue the DISPLAY SYMBOLS command on each system on which you plan to use the symbols. The DISPLAY SYMBOLS command is an MVS command that displays defined symbols and their substitution text for the system on which the command is issued. Refer to *z/OS MVS System Commands* for information about how to use the DISPLAY SYMBOLS command.

Additional installation-defined symbols can be defined if you need them.

Consider how to best use the defined symbols and any additional symbols you might want to define. System symbols in VTAMLST are most useful for defining system identifiers and names when coding:

- Start options, especially those that specify system names and identifiers; for example, CONFIG, SSCPID, and SSCPNAME.
- A configuration list. Use the symbols in the names of the major node members contained in the configuration list.
- Definitions for major nodes and minor nodes. Use the symbols in the names of definition statements and in the values you specify on operands on the definition statements.
- Definitions for routing and dynamic reconfiguration. Use the symbols in the names of definition statements and in the values you specify on operands on the definition statements.
- The following user-definable tables:
  - APPN Class of Service (COS) definitions
  - APPN-to-subarea COS mapping table
  - Associated LU table
  - Message-flooding prevention table
  - Model name table
  - Subarea-to-APPN COS mapping table

  Use the symbols in the names of macroinstructions and in the values you specify on operands on the macroinstructions.

To use MVS system symbols in VTAM network operator commands and TPF logon manager operator commands, use the symbols in the values you specify on command operands, especially those that specify resource names (for example, ID). You can also use the static system symbols shown in Table 3.

Table 3. MVS static system symbols that can be used in VTAM

Symbol name	Description	Length of text	Where defined	Default substitution text
&SYSCLONE	Shorthand notation for the name of the system; often used in fields that are limited to two characters.	1–2 characters	IEASYMxx parmlib member	Last two characters of substitution text are defined to &SYSNAME symbol.
&SYSNAME	The name of the system	1–8 characters	IEASYMxx or IEASYSxx parmlib member	The processor identifier. Refer to the *z/OS MVS Initialization and Tuning Reference* and the *z/OS MVS Initialization and Tuning Guide* for information about the processor identifier.
&SYSPLEX	The name of the sysplex	1–8 characters	COUPLExx or LOADxx parmlib member	If LOADxx does not specify the sysplex name, &SYSPLEX defaults to LOCAL until the COUPLExx member is processed.
&*installation-defined system symbol*	Up to 100 system symbols defined by your installation	1–8 characters	IEASYMxx parmlib member	None.

During VTAM startup, major node activation, resource activation, table activation, and command processing, MVS substitutes the text defined for these system symbols wherever the symbols appear in VTAMLST, TSOKEY00, and in commands.

Note: Other products, such as NCP and NetView, use VTAMLST to obtain certain definitions. Refer to the other products' publications to determine how they support MVS system symbols in VTAMLST.

Coding guidelines

Substitution text for system symbols is defined in MVS. For complete information about how to define the symbols, including default substitution text, refer to the *z/OS MVS Initialization and Tuning Reference* and the *z/OS MVS Initialization and Tuning Guide*. This section provides guidelines for coding the symbols in VTAMLST.

The symbols must begin with an ampersand (&) and should end with a period (.). However, if MVS finds a system symbol that does not end with a period, it substitutes text for the symbol when the next character is one of the following:

- Null (the end of text is reached)
- A character that is not alphabetical, numeric, or one of the national characters, @, #, or $

For example, if &SYSCLONE is defined in MVS to equal 01—SYSCLONE(01)—and &SYSNAME is defined to equal HOST01A—SYSNAME(HOST01A)—, you could code the symbols in VTAMLST as follows:

```
...CONFIG=&SYSCLONE.,SSCPNAME=HOST&SYSCLONE.A
```

or:

```
...CONFIG=&SYSCLONE.,SSCPNAME=&SYSNAME
```

It is recommended that you always end MVS system symbols with a period. Otherwise, you risk a higher incidence of syntax errors.

Note: For ease of reading, periods are not shown at the end of symbols in this document, except when the symbols are used in examples.

A symbol name can have one to eight characters between the ampersand and the period.

After MVS substitutes the symbols with the text defined in MVS, VTAM will ensure that the resulting values are valid. Current VTAM coding conventions still apply when MVS system symbols are used in VTAMLST. So, for example, if &SYSCLONE is defined in MVS to equal 01, and CONFIG=&SYSCLONE is coded in VTAMLST, VTAM accepts the resolved value, CONFIG=01. If, however, CONFIG=01&SYSCLONE is coded, VTAM rejects the resolved value, CONFIG=0101, because the value specified on the CONFIG start option must be two characters in length. For information about current VTAMLST coding conventions, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Example of VTAM definitions in a sysplex environment without the use of MVS system symbols

Following is an example of the VTAM definitions required to enable VTAM to run on two MVS hosts in a sysplex without the use of MVS system symbols.

Two MVS hosts, HOST01 and HOST02, exist in a sysplex. Each host starts VTAM and is running the same VTAM application programs on its system. The following members are defined in the VTAMLST data set to enable this configuration:

```
ATCSTR01

  CONFIG=01,
  SSCPID=01,
  SSCPNAME=HOST01,
   ⋮

ATCSTR02

  CONFIG=02,
  SSCPID=02,
  SSCPNAME=HOST02,
   ⋮

ATCCON01

  APPLS01
```

```
  ⋮

ATCCON02

  APPLS02
  ⋮

APPLS01

  APPL01A  APPL ...
  APPL01B  APPL ...
  APPL01C  APPL ...
  ⋮

APPLS02

  APPL02A  APPL ...
  APPL02B  APPL ...
  APPL02C  APPL ...
  ⋮
```

VTAM is started from MVS HOST01 with the following command:

```
START VTAM,LIST=01
```

VTAM is started from MVS HOST02 with the following command:

```
START VTAM,LIST=02
```

Two separate start option list members are defined in VTAMLST to enable VTAM to be started on both systems:

- ATCSTR01, which specifies names and identifiers to be used when VTAM is started on HOST01
- ATCSTR02, which specifies names and identifiers to be used when VTAM is started on HOST02

Each start option list specifies the name of a configuration list to be used when VTAM is started on that system. Two configuration list members are defined in VTAMLST:

- ATCCON01, which specifies the major nodes to be activated when VTAM is started on HOST01
- ATCCON02, which specifies the major nodes to be activated when VTAM is started on HOST02

Even though identical application programs are to be activated on both systems, because they must have unique names within the sysplex, they must be defined in two separate major nodes with unique names. One major node is specified in one configuration list and the other major node is specified in the other configuration list.

Three application programs are defined in major node APPLS01 with the following names:

- APPL01A
- APPL01B
- APPL01C

Identical application programs are defined in major node APPLS02 with the following names:

- APPL02A
- APPL02B
- APPL02C

APPLS01 is specified in configuration list ATCCON01, indicating that VTAM should activate the major node defined by APPLS01 when VTAM is started on HOST01. Within member APPLS01, the APPL definition statements, APPL01A, APPL01B, and APPL01C are defined, indicating that the application programs defined by these statements should be activated when the major node defined by APPLS01 is activated.

APPLS02 is specified in configuration list ATCCON02, indicating that VTAM should activate the major node defined by APPLS02 when VTAM is started on HOST02. Within member APPLS02, the APPL definition statements APPL02A, APPL02B, and APPL02C are defined, indicating that the application programs defined by these statements should be activated when the major node defined by APPLS02 is activated.

You can see that the more systems you have in a sysplex environment, each running identical VTAM application programs, the more VTAM definitions you must code in VTAMLST.

Using MVS system symbols in a sysplex environment

Following is an example of how the definitions in the previous section can be reduced by using MVS system symbols.

The following members are defined in a VTAMLST data set that is accessible to two MVS systems, HOST01 and HOST02, in a sysplex:

```
ATCSTR00

  CONFIG=00,
  SSCPID=&SYSCLONE.,
  SSCPNAME=&SYSNAME.,
   ⋮

ATCCON00

  APPLS
   ⋮

APPLS

  APPL&SYSCLONE.A  APPL ...
  APPL&SYSCLONE.B  APPL ...
  APPL&SYSCLONE.C  APPL ...
   ⋮
```

How the symbols resolve when HOST01 starts VTAM:

The following substitution text is defined in MVS HOST01:

```
SYSCLONE(01)
SYSNAME(HOST01)
```

VTAM is started from MVS HOST01 with the following command:

```
START VTAM,LIST=00
```

LIST=00 is not required on the START command because it is the default. VTAM substitutes every instance of &SYSCLONE in VTAMLST with 01 and every instance of &SYSNAME with HOST01. As a result, VTAM uses the following definitions:

```
ATCSTR00

  CONFIG=00
  SSCPID=01,
  SSCPNAME=HOST01,
```

```
  ⋮
ATCCON00

  APPLS
  ⋮

APPLS

  APPL01A  APPL ...
  APPL01B  APPL ...
  APPL01C  APPL ...
  ⋮
```

How the symbols resolve when HOST02 starts VTAM:

The following substitution text is defined in MVS HOST02:

```
SYSCLONE(02)
SYSNAME(HOST02)
```

VTAM is started from MVS HOST02 with the following command:

```
START VTAM,LIST=00
```

LIST=00 is not required on the START command because it is the default. VTAM substitutes every instance of &SYSCLONE in VTAMLST with 02 and every instance of &SYSNAME with HOST02. As a result, VTAM uses the following definitions:

```
ATCSTR00

  CONFIG=00
  SSCPID=02,
  SSCPNAME=HOST02,
  ⋮

ATCCON00

  APPLS
  ⋮

APPLS

  APPL01A  APPL ...
  APPL01B  APPL ...
  APPL01C  APPL ...
  ⋮
```

Using MVS system symbols in a non-sysplex environment

If multiple VTAMs are not in a sysplex environment, but are nonetheless related (for example, in a state university environment where one VTAM is started on campus A and one VTAM is started on campus B) one VTAMLST data set might be copied from the campus A system and installed and used to start VTAM on the campus B system. MVS system symbols can be used in this situation to reduce the system definitions in VTAMLST, much like they are used in the sysplex example 44. Instead of being two MVS hosts in a sysplex, HOST01 and HOST02 might be the two hosts in the state university environment, HOST01 on campus A and HOST02 on campus B.

Verifying a VTAM network

This section suggests steps to follow to verify that your VTAM network functions properly. Before you perform these steps, make sure that the status of all resources in your network is inactive.

For specific information about how to perform these steps and for the syntax of operator commands, refer to *z/OS Communications Server: SNA Operation*. If these steps do not yield the expected results, double-check the definition statements for the resources that are affected. If this does not solve the problem, refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.

To verify that your VTAM network functions properly, do the following:

1. Start VTAM and the appropriate trace facility (such as GTF).
2. Activate a channel-attachment major node, begin tracing, and activate lines and peripheral physical units.
3. Activate an NCP, begin tracing and intensive mode error recording, and verify that you can dump the NCP.
4. Start the NetView program, if it is installed.
5. Activate a switched major node, and begin tracing for the physical units.
6. Start the VTAM or host subsystem application programs.
7. Start VTAM traces, and activate the logical units at the peripheral physical unit.
8. Use DISPLAY commands to verify:
   - Peripheral terminals and links
   - Route structure
   - Sessions between application programs and logical units
9. Activate and verify additional physical units and NCPs.
10. Simulate normal operation with all physical units.
11. Try backup and recovery procedures.
12. Halt VTAM.

Verifying a multiple-domain subarea network

This section suggests steps to follow to verify that your multidomain subarea network functions properly. However, before you perform the steps in this section, you must perform the steps "Verifying a VTAM network" for each domain in your network. After you have done this, make a list of the most important normal and backup operations in your network, and verify each of them one at a time. Then verify the network over a length of time, as in normal operation.

To verify that your multiple-domain, subarea network functions properly do the following:

1. Activate cross-domain paths and links, cross-domain resource managers (CDRMs), and cross-domain resources (CDRSCs).
2. Verify cross-domain sessions by exchanging requests across domains, between terminals and hosts, and between application programs.
3. Verify the NetView program, if it is installed.
4. Verify host backup using a channel-attached, shared NCP.
5. Verify host backup over a cross-domain link.
6. Verify automatic SSCP-SSCP session restart.
7. Halt VTAM in each domain.

Verifying a multiple-network environment

This section suggests steps to follow to verify that your multiple-network environment functions properly. However, before you perform the steps in this section, perform the steps in both "Verifying a VTAM network" on page 46 and "Verifying a multiple-domain subarea network" on page 46 (in that order) for each of your networks.

To verify that your multiple-network environment functions properly, do the following:

1. Activate each gateway VTAM and gateway NCP.
2. Activate major nodes required for cross-network sessions.
3. Establish required cross-network SSCP-SSCP sessions.
4. Verify cross-network sessions by exchanging requests across networks, between terminals and hosts, and between application programs.
5. Verify termination processing when an SSCP-SSCP or SSCP-gateway NCP session is deactivated.
6. Try backup and recovery procedures.
7. Verify the NetView program, if it is installed.
8. Halt VTAM in each domain in each network.

Verifying a VTAM APPN network

To verify that your VTAM APPN network functions properly, follow these steps:

1. Start VTAMs with appropriate APPN start options.
2. Activate NCP major node, XCA major node for 3172 connection, or an OSA connection.
3. Activate CP-CP session-capable APPN connections (for example, lines and PUs) to adjacent nodes and verify that CP-CP sessions are automatically started. The following message is issued:

   ```
   IST1096I CP-CP SESSIONS WITH adjcpname ACTIVATED
   ```

4. Display adjacent control points. For example, if VTAM was started with DYNADJCP=YES, issue:

   ```
   D NET,ID=ISTADJCP,E.
   ```

5. From VTAM network nodes, display the topology of the APPN network. For example:

   ```
   D NET,TOPO,ID=cp_name,LIST=ALL.
   ```

6. From VTAM end nodes, display the network node server list. For example:

   ```
   D NET,NETSRVR,E.
   ```

7. Terminate one of the CP-CP sessions. For example, issue:

   ```
   V NET,TERM,LU1=cp_name,LU2=adjcp_name.
   ```

8. Reestablish the CP-CP session. For example, issue:

   ```
   V NET,ACT,ID=adjcp_name,IDTYPE=CP.
   ```

9. To verify end node registration, first activate a resource defined with REGISTER=NETSRVR on the end node (note that application program minor nodes default to REGISTER=NETSRVR). Then display the resource from the network node server directory database. For example, issue:

   ```
   D NET,DIRECTRY,ID=resource_name.
   ```

10. Verify network connectivity with any LU 6.2 resource (including control points). For example, issue:

    ```
    D NET,APING,ID=resource_name.
    ```

11. From VTAM network nodes, verify that an SSCP, CP, or LU in the network can be reached, by initiating a search. Also verify that a particular name is not duplicated in the network, which might result in session failures. For example, issue:

    ```
    D NET,DIRECTRY,ID=resource_name,SCOPE=NSEARCH.
    ```

12. Verify sessions by exchanging requests between APPN nodes.
13. Halt VTAM. Note that the HALT or HALT,QUICK commands checkpoint the topology and routing services database.

Chapter 4. Connecting an APPN node to VTAM

This topic includes information about the following:

- Connections through boundary function-based transmission groups
- Multiple Connections with Parallel Transmission Groups
- Channel connections between APPN nodes
- Leased connections between APPN nodes
- IBM 3172 Nways interconnect controller connections between APPN nodes
- APPN multiple network connectivity
- Virtual-Route-Based transmission groups
- Selecting the network node server for end nodes

For a node to communicate with other nodes as a peer, rather than as a participant in a hierarchical relationship, the node must be aware of other APPN nodes. VTAM nodes use a combination of the following available definitions and information to learn about adjacent nodes:

- Definition of the physical line or port
- Definition of the adjacent link station, either defined dynamically or predefined with PU definition statements
- Transmission group number
- Adjacent control point name

Before and during link activation between adjacent link stations, nodes also obtain information about the capability of an adjacent node through a message unit called an exchange identification (XID), which is used to convey node and link characteristics. XIDs are exchanged by adjacent nodes to establish and negotiate link and node characteristics. After link activation, XIDs are used to communicate changes in these characteristics and to negotiate the TG number used to represent the connection.

APPN nodes are connected using type 2.1 connections. APPN connections provide the same benefits as LEN connections. With APPN connections, you do not need to define CDRMs, PATH definitions, or ADJSSCP tables. You can dynamically define switched connections and independent logical units, and the connectivity of applications and dependent LUs is not limited.

Notes:

1. If you use VR-based transmission groups, CDRMs, PATH definitions, and ADJSSCP tables must be considered.
2. When changing the node roles of nodes in the network, a node should be taken down using normal takedown procedures. Forcing a node down and then bringing it up with a different node role can cause errors in the topology database, because error recovery might still be in progress when the node is brought back up.

© Copyright IBM Corp. 1984, 2011

Connections through boundary function-based transmission groups

Nodes are connected by transmission groups (TGs). A TG connects a pair of adjacent link stations across the transmission medium. A boundary function (BF)-based APPN TG is used to connect two APPN nodes that use the NCP or VTAM boundary function.

Transmission groups are described with the following standard APPN characteristics:

- Cost per connect time
- Cost per byte
- Security
- Propagation delay
- Effective capacity
- User-defined values

The values specified for these characteristics are saved in the topology database and are used to determine the weight of the TG for route calculation. The values you specify represent the characteristics of the link; the values do not affect the actual physical characteristics of the link. If the physical characteristics of the link change, the values specified for the TG characteristics should probably be changed to reflect the new physical characteristics. The topology database is discussed further under "Topology database" on page 18.

By default, all TGs assume the same characteristics. To distinguish one TG from another, TG characteristics can be specified in the following ways:

- As TG profiles, which are groups of TG characteristics that can be applied to several PUs
- On definition statements

The characteristics of TGs owned by other nodes are learned through topology database updates (TDUs).

Multiple connections with parallel transmission groups

Parallel TGs for multiple connections between APPN nodes are supported. This support means that a node can have multiple links to the same adjacent node. Parallel TGs have different TG numbers and can have different TG characteristics assigned to them. Multiple-link TGs (many links with one TG number) are not supported in APPN by VTAM. The support for parallel TGs matches that of most other APPN products.

If you have parallel TGs with CONNTYPE=APPN coded on the adjacent link stations, you can establish one or more APPN connections or one LEN connection between the two nodes. You cannot combine LEN and APPN connections or establish more than one LEN connection to the same adjacent node over APPN-capable TGs.

Support for multiple connections allows an independent LU in the APPN network to establish sessions through or into the subarea network by using several boundary-function TGs. An LU can have multiple sessions with one or more LUs through multiple boundary-function TGs and through different physical paths that function as parallel TGs. This support is also available for TGs attached to LEN nodes.

When coding CONNTYPE=APPN (or letting it default to that value), XID exchange rules for parallel TGs are enforced. This means that if you have PUs with duplicate CPNAMES, you should resolve this duplication before that node becomes an APPN node. If the node is to remain a LEN node, then CONNTYPE=LEN should be coded on the PU statement. Coding CONNTYPE=LEN results in VTAM avoiding the test for duplicate CPNAMEs.

Channel connections between APPN nodes

Type 2.1 channel connections can be used to connect adjacent APPN nodes. Adjacent APPN nodes can be other S/390® or zSeries hosts or other devices that support APPN channel connections.

The following types of channel connections are discussed in this section:
- Multipath channel connections
- Composite network node channel connections

Multipath channel connections

Multipath channel (MPC) connections allow you to code a single transmission group (TG) that uses multiple write-direction and read-direction subchannels. The subchannels may be assigned to one or more physical channels based on how you have defined your I/O subsystem to the operating system. Because each subchannel operates in only one direction, the half-duplex turnaround time that occurs with other channel-to-channel connections is reduced.

Levels of MPC capability provided by VTAM

VTAM provides the following levels of MPC capability:

- High-performance data transfer (HPDT)

  HPDT MPC connections provide more efficient transfer of data than non-HPDT MPC connections. They do this by using HPDT services to provide the following functions:

  – Data packing without data movement: This process decreases consumption of CPU cycles by reducing the internal movement of data, thus increasing the availability of MIPS (million instructions per second) for user processing.

  – Chain scheduling of channel programs: This process reduces operating system I/O invocations and CPU overhead.

  HPDT MPC can also run in HPDT Packing mode. HPDT Packing reduces data stream fragmentation. See “HPDT Packing” on page 56 for more information.

  HPDT services are available over connections to other nodes that implement HPDT MPC. Applications that use high-performance data transfer services rely on HPDT MPC connections for path length reductions and performance enhancements when sending and receiving data. For more information about high-performance data transfer services, refer to the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

  HPDT MPC uses the services of the communications storage manager (CSM). CSM is a storage facility that allows authorized host applications to set up data in buffers that can be addressed and accessed by other authorized host applications, eliminating the need to copy data as it is being prepared to be sent. CSM is installed with VTAM. Maximum CSM storage limits are defined in the CSM parmlib member, IVTPRM00. For more information about CSM, see “Communications storage manager (CSM)” on page 425.

- Non-HPDT

Non-HPDT MPC connections do not use HPDT services. Non-HPDT MPC connections can be considered synonymous with APPN host-to-host channel (AHHC) connections. See "Using MPC connections in an APPN network."

- Subarea

  Subarea MPC connections are described in "Multipath channel connections" on page 105.

Using MPC connections in an APPN network

You can use MPC connections in an APPN network to connect VTAM to adjacent APPN nodes and to connect VTAM to a port on the IBM Open Systems Adapter that provides native access to an ATM network.

How VTAM uses HPDT MPC connections in an APPN network

HPDT MPC is used by VTAM in an APPN network to enable the following connections:

- APPN node-to-node connections (ANNCs)

  ANNC connections can be between a VTAM APPN host node and an adjacent VTAM APPN host node or between a VTAM APPN host node and an adjacent nonhost APPN node.

  Connections between two VTAM APPN host nodes in a sysplex can also use the cross-system coupling facility (XCF). See "Dynamic definition of VTAM-to-VTAM connections" on page 445 for more information.

- ATM Native connections

  HPDT MPC enables a connection between a VTAM APPN host node and an IBM Open Systems Adapter port that provides native access to an ATM network. See "ATM native connections" on page 69 for more information.

How VTAM uses non-HPDT MPC connections in an APPN network

Non-HPDT MPC is used by VTAM to enable connections only between a VTAM APPN host node and an adjacent VTAM APPN host node. These connections are called APPN host-to-host channel (AHHC) connections. AHHC connections can be considered synonymous with non-HPDT MPC connections.

Steps involved in using HPDT and non-HPDT MPC connections in an APPN network

To use an MPC connection in an APPN network, you must complete the following steps:

1. Define the MPC connection.
2. Define the partner node.
3. Activate the resources that enable the MPC connection.

Note: If VTAM is using the functions of XCF in a sysplex, these steps can be performed dynamically. See "Dynamic definition of VTAM-to-VTAM connections" on page 445 for more information.

Defining an MPC connection:

You define an MPC connection in the transport resource list (TRL) major node. The characteristics of the connection are defined on a TRLE definition statement. You must code a TRLE definition statement for each MPC connection.

Defining the partner node:

As explained earlier, VTAM uses an MPC connection to connect to:

- An adjacent APPN node

  You define an adjacent APPN node (host or nonhost) in the local SNA major node. The TRLE operand on the PU definition statement in the local SNA major node specifies the name of the TRLE definition statement in the TRL major node. The TRL major node defines the MPC connection to be used between VTAM and the adjacent APPN node.

  For TCP/IP to use an MPC connection, the device name on the DEVICE statement must match either the MPC TRLE name (for MPCPTP devices) or the TRLE PORTNAME (for MPCIPA or ATM devices). TCP/IP cannot use MPC connections with an MPCLEVEL of NOHPDT. For more information, refer to the *z/OS Communications Server: IP Configuration Guide* and the *z/OS Communications Server: IP Configuration Reference*.

- A port on an IBM Open Systems Adapter that provides native access to an ATM network

  You define a port on an IBM Open Systems Adapter in the external communications adapter (XCA) major node. For information about how to define an IBM Open Systems Adapter port and how to specify the MPC connection to be used between VTAM and the port, see "Defining ATM native connections to VTAM" on page 69.

Activating the resources that enable the MPC connection:

After creating the TRL major node and the major node that defines the partner node (local SNA or XCA), you can use the MPC connection by activating the appropriate resources. For an MPC connection between VTAM and an adjacent APPN node, follow the steps below:

1. Activate the TRL major node.
2. Activate the local SNA major node.
3. Activate the PU that defines the connection to the adjacent node as an APPN PU.

For an MPC connection between VTAM and a port on an IBM Open Systems Adapter, see "ATM native connections" on page 69 for information on how to activate the appropriate resources.

Multiple TRL major nodes are allowed to exist in VTAM, and each can be dynamically modified using V NET,ACT,ID= with the UPDATE operand.

If UPDATE=ALL is specified, the VTAMLST member designated with the ID operand replaces the existing TRL major node. However, note that TRLEs in use are not deleted and remain in the TRL major node along with the new TRLEs.

Note: When there is only one active TRL major node, UPDATE=ALL deletes or replaces all inactive TRLEs. When there are multiple active TRL major nodes, UPDATE=ALL deletes or replaces only the inactive TRLEs that exist in the TRL major node being activated. If an inactive TRLE is defined in another TRL major node, it will not be deleted and a name conflict will result if an attempt is made to replace it later.

If UPDATE=ADD is specified, the TRLE entries in the VTAMLST member identified on the ID operand are added to the TRL major node as long as they do not already exist in any TRL major node.

MPC connection example

Figure 6 shows two VTAM network nodes connected through an MPC connection. Corresponding sample definitions follow Figure 6.

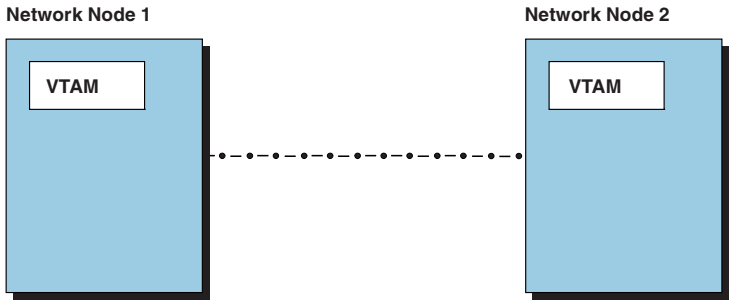


Figure 6. MPC connection between two VTAM network nodes

Definitions for network node 1	Definitions for network node 2
TRL Major Node TRLNN1 VBUILD TYPE=TRL MPCLN1 TRLE LNCTL=MPC, MAXBFRU=16, READ=(BCA), WRITE=(BCB) **Local SNA Major Node** LSNA1 VBUILD TYPE=LOCAL LSNAPU1 PU TRLE=MPCLN1, ISTATUS=ACTIVE, CONNTYPE=APPN, CPCP=YES	**TRL Major Node** TRLNN2 VBUILD TYPE=TRL MPCLN2 TRLE LNCTL=MPC, MAXBFRU=16, READ=(BCB), WRITE=(BCA) **Local SNA Major Node** LSNA2 VBUILD TYPE=LOCAL LSNAPU2 PU TRLE=MPCLN2, ISTATUS=ACTIVE, CONNTYPE=APPN, CPCP=YES

How the level of MPC is determined

The level of MPC used for an MPC connection is automatically determined by a negotiation between the partner nodes. You can control the level used by coding the MPCLEVEL operand on the TRLE definition statement that defines the MPC connection.

When HPDT MPC is used

If both partner nodes support HPDT MPC, HPDT MPC is automatically used.

VTAM supports HPDT MPC if it is defined as an HPR APPN node that provides RTP-level HPR support. See "High-Performance Routing (HPR)" on page 491 for information on how VTAM is defined as an HPR APPN node that provides RTP-level HPR support.

In Figure 6, assume that both VTAM network nodes are nodes higher than VTAM Version 4 Release 4, each providing RTP-level HPR support. The MPC connection between them, therefore, is an HPDT MPC connection.

When Non-HPDT MPC is used

If either partner node does not support HPDT MPC, non-HPDT MPC is automatically used.

Keep in mind that VTAMs prior to Version 4 Release 4 do not support HPDT MPC, and therefore, non-HPDT MPC will be used.

In Figure 6 on page 54, assume that VTAM network node 1 is a VTAM Version 4 Release 4 or higher node that does not provide RTP-level HPR support and that VTAM network node 2 is a VTAM node prior to Version 4 Release 4. Because VTAM releases prior to Version 4 Release 4 do not support HPDT MPC, the MPC connection between them is a non-HPDT MPC connection.

Of course, if both nodes are VTAM Version 4 Release 4 or lower network nodes, the MPC connection between them would also be a non-HPDT MPC connection.

Controlling the level of MPC used

Use the MPCLEVEL operand on the TRLE definition statement to control the level of MPC used for the connection. When VTAM and the adjacent node support HPDT MPC, the connection automatically uses HPDT MPC, unless VTAM includes MPCLEVEL=NOHPDT on its TRLE definition statement. When the adjacent node is another VTAM node that supports HPDT MPC, if MPCLEVEL=NOHPDT is coded on the TRLE definition statement of at least one of the VTAM nodes, the connection uses non-HPDT MPC.

Note: There are situations in which you must code MPCLEVEL=NOHPDT. Otherwise, the connection or session requests over the connection will fail. These situations include:

- Both nodes are VTAM Version 4 Release 4 or higher nodes and one of the VTAM nodes does not provide RTP-level HPR support. In this case, MPCLEVEL=NOHPDT must be specified on the TRLE definition statement in the VTAM node that does not provide RTP-level HPR support or the connection attempt will fail. See the following example for a full explanation of this situation.
- Both nodes are VTAM Version 4 Release 4 nodes that provide RTP-level HPR support, but the HPR operand on the PU definition statement in the local SNA major node in a least one of the VTAM nodes is coded as follows:
  - HPR=NO
  - HPR=YES, when the HPR start option specifies HPR=(RTP,NONE) or HPR=(RTP, ANR)

  In these cases, MPCLEVEL=NOHPDT must be specified on the TRLE definition statement in at least one of the VTAM nodes or the connection attempt will fail.

In Figure 6 on page 54, assume that both VTAM network nodes are VTAM nodes, but that VTAM network node 1 provides RTP-level HPR support and VTAM network node 2 does not. Because the default value for the MPCLEVEL operand is HPDT, during negotiation, both VTAM nodes agree to use an HPDT MPC connection. But, because VTAM network node 2 does not provide RTP-level HPR support, the connection attempt fails. For the connection to be successfully established, it must be a non-HPDT MPC connection. You must code MPCLEVEL=NOHPDT on the TRLE definition statement that defines the connection in VTAM network node 2 to make this a non-HPDT MPC connection, as follows:

Definitions for network node 1 HPR=RTP	Definitions for network node 2 HPR=NONE
TRL Major Node TRLNN1 VBUILD TYPE=TRL MPCLN1 TRLE LNCTL=MPC, MAXBFRU=16, READ=(BCA), WRITE=(BCB) **Local SNA Major Node** LSNA1 VBUILD TYPE=LOCAL LSNAPU1 PU TRLE=MPCLN1, ISTATUS=ACTIVE, CONNTYPE=APPN, CPCP=YES	**TRL Major Node** TRLNN2 VBUILD TYPE=TRL MPCLN2 TRLE LNCTL=MPC, MAXBFRU=16, READ=(BCB), WRITE=(BCA), MPCLEVEL=NOHPDT **Local SNA Major Node** LSNA2 VBUILD TYPE=LOCAL LSNAPU2 PU TRLE=MPCLN2, ISTATUS=ACTIVE, CONNTYPE=APPN, CPCP=YES

MPC dynamics

With MPC dynamics support, if at least one WRITE and one READ subchannel path is allocated successfully, an MPC connection is activated. Additional paths (defined but not ONLINE) in an MPC group can later be dynamically added to the active group using the MVS command `VARY device ONLINE`. For example, if there is a need for an increase in capacity to allow for extra traffic over a channel, additional paths can be added to the active group without disruption. Similarly, paths can be deleted from the active group when no longer needed using the MVS command `VARY device OFFLINE`.

Note: By default, the last READ and last WRITE subchannel paths in an MPC group cannot be deleted. To be able to use `VARY device OFFLINE` to delete the last READ or last WRITE subchannel path in an MPC group, include LASTRW=ALLOW on the TRLE definition statement for the MPC connection. For details on the TRLE definition statement, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

If a path fails, it can be dynamically added back to the active group without disruption [automatically using the Event Notification Facility (ENF) channel recovery support or by issuing the MVS command `VARY device ONLINE`].

When an MPC connection is activated, if the minimum number of WRITE and READ subchannel paths cannot be allocated successfully, either because they are offline or no valid path exists to the connecting host, then activation is suspended until the minimum number becomes available. This is true unless the start option specification MPCACT=NOWAIT is in effect.

HPDT Packing

Restriction: HPDT Packing can only be enabled for non-XCF point-to-point connections. Therefore XCF, ATM, and OSA devices are excluded.

The HPDT device driver was initially designed for large data transfer and within the architectural limitations of the I/O subsystem. HPDT is efficient at transporting large pieces of data but relatively inefficient when building a data stream with multiple small to medium size packets (packets from 257 to 2KB bytes, including headers). A write data stream built from this type of workload results in a

significant number of alignment bytes. These alignment bytes are transmitted across the media but are not used by the receiver. In effect, the alignment bytes significantly reduce the media bandwidth.

HPDT Packing can be enabled to eliminate most of these alignment bytes and effectively increase the media bandwidth. However, HPDT Packing is a compromise between CPU cycles, storage, and channel bandwidth. There are two side effects of enabling HPDT Packing:

- CPU use increases because of the cost of moving the data into the transmit packing buffer.
- Storage use increases because fixed packing transmit buffers are allocated.

Recommendation: In cases where the HPDT MPC connection is to a router or through a channel extender (and packet size is small to medium), you should consider enabling HPDT Packing. In these cases, the benefits of a densely packed data stream might exceed the costs in CPU and storage resources. If channel bandwidth is not a concern, enabling HPDT Packing is not recommended.

Determining the storage cost of enabling HPDT Packing: HPDT Packing is enabled on a TRLE basis. When enabled, seven packing buffers are acquired for each write device defined in the TRLE. Packing buffer size does not vary within the TRLE. Packing buffers are obtained from the smallest CSM data space pool that can contain the entire buffer.

Determining packing buffer size

The packing buffer size is equal to the number of 4 KB pages (minus 1 page) specified by the MAXBFRU parameter on the TRLE definition in the adjacent link station (or equivalent when the adjacent link station is a router). This value can be determined without accessing the adjacent link station when the TRLE is active. Issue the DISPLAY NET,ID=trlename command for the TRLE for which you are considering enabling HPDT Packing. The MAXBFRU value of the adjacent link station is contained in the response. The response looks similar to the following:

```
IST097I DISPLAY ACCEPTED
 IST075I NAME = TRLE1A, TYPE = TRLE
 IST486I STATUS= ACTIV----E, DESIRED STATE= ACTIV
 IST087I TYPE = LEASED             , CONTROL = MPC , HPDT = YES
 IST1715I MPCLEVEL = HPDT       MPCUSAGE = SHARE
 IST1717I ULPID = AHHCPU1
 IST1801I UNITS OF WORK FOR NCB AT ADDRESS X'02EAE018'
 IST1802I CURRENT = 0 AVERAGE = 1 MAXIMUM = 2
>IST1577I HEADER SIZE = 4092 DATA SIZE = 16 STORAGE = ***NA***
 IST1221I WRITE DEV = 0508 STATUS = ACTIVE     STATE = ONLINE
 IST1577I HEADER SIZE = 4092 DATA SIZE = 16 STORAGE = DATASPACE
 IST1221I READ  DEV = 0408 STATUS = ACTIVE     STATE = ONLINE
 IST314I END
```

The DATA SIZE value reported in the first IST1577I message reflects the MAXBFRU parameter in the adjacent link station. In this case, the adjacent link station defined a data stream maximum size of 16 pages. Therefore, if HPDT Packing is enabled for this TRLE, the packing buffer size will be 15 pages, or 60 KB. Because 7 packing buffers are allocated for the single write device, the total storage cost in CSM fixed data space is 7*60 KB or 420 KB.

Packing buffer size versus CSM data space pool size

The packing buffers are obtained from the smallest CSM data space pool

that can contain the entire buffer. If the packing buffer size does not equal one of the CSM data space pool sizes, storage is wasted.

In the example above, the packing buffer size is exactly equal to the second largest CSM data space pool size; therefore no storage is wasted. The worst case occurs when DATA SIZE is equal to 10, meaning 9 page (36 KB) packing buffers are used. Since the packing buffers cannot be acquired from the 32 KB pool, they are acquired from the 60 KB pool. The last 24 KB of each packing buffer is wasted. 7*24 KB equals 160 KB of wasted storage for a single write device.

The previous example explains why you should adjust the MAXBFRU value in the adjacent node (if necessary) prior to enabling HPDT Packing, so that the packing buffer size is equal to one of the CSM data space pools sizes.

Table 4 shows the correlation between DATA SIZE (MAXBFRU in the adjacent link station), the packing buffer size, the CSM data space pool size from which the packing buffers are allocated, and any apparent storage waste due to mismatch of packing buffer size and CSM data space pool size.

Table 4. HPDT Packing - packing buffer size, CSM pools size, and waste per packing buffer

DATA SIZE	Packing buffer size	CSM pool used	Waste per packing buffer
2	4 KB	4 KB	None
3	8 KB	16 KB	8 KB
4	12 KB	16 KB	4 KB
5	16 KB	16 KB	None
6	20 KB	32 KB	12 KB
7	24 KB	32 KB	8 KB
8	28 KB	32 KB	4 KB
9	32 KB	32 KB	None
10	36 KB	60 KB	24 KB
11	40 KB	60 KB	20 KB
12	44 KB	60 KB	16 KB
13	48 KB	60 KB	12 KB
14	52 KB	60 KB	8 KB
15	56 KB	60 KB	4 KB
16	60 KB	60 KB	None

Composite network node channel connections

Figure 7 on page 59 shows how to connect an NCP in a composite network node to a network node.

Note: This example does not include all definitions needed for an NCP major node. For further information on defining the NCP to VTAM, see "Connecting an NCP to VTAM" on page 115.

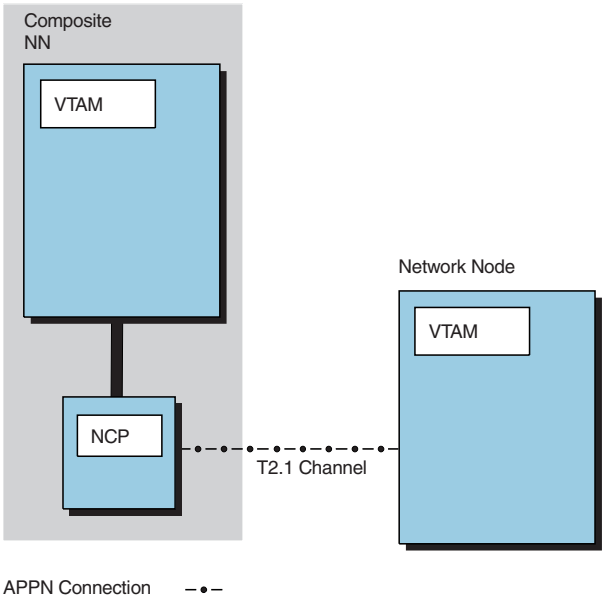


Figure 7. Type 2.1 channel connection between a composite network node and a network node

Composite network node definitions

Define the NCP major node as follows:

```
⋮
GRP3AAA1 GROUP LNCTL=CA,CA=TYPE5,NCPCA=ACTIVE
LN3AAA1  LINE  ADDRESS=04,TIMEOUT=840.0,CASDL=420.0,TRANSFR=254,       X
               INBFRS=128,ANS=CONT
P3A21AA1 PU    PUTYPE=2,XID=YES,                                       X
               CONNTYPE=APPN,CPCP=YES
⋮
```

Network node definitions

Define the local SNA major node as follows:

```
LSNA3AA  VBUILD TYPE=LOCAL
LSNA3APA PU    PUTYPE=2,CUADDR=050,XID=YES,                            *
               MAXBFRU=15,CONNTYPE=APPN,                               *
               CPCP=YES
```

Leased connections between APPN nodes

A leased connection can be used to connect a composite network node to a composite network node or to other nodes that support leased connections. Figure 8 on page 60 shows a leased connection.

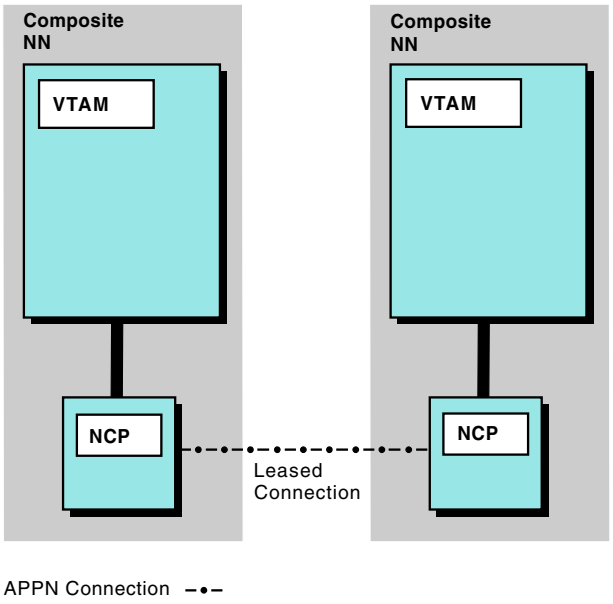


Figure 8. Leased connection between two composite network nodes

The following examples show how to connect two composite network nodes.

Connect the NCP major node for the first host as follows:

```
GRP4A6   GROUP LNCTL=SDLC,DIAL=NO,MODE=SEC,TYPE=NCP,                  X
               CLOCKNG=EXT,DUPLEX=HALF,SPEED=1200
*
LN4A6    LINE  ADDRESS=04E, TADDR=C1
*
         SERVICE ORDER=(P4A4956C)
*
P4A4956C PU    PUTYPE=2,XID=YES,                                      X
               MAXDATA=256,MAXOUT=1,PASSLIM=1,                        X
               ANS=CONTINUE,CONNTYPE=APPN,CPCP=YES
```

Connect the NCP major node for the second host as follows:

```
GRP3A9   GROUP LNCTL=SDLC,DIAL=NO,TYPE=NCP,                          X
               CLOCKNG=EXT,DUPLEX=HALF,SPEED=1200
*
LN3A11   LINE  ADDRESS=03F
*
         SERVICE ORDER=(P3A4956M)
*
P3A4956M PU    PUTYPE=2,ADDR=C1,XID=YES,                              X
               MAXDATA=256,MAXOUT=1,PASSLIM=1,                        X
               ANS=CONTINUE,CONNTYPE=APPN,CPCP=YES
```

IBM 3172 Nways Interconnect Controller connections between APPN nodes

You can use an IBM 3172 Nways Interconnect Controller connection to connect a composite network node, end node, or network node with any other type of APPN node.

Note: VTAM uses single-route broadcasts (rather than all-routes broadcasts) when attempting to connect to nodes on token-ring networks attached through the IBM 3172 Nways Interconnect Controller. When a node can be reached only

through bridges, there must be a route from the IBM 3172 Nways Interconnect Controller to the node, and that route must traverse bridges that are configured to route single-route broadcasts. Refer to the *Token-Ring Network Architecture Reference* for more information on all-routes and single-route broadcasts.

Figure 9 shows how to connect two network nodes using IBM Nways Interconnect Controller.

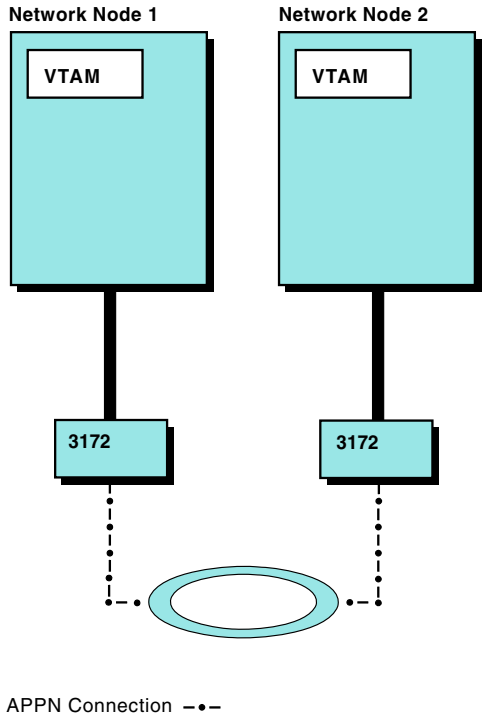


Figure 9. Two network nodes connected using an IBM 3172 Nways Interconnect Controller

Example

As shown in the following example, by coding DYNPU in the XCA major node, network node 1 does not need a switched major node; however, network node 2 must initiate the session using the DIALNO coded in its switched major node.

XCA major node in network node 1

```
XCA1A1L  VBUILD  TYPE=XCA
PORT1A1L PORT    MEDIUM=RING,ADAPNO=1,SAPADDR=4,CUADDR=500,TIMER=254
GP1A1L1  GROUP   DIAL=YES,ANSWER=ON,CALL=INOUT,                        X
                 DYNPU=YES,DYNPUPFX=BA
L1A1L1AA LINE
P1A1L1AA PU
```

XCA major node in network node 2

```
XCAAA1L  VBUILD  TYPE=XCA
PORTAA1L PORT    MEDIUM=RING,ADAPNO=1,SAPADDR=4,CUADDR=503,TIMER=254
GPAA1L1  GROUP   DIAL=YES,ANSWER=ON,CALL=INOUT
LAA1L11A LINE
PAA1L11A PU
```

Switched major node in network node 2

```
SWXCAAA  VBUILD TYPE=SWNET
SWP1A1   PU     MAXDATA=256,ADDR=11,CPNAME=SSCP1A,                    X
                PUTYPE=2
PTH1A1L  PATH   DIALNO=1A04003A11111111,                              X
                GRPNM=GPAA1L1
```

Note: Asynchronous transfer mode (ATM) networks accessed through LAN emulation appear to VTAM to be Ethernet or Ethernet-type LANs or token-ring networks and are defined to VTAM as such. ATM networks accessed through native ATM are defined to VTAM differently than those accessed through LAN emulation. See "ATM native connections" on page 69 for information about defining ATM native connections.

Using a connection network

A connection network is a representation of a shared access transport facility (SATF), such as a local area network (LAN). This arrangement enables nodes identifying their connectivity to the SATF by a common virtual routing node to communicate without having individually defined connections to one another. This is illustrated in the following figures.

In Figure 10 on page 63, when an LU-LU session between resources at AS400A and HOSTB is set up, the optimal route is directly through the token ring between the two nodes. However, if AS400A and HOSTB are not directly defined to each other or are not connected to the same connection network, the indirect route is selected through HOSTA.

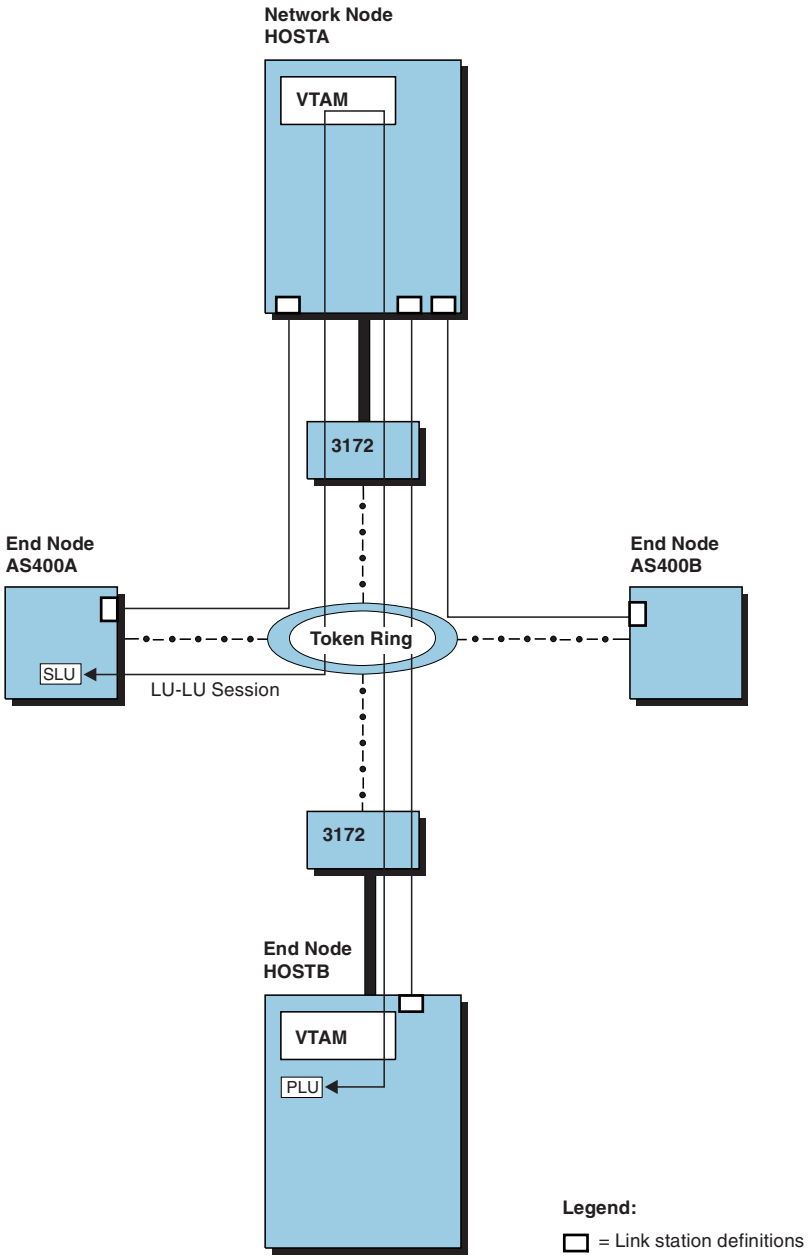


Figure 10. VTAM attachment to a LAN—No meshed connection definitions

Optimal route calculation is achieved in one of two ways:
- Meshed connection definitions
- Connection network definition

A definition for the connection between each pair of nodes, (n*(n-1)) definitions where n equals the number of nodes, enables optimal route calculation between all nodes on the SATF. In Figure 11 on page 64, for example, the pictured LU-LU session no longer traverses HOSTA.

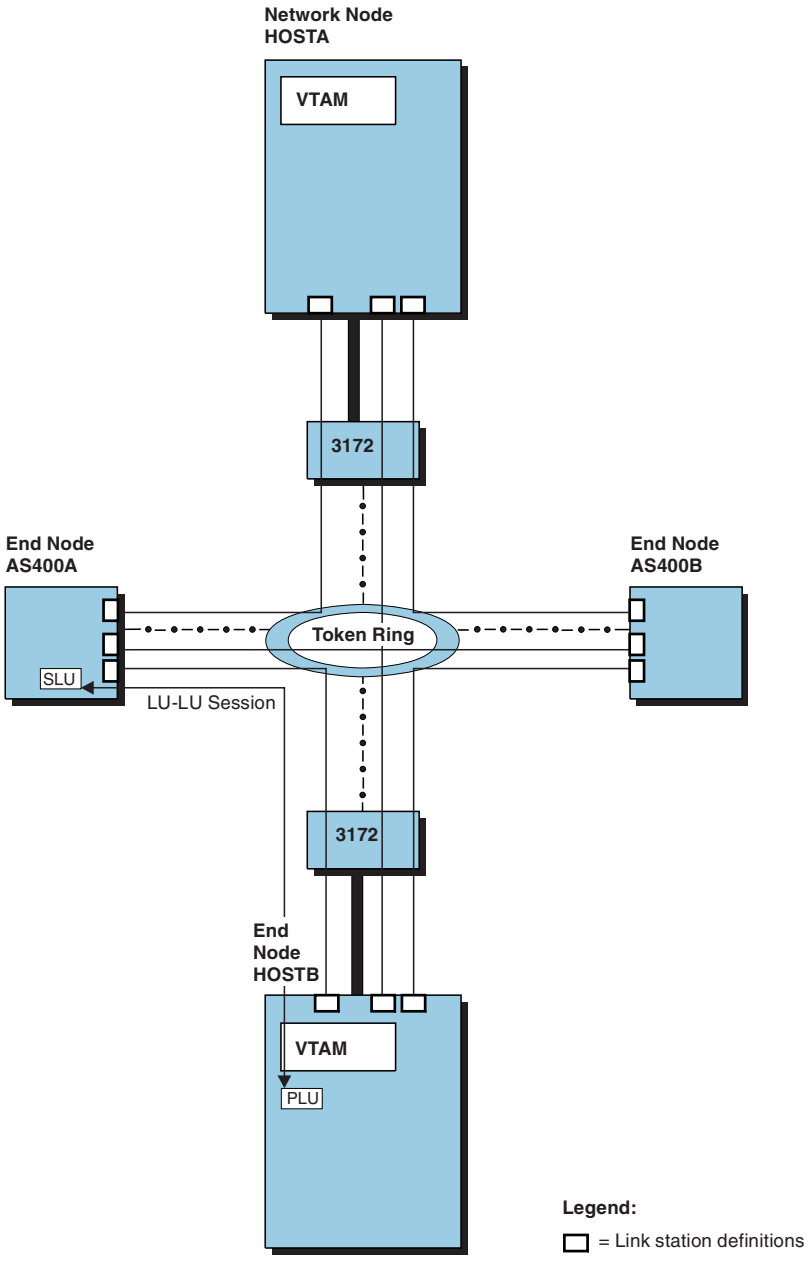


Figure 11. VTAM attachment to a LAN—Meshed connection definitions provide optimal route calculation

In a large network, however, this system definition is extensive. System definition is greatly reduced by defining a connection network to represent the shared access transport facility (here, the token ring). In a connection network, end nodes need only define a connection to a virtual node, as shown in Figure 12 on page 65.

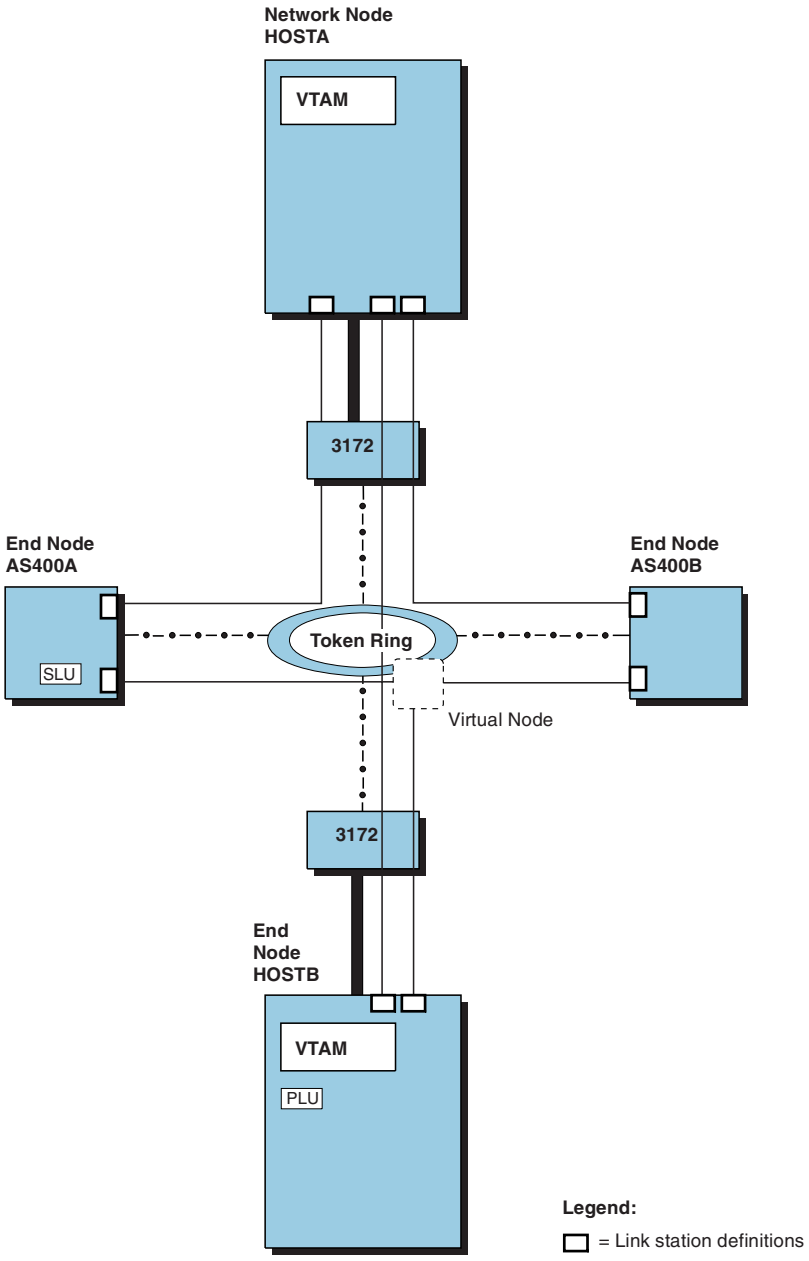


Figure 12. VTAM attachment to a connection network reduces required connection definitions (token ring)

The virtual node is reported to the topology database and can be chosen as the intermediate node during route calculation. As shown in Figure 13 on page 66, BINDs can then be routed directly to the destination node over a dynamically created TG. Thus, the definition of a connection network to represent the token ring reduces required system definition and enables optimal route calculation.

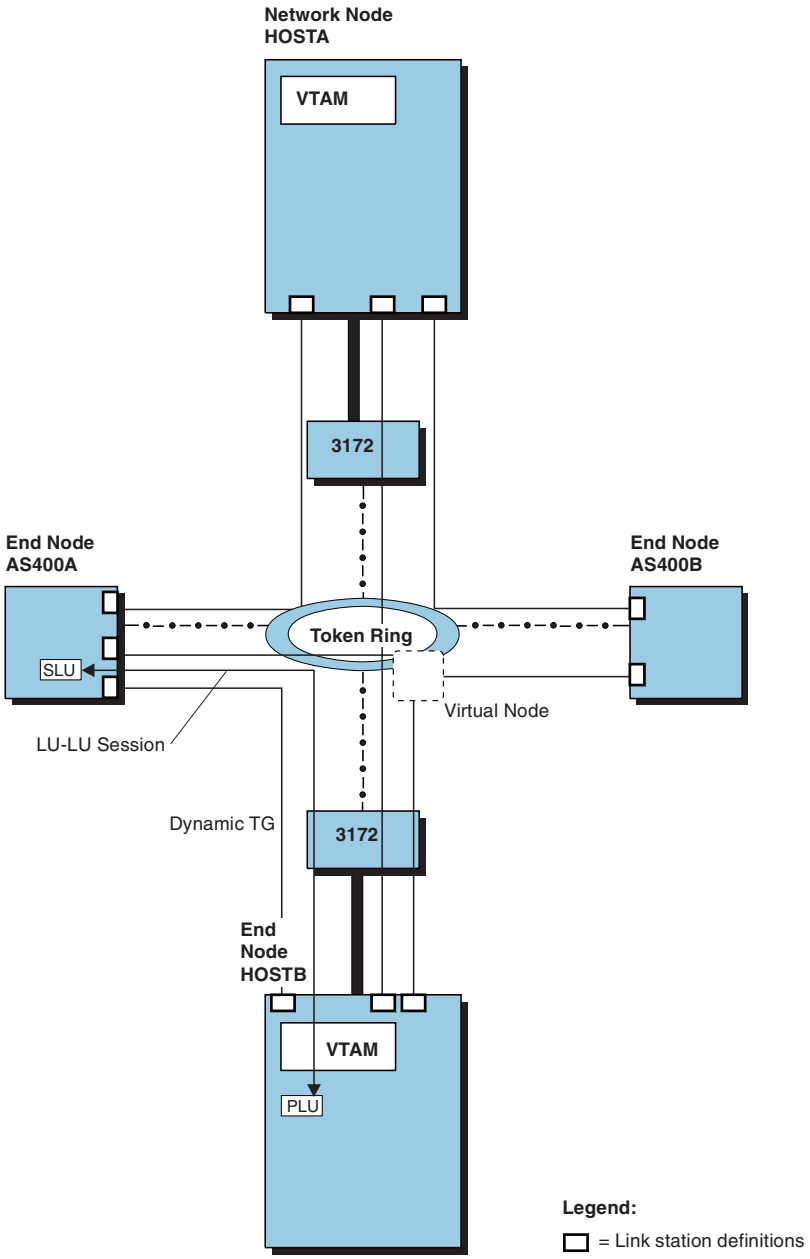


Figure 13. VTAM attachment to a connection network also enables optimal route calculation (token ring)

To define a connection to a connection network, include the following in the NCP major node, the XCA major node, or the LAN major node:

- In the NCP major node [NTRI (NCP token ring interconnect) physical lines]:
  - Code the VNNAME and VNGROUP operands on the LINE definition statement. The VNNAME operand specifies the CP name of the virtual node. The VNGROUP operand specifies the name of the logical group containing dial-out links through the connection network named on the VNNAME operand.

Note: DYNPU=YES is the default when VNNAME and VNGROUP are coded. For more information on DYNPU, see Chapter 9, "Defining resources dynamically," on page 227.

- Optional operands on the LINE definition statement include the TGP operand to indicate a TG profile containing TG characteristics, or operands for individual TG characteristics.

- In the XCA major node (IBM 3172 Nways Interconnect Controller lines):
  - Code the VNNAME and VNGROUP operands on the PORT definition statement. The VNGROUP operand specifies the GROUP in this major node containing the lines for the virtual node named on the VNNAME operand.

    Note: DYNPU=YES is the default when VNNAME and VNGROUP are coded. DIAL=YES is required on the GROUP named on the VNGROUP operand.
  - Optional operands on the PORT definition statement include the TGP operand to indicate a TG profile containing TG characteristics, or operands for individual TG characteristics.
- In the LAN major node:
  - Code the VNNAME and VNGROUP operands on the PORT definition statement. The VNGROUP operand specifies the GROUP in this major node containing the lines for the virtual node named on the VNNAME operand.

    Note: DYNPU=YES is the default when VNNAME and VNGROUP are coded. DIAL=YES is required on the GROUP named on the VNGROUP operand.
  - Optional operands on the PORT definition statement include the TGP operand to indicate a TG profile containing TG characteristics, or operands for individual TG characteristics.

Notes:

1. NCP Version 7 Release 1 is required for NCP/Token-Ring interconnection (NTRI) support of connection network.
2. If CP-CP sessions are required between two nodes on the shared access transport facility, the dialing-out node must define the PU for any node it is to call; routes traversing a virtual node cannot be used for CP-CP sessions.
3. The dial-out node can use only a dynamically defined PU for connectivity. The dial-in node attempts to use a predefined PU, if one exists. Otherwise, it uses a dynamically defined PU.
4. An ADJCP definition statement with the VN=YES operand can optionally be placed into the adjacent control point major node. The VN=YES operand indicates that this adjacent CP is a virtual node. The VN operand cannot be specified when the NN operand or the DYNLU=YES operand is also specified.
5. Both VTAM end nodes and network nodes can define a connection to a virtual node.
6. A connection network can be a node in an HPR route.
7. Available lines must exist for the call through the connection network both within the dial-out node and the dial-in node. For the dial-out node, available lines must exist in the group specified by the VNGROUP keyword. The dial will fail if no connectable line is found. With multiple virtual node definitions between two real nodes, if the VNGROUP keywords specify different groups, because either route through the virtual node may be selected (depending upon the topological weight), it causes the connection network function to attempt to locate a connectable line in a group without one. For the dial-in node, a

connectable line must be available for the call through the connection network. If all lines are busy with other calls, the call through the connection network will fail.

8. By default, dynamic connection network PUs take a name in the format *CNVxxxxx*, where *xxxxx* is a unique value that is generated and concatenated to the default *CNV* prefix to create the eight-character PU name. To specify a different prefix (two characters), use the DYNVNPFX start option.
9. A model PU definition can be created to customize the characteristics of dynamically created PUs. Use the DYNTYPE=VN operand for the model PU definition in the model major node.

IBM Open Systems Adapter connections between APPN nodes

VTAM can connect to the following shared access transport facilities (SATFs) through the IBM Open Systems Adapter:

- Local area network (LAN) connections
- Asynchronous transfer mode (ATM) networks accessed through LAN emulation or native ATM

LAN connections

VTAM supports the following types of LANs through the IBM Open Systems Adapter (configured in SNA mode):

- Ethernet or Ethernet-type LAN
- Token-ring network
- Fiber distributed data interface (FDDI)

LANs attached to VTAM through an IBM Open Systems Adapter are defined to VTAM in the same way as LANs attached through an IBM 3172 Nways Interconnect Controller. To define LAN connections through the IBM Open Systems Adapter, apply the information about defining LAN connections through the IBM 3172 Nways Interconnect Controller, found in "External communication adapter (XCA) connections" on page 258 and "IBM 3172 Nways Interconnect Controller connections between APPN nodes" on page 60.

ATM connections

ATM is a switching technology that provides fast, reliable, simultaneous transfer of data, voice, and video. VTAM supports ATM technology by enabling communication across ATM networks—both public (wide area networks, or WANs) and private (campus networks)—accessed through:

- LAN emulation
- Native ATM

ATM LAN emulation connections: ATM networks accessed through LAN emulation are attached to VTAM through an IBM Open Systems Adapter (configured in SNA mode). They appear to VTAM as though they are Ethernet or Ethernet-type LANs or token-ring networks. ATM LAN emulation connections offer the benefits of fast data transfer, but they do not offer the full services of ATM networks, such as guaranteed bandwidth and Quality of Service.

ATM LAN emulation connections through an IBM Open Systems Adapter are defined to VTAM in the same way as LAN connections through an IBM 3172 Nways Interconnect Controller. To define ATM LAN emulation connections through the IBM Open Systems Adapter, apply the information about defining LAN connections through the IBM 3172 Nways Interconnect Controller, found in "External communication adapter (XCA) connections" on page 258 and "IBM 3172 Nways Interconnect Controller connections between APPN nodes" on page 60.

ATM native connections: ATM networks accessed through native ATM are also attached to VTAM through an IBM Open Systems Adapter (configured in HPDT ATM native mode). They offer the full services of ATM networks, such as guaranteed bandwidth and Quality of Service.

VTAM enables multiprotocol attachments to ATM networks (both public and private) accessed through native ATM. These multiprotocol attachments are achieved through a high-performance data transfer (HPDT) multipath channel (MPC) connection to an IBM Open Systems Adapter. A single HPDT MPC connection can be simultaneously shared by multiple products using different communication protocols, provided those other products also support HPDT MPC connections. For example, APPN and non-APPN data can be simultaneously sent and received across an ATM network over the same HPDT MPC connection. For more information about HPDT MPC connections, see "Multipath channel connections" on page 51.

Note: VTAM can route only HPR APPN data across ATM native connections. Routing of subarea and non-HPR APPN data across ATM networks requires ATM LAN emulation connections.

Defining ATM native connections to VTAM

Figure 14 shows a basic ATM configuration enabling HPR APPN communication through native access.

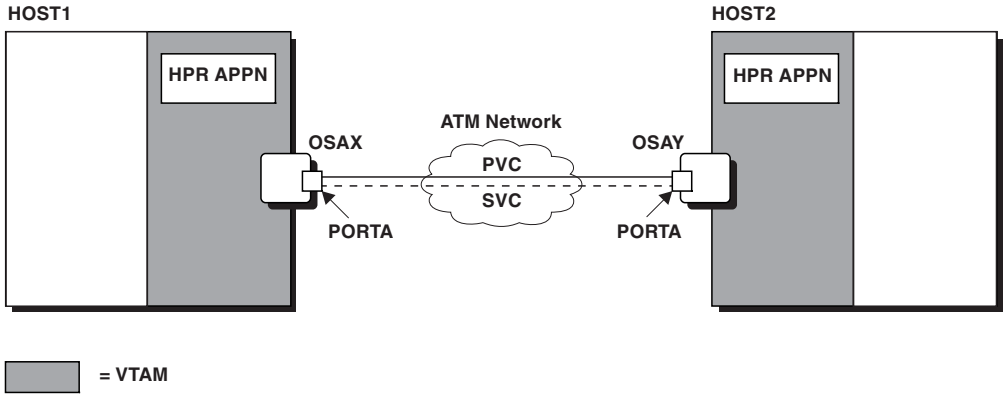


Figure 14. Basic ATM configuration

The major elements of an ATM configuration that must be defined to VTAM are:

- VTAM connection to the IBM Open Systems Adapter
- The port on the IBM Open Systems Adapter through which the ATM network is accessed
- The transmission groups (TGs) that route data:
  - Across permanent virtual channels (PVCs)
  - Across switched virtual channels (SVCs)
  - To connection networks

Defining VTAM connection to the IBM Open Systems Adapter

Figure 15 on page 70 illustrates a VTAM connection to the IBM Open Systems Adapter.

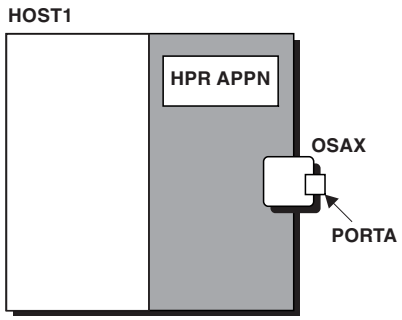


Figure 15. VTAM connection to the IBM Open Systems Adapter

VTAM communicates with the IBM Open Systems Adapter using an HPDT MPC connection. For more information about HPDT MPC connections, see "Multipath channel connections" on page 51.

Define the following key characteristics of the MPC connection in the transport resource list (TRL) major node:

- **Characteristics you must specify**
  - Name of the IBM Open Systems Adapter
  - Subchannel addresses of the WRITE and READ paths
  - Name of the port associated with the IBM Open Systems Adapter to which the WRITE and READ paths are connected
- **Characteristics specified by default**
  - Line control (LNCTL=MPC)

    LNCTL=MPC is required for ATM native connections and is automatically defined for you when you code the PORTNAME operand on the TRLE definition statement in the TRL major node. The only possible value for ATM native connections is LNCTL=MPC.
  - MPC level (MPCLEVEL=HPDT)

    MPCLEVEL=HPDT is required for ATM native connections and is automatically defined for you when you code the PORTNAME operand on the TRLE definition statement in the TRL major node. The only possible value for ATM native connections is MPCLEVEL=HPDT.

In addition to required user-specified characteristics, optional characteristics can be defined. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for information about how to code the definition statements and operands used to define both the key and optional characteristics.

The example shown in Figure 16 on page 71 is based on the configuration in Figure 15 and represents definitions in the VTAMLST data set for the VTAM in HOST1. Following Figure 16 on page 71 are descriptions of the major nodes, definition statements, and operands used in Figure 16 on page 71 to define VTAM connection to the IBM Open Systems Adapter.

```
TRL1    VBUILD    TYPE=TRL

OSAX    TRLE      WRITE=(501),
                  READ=(500),
                  PORTNAME=PORTA,
                  LNCTL=MPC,
                  MPCLEVEL=HPDT,
                  .
                  .
                  .
```

Figure 16. Definition of VTAM connection to the IBM Open Systems Adapter

Definition statements	Description
Name of the IBM Open System Adapter	The name field of the TRLE definition statement indicates that the name of the IBM Open Systems Adapter is OSAX. The name specified here is the name of the IBM Open Systems Adapter. The name of the IBM Open Systems Adapter is defined during IBM Open Systems Adapter configuration on the ATM Native Settings panel, which is part of the OSA Configuration pull-down option accessed through the OSA/SF OS/2 interface.
Subchannel Addresses of the WRITE and READ Paths	The WRITE operand on the TRLE definition statement specifies that the subchannel address of the WRITE path is 501. The READ operand specifies that the subchannel address of the READ path is 500. The last two digits of the subchannel address of the READ path must match the Even Unit Address. The Even Unit Address and the ATM OSA-2 physical port are defined during IBM Open Systems Adapter configuration on the ATM Native Settings panel, which is part of the OSA Configuration pull-down option accessed through the OSA/SF OS/2 interface. **Note:** Only one subchannel address is specified for the READ path and one for the WRITE path. The address specified for the READ path must be an even number that is one less than the address specified for the corresponding WRITE path.
Name of the Port Associated with the IBM Open Systems Adapter to which the WRITE and READ Paths are Connected	The PORTNAME operand on the TRLE definition statement specifies that the name of the port associated with this IBM Open Systems Adapter—through which an ATM network can be accessed—is PORTA. The name specified here is the name of the ATM OSA-2 physical port. The Even Unit Address and the ATM OSA-2 physical port are defined during IBM Open Systems Adapter configuration on the ATM Native Settings panel, which is part of the OSA Configuration pull-down option accessed through the OSA/SF OS/2 interface.

Defining the port on the IBM Open Systems Adapter through which the ATM network is accessed

Figure 17 on page 72 shows a port example.

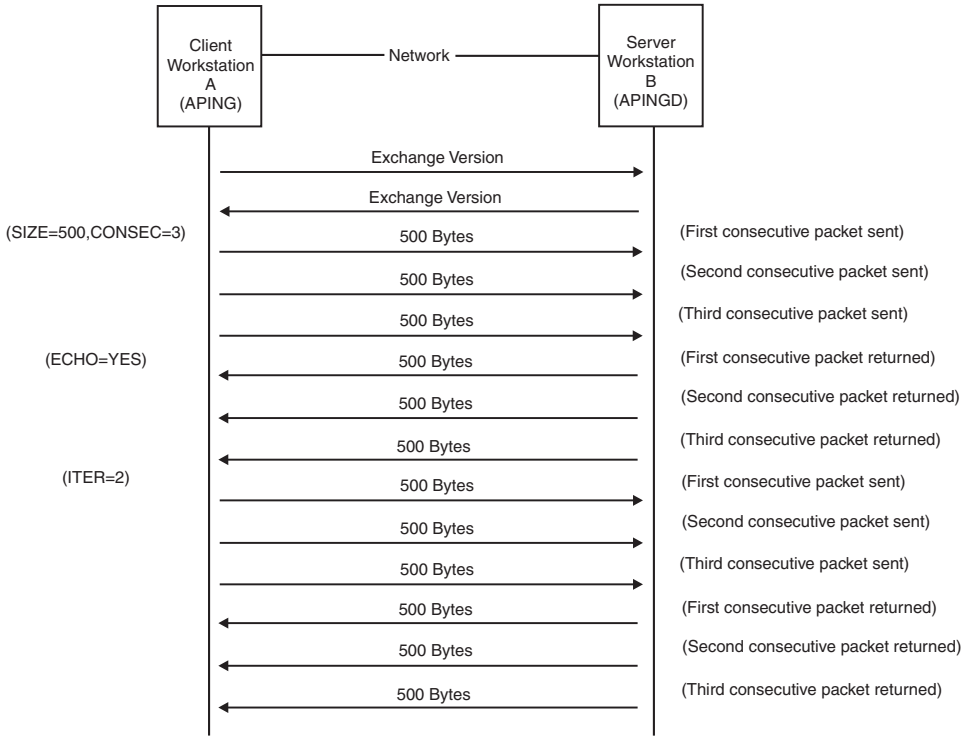


Figure 17. Port on the IBM Open Systems Adapter through which the ATM network is accessed

VTAM accesses the ATM network through a port on the IBM Open Systems Adapter. Associated with the port are links used for permanent virtual channels (PVCs) and switched virtual channels (SVCs). These PVCs and SVCs carry data across the ATM network. Define the following characteristics of the port in the external communication adapter (XCA) major node:

- **Characteristic you must specify** is the name of the port.
- **Characteristic specified by default** is the type of shared access transport facility (SATF) accessed through the port (MEDIUM=ATM).

  MEDIUM=ATM is required for ATM native connections and is automatically defined for you when you do not code the ADAPNO and CUADDR operands on the PORT definition statement in the XCA major node. The only possible value for ATM native connections is MEDIUM=ATM.

Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for information about how to code the definition statements and operands used to define these characteristics. No other characteristics of the port need to be defined.

Figure 18 on page 73 is based on the configuration in Figure 17 and represents definitions in the VTAMLST data set for the VTAM in HOST1. Following Figure 18 on page 73 are descriptions of the major nodes, definition statements, and operands used in Figure 18 on page 73 to define the port on the IBM Open Systems Adapter through which the ATM network is accessed.

```
OSAXCA1   VBUILD    TYPE=XCA

          PORT      PORTNAME=PORTA,
                    MEDIUM=ATM
```

Figure 18. Definition of port on the IBM Open Systems Adapter through which the ATM network is accessed

The PORTNAME operand on the PORT definition statement specifies that the name of the port on the IBM Open Systems Adapter with which the links used for PVCs and SVCs are associated is PORTA. The name specified here must also be defined on the PORTNAME operand on a TRLE definition statement in the TRL major node. Both specifications of the port name must match the name of the ATM OSA-2 physical port.[1]

Defining the APPN transmission groups that route data across the ATM network

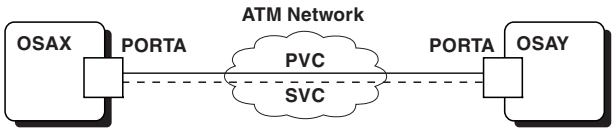


Figure 19. TGs that route data across the ATM network

VTAM uses transmission groups (TGs) to route data across the ATM network. These TGs can be over PVCs and SVCs, as shown in Figure 19. One TG is associated with each PVC or SVC.

In networks where multiple nodes can communicate with one another across an ATM network, you can define TGs to connection networks. This can minimize the amount of definition required to establish routes among the multiple nodes.

Defining transmission groups over permanent virtual channels:

Permanent virtual channels (PVCs) represent permanent connections. They are reserved by the ATM network and are available as long as the network is active. PVCs appear to VTAM as nonswitched lines. As such, the TGs that are assigned to them are defined in groups headed by a GROUP definition statement that specifies DIAL=NO.

Because PVCs are associated with a port on the IBM Open Systems Adapter, the TGs that are assigned to them are defined in the XCA major node that defines the port with which the PVCs are associated.

Define the following key characteristics of a TG over a PVC in the XCA major node:

- **Characteristic you must specify** is the name of the PVC.
- **Characteristics you should specify**
  - Name of the remote node with which VTAM can communicate over the TG
  - Route calculation characteristics
    - Cost per connect time
    - Cost per byte
    - Security

1. The name of the ATM OSA-2 physical port is defined during IBM Open Systems Adapter configuration on the ATM Native Settings panel, which is part of the OSA Configuration pull-down option accessed through the OSA/SF OS/2 interface.

- Propagation delay
- Effective capacity
- User-defined values

Note: IBM supplies default TG profiles that define these characteristics for best effort and reserved bandwidth connections across public and private ATM networks. IBM recommends that you use these default TG profiles. For examples of the default TG profiles, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

- **Characteristics specified by default**

  These characteristics are required for ATM native connections and are automatically defined for you when you do not code the ADAPNO and CUADDR operands on the PORT definition statement in the XCA major node, which is the equivalent of coding MEDIUM=ATM. The only possible values for ATM native connections are those shown in parentheses in the following list of characteristics.
  - PU type of the remote node (PUTYPE=2)
  - Type of connection (CONNTYPE=APPN)
  - High-performance routing (HPR) enablement (HPR=YES)
  - Channel contact procedure (XID=YES)

Figure 20 shows the definition of a TG over a PVC. Following Figure 20 are descriptions of the major nodes and the definition statements and operands used in Figure 20 to define a TG over a PVC.

```
OSAXCA1   VBUILD    TYPE=XCA

          PORT      PORTNAME=PORTA,
                    MEDIUM=ATM

          GROUP     DIAL=NO

 PVCLN1   LINE      PVCNAME=PVC1,
                    .
                    .
                    .

 PVCPU1   PU        CPNAME=HOST2,
                    TGP=ATMPVCCB,
                    CONNTYPE=APPN,
                    HPR=YES,
                    PUTYTPE=2,
                    XID=YES
                    .
                    .
                    .
```

Figure 20. Definition of a TG over a PVC

Name of the PVC:

The PVCNAME operand on the LINE definition statement specifies that the name of the PVC is PVC1. The name specified here is the name of the PVC associated with the ATM OSA-2 physical port. [2] The ATM network provider must configure the ATM network to provide the PVC connection.

2. The name of the PVC is defined during IBM Open Systems Adapter configuration on the ATM Native Settings panel, which is part of the OSA Configuration pull-down option accessed through the OSA/SF OS/2 interface.

Note: The IBM Open Systems Adapter limits the total number of PVCs that can be associated with one port. Refer to *zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference* for information about this limitation.

Name of the remote node with which VTAM can communicate over the TG:

The CPNAME operand on the PU definition statement specifies that the name of the remote node with which VTAM can communicate over this TG is HOST2.

Route calculation characteristics:

You can specify route calculation characteristics on the PU definition statement on the following operands:
- COSTTYPE
- COSTBYTE
- SECURITY
- PDELAY
- CAPACITY
- UPARM1
- UPARM2
- UPARM3

Or, they can be determined by the TGP operand on the PU definition statement, which specifies the name of an IBM-supplied APPN TG profile definition. In Figure 20 on page 74, they are determined by the TGP operand, which specifies the name of the IBM-supplied profile definition for ATM TGs over PVCs, campus best effort.

Defining transmission groups over switched virtual channels:

Figure 21 illustrates how TGs are defined over switched virtual channels.

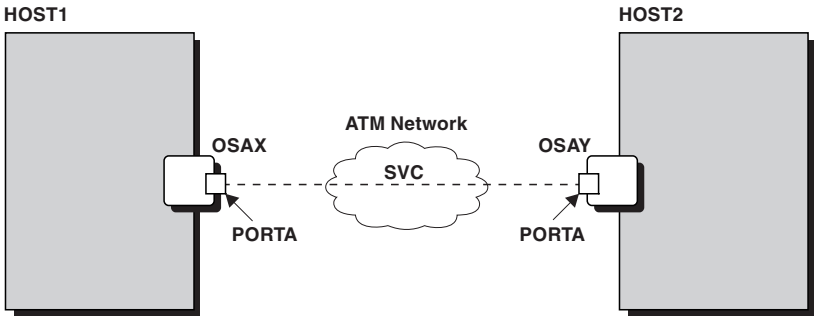


Figure 21. TG over an SVC

Switched virtual channels represent temporary connections. They are established through a dial operation and are available for as long as the connection is desired. SVCs appear to VTAM as switched lines. As such, the TGs that are assigned to them are defined in groups headed by a GROUP definition statement that specifies DIAL=YES.

Because SVCs are associated with a port on the IBM Open Systems Adapter, the TGs that are assigned to them are defined in the XCA major node that defines the port with which the SVCs are associated.

Define the following key characteristics of a TG over an SVC in the XCA major node:

- **Characteristic you must specify** is the maximum number of simultaneous connections using the port
- **Characteristics you should specify**
  - Whether the line used for the TG can be used for calls initiated by a remote node, VTAM or both
  - Whether PUs are to be dynamically created during call-in operations or explicitly defined in the switched major node

If a line used for a TG over an SVC is used for calls initiated by VTAM (CALL=OUT or CALL=INOUT), define the following key characteristics of the TG in the switched major node:

- **Characteristics you must specify**
  - The type of APPN routes that can use the SVC
  - The name of the group that contains the dial-out line definitions for the TG
  - Name of the remote node with which VTAM can communicate over the TG
  - Address through which the remote node can be reached
  - ATM channel characteristics
    - Best effort indicator
    - Cell rates
    - Traffic management options
- **Characteristics you should specify**
  - Route calculation characteristics
    - Cost per connect time
    - Cost per byte
    - Security
    - Propagation delay
    - Effective capacity
    - User-defined values

    Note: IBM supplies default TG profiles that define these characteristics for best effort and reserved bandwidth connections across public and private ATM networks. It is recommended that you use these default TG profiles. For examples of the default TG profiles, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.
  - ATM channel characteristics including the Quality of Service (QoS) class

In addition to required characteristics, optional characteristics can be defined. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for information about how to code the definition statements and operands used to define both the key and optional characteristics.

Figure 22 on page 77 is based on the configuration in Figure 21 on page 75 and represents definitions in the VTAMLST data set for the VTAM in HOST1. Following Figure 22 on page 77 are descriptions of the major nodes and the definition statements and operands used in Figure 22 on page 77 to define a TG over an SVC.

```
OSAXCA1  VBUILD    TYPE=XCA

         PORT      PORTNAME=PORTA,
                   MEDIUM=ATM

SVCGRP1  GROUP     DIAL=YES,
                   CALL=INOUT,
                   DYNPU=YES,
                   .
                   .
                   .

SVCLN1   LINE      .
                   .                                          .

SVCPU1   PU        .
                   .
                   .

SVCLN2   LINE      .
                   .
                   .

SVCPU2   PU        .
                   .
                   .

OSASWT1  VBUILD    TYPE=SWNET

SWTPU1   PU        CPNAME=HOST2,
                   TGP=ATMSVCCR,
                   .
                   .
                   .

         PATH      GRPNM=SVCGRP1,
                   DLCADDR=(1,C,ATMSVC,EXCLUSIVE),
                   DLCADDR=(7,BCD,03,00,
00006000,00004000,
                           00000191,00,
00006000,00004000,
                           00000191,00),
                   DLCADDR=(8,X,03,03,03),
                   DLCADDR=(21,X,0002,
                           399999999999999999),
999801013150455354346O),
                   .
                   .
                   .
```

Figure 22. Definition of a TG over an SVC

Maximum number of simultaneous connections:

The number of sets of LINE and PU definition statements in the XCA major node represents the maximum number of simultaneous connections that can be established using the port. In Figure 22, the maximum number of simultaneous connections is 2. The LINE definition statements serve as placeholders for lines used to connect to remote nodes that are defined either dynamically or in a switched major node, and are used when those nodes are activated.

Note: The IBM Open Systems Adapter limits the total number of SVCs that can be associated with one port. Refer to *zEnterprise 196, System*

z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference for information about this limitation.

Who can initiate calls:

The CALL operand on the GROUP definition statement in the XCA major node specifies that the lines in this group can be used for calls initiated by both the VTAM in HOST1 and remote nodes.

Dynamic creation of PUs:

The DYNPU operand on the GROUP definition statement in the XCA major node specifies that PUs are to be dynamically created when remote nodes call in to VTAM. By enabling PUs to be dynamically created, you eliminate the need to code PU and PATH definition statements in the switched major node.

The type of APPN routes that can use the SVC:

In Figure 22 on page 77, EXCLUSIVE on the DLCADDR operand with a subfield of 1 on the PATH definition statement in the switched major node indicates that the TG can be assigned to an SVC that is used only for this TG, and it cannot be assigned to an SVC that is used for connection network TGs.

Meanings of the values coded on the DLCADDR operand are directly related to ATM channel characteristics. For additional information about what the values indicate, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Name of the group that contains the dial-out line definitions for the TG:

The GRPNM operand on the PATH definition statement in the switched major node specifies that the line to be used for the TG is defined in the group named SVCGRP1 in the XCA major node. This group follows a PORT definition statement, and identifies the lines to be used for the TG as those associated with the port defined on that PORT statement (PORTA in Figure 22 on page 77).

Name of the remote node with which VTAM can communicate over the TG:

The CPNAME operand on the PU definition statement specifies that the name of the remote node with which VTAM can communicate over this TG is HOST2.

Address through which the remote node can be reached:

The DLCADDR operand with subfield 21 on the PATH definition statement in the switched major node specifies that the 20-byte address through which HOST2 can be reached is identified as 39999999999999999999980101315045535434 60 and that this is an International Organization for Standardization (ISO) network service access point (NSAP) address, indicated by the 0002 preceding the address. The address through which HOST2 can be reached was obtained through configuration of OSAY in Figure 22 on page 77. OSAY is the IBM Open Systems Adapter through which HOST2 is connected to the ATM network. When the remote node is a VTAM node, configuration of the IBM Open Systems Adapter through which the remote node is connected should be completed before definition of the calling VTAM node is attempted. Refer to *zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express*

Customer's Guide and Reference for information on how to determine the ATM address through which the remote VTAM node can be reached.

Note: You can also determine the ATM address through which the remote VTAM node can be reached by starting the remote VTAM, then activating the following resources in the remote VTAM:

- The TRL major node
- The XCA major node that defines the port on the IBM Open Systems Adapter through which the remote VTAM can be reached
- Any line defined in the XCA major node

After activating the remote VTAM resources, issue a DISPLAY ID command for the XCA major node. The resulting display shows the ATM address for the remote VTAM node.

You can follow this same procedure to obtain the ATM address for the local VTAM as well.

Route calculation characteristics:

You can specify route calculation characteristics on the PU definition statement on the following operands:

- CAPACITY
- COSTBYTE
- COSTTYPE
- PDELAY
- SECURITY
- UPARM1
- UPARM2
- UPARM3

Or, they can be determined by the TGP operand on the PU definition statement, which specifies the name of an IBM-supplied APPN TG profile definition. In Figure 22 on page 77, they are determined by the TGP operand, which specifies the name of the IBM-supplied profile definition for ATM TGs over SVCs (campus reserved bandwidth).

Best effort indicator, cell rates, and traffic management options:

The DLCADDR operand with a subfield of 7 specifies the following values:

03	The format of the best effort indicator, cell rates, and traffic management options is defined by the ATM network.
00	Guaranteed bandwidth is desired.
00006000	The forward peak cell rate (cells per second), in binary coded decimal, for cell loss priority 0+1 (CLP=0+1) is 6000.
00004000	The forward sustainable cell rate (cells per second), in binary coded decimal, for (CLP=0) is 4000.
00000191	The forward maximum burst size (cells), in binary coded decimal, for (CLP=0) is 191.
00	Tagging is not requested in the forward direction.
00006000	The backward peak cell rate (cells per second), in binary coded decimal, for cell loss priority 0+1 (CLP=0+1) is 6000.
00004000	The backward sustainable cell rate (cells per second), in binary coded decimal, for (CLP=0) is 4000.

00000191 The backward maximum burst size (cells), in binary coded decimal, for (CLP=0) is 191.

00 Tagging is not requested in the backward direction.

Quality of Service (QoS) class:

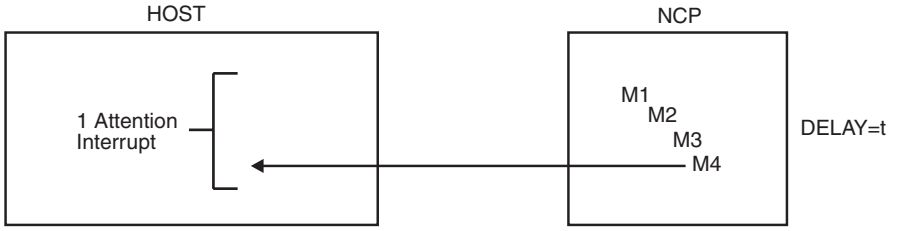
The DLCADDR operand with a subfield of 8 specifies the following:

Value	Meaning
03	The format of the QoS class is defined by the ATM network.
03	The QoS class is connection-oriented for outbound (forward) data traffic.
03	The QoS class is connection-oriented for inbound (backward) data traffic.

Defining transmission groups to connection networks

Figure 23 shows an ATM configuration in which multiple nodes can communicate with one another across an ATM network. HOST2 and HOST3 are end nodes that dial in to the network node server, HOST1. All nodes can establish switched connections with one another.

To enable optimal route selection among these nodes, an extensive number of TGs needs to be defined. Specifically, each node needs to define TGs to every other node. So, in Figure 23, HOST1 needs to define TGs to HOST2 and HOST3, HOST2 needs to define TGs to HOST1 and HOST3, and HOST3 needs to define TGs to HOST1 and HOST2. In a large configuration with many nodes connected by many SVCs, the definition of TGs can become overwhelming.



In this example, DELAY is equal to *t*. This time period is large enough to allow four messages to arrive at the NCP. The time *t* has expired, and the NCP presents an attention interrupt. VTAM reads all four messages, having received only one attention interrupt.

Figure 23. Multiple nodes communicating across an ATM network

APPN's connection network function reduces this extensive TG definition for connections among multiple nodes across an ATM network. A connection network is a representation of a shared access transport facility, such as an ATM network, that handles the routing of data among the nodes communicating across the shared access transport facility. It does this by enabling the shared access transport facility to be defined as a virtual node. As a result, end nodes need to define TGs only to the virtual node and to the network node server. The network node server, though it does not always need to, should also define a TG to the virtual node.

Figure 24 on page 81 shows the ATM configuration in Figure 23 with a virtual node. The virtual node represents the ATM network to the nodes

that can communicate with one another across the ATM network.

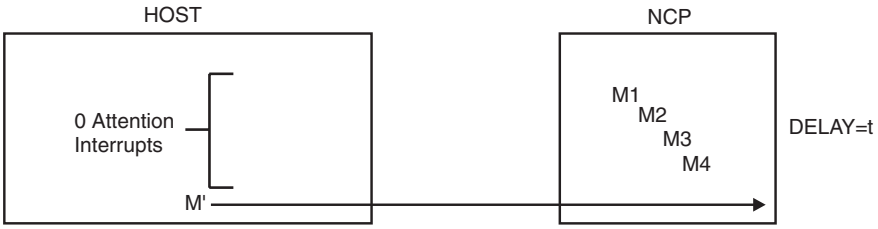


Figure 24. ATM configuration with a connection network

Now, HOST1 needs to define a TG only to VNODE1, as does HOST2 and HOST3. HOST2 and HOST3 must also each define a TG to HOST1 because HOST1 is their network node server. The same routing capability achieved by six TG definitions for the configuration in Figure 23 on page 80 is now achieved by five TG definitions using the virtual node in Figure 24. In large networks with many nodes connected by many SVCs, the reduction in TG definitions is much more significant. For example, without the use of a connection network, a configuration with five end nodes that dial into one network node server requires 30 TG definitions to enable optimal route selection. With the use of a connection network, that same configuration requires only 11 TG definitions to enable the same optimal route selection.

If another network node server and its associated end nodes wanted to communicate with HOST1, HOST2, and HOST3 through the same connection network, each new end node would need to define a TG to the virtual node and to the new network node server. The new network node server would also need to define a TG to the virtual node and to HOST1.

Multiple TGs through one port can be to the same virtual node. In addition, TGs to multiple virtual nodes can be through the same port.

Connections through connection networks occur over SVCs, which appear to VTAM as switched lines. Thus, TGs to connection networks are defined in groups headed by a GROUP definition statement that specifies DIAL=YES.

Because SVCs are associated with a port on the IBM Open Systems Adapter, TGs to connection networks are defined in the XCA major node that defines the port with which the SVCs are associated.

To define the following key characteristics of a TG to a connection network in the XCA major node:

- **Characteristics you must specify**
  - The type of APPN routes that can use the SVC
  - Name of the virtual node that represents the ATM network
  - The maximum number of simultaneous connections using the port
  - ATM channel characteristics
    - Best effort indicator

- Cell rates
- Traffic management options

- **Characteristics you should specify**
  - Whether the line used for the TG can be used for calls initiated by a remote node, VTAM, or both
  - Whether PUs are to be dynamically created during call-in operations or explicitly defined in the switched major node
  - Route calculation characteristics
    - Cost per connect time
    - Cost per byte
    - Security
    - Propagation delay
    - Effective capacity
    - User-defined values

    Note: IBM supplies default TG profiles that define these characteristics for best effort and reserved connections across public and private ATM networks and recommends use of these default TG profiles. For examples of the default TG profiles, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.
  - ATM channel characteristics
    - Quality of Service (QoS) class
- **Characteristics specified by default**
  - High-performance routing (HPR) enablement (HPR=YES)

    HPR=YES is required for ATM native connections and is automatically defined for you when you do not code the ADAPNO and CUADDR operands on the PORT definition statement in the XCA major node, which is the equivalent of coding MEDIUM=ATM. The only possible value for ATM native connections is HPR=YES.

In addition to the required characteristics, other optional characteristics can be defined. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for information about how to code the definition statements and operands used to define both the key and optional characteristics.

Figure 25 on page 83, Figure 26 on page 84, and Figure 27 on page 85 are based on the configuration in Figure 24 on page 81 and represent definitions in the VTAMLST data sets for the VTAMs in HOST1, HOST2, and HOST3. Following Figure 27 on page 85 are descriptions of the major nodes, definition statements, and operands used in Figure 25 on page 83, Figure 26 on page 84, and Figure 27 on page 85 to define TGs to a connection network.

```
HOST1

                     OSAXCA1  VBUILD    TYPE=XCA

                              PORT      PORTNAME=PORTA,
                                        MEDIUM=ATM

                      CNGRP1  GROUP     DIAL=YES,
                                        CALL=INOUT,
                                        DLCADDR=(1,C,ATMSVC, &cont
VNODE1,EXCLUSIVE),
                                        DLCADDR=(7,BCD,03,00, &cont
00006000,
                                                00004000,00000191, &cont
00),
                                        DLCADDR=(8,X,03,03),
                                        DYNPU=YES,
                                        TGP=ATMSVCCR,
                                        HPR=YES,
                                        .
                                        .
                                        .
                      CNLN1   LINE

                      CNPU1   PU

                      CNLN2   LINE

                      CNPU2   PU

                      CNLN3   LINE

                      CNPU3   PU
```

Figure 25. Definitions in VTAMLST for the VTAM in HOST1

```
HOST2

                     OSAXCA1  VBUILD    TYPE=XCA

                              PORT      PORTNAME=PORTA,
                                        MEDIUM=ATM

                      CNGRP1  GROUP     DIAL=YES,
                                        CALL=INOUT,
                                        DLCADDR=(1,C,ATMSVC,
VNODE1,EXCLUSIVE),
                                        DLCADDR=(7,BCD,03,00,
00006000,
                                                00004000,00000191,
00),
                                        DLCADDR=(8,X,03,03),
                                        DYNPU=YES,
                                        TGP=ATMSVCCR,
                                        HPR=YES,
                                        .
                                        .
                                        .
                      CNLN1   LINE

                      CNPU1   PU

                      CNLN2   LINE

                      CNPU2   PU

                      CNLN3   LINE

                      CNPU3   PU

                     OSASWT1  VBUILD    TYPE=SWNET

                     SWTPU1   PU        CPNAME=HOST1,
                                        TGP=ATMSVCCR,
                                        .
                                        .
                                        .
                              PATH      GRPNM=CNGRP1,
                                        DLCADDR=(1,C,ATMSVC,EXCLUSIVE),
                                        DLCADDR=(7,BCD,03,00,
00006000,00004000,
                                                00000191,00,
00006000,00004000,
                                                00000191,00),
                                        DLCADDR=(8,X,03,03,03),
                                        DLCADDR=(21,X,0002,
                                                399999999999999999
99980101315045535434 60),
                                        .
                                        .
                                        .
```

Figure 26. Definitions in VTAMLST for the VTAM in HOST2

```
HOST3

                     OSAXCA1  VBUILD    TYPE=XCA

                              PORT      PORTNAME=PORTA,
                                        MEDIUM=ATM

                      CNGRP1  GROUP     DIAL=YES,
                                        CALL=INOUT,
                                        DLCADDR=(1,C,ATMSVC,
VNODE1,EXCLUSIVE),
                                        DLCADDR=(7,BCD,03,00,
00006000,
                                                00004000,00000191,
00),
                                        DLCADDR=(8,X,03,03),
                                        DYNPU=YES,
                                        TGP=ATMSVCCR,
                                        HPR=YES,
                                        .
                                        .
                                        .
                      CNLN1   LINE

                      CNPU1   PU

                      CNLN2   LINE

                      CNPU2   PU

                      CNLN3   LINE

                      CNPU3   PU

                     OSASWT1  VBUILD    TYPE=SWNET

                     SWTPU1   PU        CPNAME=HOST1,
                                        TGP=ATMSVCCR,
                                        .
                                        .
                                        .
                              PATH      GRPNM=CNGRP1,
                                        DLCADDR=(1,C,ATMSVC,EXCLUSIVE),
                                        DLCADDR=(7,BCD,03,00,
00006000,00004000,
                                                00000191,00,
00006000,00004000,
                                                00000191,00),
                                        DLCADDR=(8,X,03,03,03),
                                        DLCADDR=(21,X,0002,
                                                399999999999999999
99980101315045535434 60),
                                        .
                                        .
                                        .
```

Figure 27. Definitions in VTAMLST for the VTAM in HOST3

The type of APPN routes that can use the SVC:

In Figure 25 on page 83 through Figure 27, EXCLUSIVE on the DLCADDR operand with a subfield of 1 on the PATH definition statement in the XCA major node indicates that separate TGs and SVCs can be established. These handle sessions only over connection-network connections. Sessions through connection networks cannot be routed over an SVC already established to the same remote node if that SVC was not established through a connection network.

Meanings of the values coded on the DLCADDR operand are directly related to ATM channel characteristics. For information about what the values indicate, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Name of the virtual node that represents the ATM network:

The DLCADDR operand, with subfield 1 on the GROUP definition statement in the XCA major node, specifies the name of the virtual node representing the ATM network to VTAM is VNODE1.

Maximum number of simultaneous connections:

The number of sets of LINE and PU definition statements in the XCA major node represents the maximum number of simultaneous connections that can be established using the port. In Figure 22 on page 77, the maximum number of simultaneous connections is two. The LINE definition statements serve as placeholders for lines used to connect to remote nodes that are defined either dynamically, or in a switched major node, and are used when those nodes are activated.

Note: The IBM Open Systems Adapter limits the total number of SVCs that can be associated with one port. Refer to *zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference* for information about this limitation.

Who can initiate calls:

The CALL operand on the GROUP definition statement in the XCA major node specifies that the lines in this group can be used for calls initiated by both the VTAM in HOST1 and remote nodes.

Dynamic creation of PUs:

The DYNPU operand on the GROUP definition statement in the XCA major node specifies that PUs are to be dynamically created when remote nodes call in to VTAM. By enabling PUs to be dynamically created, you eliminate the need to code PU and PATH definition statements in the switched major node. If you explicitly define PUs in the switched major node, you lose the major benefit of the connection network: reduction of resource definition.

Route calculation characteristics:

Route calculation characteristics can be specified on the GROUP definition statement in the XCA major node on the following operands:

- CAPACITY
- COSTBYTE
- COSTTYPE
- PDELAY
- SECURITY
- UPARM1
- UPARM2
- UPARM3

They can also be determined by the TGP operand on the GROUP definition statement, which specifies the name of an IBM-supplied APPN TG profile definition. In Figure 25 on page 83 through Figure 27 on page 85, the route calculation characteristics are determined by the TGP operand, which specifies the name of the IBM-supplied profile definition for ATM TGs over SVCs (campus reserved bandwidth).

Best effort indicator, cell rates, and traffic management options:

The DLCADDR operand with a subfield of 7 specifies the following:

Value	Meaning
03	The format of the best effort indicator, cell rates, and traffic management options is defined by the ATM network.
00	Guaranteed bandwidth is desired.
00006000	The forward peak cell rate (cells per second) for cell loss priority 0+1 (CLP=0+1) is 6000.
00004000	The forward sustainable cell rate (cells per second) for (CLP=0) is 4000.
00000191	The forward maximum burst size (cells) for (CLP=0) is 191.
00	Tagging is not requested in the forward direction.

Quality of Service (QoS):

The DLCADDR operand with a subfield of 8 specifies the following:

Value	Meaning
03	The format of the QoS class is defined by the ATM network.
03	The format of the QoS class is defined by the ATM network.
03	The QoS class is connection-oriented for outbound (forward) data traffic.

Establishing a TG over a permanent virtual channel (PVC)

For a PVC connection to be established, both nodes must activate the PVC. To activate the PVC from VTAM, follow these steps:

1. Accept the default value or specify YES on the DYNADJCP start option in the VTAM start list. Or, activate the adjacent control point (ADJCP) major node that defines the remote node with which VTAM can communicate over the TG assigned to this PVC.

   ```
   ATCSTR01  DYNADJCP=YES
   ```

   or

   ```
   VARY ACT ID=ADJCP1
   ```

2. Activate the transport resource list (TRL) major node that defines VTAM connection to the IBM Open Systems Adapter.

   ```
   VARY ACT ID=TRL1
   ```

3. Activate the external communication adapter (XCA) major node that defines the port on the IBM Open Systems Adapter used to access the ATM network.

   ```
   VARY ACT ID=OSAXCA1
   ```

4. Activate the LINE and PU definition statements in the XCA major node that define the line to which the PVC connection can be assigned and the remote node with which VTAM can communicate over the PVC connection.

   ```
   VARY ACT ID=PVCLN1

   VARY ACT ID=PVCPU1
   ```

Establishing a TG over a switched virtual channel (SVC)

For an SVC connection to be established, both nodes must be prepared to call or receive calls.

To prepare VTAM to call or receive calls, follow these steps:

1. Accept the default value or specify YES on the DYNADJCP start option in the VTAM start list. Or, activate the ADJCP major node that defines the remote node with which VTAM can communicate over the TG assigned to this SVC.

   ```
   ATCSTR01  DYNADJCP=YES
   ```

   or

   ```
   VARY ACT ID=ADJCP1
   ```

2. Activate the TRL major node that defines VTAM connection to the IBM Open Systems Adapter.

   ```
   VARY ACT ID=TRL1
   ```

3. Activate the XCA major node that defines the port on the IBM Open Systems Adapter used to access the ATM network.

   ```
   VARY ACT ID=OSAXCA1
   ```

4. Activate the LINE definition statements in the XCA major node that serve as placeholders for the lines used to connect to the remote nodes with which VTAM can communicate over the SVC connection.

   ```
   VARY ACT ID=SVCLN1

   VARY ACT ID=SVCLN2
   ```

   The LINE definition statements in the XCA major node serve as placeholders for lines used to connect to remote nodes that are defined either dynamically, or in a switched major node, and are used when those nodes are activated.

To call a remote node from VTAM, follow these steps:

1. Activate the switched major node that defines the remote node with which VTAM can communicate over the TG assigned to the SVC.

   ```
   VARY ACT ID=OSASWT1
   ```

2. Activate the physical units defined in the switched major node.

   ```
   VARY ACT ID=SWTPU1
   ```

3. Dial the physical units defined in the switched major node.

   ```
   VARY DIAL ID=SWTPU1
   ```

Establishing a TG to a connection network

To establish a connection-network connection, follow these steps:

1. Accept the default value or specify YES on the DYNADJCP start option in the VTAM start list. Or, activate the ADJCP major node that defines the remote nodes with which VTAM can communicate through the connection network.

   ```
   ATCSTR01  DYNADJCP=YES
   ```

   or

   ```
   VARY ACT ID=ADJCP1
   ```

2. Activate the TRL major node that defines VTAM connection to the IBM Open Systems Adapter.

   ```
   VARY ACT ID=TRL1
   ```

3. Activate the XCA major node that defines the port on the IBM Open Systems Adapter used to access the ATM network.

   ```
   VARY ACT ID=OSAXCA1
   ```

4. Activate the LINE definition statements in the XCA major node that serve as placeholders for the SVCs to the remote nodes with which VTAM can communicate over the connection-network connection.

```
VARY ACT ID=CNLN1

VARY ACT ID=CNLN2

VARY ACT ID=CNLN3
```

The LINE definition statements in the XCA major node serve as placeholders for SVCs to remote nodes that are defined either dynamically, or in a switched major node, and are used when sessions to those nodes are established.

APPN multiple network connectivity

APPN multiple network connectivity support enables connectivity between APPN networks, without the exchange of topology information between the networks. APPN multiple network connectivity support may exist between APPN networks that have different network IDs, and also between APPN subnetworks that have the same network ID. APPN multiple network connectivity support can also be used with High-Performance Routing (HPR) support to enable HPR routes between nodes in different subnetworks. See "High-Performance Routing (HPR)" on page 491 for information about HPR support.

APPN multiple network connectivity support uses border nodes at the boundaries between APPN networks or subnetworks. With APPN multiple network connectivity support, full APPN directory and session initiation support is available across these boundaries. APPN directory search activity can be managed through several user-definable parameters for border nodes. Class of Service definitions do not need to be consistent across network or subnetwork boundaries, allowing for independent management.

APPN multiple network connectivity support enables you to eliminate topology exchange between APPN networks or subnetworks when the exchange of such data is undesirable (for example, for networks of different enterprises). Topology exchange is eliminated between border nodes. This way, you can protect smaller subnetworks from the volume of topology in a larger subnetwork or from processing searches for resources that do not exist in the smaller subnetwork. Thus, APPN multiple network connectivity support can be used to manage the increase in topology and directory search activity as an APPN network grows.

For example, the APPN network in Figure 28 on page 90 is divided into four subnetworks. With APPN multiple network connectivity support, full APPN directory and session initiation support is available across all subnetwork boundaries, and yet, the creation of smaller subnetworks protects each from the volume of topology and search processing that would exist in the entire APPN network (which could, of course, extend beyond what is shown).

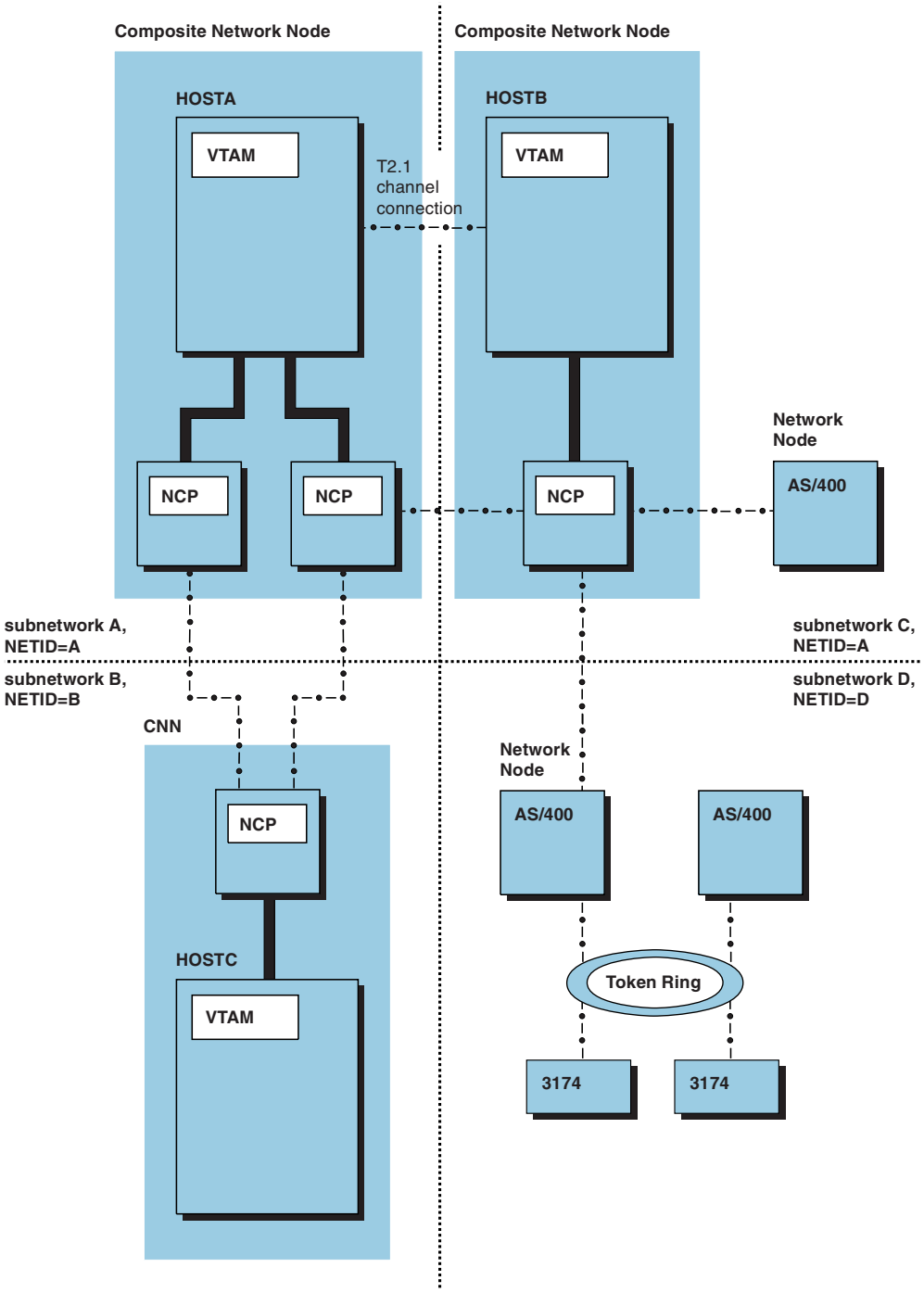


Figure 28. APPN subnetworks through APPN multiple network connectivity support

There are two types of border node boundaries:
- Peripheral subnetwork boundaries
- Extended subnetwork boundaries

Both of these types of border node boundaries are discussed below.

Restriction: APPN multiple network connectivity support cannot be used to connect two APPN networks or subnetworks through a connection network unless

the connection network uses Enterprise Extender and is defined as VNTYPE=GLOBAL. For more information, see “Configuring the EE connection network” on page 161.

Peripheral subnetwork boundaries

A peripheral subnetwork boundary is a connection over a subnetwork boundary between a border node (VTAM with APPN multiple network connectivity support enabled) and a network node with no border node function. The VTAM border node portrays an end node image to the network node, thus preventing topology information from being sent by the network node to VTAM.

VTAM offers the peripheral subnetwork boundary to provide APPN subnetwork interconnection with nodes that do not have border node function. Because the nonborder node in a peripheral boundary is not aware that the subnetwork boundary exists, compared to an extended boundary, there is considerable loss of control over search order, redundant searching, and looping search prevention. For these reasons, it is recommended that you use an extended boundary whenever possible. When using a peripheral boundary, it is wise to use a single subnetwork boundary connection to the peripheral subnetwork (the subnetwork of the nonborder node) to prevent locate requests from exiting the subnetwork over one boundary and then reentering over another boundary. If a peripheral subnetwork has more than one boundary, the user should customize routing by a user exit or routing definitions to ensure that the requests do not reenter the same peripheral subnetwork. This precaution is necessary to avoid possible failed sessions due to a looping locate path.

Guideline: Because these controls are not present for peripheral subnet boundaries, to avoid looping problems, limit connections to a single connection between networks when peripheral subnet boundaries are used.

Notes:

1. A peripheral subnetwork boundary can be used as either the first or last boundary (or both) on a multiple subnetwork search path. If there is an intermediate subnetwork on a multiple subnetwork search path (that is, neither the search originator nor the search target are members of that subnetwork), only extended border nodes can be used to manage the intermediate subnetwork's boundaries.
2. Not all network nodes are capable of attaching to a VTAM border node through a peripheral subnetwork boundary. Ensure the product and release support interoperability with peripheral border nodes (APPN option set 1013) before attempting to establish subnetwork boundaries between VTAM and a network node.
3. It is possible, although strongly discouraged, to get a peripheral boundary between two VTAMs that both support extended border node by setting BN=NO in one of the two nodes. This is strongly discouraged because it bypasses the valuable looping and redundant search controls that are present if the connection is an extended subnetwork boundary.

Extended subnetwork boundaries

An extended subnetwork boundary is a connection between two border nodes that support extended subnetwork boundaries. Although topology flows are still restricted across the boundary, both nodes recognize that a network node exists on the other side, allowing for additional subnetwork hops and increased control over routing.

Note: Some border nodes do not support extended subnetwork boundaries. If one of two communicating border nodes does not support extended subnetwork boundaries, a peripheral subnetwork boundary is used.

APPN multiple network connectivity support

To enable APPN multiple network connectivity support, use the BN=YES start option at a VTAM network node (that is, NODETYPE=NN must also be specified). BN=YES indicates that VTAM is capable of establishing subnetwork boundaries.

Note: NCP Version 7 Release 1 or higher is required for both extended and peripheral subnetwork boundaries through an NCP.

You need to code only the BN=YES start option to take advantage of APPN multiple network connectivity support. Some controls are available for customization:

- An adjacent cluster routing definition list (start with the VBUILD TYPE=ADJCLUST definition statement) can be used to customize routing between subnetworks. The ADJCLUST definition is used when a border node determines it is unable to satisfy a search request [using information in its DS database, TRS database, or symbolic resolution table (SRT)] for a resource that is not in its domain. The border node builds a subnetwork routing list, used to control searching both the native subnetwork and nonnative subnetworks. The subnetwork routing list is built from the ADJCLUST definition, the network topology database (which is aware of all native border nodes), SNVC start option, and learned information from prior searches. The BNDYN and BNORD start options control how this information is used to build the subnetwork routing list.

  The subnetwork routing list is built differently at exit border nodes (a border node receiving a request from another node in its native subnetwork) when compared to entry border nodes (a border node receiving a request across a subnetwork boundary from a nonnative node) and origin border nodes (a border node that is the OLU, or a CDS where the OLU is not a border node). An exit border node includes only adjacent nonnative network nodes in its subnetwork routing list. Entry and origin border nodes include these also, as well as its own name (used to search the native APPN subnetwork) and all other border nodes in the native APPN subnetwork. Any nodes in the subnetwork routing list that cannot be routed to are automatically pruned from the list.

  Once the subnetwork routing list is built, the border node sequentially sends a directed Locate to each node on the list, progressing down the list only if a negative reply is received. If a border node encounters its name in the subnetwork routing list, it performs searching of its locally attached subarea network and native APPN subnetwork, possibly including a network broadcast search and interchange node search. The interchange node search allows any subarea network attached to the native APPN subnetwork to be searched as well.
- The BNDYN start option is used to define how VTAM adds entries to the adjacent cluster routing table. By default, VTAM adds a limited set of nodes to the table.

  Notes:

  1. When coding BNDYN=NONE, you will need to code an adjacent cluster table, otherwise session requests might not leave this node. Be sure to code this node's name in the table to allow the native subnet to be searched.

2. In order to search its own network, when BNDYN=NONE and VTAM is the NNS(OLU) or CP(OLU), a minimum of two adjacent cluster tables are needed (one to use when the NETID of the resource is known, and a default adjacent cluster table to use when the NETID of the resource is not known). Adjacent cluster tables for other NETIDs might also be needed when VTAM directory information does not reflect the actual location of the DLU. *z/OS Communications Server: SNA Resource Definition Reference* provides an example.

- The BNORD start option is used to control the search order when searching across subnetwork boundaries. By default, VTAM gives preference to nodes for which the most recent search was successful and nodes whose network ID matches that of the destination logical unit (DLU). Specify BNORD=DEFINED to use the defined search order.
- The SNVC start option is a number in the range of 1–255 that specifies the maximum number of subnetworks to which a request can be forwarded, including this subnetwork. For example, the default for the SNVC start option is 3, indicating that a request can be routed from this subnetwork across up to two subnetwork boundaries, for a total of three subnetworks visited when you count the original subnetwork. Given the connectivity shown in Figure 28 on page 90, the default would not allow the routing of a request from subnetwork D to subnetwork B (SNVC=4 is needed).

  SNVC processing is summarized as follows:

  – When a request is received across a subnetwork boundary, a border node decrements the SNVC, if present. If the SNVC is 0 after being decremented, the request is rejected without further processing.

    When a request is received from another node in the same subnetwork as the border node, the border node does not decrement the SNVC.

  – Before sending a request to another node, a border node compares the SNVC on the request received (after decrementing, if received over a subnetwork boundary) to the SNVC defined at the border node for the destination node. The SNVC on the request sent is set to the lower of these two values. If no SNVC was received on the request, the defined value is used.

    The SNVC defined at a border node is determined from the SNVC start option and the SNVC values specified in the adjacent cluster routing definition list, with the adjacent cluster routing definition list taking priority.
- Class of Service mapping definitions can be placed in a COS mapping table to map the following:
  – Nonnative COS definitions to native COS definitions
  – Native COS definitions to nonnative COS definitions
- The APPNCOS start option is used to specify the default COS used for requests received over a subnetwork boundary with an unrecognizable COS listed. The value for APPNCOS is used only after COS mapping is attempted. The default value is NONE.
- The CACHETI start option defines the number of seconds that routing information about a previous locate search is stored, if applicable. The default is eight minutes.
- The ALIASRCH operand is specified on the ADJCP minor node definition for an adjacent non-native CP. The ALIASRCH operand is valid only when BN=YES is specified. Code ALIASRCH=NO if you want this border node to halt inbound ALIAS searches from the adjacent non-native node.
- The AUTHNETS operand is specified on the ADJCP minor node definition for an adjacent non-native CP. The AUTHNETS operand is valid only when

BN=YES is specified. Specify AUTHNETS= if you want to limit searches through this border node from the adjacent non-native node, based on the NETID value of the DLU resource.

- The NATIVE operand is specified on the ADJCP definition statement in the adjacent CP major node, or on a PU definition statement defining an APPN connection. The NATIVE operand is valid only when BN=YES is specified. Code NATIVE=NO to define a subnetwork boundary between this node and the named adjacent CP, or between this node and the CP represented by the PU statement. Use NATIVE=NO when both nodes have the same network ID, but a subnetwork boundary is desired. For example, in Figure 28 on page 90, NATIVE=NO was used to create the subnetwork boundary between subnetwork A and subnetwork C. The NATIVE operand is required on only one side of a network or subnetwork boundary, though it is allowed on both sides as long as they do not conflict.

  Note: Make sure to use the NATIVE operand consistently in your network. For example, do not specify a native connection (NATIVE=YES) from one node to a second node when the second node has a native connection to a third node that is nonnative (NATIVE=NO) to the first node. This is not valid.

- The RTPONLY operand is specified on the ADJCP definition statement in the adjacent CP major node. The RTPONLY operand is valid only when BN=YES is specified. Code RTPONLY=YES on the ADJCP definitions for nonnative nodes if you want this border node to maintain awareness of all sessions established to, from, or through the adjacent nonnative node being defined. RTPONLY=YES can be used only when the HPR start option specified RTP as the first operand. Due to the potential increase in storage and CPU use, RTPONLY=YES is recommended only when there is a crucial need to maintain session awareness (such as for accounting purposes).

  By coding RTPONLY=YES, you are instructing VTAM to disallow use of the high performance routing (HPR) automatic network routing (ANR) function for any new RTPs that are established or path switched to, from, or through the adjacent nonnative node being defined. (Allowing the use of ANR could result in RTPs being established through this border node, thereby preventing the border node from maintaining awareness of any sessions that use these RTPs.) Instead of allowing RTPs to be established through this border node, coding RTPONLY=YES will result in VTAM forcing these RTPs to terminate on this border node or forcing the use of intermediate session routing (ISR) instead of HPR (or a combination of the two).

  Restrictions:

  1. Use of RTPONLY=YES at any APPN subnetwork boundary on a session setup path prevents the use of Global VRNs (GVRNs) for intersubnetwork connectivity, since using GVRNs could result in sessions being established across this subnetwork boundary without this border node maintaining awareness of these sessions.
  2. Use of RTPONLY=YES can result in an increase in network traffic in the form of additional Route_Setup flows used for RTP establishment. These additional Route_Setup flows will occur only during the establishment of sessions that cross subnetwork boundaries defined with RTPONLY=YES.
  3. Use of RTPONLY=YES can result in an increase in storage and CPU utilization due to VTAM maintaining awareness of these sessions and performing ISR instead of HPR/ANR for these sessions.

Virtual-route-based transmission groups

A virtual-route-based transmission group (VR-based TG) connects two APPN-capable VTAM nodes through a subarea network and allows them to act as APPN peers. The VR-based TG (whose TG number is always 255) then functions as any other TG in an APPN network. APPN CP-CP sessions can be established between these two nodes, which allows all APPN functions to be performed through the subarea network. For example, if the two nodes are attached to disjoint APPN subnetworks, the topologies of these two subnetworks are exchanged over the CP-CP sessions, thereby combining them into one large APPN network. Resource location and session establishment can also be performed through the subarea network using APPN protocols, rather than transforming APPN flows to subarea flows and back to APPN flows. In addition, VR-based TG support can be used in conjunction with HPR support, enabling an HPR route to traverse a VR-based TG. See "High-Performance Routing (HPR)" on page 491 for information about HPR support.

Only one VR-based TG can be defined between any two VTAM nodes, but this does not limit the use of this TG to only one virtual route (VR). In fact, the VR-based TG is used to represent all possible VRs between any two subareas in the domains of the two VTAMs. For example, in Figure 29 on page 96, if an application on HOSTA starts a session with an application on HOSTB, the VR-based TG can be selected as the route for this session. When this occurs, VR1 can be assigned as the underlying VR for the session, which will use the channel-to-channel connection between the two hosts. If an independent LU on network node A starts a session with an independent LU on network node D, the VR-based TG can be selected for a portion of the session route. However, in this case, VR2 can be assigned as the underlying VR, so that the direct connection between the two NCPs will be used.

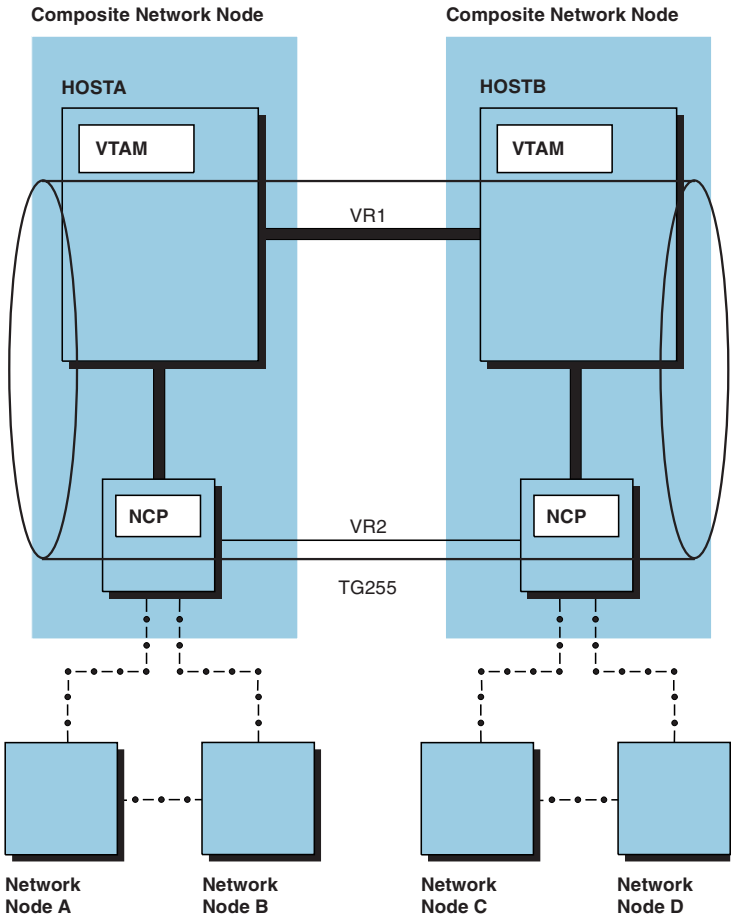


Figure 29. VR-based TG between composite network nodes

Because an underlying VR is always assigned to sessions established using VR-based TGs, both APPN and subarea functions must be enabled on both nodes. This means that each node must be defined as an interchange node or a migration data host. (That is, the NODETYPE start option must be coded and each host must define a unique HOSTSA number.) Also, because a limited number of subarea flows must be sent between these two nodes to determine which underlying VR should be used, VR-based TG support also depends on a CDRM session being active between the two nodes.

Notes:

1. It is possible for a session route to be calculated that includes two or more consecutive VR-based TGs, as shown in Figure 30 on page 97. When this occurs, consecutive VR-based TGs are merged together into one VR-based TG. This is referred to as RSCV pruning. The resulting VR-based TG appears to directly connect the first and last VTAM nodes (the endpoints of the consecutive VR-based TGs), even though a VR-based TG was not actually defined between these two nodes. For this reason, it is not sufficient to have active CDRM sessions only between VR-based TG partner nodes. The existing subarea requirement that every SSCP in the network have an active CDRM session with every other SSCP in that network still holds.
2. VR-based TGs cannot be used across APPN or subarea network boundaries (that is, between nodes with different network IDs) or across APPN subnetwork boundaries created using the NATIVE=NO operand. For information on APPN

subnetwork boundaries created using the NATIVE=NO operand, see “APPN multiple network connectivity” on page 89.

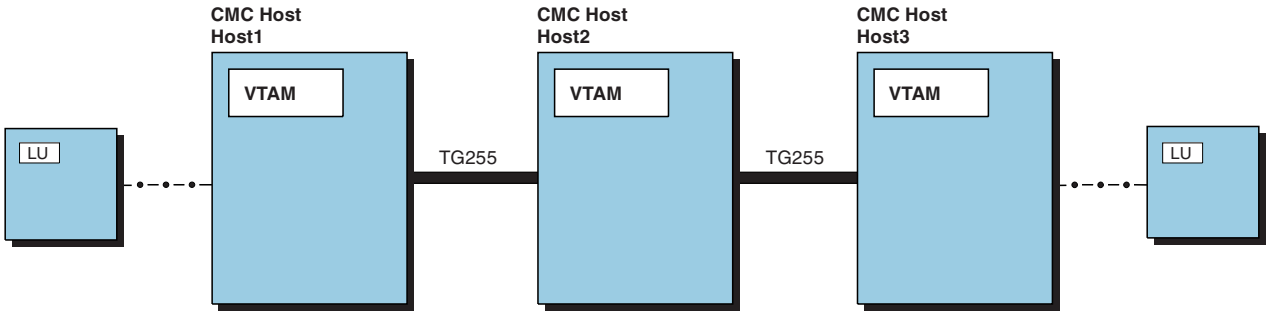


Figure 30. Multiple contiguous VR-based TGs

Defining a VR-based TG

To define a VR-based TG between two VTAM APPN nodes, code VRTG=YES as a start option at both VTAMs or on the CDRM definition statement for the adjacent VTAM. If VRTG=YES is coded at both VTAMs, then a VR-based TG is activated automatically when the CDRM session with the adjacent VTAM is established. If the VR-based TG connects two VTAM network nodes, a topology database update (TDU) is broadcast throughout the network, so that the VR-based TG is added to the topology database of all network nodes. If either VTAM is defined as an end node, then the VR-based TG is registered with the end node network node server. Similarly, when the CDRM session between these two VTAMs terminates, the VR-based TG is deactivated and the appropriate nodes are notified that the TG is no longer active (either by sending another TDU through the network or by deleting a registered TG with an end node network node server).

Note: To enable VR-based TGs with a specific adjacent CDRM without disrupting existing LU-LU sessions, use the SAVESESS operand when deactivating the adjacent CDRM. Keep in mind, however, that deactivation of a CDRM with SAVESESS disassociates the active sessions from the CDRM. Subsequent activations and deactivations of the CDRM have no effect on these sessions.

If there are no CP-CP sessions active between the two VTAM nodes, CP-CP session establishment is automatically initiated when a VR-based TG is activated. The CP-CP sessions might use the VR-based TG or any other APPN link between the two VTAMs that supports CP-CP sessions. If CP-CP sessions are not desired over a VR-based TG, code VRTGCPCP=NO as a start option or on the CDRM definition statement for the adjacent VTAM. Note, however, that an alternate CP-CP session path must exist between the two VTAMs, so that network traffic can be sent. For example, in Figure 29 on page 96, if a connection existed between network node B and network node C and CP-CP sessions were established between them, VRTGCPCP=NO could be used to prevent CP-CP sessions from being established over the VR-based TG between the two VTAMs. When a session between two VTAM applications is initiated, the network traffic sent to locate the target resource would have to be sent through network node B and network node C, but the session could still be established over the direct VR-based TG connection.

The choice of which APPN TGs can be used for a given session is based on the APPN Class of Service and the characteristics defined for the links in the APPN network. As such, APPN link characteristics must be defined for all VR-based TGs as well. This is done by coding the TGP operand on the adjacent CDRM definition

statement, to specify the TG profile that describes the desired characteristics of the VR-based TG. For general information on how an APPN TG is chosen for a given session, see "Network routing and resource location for APPN nodes" on page 325. For details on the operands that you can code in a transmission group profile, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

When a VR-based TG has been selected for a session, the mode name for the session is used to select the subarea Class of Service, which is then used to determine which underlying VRs can be used for the session. Therefore, it is necessary to ensure that VRs and ERs are defined for every subarea Class of Service that might be selected when a VR-based TG is included in the route that is calculated during APPN session establishment.

VR-based TG recommendations

When defining the CP-CP sessions in an APPN network, it is important to remember that APPN networks do not require meshed CP-CP sessions the way subarea networks require meshed SSCP-SSCP sessions. In APPN networks, direct CP-CP sessions between all pairs of network nodes are impractical due to the number of network nodes, and it is not possible in many cases because CP-CP sessions are allowed only with physically adjacent nodes. Instead, only a path of active CP-CP sessions between the origin and destination nodes is needed to deliver a session establishment request, determine an appropriate session route, and establish the session.

In fact, too many CP-CP sessions in an APPN network can cause network nodes to spend too much time processing duplicate requests. For example, when network topology database updates and broadcast search requests are performed, every network node must send or receive every request over every pair of CP-CP sessions that it has with other network nodes. (This ensures that every network node receives the request.) For this reason, establishing CP-CP sessions with every adjacent node might also prove to be undesirable.

At the same time, not enough CP-CP sessions in an APPN network can cause portions of the network to become disjoint, so that CP-CP sessions fail and network topology database updates and broadcast search requests can no longer be delivered to every network node. Therefore, you need to consider (which takes into account network availability requirements) the number of CP-CP sessions and the placement of those sessions. The ideal configuration is one that minimizes the number of CP-CP sessions while still ensuring that, at any time, every network node is logically connected to every other network node by a path of active CP-CP sessions.

The same considerations are necessary when defining VR-based TGs and CP-CP sessions through a subarea network. The following recommendations can simplify the task of determining how many VR-based TGs and CP-CP sessions should be defined, and where in the network they should be located.

Physical adjacency

VR-based TGs and CP-CP sessions should be defined only between physically adjacent VTAM nodes (or composite network nodes). They should not be defined through an intermediate VTAM. For example, in Figure 30 on page 97, VR-based TGs and CP-CP sessions should only be defined between HOST1 and HOST2 and between HOST2 and HOST3. CP-CP sessions between HOST1 and HOST3 (through HOST2) are unnecessary, because session setup flows can be sent from HOST1 to HOST3 using HOST2 as an intermediate APPN node. Also, because RSCV pruning is performed, a two-hop VR-based TG route from HOST1 to HOST2

to HOST3 is pruned to a one-hop VR-based TG route from HOST1 directly to HOST3. This pruning allows a VR-based TG to be used between HOST1 and HOST3, even though it was not explicitly defined.

CMC connectivity

If the subarea network is set up as a communications management configuration (CMC) with multiple VTAMs acting as CMCs, there must be enough VR-based TGs and CP-CP sessions defined between the CMCs to satisfy network availability requirements, based on the considerations described earlier. Although some redundant VR-based TGs and CP-CP sessions might have to be defined, to provide connectivity to all network nodes when TGs carrying CP-CP sessions fail abnormally, meshed VR-based TGs and CP-CP sessions between every pair of VTAM nodes will rarely be required, even when every VTAM is physically adjacent to every other VTAM.

Migration data host connectivity

Migration data hosts are VTAM end nodes that also support CDRM sessions. As end nodes, migration data hosts require the services of a network node server in order to establish sessions with resources in other nodes. The network node server for VTAM end nodes is typically a VTAM network node or composite network node, possibly a CMC host.

If all of the migration data hosts are served by the same network node server, then VR-based TGs and CP-CP sessions need only be defined between each migration data host and its network node server (not between pairs of migration data hosts), as shown in Figure 31. In some cases, a session is requested between two migration data hosts, and the route is calculated as a two-hop VR-based TG route through the network node server. RSCV pruning reduces this route to a one-hop VR-based TG route directly between the two migration data hosts, and an appropriate VR is selected for the session.

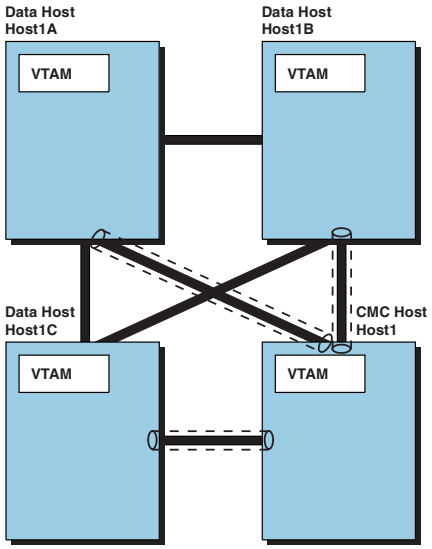


Figure 31. VR-based TGs in a communication management configuration

The same VR selection process is used when the migration data hosts are served by different network node servers and there is a VR-based TG path between the two network node servers. In this case, each migration data host should define VR-based TGs to all potential network node servers. In some cases, a session is requested between two migration data hosts, and the route is calculated as a

two-hop VR-based TG route through the network node server. RSCV pruning reduces this route to a one-hop VR-based TG route directly between the two migration data hosts, and an appropriate VR is selected for the session.

If two migration data hosts are served by different network node servers and no VR-based TG path exists between the two network node servers, each migration data host should define VR-based TGs to each adjacent VTAM network node and migration data host to improve the chances of finding a route between the migration data hosts that consists of entirely VR-based TGs.

Selecting the network node server for end nodes

You might want to control which network node is selected by an end node as its network node server (NNS). For example, you might want to exclude a particular network node from network node server responsibilities because acting as a server involves some overhead, such as originating search requests or executing domain broadcasts. You might also want to specify which network nodes should be allowed as backup network node servers in the event the primary network node server becomes unavailable.

The network node server list allows you to indicate the network nodes that can act as servers for an end node. An end node detects adjacent network nodes capable of acting as its network node server through link activation or system definition. If you do not use a network node server list, the adjacent network node chosen as an end node server is directly related to when the end node becomes aware of that particular network node, relative to other adjacent network nodes. Without a network node server list, an end node establishes a CP-CP session with the first such network node that it becomes aware of, and this network node acts as the end node server. There is no predictability in this scheme; it is all timing-dependent. Therefore, if you want the network node server list to be used, you need to activate the list prior to activating links to network nodes. The configuration list should have the network node server list as one of the first resources to activate.

Initially, a CP-CP-capable physical connection might be temporarily unavailable between an end node and the network node that you prefer to have as the end node primary network node server. This situation requires the end node to establish CP-CP sessions with a backup server specified by the network node server list or, if no network node server list has been defined, with the first network node it can activate a CP-CP capable physical connection with.

By specifying the name of the end node preferred network node server on the end node NNSPREF start option, you allow the end node to automatically switch CP-CP sessions from its current NNS to the preferred NNS when a CP-CP-capable connection to the preferred network node server becomes available.

Creating a network node server list

To restrict the network node server to one that has the same level of support for SLU-initiated sessions as the end node, you can use the SLUINIT operand on the NETSRVR definition statement. For example, to require a network node server to be able to support SLU-initiated sessions, you can define SLUINIT=REQ. The SLUINIT=REQ specifies that CP-CP sessions can only be established with a network node that supports SLU-initiated sessions. SLUINIT=REQUIRED is the

default. If you define SLUINIT=OPT, CP-CP sessions are established with a network node server regardless of whether the network node supports SLU-initiated sessions.

You can create a prioritized network node server list by allowing the ORDER operand to default to FIRST on the NETSRVR definition statement. ORDER=FIRST causes the end node to always start at the top of the list when searching for a network node server. If ORDER=NEXT, the end node selects the network node in the list following the last node selected. ORDER=NEXT gives no preference to any node on the list.

To create a network node server list for an end node, create a VTAMLST member containing a VBUILD TYPE=NETSRVR definition statement, and one or more NETSRVR definition statements for each network node that you want in that network node server list. You can also include a nameless entry to allow the end node to select any other known adjacent network node that meets the defined criteria as its network node server. A sample network node server list follows.

```
       VBUILD TYPE=NETSRVR
CP1A   NETSRVR  NETID=NETA
CP2A   NETSRVR  NETID=NETA
CP3A   NETSRVR  NETID=NETA
       NETSRVR
```

Note: Any network node named on a NETSRVR definition statement previously considered and rejected as a network node server is not retried during nameless entry processing.

Activating, replacing, and displaying a network node server list

You can activate a network node server list at startup by specifying the appropriate VTAMLST member name in your configuration list, or by using the VARY ACT command at an end node. If you use the VARY ACT command to activate the network node server list after a CP-CP session has already been established, the list is not used until the existing CP-CP session between the end node and its current network node server is lost or terminated. Use the VARY TERM command at the end node to deactivate the CP-CP session between the end node and the network node acting as the end node server. VTAM will then use the network node server list in selecting a new server for the end node.

You can also dynamically replace an existing network node server list. To do this, create a new list and, as described above, use the VARY ACT command to activate it and the VARY TERM command to force VTAM to use the new list.

To display server information for end nodes, use the DISPLAY NETSRVR command at an end node to display the end node network node server list, the end node current network node server, and other network nodes known to the end node.

Using the NNSPREF start option

The NNSPREF start option gives the end node the ability to dynamically switch CP-CP sessions from the current network node server (NNS) to a different (preferred) NNS. This switch occurs automatically in either of the following circumstances:

- A physical connection is successfully established between the end node and the preferred NNS.

- The end node operator manually modifies the NNS preference from the name of one adjacent network nodes to the name of another adjacent network node (NN). In this case, a physical connection to the NN that is taking over the NNS function for the end node must exist.

If you do not specify NNSPREF, or specify NNSPREF with a value of NONE (the default), dynamic NNS switching is not enabled. You can modify NNSPREF using the MODIFY VTAMOPTS command. The value specified on MODIFY NNSPREF effectively gets dynamically added to the existing NETSRVR list table that is active. (To display the value of NNSPREF, use the D NET, VTAMOPTS,OPTION=NNSPREF command.) For more information about the DISPLAY VTAMOPTS command, refer to *z/OS Communications Server: SNA Operation*.

Chapter 5. Connecting a subarea node to VTAM

If a VTAM is connected to a subarea node, a physical link between the nodes must be connected and functional at both ends. A subarea node can be a host processor (containing another VTAM) or a communication controller (containing an NCP).

The following list shows where to find information on the following connections:

- "Connecting two VTAMs using channels"
- "Connecting two VTAMs through communication controllers" on page 106
- "Connecting two VTAMs using an external communication adapter" on page 112
- "Connecting an NCP to VTAM" on page 115
- "Connecting an NCP to another NCP" on page 127

Note: Earlier releases of VTAM support type 2.1 casual connection, a way of connecting two VTAMs in a peer-to-peer relationship using low-entry networking (LEN). While type 2.1 casual connection is still supported, IBM recommends that you use APPN to connect two VTAMs in a peer-to-peer relationship. See Chapter 4, "Connecting an APPN node to VTAM," on page 49 for information on connecting two VTAMs using APPN.

Connecting two VTAMs using channels

The following channel connections can be used to connect two VTAM subareas:

- Channel-to-channel adapter connection
- "Multipath channel connections" on page 105

Channel-to-channel adapter connection

A channel-to-channel adapter connection between two host processors is defined using a channel-attachment major node. You define two channel-attachment major nodes for each connection. Each channel-attachment major node represents each VTAM's view of the connection. A GROUP definition statement defines the type of links that follow and also carries other operands that sift down to following definition statements that do not explicitly override them.

Code one LINE definition statement for each channel-to-channel adapter. The LINE definition statement defines to VTAM the characteristics of the line side of the adapter. Code one PU definition statement for each LINE definition statement.

The definition statements in the following example define the channel link between the two VTAMs shown in Figure 32 on page 105. On the GROUP definition statement, the missing interrupt handler (MIH) operand is coded with a value of YES so that the channel link becomes inoperative after the time period specified on the REPLYTO operand expires. (Otherwise, the channel link appears operative, but VTAM cannot use it.) The default for the MIH operand is NO.

The LINE definition statement is used to specify the channel unit address (ADDRESS=51A) used in VTAM at one end of the channel link.

You can use the following operands on the LINE definition:

© Copyright IBM Corp. 1984, 2011

- MAXBFRU defines the number of 4 KB pages of storage that are used to buffer PIUs for transmission over the channel link. The PIUs are blocked into the 4 KB pages, transferred over the channel, and then deblocked into VTAM I/O buffers in the other VTAM.
- DELAY can be used to delay the operation of the data transfer so that more PIUs can be buffered and transferred in a single channel I/O operation.

For more information about selecting values for these operands, see Chapter 23, "Tuning VTAM for your environment," on page 663.

Though the PU definition statement identifies the physical unit type at the other end of the connection to be an NCP (PUTYPE=4), the channel link actually connects to another VTAM, a physical unit type 5. The TGN operand specifies a transmission group number for the channel-link transmission group. Because parallel transmission groups between two subareas require unique TG numbers, the TGN operand is used.

```
TSCTCAX VBUILD TYPE=CA              CHANNEL-TO-CHANNEL (CTC) CONNECTION
*
CTCGRPX  GROUP LNCTL=CTCA,           CTC CONNECTION
               MIH=YES,              LINK INOPERATIVE - START I/O TIMEOUT
               REPLYTO=10.0,         CHANNEL PROGRAM TIMEOUT

⋮
               ISTATUS=ACTIVE        VTAM INITIAL STATUS
*
CTCSA51L LINE  ADDRESS=51A,          CHANNEL UNIT ADDRESS
               DELAY=.1,             CHANNEL DELAY TIMER
               MAXBFRU=8,            4K PAGE BUFFER FOR CTC DATA TRANSFER

⋮
               ISTATUS=ACTIVE        VTAM INITIAL STATUS
*
CTCSA51P PU    PUTYPE=4,             PHYSICAL UNIT TYPE = VTAM
               TGN=3,                TRANSMISSION GROUP NUMBER

⋮
               ISTATUS=ACTIVE        VTAM INITIAL STATUS
```

The capability of having parallel transmission groups allows you to add multiple channel attachments to an adjacent subarea host. With parallel transmission group support, VTAM permits the transmission group number to be specified on the PU definition statement of the channel-attachment major node and on the PATH definition statement. For information about how this affects network routing, see "Parallel sessions using parallel transmission groups" on page 368.

Although you can have as many as 255 transmission groups, only 16 of these can be defined because the maximum number of explicit routes that can be defined between two adjacent VTAMs is 16. Each transmission group is single-link capable only. Figure 32 on page 105 illustrates a possible configuration of parallel transmission groups.

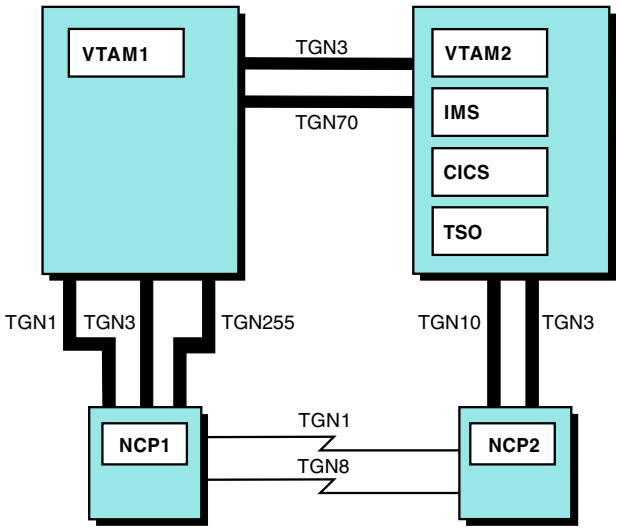


Figure 32. Parallel transmission groups in multiple domain environment with NCP

Two channel-to-channel connections are configured in Figure 32. A LINE and a PU definition statement are required for each of these channel connections between the two VTAMs. The TGN operand for one of the channel connections specifies a transmission group number of 3, as shown in the previous example. The LINE definition statement that defines the other channel-link connection must have a TGN operand that specifies a transmission group number of 70. Parallel transmission group support limits each transmission group to a single channel-to-channel link. The two channel-link connections (TG3 and TG70) between the VTAMs cannot be defined as a single logical transmission group composed of the two channel links.

Multipath channel connections

Multipath channel (MPC) connections allow you to code a single transmission group (TG) for host-to-host communication that uses multiple write-direction and read-direction subchannels. The subchannels can be assigned to one or more physical channels based on how you have defined your I/O subsystem to the operating system. Because each subchannel operates in only one direction, the half-duplex turnaround time that occurs with other channel-to-channel connections is reduced.

To define an MPC connection between two subarea nodes, code a channel-attachment major node (VBUILD=CA) with LNCTL=MPC on the GROUP definition statement. For more detailed information about defining subarea MPC connections, refer to *z/OS Communications Server: SNA Resource Definition Samples*.

If you code a TG in which the subchannels are divided between two physical channels, you can increase availability because the TG will have a path to use, even if one physical channel is down. Because each TG can use more than one channel, and because the turnaround time required for half-duplex is reduced, throughput is increased.

The subchannels on the physical channel are represented by the subchannel addresses coded on the READ and WRITE operands on the LINE definition statement in the channel-attachment major node. One READ subchannel in one host and the corresponding WRITE subchannel in the other host form a complete path. The READ subchannel address and the corresponding WRITE subchannel

address must reference the same physical connection between the two nodes; the two addresses do not need to be identical. You should consider the ratio of inbound and outbound traffic for a host in deciding how many write subchannels and read subchannels you want to code.

If multiple write subchannels are used, traffic is distributed among the subchannels as traffic requests increase. Arriving data is resequenced before being presented to the application. Priority is assigned to the write subchannels in the order that they are defined.

If you want two TGs, with each TG dividing its subchannels across two parallel physical channels, you have to code two groups, one for each TG. The LINE definition statement for each TG will include subchannel addresses on the READ and WRITE operands from each of the two physical channels.

If a subchannel becomes inoperative, the TG remains active as long as one subchannel is available in each direction. If the last subchannel in either direction becomes inoperative, issue a VARY INACT command, correct the problem (all subchannels must be allocatable when activated or activation will fail), and then issue a VARY ACT command to reactivate the major node.

Note: MPC connections in an APPN network enable two nodes to communicate using APPN protocols over MPC connections. For a description of MPC connection support in an APPN network, see "Multipath channel connections" on page 51.

Connecting two VTAMs through communication controllers

The following NCP connections can be used to connect domains:
- Nonswitched SDLC subarea connection
- "Switched SDLC subarea connection" on page 108
- "X.25 subarea connection (NPSI support)" on page 110
- "Token-ring connection (NTRI support)" on page 111

Nonswitched SDLC subarea connection

A nonswitched synchronous data link control (SDLC) link between two NCPs can be used to connect two VTAM domains. In Figure 32 on page 105, the SDLC connection between NCP1 and NCP2 is used to connect the two VTAM domains (VTAM1 and VTAM2). This connection can operate at whatever speed the communication controller and NCP are capable of supporting. Each end of the SDLC connection is defined in each NCP using NCP definition statements. The connection can be composed of multiple links in the same transmission group because it is a connection between NCPs. To activate the link, one of the following must occur:

- If each VTAM controls its respective NCP (NCP1 and NCP2), each VTAM domain (VTAM1 and VTAM2) must activate its end of the connection.
- If one VTAM controls both NCPs (NCP1 and NCP2), the RNAME operand must be coded on either the PCCU definition statement or the operator command that is used to activate NCP.

The GROUP and SDLCST definition statements indicate to the NCP the link characteristics that are to be used when the NCP is operating as the primary or secondary end of the link.

The LINE and PU definition statements in the NCP define the link and link station connection between the NCPs. The majority of the operands used to define this connection are associated with NCP. However, you can specify VTAM-only operands such as ISTATUS. For information on the operands that are VTAM-only and are applicable to this connection, refer to "Connecting an NCP to VTAM" on page 115.

The sample coding in "Sample coding for SDLC primary and secondary links and link stations" and "Sample coding for an intermediate routing node link" illustrates the NCP definitions used to define a high-speed connection (1.47 megabytes per second) between NCP1 and NCP2. A set of definitions is required in each NCP.

Sample coding for SDLC primary and secondary links and link stations

The following sample illustrates how to code for SDLC primary and secondary links and link stations.

```
PRIM     SDLCST GROUP=GPRIM,               PRIMARY GROUP NAME
                MAXOUT=127,                LINK LEVEL RESPONSE
                MODE=PRIMARY,              PRIMARY MODE
                PASSLIM=254,               MAXIMUM PIU SEND COUNT
                RETRIES=(7,3,5),           RETRY COUNTS

⋮
                SERVLIM=254                SERVICE ORDER TABLE SCAN
*
SEC      SDLCST GROUP=GSEC,                SECONDARY GROUP NAME
                MAXOUT=127,                LINK LEVEL RESPONSE
                MODE=SECONDARY,            SECONDARY MODE
                PASSLIM=254,               MAXIMUM SEND COUNT
                RETRIES=(7),               RETRY COUNT

⋮
                TADDR=C1                   SECONDARY POLL ADDRESS
*
*
GPRIM    GROUP  LNCTL=SDLC,                SDLC LINK CONTROL
                TYPE=NCP,                  NCP LINK
                DIAL=NO,                   NONSWITCHED CONNECTION
                REPLYTO=(1.0,3.0),         REPLY TIMEOUT
                MODE=PRIMARY,              PRIMARY link station

⋮
                MODULO=128                 MODULO FOR LINK LEVEL RESPONSE
*
GSEC     GROUP  LNCTL=SDLC,                SDLC LINK CONTROL
                TYPE=NCP,                  NCP LINK
                DIAL=NO,                   NONSWITCHED CONNECTION
                REPLYTO=(NONE,NONE),       NO REPLY TIMEOUT ON SECONDARY
                MODE=SECONDARY,            SECONDARY link station
                MODULO=128,                MODULO FOR LINK LEVEL RESPONSE

⋮
                ACTIVTO=120                ACTIVATE TIMEOUT
```

Sample coding for an intermediate routing node link

The following sample illustrates how to code for an intermediate routing node link.

```
G6INN    GROUP ACTIVTO=120,                ACTIVATE TIMEOUT
               DIAL=NO,                    NONSWITCHED CONNECTION
               LNCTL=SDLC,                 SDLC LINK CONTROL
               REPLYTO=(10.0,15.0),        RETRY COUNTS
```

```
               TYPE=NCP,                NCP LINK
⋮
               ISTATUS=ACTIVE           INITIAL VTAM STATUS
*
L51028   LINE  ADDRESS=(1028,FULL),     LINE ADDRESS
               SPEED=1474560,           LINK SPEED
               CLOCKNG=EXT,             EXTERNAL CLOCKING
               DUPLEX=FULL,             FULL DUPLEX OPERATION
               PAUSE=(1.0,1.0),         PRIMARY AND SECONDARY PAUSE
               SDLCST=(PRIM,SEC),       SDLCST PRIMARY/SECONDARY
⋮
               ISTATUS=ACTIVE           INITIAL VTAM STATUS
*
C51028A  PU    ANS=CONTINUE,            CONTINUE AFTER VTAM FAILURE
               PUTYPE=4,                PHYSICAL UNIT TYPE = NCP
               TGN=8,                   TRANSMISSION GROUP NUMBER
⋮
               ISTATUS=ACTIVE           INITIAL VTAM STATUS
```

Switched SDLC subarea connection

A switched SDLC line between two NCPs can be used to connect two VTAM domains. One or more of switched lines can be defined as a switched line group in an NCP major node definition. Associated with each line in the NCP major node definition is a PU definition statement to represent a port for the actual physical unit that can be connected to the NCP on that line. The actual physical units must be defined to VTAM in the switched major node.

Note: An automatic calling unit (ACU) is required for switched SDLC subarea connection.

The PU definition statements in the switched major node define subareas attached over switched lines that can be dialed in to or dialed out from a communication controller using NCP. These lines are defined in switched line groups in an NCP major node and a VTAM switched major node. For dial-out operations, the PATH definition statement defines the possible paths to be used to establish connection between the communication controller or host processor and the other subarea.

Sample coding for a switched major node for a switched subarea

The following coding is an example of definition statements for a switched major node for switched subarea connections.

```
SWN112   VBUILD TYPE=SWNET          SDLC SWITCHED MAJOR NODE
*
PU2      PU     ADDR=C1,            PUTYPE4 STATION ADDRESS
                ANS=CONTINUE,       LINK CONNECTION AFTER VTAM FAILURE
                MAXDATA=4096,       MAXIMUM DATA TRANSFER
                SUBAREA=112,        SUBAREA ADDRESS NUMBER
                IDNUM=11201,        IDENTIFICATION NUMBER
                PUTYPE=4,           PHYSICAL UNIT TYPE = NCP
                NETID=USIBMCTS,     NETWORK IDENTIFIER
⋮
                TGN=19              TRANSMISSION GROUP NUMBER
*
         PATH   DIALNO=1234567,     DIAL OUT TELEPHONE NUMBER
                GID=1,              GROUP IDENTIFIER
                PID=1,              PATH IDENTIFIER
                REDIAL=2,           RETRY DIAL OPERATION TWICE
```

```
⋮
                GRPNM=GN112SW       NCP GROUP NAME
```

The IDNUM operands associated with both ends of a specific switched link must match for the connection to be established. If there are multiple switched links between two subarea nodes, the IDNUM associated with each link between the subarea nodes must be different for each link. Additionally, the combinations of IDNUM, NETID, and SUBAREA must be unique.

Sample NCP definition statements for switched subarea link connection

The following code illustrates some of the definition statements that are used in the NCP to establish a switched subarea link connection.

```
GN112SW  GROUP ACTIVTO=480,        ACTIVITY TIMEOUT
               DIAL=YES,           SWITCHED CONNECTION
               LNCTL=SDLC,         SDLC LINK CONTROL
               PUTYPE=4,           PU TYPE 4 (NCP)

⋮
               BRKCON=NOWNER       BREAK CONNECTION = NO OWNER (ANS)
*
N112L001 LINE  ADDR=008,           LINK ADDRESS
               SPEED=9600,         LINK SPEED
               CLOCKNG=EXT,        EXTERNAL CLOCKING
               SDLCST=(PRIM,SEC),  SDLCST PRIMARY/SECONDARY

⋮
               ISTATUS=ACTIVE      INITIAL VTAM STATUS
*
N112P001 PU    PUTYPE=4,           PU TYPE 4 STATION ADDRESS

⋮
               ISTATUS=ACTIVE      INITIAL VTAM STATUS
```

These switched connections are established using automatic dial and answer. Either subarea can initiate or terminate the switched connection using a VTAM operator command. Use the VARY DIAL operator command to establish a switched connection. The operator at either VTAM subarea can terminate the switched connection using the VARY HANGUP or VARY INACT operator commands.

Note: A VTAM or NCP capable of switched subarea support cannot communicate across a switched subarea link with an adjacent down-level VTAM or NCP that does not contain switched support.

Controlling connections

You should also specify ANS=CONT for the switched link so that the connection is not broken and the session can continue. That is, if the NCP enters automatic network shutdown (ANS), the connection is maintained, and session traffic can continue to flow even though the controlling SSCP has lost contact with the NCP. This operand can also be used with the ACTIVTO and BRKCON operands on NCP definition statements to control breaking the switched connection. The ACTIVTO timer value specifies the idle time on the connection before the switched subarea link is disconnected. The BRKCON operand indicates when to initiate the activity timer for the switched connection. You can either enable the inactivity timer when the connection is established (CONNECTO) or start the timer after the NCP completes automatic network shutdown (NOWNER).

Call security verification provides added security for subarea dial operations by providing you with a two-way password verification system without transmitting actual passwords on the network. Each side of the telephone line (the caller and the receiver subarea) generates random data and sends it to the other side. The other side encrypts the random data, using the password as the encryption key, and sends the encrypted data back to the originator, which verifies the received data against its own encryption. The password is defined using the PRTCT operand on the PU definition statement in the switched major node that describes the caller or receiver subarea. The passwords are not transmitted outside of VTAM even when the caller or the receiver subarea is an NCP.

The level of protection provided by call security verification can be equated with a two-way password verification scheme that requires the transmission of each other's passwords. With the call security verification scheme, the possibility of exposing the passwords to a third party on the line is almost entirely eliminated. However, VTAM does not provide any facility to prevent the passwords from printing in the switched major node definition listings. No software password verification scheme can provide full protection from security exposures. For more security, you should use either session-level password verification or session-level cryptography.

Supporting a remote controller

VTAM also supports the use of a switched connection to a link-attached NCP. A switched link can be used to load, activate, or dump the NCP using normal operator commands. No VTAM definition changes are required, but because NCP does not support a password and encryption process, no call security verification is supported. You cannot code the PRTCT password if this support is used to load and activate an NCP over a switched link because a VTAM is required at the other link station to perform the encryption process.

For example, in Figure 32 on page 105, if VTAM2 fails, VTAM1 can reload and activate NCP2 using a switched connection. However, you cannot use call security verification because VTAM2 is not available to perform the encryption process. Therefore, you cannot code the PRTCT password operand on the switched definition for the PU type 4 connection; if you do, the contact process over the switched link fails.

Prior to communication, however, information must be exchanged to determine whether the connection can be established. VTAM uses the IDNUM value coded on the PU definition statement to determine if the connection is to a defined NCP subarea node. The IDNUM specification must match the IDNUM specification coded in the NCP on the SWMMT definition statement. For more information on this type of connection, refer to “Enhanced remote controller support” on page 129.

X.25 subarea connection (NPSI support)

An X.25 packet switching data network (PSDN) can be used to connect two NCP subarea nodes. Support for an X.25 subarea connection requires the X.25 NCP Packet Switching Interface (NPSI) licensed program. Both permanent virtual circuit (PVC) and switched virtual circuit (SVC) connectivity options are supported.

When the X.25 NCP Packet Switching Interface (NPSI) receives a virtual call request for a switched connection, VTAM enables X.25 NPSI to identify the caller using call security verification (DTE ID verification). The DTE ID carried on the incoming call request packet is checked against the DTE IDs defined on the VERID operand of the PATH definition statement in the switched major node. This

verification applies to X.25 subarea node connections. For details on X.25 NPSI system generation, see *X.25 Network Control Program Packet Switching Interface Planning and Installation*.

Token-ring connection (NTRI support)

Two VTAM subarea nodes can be connected using a local area network (LAN) token-ring connection between two NCPs. The NCP/Token Ring Interconnection (NTRI) is used as the LAN connection and is defined to VTAM as a nonswitched SDLC link. Figure 33 illustrates NCP-to-NCP token-ring connection.

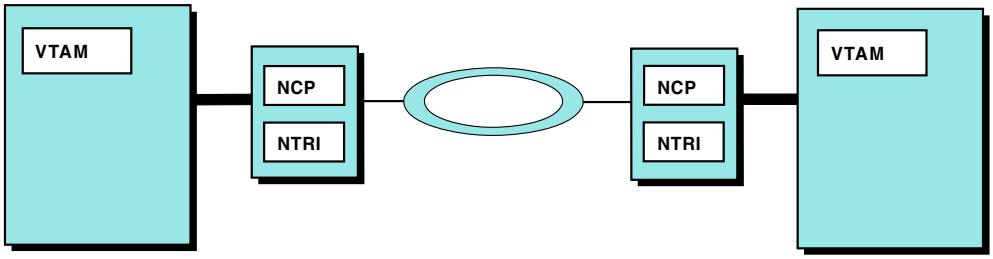


Figure 33. Attaching domains using NCP-to-NCP token-ring connection

Sample NCP definition statements for token-ring connection

The following sample shows some of the NCP definition statements that are used in one NCP to define a token-ring connection between two NCP subarea nodes. Define both the physical connection and the logical connection as nonswitched links. Therefore, in contrast to a peripheral node connection, you do not define a VTAM switched major node for a subarea node token-ring connection.

```
N112TGP1 GROUP ECLTYPE=(PHYSICAL,SUBAREA)  TOKEN-RING PHYSICAL GROUP
*
N112TL01 LINE  ADDRESS=(1088,FULL),        TIC ADDRESS
               LOCADD=400000003023,        LAN ADAPTER ADDRESS
               MAXTSL=1108,                TRANSMIT MAX DATA (MAX)
               PORTADD=1,                  PHYSICAL PORT ADDRESS

⋮
               RCVBUFC=4095                RECEIVE MAX DATA
*
N112TP01 PU
*
N112TLU1 LU
*
N112TGL1 GROUP ECLTYPE=(LOGICAL,SUBAREA),  TOKEN-RING LOGICAL GROUP
               PHYPORT=1,                  ASSOCIATE WITH PHYSICAL PORT
               SDLCST=(PRIM,SEC)           SDLCST LINK CHARACTERISTICS
*
N112TR01 LINE
*
N112TP01 PU    ADDR=04400000000030         TOKEN-RING ADDRESS
```

You can also specify that a token-ring logical group is for a frame-relay connection by coding ECLTYPE=(LOGICAL,FRELAY). This provides a private frame-relay network over the token-ring connections without configuring additional physical lines.

Connecting two VTAMs using an external communication adapter

A VTAM attached through an external communication adapter (XCA), such as an IBM 3172 Nways Interconnect Controller or an IBM Open Systems Adapter, can communicate to other SNA domains through an Ethernet or Ethernet-type local area network (LAN), token ring, or FDDI LAN. You define an XCA LAN connection to VTAM using the external communication adapter (XCA) major node.

For each LAN connected through an XCA, code a VBUILD TYPE=XCA definition statement. The PORT statement identifies the connection between VTAM and the LAN, so you can code only one PORT definition statement for each major node.

A VTAM attached to a LAN through an XCA is defined to its peer processor by the LINE and PU definition statements. These statements must be coded as a nonswitched group using the GROUP definition statement.

If you have multiple VTAM hosts on a local area network attached through one XCA, you can specify which host to contact by using the service access point address (SAPADDR operand) and medium access control address (MACADDR operand) to specify the host. Each host should have a different combination of the SAPADDR operand and MACADDR operand.

A VTAM host connected through an XCA cannot load an NCP across a local area network.

Note: ATM networks accessed through LAN emulation appear to VTAM to be Ethernets or Ethernet-type LANs or token-ring networks and are defined to VTAM as such. ATM networks can be accessed through native ATM in APPN networks. ATM networks accessed through native ATM are defined to VTAM differently from those accessed through LAN emulation. See "ATM native connections" on page 69 for information about defining ATM native connections.

Sample configuration with Ethernet or Ethernet-type LAN

Figure 34 on page 113 shows a multiple domain XCA configuration with an Ethernet or Ethernet-type LAN and two IBM 3172 Nways Interconnect Controllers.

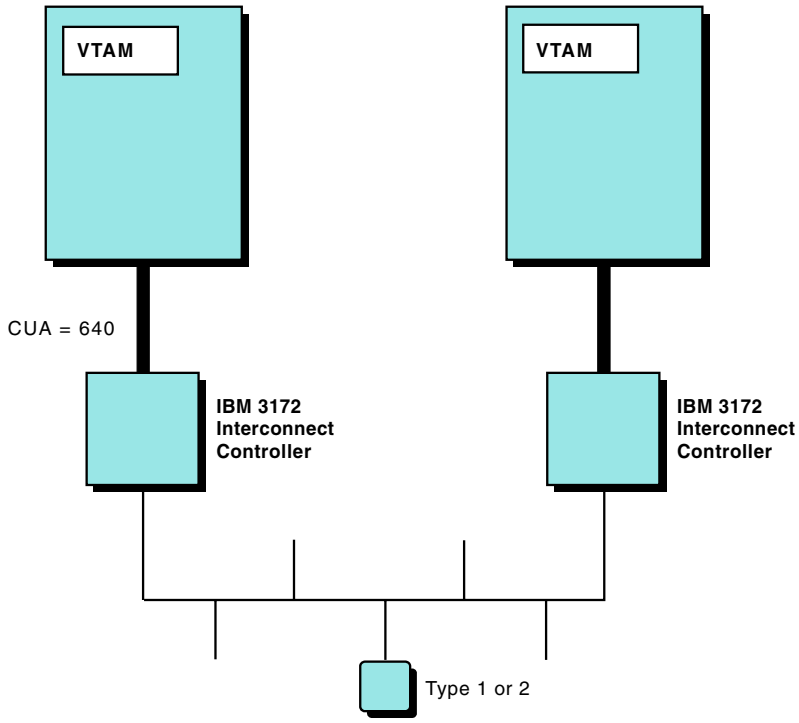


Figure 34. XCA multiple domain configuration with Ethernet or Ethernet-type LAN

To define the configuration in Figure 34, define the following in the first VTAM:

- One external communication adapter major node to represent the physical unit in the IBM 3172 Nways Interconnect Controller
- One external communication adapter major node for each LAN connected to the IBM 3172 Nways Interconnect Controller
- Major nodes for any devices connected to the LAN

Following is the sample code for the configuration in Figure 34:

1. Define an external communication adapter major node to represent the physical unit in the IBM 3172 Nways Interconnect Controller:

```
XCACON1  VBUILD TYPE=XCA               XCA MAJOR NODE
PORT1    PORT  MEDIUM=BOXMGR,         3172 MANAGER
               CUADDR=C04              CHANNEL UNIT ADDRESS
*
GROUP1   GROUP ISTATUS=ACTIVE          ACTIVATED AT GEN
LINE1    LINE
PU1      PU
```

2. Define another external communication adapter major node for the Ethernet or Ethernet-type LAN:

```
XCACON2  VBUILD TYPE=XCA               XCA MAJOR NODE
PORT2    PORT  MEDIUM=CSMACD,          Ethernet or Ethernet-type LAN
               SAPADDR=4,              SERVICE ACCESS POINT ADDRESS
               ADAPNO=1,               ADAPTER NUMBER
               CUADDR=640              CHANNEL UNIT ADDRESS
*
GROUP2A  GROUP DIAL=YES,               SWITCHED PERIPHERAL NODE
               ANSWER=ON,              DIAL WILL ACCEPT CALLS
               CALL=INOUT,             DIAL IN AND OUT
               ISTATUS=ACTIVE          ACTIVATED AT GEN
LINE2A   LINE  ANSWER=ON               PU CAN DIAL IN
PU2A     PU
*
```

```
GROUP2B  GROUP DIAL=NO                SUBAREA
               ISTATUS=ACTIVE         ACTIVATED AT GEN
LINE2B   LINE
PU2B     PU    MACADDR=02608C1C0A21   LAN MEDIUM ACCESS CONTROL ADDRESS
               PUTYPE=5,              VTAM HOST
               SAPADDR=4,             SERVICE ACCESS POINT ADDRESS
               SUBAREA=2,             HOST'S SUBAREA NUMBER
               TGN=2                  TRANSMISSION GROUP NUMBER
```

3. Define a switched major node for the peripheral node attached to the LAN:

```
SWNODE1  VBUILD TYPE=SWNET
PU2      PU    DISCNT=YES,             DISCONNECT FACILITY
               IDBLK=002,              BLOCK IDENTIFICATION
               IDNUM=00002,            IDENTIFICATION NUMBER
               PUTYPE=2                PHYSICAL UNIT TYPE
*
LANPATH2 PATH  DIALNO=010402608C1C0B35, TIC, SAP, MACADDR
               GRPNM=GROUP2A           LOGICAL GROUP DEFINITION
LU2      LU    LOCADDR=2,              DEPENDENT LU
               ISTATUS=INACTIVE        INITIAL STATUS INACTIVE
```

Sample configuration with a token-ring local area network

Figure 35 shows a multiple domain XCA configuration.

To be in session across the XCA connection, application programs in each VTAM must be compatible application programs (that is, they must both be SNA applications programs).

To define the configuration in Figure 35, define the following in the VTAM connected through the IBM 3172 Nways Interconnect Controller:

- One external communication adapter major node to represent the physical unit in the IBM 3172 Nways Interconnect Controller. This physical unit is used for network management purposes. If you are running the NetView program, it is recommended that you code this definition statement. By defining this physical unit, you can have the IBM 3172 Nways Interconnect Controller forward the same type of information that a 3174 cluster controller forwards. For example, any alerts detected by the IBM 3172 Nways Interconnect Controller would be forwarded.
- One external communication adapter major node for each LAN connected to the IBM 3172 Nways Interconnect Controller.
- Major nodes for any devices connected to the LAN.

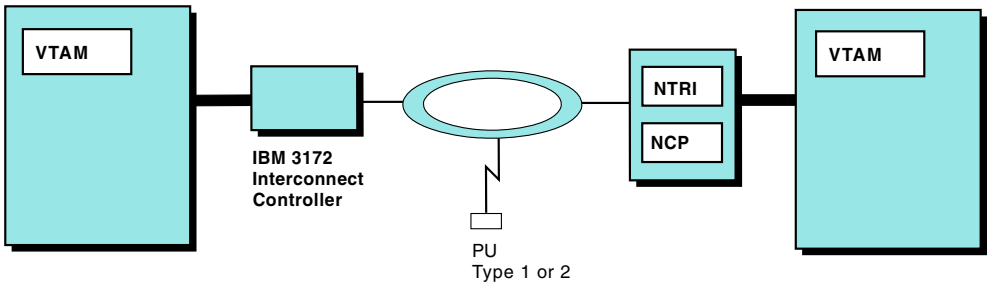


Figure 35. XCA multiple domain configuration

Following is an example of coding used for the configuration in Figure 35 (VTAM attached through the 3172 Nways Interconnect Controller):

1. Define an external communication adapter major node to represent the physical unit in the IBM 3172 Nways Interconnect Controller:

```
XCACON1  VBUILD TYPE=XCA               XCA MAJOR NODE
PORT1    PORT  MEDIUM=BOXMGR,          3172 MANAGER
               CUADDR=500              CHANNEL UNIT ADDRESS
*
GROUP1   GROUP ISTATUS=ACTIVE          ACTIVATED AT GEN
LINE1    LINE
PU1      PU
```

2. Define another external communication adapter major node for the token ring:

```
XCACON2  VBUILD TYPE=XCA               XCA MAJOR NODE
PORT2    PORT  MEDIUM=RING,            TOKEN-RING
               SAPADDR=4,              SERVICE ACCESS POINT ADDRESS
               ADAPNO=1,               ADAPTER NUMBER
               CUADDR=BC0              CHANNEL UNIT ADDRESS
*
GROUP2A  GROUP DIAL=YES,               SWITCHED PERIPHERAL NODE
               ANSWER=ON,              DIAL WILL ACCEPT CALLS
               CALL=INOUT,             DIAL IN AND OUT
               ISTATUS=ACTIVE          ACTIVATED AT GEN
LINE2A   LINE  ANSWER=ON               PU CAN DIAL IN
PU2A     PU
*
GROUP2B  GROUP DIAL=NO                 NCP SUBAREA
               ISTATUS=ACTIVE          ACTIVATED AT GEN
LINE2B   LINE
PU2B     PU    MACADDR=400007777757,   LAN MEDIUM ACCESS CONTROL ADDRESS
               PUTYPE=4,               NCP PU
               SAPADDR=8,              SERVICE ACCESS POINT ADDRESS
               SUBAREA=22,             NCP'S SUBAREA NUMBER
               TGN=1                   TRANSMISSION GROUP NUMBER
```

3. Define a switched major node for the peripheral node attached to the LAN:

```
SWNODE1  VBUILD TYPE=SWNET
PU2      PU    DISCNT=YES,             DISCONNECT FACILITY
               DYNLU=YES,              DYNAMIC DEFINITION OF ILU (CDRSC)
               IDBLK=002,              BLOCK IDENTIFICATION
               IDNUM=000002            IDENTIFICATION NUMBER
               PUTYPE=2                PHYSICAL UNIT TYPE
*
LANPATH2 PATH  DIALNO=0104000069010C59, TIC, SAP, MACADDR
               GRPNM=GROUP2A           LOGICAL GROUP DEFINITION
LU2      LU    LOCADDR=2,              DEPENDENT LU
               ISTATUS=INACTIVE        INITIAL STATUS INACTIVE
```

Notes:

1. You need to code one LINE and PU definition statement in the dial GROUP definition statement for every simultaneous peripheral node connection.
2. Switched devices connected to a LAN are defined to VTAM using the model major node. See Chapter 9, “Defining resources dynamically,” on page 227.
3. Multiple VTAM hosts can be in session with different stations on the LAN through one IBM 3172 Nways Interconnect Controller. To implement this support, code a different SAPADDR and MACADDR value combination to distinguish between the two hosts.

Connecting an NCP to VTAM

An NCP major node, which is a type 4 subarea node, is defined by filing NCP definition statements in the VTAM definition library. The name under which the major node is filed must match the NCP load module name on the NEWNAME operand of the BUILD definition statement.

Additional information required by VTAM to communicate with the NCP and its attached devices is provided to VTAM by coding VTAM-only definition statements and VTAM-only operands on NCP definition statements. The VTAM-only definition statements and operands must appear in the NCP definition statements that define this NCP to VTAM, even though they provide no information to the NCP. If VTAM-only operands are changed or replaced, no new NCP generation is required.

Channel-attached NCP

You can either statically or dynamically define a channel-attached NCP. See "Dynamic configuration of channel-attached devices" on page 230 for information on implementing dynamic definitions.

To define your channel-attached NCP as a type 2.1 device, and define the logical unit either dynamically or statically, see "Defining type 2.1 peripheral nodes" on page 253.

To define a channel-attached NCP:

1. Code a PCCU definition statement.
   - The CUADDR operand on the PCCU definition statement specifies the channel address of the channel adapter on the NCP that is used to perform the load operation. The operator can also specify the channel unit address on the VARY ACT operator command. VTAM identifies the channel link and link station by using the channel unit address.
   - The LOADSTA and DUMPSTA operands on the PCCU definition statement can also be used to identify to VTAM the channel-link station to be used to load and dump the NCP.

   Following is a sample PCCU definition statement for the channel-attached NCP in Figure 36 on page 117:

```
PCCU116  PCCU  CUADDR=016,          3720 IPL ADDRESS
               DUMPDS=NCPDUMP,     NCP DUMP DATA SET DDNAME
               MDUMPDS=MOSSDUMP,   MOSS DUMP DATA SET DDNAME
               CDUMPDS=CSPDUMP,    CSP  DUMP DATA SET DDNAME
               VFYLM=YES,          NOTIFY OPERATOR IF WRONG LOAD MODULE
               VFYC=YES,           VERIFY LOAD MODULE AND RRT MATCH
⋮
               SUBAREA=116         OWNING SSCP SUBAREA
```

   In Figure 36 on page 117, VTAM uses the CUADDR operand from the PCCU definition statement (016) to define the channel link (016-L) and channel link station (016-S). VTAM will load the 3720 using channel link 016-L.

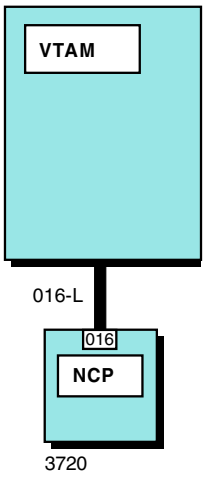


Figure 36. Channel-attached 3720 Communication Controller

VTAM chooses default link stations that are adjacent to the NCP and ready for the dump operation. Preference is given to channel-link stations.

If a channel-attachment major node and its link and link station minor nodes are active when this NCP is activated, use the channel-link station name that is defined by the PU definition statement in the channel-attachment major node for the DUMPSTA, LOADSTA, and RNAME operands of PCCU.

2. Code a BUILD definition statement.

   Following is an example of the operands on the BUILD definition statement:

```
BUILD    BUILD BFRS=240,                        SIZE OF NCP BUFFERS
⋮
               CA=(TYPE5,TYPE5),                CA TYPES
               DELAY=(.2,.2),                   CA ATTENTION DELAYS
               NCPCA=(ACTIVE,ACTIVE),           CA STATUS
               TIMEOUT=(30,30)                  CA ANS – HOST FAILURE
```

   One of the channel links is used to load the NCP.

3. You can also define channel links using the GROUP, LINE, and PU definition statements of the NCP for a 3745 and optionally for a 3725 and 3720. The information about the channel-link characteristics can be coded as operands on these definition statements, similar to those used on the BUILD definition statement (CA, NCPCA, DELAY, and TIMEOUT). Code a LINE and a PU definition statement for each channel adapter. The names used to identify the channel link and link station to VTAM are the names specified on the LINE and PU definition statements, respectively.

   Following is an example of how channel links can be defined using the GROUP, LINE, and PU definition statements of the NCP:

```
N111CA   GROUP LNCTL=CA             CHANNEL ADAPTER DEFINITIONS
L5CA05   LINE  ADDRESS=00,          CHANNEL ADAPTER 5 RELATIVE ADDRESS
               CASDL=125,           CHAN ADAPTER SLOWDOWN DELAY
               NPACOLL=YES,         NPA COLLECTION
⋮
               CA=TYPE6,            CHANNEL ADAPTER TYPE
               DELAY=.1,            CHANNEL ATTENTION DELAY
               NCPCA=ACTIVE,        CHANNEL ADAPTER STATUS
               TIMEOUT=180,         CHANNEL ADAPTER ANS - HOST FAILURE
               TRANSFR=20,          MAXIMUM DATA TRANSFER SIZE
               ISTATUS=ACTIVE       VTAM STATUS
```

```
*
P5CA05   PU    PUTYPE=5,            VTAM HOST CONNECTION - PUTYPE5
               TGN=1                channel link TRANSMISSION GROUP NUMBER
```

Using parallel transmission groups between VTAM and a channel-attached NCP

A transmission group is one or more physical links connecting two subareas. These physical links are viewed as a single logical connection between the two subareas.

VTAM, with NCP Version 5 Release 3 or higher for the 3745 and 3720, allows you to have multiple transmission groups between VTAM and a channel-attached NCP subarea node. You can add up to 16 transmission groups using up to 16 channel links between the same VTAM and NCP subarea nodes. However, because only single-link transmission groups are supported, several channel links between VTAM and NCP cannot be defined as a multilink transmission group.

To define a parallel transmission group:

1. Code transmission group numbers (TGN1, TGN3, TGN255) on the PU definition statement for each channel-link station.

   In the following LINE and PU definition statements for the NCP, each channel link is defined as a separate transmission group (TGN1, TGN3, TGN255) by coding the TGN operand on the PU definition statement. However, the transmission groups TG7 and TG8 between the two NCPs can consist of multiple SDLC links.

   ```
   N111CA   GROUP LNCTL=CA            CHANNEL ADAPTER DEFINITIONS
   L5CA05   LINE ADDRESS=00,          CHANNEL ADAPTER 5 RELATIVE ADDRESS
   ⋮
                CA=TYPE6,             CHANNEL ADAPTER TYPE
                ISTATUS=ACTIVE        VTAM STATUS
   *
   P5CA05   PU   PUTYPE=5,            VTAM HOST CONNECTION - PUTYPE5
                 TGN=1                CHANNEL-LINK TRANSMISSION GROUP NUMBER
   *
   L6CA06   LINE ADDRESS=01,          CHANNEL ADAPTER 6 RELATIVE ADDRESS
   ⋮
                CA=TYPE6,             CHANNEL ADAPTER TYPE
                ISTATUS=ACTIVE        VTAM STATUS
   *
   P6CA06   PU   PUTYPE=5,            VTAM HOST CONNECTION - PUTYPE5
                 TGN=3                CHANNEL-LINK TRANSMISSION GROUP NUMBER
   *
   L7CA07   LINE ADDRESS=02,          CHANNEL ADAPTER 7 RELATIVE ADDRESS
   ⋮
                CA=TYPE6,             CHANNEL ADAPTER TYPE
                ISTATUS=ACTIVE        VTAM STATUS
   *
   P7CA07   PU   PUTYPE=5,            VTAM HOST CONNECTION - PUTYPE5
                 TGN=255              CHANNEL-LINK TRANSMISSION GROUP NUMBER
   ```

2. Code a transmission group number on the PCCU definition statement in the NCP major node or on the PU definition statement in a VTAM channel-attachment major node.

   If a TG number is specified on the VTAM definition statement (PCCU or PU), it must match the corresponding TG number specification on the PU definition statement in the NCP for that channel-link station. Otherwise, code TGN=ANY, which allows the TG number to be assigned by the adjacent NCP, but the NCP subarea node must then specify a TG number. The transmission group number

used is the TGN value that is coded on the PU definition statement in the NCP and the TG number that VTAM receives in the exchange identifier (XID) exchange during activation of the channel link.

Transmission group number on the PCCU definition statement
One of the channel attachments must be used to load and activate the NCP from the VTAM host. The transmission group number that is used for that channel attachment can be specified on the PCCU definition statement.

Transmission group number on the PU definition statement
The channel links and link stations between VTAM and a channel-attached NCP must be defined in a channel-attachment major node.

3. Code PATH definition statements to define different routes using different channel links between the same VTAM and NCP subarea nodes. Although you can define up to 16 explicit routes between VTAM and a channel-attached adjacent NCP, the routing capability between these two subareas can be limited by the number of virtual routes that can be defined. The maximum is eight virtual routes using three different transmission priorities for a total of 24 potential routes. For information about how this affects network routing, see "Network routing for subarea nodes" on page 348.

Following are two examples of how parallel transmission groups are defined:

- Figure 37 illustrates a possible configuration of parallel transmission groups.

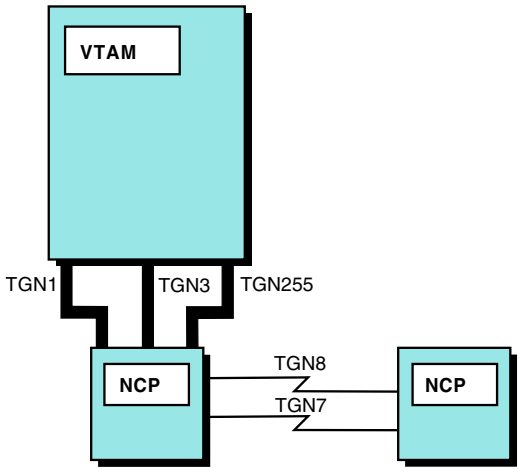


Figure 37. Parallel transmission groups between VTAM and NCP

In Figure 37, the channel-link station TGN1 can be used to load and activate the channel-attached NCP. In this case, the PCCU definition statement is used to specify the TG number (default is 1). The other two channel-link stations, TGN3 and TGN255, can be defined using a channel-attachment major node for each. In these two cases, the PU definition statement for each of the channel-link stations specifies the transmission group number. Because VTAM does not support the capability of defining multiple channel links as a single transmission group, each channel link in the NCP must be defined as a separate transmission group.

- The following is a PCCU definition statement:

```
PCCU116  PCCU  AUTODMP=NO,         AUTOMATIC DUMP OPTION
               CUADDR=016,         3745 IPL ADDRESS
```

```
⋮
                TGN=ANY,            USE TGN OF NCP FROM XID
⋮
                SUBAREA=116         OWNING SSCP SUBAREA
```

In this example, the transmission group specification is TGN=ANY. Therefore, when contacted over the channel link identified by the CUADDR operand, VTAM uses the TG number that is specified on the PU definition statement for the channel-link station in the NCP. Because there can be multiple channels between VTAM and an NCP, each channel attachment must use a different TG number so that the PATH statements between the subarea nodes can identify different network routes. Other channel attachments to the same NCP can be defined using the channel-attachment major node specification. In this case, the TG number specification is not on the PCCU definition statement but on the PU definition statement in the channel-attachment major node that represents that channel-link station.

Channel-attached NCP to a VTAM data host

You can define a channel attachment between a host and an NCP in a channel-attachment major node. This data host attachment eliminates the requirement for the host to have ownership (control) of network resources. It provides the ability for VTAM to contact a channel-attached NCP without activating the NCP. By using this definition process, a VTAM data host is not required to establish an SSCP-PU session for data transfer to occur over the channel link to the communication controller. This method of attaching to the network prevents the data host from issuing requests on the SSCP-PU session that can affect the configuration of the local NCP major node or other nodes.

The only definition required in the VTAM data host is a channel-attachment major node definition; no NCP load module, NCP resource resolution table (RRT), or copy of the NCP major node definition is needed. However, the NCP must be loaded and active if VTAM is to contact the NCP and not establish an SSCP-PU session.

The GROUP definition statement defines channel-to-NCP links. It also carries other operands that sift down to following definition statements that do not explicitly override them. When you code LNCTL=NCP, VTAM does not have to have knowledge of the resources attached to the NCP. Code one LINE definition statement for each channel-to-NCP link and one PU definition statement for each LINE definition statement. The LINE definition statement defines the following to VTAM:

- The name of the link
- The channel unit address of the link
- Certain procedural options to be used for this line

In Figure 38 on page 121, the two channel connections between NCP2 and VTAM2 can be defined using a channel-attachment major node. In this case, VTAM1 is a host that owns all of the network resources attached to both NCP1 and NCP2. VTAM1 loads the NCPs and activates all of the resources.

VTAM2 in this configuration is designated as the data host and thus does not own any network resources. The primary task of the data host (VTAM2) is to support application programs such as IMS™, CICS, or TSO. All LU-LU sessions are therefore cross-domain sessions.

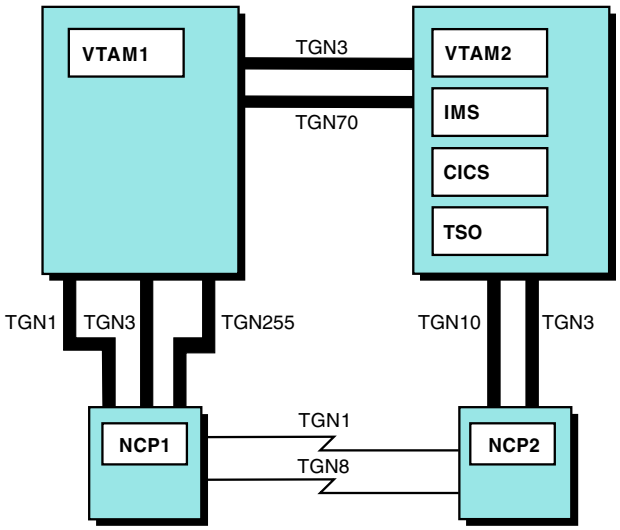


Figure 38. Example NCP to VTAM data host connection

The following example illustrates the definition statements used in the VTAM data host (VTAM2) to define one of the channel links between VTAM2 and NCP2. Another LINE definition statement and PU definition statement are required to define the other channel link (TG3) between VTAM2 and NCP2.

The LINE definition statement defines the channel unit address for the channel link to the communication controller. The MAXBFRU operand defines the maximum number of I/O buffers that are used by the VTAM host to receive data from the NCP. This operand has the same meaning as the MAXBFRU operand that is specified on the HOST definition statement in the NCP and should contain the same value.

The value of the ISTATUS operand on the LINE definition statement is INACTIVE. If ISTATUS=ACTIVE is specified, the activation process causes VTAM to issue the IST380I message with sense information (081C0010) that is unnecessary. By specifying INACTIVE as the status, the major node activation process completes, and the message is eliminated.

The MAXDATA operand on the PU definition statement also has the same meaning as the MAXDATA operand on the PCCU definition statement. The operand specifies the maximum number of bytes that VTAM can send in a single-segment PIU. The TGN operand identifies the unique transmission group number (10) that is associated with this channel-link connection.

```
CTSNCP2  VBUILD TYPE=CA              NCP CHANNEL LINK CONNECTION
*
CTSNCP2G GROUP  LNCTL=NCP            NCP CONNECTION
                CHANCON=COND,        CONTACT CONDITIONAL — SUBAREA IN USE

⋮
                ISTATUS=ACTIVE        VTAM INITIAL STATUS
*
N02CA    LINE   ADDRESS=32A,          CHANNEL UNIT ADDRESS
                MAXBFRU=32,          MAXIMUM VTAM I/O BUFFERS FOR RECEIVE

⋮
                ISTATUS=INACTIVE      VTAM INITIAL STATUS
*
N02TG10P PU     PUTYPE=4,            PHYSICAL UNIT TYPE = NCP
```

```
                MAXDATA=4096,         MAXDATA TRANSFER
                TGN=10,               TRANSMISSION GROUP NUMBER

⋮

                ISTATUS=ACTIVE        VTAM INITIAL STATUS
```

In a multiple domain environment, parallel transmission groups can be defined between VTAM and NCP similar to those in a single domain environment. This capability allows you to add multiple channel attachments to an NCP subarea. In Figure 32 on page 105, the channel connections between VTAM2 and NCP2 can be defined using channel-attachment major node definitions. In this case, both TG10 and TG3 can be used to communicate with NCP2 without activating NCP1. The transmission group number is specified on the PU definition statement in the channel-attachment major node. Each transmission group is limited to a single link TG. Although you can have as many as 255 transmission groups, only 16 of these can be defined due to the following limitations:

- The number of physical attachments (16 channel adapters) available on a 3745
- The number of explicit routes (16) that can be defined between VTAM and a channel-attached adjacent NCP

See “Network routing for subarea nodes” on page 348 for information about how this affects network routing.

Note: NCP2 can be activated by the data host if ownership of the NCP2 resources is required as a part of a backup and recovery scenario. This establishes the necessary SSCP-PU session, but it requires the data host to maintain a copy of the NCP major node in its VTAMLST and the NCP resource resolution table (RRT). You could require a copy of the NCP load module at the data host if the NCP needs to be reloaded and the communication controller does not have a disk (the IBM 3725, for example).

NCP definition statements

The definition statements contained in the NCP major node contain information that can be divided into two categories:

- Information used by VTAM
- Information used by the NCP

NCP definition statements for VTAM

The PCCU, BUILD, and HOST definition statements in the NCP major node can contain information for VTAM.

PCCU definition statement

The programmed communication control unit (PCCU) definition statement identifies the communication controller where the NCP is loaded. It is required in VTAM systems and defines the VTAM functions that are provided for this NCP. Code at least one PCCU definition statement for each NCP.

Sample PCCU definition statement

Following is a sample PCCU definition statement:

```
PCCU116  PCCU  AUTODMP=NO,           AUTOMATIC DUMP OPTION
               AUTOIPL=NO,           AUTOMATIC RE-IPL OPTION
               AUTOSYN=YES,          SYNCHRONIZE WITH ACTIVE NCP
               CUADDR=016,           3745 IPL ADDRESS
               LOADFROM=EXTERNAL,    LOAD 3745 FROM 3745 DISK
               DUMPLD=YES,           DUMPLOAD USING 3745/3720 DISK
               DUMPDS=NCPDUMP,       NCP DUMP DATA SET DDNAME
               MDUMPDS=MOSSDUMP,     MOSS DUMP DATA SET DDNAME
```

```
                    CDUMPDS=CSPDUMP,       CSP  DUMP DATA SET DDNAME
                    MAXDATA=5000,          MAXIMUM PIU SIZE
                    NETID=USIBMCSS,        NETWORK IDENTIFIER
                    VFYLM=YES,             NOTIFY OPERATOR IF WRONG LOAD MODULE
                    VFYC=YES,              VERIFY LOAD MODULE AND RRT MATCH
                    .
                    .
                    .
                    SUBAREA=116            OWNING SSCP SUBAREA
```

SUBAREA operand

The SUBAREA operand associates the communication controller with a VTAM host when the NCP is activated. If each host provides the same function for the NCP, only one PCCU definition statement is necessary, but the operator might need to override unique attachment requirements.

Multiple PCCU definition statements

If you code multiple PCCU definition statements, VTAM compares its network identifier and subarea number to the values of the NETID and SUBAREA operands on all the PCCU definition statements to locate the one that applies to it. However, VTAM selects this PCCU definition statement in the following order:

1. The VTAM host processes the first PCCU definition statement whose NETID and SUBAREA operands match those in VTAM.
2. If no such PCCU definition statement exists, the VTAM host processes the first one that has the same NETID as VTAM and that does not have a SUBAREA.
3. If a PCCU definition statement as described in step 1 or 2 does not exist, the VTAM host processes the first PCCU definition statement without a NETID operand but with a SUBAREA value equal to that found in VTAM.
4. Finally, if a PCCU definition statement as described in step 1, 2, or 3 does not exist, the VTAM host processes the first PCCU statement for which neither the NETID nor SUBAREA operand is coded.

The NCP for an IBM 3745 or 3720 Communication Controller can be loaded from either of the following:

- The NCP load library at the host (LOADFROM=HOST)
- The 3745 or 3720 disk (LOADFROM=EXTERNAL)

When the VARY ACT command is issued to activate NCP, the LOADFROM operand on the PCCU definition statement is used if the LOADFROM operand is not specified on the VARY ACT command.

Automatic dump and reload

The AUTODMP and AUTOIPL operands on the PCCU definition statement inform VTAM whether to automatically dump and reload the NCP if an error is encountered. To have an IBM 3745 or 3720 Communication Controller dump and reload the NCP from its disk, do one of the following:

- Code the DUMPLD=YES operand on the PCCU definition statement.
- Specify the DUMPLOAD=YES operand on the VARY ACT command.

These operands are then ignored if the operation is successful. The NCP dump can then be transmitted to the VTAM host from the communication controller disk at a later time.

If the communication controller is a 3725 or the dump is not stored on the communication controller disk, it is transmitted to the VTAM host and

stored in the file identified by the DUMPDS operand on the PCCU definition statement. If you specify the CDUMPDS and MDUMPDS operands on the PCCU definition statement, they identify the files in which a CSP and a MOSS dump, respectively, are to be stored in the VTAM host. If you do not specify these operands, the corresponding dump is stored in the file identified by the DUMPDS operand.

Verifying the load module

When you load the NCP, you can use the AUTOSYN operand of the PCCU definition statement to request VTAM to compare the name of the load module currently in the communication controller to the NEWNAME operand on the BUILD definition statement. If you code AUTOSYN=YES, VTAM uses the current NCP load module; if you code AUTOSYN=NO, the operator must determine whether the communication controller should be reloaded with a new copy of the NCP. If the NCP load module and the NEWNAME operand do not match, the VFYLM operand determines whether the operator is to be informed of the mismatch (VFYLM=YES) or whether the operator must simply reload the communication controller with the different load module (VFYLM=NO).

For a channel-attached NCP, the CUADDR operand of the PCCU definition statement indicates the channel unit address of the communication controller that VTAM should use to load the NCP.

Verifying time stamps

When the NCP/EP definition facility (NDF) is used to generate NCP, a resource resolution table (RRT) is created in addition to the NCP load module. At initialization, VTAM reads the RRT to determine what resources are contained in the NCP load module. VTAM examines the time stamp of the RRT and verifies that it corresponds to the NCP load module associated with the RRT.

The VFYC operand on the PCCU definition statement enables you to specify what action VTAM should take if the time stamps do not match. If you code VFYC=NO or allow it to default and VTAM detects a correlator mismatch, VTAM sends a message to the operator explaining that the NCP is being loaded due to the mismatch. If you specify VFYC=IGNORE, VTAM continues the NCP activation process regardless of the time stamps of the NCP load module and the RRT; no message is issued.

BUILD definition statement

The operands on the BUILD definition statement are primarily used by the NCP generation process. The SUBAREA operand identifies the subarea address of the NCP to VTAM. The subarea address number specified must be within a subarea address range that is acceptable to VTAM; this depends on whether the network contains nonextended network addressing nodes, extended network addressing nodes, or extended subarea addressing nodes.

Sample BUILD definition statement

Following is a sample BUILD definition statement:

```
BUILD    BUILD BFRS=240,          SIZE OF NCP BUFFERS
               MEMSIZE=4M,        3745 MEMORY SIZE
               MODEL=3745-210,    CONTROLLER IS AN IBM 3745
               VERSION=V5R2.1,    NCP VERSION
               USGTIER=4,         USAGE TIER 4

⋮
               LOCALTO=1.5,       (NTRI) LOCAL ACKNOWLEDGEMENT TIMER
               MXRLINE=2,         (NTRI) 2 TICS ATTACHED
               MXVLINE=16,        (NTRI) 16 PUS MAXIMUM
```

```
                REMOTTO=1.5,        (NTRI) REMOTE ACKNOWLEDGEMENT TIMER
:
                NETID=CSSNET,       NETWORK ID
                SUBAREA=111,        NCP SUBAREA NUMBER
                SALIMIT=255,        MAXIMUM SUBAREA ADDRESS LIMIT
                ERLIMIT=8,          USE ONLY 8 EXPLICIT ROUTES
                NEWNAME=N111A2,     NCP LOAD MODULE NAME-NCP 111, MONTH 10, CYCLE 2
                PUNAME=NCP111,      NCP PU NAME
                LOADLIB=NCPLOAD     LOAD MODS IN NCPLOAD LIB
```

Subarea addressing

In the preceding sample, the MAXSUBA operand is not coded, which means that the network does not contain nonextended network addressing nodes. The SALIMIT operand, which is used in an extended subarea addressing node, limits the subarea address range for the sample NCP to 255 subareas. This subarea address range is the same as for an extended network addressing environment. The maximum subarea address range for an attached VTAM can be greater than 255; however, because SALIMIT=255 for the NCP, its subarea address must be less than or equal to that limit.

You can limit the number of explicit routes that can be used in an extended subarea network by coding the ERLIMIT operand on the NCP BUILD definition statement. Although VTAM supports up to 16 explicit routes, if the number of explicit routes in the NCP is limited to eight, VTAM must use explicit route numbers 0 through 7 to communicate with that NCP.

For a single-domain network, the NETID operand on the BUILD definition statement must be the same as the network identifier used to start VTAM. For information on coding the BUILD definition statement, refer to the *NCP, SSP, and EP Resource Definition Reference*.

Identifying the NCP load module

To identify the NCP load module, perform the following steps:

1. Code the LOADLIB operand on the BUILD definition statement to identify the file in which the NCP load module is stored.
2. Code the NEWNAME operand on the BUILD definition statement to identify the name of the load module. This is also the name used to file the NCP in VTAMLST.
3. If multiple copies of an NCP must be maintained at a single host, alter the load module name by changing the value of the NEWNAME operand.
4. You can refer to the NCP using the same physical unit name by coding the same value on the PUNAME operand for each different NCP load module.
5. Activate the NCP by specifying ID=*puname* and LOADMOD=*newname*. If you specify ID=*puname* on the VARY ACT command, a DISPLAY command for the NCP is issued specifying ID=*puname*.

If the NCP is to be automatically activated using a VTAM configuration list (CONFIG start option), the NCP major node name must be the same as the load module name specified on the NEWNAME, because it is filed in VTAMLST using that name. For example, in the preceding BUILD definition statement, you can always refer to the NCP by its physical unit name (PUNAME) of NCP111. The load module name specified on the NEWNAME operand can contain other information, such as the month (October is 10 or X'A') and the cycle number of the NCP generation for that month.

You can also maintain different load modules for the same NCP by filing the NCPs under different temporary names. Before activating one of them as a VTAM major node, rename the member to match the load module name (NEWNAME) on the BUILD definition statement.

HOST definition statement

VTAM examines the HOST definition statement to determine the amount of data that VTAM must be prepared to receive from the NCP over the channel. The value VTAM uses is the MAXBFRU specification times the size of VTAM buffers (IOBUF size). In exchange identification (XID) processing, VTAM informs NCP of the maximum amount of data that it can receive from the NCP. NCP also informs VTAM of the maximum amount of data that it can receive. The value NCP uses in the XID process is the NCP buffer size (BFRS) times the value specified on the TRANSFR operand of either the BUILD or LINE definition statements. VTAM compares this value to the MAXDATA operand on the PCCU definition statement and returns the smaller of the two values.

Sample HOST definition statement

Following is a sample HOST definition statement:

```
VTAM116  HOST  BFRPAD=0,          BUFFER PAD FOR VTAM
               MAXBFRU=50,        INITIAL UNITS FOR INPUT
               SUBAREA=116,       HOST SUBAREA
               UNITSZ=512         BUFFER UNIT SIZE
```

Note: With NCP Version 4 Release 2 and later releases, the UNITSZ specification for an NCP is ignored by VTAM and is no longer used in channel programming. If specified, it should match the size specified for the IOBUF buffer pool start option, but it does not affect the success or failure of link establishment.

Specifying a value for MAXDATA

For NCP Version 4 Release 3 and later releases, the MAXDATA operand is redundant and need not be specified. MAXDATA is set to 65 535 by default. The maximum size of data transfers between VTAM and NCP is established during XID exchange. For NCP releases prior to 4.3, the value of MAXDATA can affect VTAM, NCP, and network performance. The following section provides guidelines for specifying a MAXDATA value.

The MAXDATA operand should be smaller than or equal to the size of the largest PIU the network will handle. Use the following criteria for choosing the value of MAXDATA:

- MAXDATA is related to the BFRS and TRANSFR operands of the BUILD definition statement. MAXDATA should not exceed the product of BFRS and TRANSFR, minus 18 (BFRS x TRANSFR – 18).
- MAXDATA should not exceed the size of the VTAM IOBUF buffer pool.
- MAXDATA should not exceed the product of the MAXBFRU and UNITSZ values for the NCP.
- If a MODIFY LOAD command will be issued for a local or remote NCP, the size of MAXDATA must be at least 2082, which is 2048 plus the size of the PIU header (34).
- Although MAXDATA applies only to the PCCU statement for a channel-attached NCP, its value should not exceed the capacities of any link-attached or cross-domain NCPs. An SNA path error can occur if MAXDATA exceeds the capacity of any NCP along the route taken by a message.

For more information on selecting the values for VTAM, see Chapter 23, "Tuning VTAM for your environment," on page 663.

Network control point definition statements for NCP

The following definition statements contain information that the NCP needs for proper operation.

BUILD
Contains information about NCP characteristics, such as controller storage size, subarea address of the NCP, and characteristics of channel adapters.

SYSCNTRL
Identifies the dynamic control facilities included in the NCP. These facilities allow the NCP to execute requests from VTAM to change certain NCP parameters or to determine the status of resources, such as lines and stations.

HOST Contains characteristics of VTAM. Knowledge of these characteristics is needed to conduct channel operations for communication with VTAM.

LINE, PU, and LU definition statements
Define SDLC resources to NCP. The LINE, CLUSTER, and TERMINAL definition statements define BSC resources to NCP. The PUDRPOOL and LUDRPOOL definition statements define pools of addresses. These addresses are used to support NCP dynamic reconfiguration (both PUDRPOOL and LUDRPOOL) and switched connections to NCP (LUDRPOOL only).

Connecting an NCP to another NCP

A link-attached NCP can use any of the following connections:
- Nonswitched SDLC lines
- Switched SDLC lines
- Token-ring networks
- 3172 Nways interconnect controller

The use of a switched SDLC link provides the same capabilities as a nonswitched link except that a static dump of controller storage cannot be transferred directly to the host. The use of the other connections requires that an initial load module be available on the controller disk. However, after the initial load module is activated by VTAM, subsequent load modules can be transferred over the token-ring or X.25 switched virtual circuit (SVC) connections, using 3720 and 3745 multiple load module support.

Recovery operations

If a functional NCP is not present in the controller, the controller does not accept the transfer of load modules over these connections. You will need to perform a recovery operation using one of the following methods:

- Reload NCP over a switched SDLC link.
- Using a diskette procedure, transfer an initial load module to the controller disk, and then reload the controller.

The capability to load and activate or dump a link-attached NCP over a token-ring or an X.25 switched virtual circuit (SVC) connection requires additional planning.

Defining a link-attached NCP

Figure 39 illustrates a link-attached NCP (NCP2). The connection between a channel-attached and a link-attached NCP is defined in each of the NCP subarea nodes. You define each end of the connection in each NCP using a LINE definition statement for the link and a PU definition statement for the link station.

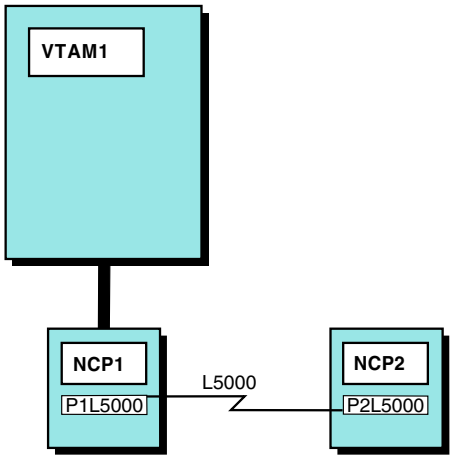


Figure 39. Defining, loading, and activating a link-attached NCP

The following example illustrates the definitions (at the channel-attached NCP, NCP1) for the link and link station when using a nonswitched SDLC link to the link-attached NCP subarea node:

```
*------------------------------------------------------------------------
*  256 KILOBYTE LINK
*------------------------------------------------------------------------
G5INN    GROUP ACTIVTO=120,               ACTIVATE TIME OUT VALUE
               DIAL=NO,                   NONSWITCHED LINK
               LNCTL=SDLC,                SDLC LINE CONTROL
               REPLYTO=(10.0,15.0),       REPLAY TIMEOUT VALUES

⋮
               ISTATUS=ACTIVE             VTAM STATUS
*
L5000    LINE  ADDRESS=(000,FULL),        LINE ADDRESS — FULL DUPLEX
               SPEED=245760,              LINK SPEED
               HISPEED=YES,               HIGH SPEED LINK
               MONLINK=YES,               SDLC MONITOR MODE ON
               IPL=YES,                   ACCEPT IPL REQUEST

⋮
               MODULO=128,                MODULO COUNT = 128
               PAUSE=(1.0,1.0),           PRIMARY AND SECONDARY PAUSE
               RETRIES=(3,12,5),          RETRY VALUES
               SDLCST=(PRIM,SEC),         PRIMARY AND SECONDARY CHARACTERISTICS
               ISTATUS=ACTIVE             VTAM STATUS
*
P1L5000  PU    ANS=CONTINUE,              AUTOMATIC NETWORK SHUTDOWN
               MAXOUT=127,                FRAMES PRIOR TO LINK-LEVEL REPLY

⋮
               PUTYPE=4,                  TYPE 4 — NCP CONNECTION
               TGN=7,                     TRANSMISSION GROUP NUMBER
               ISTATUS=ACTIVE             VTAM STATUS
```

To load and activate the link-attached NCP:

1. Specify the link station name of the channel-attached NCP (in the example, P1L5000) in the RNAME operand of either the PCCU definition statement for the link-attached NCP (NCP2) or the VARY ACT command.

```
On the PCCU Definition Statement for NCP2:

PCCU  SUBAREA=1,
⋮
      LOADSTA=P1L5000,
      RNAME=P1L5000,
⋮
On the VARY ACT Operator Command:

VARY ACT,ID=NCP2,LOAD=YES,RNAME=P1L5000
```

2. Define the characteristics of the link. The NCP subarea nodes exchange information over the link using the XID process. The subarea node with the higher subarea number performs the role of the primary and performs the polling unless this role relationship is predefined. The characteristics associated with the link, depending on primary or secondary role, must be defined using the SDLCST and GROUP definition statements in the NCP subarea nodes.

Cross-subarea links can be directly or indirectly activated and deactivated. For information about using links, see "Activating links" on page 647 and "Deactivating links" on page 653.

Enhanced remote controller support

Enhanced remote controller support enables you to load, activate, and dump NCP using a switched SDLC link, including a token-ring connection and an X.25 switched virtual circuit (SVC). However, the token-ring and X.25 SVC switched connections require an NCP to be operational in the communication controller or available on the controller disk.

Following is a sample switched major node for enhanced remote controller support:

```
NCPSW111 VBUILD TYPE=SWNET               SWITCHED MAJOR NODEE
*
P5001T4  PU    ADDR=C1,                  PU TYPE 4 STATION ADDRESS
               MAXDATA=4096,             MAXIMUM DATA TRANSFER
               NETID=USIBMCSS,           NETWORK IDENTIFIER
               SUBAREA=112,              SUBAREA NUMBER
               IDNUM=11201,              UNIQUE IDNUM
               PUTYPE=4,                 PU TYPE 4

⋮
               TGN=8,                    TRANSMISSION GROUP NUMBER
*
         PATH  DIALNO=1234567,           TELEPHONE DIAL NUMBER
               GID=2,                    GROUP IDENTIFIER
               PID=2,                    PATH IDENTIFIER
               REDIAL=4,                 RETRY DIAL 4 TIMES

⋮
               GRPNM=G6INN               GROUP NAME IN NCP
```

To use enhanced remote controller support:

1. Define the switched physical connection by coding the NCP LINE and PU definition statements with the DIAL operand specified as YES.
2. Define the corresponding logical definition to VTAM by coding a PU definition statement for a type 4 PU in a switched major node.

3. Because a subarea physical unit can have multiple connections, code a connection identifier (IDNUM) to distinguish the multiple PU definition statements that can apply to a subarea physical unit. The value of the IDNUM operand on the PU definition statement must match the value of the IDNUM operand on the SWMMT definition statement for the remote communication controller; when the IDNUM operand is concatenated with the subarea number for the NCP, it must be unique within the network. Therefore, you can use the same IDNUM to represent connection endpoints in different subareas. If IDNUM is duplicated within the network, a connection can be established with the wrong subarea, which may result in a connection failure or other errors later in the process. IDBLK is not used by VTAM and is set to a value of X'FFF' in the station identifier that is used for the XID exchange during the dial procedure.
4. Code a PATH definition statement so that VTAM performs a dial-out operation on the switched link and contacts the remote device (PUTYPE=4). The PATH definition statement also specifies the group name (GRPNM) of the group of switched lines in the NCP over which the connection can be made.
5. Make sure the value of the NETID operand on the PU definition statement in the switched or model major node matches the NETID operand on the SWMMT definition statement for the corresponding switched line in the remote communication controller.

   Note: The IDNUM and NETID operands are maintained within the NCP load module so an NCP generation is required to change these values.
6. For security reasons, you might want to code a password on the PU definition statement using the PRTCT operand. If the check fails, the connection fails.

   Note: In a single-domain network where a single VTAM and NCP control both link stations, VTAM does not support call security verification. (That is, if you code either the MONLINK=YES or MONLINK=CONT operand for the switched line in the link-attached NCP, VTAM does not support call security verification.)

NCP switched SDLC connections

In the following sections about dial-out and dial-in connections, normal dial-in refers to a call originating from a PU; automatic dial-out refers to a call established by VTAM using a communication controller. Manual dial-out refers to a VTAM-operator-assisted call from the communication controller or communication adapter to the PU.

Dial-out connections

When an application program requests a session, VTAM selects a dial-out line, unless the application program specifies the dial-out line. Depending on the capabilities of the switched line, the dial-out call can be handled automatically (done entirely by VTAM or by VTAM and the NCP) or manually (done partly by the VTAM operator).

Dial-out requests can be placed on one of a number of switched paths to a physical unit as defined within a switched major node. The dial-out path assignment for switched physical units can be controlled by making switched paths usable or not usable with the VARY PATH command. This command can be used for individual switched paths by specifying the path identifier (PID), or for a group of switched paths by specifying the group identifier (GID). For example, all switched paths that

use Wide Area Telephone Service (WATS) lines for dial-out connections can be associated with a GID in the switched major node definition, and can be made usable or not usable as a unit.

For a manual dial-out call, VTAM displays a message with the line and telephone number to use. The VTAM operator manually calls the physical unit. VTAM is notified when the switched connection is complete and then verifies the identity of the physical unit. From this point, the process continues as for a dial-in connection.

If the VTAM operator is unable to complete a manual dial-out call, the VARY INOP command can be used to specify whether VTAM is to select an alternate line or terminate the dial-out operation.

Dial-in connections

For normal dial-in operation, the terminal operator calls the communication controller or communication adapter, and VTAM is notified of the incoming call. The NCP then requests that the physical unit identify itself. When the physical unit responds, VTAM locates the proper physical unit definition in an active switched major node. After VTAM completes activation of the physical unit and its subordinate logical units, sessions involving the logical units can be initiated as if the line were nonswitched.

Switched lines with dial-in capability can allow or disallow incoming calls. For an incoming call to be allowed, the line must be in answer mode. The mode setting can be specified in the definition statement for the line and can be changed with the ANS operand of the VARY ACT command when activating the line, or with the VARY ANS command after the line is active.

Chapter 6. Using Enterprise Extender (EE)

Companies continue to rely on older applications residing on mainframes. Initially, access to these applications was through SNA. Today, newer applications are generally based on TCP/IP. Initially, companies supported both these networks separately, but are now seeking ways to consolidate their SNA traffic onto the TCP/IP network. However, it is not economically practical to convert existing SNA applications to TCP/IP and, in many cases, conversion might even be technically impractical because of the lack of source code, adequate skills, or both.

With Enterprise Extender (EE), you can extend the reach of SNA applications and data to include TCP/IP networks and IP-attached clients with levels of reliability, scalability, and control similar to those that SNA users have used. EE integration uses standard IP technology and does not require new hardware or new software in the IP backbone.

You can use an IP network for SNA sessions with Enterprise Extender, which provides enablement of IP applications and convergence on a single network transport while preserving SNA application and endpoint investment. An EE connection is a logical connection that represents IP connectivity from a host to a specified IP address or host name. Conceptually, an IP network looks like an APPN/HPR transmission group (TG) in a session route.

In summary, EE is an extension of High Performance Routing (HPR) technology that provides encapsulation of SNA application traffic within UDP frames by HPR-capable devices at the edges of an IP network. To the IP network, the SNA traffic is UDP datagrams that get routed without hardware or software changes to the IP backbone. To the user, the session is normal SNA with predictable performance and high availability. By wrapping the SNA application in this way, EE enables SNA data to be carried over an IP backbone without changing either the SNA applications or the IP hardware.

This information is organized into the following sections:

- Overview
- Designing the EE network
- Configuring the EE network
- Configuring the EE connection network
- EE security considerations
- Tuning the EE network
- Advanced EE considerations
- Troubleshooting EE problems

In addition, this topic contains information about various topics including:

- EE connection network reachability awareness
- Benefits of defining multiple Enterprise Extender virtual routing nodes
- Parallel Enterprise Extender connections
- TCP/IP MTU size for Enterprise Extender
- Running EE in constrained or virtualized environments
- RTP transmission stall operator awareness and recovery support

© Copyright IBM Corp. 1984, 2011

- Load balancing with Enterprise Extender
- Transmission group profiles for Enterprise Extender

For more information about Enterprise Extender, see *Migrating Subarea Networks to an IP Infrastructure*, SG24-5957-00 (a member of the IBM Redbooks library) http://www.redbooks.ibm.com/abstracts/sg245957.html?Open

Overview

This section provides information about the following:

- Benefits of EE
- Availability of EE
- Hardware requirements
- EE reliability and strategy
- Using EE and extended border node (EBN) as a replacement for SNI
- EE implementation considerations

Benefits of Enterprise Extender

EE provides the following benefits:

- SNA transport over a native IP network with no changes required to SNA applications
- SNA application connectivity using an IP backbone that supports preservation of SNA transmission priority
- When used with an extended border node, EE provides a mechanism to replace SNA Network Interconnection (SNI) function in a way that does not require SNA legacy hardware, such as the 3745. See Using EE and EBN as a replacement for SNI for more information.
- SNA traffic can use OSA-E Gigabit and OSA-E 10 Gigabit Ethernet (EE can use any z/OS-supported IP network connection)
- Support for IPv4 and IPv6 addressing models
- End-to-end failure protection and data prioritization
- Compatibility with IPsec and SNA session-level encryption

Requirement: EE is valid only for APPN configurations and is an extension of APPN and high performance routing (HPR) protocols. Your subarea network must be migrated to APPN for the subareas where EE is to be deployed. For more information about APPN, see VTAM networking concepts and Connecting an APPN node to VTAM. Also see *Migrating Subarea Networks to an IP Infrastructure* SG24-5957-00.

Availability of Enterprise Extender

Enterprise Extender can be implemented on the following platforms:

- Communications Server NT
- Communications Server for Linux
- Communications Server for AIX®
- Communications Server for OS/2
- z/OS Communications Server
- Personal Communications
- Microsoft Host Integration Server

- Cisco SNASw

Restriction: EE is not supported by VTAM on any VM or VSE platform.

Hardware requirements

EE can use any interface supported by the TCP/IP stack. Use OSA-Express adapter in QDIO mode and HiperSockets for optimal performance and function. With OSA-Express3 adapters or later, you can use QDIO inbound workload queueing to further improve throughput for both inbound Enterprise Extender (EE) packets and for inbound non-EE traffic over the same interface. For more information, see QDIO inbound workload queueing in *z/OS Communications Server: IP Configuration Guide*. For intra-CEC communication, EE-over-HiperSockets provides superior performance unless CPU availability is limited. If CPU availability is limited, then EE communication using a shared OSA adapter provides optimal performance.

EE reliability and strategy

Because EE relies on IP network strategy and integrity, effective IP network design is essential to ensure successful EE implementation. The robustness of EE depends on the stability of your IP network configuration, just as any mission-critical TCP application does. With a stable IP network configuration in place, EE ensures the availability of SNA applications.

In planning your EE environment, discuss your configuration with the WAN environment architect to determine whether VPN or your firewall will be affected. You should also communicate with those responsible for MVS, because EE can affect the way OSA, OSAE, or CIP are defined.

Base a well-designed IP Infrastructure on the following considerations:

- The dynamic routing update protocols being used (such as OSPF, RIP, RIPV2, EIGRP, and so on)
- Use of a private intranet in contrast to the public Internet
- The type of network interfaces used, as well as MTU size
- The IP security mechanisms (NAT, IPSec, firewall)
- The quality of service (QoS) for the WAN

Figure 40 on page 136 shows the differences between a SNA network and EE using an IP network.

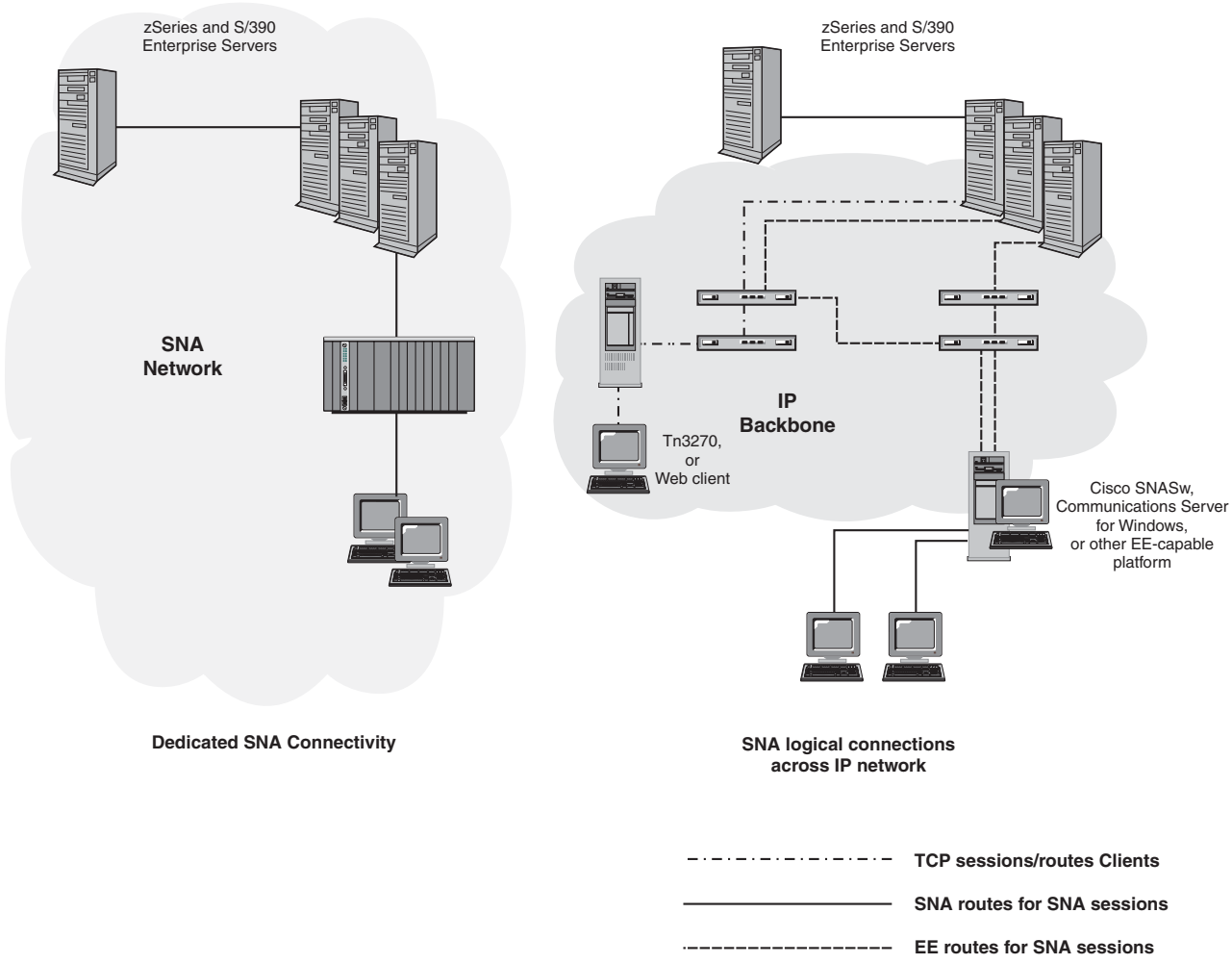


Figure 40. Comparison between an EE network and a SNA network

Using EE and extended border node (EBN) as a replacement for SNI

Enterprise Extender used with the APPN extended border node function can replace SNA Network Interconnection (SNI) function in a way that does not require SNA legacy hardware, such as the 3745. Figure 41 on page 137 shows how EE and EBN work together. Because EE with EBN requires that both endpoints be migrated from SNI, using EE with EBN as an SNI replacement technology requires cooperative changes with other partner companies.

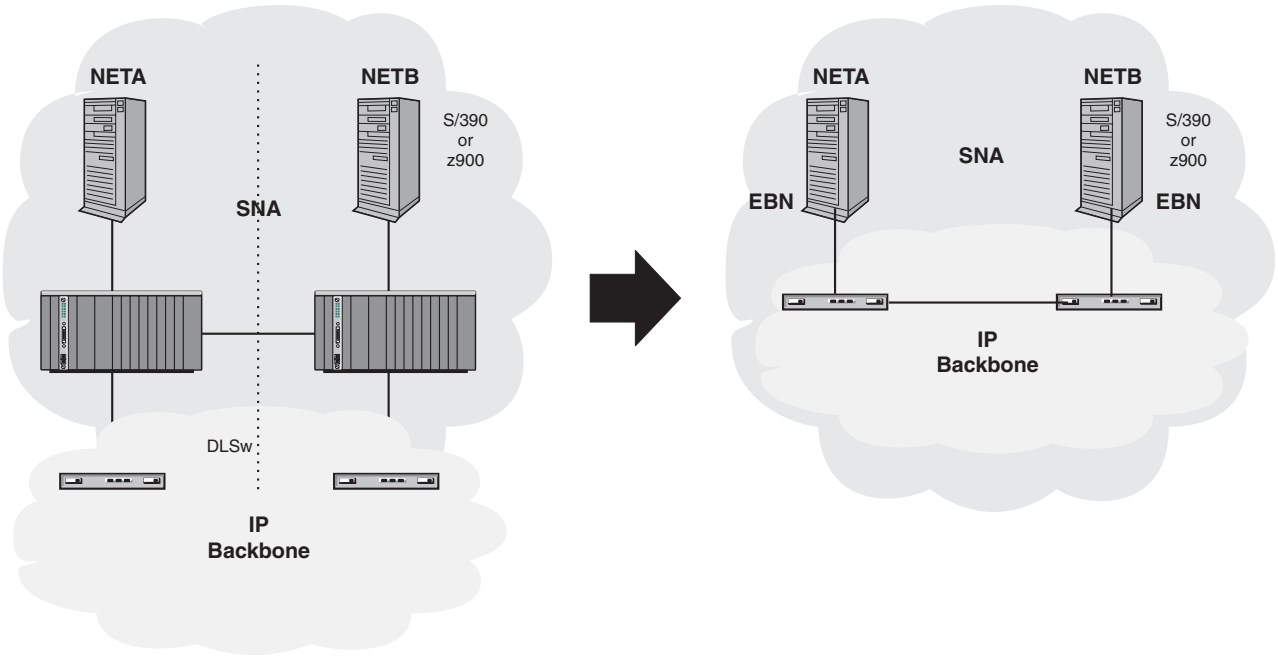


Figure 41. How EE and EBN work together

EE implementation considerations

Before implementing EE, carefully consider your business requirements, especially the following:

- Identify your preferences; will you need dynamic or predefined connections?
- Evaluate your priorities. Which of the following considerations is most important in your environment?
  - Less maintenance
  - Less definition
  - More control
  - More security
- Do you want to use host names or IP addresses for IP configuration? Do you maintain a DNS to resolve host names to IP addresses? The benefit of maintaining your own DNS is that you can keep all IP addresses in a central location. If the IP addressing scheme changes, you do not have to update VTAM definitions because DNS resolves the host names to the correct IP addresses.
- Decide whether you want to use EE connection network and which EE partner nodes will participate in the EE connection network.
- Determine whether network address translation (NAT) is being used in your IP network. See Network address translation (NAT) considerations for more information related to using host names.

Designing the EE network

Proper design is critical as you plan how to implement EE in your network. To make your implementation effective, consider these design guidelines:

- Spread EE connections for a balanced workload and avoid sending all your EE connections through your primary host.

- Use direct EE connections to endpoints.
- Consider an EE connection network, which might simplify definitions.
- Avoid using VTAM as a router.

This section explains how static VIPA addresses represent EE endpoints to z/OS Communications Server, how you can use multiple VIPA addresses to your advantage, how you need to consider class of service (COS) and quality of service (QoS), and how to decide whether you should specify addresses using host name definitions or IP address definitions.

Distinctions between an EE network and an EE connection network

You can implement EE in one of two ways:

- A simple EE connection. In this implementation there are individual connections between endpoints. This connection can be either static with each endpoint of the connection defined with a predefined switch PU or dynamic with the DYNPU operand allowing for dynamically defined dial-in connections.
- An EE connection network. This implementation uses a shared access transport facility (SATF) involving a common virtual routing node for communication. This connection is dynamically built as needed. EE connection network is explained in more detail in the section Configuring the EE connection network .

Evaluate whether an EE connection network applies to your network. A connection network is a representation of a shared access transport facility (SATF) that enables nodes to identify their connectivity to the SATF by a common virtual routing node in order to communicate without having individually defined connections to one another.

Many customers prefer to use an EE connection network because it provides a balance between control and usability. You do not need an EE connection network to have dynamic definitions; but if you use dynamic definitions, an EE connection network can be a simplified means of enabling dynamic connectivity between a set of EE partners.

Generally, all PUs associated with an EE connection network environment are dynamic. There are no predefined EE partner addresses, just the VRN definitions. Each node must have a predefined APPN connection (EE or otherwise) that provides end-to-end APPN locate capability in addition to the VRN definitions. With an EE connection network, the following are reduced:

- Definitions
- Activation actions
- Maintenance of switched major nodes

Characteristics of EE connections

EE is a simple set of extensions to the existing, open HPR technology. It integrates the HPR frames using User Datagram Protocol/Internet Protocol (UDP/IP) packets to deliver dependable SNA networking benefits to corporate intranets and the Internet. EE provides the following:

- Ability to use all z/OS Communications Server IP DLCs
- Seamless IP/SNA integration

For more information about HPR, see Chapter 17, “Implementing an APPN network,” on page 487.

From the perspective of the HPR network, the connection across the IP backbone looks like a logical link; in the IP network, the SNA traffic is just UDP application datagrams that are routed to the IP backbone. Unlike gateways, there is no protocol transformation; and unlike common tunneling mechanisms, the integration is performed at the routing layers without the overhead of additional transport functions.

Sessions do not have to go through a specific subarea (VTAM and NCP) attachment point (boundary function) because EE is based on peer networking. This helps to optimize network flow.

Restriction: Because EE relies on the APPN/HPR protocols, only SNA sessions are supported. SNA sessions involving both independent LUs and dependent LUs are supported across EE connections, with DLUR providing the boundary services for the dependent LUs.

Static VIPA considerations

EE endpoints are represented to the z/OS Communications Server network using static VIPA addresses. An EE endpoint can have one or more EE static VIPA addresses. A VIPA address can be dedicated to an EE endpoint, or it can be shared with other applications.

Guideline: Use a dedicated EE VIPA address that is not shared with other applications to simplify problem determination and network management.

Restriction: EE does not support dynamic VIPA.

Using multiple VIPAs

There are several advantages to having multiple VIPAs.

- You can provide different VIPAs to different vendors, which eases the tasks of network management, problem determination, and firewall administration.
- You can define multiple connection networks.
- There is more potential for flexibility in making IP routing decisions.

Requirement: All local Enterprise Extender VIPAs must belong to the same TCP/IP stack. VTAM establishes affinity to a single TCP/IP stack when the first EE line is activated; VTAM cannot use multiple stacks concurrently.

Parallel transmission groups (TGs)

You have the option of defining parallel EE TGs by using different EE VIPAs for one or both of the endpoints. This method of defining parallel TGs might be useful in some cases, because, depending on the IP network topology and the IP routing protocol in use, different EE TGs can map to different IP routes.

Rule: Do not code different SAP values to define parallel TGs.

Rule for multiple EE connections

z/OS Communications Server restricts how multiple EE connections are established between two EE endpoints.

When establishing multiple EE connections between the same two EE endpoints (parallel EE connections), you cannot use the same IP address pair for more than

one EE connection. Instead, every EE connection must use a unique IP address pair. You can define parallel EE connections by using a different local static VIPA address or remote IP address (or both) for each connection. z/OS Communications Server does not prevent parallel EE connections from being established as long as each connection uses a unique IP address pair. There is no limit to the number of EE connections that can be established using unique IP address pairs.

Before V1R8, an alternate method allowed parallel EE connections between the same IP address pair. As of V1R8, the initiation of parallel EE connections between the same IP address pair is no longer supported. This restriction applies only to parallel EE connections that are initiated by VTAM. An inbound parallel EE connection request, that uses the same IP address pair and different SAP values, will be accepted by VTAM.

Rule: Do not define parallel EE connections between the same IP addresses. Because each of these parallel connections traverse the same IP network, there are no advantages with regards to either bandwidth exploitation or availability.

IP multipath considerations

The IPCONFIG MULTIPATH and IPCONFIG6 MULTIPATH parameters in the TCP/IP profile enable the multipath routing selection algorithm for outbound IP traffic (the default value is NOMULTIPATH.). When multipath routing is enabled, there are two options: PERCONNECTION and PERPACKET; PERCONNECTION is the default value. If multipath routing is enabled in the TCP/IP stack for EE traffic, it does not matter which option you choose. In either case, IP uses a per-batch of packets from VTAM approach. Because there is no UDP connection for EE, true per-connection multipath is not possible. Per-packet multipath routing is too granular because it leads to too much resequencing overhead at the RTP receiving endpoint.

Multipath for Enterprise Extender (EE) is disabled by default. Enabling multipath for EE can lead to poor performance. A common problem is that one or more of the routes might not have a path to the destination IP address. This can result in packet loss of up to as much as 50 percent. Another problem is that one route might be slower than the others, which can cause half of the packets to arrive out of order. Both of these conditions result in poor send rates for the HPR connections that are using the EE connections. Multipath for EE is controlled with the VTAM start option MULTPATH. To enable multipath for EE, code MULTIPATH in the TCP/IP profile and code MULTPATH=TCPVALUE in the VTAM start options. To disable multipath for EE, code the VTAM start option MULTPATH to the value NO or allow MULTPATH to default. If MULTIPATH was coded in the IPCONFIG statement, all other applications will continue to use multipath.

Class of Service preservation dependencies

With EE, you can adhere to SNA transmission priorities across IP networks, sending batch transfers as well as smaller interactive sessions efficiently. EE maps SNA transmission priorities to the IP type of service (ToS) header. To take advantage of the mapping, you must correctly configure IP routers and switches to accommodate the IP ToS settings. If the configuration is incorrect, SNA priorities are not applied to data as it travels across the IP network.

Distinct routing algorithms can differentiate among IP packet priorities, and these algorithms have to be established as you set up routing hardware. You need to set up routers to process the ToS value that you want. SNA transmission priorities are

competing against native IP applications and every other application using the WAN; for example, FTP, Telnet, and Web-based traffic. EE is just another application competing for priority.

Establish traffic prioritization (ToS settings) as you design your EE network. Consult the appropriate documentation specific to your router to establish the correct traffic prioritization. Host personnel and network engineers need to agree on ToS values so that applications get the priority they need. If this is not done before implementation, EE will not be effective and SNA traffic will not get the proper priority on the router. Figure 42 shows how ToS settings affect IP traffic.

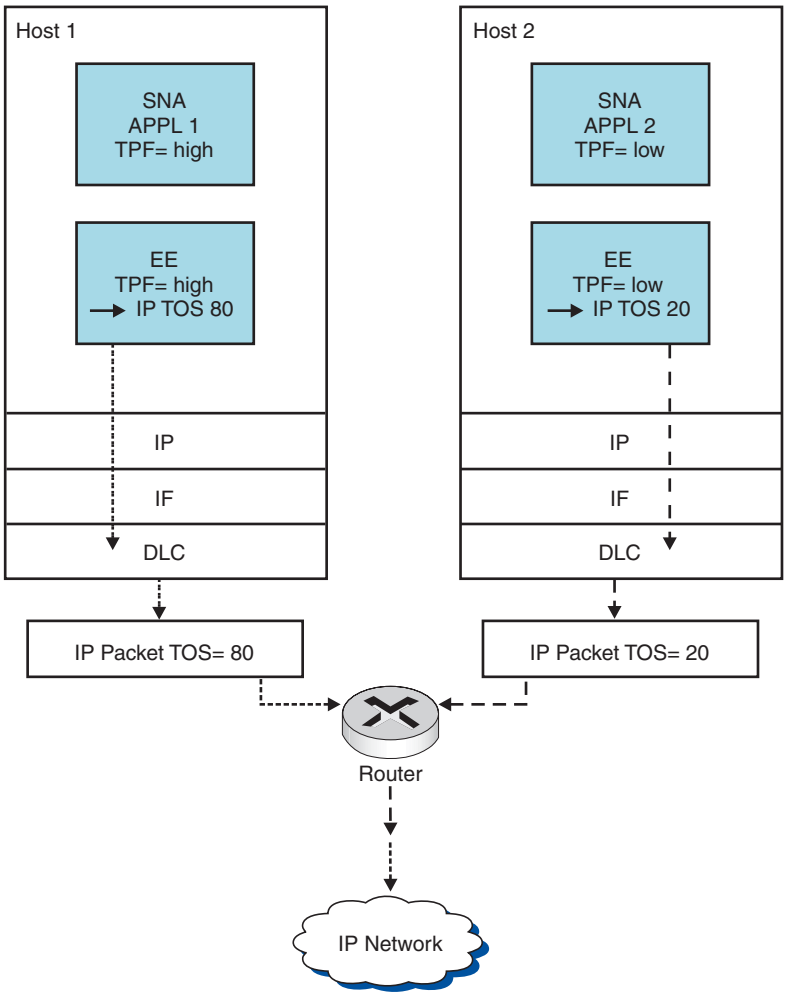


Figure 42. How ToS settings affect IP traffic

See *z/OS Communications Server: IP Configuration Guide* for more information on ToS.

Comparing host name and IP address definitions

To configure EE, the IP addresses on both sides of a connection need to be known; one on the local side, and one on the remote side. EE creates the logical connection between these two addresses. The following rules apply to using a host name instead of an IP address.

Rules:

- If you use EE across IPv6 networks, you must use host names to define endpoints.
- If you define EE connection networks, use host name definitions instead of IP addresses.
  - If you are using an EE connection network over an IP network with NAT boundaries, you must use host names.
  - If you are using an EE connection network over an IP network without NAT boundaries, using host names ensures compatibility with future IP addressing schemes.

Types of definitions

When defining EE connections, you can choose between predefined (static) or dynamic definitions.

Static definitions are more secure, which is often a priority for host connections. The disadvantage is that more labor is required to maintain definitions.

Dynamic definitions are less secure but require fewer definitions to maintain.

Table 5. Comparison between static and dynamic definitions

Predefined (static) definitions	**Dynamic definitions**
Elaborate implementation; many individual definitions; DYNPU=NO	Streamlined implementation; no individual definitions; DYNPU=YES
More secure	Less secure
Ability to manage specific connections with user-defined names	Minimal control over assigned names
More flexible - different characteristics for different connections	Less flexible - most connections have same set of characteristics
Transmission group number (TGN) helps identify partner; specific TGNs can differentiate media type to enhance problem determination	Control of TG number only by defining TGN on the DYNTYPE=EE PU entry in a model major node
More resources (definitions) to maintain	Self-managing

It is possible to have EE connections between the following endpoints:

- Two static end points
- One static end point and one dynamic end point using DYNPU=yes
- Two dynamic end points defined with connection network

Configuring the EE network

Before you begin to configure your EE network, you need to plan how you want to configure it. The following steps might help in the planning phase:

1. Draw a logical picture of your network and decide what types of EE connectivity you want to implement. Four choices are represented in Figure 43 on page 143.
   - Host-to-host with same NETID

     This type of EE connection is between z/OS hosts within your data center.
   - Host-to-host with extended border node to vendor

This type of EE connection uses EE along with VTAM extended border node to connect two vendors, replacing existing SNI connectivity.

- Host-to-branch

  This type of EE connection sets a path to one or more branches across a WAN. This connection requires an EE-capable access point in the branch such as IBM Communications Server for Linux on Intel or PCOMM.

- HiperSockets within a CEC

  This type of EE connection is made between LPARs within a CEC.

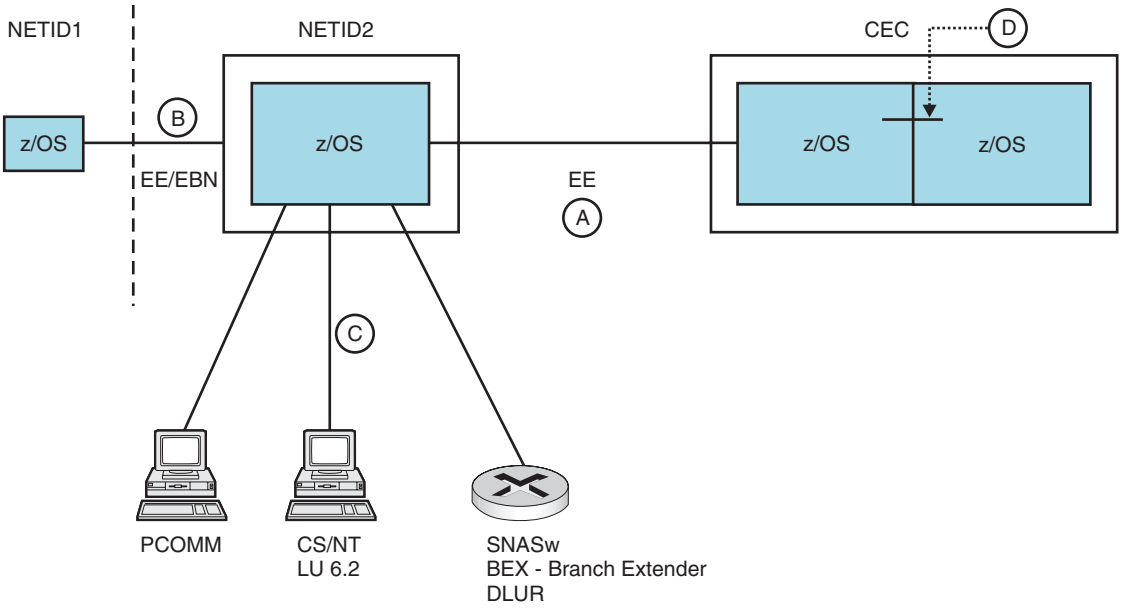


Figure 43. Four types of EE connectivity

2. Decide on the level of control, flexibility, and ease of use you need, as well as a naming scheme. See the discussion in the section called "Types of definitions" on page 142.
3. Review the following terms:

   model major node
   Defines a model PU to be used for dynamically created EE PUs on inbound connections.

   switched major node
   Defines the switched TGs with the following statements:
   - The PU statement defines the remote node.
   - The PATH statement defines the remote IPv4 address, or a host name that can be resolved into an IPv4 or an IPv6 address.

   XCA major node
   Defines the IP port with the following statements:
   - The PORT statement identifies the name of a port through which an HPR connection through the IP network is made. The PORT statement must be set to MEDIUM=HPRIP for EE.
   - One or more GROUP statements define a set of lines and PUs that represent potential EE partners. All lines and PUs under a given group are associated with a particular local IP address or host name.

4. Decide what types of EE connectivity you want to implement. All four types have these benefits:
   - Increased bandwidth because EE can use OSA Express adapters.
   - Elimination of dependency on SNA legacy hardware such as the 3745 and ESCON-based channel connectivity.

   In addition, for host-to-host with EBN, another benefit is the elimination of Gateway NCP definitions. With HiperSockets, EE provides SNA applications with the ability to communicate across a high-speed low-latency data communication path, with no network connection and no special hardware.

Requirement: To fully evaluate the best option for your environment, you must ensure that your WAN hardware and firewalls accommodate the EE connectivity of your choice. Any hardware device performing IP filtering must allow UDP traffic on ports 12000 through 12004 both inbound and outbound. If your organization does not want to allow UDP packets to flow through the firewall, you can limit UDP traffic to a subset of trusted partners using IP filtering.

Rule: Do not modify the default value for the IPPORT operand or EE interoperability problems might result.

Remember that EE works no better than your IP network. A healthy WAN environment and stable IP connectivity ensures successful EE connections.

Preservation of SNA transmission priority

In Logmode/COS assignment, every session is assigned an APPN COS with an associated transmission priority so that data flowing over that session is one of the four transmission priorities (network, high, medium, or low). EE attempts to preserve that transmission priority across the WAN by mapping the transmission priority to an appropriate IP type of service (ToS) byte setting and UDP port. Configure IP WAN routers to handle appropriate transmission priorities.

Guideline: EE also attempts to preserve SNA transmission priority by mapping each SNA TP to a corresponding UDP port. However, traffic prioritization using ToS byte management is generally preferable to port-based prioritization, because port-based prioritization schemes do not work well in the presence of IP fragmentation or IPSec-encrypted traffic.

Network address translation (NAT) considerations

Because EE depends on IP to establish its connectivity, NAT can pose difficulties in some situations. If you choose to use NAT, EE requires static NAT. Dynamic NAT is not appropriate because an address needs to be determined at configuration time. If a NAT boundary is associated with an EE connection network path, then host name-based EE definitions are required. See “Configuring the EE connection network” on page 161 for more information.

Restrictions:

- Do not use Network Address Port Translation (NAPT) if you are using EE.
- Do not use dynamic NAT if you are using EE.

Steps for configuring and activating an EE network

Following is an overview of the steps required for configuring and activating an EE network:

To configure an EE network:

1. Prepare your TCP/IP configuration
2. Evaluate IP address resolution
3. Prepare VTAM definitions

To activate an EE network:

1. Enable z/OS Communications Server to accept EE connections
2. Establish outbound EE connections
3. Accept inbound EE connections

Step 1: Prepare your TCP/IP configuration

1. Reserve the UDP ports for EE using the PORTRANGE statement or PORT statements to prevent another application from using one of the EE ports.

   Tips:

   - If you do not reserve the UDP ports, you might have a conflict when another application uses the port. For example:

     ```
      PORTRANGE
     12000 5 UDP NET  ; RESERVE UDP PORTS 12000-12004 FOR EE
     ```

   - Use the MVS job name associated with the VTAM started task to reserve UDP ports that are to be used for EE network connections. The MVS job name for a given started task can be assigned based on various inputs. These inputs are examined in the following order:

     a. The job name specified in the JOBNAME= parameter or the identifier specified on the MVS START command. In the PORTRANGE example, VTAM was started with an identifier of NET and therefore the job name specified was NET.

     b. The job name specified on the JOB JCL statement within the member.

     c. The member name.

2. Evaluate your IP routing environment. See Static VIPA considerations for more information. Also identify the number of static VIPAs.
3. Define one or more static VIPAs for the EE connection. The following are the TCP/IP profile requirements for Enterprise Extender:

   - For IPv4, define the static VIPA addresses for EE by using the DEVICE, LINK, and HOME statements. The following example defines two local static IPv4 VIPA addresses for EE:

     ```
     ;*******************************
     ; Enterprise Extender VIPAs, IPv4
     ;*******************************
     DEVICE VIPA01 VIRTUAL 0
     LINK LVIPA1 VIRTUAL 0 VIPA01
     DEVICE VIPA02 VIRTUAL 0
     LINK LVIPA2 VIRTUAL 0 VIPA02
     HOME
       92.1.1.1 LVIPA1
       92.1.1.2 LVIPA2
     ```

   - For IPv6, define the static VIPA addresses for EE by using the INTERFACE statement. The following example defines one static IPv6 VIPA address for EE:

     ```
     ;*******************************
     ; Enterprise Extender VIPA, IPv6
     ;*******************************
     INTERFACE VIPA6 DEFINE VIRTUAL6 IPADDR 2001:0db8::91:1:1:1
     ```

- Optionally, enable SOURCEVIPA. The EE static VIPA is used as the source IP address for all EE traffic, regardless of the SOURCEVIPA setting. EE uses the static VIPA as the source IP address in all outbound traffic regardless of the SOURCEVIPA setting. If you have NOSOURCEVIPA coded or defaulted, non-EE traffic uses the outbound interface address as the source IP address but EE traffic still uses the static VIPA as the source IP address.

4. Define IUTSAMEH connectivity either by using DYNAMICXCF or by creating manual IUTSAMEH definitions in the TCP/IP profile.

   If you configure manual IUTSAMEH definitions and also configure the DYNAMICXCF definition, the manual IUTSAMEH definitions will coexist with the other connectivity (XCF and HiperSockets) definitions that dynamic XCF can generate.

   To dynamically configure IUTSAMEH, do the following:

   - IUTSAMEH definitions are dynamically defined when dynamic XCF support is enabled using IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF. For IPv4, the device name is IUTSAMEH and the link name is EZASAMEMVS. For IPv6, the interface name is EZ6SAMEMVS.

   To manually configure IUTSAMEH definitions, do the following:

   - For IPv4, define and start IUTSAMEH using the DEVICE, LINK, HOME, and START statements.
   - For IPv6, define and start IUTSAMEH using the INTERFACE and START statements.

   Following is an example of a manual configuration:

```
;*******************************
; VTAM to TCPIP Stack, IPv4
;*******************************
DEVICE IUTSAMEH MPCPTP
LINK EELINK MPCPTP IUTSAMEH
HOME 93.1.1.1 EELINK
START IUTSAMEH


; ****************************
; VTAM to TCPIP Stack, IPv6
; ****************************
INTERFACE EELINK6 DEFINE MPCPTP6
   TRLENAME IUTSAMEH
   IPADDR 2001:0db8::91:1:1:1
START EELINK6
```

   Tip: To simplify EE definitions, consider defining IUTSAMEH connections using the IPCONFIG DYNAMICXCF statement for IPv4 and the IPCONFIG6 DYNAMICXCF statement for IPv6.

Step 2: Evaluate IP address resolution

Consider the following questions:

- Do you have a DNS infrastructure?

  Define name-to-address resolutions in a local hosts file (for example, ETC.IPNODES) or in the appropriate DNS zone files. The following are examples of how to code the resolutions in ETC.IPNODES:

```
92.3.1.1             SSCP2A.RALEIGH.COM
2001:0db8::91:1:1:1  SSCP2AV6.RALEIGH.COM
```

- Do you use host name support or explicit address coding?

  There are several options for specifying the source VIPA address to be used for a specific EE connection:

– Use the TCPNAME start option to identify the TCP/IP stack name from which the IPv4 source VIPA address is to be obtained.
– Use the IPADDR start option to explicitly identify the IPv4 source VIPA address.
– Use the HOSTNAME start option to identify the host name value to be resolved using name-to-address translation into the correct IPv4 or IPv6 source VIPA address.
– Use the HOSTNAME or IPADDR parameters on individual GROUP definition statements within the EE XCA major node.

Guideline: If using the HOSTNAME parameter in a common INET environment, see TCPNAME in *z/OS Communications Server: SNA Resource Definition Reference*.

Tip: The HOSTNAME and IPADDR parameters are mutually exclusive. The HOSTNAME parameter value overrides the IPADDR value. See *z/OS Communications Server: SNA Resource Definition Reference* for details.

For predefined EE connections, you can specify or obtain the remote IP address of an EE connection in one of the following ways:

– Specify the IPv4 address of the remote partner on the IPADDR operand on the PATH statement of the switched major node used for EE.
– Specify the TCP/IP host name of the remote address. This is specified using the HOSTNAME operand of the PATH statement of the switched major node used for EE. The host name can be resolved to an IPv4 or an IPv6 address.

If you use host name instead of an IP address, you must perform the following additional steps to enable this function:

1. Configure the resolver.

   VTAM uses the TCP/IP system resolver to perform name-to-address resolution. The system resolver, in general, first attempts to resolve the name by way of one or more name servers (as defined by resolver configuration statements). If unsuccessful, the resolver then attempts to resolve the name using a local host table, such as HOSTS.SITEINFO (only for IPv4 addresses) or ETC.IPNODES. For more information about defining name servers to the resolver, see *z/OS Communications Server: IP Configuration Guide*.

   The search order for selecting the TCPIP.DATA file to use is documented in *z/OS Communications Server: IP Configuration Guide*. VTAM uses the native MVS search order. The concept of data set concatenation does not apply here. You may choose to use SYSTCPD for allocating the TCPIP.DATA file. In this case, this statement must be included in your VTAM start procedure.

   If you want to use a local table to resolve host names instead of a name server, see *z/OS Communications Server: IP Configuration Guide*.

   Note: Enterprise Extender processing uses the native MVS search order for choosing the correct local host table. See *z/OS Communications Server: IP Configuration Guide* for details on the search order.
2. Configure the TCPIP.DATA file.
3. See *z/OS Communications Server: IP Configuration Guide* for information about UNIX Systems Services security considerations and required security definitions.

Guidelines:

1. While architecturally able to extend to a maximum of 255 characters, the length of the EE-related HOSTNAME operand is restricted by z/OS Communications Server to 64 characters. However, the length of the host name you choose to represent the static VIPA should be much shorter than

the maximum. The host name is exchanged between APPN nodes in APPN topology and during APPN session setup flows. Those flows carry the host name within APPN control vectors, which are limited to 255 bytes. Calculations using the contents of the APPN Route Selection Control Vector (RSCV) indicate that limiting the fully qualified host name to 40 characters should allow the host name to be exchanged comfortably within reasonably sized APPN networks.

2. If you choose to use the host name function, it is crucial for consistent and predictable EE connection establishment that a given host name resolves to a single IP address. At the TCP/IP stack that owns the host name, the name-to-address resolution should produce the static VIPA address associated with the host name. At a remote EE endpoint wanting to connect to the TCP/IP stack that owns the host name, the name-to-address resolution should produce one of two addresses:
   - If no network address translation is required (for instance, if both nodes are behind the same firewall), the name-to-address resolution at the remote EE partner should generate the static VIPA address of the TCP/IP stack that owns the host name.
   - If network address translation (NAT) is required (for instance, the nodes are separated by a firewall), the name-to-address resolution at the remote EE partner should generate a NAT address. The NAT address is the address that should point to the static VIPA address at the target TCP/IP stack.
3. If you use a name server to resolve host names and your name server runs on a z/OS Communications Server node that uses SOURCEVIPA, code the VIPA address for NSINTERADDR. For more information about SOURCEVIPA, see *z/OS Communications Server: IP Configuration Guide*.
4. If you use a name server to resolve host names and your name server runs on a z/OS Communications Server node that uses VIPA addresses, but does not use SOURCEVIPA, code a non-VIPA address for NSINTERADDR.
5. If you use a name server to resolve host names and your name server runs on a z/OS Communications Server node that does not use VIPA addresses or if the name server runs on a non-z/OS Communications Server node, there are no additional coding requirements to TCPIP.DATA.

Step 3: Prepare your VTAM definitions

There are four categories of definitions you need for your APPN-enabled VTAM:

- Start options
- EE XCA major node
- Switched major node definitions
- Model major node definitions

Start options:

- EEVERIFY

  The default value for this option is ACTIVATE, which is the recommended value for EE networks. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the EEVERIFY start option.
- HPRARB

  The default value for this option is RESPMODE, which is the recommended value for EE networks. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the HPRARB start option.
- HPRPSDLY

Use this start option to specify the minimum amount of time that VTAM delays before an RTP pipe (that has an unresponsive partner) enters a path switch state. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the HPRPSDLY start option.

- HPRPSMSG

  Indicates whether VTAM should limit the number of HPR path switch messages that are issued in the event of a network outage. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the HPRPSMSG start option.

- HPRPST

  The maximum time that VTAM tries a path switch before ending a connection. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the HPRPST start option.

- HPRSESLM

  Specifies whether VTAM should limit the number of sessions assigned to each RTP connection. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the HPRSESLM start option.

- HPRSTALL

  Use this start option to set the amount of time an RTP pipe may remain continuously stalled before being stopped automatically. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the HPRSTALL start option.

- IPADDR/HOSTNAME

  Use one of these options to specify a local static VIPA address that other EE endpoints will use to communicate with this VTAM. The decision whether to use IPADDR or HOSTNAME should be based on what you decided in the previous section, "Step 2: Evaluate IP address resolution" on page 146.

- MULTPATH

  Allows you to disable IP multipath routing support for IPv4 and IPv6 Enterprise Extender connections. For more information about the MULTPATH start option, see *z/OS Communications Server: SNA Resource Definition Reference*. For more information about the IP multipath routing, see *z/OS Communications Server: IP Configuration Guide*.

- PMTUD

  Allows you to disable path MTU discovery for IPv4 and IPv6 Enterprise Extender connections. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the PMTUD start option.

- TCPNAME

  See TCPNAME in *z/OS Communications Server: SNA Resource Definition Reference* for more information.

  Tip: Although specifying TCPNAME is optional, you should specify a name for the TCP/IP stack that defines the local static VIPA for EE because it causes more reliable line activation. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the TCPNAME start option.

  Restriction: All local Enterprise Extender static VIPA addresses must be associated with a single TCP/IP stack. Even in common-INET environments, EE always establishes affinity with only one TCP/IP stack.

- T1BUF/T2BUF

  These are two buffer pool start options designed to optimize data transmission for Enterprise Extender configurations that use QDIO/iQDIO device drivers. See *z/OS Communications Server: SNA Resource Definition Reference* for information

about the T1BUF/T2BUF start options. Find out how tuning these options can benefit EE performance in Tuning the EE network.

- UNRCHTIM

  Specifies the options associated with Enterprise Extender connection network reachability awareness. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the UNRCHTIM start option.

EE XCA major node:

Activation of the EE XCA major node enables the use of EE, and establishes the following relationships.

- The PORT definition represents a relationship between VTAM and TCP/IP for sending and receiving EE packets.
- Each GROUP entry is composed of a set of LINE definitions sharing the same characteristics, such as IP address or host name. For groups that define a connection network, the APPN TG characteristics such as CAPACITY and SECURITY are defined on the GROUP statement in the XCA major node. The EEVERIFY operand on the GROUP statement overrides the EEVERIFY start option value. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the EEVERIFY operand under XCA major node for EE.

  Guideline: Ensure that the CAPACITY specified is a value representing the effective capacity of the end-to-end IP connectivity. Note that this might be a different capacity than the TCP/IP interface.
- Each LINE definition represents a potential switched connection with a partner EE endpoint.

These relationships enable VTAM and TCP/IP to communicate across a connection that is associated with both the remote and local IP addresses. This communication begins when the first line, associated with a unique IP address, is activated.

The XCA major node defines the IP port with the following statements, operands and functions:

- The PORT statement identifies the name of a port through which an HPR connection through the IP network is made. The PORT statement must be set to MEDIUM=HPRIP for EE.

  The IPPORT operand on the PORT statement specifies the first of five consecutive UDP ports intended for all EE communication from this host.

  Requirements:

  – The IPTOS operand on the PORT statement specifies values for the type of service bytes for each of the five UDP ports defined for EE. These IP precedence values within the ToS byte must be consistent throughout the network. By coding or defaulting the IPTOS operand, you preserve SNA transmission priority through the IP type of service byte that is recognized by IP networking hardware. Further customization of the ToS byte can be done through the z/OS IP policy agent. See the sections on IPTOS, XCA major node, Customizing IP type of service, and QoS preservation dependency in *z/OS Communications Server: SNA Resource Definition Reference* for more information.

  – Use the default value of 12000 when specifying IPPORT.

  The LIVTIME, SRQTIME, and SRQRETRY logical data link control (LDLC) parameters are specified on the PORT or GROUP statement. More information about these parameters is in *z/OS Communications Server: SNA Resource Definition Reference*.

- The GROUP statement must be set to DIAL=YES to define a TG. Optionally, the GROUP statement can specify the source VIPA address explicitly or implicitly, using a host name value to be resolved into the address. If the IP address or host name is not defined on the XCA major node, the IP address or host name defined using VTAM start options is used as the source VIPA address. You must define separate GROUPs for IPv4 and IPv6 EE connections.

Guideline: Define at least as many lines as the maximum number of concurrent EE connections that are expected. Specifying DYNPU=YES on the GROUP statement indicates lines within the group that are eligible for dynamic PU assignments for inbound connections. This eliminates the need for explicit switched PU definitions for those connections. You can customize dynamic PUs for EE by defining a DYNTYPE=EE entry in a model major node. See Model major node definition for more information.

Requirements:

- For EE COS preservation, IP WAN devices must be enabled for QoS.
- All EE definitions must be coded within a single XCA major node.

Following is a sample XCA major node definition for Enterprise Extender:

```
************************************************************
*
* NAME: XCA1A   XCA MAJOR NODE FOR HOST 1A Enterprise Extender
*
************************************************************
XCA1A    VBUILD  TYPE=XCA
PORT1A   PORT   MEDIUM=HPRIP,IPPORT=12000,
                IPTOS=(20,40,80,C0,C0),LIVTIME=(10,30),
                SRQTIME=15,SRQRETRY=3
*
GP1A2A   GROUP  DIAL=YES,ANSWER=ON,ISTATUS=INACTIVE,
                CALL=INOUT,
                EEVERIFY=60,
                IPADDR=9.1.1.1,LIVTIME(15,40),SRQTIME=20,SRQRETRY=5

LN1A2A   LINE
P1A2A    PU
*
GP1A2A1  GROUP  DIAL=YES,ANSWER=ON,ISTATUS=INACTIVE,
                CALL=IN,DYNPU=YES,EEVERIFY=NEVER,
                IPADDR=9.1.1.2
LN1A2A2  LINE
P1A2A2   PU
*
* HOSTNAME resolves to IPv6 address
GP1A2A2  GROUP  DIAL=YES,ANSWER=ON,ISTATUS=INACTIVE,CALL=INOUT,
                HOSTNAME=HOST.DOMAIN.COM
LN1A2A2  LINE
P1A2A2   PU
```

Switched major node definition:

The switched major node defines the Enterprise Extender switched TGs with the following statements:

- The PU statement defines the remote node
- The PATH statement defines the remote IPv4 address, or a host name that can be resolved into an IPv4 or an IPv6 address.

The PU definitions can be used for dialing outbound and for identifying PUs for incoming calls to this host. See the section on switched major node definitions in *z/OS Communications Server: SNA Resource Definition Reference* for more information.

Dial guidelines: For predefined EE connections:

- Coding a PATH definition statement with GRPNM enables an incoming call to prefer one switched PU over another. When the GROUP name that is associated with the line selected for the inbound call matches the GRPNM on the switched PATH, the switched PU associated with the PATH is preferred over another switched PU that does not match, or does not have a PATH definition statement coded.
- Specify APPN TG characteristics on the switched PU used for dial-out connections or on the DYNTYPE=EE model PU for dynamically defined dial-in connections.
- Specify the EEVERIFY operand on the switched PU used for dial-out connections or on the DYNTYPE=EE model PU for dynamically defined dial-in connections to override the EEVERIFY start option value. See *z/OS Communications Server: SNA Resource Definition Reference* for information about the EEVERIFY operand in the Switched major node or Model Major node sections, under the DYNTYPE=EE description.

Following is a sample switched major node definition for Enterprise Extender.

```
************************************************************
*
*  NAME: SWXCA1A   SWITCHED MAJOR NODE FOR HOST 1A
*
************************************************************
SWXCA1A  VBUILD  TYPE=SWNET,MAXNO=256,MAXGRP=256
*
SW1A2A   PU     IDBLK=003,IDNUM=00003,ADDR=01,
                CPNAME=SSCP2A,EEVERIFY=60,
                CPCP=YES,HPR=YES,
                PUTYPE=2,TGP=GIGENET
PATH2A   PATH   IPADDR=93.1.1.1,
                GRPNM=GP1A2A
*
* HOSTNAME resolves to IPv4 address
SW1A2A1  PU     IDBLK=004,IDNUM=00004,ADDR=02,
                CPNAME=SSCP2A,
                CPCP=YES,HPR=YES,
                PUTYPE=2,TGP=GIGENET
PATH2A1  PATH   HOSTNAME=SSCP2A,
                GRPNM=GP1A2A1
*
* HOSTNAME resolves to IPv6 address
SW1A2A2  PU     IDBLK=005,IDNUM=00005,ADDR=03,
                CPNAME=SSCP2A,
                CPCP=YES,HPR=YES,
                PUTYPE=2,TGP=GIGENET
PATH2A2  PATH   HOSTNAME=SSCP2AV6,
                GRPNM=GP1A2A2
```

Model major node definition:

To define the TG characteristics of a dynamic PU for EE, code a model EE PU within the model major node. This PU model definition is used only if all of the following are true:

- DYNPU=YES is explicitly coded on a GROUP statement under an EE PORT within an XCA major node

- A LINE statement coded under that GROUP statement is eligible to accept incoming calls (CALL=IN or CALL=INOUT, and ANSWER=ON are coded, sifted down from the GROUP statement, or taken as defaults)

  Tip: Using CALL=INOUT provides the most flexibility.
- A predefined switched PU in connectable state that matches the partner cannot be found at dial-in time

Tip: Using the model PU enables you to specify things such as non-default TG characteristics and an installation-specific DISCNT value without having to code a PU statement for each remote node that can dial in. Also, by using the DWINOP, REDIAL, and REDDELAY parameters, you can specify whether you want VTAM to attempt to reconnect to the partner when the connection becomes inoperative. You can also specify up to four predefined TG numbers in the order you prefer for your EE connections. Include ANY as one of the four if you want to have the EE connection established even if your preferred TG numbers are not available. ANY can be placed anywhere in the list.

The EEVERIFY operand on the PU statement overrides the EEVERIFY start option value.

Following is a sample model major node definition that defines a model PU to be used for dynamically created, non-connection-network EE PUs for inbound connections.

```
************************************************************
*
*  NAME: MODELA1A   MODEL MAJOR NODE FOR HOST 1A
*
************************************************************
MODELA1A VBUILD  TYPE=MODEL
*
EEMODEL  PU     DYNTYPE=EE,
                CAPACITY=100M,
                COSTTIME=0,
                CPCP=YES,
                EEVERIFY=60,
                DISCNT=NO,
                DWINOP=YES,
                REDIAL=30,
                REDDELAY=60,
                TGN=(11,8,15,ANY)
```

Activating EE

Perform the following steps to start EE. Before you begin, define static VIPAs to be used for EE connections, start the TCP/IP stack enabled for EE, and activate IUTSAMEH by static definition or by specifying DYNAMICXCF. See “Step 1: Prepare your TCP/IP configuration” on page 145 for more information on preparing your TCP/IP configuration.

1. Enable z/OS Communications Server to accept or initiate EE connections by activating the EE XCA major node
   - The first line activated in the EE XCA major node establishes a VTAM connection to the TCP/IP stack supporting Enterprise Extender. When this communication is established, the following messages appear on the console:

     ```
     EZZ4313I INITIALIZATION COMPLETE FOR DEVICE IUTSAMEH
     EZZ4324I CONNECTION TO 9.67.1.3 ACTIVE FOR DEVICE IUTSAMEH
     ```
   - For the first line activation associated with each unique static VIPA address (IPADDR or HOSTNAME), VTAM registers this VIPA address with the

TCP/IP stack for EE communications. Any other line you activate from a group results in a new message that indicates VIPA activation. The following message appears on the console:

```
EZZ4324I CONNECTION TO 9.67.1.5 ACTIVE FOR DEVICE IUTSAMEH
EZZ4324I CONNECTION TO 9.67.1.7 ACTIVE FOR DEVICE IUTSAMEH
```

Each message signifies that EE is registering the new static VIPA address with the TCP/IP stack and that the VIPA is ready to use.

2. Establish outbound EE connections by configuring a predefined EE network. Perform the following actions:
   - Activate the EE switched major node.
   - Dial the associated switched PU.

     ```
     V NET,DIAL,ID=SWIP2A1

      IST097I VARY ACCEPTED
      IST590I CONNECTOUT ESTABLISHED FOR PU SWIP2A1 ON LINE LNIP1
      IST1086I APPN CONNECTION FOR NETA.SSCP2A IS ACTIVE - TGN = 21
      IST241I VARY DIAL COMMAND COMPLETE FOR SWIP2A1
      IST1488I ACTIVATION OF RTP CNR00002 AS PASSIVE TO NETA.SSCP2A
      IST1488I ACTIVATION OF RTP CNR00001 AS ACTIVE TO NETA.SSCP2A
      IST1096I CP-CP SESSIONS WITH NETA.SSCP2A ACTIVATED
     ```

     If the partner is ready to accept connections, connections will be established. To establish connections using EE connection network, see "Configuring the EE connection network" on page 161.
3. Accept inbound EE connections by activating a switched PU to receive inbound connections. Use one of the following methods to identify the switched PU that will receive the EE connection.
   - Activate a switched PU to receive the incoming connection.
   - Specify DYNPU=YES on a GROUP within the XCA major node to indicate that a dynamic PU should be created when inbound connections arrive. Optionally, specify a model to be used for creation of dynamic physical units for incoming calls by activating a model major node containing a PU statement with DYNTYPE=EE specified.

Activating EE automatically:

If you want to automate EE activation and connection to EE partners with minimal operator intervention, specify the EE XCA major node and any related EE switched major nodes in the VTAM configuration start list (ATCCONxx). Specifying DWACT=YES on the switched PU causes VTAM to automatically dial the switched PU upon activation.

Tip: The connection attempt might initially fail. Consider coding REDIAL and REDDELAY on the switched path definition statement. For more information about path definition statements see *z/OS Communications Server: SNA Resource Definition Reference*. For EE to automatically recover after a connection loss, consider coding DWINOP on the switched PU.

Guideline: Use caution if both EE endpoints support DWINOP. See Dial usability - DWACT, DWINOP, KEEPACT, REDIAL, and REDDELAY for more information.

Verifying activation

The following commands are useful in verifying that your EE network is correctly configured and performing properly.

- DISPLAY EE

  This command displays general Enterprise Extender (EE) information.

```
d net,ee

 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = EE
 IST2000I ENTERPRISE EXTENDER GENERAL INFORMATION
 IST1685I TCP/IP JOB NAME = TCPCS
 IST2003I ENTERPRISE EXTENDER XCA MAJOR NODE NAME = XCAIP
 IST2004I LIVTIME = (10,0)     SRQTIME =    15  SRQRETRY =     3
 IST2005I IPRESOLV =    0
 IST924I -------------------------------------------------------------
 IST2006I PORT PRIORITY =  SIGNAL    NETWORK     HIGH   MEDIUM     LOW
 IST2007I IPPORT NUMBER =   12000      12001    12002    12003   12004
 IST2008I IPTOS VALUE   =      C0         C0       80       40      20
 IST924I -------------------------------------------------------------
 IST2017I TOTAL RTP PIPES =          2     LU-LU SESSIONS =          2
 IST2018I TOTAL ACTIVE PREDEFINED EE CONNECTIONS          =          1
 IST2019I TOTAL ACTIVE LOCAL  VRN EE CONNECTIONS          =          0
 IST2020I TOTAL ACTIVE GLOBAL VRN EE CONNECTIONS          =          0
 IST2021I TOTAL ACTIVE EE CONNECTIONS                     =          1
 IST314I END
```

For more information about detailed breakdowns of the DISPLAY outputs see *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.

- DISPLAY EEDIAG

  This command displays Enterprise Extender (EE) connections that meet or exceed a specified retransmission threshold.

```
d net,eediag,rexmit=0,id=swip2a1

IST097I DISPLAY ACCEPTED
  IST350I DISPLAY TYPE = EEDIAG
  IST2065I ENTERPRISE EXTENDER CONNECTION REXMIT INFORMATION
  IST2067I EEDIAG DISPLAY ISSUED ON 06/13/05 AT 12:01:03
  IST924I -------------------------------------------------------------
  IST1680I LOCAL IP ADDRESS 9.67.1.1
  IST1680I REMOTE IP ADDRESS 9.67.1.2
  IST1909I REMOTE HOSTNAME SSCP2A
  IST2023I CONNECTED TO LINE LNIP1
  IST924I -------------------------------------------------------------
  IST2035I TOTALS FOR ALL PORT PRIORITIES
  IST2036I   NLPS SENT           =               42799  ( 042K )
  IST2037I   BYTES SENT          =             2995730  ( 002M )
  IST2038I   NLPS RETRANSMITTED  =                   0  ( 000K )
  IST2039I   BYTES RETRANSMITTED =                   0  ( 000K )
  IST2040I   NLPS RECEIVED       =               44186  ( 044K )
  IST2041I   BYTES RECEIVED      =             3068884  ( 003M )
  IST314I END
```

For more information about detailed breakdowns of the DISPLAY outputs see *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.

- DISPLAY TOPO

  This command displays node and link information from the topology data base.

```
d net,topo,orig=neta.sscp1a,dest=neta.sscp2a,appncos=#inter
  IST097I DISPLAY ACCEPTED
  IST350I DISPLAY TYPE = TOPOLOGY
  IST1299I TRANSMISSION GROUPS ORIGINATING AT CP NETA.SSCP1A
  IST1357I                                              CPCP
  IST1300I DESTINATION CP    TGN      STATUS   TGTYPE    VALUE WEIGHT
  IST1301I NETA.SSCP2A       21       OPER     INTERM    YES   60
  IST1579I                   ----------------------------------------
  IST2241I                                              TIME  ISL
  IST1163I                   RSN              HPR       LEFT  WEIGHT
  IST1164I                   10               YES       15    *NA*
  IST1579I                   ----------------------------------------
  IST1302I                   CAPACITY PDELAY   COSTTIME  COSTBYTE
  IST1303I                   4M       TERRESTR 0         0
```

```
IST1579I                  ----------------------------------------
IST1304I                  SECURITY UPARM1   UPARM2      UPARM3
IST1305I                  UNSECURE 128      128         128
IST1579I                  ----------------------------------------
IST1736I                  PU NAME
IST1737I                  SWIP2A1
IST924I -------------------------------------------------------------
IST2275I TDU INFORMATION SINCE LAST RESET ON 04/28/08 AT 04:18:25
IST1769I LAST TDU RECEIVED - 04/08/08 11:11:45 FROM NETA.SSCP2A
IST2281I LAST TDU SENT - 04/28/08 11:11:48
IST2282I TDU COUNTS:
IST1771I   SENT     - 9      RECEIVED - 1      ACCEPTED - 0
IST1772I   REJECTED - 1      IGNORED  - 0
IST314I END
```

For more information about DISPLAY TOPO see *z/OS Communications Server: SNA Operation*.

- DISPLAY ID=EE *puname*

  This command displays information about the EE switched PU such as local and remote IP addresses.

```
d net,id=swip2a1
  IST097I DISPLAY ACCEPTED
  IST075I NAME = SWIP2A1, TYPE = PU_T2.1
  IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
  IST1043I CP NAME = SSCP2A, CP NETID = NETA, DYNAMIC LU = YES
  IST1589I XNETALS = YES
  IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
  IST1106I SWIP2A1  AC/R    21 YES   98750000000000000000017100808080
  IST1482I HPR = RTP - OVERRIDE = N/A - CONNECTION = YES
  IST1510I LLERP = NOTPREF - RECEIVED = NOTALLOW
  IST1680I LOCAL IP ADDRESS 9.67.1.1
  IST1680I REMOTE IP ADDRESS 9.67.1.2
  IST1909I REMOTE HOSTNAME SSCP2A
  IST2114I LIVTIME:    INITIAL =   10   MAXIMUM =    0   CURRENT =   10
  IST136I SWITCHED SNA MAJOR NODE = TOIP2A
  IST081I LINE NAME = LNIP1, LINE GROUP = GPIP, MAJNOD = XCAIP
  IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
  IST1500I STATE TRACE = OFF
  IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
  IST1657I MAJOR NODE VTAMTOPO = REPORT
  IST314I END
```

For more information about DISPLAY ID see *z/OS Communications Server: SNA Operation*.

- Verify the connections by issuing the appropriate VTAM DISPLAY command:

```
d net,ee,id=swip2a1
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = EE
 IST2001I ENTERPRISE EXTENDER CONNECTION INFORMATION
 IST075I NAME = SWIP2A1, TYPE = PU_T2.1
 IST1680I LOCAL IP ADDRESS 9.67.1.1
 IST1680I REMOTE IP ADDRESS 9.67.1.2
 IST1909I REMOTE HOSTNAME SSCP2A
 IST2022I EE CONNECTION ACTIVATED ON 06/13/05 AT 11:53:51
 IST2114I LIVTIME:    INITIAL =   10   MAXIMUM =    0   CURRENT =   10
 IST2023I CONNECTED TO LINE LNIP1
 IST2025I LDLC SIGNALS RETRANSMITTED AT LEAST ONE TIME    =          0
 IST2026I LDLC SIGNALS RETRANSMITTED SRQRETRY TIMES       =          0
 IST2009I RTP PIPES =          2      LU-LU SESSIONS      =          2
 IST2027I DWINOP  = NO      REDIAL = *NA*        REDDELAY =       *NA*
 IST2028I KEEPACT = YES
 IST924I -------------------------------------------------------------
 IST2035I TOTALS FOR ALL PORT PRIORITIES
 IST2036I   NLPS SENT           =                  31  ( 000K )
 IST2037I   BYTES SENT          =                2467  ( 002K )
```

```
IST2038I   NLPS RETRANSMITTED  =                   0  ( 000K )
IST2039I   BYTES RETRANSMITTED =                   0  ( 000K )
IST2040I   NLPS RECEIVED       =                  32  ( 000K )
IST2041I   BYTES RECEIVED      =                2545  ( 002K )
IST314I END
```

- DISPLAY NETSTAT DEVLINKS

  This command generates a report that displays the information about devices and their interfaces or links that are defined to the TCP/IP stack. For more information about the DISPLAY NETSTAT command see *z/OS Communications Server: IP System Administrator's Commands*.

  d tcpip,,netstat,devlinks

```
.
.
 DEVNAME: TRLE1A            DEVTYPE: MPCPTP
  DEVSTATUS: READY
 LNKNAME: TRLE1AL           LNKTYPE: MPCPTP     LNKSTATUS: READY
     NETNUM: N/A  QUESIZE: N/A
     ACTMTU: 14336
     SECCLASS: 255
 BSD ROUTING PARAMETERS:
   MTU SIZE: N/A               METRIC: 00
   DESTADDR: 9.67.16.2         SUBNETMASK: 255.0.0.0
 MULTICAST SPECIFIC:
   MULTICAST CAPABILITY: YES
   GROUP             REFCNT
   -----             ------
   224.0.0.1         0000000001
 LINK STATISTICS:
   BYTESIN                          = 12572401
   INBOUND PACKETS                  = 129102
   INBOUND PACKETS IN ERROR         = 0
   INBOUND PACKETS DISCARDED        = 0
   INBOUND PACKETS WITH NO PROTOCOL = 0
   BYTESOUT                         = 12244450
   OUTBOUND PACKETS                 = 125059
   OUTBOUND PACKETS IN ERROR        = 0
   OUTBOUND PACKETS DISCARDED       = 0
 DEVNAME: IUTSAMEH          DEVTYPE: MPCPTP
  DEVSTATUS: READY
  LNKNAME: TOVTAM            LNKTYPE: MPCPTP     LNKSTATUS: READY
      NETNUM: N/A  QUESIZE: N/A
      ACTMTU: 65535
     SECCLASS: 255
    BSD ROUTING PARAMETERS:
      MTU SIZE: N/A              METRIC: 00
      DESTADDR: 0.0.0.0          SUBNETMASK: 255.0.0.0
    MULTICAST SPECIFIC:
      MULTICAST CAPABILITY: YES
      GROUP             REFCNT
      -----             ------
      224.0.0.1         0000000001
    LINK STATISTICS:
      BYTESIN                          = 0
      INBOUND PACKETS                  = 0
      INBOUND PACKETS IN ERROR         = 0
      INBOUND PACKETS DISCARDED        = 0
      INBOUND PACKETS WITH NO PROTOCOL = 0
      BYTESOUT                         = 0
      OUTBOUND PACKETS                 = 0
      OUTBOUND PACKETS IN ERROR        = 0
```

```
    OUTBOUND PACKETS DISCARDED        = 0
  .
  .
40 OF 40 RECORDS DISPLAYED
```

In addition to DISPLAY commands, z/OS Communications Server has a network management interface that provides EE configuration and performance data. There are various network management applications that use this interface to provide a graphical presentation of your EE network. An example is Tivoli® ITM/NP V2. See *z/OS Communications Server: IP Programmer's Guide and Reference* for more information about the z/OS network management interface

Verifying the health of EE connection

VTAM sends a Logical Data Link Control (LDLC) probe to the remote partner to determine whether all ports are accessible during the activation of the EE connection by default. EE health verification for this connection activation fails if VTAM does not receive a response, or receives an error response, from the remote partner.

VTAM also verifies the health of EE connections at other user-specified time intervals. To do this, VTAM sends an LDLC probe to the remote partner on a user-specified interval. If the LDLC probe does not reach any port, VTAM issues the eventual action error message if it is not already present. The eventual action error message stays on console until the customer clears the message or all active EE connections receive successful EE health verification during the most recent LDLC probe.

The D NET,EE,LIST=EEVERIFY command displays all lines that failed EE health verification during the most recent LDLC probe.

The D NET,EE,ID=puname or linename command displays EE health verification information of the most recent LDLC probe for a particular connection.

VTAM provides the EE health verification information of the most recent LDLC probe to the NMI application upon its request. It includes success and failure information for all ports and contains the round-trip time information.

VTAM issues the health verification failure message if health verification fails during the activation of EE connection when it sends the LDLC probe to their remote partner.

```
IST2330I EE HEALTH VERIFICATION FAILED FOR puname AT time
```

During the activation of the EE connection, VTAM senda Logical Data Link Control (LDLC) probes to the remote partner to determine whether all the five ports are accessible. VTAM does not receive a response to any of the LDLC probe requests. VTAM continues with the activation of the EE connection between this node and the remote partner. Because VTAM receives no replies to its LDLC probe requests, VTAM determines that the remote partner does not support EE health verification and VTAM issues the following message:

```
IST2342I EE HEALTH VERIFICATION NOT SUPPORTED BY puname
```

When the EE Health verification fails on active connections, VTAM issues the following eventual action message. This message remains on the console until a subsequent LDLC probe to the remote partner is successful or is erased by the operator.

```
IST2323E EE HEALTH VERIFICATION FAILED FOR ONE OR MORE CONNECTIONS
```

When you see the above message, you can issue the D NET,EE,LIST=EEVERIFY command to display each line, which failed health verification on the most recent LDLC probe to its remote partner. The command displays the following message for each line with the line name, pu name, date and time.

```
IST2325I LINE linename PU puname ON date AT time
```

Display EE,LIST=EEVERIFY command example: This output shows all lines with active EE connections and failed EE health verification on last LDLC probe.

```
D NET,EE,LIST=EEVERIFY
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2000I ENTERPRISE EXTENDER GENERAL INFORMATION
IST1685I TCP/IP JOB NAME = TCPCS1
IST2003I ENTERPRISE EXTENDER XCA MAJOR NODE NAME = XCAEE
IST2004I LIVTIME = (10,0) SRQTIME = 15 SRQRETRY = 3
IST2005I IPRESOLV = 0
IST2231I CURRENT HPR CLOCK RATE = STANDARD
IST924I -------------------------------------------------------------
IST2006I PORT PRIORITY = SIGNAL NETWORK HIGH MEDIUM LOW
IST2007I IPPORT NUMBER = 12000 12001 12002 12003 12004
IST2008I IPTOS VALUE = C0 C0 80 40 20
IST924I -------------------------------------------------------------
IST2324I EE HEALTH VERIFICATION: FAILED CONNECTION INFORMATION
IST2325I LINE LNEE4000 PU SWEE42AI ON 08/11/09 AT 20:52:43
IST2326I EE HEALTH VERIFICATION TOTAL CONNECTION FAILURES = 1
IST2017I TOTAL RTP PIPES = 5 LU-LU SESSIONS = 5
IST2018I TOTAL ACTIVE PREDEFINED EE CONNECTIONS = 1
IST2019I TOTAL ACTIVE LOCAL VRN EE CONNECTIONS = 0
IST2020I TOTAL ACTIVE GLOBAL VRN EE CONNECTIONS = 0
IST2021I TOTAL ACTIVE EE CONNECTIONS = 1
IST314I END
```

This output shows EE health verification message from the most recent LDLC probe.

```
D NET,EE,ID=SWEE2A1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2001I ENTERPRISE EXTENDER CONNECTION INFORMATION
IST075I NAME = SWEE2A1, TYPE = PU_T2.1
IST1680I LOCAL IP ADDRESS FC00::67:1:1
IST1910I LOCAL HOSTNAME VIPA16.SSCP1A.TCP.RALEIGH.IBM.COM
IST1680I REMOTE IP ADDRESS FC00::67:1:2
IST1909I REMOTE HOSTNAME VIPA16.SSCP2A.TCP.RALEIGH.IBM.COM
IST2114I LIVTIME: INITIAL = 10 MAXIMUM = 30 CURRENT = 20
IST2022I EE CONNECTION ACTIVATED ON 08/11/09 AT 20:22:55
IST2023I CONNECTED TO LINE LNEE1000
IST2327I EE HEALTH VERIFICATION OPTION - EEVERIFY = 3 MINUTES
IST2328I EE HEALTH VERIFICATION FAILED ON 8/11/09 AT 20:55:57
IST2339I EE HEALTH VERIFICATION LAST SUCCESS ON 08/11/09 AT 20:30:00
IST2025I LDLC SIGNALS RETRANSMITTED AT LEAST ONE TIME = 0
IST2026I LDLC SIGNALS RETRANSMITTED SRQRETRY TIMES = 0
IST2009I RTP PIPES = 4 LU-LU SESSIONS = 3
IST2027I DWINOP = NO REDIAL = *NA* REDDELAY = *NA*
IST2028I KEEPACT = NO
IST924I -------------------------------------------------------
IST2035I TOTALS FOR ALL PORT PRIORITIES
IST2036I NLPS SENT = 49782 ( 049K )
IST2037I BYTES SENT = 4576487 ( 004M )
IST2038I NLPS RETRANSMITTED = 0 ( 000K )
IST2039I BYTES RETRANSMITTED = 0 ( 000K )
IST2040I NLPS RECEIVED = 49780 ( 049K )
IST2041I BYTES RECEIVED = 4576724 ( 004M )
IST314I END
```

Display EE,ID=line name command example: Display NET,EE,ID=line name shows EE health verification from the most recent LDLC probe.

```
D NET,EE,ID=LNEE1000,LIST=DETAIL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2001I ENTERPRISE EXTENDER CONNECTION INFORMATION
IST075I NAME = LNEE1000, TYPE = LINE
IST1680I LOCAL IP ADDRESS FC00::67:1:1
IST1910I LOCAL HOSTNAME VIPA16.SSCP1A.TCP.RALEIGH.IBM.COM
IST1680I REMOTE IP ADDRESS FC00::67:1:2
IST1909I REMOTE HOSTNAME VIPA16.SSCP2A.TCP.RALEIGH.IBM.COM
IST2114I LIVTIME: INITIAL = 10 MAXIMUM = 30 CURRENT = 20
IST2022I EE CONNECTION ACTIVATED ON 08/11/09 AT 14:14:26
IST2024I CONNECTED TO SWITCHED PU SWEE2A1
IST2327I EE HEALTH VERIFICATION OPTION - EEVERIFY = 3 MINUTES
IST2328I EE HEALTH VERIFICATION FAILED ON 8/11/09 AT 20:55:57
IST2339I EE HEALTH VERIFICATION LAST SUCCESS ON 08/11/09 AT 20:30:00
IST2025I LDLC SIGNALS RETRANSMITTED AT LEAST ONE TIME    =          0
IST2026I LDLC SIGNALS RETRANSMITTED SRQRETRY TIMES       =          0
IST2009I RTP PIPES = 4 LU-LU SESSIONS = 3
IST2027I DWINOP = NO REDIAL = *NA* REDDELAY = *NA*
IST2028I KEEPACT = NO
IST924I -------------------------------------------------------------
IST2030I PORT PRIORITY = SIGNAL
IST2029I MTU SIZE = 1232
IST2036I NLPS SENT = 4 ( 000K )
IST2037I BYTES SENT = 526 ( 000K )
IST2038I NLPS RETRANSMITTED = 0 ( 000K )
IST2039I BYTES RETRANSMITTED = 0 ( 000K )
IST2040I NLPS RECEIVED = 4 ( 000K )
IST2041I BYTES RECEIVED = 577 ( 000K )
IST924I -------------------------------------------------------------
IST2031I PORT PRIORITY = NETWORK
IST2029I MTU SIZE = 1232
IST2036I NLPS SENT = 91 ( 000K )
IST2037I BYTES SENT = 7036 ( 007K )
IST2038I NLPS RETRANSMITTED = 0 ( 000K )
IST2039I BYTES RETRANSMITTED = 0 ( 000K )
IST2040I NLPS RECEIVED = 90 ( 000K )
IST2041I BYTES RECEIVED = 6693 ( 006K )
IST924I -------------------------------------------------------------
IST2032I PORT PRIORITY = HIGH
IST2029I MTU SIZE = 1232
IST2036I NLPS SENT = 56390 ( 056K )
IST2037I BYTES SENT = 5184997 ( 005M )
IST2038I NLPS RETRANSMITTED = 0 ( 000K )
IST2039I BYTES RETRANSMITTED = 0 ( 000K )
IST2040I NLPS RECEIVED = 56394 ( 056K )
IST2041I BYTES RECEIVED = 5185656 ( 005M )
IST924I -------------------------------------------------------------
IST2033I PORT PRIORITY = MEDIUM
IST2029I MTU SIZE = 1232
IST2036I NLPS SENT = 0 ( 000K )
IST2037I BYTES SENT = 0 ( 000K )
IST2038I NLPS RETRANSMITTED = 0 ( 000K )
IST2039I BYTES RETRANSMITTED = 0 ( 000K )
IST2040I NLPS RECEIVED = 0 ( 000K )
IST2041I BYTES RECEIVED = 0 ( 000K )
IST924I -------------------------------------------------------------
IST2034I PORT PRIORITY = LOW
IST2029I MTU SIZE = 1232
IST2036I NLPS SENT = 0 ( 000K )
IST2037I BYTES SENT = 0 ( 000K )
IST2038I NLPS RETRANSMITTED = 0 ( 000K )
IST2039I BYTES RETRANSMITTED = 0 ( 000K )
IST2040I NLPS RECEIVED = 0 ( 000K )
```

```
IST2041I BYTES RECEIVED = 0 ( 000K )
IST924I -------------------------------------------------------------
IST2035I TOTALS FOR ALL PORT PRIORITIES
IST2036I NLPS SENT = 56485 ( 056K )
IST2037I BYTES SENT = 5192559 ( 005M )
IST2038I NLPS RETRANSMITTED = 0 ( 000K )
IST2039I BYTES RETRANSMITTED = 0 ( 000K )
IST2040I NLPS RECEIVED = 56488 ( 056K )
IST2041I BYTES RECEIVED = 5192926 ( 005M )
IST314I END
```

Configuring the EE connection network

Your EE network might benefit from having an EE connection network. A connection network is a representation of a shared access transport facility (SATF) which enables nodes that identify their connectivity to the SATF by a common virtual routing node (VRN) to communicate without having individually defined connections to one another. Figure 44 shows VRN connectivity. A connection network is also referred to as a virtual routing node (VRN).

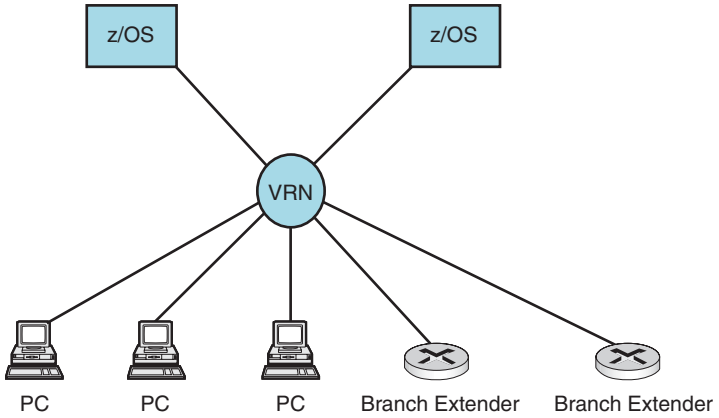


Figure 44. VRN connectivity

Before you begin, you should understand the benefits and liabilities of an EE connection network. These are shown in the following figures.

Figure 45 on page 162, when an LU-LU session between resources on Communications Server for Linux and HOSTB is set up, the optimal route is directly through the network. However, if Communications Server for Linux and HOSTB are not directly defined to each other or are not connected to the same connection network, the indirect route is selected through HOSTA.

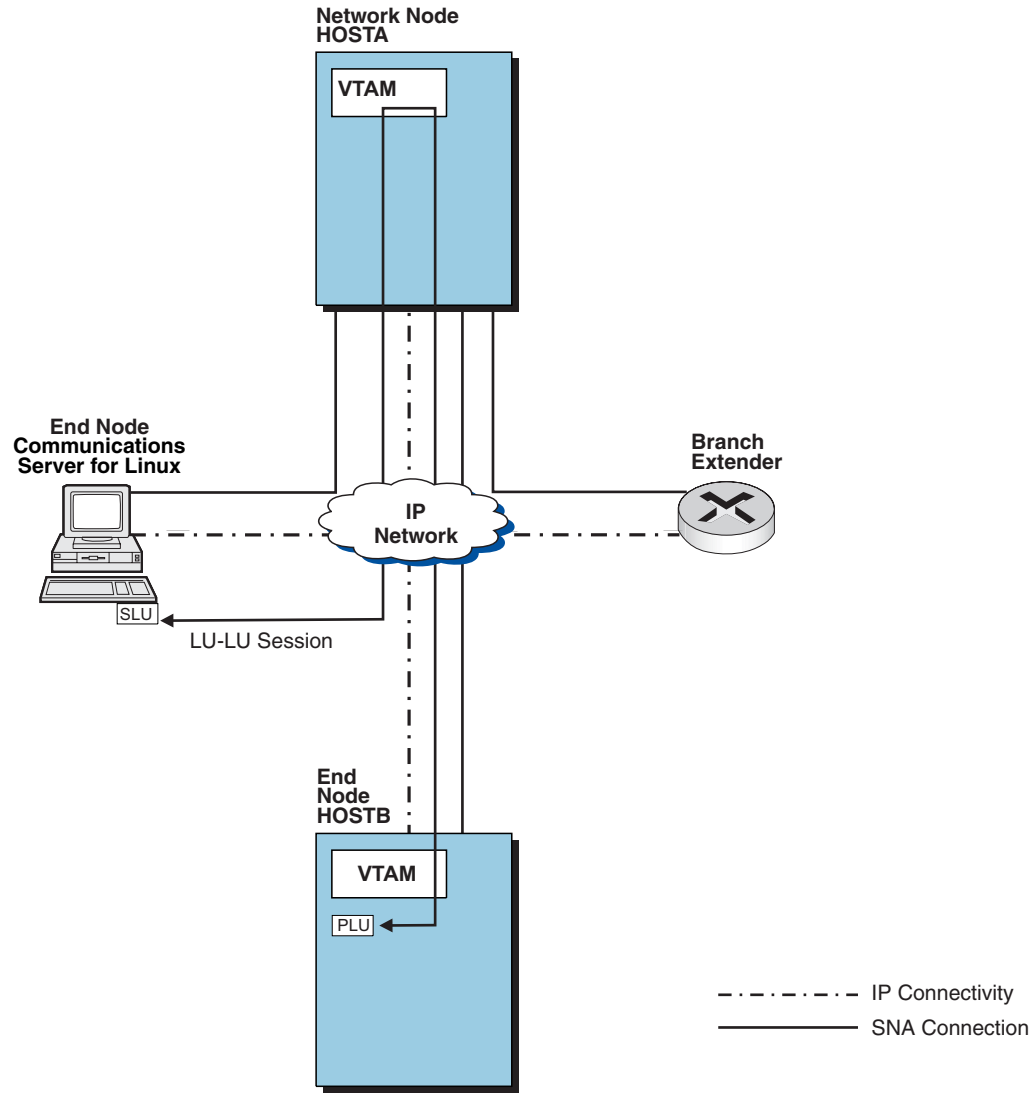


Figure 45. VTAM routing with an SATF

Optimal route calculation is achieved in one of two ways:

- Meshed connection definitions (define connections from every node to every other node)
- Connection network definition (define a connection to a connection network or virtual node)

Defining connections between each pair of nodes enables optional route calculation, but requires (n*(n-1)) definitions (where n is the number of nodes). In Figure 46 on page 163, for example, the LU-LU session no longer traverses HOSTA.

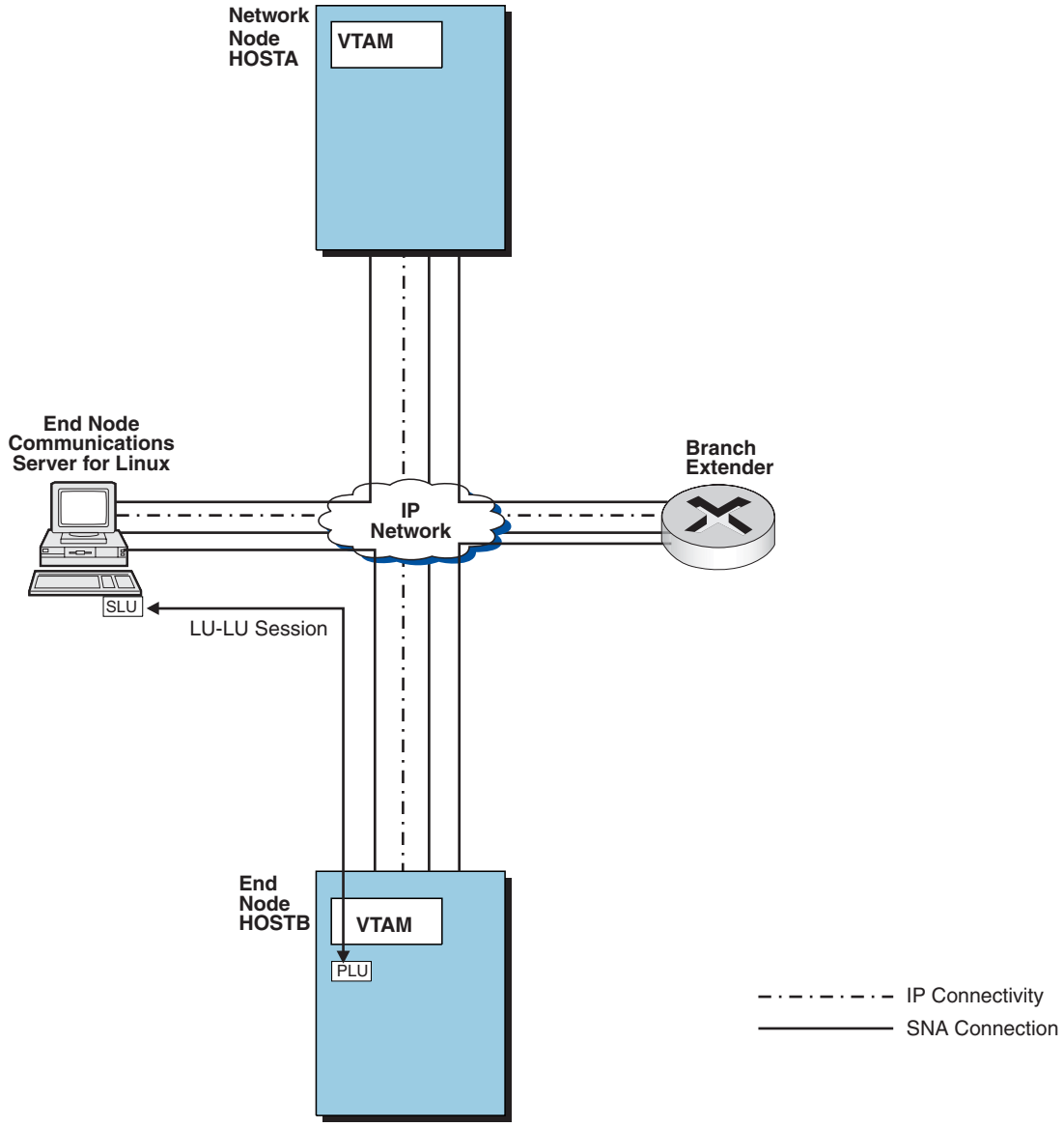


Figure 46. VTAM routing meshed connections

As the number of nodes in the network increases, the number of required connection definitions increases at an exponential rate. In a large network these definitions can be extensive. If you define a connection network to represent the shared access transport facility, system definition is reduced. With connection networks, end nodes need to define only a connection to a virtual node and to their network node server, which means only about (2*n) connection definitions. Figure 47 on page 164 shows this concept.

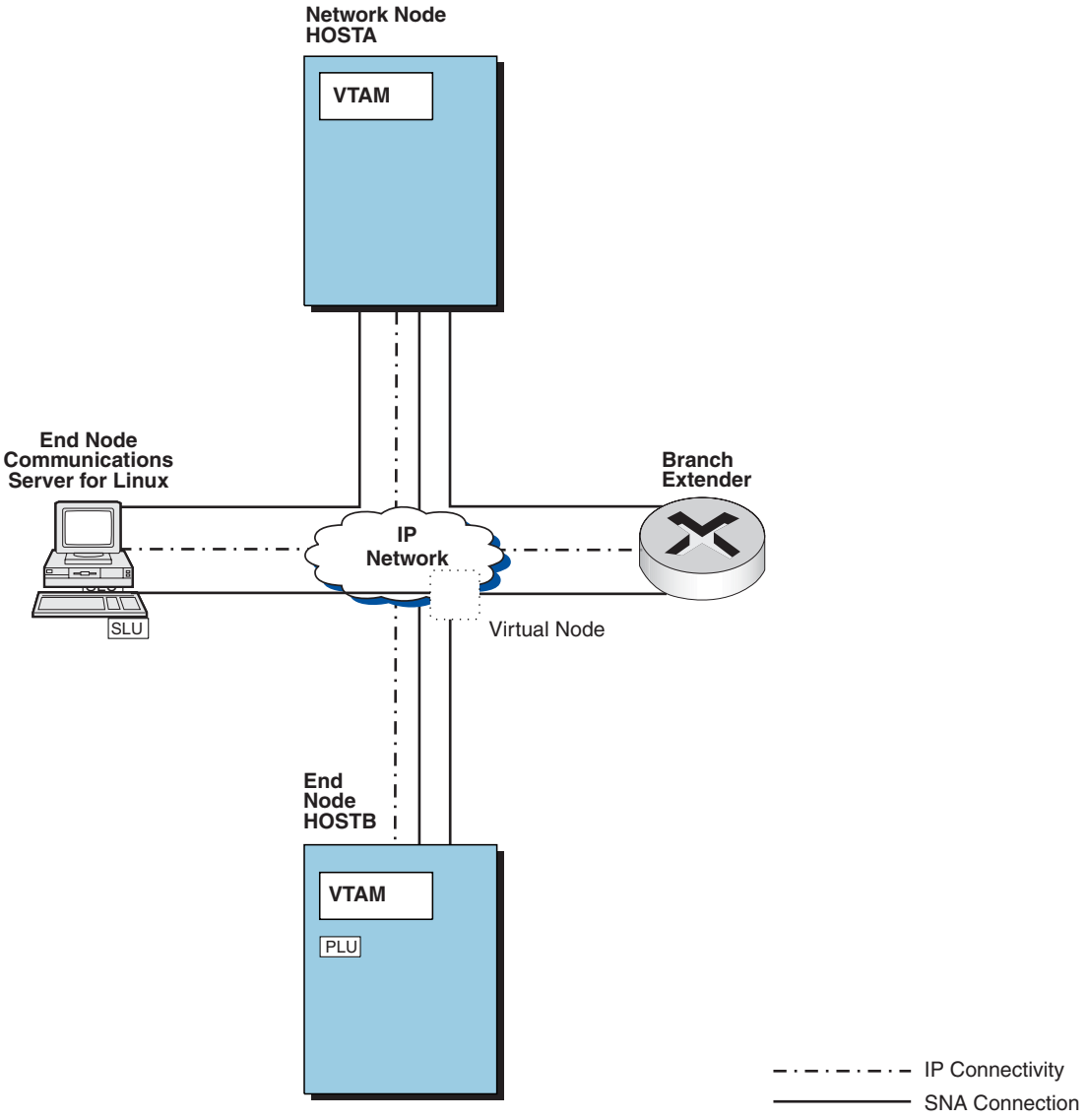


Figure 47. VTAM attachment to a connection network

The virtual node is reported to the topology database and can be chosen as the intermediate node during route calculation. This is shown in Figure 48 on page 165.

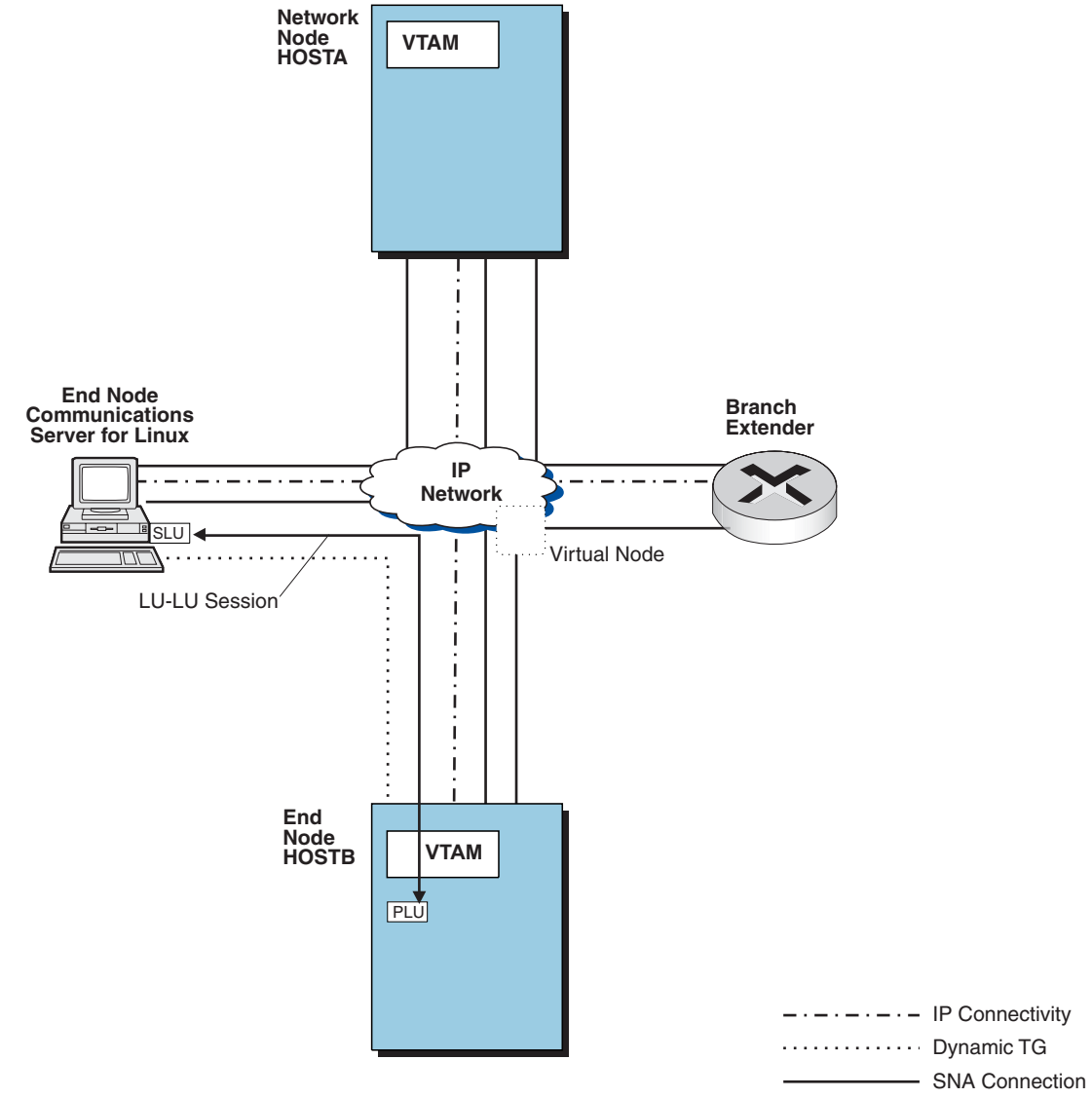


Figure 48. VTAM optimal route calculations

Connection network concepts

To understand connection networks, consider the following concepts:

- A connection network provides direct communication between all nodes on a shared access transport facility (such as an IP network) with the addition of just one definition on each node; namely, a link to the named connection network.
- A connection network is also referred to as a virtual node (VN) or a virtual routing node (VRN).
- Each EE connection network represents a single IP network of a given address family (IPv4 or IPv6) that is used for communicating between a set of EE-enabled partners.
- Connection network names are specified as fully qualified names of the form *netid.vnname*. All nodes on the same shared media that also have connection network functions can specify the same *netid.vnname*, which enables the node

computing the route to recognize that all of these nodes are on the same shared media and can bringing up dynamic links for LU-LU sessions.

- Connection network links are not used for CP-CP sessions. End nodes need to predefine a link to their network node server (and usually to their backup network node server) and can also predefine links to other nodes. The predefined link to the network node server is used for CP-CP sessions. Dynamically created connection network links are used only for LU-LU sessions and are typically brought down when the sessions end. (One exception to this is when a DYNTYPE=VN model is coded to change the setting of the DISCNT parameter for connection network PUs.) Similarly, network nodes also predefine links to adjacent network node partners for CP-CP sessions, with connection network links used only for LU-LU sessions.
- All participants on a local connection network (VNTYPE=LOCAL) must be located within the same APPN topology subnetwork. In a global connection network (VNTYPE=GLOBAL) nodes located in different APPN topology subnetworks can share the same connection network, thereby providing the capability for direct communication between the nodes defining the global VRN without requiring that all session traffic traverse the extended border nodes (EBNs) that interconnect those subnetworks. The EBNs must still be defined and participate in the session setup process, however.

EE connection network rules

- All partner nodes on an EE connection network must be able to communicate with each other over the IP WAN.
- Separate connection networks are needed for IPv4 and IPv6 traffic. If the TCP/IP stack supports both protocols, it can connect to both types of connection networks concurrently, if an IPv4 and an IPv6 local VIPA address have been defined to the stack.
- If host names are used for EE connection networks, then for consistent results, ensure that a given fully qualified host name resolves to a single source VIPA address. If not, the address returned by DNS becomes unpredictable.
- Host names are required when using IPv6 addressing. EE connection networks that use IPv6 addressing require name-to-address resolution for acquiring the source VIPA address of the local and remote EE endpoints.
- EE connection networks that use IPv4 addressing can also use name-to-address resolution for acquiring the source VIPA address of the local and remote EE endpoints, if both endpoints are z/OS V1R5 or later.

Contrasting local and global networks

If all connection network partners are located within the same APPN topology subnetwork, then you can define the connection network as a local connection network (VNTYPE=LOCAL). If some of the connection network partners are in different APPN topology subnetworks, then you must define the connection network as a global connection network (VNTYPE=GLOBAL).

Tip: Some products do not provide a choice between local and global connection networks. In general, if a product supports global connection networks but does not provide a choice of local or global (as part of connection network definition), then any connection networks that are defined are considered global.

Communication between partner nodes in different APPN topology subnetworks using a global connection network requires the assistance of extended border node

connections. Based on these considerations, begin planning your EE connection network by drawing a logical diagram of your network showing each connection network and all participating partners.

Benefits of defining multiple Enterprise Extender virtual routing nodes

The following sections describe some of the benefits that can be achieved by defining multiple local VRNs (LVRNs), global VRNs (GVRNs), or both.

Defining local and global VRNs

You can define a local and global VRN to independently represent the connectivity within your own intranet, versus the connectivity over an extranet or the Internet. This enables you to provide direct access to specific systems for external vendors, while still giving internal systems direct access to all systems within your network. Figure 49 shows a network using multiple Enterprise Extender VRNs.

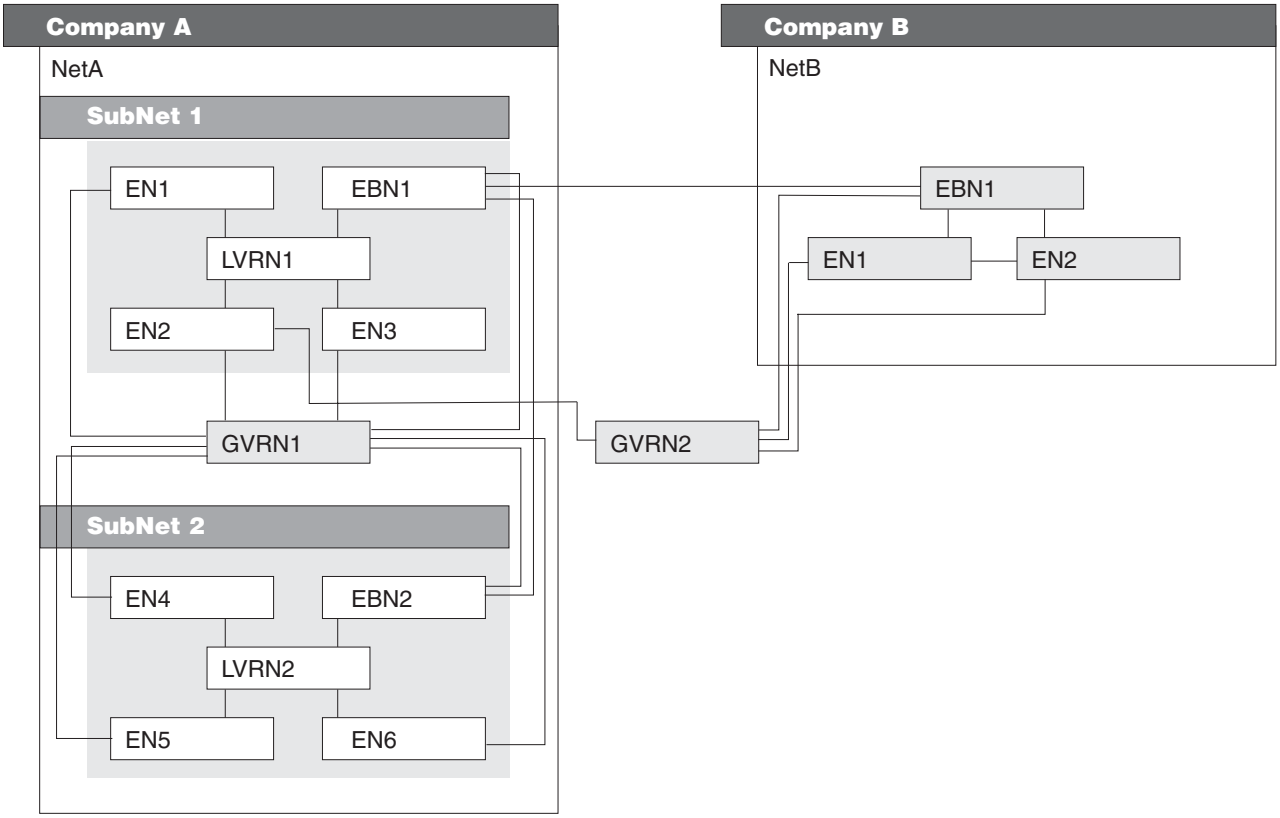


Figure 49. Defining multiple EE VRNs

In this figure, only the nodes in NetA SubNet 1 have access to the local VRN called LVRN1; and only the nodes in NetA SubNet 2 have access to LVRN2. These local VRNs can be used only to establish direct communication between nodes in the same subnetwork. But all of the nodes in both NetA SubNet 1 and NetA Subnet 2 have access to the global VRN called GVRN1. This allows every node in either of these two subnetworks to dynamically establish direct communication to any other node in either of these two subnetworks. GVRN2 is another global VRN which allows nodes in NetB to establish direct communication with only one end node (EN2) in NetA Subnet 1; sessions from NetB to any other node in NetA (SubNet 1 or SubNet 2) will include NetA.EBN1 as an intermediate node on the session path.

Defining multiple global VRNs

If a network is designed with multiple subnetworks, firewalls probably do not exist between these subnetworks. But if this same network also has APPN EBN connectivity to external vendors, a firewall probably exists between them. In this case, it would be beneficial to define two different global VRNs:

- A global VRN that is defined only by nodes within one of your own subnetworks
- A global VRN that is defined by your external vendor and a subset of the nodes within your own subnetworks

This enables you to control which systems external vendors can connect to directly (using global VRN), while still allowing internal systems (in any subnetwork) to directly connect to any of the other system in your subnetworks.

Additionally, defining multiple global VRNs enables you to avoid problems that might occur when multiple external customers connect to your network using the same global VRN. For example, assume Company A connects to Company B using a given global VRN (as shown in Figure 49 on page 167). If Company B connects to a second company using the same global VRN, it might then be possible for Company A to inadvertently connect directly to that second company. Although you can restrict this type of access by implementing the appropriate searching controls, defining a separate global VRN for connectivity to each external vendor increases your control over network access.

Using multiple local or global VRNs for EE load balancing

If you define links to multiple different VRNs (using different local IP addresses), you can choose different TG characteristics on these definitions to force sessions using different APPN COS names to flow over different VRNs (and use different local IP addresses). You can also configure the IP network to route traffic for different APPN COS names (IP ToS values) using different physical IP interfaces. For example, #INTER traffic could be sent over a Gigabit Ethernet connection and #BATCH traffic could be sent over a slower OSA link.

Using this type of configuration, you can monitor APPN traffic flows by displaying the RTPs that flow through a specific VRN. You can do this by specifying the FIRSTCP=*vrnname* operand, the FIRSTTG=*tgnumber* operand, or both on the DISPLAY RTPS command. This configuration also enables you to monitor IP traffic flows by tracing flows based on source or destination IP addresses.

If the APPN product you are using for a remote EE partner does not allow you to define multiple VRNs at the same time, then instead you can define two links to the same VRN at your z/OS system (using different local IP addresses) and force sessions to be load balanced over those connections. You can still monitor APPN and IP traffic by specifying both the FIRSTCP= and FIRSTTG= operands on the DISPLAY RTPS command in order to see what traffic is flowing over which link.

Finally, you might also want to use only a single local IP address, but still load balance sessions through parallel VRNs. With this type of configuration, it is not possible to monitor IP traffic over each of these connections, because there is only a single local IP address being used. But you would still be able to monitor APPN session traffic over the parallel routes (VRNs) by using the FIRSTCP= and FIRSTTG= operands on the DISPLAY RTPS command. Furthermore, because only a single IP address is being used, this type of configuration might simplify the coordination of system definitions.

Traversing multiple APPN network boundaries

To enable an EE connection network to traverse multiple APPN network boundaries, you can create an EE global connection network. In an EE global connection network, the following are true:

- An IP wide area network (WAN) is a shared access transport facility (SATF) represented by a global virtual routing node (GVRN).
- Nodes can indicate their connectivity to the SATF through the GVRN, thereby enabling nodes in the same or different APPN subnetworks to establish direct connectivity without having explicit routes (TGs) defined to one another.
- Part or all of a session path can traverse a common IP network (intranet or internet) and can bypass the extended border nodes (EBNs) completely.
- The computed session route does not have to traverse the adjacent border nodes (which is required when global VRNs are not used). This often provides for more optimal session routes and can remove the EBNs as a potential data bottleneck.

A GVRN can be used for cross-network EE connections if the following are true:

- The GVRN must be defined in the subnetwork of the proposed session endpoints.
- The GVRN must be defined on the endpoint node; or on the EBN that is acting as the entry point into (or exit point out of) the subnetwork of the endpoint node; or, under certain circumstances, on a network node in the subnetwork of the PLU. A GVRN defined on a network node can be used when that network node resides in the same subnetwork as the PLU and the final session route is calculated by PLU node's network node server.
- The session path is completely APPN, and every subnetwork boundary along the session setup path is an extended subnetwork boundary. (That is, there is an EBN on both sides of every subnetwork boundary.) Each of the EBNs on the session setup path must be z/OS V1R2 Communications Server or later VTAMs.

Restrictions: The following restrictions apply when an EE global connection network is used for session establishment.

- GVRNs do not allow dynamic connections to end nodes residing below a branch extender node (BrNN). Assuming the BrNN has a connection to the GVRN, the BrNN is the concentration point for the EE global connection network for all nodes residing downstream of the BrNN.
- A GVRN is not used in intermediate subnetworks along a session setup path (except, possibly, as a local VRN).
- In most cases, a GVRN is not used if either session endpoint resides in or the session path traverses a subarea network (through an interchange node). However, the use of a DSME in intermediate border nodes might allow using a GVRN when the SLU resides in or the session path traverses a subarea network.
- GVRNs are not used for session setup requests that traverse APPN subnetwork boundaries defined with the value RTPONLY=YES. (The value RTPONLY=YES enables border nodes to maintain awareness of all sessions established over the specified APPN subnetwork boundaries; but using a GVRNs instead could result in an EBN not maintaining awareness of one or more sessions, which would defeat the purpose of RTPONLY=YES.) For more information on the RTPONLY operand, see RTPONLY operand in *z/OS Communications Server: SNA Resource Definition Reference*.

Rules for identifying the source of the local IP address associated with a connection network:

- Each connection network partner must identify the IP address associated with its endpoint on the connection network. Specify either the host name or the IP address on the group definition where the connection network is defined.
- If the connection network represents an IPv6 network, then the connection network endpoint address must be specified using the HOSTNAME parameter.
- If the connection network represents an IPv4 network, then the endpoint can be defined by either the IP address parameter or the HOSTNAME parameter; however, the HOSTNAME parameter provides maximum flexibility because it provides compatibility with networks that use network address translation (NAT).

Defining an EE connection network in the EE XCA major node

Perform the following steps to configure the EE XCA major node to include the definition of a connection network.

1. Use the PORT and GROUP definition statements to define a connection to a connection network virtual routing node. Following is a sample Enterprise Extender XCA major node GROUP definition for a local VRN:

```
GRPEE1   GROUP  DIAL=YES,                        (a)
                CALL=INOUT,                     (b)
                HOSTNAME=NETA.HOST1,            (c)
                VNNAME=NETA.LCLVRN,             (d)
                VNTYPE=LOCAL,                   (e)
                CAPACITY=1000M,                 (f)
                AUTOGEN=(10,L,P)                (g)
```

   Perform the following steps to define the appropriate operands:

   a. Code DIAL=YES on the GROUP statement that defines the virtual node or on the GROUP statement named on the VNGROUP operand of the PORT definition statement.

   b. Code the CALL operand to indicate that lines are available for both dial out and dial in connections.

   c. Code the HOSTNAME operand to specify the host name to be resolved to obtain this static VIPA address. Alternatively, code the IPADDR operand to specify the static VIPA address that Enterprise Extender connections will use to communicate with this host. Code the IPADDR or HOSTNAME on the GROUP definition statement, or it will default from the similarly named VTAM start option.

      Tips:

      - If your environment requires multiple source VIPA addresses, or if your EE connections require different addressing protocols (IPv4 versus IPv6), code the IPADDR operand (for IPv4 protocols) or HOSTNAME (for IPv4 or IPv6 protocols) on the GROUP statement. The IPADDR operand explicitly identifies the source VIPA address to be used for this EE connection. The HOSTNAME operand specifies the name to use for name-to-address resolution, at this node and at remote EE endpoints connected to this same VRN, for obtaining the source VIPA address representing this stack.
      - If your environment requires unique LDLC inactivity timer settings for different EE connections, then you can specify different LIVTIME, SRQTIME, and SRQRETRY values on the GROUP definition statements (for each unique local IP address).

   d. Code the VNNAME operand to define the name of the virtual node. IPv4 or IPv6 protocols can be used for both local or global VRN definitions. Supply a different virtual node name (VNNAME) for connection networks using different addressing protocols.

e. Code the VNTYPE operand to indicate whether the connection network is allowed to span APPN subnetwork boundaries. The default for VNTYPE is LOCAL (the connection network cannot span subnetwork boundaries). If the connection network should be allowed to span subnetwork boundaries, specify a VNTYPE of GLOBAL. (A VRN defined with VNTYPE=GLOBAL can also be used locally.) Attempts to define the same VRN as GLOBAL in some nodes and LOCAL in others might not produce the results you want. If a VRN is to be used across subnetwork boundaries, then define it as VNTYPE=GLOBAL in all nodes.

f. Use the CAPACITY operand to specify the media speed for the connectivity to the partner. The previous example specifies 1000 Mb for a gigabit Ethernet infrastructure. See "Advanced coding considerations for EE" on page 184 for more detail on specifying connection (TG) characteristics.

g. Coding AUTOGEN=(10,L,P) will automatically generate 10 lines and PUs for the group. The number of lines and PUs defined should be greater than or equal to the number of expected concurrent EE partners.

2. Define at least as many lines as the maximum expected concurrent connections. Because lines are associated with a specific GROUP statement, make sure that there are enough lines for that specific GROUP. For instance, a GROUP statement could be used for predefined EE connections (used for actual dialed EE connections) or for an EE connection network (either global or local). Make sure that there are enough EE lines for all GROUP statements defined in the XCA major node. These lines can be created manually or with AUTOGEN.
3. Define as many local or global connection networks as your configuration requires. By allowing multiple global and local virtual routing nodes, users can define VRNs based on link characteristics. For example, a subset of users might require secure links, while others might use unsecured links. Depending on the requirements of the sessions, users can connect to the z/OS network using the appropriate VRN for the session characteristics. Definition of multiple local or global connection networks is necessary if both IPv4 and IPv6 protocols are to be used for local or global VRNs.
4. To maximize availability, consider enabling the EE connection network reachability awareness function.
5. Activate the XCA major node if it is not already activated. Also activate the group by issuing VARY ACT, ID=GRPEE1.

You can verify that you have successfully defined a connection network in the EE XCA major node by looking for message:

```
IST1168I VIRTUAL NODE NETA.LCLVRN CONNECTION ACTIVE
```

which indicates that the connection network link is now active.

The following guidelines and restrictions apply.

Guidelines:

- When a VNTYPE of GLOBAL is specified and a VNNAME value is not specified, the virtual node is assigned the name IP.IP.
- The VNNAME/IPADDR pair (whether explicitly coded as an IP address using the IPADDR operand, or resolved from a host name using the HOSTNAME operand) must be unique. For example, if VRN1/HOST1 and VRN1/HOST2 are defined and if HOST1 and HOST2 resolve to the same IP address, then the first connection to be activated will succeed, but activation of the second connection will fail because these two definitions are not considered unique.

- You can define local connection networks (used for sessions within a single APPN topology subnetwork) and global connection networks (used for sessions that span APPN subnetwork boundaries). You can define as many local or global virtual nodes as your configuration requires. When a virtual node is defined on the PORT definition statement, the GROUP statement referenced by VNGROUP cannot also define a virtual node.

Restrictions:

- A local VRN cannot use the same name as a global VRN. A different VRN name (VNNAME) must be supplied for connection networks using different VNTYPEs.
- If CP-CP sessions over EE are required between two nodes on the shared access transport facility, the dialing-out node must define the PU for every node that it will call. Routes traversing a virtual node cannot be used for CP-CP sessions.
- The dial-out node can use only a dynamically defined PU for connectivity through a connection network. The dial-in node attempts to use a predefined PU, if one exists. Otherwise, it uses a dynamically defined PU or a PU allocated using the configuration services XID exit.
- When a virtual node is defined on a GROUP definition statement, the TG characteristics for the link to that virtual node must also be specified on that GROUP definition statement.
- When a virtual node is defined on a PORT definition statement, the TG characteristics for the link to that virtual node must also be specified on the PORT definition statement.

EE security considerations

There are several techniques to consider when evaluating EE security options:

- SNA session level encryption (SLE)
- IPSec
- Network Address Translation (NAT)
- OEM security products

SNA session level encryption (SLE)

You can use SNA session-level encryption in the same way that it is currently used in SNA networks. In this way, the application can be assured that the traffic is received from the sender and that it has not been modified. This provides a reliable transmission method that is useful if sections of the network remain SNA (non-EE) and the data being transmitted is especially sensitive in nature. If the only section of the network that is unsecured is the IP network, or if authentication is the only requirement, other methods might work better for your needs.

Requirement: SLE requires the sharing of encryption keys. Ensure that you have a trusted partner.

IP security (IPSec)

IPSec is an industry-standard protocol that provides end-to-end authentication and encryption. IPSec provides an excellent method of securing EE connections. z/OS itself can be an IPSec endpoint or IP security can be offloaded to an attached router platform.

- By placing the IPSec endpoint on z/OS, you have end-to-end protection but your zSeries CPU will incur the cost of the encryption.

  - Additional assistance for IPSec protocol traffic is available with any IBM System z® Integrated Information Processor (IBM zIIP). You might need to modify some of your Workload Manager (WLM) definitions when you use zIIPs for IPSec-enabled EE connections. See *z/OS Communications Server: IP Configuration Guide* for information about modifying WLM definitions for zIIPs.
- By offloading the IPSec function to a router, you offload the encryption cost but you have an unprotected segment between z/OS and the router hosting the IPSec endpoint.

Providing authentication

If you need to provide authentication, you can use IPSec, which includes an authentication scheme. You can use IPSec authentication in one of three ways:

- Stand-alone
- In combination with IPSec encryption
- In combination with SNA session-level encryption

Using IPSec's authentication scheme lets the system performing authentication verify that the data is received from the partner and has remained unchanged during transmission. When you use IPSec's authentication scheme in combination with encryption, you can authenticate the partner before wasting extra cycles for data decryption.

Tip: SNA also offers Message Authentication Code (MAC), but it does not authenticate the IP header portion.

Providing encryption

If the IP network is the only unsecured section, you can use IPSec between the two EE nodes to ensure that the transmitted data is not modified or viewed along the path.

- You can use IPSec between firewalls if there is a secure intranet and an unsecured Internet portion of the session path.
- You can run IPSec on the host to establish a VPN.
- If the EE nodes are the session partners, you can use either SNA session-level encryption or IPSec to encrypt the data.

The most significant difference between IPSec and SLE is that IPSec encrypts part of the UDP header, but SNA session-level encryption does not. See *z/OS Communications Server: SNA Resource Definition Reference* for specifics about session-level encryption.

Tip: The SNA header is encrypted only if IPSec is used.

If you use SNA encryption, use the filtering rule on the EE UDP port to allow traffic to flow without subsequent IPSec encryption. You can also use a combination of SNA encryption and IPSec authentication, where IPSec authentication is designed using filter rules on the same EE UDP port.

For more information about IPSec and IP filtering, see *z/OS Communications Server: IP Configuration Guide*.

Using EE with network address translation (NAT)

Network address translation (NAT) is a technique where a one-to-one address translation function is performed, translating a single internal IP address to a

single public IP address. NAT is a broad term that encompasses both a one-to-one address translation function, and network address/port translation (NAPT)

An internal-external IP address mapping is maintained by the NAT device. IP addresses are translated, but ports are unchanged. The mapping can be static or dynamic. For a static mapping, there is a definition in the NAT that always translates IP address x.x.x.x to IP address y.y.y.y. For a dynamic mapping, the NAT has a pool of IP addresses that are assigned as needed, so IP address x.x.x.x might be mapped to IP address y.y.y.y one time, and to IP address z.z.z.z at another time.

Restriction: Because EE requires a unique IP address mapping, dynamic network address translation is generally incompatible.

The following considerations apply when using EE on IP networks implementing network address translation:

- For predefined EE connections, ensure that the remote IP address is the public address translated or mapped by the NAT device.
- To enable EE connection networks to coexist with NAT, define the connection network using the HOSTNAME operand, not the IPADDR operand.

Network address/port translation (NAPT)

NAPT is a technique in which multiple internal IP addresses are translated into a single public IP address. As part of this translation process, the TCP and UDP ports in the packets are translated.

Restriction: Because EE has dependencies on port assignments, and a one-to-one IP address mapping, NAPT is incompatible with EE.

IP filtering

Use IP filtering to control the flow of network traffic. An IP security policy can define filters that deny or allow a packet access to a z/OS Communications Server system. Enterprise Extender (EE) IP filters use information in the IP and UDP packet headers to either permit or deny traffic coming into the z/OS CS enterprise. You can write an EE filter rule to allow inbound UDP data from a set of IP addresses to use the defined EE ports. You can write a second rule to allow outbound EE data access to a set of predefined IP addresses. You can also set up the policy to deny any traffic that does not match the previous rules. IP filtering rules can be written to log messages if a packet is denied because the packet did not match the rule. See *z/OS Communications Server: IP Configuration Guide* for more information about IP filtering.

IDS for Enterprise Extender

Use intrusion detection services (IDS) to protect z/OS Communications Server from attacks. You can use IDS to detect patterns that might indicate an impending attack. IDS supports attack rules that define the type of behavior to monitor. Define a set of attack rules to specify what events to monitor and the actions to take if an attack is detected. The action associated with the rule lets you specify what type of logging is to be performed and the action that the system should perform when an attack is detected.

Use the syslog daemon (syslogd) to log the attack. Syslogd is an application that writes messages to an OMVS file. You can set a statistics interval. At the end of each statistics interval, syslogd logs a message. The attack rule can write the detected packet to the IDS trace SYSTCPIS. The action can be defined to either discard or not discard the packet. You can code an exclusion list for each

Enterprise Extender (EE) IDS attack type. Specific IP addresses in the exclusion list that would have been detected as an attack continue to be accepted to preserve existing behavior.

See *z/OS Communications Server: IP Configuration Guide* for more IDS information.

IDS defines the following new attack types for EE.

EE Malformed Packet
Specify this attack type to allow IDS to check inbound EE packets to determine whether the packet is malformed.

EE Port Check
Specify this attack type to allow IDS to verify the port value of inbound EE packets. If the port values are not correct, IDS flags the packet as a port check attack.

EE LDLC Check
Specify this attack type to allow IDS to verify EE LDLC commands. All EE data is sent by using LDLC commands. IDS checks the LDLC type to verify that the packet is received on the correct port.

EE XID Flood
Specify this attack type to allow IDS to detect suspicious XID activities and EE XID timeouts. EE XID timeouts might lead to an EE XID flood. When IDS detects an XID timeout, the following series of events occurs:

The local EE endpoint resends the XID reply three times before it fails the activation request and issues a timeout message. An XID flood occurs when the number of inbound XID timeouts is equal to the value of the EEXIDTimeout value in the IDS XID flood attack rule. Each inbound activation XID that is received by VTAM is assigned an available line for the connection. A partner EE that sends an XID and that does not continue activation of the connection will occupy an available line for about 1 minute. A flood of these occurrences can quickly use all available lines. This kind of attack is a denial of service attack. Valid XIDs will fail because a line is not available for the connection request.

The EEXIDTimeout value defines a threshold value that specifies the number of XIDs that can time out in 1 minute before IDS detects an EE XID flood. When IDS detects a flood, TCPIP writes a message to the console and to an OMVS file using syslogd (if such a message is required). The XID flood ends when the number of XIDs that are received in 1 minute is below the threshold value. You can enable statistics logging to assist in determining a threshold value. The statistics provide the number of XID timeouts that occurred during the interval and the maximum number of timeouts that occurred in any minute during the interval.

Note: The EE XID flood attack type does not support packet discard. It also does not support writing the packet to the IDS trace, SYSTCPIS.

OEM security products — EE proxy solutions

Because EE is based on UDP, firewalls in the underlying network must permit UDP traffic on the EE ports (12000 - 12004). For corporations with policies that absolutely forbid allowing UDP packets through the firewall, it is possible to proxy the UDP traffic onto TCP connections. Although IBM does not currently provide a TCP proxy for EE, other vendors might provide such a solution.

An additional advantage of using an EE proxy is that the EE traffic then becomes eligible for SSL encryption (SSL is a TCP protocol.) The disadvantage of an EE proxy (with or without SSL) is the additional CPU overhead incurred for the TCP transport.

Tuning the EE network

This section describes several ways to tune your EE network to improve performance.

Tuning Enterprise Extender-specific buffer pools

Enterprise Extender performance degradation can be caused by poorly tuned buffers. Monitor and tune the T1BUF and T2BUF buffer pools to minimize the number of expansions. Minimizing buffer pool expansions decreases internal buffer overhead processing, which should increase throughput while reducing CPU consumption. Also, the T1BUF and T2BUF pools are used instead of the TIBUF buffer pool. Increasing the buffer pool values for the T1BUF, T2BUF, or both might decrease the buffers necessary for the TIBUF pool. You should also monitor and tune the TIBUF pool.

Use IBM Health Checker for z/OS to check whether the T1BUF and T2BUF start option definitions for the initial number of buffers in the pool are greater than the default value when Enterprise Extender (EE) is not being used. You can also use Health Checker to check whether the initial number of buffers specified for these two buffer pools is equal to the default value when EE is being used with QDIO or HiperSockets. For more details about IBM Health Checker for z/OS, see *z/OS Communications Server: IP Diagnosis Guide*.

The following is a buffer pool display example.

```
d net,bfruse,buf=(ti,t1,t2)
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = BUFFER POOL DATA
IST920I TI00     BUFF SIZE  632         EXP INCREMENT   60
IST921I          TIMES EXP  0           EXP/CONT THRESH 120   / *NA*
IST922I          CURR TOTAL 360         CURR AVAILABLE  85
IST923I          MAX TOTAL  360         MAX USED        302
IST924I ------------------------------------------------------------
IST920I T100     BUFF SIZE  1004        EXP INCREMENT   32
IST921I          TIMES EXP  3           EXP/CONT THRESH 15    / 111
IST922I          CURR TOTAL 112         CURR AVAILABLE  112
IST923I          MAX TOTAL  112         MAX USED        69
IST924I ------------------------------------------------------------
IST920I T200     BUFF SIZE  2028        EXP INCREMENT   32
IST921I          TIMES EXP 406920       EXP/CONT THRESH 7     / 103
IST922I          CURR TOTAL 104         CURR AVAILABLE  104
IST923I          MAX TOTAL  1608        MAX USED        1576
IST314I END
```

In this example, the T2BUF (T200) buffer pool expanded 406,920 times. At any given time, the maximum number of T2BUF buffers used at one time was 1576. This example indicates that the T2BUF buffer pool is thrashing, constantly expanding and contracting to meet the needs of its environment. A large number of expansions indicates that buffer pool tuning is warranted. In this case, coding a base allocation of 1576 buffers or more will eliminate the buffer pool thrashing for the T2BUF buffer pool. Use a similar approach when tuning your T1BUF buffers. In this example, the T1BUF pool does not need tuning (three expansions is considered minor).

Timers

EE and HPR depend on several different timers to determine when EE connections and RTP pipes should be terminated. There are timers for the RTP pipe itself, the EE switched PU, and at the logical data link control layer beneath the switched PU.

When does the RTP pipe go away?

With z/OS Communications Server, the RTP endpoint is represented by a dynamic PU, and acts like a delayed disconnect PU. This PU is dynamically given the name CNR*xxxxx*, where *xxxxx* represents a number incrementing from 00001 as the dynamic PUs are created. When the last session using the RTP terminates, the RTP goes inactive if no new session is queued to it before the disconnect time expires. You can alter the disconnect time using an operand on the DISCNT keyword on a model PU in a model major node (DYNTYPE=RTP indicates the model for HPR PUs). If you code a model RTP PU, code CPCP=YES if you want your CPCP sessions to traverse HPR pipes.

Guideline: Avoid using DISCNT=YES or a disconnect time of less than 10 seconds for DYNTYPE=RTP model PUs.

In the following example, 60 seconds is shown in the model major node definition for RTP PUs. Both RTP endpoints need to have similar types of definition. If dissimilar, the RTP endpoint with the lowest disconnect time would be the one deactivating the RTP, regardless of the (higher) value coded at the partner node.

```
MODMAJND VBUILD  TYPE=MODEL
RTPPUMOD    PU   DYNTYPE=RTP,DISCNT=(DELAY,,60),CPCP=YES
```

When does the EE connection go away?

Because the EE connection is itself represented by a switched PU, you can code the DISCNT operand to disconnect the EE switched PU. If DISCNT=YES is coded, the disconnect time used is that specified on the VTAM start option DISCNTIM (default is 15 seconds). Alternatively, you can specify DISCNT=(DELAY,,*s*) on an EE PU where *s* is equal to the number of seconds to be used for the disconnect time.

Because the EE PU (unlike RTP PUs) has no sessions queued directly to it, the disconnect decision is based on a period of inactivity rather than the termination of the last session.

If RTP's ALIVE timer is set to half of the disconnect time, RTP status messages (keep-alive flows) will keep the PU from dropping as long as at least one RTP is using the EE PU.

The Logical Data Link Control (LDLC) layer monitors the EE connection, and terminates the EE connection if contact is lost with the partner. The LDLC inactivity trigger is controlled by three parameters on the PORT or GROUP statements:

- LIVTIME - The amount of inactivity time that can lapse before LDLC tests the connection
- SRQTIME - The amount of time LDLC waits for a response to its test
- SRQRETRY - The number of times the test is retried

The connection is terminated if no activity or response occurs when using the following formula:

LIVTIME + ((SRQRETRY+1) * SRQTIME) seconds

This is an approximate formula. The actual duration of the LDLC layer outage detection window should be within plus or minus one LIVTIME interval of the calculated amount.

Guidelines:

- Lengthen the HPR path switch timers (HPRPST) as necessary to ensure that all four timers are longer than the LDLC timeout interval. This ensures that RTP pipes stay in path switch long enough during IP network instability for the EE link to disconnect, and enables another path to be selected.
  - Alternatively, specify HPRPSDLY=EEDELAY on the definition statements that represent EE connections. The HPRPSDLY parameter is available on the PU definition statement in the switched and model (DYNTYPE=EE) major nodes, and also on the connection network GROUP definition statements in the EE XCA major node. For more information about the HPRPSDLY parameter, see *z/OS Communications Server: SNA Resource Definition Reference*.
- When possible, ensure that the values of the LDLC parameters are consistent between the endpoints of the connection. If the EE partner is not a z/OS endpoint, see that product's documentation for details on setting its LDLC parameters.
- For predefined EE PUs, specify DISCNT=NO.

Tip: The LIVTIME, SRQTIME, and SRQRETRY parameters are customer-configurable parameters that control the ability of the LDLC-layer to detect TCP/IP network connectivity problems; these parameters should not be confused with the similarly-named variables associated with HPR's Adaptive Rate-Based congestion control (ARB) algorithm that is used to monitor each RTP connection. Each RTP has associated with it a liveness timer, a short request timer, and a short request retry counter. However, the RTP values are algorithmically maintained, are not configurable, and are independent of the LDLC parameters. (It is possible to influence the RTP's liveness timer value by coding DISCNT=YES for the connections used by the RTP.)

See the SRQRETRY description in *z/OS Communications Server: SNA Resource Definition Reference* for more information.

Inactivity timer example 1

Consider the following time-annotated console:

```
d net,rtps
    ...
IST1695I PU NAME       CP NAME      COS NAME SWITCH CONGEST  SESSIONS
IST1696I CNR00004 NETA.SSCP2A       #INTER     NO      NO           1
IST1696I CNR00003 NETA.SSCP2A       RSETUP     NO      NO           0
    ...
(11:43:32) Break the IP connectivity.
v tcpip,,stop,trle1a
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,STOP,TRLE1A
EZZ0053I COMMAND VARY STOP COMPLETED SUCCESSFULLY
EZZ4315I DEACTIVATION COMPLETE FOR DEVICE TRLE1A
(11:44:14) RTP CNR00004 detects failure and goes into path switch.
IST1494I PATH SWITCH STARTED FOR RTP CNR00004
(11:44:49) The LDLC layer detects loss of IP connectivity and
           disconnects the EE connection.
IST1411I INOP GENERATED FOR LNIP1
IST1430I REASON FOR INOP IS XID OR LDLC COMMAND TIMEOUT
IST314I END
IST259I INOP RECEIVED FOR SWIP2A1 CODE = 01
```

```
IST619I ID = SWIP2A1 FAILED - RECOVERY IN PROGRESS
IST1196I APPN CONNECTION FOR NETA.SSCP2A INACTIVE - TGN = 21
IST590I CONNECTION TERMINATED FOR PU SWIP2A1 ON LINE LNIP1
IST621I RECOVERY SUCCESSFUL FOR NETWORK RESOURCE SWIP2A1
IST1488I INACTIVATION FOR RTP CNR00003 AS PASSIVE PARTNER COMPLETED
IST619I ID = CNR00003 FAILED - RECOVERY IN PROGRESS
IST129I UNRECOVERABLE OR FORCED ERROR ON NODE CNR00003 - VARY INACT SCHED
IST105I CNR00003 NODE NOW INACTIVE
IST871I RESOURCE CNR00003 DELETED
(11:46:14) CNR00004's path switch timer (2 min for interactive TP) expires.
           CNR00004 goes inactive.
IST1494I PATH SWITCH FAILED FOR RTP CNR00004
IST1495I NO ALTERNATE ROUTE AVAILABLE
IST314I END
IST1488I INACTIVATION FOR RTP CNR00004 AS ACTIVE PARTNER COMPLETED
IST619I ID = CNR00004 FAILED - RECOVERY IN PROGRESS
IST129I UNRECOVERABLE OR FORCED ERROR ON NODE CNR00004 - VARY INACT SCHED
IST105I CNR00004 NODE NOW INACTIVE
IST871I RESOURCE CNR00004 DELETED
```

This is an example of how the various parameters influence the termination of an HPR connection using an EE TG. In this example, the RTP's disconnect time is the VTAM start option's DISCNTIM default, 15 seconds. The keyword DISCNT=YES is coded on the Enterprise Extender PU in its switched major node and, in the XCA major node on the PORT statement, LIVTIME is set to 10, SRQTIME to 15, and SRQRETRY to 3.

Based on the LDLC parameters, the LDLC layer should detect an outage and disconnect the connection in 70 seconds, plus or minus one SRQTIME interval. This results from the following equation:

```
10 + ((3+1) * 15) = 70 seconds
```

With DISCNTIM set to 15 seconds, RTP connections using the EE connection will have a liveness timer value of 7.5 seconds (half the DISCNTIM value), which means that they will be likely to detect the outage faster than the LDLC layer.

As shown in the example, at 11:43:32 the IP TRLE is taken down and after about 40 seconds the RTP connection has detected the failure and has started a path switch. The LDLC layer detects the failure at 11:44:49, 77 seconds (close to the predicted value of 70 seconds) after the outage, and ends the EE connection. With no alternate path available, all path switch attempts fail, and the RTP pipe (CNR00004) fails at 11:46:14 (2 minutes after it went into path switch, based on the path switch timer value for interactive transmission priorities as set by HPRPST).

Inactivity timer example 2

Consider the following time-annotated console:

```
d net,rtps
    ...
IST1695I PU NAME       CP NAME      COS NAME SWITCH CONGEST  SESSIONS
IST1696I CNR00004 NETA.SSCP2A       #INTER     NO      NO           1
IST1696I CNR00003 NETA.SSCP2A       RSETUP     NO      NO           0
    ...
(11:34:56) Break the IP connectivity.
v tcpip,,stop,trle1a
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,STOP,TRLE1A
EZZ0053I COMMAND VARY STOP COMPLETED SUCCESSFULLY
EZZ4315I DEACTIVATION COMPLETE FOR DEVICE TRLE1A
(11:36:23) The LDLC layer detects loss of IP connectivity and
           disconnects EE connection.
IST1411I INOP GENERATED FOR LNIP1
IST1430I REASON FOR INOP IS XID OR LDLC COMMAND TIMEOUT
```

```
IST314I END
IST259I INOP RECEIVED FOR SWIP2A1 CODE = 01
IST619I ID = SWIP2A1 FAILED - RECOVERY IN PROGRESS
IST1196I APPN CONNECTION FOR NETA.SSCP2A INACTIVE - TGN = 21
IST590I CONNECTION TERMINATED FOR PU SWIP2A1 ON LINE LNIP1
IST1494I PATH SWITCH STARTED FOR RTP CNR00004
IST621I RECOVERY SUCCESSFUL FOR NETWORK RESOURCE SWIP2A1
IST1488I INACTIVATION FOR RTP CNR00003 AS PASSIVE PARTNER COMPLETED
IST619I ID = CNR00003 FAILED - RECOVERY IN PROGRESS
IST129I UNRECOVERABLE OR FORCED ERROR ON NODE CNR00003 - VARY INACT SCHED
IST105I CNR00003 NODE NOW INACTIVE
IST871I RESOURCE CNR00003 DELETED
(11:38:23) CNR00004's path switch timer (2 min for interactive TP) expires.
           CNR00004 goes inactive.
IST1494I PATH SWITCH FAILED FOR RTP CNR00004
IST1495I NO ALTERNATE ROUTE AVAILABLE
IST314I END
IST1488I INACTIVATION FOR RTP CNR00004 AS ACTIVE PARTNER COMPLETED
IST619I ID = CNR00004 FAILED - RECOVERY IN PROGRESS
IST129I UNRECOVERABLE OR FORCED ERROR ON NODE CNR00004 - VARY INACT SCHED
IST105I CNR00004 NODE NOW INACTIVE
IST871I RESOURCE CNR00004 DELETED
```

In this example, the LDLC parameters (LIVTIME, SRQTIME, and SRQRETRY) yield a longer disconnect detection time of 95 seconds, as shown in the following equation. The keyword DISCNT=YES is coded on the Enterprise Extender PU in its switched major node and, in the XCA major node on the PORT statement, LIVTIME is set to 20, SRQTIME to 15, and SRQRETRY to 4.

```
20 + ((4+1) * 15) = 95 seconds
```

DISCNTIM is also much longer (240 seconds) than in the previous example, giving an RTP liveness timer value of 120 seconds.

In this example, the LDLC layer detects the outage first, generating an inoperative indication 87 seconds (again, close to the predicted value) after the IP connectivity is broken. In this case, the local link's inoperative transition forces the RTP connection CNR00004 to go immediately into path switch as can be seen by the IST1494I message several lines after the inoperative indication occurs at 11:36:23. Two minutes later (again based on the HPRPST value for interactive TP), at 11:38:23, the path switch timer expires and the RTP connection is terminated.

HPR ALIVE timer optimization for Enterprise Extender

In addition to the overhead of the timer maintenance itself, the I/O cost of sending keep-alive status messages and the processing of the subsequent replies has a significant impact on CPU utilization when a large number (thousands) of RTP pipes are active. You can use the HPREELIV start option to reduce the CPU impact.

The EE LDLC layer monitors the health of the TCP/IP network underlying the EE connection by monitoring for activity and sending test frames to verify partner reachability. In the event that the partner can no longer be reached, the LDLC layer generates a disconnection for the EE logical connection. For products supporting the architectural option of local link disconnection notifications to drive HPR path switch processing, this means that RTP pipes using EE connections will immediately enter path switch processing if the EE LDLC layer triggers an EE disconnection. Therefore, for RTP pipes that are composed of single EE connections (and including those pipes where the RSCV describes a two-hop path across an EE

connection network), HPR ALIVE timer processing is superfluous. As long as the EE LDLC layer has not detected a problem, the RTP partner is considered reachable.

If HPREELIV is set to or defaults to YES, then:

- An RTP pipe with a route comprised of a single EE link (including a single hop across a two-hop EE VRN path) can run without an ALIVE timer, thereby relying on the EE LDLC layer for inactivity monitoring, and saving the timer maintenance and keep-alive status request and reply overhead.
- RTP pipes comprised of multi-hop paths, or single non-EE hops use ALIVE timer processing as architected.
- An RTP that path switches between one of these types of paths (for example, from single-hop EE to multi-hop) has to dynamically enable and disable the ALIVE timer at the time of the path switch.

There is CPU savings for configurations using HPR-over-one hop Enterprise Extender connections, or two hop Enterprise Extender virtual routing node (VRN) connections. When HPR-over-EE configurations specify DISCNT=NO, significant CPU savings can be achieved. Specifying DISCNT=NO means that HPR and EE connections always stay up, even when there are no sessions.

Enterprise Extender LDLC keep-alive reduction

During periods of relatively low network activity, configurations with many EE connections can consume a substantial amount of CPU resources due to the EE LDLC keep-alive function. In addition to the overhead of the timer maintenance itself, the I/O cost of sending keep-alive messages (TEST request) and the processing of the subsequent responses has a significant impact on CPU utilization when large numbers of EE connections are active.

The Enterprise Extender LDLC keep-alive reduction function assists in reducing the total CPU impact of Enterprise Extender LDLC keep-alive processing overhead. You can specify an initial and a maximum value (through the max_value parameter) for the logical data link control (LDLC) liveness timer interval, using the LIVTIME operand (see LIVTIME operand in *z/OS Communications Server: SNA Resource Definition Reference* for more information). In this way, each Enterprise Extender connection can maintain its own liveness timer value depending on network conditions. The liveness timer specifies how long the LDLC layer will wait without receipt of a network layer packet (NLP) from the connection partner before sending a liveness message (TEST frame) to verify that the connection is still operational. During periods of inactivity, an EE connection can expand its current liveness timer up to the limit specified by the max_value parameter. During each liveness interval, if no HPR data is received from the EE partner, the current liveness interval doubles in size. This process is repeated until the maximum liveness timer value is reached, or HPR traffic resumes. If the maximum liveness timer value is reached, the liveness timer will remain at that value. When HPR traffic resumes, the current liveness window is reset to the initial value and the EE connection is immediately tested to verify partner reachability.

The amount of CPU savings realized by LDLC keep-alive reduction is system-dependent. When HPR over EE configurations specify DISCNT=NO, significant CPU savings are possible.

Enterprise Extender improved packet loss tolerance

While RTP pipes generally perform well over Enterprise Extender under ideal network conditions, the overall performance of RTP pipes can be affected when packet loss is occurring. The Enterprise Extender improved packet loss tolerance function is designed to improve the performance of RTP pipes while running in non-ideal network conditions. This function is enabled by coding the HPRCLKRT start option to the ADAPTIVE mode.

Coding ADAPTIVE for the HPRCLKRT setting allows the HPR clock to operate in adaptive mode. Running the HPR clock in adaptive mode allows the clock to toggle between standard and high modes of operation when servicing RTP pipes which are experiencing packet loss. When packet loss occurs, the clock speed adapts to run in high mode. This causes more frequent scheduling of the RTP pipes which are experiencing packet loss, allowing the RTP pipes to recover quicker. There may be some CPU overhead for running the HPR clock in high mode. When packet loss is no longer occurring, the clock returns to the standard mode.

The adaptive mode is designed to increase the overall performance of RTP pipes when packet loss is occurring. However, RTP performance and CPU overhead are system-dependent and may vary accordingly.

Restrictions: When coding HPRCLKRT=ADAPTIVE:

- This mode is only valid in Enterprise Extender configurations which have a defined capacity of 1G (gigabit) or higher access speeds.
- This mode is not recommended for uni-processor configurations or in configurations which are CPU bound.

Disconnect and inactivity summary

The LDLC layer monitors the EE connection, sending a test frame if no activity is detected for the number of seconds specified by the LIVTIME operand as coded (or defaulted) on the XCA PORT statement. If no response is received for the number of seconds specified by the SRQTIME operand, another test frame will be issued. As long as no response is received, LDLC will retry SRQRETRY + 1 times. If no response is received after LDLC tries again the last time, the EE link will be disconnected.

At a slightly higher layer, z/OS Communications Server monitors the EE switched PU if DISCNT=YES or =DELAY is specified.

- If you specify YES, a disconnect is generated if no traffic flows over that PU for the amount of time specified by the DISCNTIM start option.
- If you specify DELAY, a disconnect is generated if no traffic flows over that PU for the amount of time specified by the value you specified for the delay.

z/OS Communications Server provides the EE switched PUs disconnect time value to each RTP, and the RTP sets its liveness timer (which is independent from the LDLC liveness timer associated with the LIVTIME parameter) to half of the disconnect value, or 3 minutes, whichever is smaller. This ensures that as long as an RTP connection is actively associated with that EE switched PU, it sends status messages (generated when the liveness timer pops) often enough to keep the PU from disconnecting due to inactivity, even if there is a lull in data traffic using the RTP connection.

At yet a higher layer, the RTP connection is itself represented by a switched PU that, by default, looks like a PU with DISCNT=YES and a 10 second DISCNTIM value. Therefore, by default the RTP PU will disconnect itself if its last session goes away, and no new session is queued to it for a period of 10 seconds. If a different DISCNTIM value is desired for the RTP PU, or if DISCNT=NO behavior is desired (meaning the RTP PU will not disconnect until forced down by lack of connectivity, partner deactivation, or operator command), then you can code a model PU, DYNTYPE=RTP.

Customizing IP type of service

EE maps the SNA transmission priority to the IP type of service so that routers can preserve the SNA transmission priority across the IP network.

The IP type of service (ToS) used for IPv4, and the IP traffic class used for IPv6, are related to the transmission priority of the chosen APPN class of service (COS) and the UDP port numbers used for EE traffic.

Tip: For IPv6 protocols, IP traffic class performs the same function as IP ToS does for IPv4 protocols. The term ToS is used in this document to refer to both functions.

The APPN COS specifies a transmission priority (with the keyword PRIORITY), which can be one of the data values shown in Table 6.

Restriction: Although the IPPORT operand enables you to change the port numbers used for EE, many platforms do not permit this. Therefore, do not change the IPPORT default value.

The UDP port value specified on the IPPORT keyword (default 12000), is reserved for signal data that is involved in establishing and keeping the EE connection active. The next four sequential UDP port numbers are reserved for the SNA traffic priorities (network, high, medium, and low).

IP routers can use the UDP port number to prioritize traffic. Likewise, by default, the ToS values in Table 5 tell IP routers that support this function how to prioritize data if they do not support prioritization by the port number. This also applies in cases where IPSec is being used.

Table 6. SNA priorities and corresponding port numbers and default ToS values

Priority	Port	ToS value
LDLC Signaling *	12000	C0
Network	12001	C0
High	12002	80
Medium	12003	40
Low	12004	20

Note: * Port 12000 carries EE LDLC signaling data and does not correspond to one of the four SNA transmission priorities.

Not all routers support prioritization of traffic based on port number or ToS value. Some might support none and others might support one or both. This is why two methods are needed (both ToS and port number) for the prioritization of HPR

traffic in an IP network. In addition, if IPSec is being used, EE data can be encrypted in the IP network. This encryption includes the UDP port numbers, but not the ToS values.

Guideline: For optimum preservation of SNA transmission priority, use prioritization based on ToS value.

Rule: If HPR data prioritization in an IP network is needed while using IPSec, the IP routers in the network must support the ToS function. Also, all routers and EE endpoints must be consistently defined for ToS and the UDP port numbers.

Advanced coding considerations for EE

The topics described in this section include:

- "EE connection network reachability awareness"
- "TCP/IP MTU size for EE" on page 193
- "Running EE in constrained or virtualized environments" on page 194
- "RTP transmission stall operator awareness and recovery support" on page 195
- "Load balancing" on page 196
- "Transmission group profiles" on page 197
- "Dynamic reconfiguration" on page 197
- "Dial usability - DWACT, DWINOP, KEEPACT, REDIAL, and REDDELAY" on page 197
- "Customization for EE connection network PUs" on page 198
- "Cross-subnet routing with global VRNs" on page 199

EE connection network reachability awareness

When a dial failure or a connection failure occurs for a connection over an Enterprise Extender connection network, the partner node cannot be reached using the connection network path across the virtual routing node (VRN). This connection network path might have been chosen for this connection because it had a lower weight than any alternate path available at one of the following times:

- The time of this failing dial
- The time of the dial that set up the existing connection
- The time of a path switch to this connection network path for an existing connection

If this path still has the lowest weight of any available path to the partner node, any attempt to redial the partner node will continue to try the path over this particular VRN, which is likely to result in failure until the underlying problem with the path is corrected.

EE connection network reachability awareness detects the dial failure or connection INOP for the connection over an Enterprise Extender connection network and prevents that specific path to the partner node from being used for a period of time. The path through the Enterprise Extender VRN to the unreachable partner is unidirectional and consists of three components:

- The node on the origin side of the virtual node, which is the node that detects the problem
- The virtual node defined by the Enterprise Extender connection network
- The unreachable partner node on the destination side of the virtual node

It is possible for the path in one direction between two nodes and through the VRN to be usable for route selection, while the path in the other direction is not usable. This function can detect that distinction and when new sessions are established and HPR path switches occur, it allows routing in the direction that is usable, while preventing the path in the direction that is not usable from being selected.

Unreachable time (UNRCHTIM) considerations

With EE connection network reachability awareness, you can use a start option, UNRCHTIM, to indicate that paths to partner nodes over Enterprise Extender VRNs should not be used for route selection for a period of time after initial dial failures or connection INOPs; this provides time for the underlying connection problem to be corrected. An alternate route to the partner node is selected, if an alternate route is available, when new sessions are established and HPR path switches occur. If an alternate route is not available, the session or HPR path switch fails with sense code 80140001. UNRCHTIM can also be coded on a PORT or GROUP definition statement that defines a connection network (with the VNNAME and VNTYPE operands) in an EE XCA major node. This enables a specific connection network to have a different unreachable time value than the value specified on the UNRCHTIM start option. See the UNRCHTIM start option in *z/OS Communications Server: SNA Resource Definition Reference* for more information.

The ramifications of the unreachable time value you choose are important. The results can vary depending on the duration of failures that occur. A value that is appropriate for a short-term failure might not be appropriate for a long-term failure. For example, if the value is set very low and a long-term outage occurs, you introduce the performance impact of added network flows to communicate the unreachable partner information to all nodes each time a path is determined to still be unreachable. In this case, many session or HPR path switch failures can still occur, even though an alternate route is available. Alternatively, if the value is set high and a short-term outage occurs, any paths to unreachable partners detected at that time will remain unavailable for use until the unreachable time expires or until the unreachable partner information is cleared with the MODIFY TOPO,FUNCTION=CLRUNRCH command.

You can change the UNRCHTIM start option value while VTAM is running with the MODIFY VTAMOPTS command. You can also change the unreachable time for a specific Enterprise Extender connection network using the VARY ACT,UPDATE=ALL command with the Enterprise Extender XCA major node after changing UNRCHTIM on the PORT or GROUP definition statement. However, the unreachable time value is not changed for paths to unreachable partners that were determined to be unreachable before the value was modified. When the existing unreachable time value for a path expires, the new unreachable time value will be used if the partner node through that path is again determined to be unreachable.

Rule: The unreachable time for an Enterprise Extender connection network should be set to the same value in all network nodes. If the values are different, the unreachable time used for the Enterprise Extender connection network will be unpredictable. For global connection networks between partner companies, it is practical for unreachable times to be the same but it is not mandatory.

An end node that has detected an unreachable partner path notifies its network node server (NNS) of the unreachability so that the NNS can use that information when calculating routes for the end node. Beginning with z/OS V1R8 Communications Server, the NNS will override the unreachable time value sent by

the EN using its unreachable time value for the associated connection network, if configured, or the value specified in its UNRCHTIM VTAM start option. With this enhancement, changing the unreachable time value for an Enterprise Extender connection network will no longer require you to update every attached end node. Only the network nodes and those attached end nodes whose NNS is running z/OS V1R6 or V1R7 Communications Server will need to be updated with the new value. You must still configure an unreachable time value on every end node to activate the EE connection network reachability awareness function.

Performance consideration: By default, the UNRCHTIM start option has a value of 0 for all virtual routing nodes; therefore, EE connection network reachability awareness is disabled. While EE connection network reachability awareness can be enabled for all virtual routing nodes defined at this node (by this VTAM) by setting the UNRCHTIM start option to a value in the range 10 - 65535, the preferred mechanism is to set UNRCHTIM at the GROUP level of the EE XCA major node. This setting provides maximum flexibility in choosing the unreachable time value for each VRN, and in selectively enabling and disabling the function for each VRN.

When EE connection network reachability awareness is enabled for a VRN, APPN Topology and Routing Services begins maintaining unreachability records if failures occur while using that VRN. The presence of unreachability records in the topology database increases the number of CPU cycles that are consumed during APPN route calculation. This CPU cycle increase is magnified as the number of VRNs that contain unreachability records increases and as the average number of unreachability records increases for each of those VRNs.

To mitigate the CPU cost of enabling EE connection network reachability awareness, the following considerations apply:

- Typical EE network designs require that only few connection networks be accessible from a single APPN subnet. For network designs that employ more than four EE connection networks that are accessible from a subnet, EE connection network reachability awareness should be enabled for no more than four of those EE VRNs at any given time.
- While the mechanism employed by EE connection network reachability awareness provides the tracking of connection failures on a per-partner basis, failures to many partners typically indicate a pervasive problem with the underlying transport. You can set a limit on the number of unreachable partner records (partner_limit) associated with a VRN on the UNRCHTIM start option. A network node maintains all unreachability records that are reported from its served end nodes and from all other network nodes in the same network. When the count exceeds this limit of unreachable partner records for a VRN, the VRN is considered unusable for any access until enough existing unreachable partner records expire or are manually cleared with the MODIFY TOPO,FUNCTION=CLRUNRCH command. The count of unreachable partner records must drop below 80% of the unreachable partner limit specified or defaulted on the UNRCHTIM start option for the VRN to again be considered for route selection.

EE virtual node route selected after UNRCHTIM is detected: When an Enterprise Extender VRN is selected by VTAM Topology and Routing Services, a DIAL request is made using the dynamic PU. The session remains in ADIALIP status until the DIAL request completes or until there is a timeout on the connection.

If there is a problem with the physical connection underlying the path through the VRN, the dynamic PU eventually receives an INOP because of the timeout. The following messages are received:

```
IST259I   INOP RECEIVED FOR CNV00074 CODE = 01
IST1411I   INOP GENERATED FOR linename
IST1430I   REASON FOR INOP IS XID OR LDLC COMMAND TIMEOUT
        .
        .
IST1903I   FAILURE OVER vrnname TO CP partner_cpname
IST2050I   THIS PATH WILL NOT BE SELECTED FOR UNRCHTIM = seconds
                  SECONDS
IST314I   END
```

Next, the session fails with sense 08010000. This is normal processing.

When a DISPLAY NET,TOPO,LIST=UNRCHTIM command is entered, the path is now shown as unreachable:

```
IST2057I   UNREACHABLE PARTNER INFORMATION:
IST924I   - - - - - - - - - - - - - - - - - - - - - - - - - - -
IST2150I   VIRTUAL NODE vrnname - 1 UNREACHABLE PARTNERS
IST2052I   ORIGIN NODE    PARTNER NODE    UNRCHTIM EXPIRES
IST2055I   origin_cpname  dest_cpname     1800s  07:54:10
IST314I   END
```

However, it appears that TRS is still selecting the unreachable path through the Enterprise Extender VRN because sessions continue to fail with sense code 08010000 following the unreachable partner messages that are received for additional dynamic PUs (for example, if the first INOP is for CNV00074, there can be additional INOPs for CNV00075, CNV00076, and so on).

The reason this is occurring is that these sessions were requested, and TRS selected the route through the EE VRN, before the first PU INOP and the detection of the unreachable partner. It can take a significant amount of time from the DIAL request until the timeout occurs. During this time, a number of sessions can be requested. A DISPLAY SESSIONS,LIST=ALL command will show the session status of the session for which a DIAL request is pending as ADIALIP. The session status of the other sessions waiting for this same path will show as PRAV3. Each of these pending sessions will fail. Further displays of the unreachable partner information will show that the EXPIRES time changes each time an INOP occurs.

This is normal processing for the EE connection network reachability awareness function. Any sessions that were pending before the INOP will fail. However, for any future sessions requested after the unreachable partner information is known, TRS will attempt to route around the EE VRN.

EE connection network reachability awareness in a mixed-release environment: EE connection network reachability awareness can coexist in a mixed-release environment. However, if the PLU of a session resides on an NN with a release earlier than V1R6 (or, without EE connection reachability awareness enabled), or on an EN whose NNS is a release earlier than V1R6 (or a node without EE connection network reachability enabled), then that session will continue to fail as it did before the introduction of this function.

Consider the following configuration:

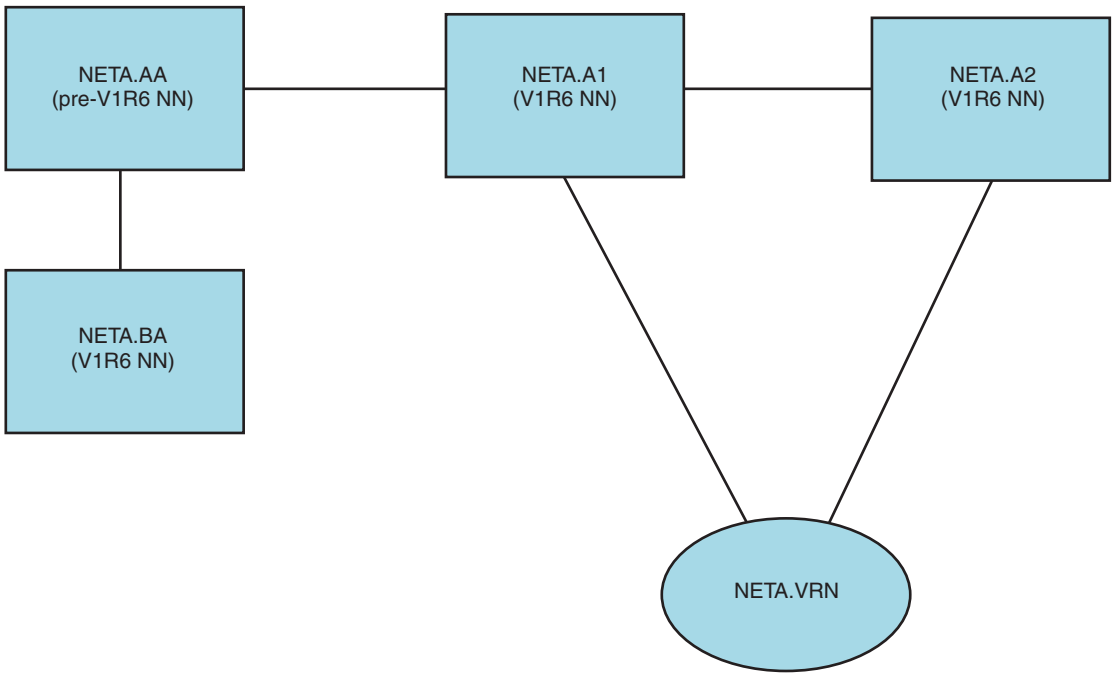


Figure 50. EE connection network reachability awareness in a mixed-release environment

Assume the PLU resides on NETA.AA, a pre-V1R6 network node or a node without EE connection network reachability awareness enabled. The SLU resides on NETA.A2, a network node that supports EE connection network reachability awareness and has the function enable for NETA.VRN.

The intermediate node, NETA.A1, is also a network node that has EE connection network reachability awareness enabled for NETA.VRN. Both NETA.A1 and NETA.A2 have the unreachable time value for the EE virtual node, NETA.VRN, set to 180 seconds. There is a problem with the connection that underlies the connection network path through NETA.VRN and it has already been detected in NETA.A1 and NETA.A2.

A DISPLAY NET,TOPO,LIST=UNRCHTIM entered at NETA.A1 would show the following output:

```
IST20571   UNREACHABLE PARTNER INFORMATION
IST924I  - - - - - - - - - - - - - - - - - - - - - - - - - -
IST2150I   VIRTUAL NODE NETA.VRN 2 UNREACHABLE PARTNERS
IST2052I   ORIGIN NODE  PARTNER NODE    UNRCHTIM   EXPIRES
IST2055I   NETA.A1      NETA.A2         180S     07:54:10
IST2055I   NETA.A2      NETA.A1         180S     07:54:12
IST314I   END
```

A request is made to VTAM Topology and Routing Services (TRS) in the PLU network node (or if the PLU is on an end node, the NNS of the EN) to select the route that is to be taken to reach the SLU node. So if the PLU is on NETA.AA, which is a pre-V1R6 node, TRS in NETA.AA has no knowledge of the unusable path through NETA.VRN and, assuming that is the best (lowest weight) path between NETA.A1 and NETA.A2, will select that route.

When NETA.A1 becomes aware of the route to be taken, a DIAL request is made across the bad connection using a dynamic PU (these PUs usually default to a name of CNVxxxx). After a period of time, the DIAL request will timeout and the PU will INOP with the following messages:

```
IST259I   INOP RECEIVED FROM CNVxxxx CODE = 01
IST1411I  INOP GENERATED FROM linename
IST430I   REASON FOR INOP IS XID OR LDLC COMMAND TIMEOUT
       .
       .
IST1903I  FAILURE OVER vrnname TO CP partner_cpname
IST2050I  THIS PATH WILL NOT BE SELECTED FOR UNRCHTIM = seconds
                    SECONDS
IST314I   END
```

The session will fail in NETA.AA with sense code 08010000. In NETA.A1, where the IST1903I and IST2050I messages were received, the EXPIRES time for the path with NETA.A1 as the origin and NETA.A2 as the partner will be reset to 180 seconds after this failure is detected. The EXPIRES time for the path with NETA.A2 as the origin and NETA.A1 as the partner will not be reset because NETA.A2 did not detect a problem in the reverse direction.

A DISPLAY NET,TOPO,LIST=UNRCHTIM entered at NETA.A1 would now show the following output:

```
IST2057I   UNREACHABLE PARTNER INFORMATION:
IST924I  - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
IST2150I   VIRTUAL NODE NETA.VRN - 2 UNREACHABLE PARTNERS
IST2052I   ORIGIN NODE      PARTNER NODE    UNRCHTIM  EXPIRES
IST2055I   NETA.A1          NETA.A2         180S    07:58:45
IST2055I   NETA.A2          NETA.A1         180S    07:54:12
IST314I   END
```

Sessions routed through the VRN by the pre-V1R6 node (or a node without EE connection network reachability awareness enabled) will continue to fail as long as the connection underlying the EE VRN is not working. However, if the PLU resides on any of the nodes with EE connection network reachability awareness enabled (for example, where that node is selecting the route), sessions will be routed around the bad connection until the unreachable time expires. This is true even when the PLU is on NETA.BA, which is not connected to either of the nodes with EE connection network reachability awareness enabled that detect the failure. This node is informed of the unreachable partner information with Topology Database Updates (TDUs). The pre-V1R6 node, NETA.AA, does not recognize the unreachable partner information about the TDUs, but will pass that information about to NETA.BA, the node with EE connection network reachability awareness enabled.

Displaying unreachable partner information

The DISPLAY TOPO,LIST=UNRCHTIM command displays Enterprise Extender connection network unreachable partner information. Unreachable partner information is maintained only by network nodes and, for this reason, this command is only valid on network nodes. Unreachable partner information identifies paths from one node through an Enterprise Extender virtual node to a second node that are considered unreachable due to a connection network dial failure or connection network link INOP condition. When this occurs, the problem path is remembered for the period of time specified as the unreachable time value for that connection network. During this period, the problem path will not be considered for new sessions or HPR path switches.

To display unreachable partner information, you can specify:

- The name of an origin node on an unreachable partner path, using the ORIG operand
- The name of a virtual node on an unreachable partner path, using the VRN operand

- The name of an unreachable partner (destination) on an unreachable partner path, using the DEST operand

The ORIG, VRN, and DEST operands can be used in any combination to control the scope of the unreachable partner information that is displayed. If you omit all three of these operands, all Enterprise Extender unreachable partner information known by the network node is displayed. Depending on the value of the DSPLYWLD start option, wildcard values can be used for the ORIG, VRN, and DEST operands.

You can use the DISPLAY command to help isolate the location of the problem with the underlying connection. For example, if the display output indicates that a specific partner is unreachable through more than one Enterprise Extender VRN, the problem might be on the partner node or a router on the partner node side of the connection. If the display output indicates that different partners are unreachable through one or more Enterprise Extender virtual nodes from the same origin node, the problem might be on the origin node's side of the connection.

Clearing unreachable partner information

After the underlying problems with the Enterprise Extender connection network paths are corrected, you can clear the unreachable partner information to make these paths available for route selection before the unreachable time value would automatically expire. The MODIFY TOPO,FUNCTION=CLRUNRCH command can be entered on a network node to clear a set of unreachable partner paths for Enterprise Extender connection networks, making those paths available for route calculation (if they were previously unavailable for route calculation as a result of unreachable partner information).

Use the SCOPE=LOCAL operand to clear the unreachable partner paths from only the network node on which the command is entered. Use the SCOPE=NETWORK operand to clear the unreachable partner paths from all network nodes in the network. (Remember that topology flows do not occur across network boundaries.)

If a problem is corrected with only one of multiple paths associated with the EE connection network and you clear the unreachable partner information for more than just that one path, the first attempt to establish a new session over a path that is still unreachable (or the first time an HPR path is switched to a path that is still unreachable) results in the following:

- The new session or HPR path switch will fail
- New unreachable path information will be created for the unavailable path

Maintaining unreachable partner information in the topology database

Unreachable partner information about a path through an Enterprise Extender connection network is sent to an end node's network node server when the failure is detected in an end node, or broadcast to all adjacent network nodes when the failure is detected in a network node, using control vectors X'44', X'46', X'47', and X'4C' on topology database updates (TDUs). Therefore, it is possible for the unreachable partner information displayed in one node to have an expiration time that is slightly different from the expiration time for the same unreachable path in another node, because of delays in network traffic.

When a CP-CP session is established between two network nodes, there is an exchange of topology information. The unreachable partner information, including the unreachable time value, is sent in the control vector X'4C'. However, it is possible that a significant amount of time has expired between the time that the

failure was detected and the time this topology information is sent. To make sure the time the unreachable partner information expires is consistent, the unreachable time value is modified in the control vector X'4C'. When this occurs, the unreachable value will not be displayed in the node that receives the modified unreachable time value. Instead, five asterisks (*****) will be displayed in the UNRCHTIM column on a DISPLAY NET,TOPO,LIST=UNRCHTIM command. However, the expiration times in the node detecting the failure and the node receiving the modified unreachable time value should be consistent.

Unreachable partner information that is detected by an end node and reported to the end node's NNS is not maintained by the end node. If the end node switches to a new NNS, a CP-CP session is established between the end node and its new NNS and the end node's topology information is sent to the NNS. However, since the end node is not maintaining unreachable partner information, any unreachable paths that had been detected by the end node and reported to the old NNS are not included in this topology information sent to the new NNS. Therefore, some failures will be necessary when attempting to establish new sessions, or when HPR path switches occur, until the new NNS learns about the unreachable paths.

Unreachable partner information is not copied to the topology data set when a MODIFY CHKPT command is issued with TYPE=ALL or TYPE=TOPO specified, or when a HALT or HALT,QUICK command is issued to end VTAM. Therefore, some failures are necessary when attempting to establish new sessions, or when HPR path switches occur, in order to refresh the unreachable partner information after restarting VTAM.

Route selection anomalies

When one path to a partner node over an Enterprise Extender VRN is unreachable, it is possible that the alternative route selected might not be the least weight route from the PLU node to the SLU node. Figure 51 and Figure 52 on page 192 are examples of configurations in which this can occur.

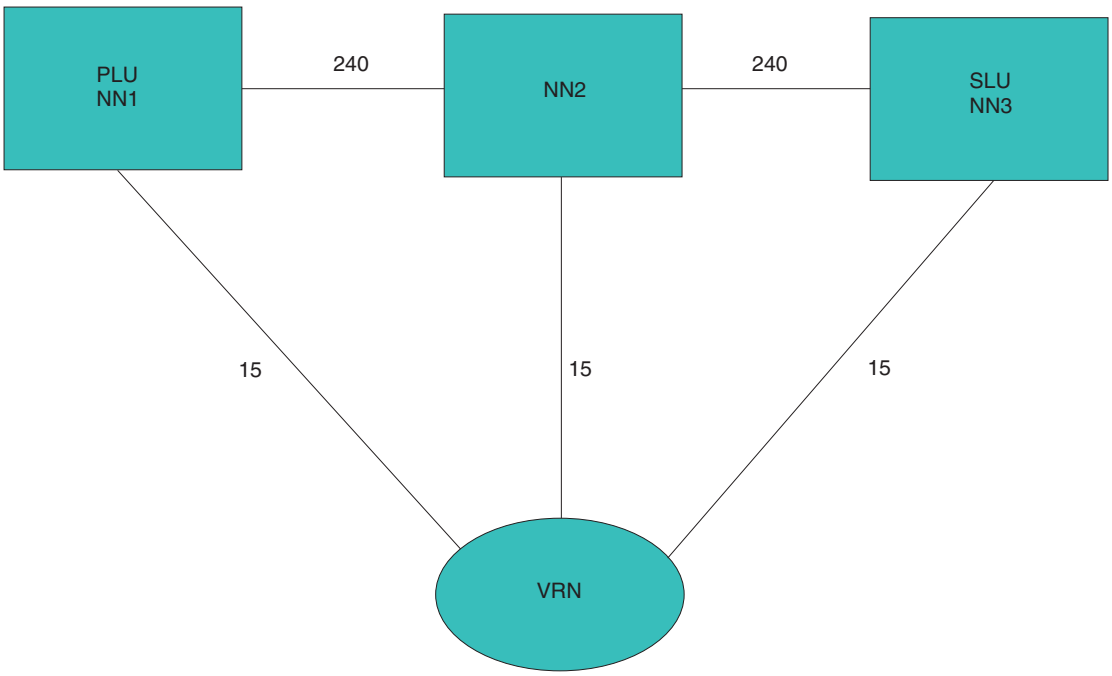


Figure 51. Connection network reachability example 1

In the configuration shown in Figure 51 on page 191, the route is calculated in the network node where the PLU resides. The weights of the TGs between the network nodes NN1 and NN2, and between the network nodes NN2 and NN3, are each 240. The weight of the TG between each network node and the VRN is 15, making the paths through the VRN preferred over the direct connections between nodes. Assume that NN3 is an unreachable partner on the path from NN1, through the VRN, to NN3, meaning that this route will not be selected for session setup. The other paths through the VRN are still acceptable for route selection.

With this unreachable path eliminated, the next least weight route goes from NN1, through the VRN, to NN2, through the VRN a second time, to NN3. However, to prevent the calculation of circular routes, VTAM Topology and Routing Services does not allow any node to exist more than one time in a selected route. Therefore, the final route is the higher weight route from NN1, through the VRN to NN2, then from NN2 to NN3.

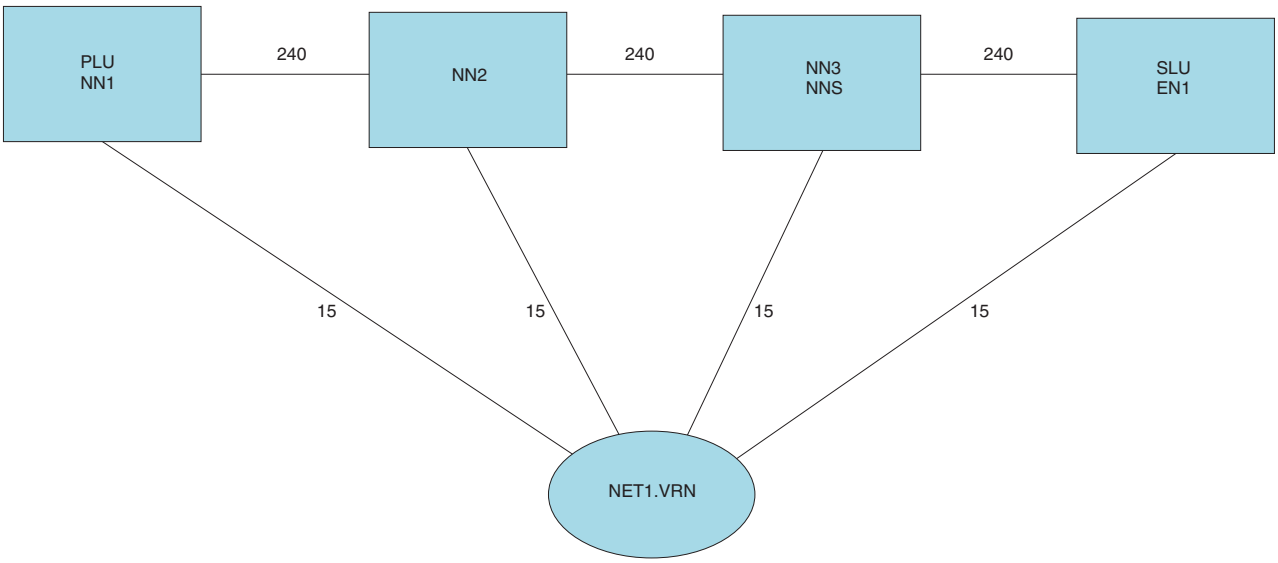


Figure 52. Connection network reachability example 2

In the configuration shown in Figure 52, the route is also calculated in the network node where the PLU resides. The weights of the TGs between the network nodes NN1 and NN2, between the network nodes NN2 and NN3, and between network node NN3 and end node EN1, are each 240. The weight of the TG between each node and the VRN is 15, making the paths through the VRN preferred over the direct connections between nodes. Assume that NN2, NN3, and EN1 are all unreachable partners on the paths from NN1 through the VRN, meaning that these routes will not be selected for session setup. The other paths through the VRN are still considered to be usable for route selection.

With the unreachable paths eliminated, the next least weight route goes from NN1 directly to NN2, then through the VRN to EN1. However, because of the way that routing trees are built, VTAM Topology and Routing Services puts the path from NN1 to the VRN on the tree first. It must remain on the tree because it will be checked later, after the tree is completed, to determine whether the endpoint TG from the VRN to EN1 can be used. This means that, because the VRN is already on the tree, it cannot be added again for the path from NN2 to the VRN (because a node is not allowed to exist more than one time in a routing tree, to prevent

circular routes). The same is true for the path from NN3 to the VRN. Therefore, the final route is the higher weight route from NN1 to NN2, from NN2 to NN3, and then from NN3 to EN1.

TCP/IP MTU size for EE

To ensure optimal performance, the TCP/IP maximum transmission unit (MTU) size should be greater than or equal to the RTP network layer packet (NLP) size. The MTU size (both IPv4 and IPv6) might change during the life of the EE connection. The MTU can change for the following reasons:

- Initially, VTAM queries the TCP/IP stack for its MTU size and sets the EE connection to use this value. This MTU size has already been reduced to account for various header lengths such as the IP, UDP and LLC headers necessary for EE traffic.
- VTAM also takes into account the VTAM MTU operand value, if specified. The MTU operand may be specified on three types of VTAM major nodes:
  - For EE connection networks, this parameter may be defined on the connection network GROUP definition statements in the EE XCA major node.
  - For dial in Enterprise Extender connections which have their associated PUs dynamically created, this parameter may be defined on the model major node (DYNTYPE=EE) PU definition statement.
  - For predefined Enterprise Extender connections, this parameter may be defined on the PU definition statement in the switched major node.
- VTAM then takes the lesser of the TCP/IP stack's computed MTU size and the VTAM defined MTU operand value (if specified). If the TCP/IP stack presents a value less than 768 bytes, VTAM sets the MTU to 768 since this is the smallest packet size allowed by the HPR architecture.
- Generally the MTU size for an EE connection is fairly constant once the EE connection is established. However, in the event the TCP/IP stack's MTU size changes, RTP pipes with endpoints on the same node as the TCP/IP stack dynamically detect these changes when their outbound packets are being transmitted. Some reasons for MTU size changes include the following:
  - New IP routes come available with different local MTU sizes
  - Existing IP routes become unavailable.
  - Path MTU discovery is enabled for IPv4 or IPv6 EE connections (see the PMTUD start option in *z/OS Communications Server: SNA Resource Definition Reference* for details), and path MTU changes are discovered in the IP network.

Table 7 shows connection conditions and related results. When establishing an RTP connection (for a CP-CP session, to transport ROUTE_SETUP GDS variables, or for an LU-LU session) over an EE connection, RTP pipes learn the MTU size when the pipes are being established (RSETUP flows). RTP then segments data to this size when transmitting outbound data.

Table 7. Connection conditions and results

If	Then
This node is the origin endpoint of the RTP connection	VTAM sets the maximum packet size equal to the lesser of the MTU size or the VTAM maximum data size

Table 7. Connection conditions and results (continued)

If	Then
This node is an intermediate node or the destination node of an RTP connection	VTAM sets the maximum packet size equal to the lesser of the MTU size, VTAM maximum data size for the next hop, or the value received on the ROUTE_SETUP GDS variable
This node is one of the endpoints of the RTP connection and a change in the EE connection's MTU size occurs.	When VTAM detects this condition (the EE connection's MTU size changes during the transmission of an NLP) the MTU size is altered. This change is specified in message IST2029I when you issue the DISPLAY EE command. Also, if this change alters the permitted NLP size (NLP size cannot be increased beyond the originally negotiated value for the RTP connection) then this change is specified in message IST1511I which is displayed as the result of the DISPLAY ID=rtp-pu command.

If the MTU size is less than 768 bytes, VTAM sets the maximum packet size to 768 (this is the smallest maximum packet size allowed by VTAM for HPR packets). This limitation can cause TCP/IP to fragment but exists because the RTP layer cannot allow the HPR header to be segmented in the RTP layer.

Running EE in constrained or virtualized environments

To proactively prevent congestion, High-Performance Routing (HPR) flow control (responsive-mode ARB) is sensitive to minor variations in round-trip time or unpredictable response times from the partner. This sensitivity can affect installations in at least two ways:

- Frequent swings in round-trip times lead to ARB rate cuts that prevent the RTP pipe from ramping up to speeds consistent with what the media and network should allow. These rate cuts result in degraded throughput and reduced response time.
- These same swings in round-trip times can cause the RTP endpoint to prematurely enter the path switch state. Although this state does not affect availability, it uses CPU cycles and can result in a significant amount of unnecessary message output.

There are a number of potential causes for these variations. A partner host that is CPU-constrained cannot guarantee consistent and timely responses to ARB flows. Furthermore, the increasing trend toward virtualized environments is making these types of problems more likely to occur.

You can use progressive mode ARB to enable HPR to perform reliably and consistently in platforms using virtualization. Progressive mode ARB is an HPR flow control mechanism that improves the performance of HPR in these environments. Progressive mode ARB is available only to single hop HPR pipes that are over TCP/IP, which includes a single physical hop across a two-hop EE virtual routing node (VRN).

You can use the HPREEARB parameter to define which EE connections will enable the progressive mode ARB. Specify the HPREEARB parameter on the PU definition

statement in the switched or model (DYNTYPE=EE) major nodes, or on the connection network GROUP definition statements in the EE XCA major node.

Once defined, VTAM negotiates with the HPR partner whether progressive mode ARB will be used on the RTP connection. If both partners desire to use progressive mode ARB and the HPR pipe is a single hop over EE (which includes a single physical hop across a two-hop EE VRN), then progressive mode ARB will be used.

Use the VTAM start option HPRPSDLY to reduce the number of unproductive path switches. With this option, you specify the minimum amount of time, in seconds, that all HPR RTP pipes must delay before entering a path switch state that is caused by an unresponsive partner. During this time, the RTP endpoint periodically tries to contact the partner to avoid switching paths. If you specify the value of 0, RTP pipes initiate path switch processing when a predetermined number of attempts to try again have been unsuccessful. You can also specify the HPRPSDLY parameter on the same three major nodes for EE configurations that were previously described for the HPREEARB parameter.

When you specify the EEDELAY setting for the HPRPSDLY parameter in the major node, VTAM calculates the number of seconds that RTP pipes must delay before entering path switch due to an unresponsive partner. The calculated value allows enough time for the EE keep-alive mechanism to end the EE connection should connectivity be lost to the partner. The benefit is that while EE is determining if there is lost connectivity, the RTP layer is not performing unnecessary path switches. When the EEDELAY value is specified for a path switch delay, and EE is the only path to the RTP partner, the HPRPST value should be specified to a value sufficiently large to allow for the EE connection to be re-established. This allows for the RTP pipe and its associated LU-LU sessions to be successfully recovered.

RTP transmission stall operator awareness and recovery support

If HPR packets are being dropped in the network, an RTP endpoint will request the partner RTP endpoint to retransmit the network layer packet (NLP). There are many reasons why the packet may not be delivered. One common problem is the NLP size which the RTP endpoint is using may be outdated or inaccurate. This situation can lead to an RTP endpoint continuously requesting the same NLP be retransmitted. The sending RTP endpoint honors the request by repeatedly retransmitting the same NLP, but the partner does not receive this NLP. When this occurs, the RTP pipe is considered to be in a transmission stall.

VTAM has support to provide operator awareness of an RTP pipe experiencing a transmission stall. When a specific NLP is retransmitted six times without being acknowledged by the RTP partner (as being received), and at least 10 seconds has elapsed since the NLP was first retransmitted, VTAM will notify the operator with message IST2245I as follows:

```
IST2245I XMIT STALL DETECTED FOR RTP puname TO cpname
```

At this point, VTAM provides RTP transmission stall recovery support for all RTP pipes whose path traverses an Enterprise Extender link. When the RTP endpoint is not directly attached to EE, but the RTP pipe consists of multiple hops with one being a REFIFO hop, VTAM assumes an EE hop is in the path. A REFIFO link is one that indicates that packets flowing across it are not guaranteed to be delivered in order. Recovery support consists one of the following two actions:

- First, if the HPR connection traverses an one hop EE or a two hop EE connection network, the RTP endpoint immediately lowers its NLP size to the

HPR-architected minimum size of 768 bytes. All outbound data is then resegmented to this value to resolve the transmission stall.

- Second, VTAM starts a path switch between two RTP endpoints. The path switch allows the EE link to provide the RTP endpoints an updated link size for the EE connection. If the stall persists, VTAM issues message IST2246I every 30 seconds until the stall is alleviated. VTAM also lowers its NLP size to the HPR-architected minimum of 768 bytes if a previous path switch did not alleviate the stalled condition. Message IST2246I follows:

  IST2246I XMIT STALL CONTINUES FOR RTP *puname* TO *cpname*

If the recovery actions taken by VTAM result in the packets being delivered to the partner, the transmission stall is considered alleviated at which point VTAM issues message IST2247I as follows:

IST2247I XMIT STALL ALLEVIATED FOR RTP *puname* TO *cpname*

At this point normal data flow resumes. After 20 minutes of operating with the reduced MTU size, this RTP pipe's original NLP size is restored in an attempt to increase RTP throughput. If the network will still not allow transport of the larger packets, the stall detection and alleviation process will repeat.

If at any point, a transmission stall extends beyond the time limit specified by the HPRSTALL VTAM start option, VTAM automatically initiates termination of the HPR pipe and issues the following message:

IST2253I HPRSTALL TIME EXCEEDED FOR RTP *puname* TO *cpname*

Load balancing

If you define multiple links to different VRNs (using different IP addresses), you can choose different TG characteristics on these definitions to force sessions using different APPN COS names to flow over different VRNs (using different local IP addresses). The IP network can also be configured to route traffic for different APPN COS names (IP ToS values) using different physical IP interfaces. For example, #INTER traffic could be sent over a Gigabit Ethernet connection and #BATCH traffic could be sent over a slower OSA link.

Using this configuration, you can monitor APPN traffic flows by displaying the RTPs that flow through a specific VRN. You can accomplish this by specifying the FIRSTCP=vrnname operand, the FIRSTTG=tgnumber operand, or both on the DISPLAY RTPS command. This configuration also enables you to monitor IP traffic flows by tracing flows based on source or destination IP addresses.

If your APPN product does not allow you to define multiple VRNs at the same time, you can define two links to the same VRN (using different IP addresses) and force sessions to be load balanced over those connections. You can still monitor APPN and IP traffic by specifying both the FIRSTCP= and FIRSTTG= operands on the DISPLAY RTPS command in order to see what traffic is flowing over what link.

You might also want to use only a single IP address, but still load balance sessions through parallel VRNs. With this configuration, it is not possible to monitor IP traffic over each of these connections, because there is only a single local IP address being used. But you would still be able to monitor APPN session traffic over the parallel routes (VRNs) by using the FIRSTCP= and FIRSTTG= operands on the DISPLAY RTPS command. Furthermore, because only a single IP address is being used, this type of configuration might simplify the coordination of system definitions.

Transmission group profiles

Six sample TGPs are provided in IBMTGPS for Enterprise Extender.

- Enterprise Extender TGs over WAN (EEXTWAN)
- Enterprise Extender TGs over campus networks (EEXTCAMP)
- Enterprise Extender TGs over Fast Ethernet (FASTENET)
- Enterprise Extender TGs over Gigabit Ethernet (GIGENET)
- Enterprise Extender TGs over 10 Gigabit Ethernet (GIGNET10)
- Enterprise Extender TGs over HiperSockets (HIPERSOC)

As with all APPN links, specify a TGP that represents the actual connectivity being used. The sample TGP FASTENET might be a reasonable choice for EE TGs traversing the Internet. If you do not specify a TGP, the default values for EEXTCAMP are used (a capacity value of 4 MB, a propagation delay value of terrestrial, and a security value of unsecure). Consider the actual capacity and propagation delay with your EE links in order to determine the appropriate TGP or link characteristic values to use.

For more information about these TGPs and details on how to use them, see *z/OS Communications Server: SNA Resource Definition Reference*.

Dynamic reconfiguration

Enterprise Extender provides substantial flexibility by enabling you to use multiple VIPA addresses or define multiple Enterprise Extender connection networks (local or global). Fully using that flexibility frequently requires coding multiple GROUP statements in the XCA major node for EE, and specifying a number of operands, such as IPADDR, HOSTNAME, VNNAME, TGP, and so on. Given all that flexibility, you might need to modify your definitions to accommodate expanding your existing Enterprise Extender deployment.

Using the VARY ACT,UPDATE command with the XCA major node for EE as the target, you can do the following:

- Change operands on a currently inactive GROUP statement
- Add or delete LINE and PU statements under a currently inactive GROUP statement
- Add new GROUP statements and remove unnecessary GROUP statements
- Change virtual node definition values on the PORT statement

When you want to change an Enterprise Extender GROUP statement, use the VARY INACT,ID=*groupname* command to deactivate the GROUP and all subordinate resources under the GROUP. Changing the values of connection network operands on the PORT statement requires deactivation of the GROUP statement identified by the VNGROUP operand.

Dial usability - DWACT, DWINOP, KEEPACT, REDIAL, and REDDELAY

To enable EE connections to automatically activate (without having to issue VARY DIAL commands), put both the EE XCA major node and the switched major node in the VTAM configuration list, and specify DWACT=YES on the EE switched PUs. When VTAM initialization completes, the EE connections will fully activate without further intervention. When the connections are activated, they should be automatically recovered after any type of failure (such as an IP network problem or even a loss of the TCP/IP stack itself).

Certain operands automate the recovery of the EE switched PUs and LINEs when initial activation fails, or when a later connection failure is detected. The operands include the following:

- KEEPACT = YES | NO

  Coding the KEEPACT value as YES on the EE LINE indicates that you want VTAM to immediately try to reactivate the LINE after it fails. This is especially useful in instances where the TCP/IP stack fails or is taken down, thereby making all of the EE LINEs inoperable. With KEEPACT=YES, the lines are reactivated automatically when the TCP/IP stack is recovered.
- DWINOP = YES | NO

  Setting Dial When INOP (DWINOP) on the switched PU to YES indicates to VTAM that you want the PU to be redialed after a failure.
- REDIAL = 0 - 254 | FOREVER

  If DWINOP=YES is specified on the switched PU, then by default VTAM will make up to 3 redial attempts. The REDIAL parameter can be coded on the PATH statement to change the number of redial attempts to a number in the range 0 - 254, or to FOREVER, meaning that VTAM will redial indefinitely.
- REDDELAY = 1 - 1200

  By default, VTAM waits 30 seconds between redial attempts. You can adjust that interval to any value in the range 1 - 1200 seconds by coding the REDDELAY parameter on the PATH statement.

When an EE connection is comprised of two z/OS EE endpoints, specifying DWACT=YES or DWINOP=YES on both endpoints can lead to both hosts attempting to initiate or recover an EE connection simultaneously. This can produce unnecessary dial collisions, resulting in delayed activation or recovery, or in some cases failed activation or recovery.

Guideline: When coding DWACT=YES, DWINOP=YES, or both, specify it on only one side of the connection to avoid PU busy conflicts.

Tip: When coding DWINOP=YES, also consider coding the REDIAL and REDDELAY operands on the corresponding switched major node PATH statement. The current defaults for these two parameters are REDIAL=3 and REDDELAY=30. With these defaults values, z/OS Communications Server retries for only approximately 120 seconds after an INOP before giving up attempts to contact the partner EE node.

Customization for EE connection network PUs

Use DYNPU=NO on the GROUP statement when you want dynamic PUs to only be created for inbound and outbound connection network connections. Following is a sample of this type of EE XCA major node:

```
XCAEE    VBUILD TYPE=XCA
PORTEE   PORT  MEDIUM=HPRIP     Enterprise Extender
GPEE1    GROUP ANSWER=ON,
               AUTOGEN=(5,LNLV1,PULV1), Create lines and PUs
               CALL=INOUT,
               DIAL=YES,
               DYNVNPFX=C1,       Connection network dynamic PU prefix
               DYNPU=NO,          No dynamic PUs (except conn network)
               ISTATUS=ACTIVE,
               TGP=GIGENET,
               VNNAME=NETA.LVRN1, Connection network CP name
               VNTYPE=LOCAL       Connection network type
```

With the EE XCA major node shown in the example, the following connections are provided for lines in group GPEE1:

- Inbound and outbound connection network links - the dynamically created PU will be named C1nnnnn
- Inbound and outbound non-connection network links when a matching active switched PU is available

Inbound non-connection network links when a matching switched PU is not available will be rejected with messages similar to the following:

```
IST680I CONNECTION REQUEST DENIED - ID = SSCP1A PU GEN NOT SUPPORTED
IST1394I CPNAME = NETA.SSCP1A        STATION ID = 0200FFF18A16
IST081I LINE NAME = LNLV1004, LINE GROUP = GPEE1, MAJNOD = XCAEE
IST314I END
```

Use DYNPU=YES on the GROUP statement when you want dynamic PUs to be created for inbound and outbound connection network connections and inbound non-connection network connections. Following is a sample of this type of EE XCA major node:

```
XCAEE    VBUILD TYPE=XCA
PORTEE   PORT  MEDIUM=HPRIP       Enterprise Extender
GPEE1    GROUP ANSWER=ON,
               AUTOGEN=(5,LNLV1,PULV1), Create lines and PUs
               CALL=INOUT,
               DIAL=YES,
               DYNPUPFX=P1,       Non-connection network dyn PU prefix
               DYNVNPFX=C1,       Connection network dynamic PU prefix
               DYNPU=YES,         Dynamic PUs allowed (all connections)
               ISTATUS=ACTIVE,
               TGP=GIGENET,
               VNNAME=NETA.LVRN1, Connection network CP name
               VNTYPE=LOCAL       Connection network type
```

With this EE XCA major node, the following connections are provided for lines in group GPEE1:

- Inbound and outbound connection network links - the dynamically created PU will be named C1nnnnn
- Inbound and outbound non-connection network links when a matching active switched PU is available
- Inbound non-connection network links when a matching switched PU is not available - the dynamically created PU will be named P1nnnnn

While the defaults values are often sufficient, you can customize EE connection network PUs in the following ways:

- By default, dynamic connection network PUs take a name in the format CNV*xxxxx*, where *xxxxx* is a unique value that is generated and concatenated to the default CNV prefix to create the eight-character PU name. To specify a different prefix (two characters), use the DYNVNPFX start option or the DYNVNPFX operand on the EE XCA GROUP definition.
- You can create a model PU definition to customize the characteristics of dynamically created PUs. Use the DYNTYPE=VN operand for the model PU definition in the model major node.

Cross-subnet routing with global VRNs

With border node architecture, topology information about nodes, TGs, and their characteristics that determine weights, is not passed across subnet boundaries by way of topology database update (TDU) flows. When a session path is determined

to cross subnet boundaries, the optimal route (lowest weight) is calculated in each subnet. The complete end-to-end route consists of contiguous route segments which are individually optimal within their native subnet, but not necessarily optimal from end to end. Global virtual routing nodes (GVRNs) can be used to provide a more direct route between two endpoints in different subnets. Figure 53 shows a border node configuration with a GVRN. The weights in the figure are those displayed with DISPLAY NET,TOPO,ID=cpname,APPNCOS=cos or DISPLAY NET,TOPO,ORIG=orig,DEST=dest,APPNCOS=cos commands.

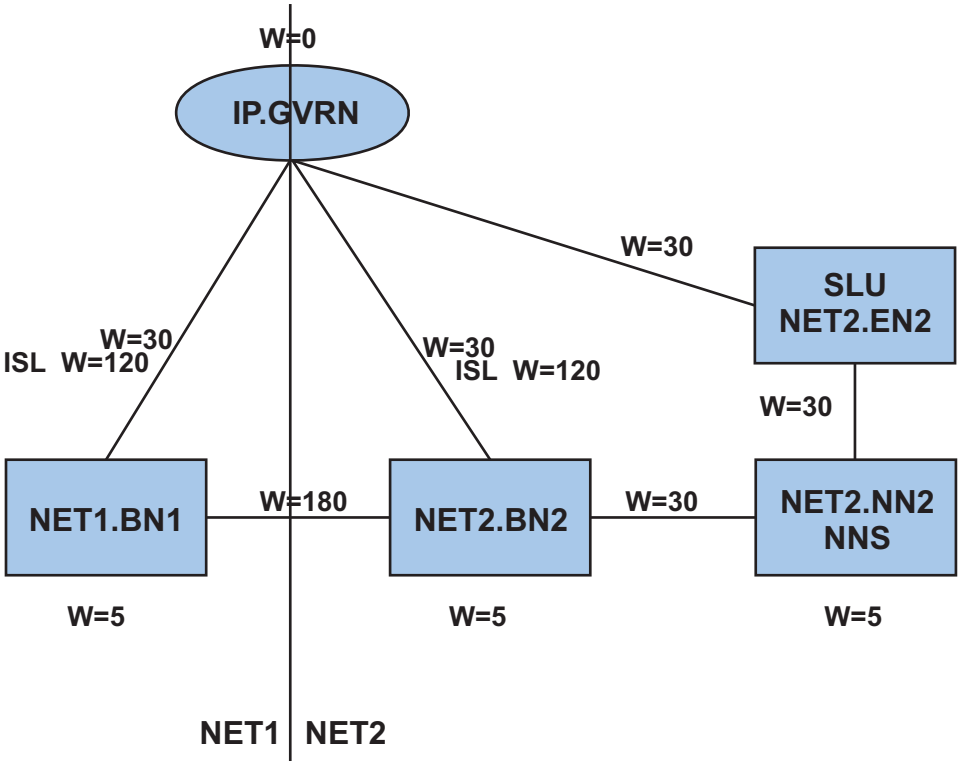


Figure 53. Global VRN with extended border nodes

A TG that originates in a border node and that has a GVRN as the destination can be used across a subnet boundary or used within a single subnet. (An intersubnet link (ISL) between two border nodes cannot be used in this way.) If the TG is used to calculate the session route and the PLU node that calculates the route is in another subnet, then the TG is being used as an intersubnet link (ISL). When the same TG is used within a single subnet, the TG is not considered to be an ISL.

In many ways, the border node in one subnet is treated the same way that an end node is, in that it is considered to be the endpoint of the route calculated in another subnet. For example, in the configuration shown in Figure 53, assume that the PLU is on NET1.BN1 and the SLU is on NET2.EN2. The only information passed to NET1.BN1 on a LOCATE would be:

- ISL TG between the two border nodes
- ISL TG from NET1.BN1 to the global VRN, IP.GVRN (because this TG is being passed cross-subnet, it is considered to be an ISL)
- TG from NET2.EN2 to the global VRN, IP.GVRN.

The ISL between the two border nodes and the ISL TG to IP.GVRN would contain the encapsulated best route from NET2.BN2 to NET2.EN2, but the LOCATE would

only include the TG characteristics (in a control vector x'47') for the ISL TGs themselves. Because the weights are determined by the TG characteristics in the x'47' control vectors, NET1.BN1 considers the two TGs to IP.GVRN to have equal weights of 30. Because the two weights are equal, and the weight of the ISL between the two border nodes has a higher weight of 180, one of the two lower weight routes is randomly selected.

For additional information about weight computation see “Example of weight computation” on page 340 and Appendix H, “Forcing an APPN route in a VTAM network,” on page 773.

Because you would normally want your session path to be the two hop route of NET1.BN1->IP.GVRN->NET2.EN2, there is a way to prefer this route over one that transverses an ISL, which might contain additional encapsulated session hops. z/OS Communications Server uses the COSTBYTE TG characteristic to prefer the more direct route. When a TG from a border node to a GVRN is being used as an ISL, the COSTBYTE value of 0 is temporarily changed to 1 (in the x'47' control vector when the TG control vectors are sent to another subnet in a LOCATE) to increase the weight of the TG. If the existing COSTBYTE value assigned to the TG is not 0, no change is made, and the weight will be the same whether the TG is used as an ISL.

The weights used in route calculation are displayed in the output of DISPLAY NET,TOPO,ORIG=orig,DEST=dest,APPNCOS=cos:

```
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP NET2.BN2
IST1357I                                          CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I IP.GVRN           21       OPER     INTERM      NO    30
IST1579I                   ----------------------------------------
IST2241I                                          TIME  ISL
IST1163I                   RSN              HPR         LEFT  WEIGHT
IST1164I                   0                YES         15    120
```

Unlike the ISL to a GVRN, where the COSTBYTE can be temporarily set to 1 when the TG is sent to another subnet in a LOCATE, the COSTBYTE of an ISL between two border nodes is defaulted to 1 when the ISL is activated. The COSTBYTE for this ISL can be changed from the default by specifying a COSTBYTE value on the PU definition for the ISL connection. This can lower the weight of the ISL, causing sessions to be routed through the direct ISL instead of taking the more optimal route through the GVRN.

When you display TG information for an ISL between two border nodes with the DISPLAY NET,TOPO,ORIG=orig,DEST=dest,APPNCOS=cos command, the weight that is displayed in messages IST1300I and IST1301I is the weight of the ISL when it is used cross-subnet:

```
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP NET2.BN2
IST1357I                                          CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I NET1.BN1          21       OPER     INTERCLUST  NO    30
IST1579I                   ----------------------------------------
IST2241I                                          TIME  ISL
IST1163I                   RSN              HPR         LEFT  WEIGHT
IST1164I                   0                YES         15    *NA*
```

Guideline: To favor the two hop GVRN route between cross-subnet session partners, you can override the COSTBYTE in a TGP assigned to a direct ISL between two border nodes. This will cause it to have a higher weight than the total

weight of a route through a GVRN between the two subnets, considering the ISL weight of a TG between a border node and a GVRN.

TG characteristics (COSTBYTE, COSTTIME, CAPACITY, etc.) can be assigned to a TG either individually or with a TG Profile (TGP), which contains a set of TG characteristics. The TG characteristics of the TGP assigned to an ISL are specified on the PU statement of the PU that is activated to bring up the ISL TG. The TG characteristics or the TGP assigned to a TG associated with a GVRN are specified on the PORT or GROUP statement that defines that GVRN in the Enterprise Extender (MEDIUM=HPRIP) XCA major node. See *z/OS Communications Server: SNA Resource Definition Reference* for additional information.

Troubleshooting EE problems

Table 8 shows some common EE problems you might encounter and suggested solutions to resolve them.

Table 8. Troubleshooting EE problems

Problem indication	Avoidance method or suggested remedy
Line activation failure: • Pending act link • PGAIN failure	When all of the rules for determining the local static VIPA address (or addresses) for EE are followed, EE line activation could still fail for one of the following reasons: • An incorrect TCP/IP stack name was specified with the TCPNAME VTAM start option. • An incorrect source VIPA address was specified with the IPADDR VTAM start option, or on the XCA GROUP definition statement. • An incorrect source VIPA address was resolved from the host name specified as the HOSTNAME VTAM start option, or on the XCA GROUP definition statement. • A host name could not be resolved for the specified source VIPA address. • The IPv4 or IPv6 same host device is not defined or is not started by the appropriate TCP/IP stack If the activation fails, perform one of the following recovery actions: • Stop TCP/IP, and activate the correct TCP/IP stack. Then LINE (thus, a PORT) activation should complete normally. • Individually deactivate (force) the LINEs. After the last LINE is deactivated, the PORT deactivates. Thus, the XCA major node is also deactivated. Then, change the TCPNAME, IPADDR, or HOSTNAME start option to the correct value, or update the XCA major node definition to specify the correct IPADDR or HOSTNAME value, and reactivate the XCA major node.

Table 8. Troubleshooting EE problems (continued)

Problem indication	Avoidance method or suggested remedy

Activation failure with the following messages:
`IST1411I INOP GENERATED FOR linename`
`IST1430I REASON FOR INOP IS`
`XID OR LDLC COMMAND TIMEOUT`
`IST314I END`
	This message group is issued when VTAM is not receiving responses to XID requests during activation. It indicates that either the partner is not responding to the request or there are connectivity problems within the IP infrastructure. Some common setup problems that cause this are the following:
	• IP connectivity has been lost within your network.
	• EE UDP ports are not defined with consistent values across the network. EE ports should be defined in the range of 12000 - 12004.
	• EE is not enabled on the remote endpoint. For example, lines and switched PUs are not defined, not activated, or both.
	• If the EE connection path traverses one or more firewalls, the firewalls must allow UDP traffic to flow for EE ports 12000 - 12004.
	• If NAT is used in the EE connection path, adhere to these rules:
	– Avoid NAPT. EE does not support NAPT.
	– When a one-to-one address translation function is performed, the name-to-address resolution mapping for the host name yields the correct NAT address.
	– If connection network is being used with NAT, you must use HOSTNAME definitions when defining your virtual routing node.
	Use the D NET, EEDIAG,TEST=YES command. Other helpful commands are PING, TRACERTE.

LU 6.2 sessions do not stay up over EE; your session unexpectedly ends	This problem usually indicates that a limited resource is in use somewhere along the session path. • For predefined EE connections, use DISCNT=NO (the default) • For EE VRN-based dynamic connections, consider coding a DYNTYPE=VN model with DISCNT=NO or a delay value of 60+ seconds. – Important note for CICS LU6.2 Users: Specifying DISCNT=NO prevents CICS from terminating its sessions at the end of every transaction.

Active EE connection unexpectedly fails with the messages:
`IST1411I INOP GENERATED FOR linename`
`IST1430I REASON FOR INOP IS`
`XID OR LDLC COMMAND TIMEOUT`
`IST314I END`
	EE connection deactivation due to LDLC timeout. EE periodically tests the EE partner to verify IP connectivity and that the partner is still there. When the tests are unanswered, the EE connection ends with these messages. Some common causes are:
	• The partner unexpectedly ended.
	• IP connectivity has been lost within your network.
	• OMPROUTE problems

Table 8. Troubleshooting EE problems (continued)

Problem indication	Avoidance method or suggested remedy
Diagnosis of connection failures or performance considerations	Use the TCP/IP packet trace to analyze Enterprise Extender-related packets flowing to and from a TCP/IP stack on a z/OS Communications Server host. You can use the PKTTRACE statement to copy IP packets as they enter or leave TCP/IP, and then examine the contents of the copied packets. The method of capturing and formatting this trace is described in *z/OS Communications Server: IP Diagnosis Guide*.
Dial collision problems	See Dial collision discussion in Dial usability - DWACT, DWINOP, KEEPACT, REDIAL, and REDDELAY for more information.
Poor throughput when using PSRETRY	After each path switch, HPR resets its sending rate to the initial value so frequent path switches can lead to reduced throughput. In particular, setting PSWEIGHT to EQUAL or SAMEROUT can lead to an excessive number of path switches. See *z/OS Communications Server: SNA Resource Definition Reference* for information on the START option.
Poor HPR throughput over EE with multipath enabled	If MULTIPATH is enabled on the TCP/IP stack, the VTAM start option MULTPATH is set to TCPVALUE and multiple equal-cost routes exist to the partner EE node, then TCP/IP will round-robin batches of EE packets across each of these routes. If one of these routes cannot reach the partner EE node, then EE may not activate, or if it does, there can be significant performance impacts. See *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for information about DISPLAY commands. Code or let start option MULTPATH default to NO. This will disable multipath routing for only EE traffic in the TCPIP stack. This can be updated dynamically using the MODIFY VTAMOPTS command.
High CPU utilization in a branch environment (with lots of EE connections always active)	If you are using many Enterprise Extender connections and these connections are kept indefinitely active (switched PU associated with the EE connection has a DISCNT=NO, either specified or defaulted), the following functions can reduce the CPU utilization by VTAM: • HPR alive timer optimization - This function by default is enabled and is controlled by the VTAM start option HPREELIV. For full details of this function, see HPR ALIVE timer optimization for Enterprise Extender. • LDLC Keep-Alive reduction - This function requires you to specify an operand for the LIVTIME Enterprise Extender PORT option. For full details of this function, see Enterprise Extender LDLC keep-alive reduction.

Table 8. Troubleshooting EE problems (continued)

Problem indication	Avoidance method or suggested remedy
EE connections through the connection network are not rerouting to an alternate path	If the EE connection network path has the lowest weight of any available path to the partner node, any attempt to redial the partner node will continue to try the path over this particular VRN. This is likely to result in failures until the underlying problem with the path is corrected. EE connection network reachability awareness is designed to detect the dial failure or connection INOP for the connection over an Enterprise Extender connection network and prevent that specific path to the partner node from being used for a period of time. Use the EE connection network reachability awareness function to indicate that the path to a partner node over an Enterprise Extender VRN should not be used for route selection for a period of time after the initial dial failure or connection INOP, providing time for the underlying connection problem to be corrected. This function can be enabled in the following manner: • Specify the UNRCHTIM start option • Specify the UNRCHTIM operand on either the EE XCA major node PORT or GROUP definition statements
A new EE connection is established between you and a partner company but sessions cannot be established.	This is generally because the firewalls are not allowing UDP traffic on all EE ports. The firewall must allow UDP traffic both inbound and outbound on all five EE ports (12000 - 12004). To assist in diagnosing new EE connection problems, use this command: D NET, EEDIAG, TEST=YES. See *z/OS Communications Server: SNA Operation* for more information.
A new EE connection fails with the EE health verification failure message: `IST2330I EE HEALTH VERIFICATION FAILED FOR puname AT time`	This is generally because the firewalls or intermediate routers are not allowing UDP traffic on all EE ports. Any firewall or intermediate router must allow UDP traffic both inbound and outbound on all five EE ports, typically 12000 - 12004. To assist in diagnosing new EE health verification failures, use the following command: `D NET,EEDIAG,TEST=YES` See *z/OS Communications Server: SNA Operation* for more information. Review the Enterprise Extender Connectivity Test output for any unsuccessful test results. See DISPLAY EEDIAG,TEST=YES in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for information about analyzing the test output.

Table 8. Troubleshooting EE problems (continued)

Problem indication	Avoidance method or suggested remedy
EE health verification fails on active connections. The following eventual action message is issued on the console: `IST2323E EE HEALTH VERIFICATION FAILED FOR ONE OR MORE CONNECTIONS`	Use the DISPLAY NET,EE,LIST=EEVERIFY command to determine which EE connections are experiencing EE health verification failures. Message IST2325I is displayed for each line or PU, which failed health verification on the most recent LDLC probe to its remote partner. Use the DISPLAY NET,ID=linename or puname command to get more information that includes local and remote IP addresses. Determine the network connectivity problems between this node and the remote partners. Use the DISPLAY NET,EEDIAG,TEST=YES command to determine the reason of the failure. See DISPLAY EEDIAG,TEST=YES in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for more information.
A new EE connection comes up with the EE health verification failure not supported message: `IST2342I EE HEALTH VERIFICATION NOT SUPPORTED BY` *puname*	During the activation of the EE connection, VTAM sends Logical Data Link Control (LDLC) probes to the remote partner to determine whether all five ports are accessible. VTAM does not receive a response to any of the LDLC probe requests. VTAM continues with the activation of the EE connection between this node and the remote partner. Because VTAM receives no replies to its LDLC probe requests, VTAM determines that the remote partner does not support EE health verification. If EE health verification is required for this PU, contact the remote PU owner about upgrading the PU to support EE health verification probes. If you think that EE health verification supported by the remote partner, use the following command: `D NET,EEDIAG,TEST=YES` See *z/OS Communications Server: SNA Operation* for more information. Review the Enterprise Extender Connectivity Test output for any unsuccessful test results. See DISPLAY EEDIAG,TEST=YES in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for information about analyzing the test output.
The EE connection link terminates due to XID or LDLC timeout.	Consider tuning the LDLC parameters as described in When does the EE connection go away?. Also consider using the D NET, EEDIAG, SRQRETRY command.
The RTP pipe fails to successfully path switch even though an alternate link is available	Due to a problem with the EE connection, an HPR pipe attempts to pathswitch but fails to connect with a message that no alternate routes are available. Ensure that values in the HPRPST start option are all greater than the EE link inoptime. See When does the EE connection go away? for more information.
Excessive path switch messages (IST1494I) flooding the system console log during large network outage.	Enable the HPR path switch message reduction function with the HPRPSMSG start option. For more information about the HPRPSMSG start option, see *z/OS Communications Server: SNA Resource Definition Reference*.

Table 8. Troubleshooting EE problems (continued)

Problem indication	Avoidance method or suggested remedy
You cannot determine the APPNCOS name associated with an RTP *puname* that unexpectedly deactivates.	Enhance the HPR activation and deactivation messages by setting the HPRITMSG start option to the value ENHANCED. Now, when an RTP is deactivated, you can locate the IST1488I (deactivation) message group on the system console log. You can then find the associated APPNCOS in messages IST1962I, IST1963I, IST1964I, or IST1965I. For more information about the HPRITMSG start option, see *z/OS Communications Server: SNA Resource Definition Reference*.

RTP transmission stalls repeatedly for pipes which use Enterprise Extender. The following messages are displayed:
`IST2245I XMIT STALL DETECTED FOR RTP`
`puname TO cpname`
 	
If the stall persists, VTAM issues the following message every 30 seconds:
`IST2246I XMIT STALL CONTINUES FOR RTP`
`puname TO cpname`
 	
If the transmission stall clears, VTAM issues the following message:
`IST2247I XMIT STALL ALLEVIATED FOR RTP`
`puname TO cpname`
 	
If the transmission stall extends beyond the time limit specified by the HPRSTALL VTAM start option, VTAM automatically initiates termination of the HPR pipe and issues the following message:
`IST2253I HPRSTALL TIME EXCEEDED FOR RTP`
`puname TO cpname`
	If path MTU discovery is enabled for IPv4 or IPv6 Enterprise Extender connections, and firewalls are used in the configuration, verify that the firewalls are configured to allow ICMP errors to flow on all hops of the connection.
	
	If problems persist, you may consider disabling path MTU discovery for IPv4 Enterprise Extender connections. This can be done by specifying PMTUD=NO in the appropriate ATCSTRxx VTAM start list or on the VTAM START command. Optionally, when VTAM is active, issue the MODIFY *procname*,VTAMOPTS, PMTUD=NO command.
	
	If path MTU discovery is not enabled for IPv4 Enterprise Extender connections, but you still suspect this is an MTU issue, you may consider limiting the maximum packet size which Enterprise Extender will transmit. Consider specifying the MTU operand available on these major nodes:
	• For EE connection networks, this parameter may be defined on the connection network GROUP definition statements in the EE XCA major node.
	• For dial in Enterprise Extender connections which have their associated PUs dynamically created, this parameter may be defined on the model major node (DYNTYPE=EE) PU definition statement.
	• For predefined Enterprise Extender connections, this parameter may be defined on the PU definition statement in the switched major node.

Chapter 7. OSA-Express

This topic provides an overview of OSA-Express for z/OS Communications Server. An OSA is an integrated S/390 or zSeries hardware feature that combines the functions of an S/390 or zSeries I/O channel with the functions of a network port to provide direct connectivity between S/390 or zSeries applications and their clients on the attached network.

For additional information about planning for and installing OSA-Express, refer to either the OSA-Express customer's guide and reference in your hardware library or the *zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference*.

OSA-Express overview

OSA-Express is an integrated hardware feature that provides direct connection to clients on local area networks (LANs). The OSA-Express feature plugs into an I/O slot just like a channel card. Fiber-optic cable connects the OSA-Express feature to the network.

OSA-Express supports direct attachment to Ethernet, token ring, and ATM LANs where clients communicate using SNA or TCP/IP. OSA-Express supports the same interfaces for Ethernet and ATM as the IBM S/390 or zSeries Open Systems Adapter (OSA). For details on using either OSA or OSA-Express for SNA, see "IBM Open Systems Adapter connections between APPN nodes" on page 68 and "Connecting two VTAMs using an external communication adapter" on page 112. For details on using either OSA or OSA-Express for TCP/IP, refer to the *z/OS Communications Server: IP Configuration Reference*. For TCP/IP, OSA-Express can be configured to use IBM Queued Direct I/O (QDIO) architecture to eliminate the need for channel control words (CCWs) and interrupts, resulting in accelerated TCP/IP data packet transmission.

OSA-Express supports QDIO for:

- Gigabit Ethernet
- Ethernet
- ATM LAN emulation
- Token ring

Ethernet links provide support for both standard Ethernet (10 Mbps) and fast Ethernet (100 Mbps).

For a summary of OSA-Express support, see Table 9 on page 211.

For details on the hardware and software required to use the OSA-Express feature, refer to the OSA-Express customer's guide and reference in your hardware library, either *zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference*.

To use the OSA-Express feature to communicate using TCP/IP, IBM recommends defining the device to use the QDIO interface. To do this, you must code a TRL definition that contains a QDIO TRLE. To define the OSA-E for IPv4, use device type MPCIPA or interface type IPAQENET in the TCP/IP profile. To define the

© Copyright IBM Corp. 1984, 2011

OSA-E for IPv6, use interface type IPAQENET6 in the TCP/IP profile.

Table 9. OSA-Express support

OSA feature	OSA mode [1]	OAT type (Input)	OAT type (Display)	VTAM TRLE resource definition MPCLEVEL[3]	TCP/IP profile (device type) [2]	TCP/IP profile (link type) [2]	TCP/IP profile (interface type)[4]	VTAM resource definitions	NDS definitions	Protocol
OSA-Express Gigabit Ethernet	QDIO (IP) [a]	N/A	MPC (QDIO)	QDIO[k]	MPCIPA[i]	IPAQENET[j]	IPAQENET[l]	N/A	N/A	IP
OSA-Express ATM 155 (using TR or ENET LANE)	TCP/IP Passthru [b]	Passthru	Passthru	N/A	LCS[i]	ETHERNet, 802.3, ETHERor802.3, IBMTR[j]	N/A	N/A	N/A	IP
OSA-Express ATM 155 (using TR or ENET LANE)	SNA [b]	SNA	SNA	N/A	N/A	N/A	N/A	XCA/ SWNET	N/A	SNA
OSA-Express ATM 155 (using ENET LANE)	QDIO (IP) [c]	N/A	MPC (QDIO)	QDIO	MPCIPA[i]	IPAQENET[j]	N/A	N/A	N/A	IP
OSA-Express ATM 155	HPDT ATM Native [d]	MPC	MPC	HPDT	ATM[i]	ATM[j]	N/A	N/A	N/A	IP
OSA-Express ATM 155	HPDT ATM Native [d]	MPC	MPC	HPDT	N/A	N/A	N/A	XCA/ SWNET	N/A	SNA
OSA-Express FENET	TCP/IP Passthru [e]	Passthru	Passthru	N/A	LCS[i]	ETHERNet, 802.3, ETHERor802.3[j]	N/A	N/A	N/A	IP
OSA-Express FENET	SNA [e]	SNA	SNA	N/A	N/A	N/A	N/A	XCA/ SWNET	N/A	SNA
OSA-Express FENET	HPDT MPC (IP) [e]	MPC	MPC (IP)	HPDT	MPCOSA[i]	OSAENET[j]	N/A	N/A	N/A	IP
OSA-Express FENET	HPDT MPC (IP) [e]	MPC	MPC (IP)	HPDT	N/A	N/A	N/A	N/A	N/A	IP
OSA-Express FENET	QDIO (IP) [f]	N/A	MPC (QDIO)	QDIO	MPCIPA[i]	IPAQENET[j]	N/A	N/A	N/A	IP
OSA-Express TR	TCP/IP Passthru [g]	Passthru	Passthru	N/A	LCS	IBMTR	N/A	N/A	N/A	IP
OSA-Express TR	SNA [g]	SNA	SNA	N/A	N/A	N/A	N/A	XCA/ SWNET	N/A	SNA

Table 9. OSA-Express support (continued)

OSA feature	OSA mode [1]	OAT type (Input)	OAT type (Display)	VTAM TRLE resource definition MPCLEVEL[3]	TCP/IP profile (device type) [2]	TCP/IP profile (link type) [2]	TCP/IP profile (interface type)[4]	VTAM resource definitions	NDS definitions	Protocol
OSA-Express TR	QDIO (IP) [h]	N/A	MPC (QDIO)	QDIO	MPCIPA	IPAQTR	N/A	N/A	N/A	IP

Notes:

1. The superscripts a through h in the OSA mode column indicate which modes can run concurrently on a single OSA feature. Modes with identical superscripts can run concurrently on a single OSA feature.
2. The superscripts i and j in the device and link type columns represent the IPv4 definitions. To define the OSA for IPv6, use the interface type IPAQENET6 in the TCP/IP profile.
3. The superscript k in the VTAM TRLE resource definition MPCLEVEL column indicates that a VTAM TRLE is not configured for QDIO OSM interfaces or for QDIO OSX interfaces that are defined with the CHPID parameter.
4. The superscript l in the interface type column represents IPv4 definitions that might be used as an alternative to device and link. To define the OSA for IPv6, use the interface type IPAQENET6 in the TCP/IP profile.

Defining an OSA-Express device to z/OS Communications Server using QDIO

To define an OSA-Express device to z/OS Communications Server using queued direct I/O (QDIO), first you need to define a QDIO transport resource list element (TRLE). This is shown in the following example.

```
TRLHYDRA VBUILD TYPE=TRL
*********************************************************************
*  TRANSPORT RESOURCE LIST FOR OSA-EXPRESS
*********************************************************************
*        10    16
HYD1     TRLE  LNCTL=MPC,                                            *
               READ=(2EC0),                                          *
               WRITE=(2EC1),                                         *
               MPCLEVEL=QDIO,                                        *
               DATAPATH=(2EC2,2EC3),                                 *
               PORTNAME=(HYD1G1),                                    *
               PORTNUM=1
```

In this example, MPCLEVEL=QDIO indicates that the direct I/O interface is used. Control data is transmitted across the one READ and one WRITE device. Normal data is transmitted across a DATAPATH device, where each ULP is assigned its own channel unit address. If you define multiple VLANs to the same OSA, you need to configure a separate INTERFACE definition in TCP/IP for each VLAN and a separate DATAPATH device is needed for each of these interfaces. When MPCLEVEL is QDIO, HPDT MPC is also used for this connection across the control channels. The port number (PORTNUM) specifies which physical port on an OSA-Express is to be used for this QDIO device. For OSA-Express and OSA-Express2, only one port, port number zero, is supported for each CHPID. For OSA-Express3, multiple ports are supported on each CHPID. Within a single logical partition (LPAR), each port must have its own unique TRLE definition, with unique read, write, and datapath channel unit addresses. The port name (PORTNAME) is the name that will be assigned to this port. All z/OS LPARs sharing the same port of an OSA-Express must define the same PORTNAME and PORTNUM on their TRLE definitions representing that OSA-Express port.

See *z/OS Communications Server: SNA Resource Definition Reference* for more information about coding a TRLE statement.

To define the OSA for IPv4, you must next define either a DEVICE and LINK statement or an INTERFACE statement in your TCP/IP profile. This is illustrated in the following example.

```
DEVICE HYD1G1 MPCIPA
LINK LHYDRA IPAQENET HYD1G1
```

In this example, HYD1G1 is the name of the device. The device name must be the port name as defined in a TRLE for a QDIO connection. IPAQENET defines an IPv4 Ethernet interface.

```
INTERFACE QDIOINTF4 IPAQENET PORTNAME HYD1G1
```

In this example, the PORTNAME value (HYD1G1) must be the port name in the TRLE definition. IPAQENET defines an IPv4 Ethernet interface.

To define the OSA for IPv6, you need an INTERFACE statement in the TCP/IP profile. This is illustrated in the following example.

```
INTERFACE HYD1G16 DEFINE IPAQENET6 PORTNAME HYD1G1
```

In this example, the PORTNAME value (HYD1G1) must be the port name in the TRLE definition. IPAQENET6 defines an IPv6 Ethernet interface.

See *z/OS Communications Server: IP Configuration Reference* for more information about coding the DEVICE, LINK, and INTERFACE statements.

To continue defining an OSA-Express device, perform the following steps:

1. Configure and vary the physical devices online.
2. Activate the TRL deck.
3. Define and start the device or interface, or both, using TCP/IP.

Note: Datapath channels for OSA-Express should not be varied offline while VTAM is active.

When z/OS Communications Server participates in an ensemble environment, QDIO TRLE definitions are dynamically generated for connectivity to the intraensemble data network (IEDN) (CHPID type OSX) if the QDIO interface is defined with CHPIDTYPE OSX and the CHPID parameter. For each dynamic TRLE for OSX, the TRLE name is IUTXT0*xx* and the corresponding port name is IUTXP0*xx* (where *xx* is the configured CHPID parameter). QDIO TRLE definitions are also dynamically generated for connectivity to the intranode management network (CHPID type OSM) if OSM CHPIDs are configured. VTAM searches for two OSM CHPIDs when the TCP/IP stack is initialized, and dynamically generates a TRLE for each CHPID. The TRLE names are IUTMT0*xx* and the corresponding port names are IUTMP0*xx*, where *xx* is the OSM CHPID that is found. See "Resources automatically activated by VTAM" on page 612 for more information about the naming convention of these TRLEs and the number of DATAPATH devices that are dynamically defined.

Tip: If the values that are chosen for the dynamically created OSX TRLE definitions do not suit your needs (for example, you need more than 17 DATAPATH devices), you can define your own QDIO TRLE and configure the QDIO interface with the PORTNAME parameter. See INTERFACE -- IPAQENET OSA-Express QDIO interfaces statement and INTERFACE -- IPAQENET6 OSA-Express QDIO interfaces statement in *z/OS Communications Server: IP Configuration Reference* for more information.

Restrictions:

- If the QDIO interface that is used for connectivity to the intraensemble data network is defined with the PORTNAME parameter, you must define a QDIO TRLE definition. The PORTNAME parameter value on the TRLE definition must match the PORTNAME parameter value. For information about restrictions on the TRLE name, see Transport resource list major node in *z/OS Communications Server: SNA Resource Definition Reference*.
- If z/OS Communications Server runs as a guest on z/VM® and guest LAN definitions is used for connectivity to the intraensemble data network, you must define the QDIO interface with the PORTNAME parameter.

QDIO TRLE definitions are always dynamically generated for connectivity to the intranode management network (CHPID type OSM). For each dynamic TRLE for OSM, the TRLE name is IUTMT0*xx* and the corresponding portname is IUTMP0*xx* (where *xx* is the value of the OSM CHPID). See TCP/IP in an ensemble in *z/OS Communications Server: IP Configuration Guide* for more information.

OSA routing

For QDIO devices, z/OS Communications Server and OSA-Express provide functions that control how incoming unicast datagrams are routed, especially when the OSA-Express is shared by multiple TCP/IP instances.

When TCP/IP activates a QDIO device, each TCP/IP registers each of its home IP addresses with OSA-Express. (TCP/IP also dynamically registers any updates to its set of home IP addresses with OSA-Express.) This allows OSA-Express to route datagrams destined for a registered IP address to the correct TCP/IP instance.

However, when packets are received for IP addresses that are not registered by any TCP/IP stack, the device needs a method for determining which stack, if any, should receive the packet. Two functions are available to accomplish this; OSA-Express Virtual MAC (VMAC) routing, and primary and secondary routing.

OSA-Express virtual MAC (VMAC) routing

If multiple TCP/IP instances are sharing an OSA-Express, the preferred method of routing is to define or generate a VMAC for each stack for each protocol being used (IPv4 or IPv6). For IPv4, this results in the OSA-Express using the VMAC address rather than the physical "burned in" MAC for all ARPs sent for that TCP/IP stack's registered IP addresses, and using the VMAC as the source MAC address for all packets sent from that stack. In this way, all routers on the same LAN as the OSA-Express will use only the VMAC address as the destination for all packets destined for that specific TCP/IP stack. From a network routing perspective, the OSA-Express with this VMAC will appear as a "dedicated" device to that TCP/IP stack.

This simplifies a shared OSA configuration significantly. The OSA-Express knows by VMAC address exactly which stack should receive a given packet. Even if the IP address is not registered with the OSA-Express, if the packet is destined for that VMAC then the router has determined which stack should be the intermediate router, and the OSA can forward the packet directly to that stack. The capability is given for a stack to indicate to the OSA it only wishes to receive packets to registered IP addresses.

The above simplication is true for IPv6 as well. TCP/IP will use the VMAC address for all neighbor discovery address resolution flows for that stack's IP addresses, and will likewise use the VMAC as the source MAC address for all IPv6 packets sent from that stack. Again, from a network perspective, the OSA-Express with this VMAC will appear as a "dedicated" device to that stack.

The VMAC address may be defined in the stack, or if desired may be generated by the OSA. If generated by the OSA, it is guaranteed to be unique from all other physical MAC addresses and from all other VMAC addresses generated by any OSA-Express feature.

The following figure illustrates how OSA-Express virtual MAC routing works:

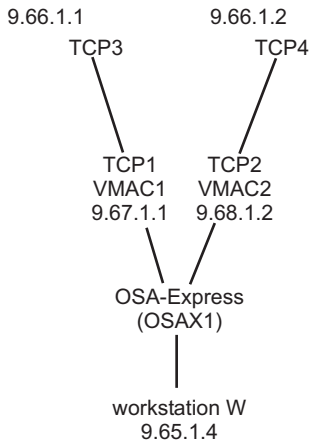


Figure 54. OSA-Express virtual MAC routing

In this example, both TCP1 and TCP2 activate OSA-Express device OSAX1. TCP1 is defined with VMAC1 and TCP2 is defined with VMAC2. TCP1 registers each of its home IP addresses (just 9.67.1.1 in this example) to the OSA-Express device when the device is started on TCP1. The OSA-Express device indicates in all ARP processing that IP address 9.67.1.1 can be reached by using VMAC1. Similarly, TCP2 registers each of its home IP addresses (just 9.68.1.2 in this example) to the OSA-Express device when the device is started on TCP2, and again the OSA-Express device indicates in all ARP processing that IP address 9.68.1.2 can be reached by using VMAC2.

If device OSAX1 receives any datagram destined to VMAC1, the OSA-Express routes that datagram to TCP1. So if routers determined either that IP address 9.66.1.1 or 9.67.1.1 could be reached by VMAC1, TCP1 receives packets destined to those addresses. Likewise, if device OSAX1 receives any datagram destined to VMAC2, the OSA-Express device routes those datagrams to TCP2. Therefore, if routers determined that either IP address 9.66.1.2 or 9.68.1.2 could be reached by VMAC2, TCP2 receives packets destined to those addresses.

Rule: If VMACs are defined in the stack, they should be defined as locally administered MAC addresses, and should be a unique address for the local LAN on which they reside. For IPv4, the VMAC is defined on the LINK statement of the MPCIPA type DEVICE or the IPAQENET type INTERFACE statement representing the shared OSA-Express. For IPv6, the VMAC is defined on the IPAQENET6 type INTERFACE statement representing the shared OSA-Express.

Guideline: If the OSA is configured for both IPv4 (using DEVICE/LINK) and IPv6 for a stack, then the same VMAC may be defined for both the INTERFACE statement and the LINK statement, or a stack may use one VMAC on the LINK statement for IPv4 usage, and one VMAC on the INTERFACE statement for IPv6 usage. If the OSA is configured for both IPv4 (using INTERFACE) and IPv6 for a stack, then the stack must use one VMAC on the INTERFACE statement for IPv4 usage, and one VMAC on the INTERFACE statement for IPv6 usage.

Guideline: A VLAN id may be associated with an OSA-Express link or interface defined with a VMAC.

You can define multiple interfaces to the same OSA for the same IP version by configuring multiple VLANs. To do this, configure a unique VLAN ID for each

IPv4 interface over the OSA-Express and a unique VLAN ID for each IPv6 interface over the OSA-Express and configure (or generate) a unique VMAC for each INTERFACE statement.

See *z/OS Communications Server: IP Configuration Reference* for how to define VMACs and VLANs in TCP/IP. See *z/OS Communications Server: IP Configuration Guide* for more considerations when using an OSA-Express configured in QDIO mode and for information about using multiple VLANs.

Refer to *zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference* for the level of OSA-Express2 that supports OSA-Express Virtual MACs.

Primary and secondary routing

The following information applies to IPv4 datagrams. For primary and secondary router specifications for IPv6 datagrams, refer to the INTERFACE statement contained in the *z/OS Communications Server: IP Configuration Reference*.

If only one TCP/IP instance is using the OSA-Express, or when multiple TCP/IP instances are using the same OSA-Express device, and you wish all instances to share the same physical MAC address of the device, you can optionally have the OSA route packets to unregistered IP addresses by designating a TCP/IP instance as the primary router or secondary router. For a given OSA-Express, only one TCP/IP instance can be registered as the primary router and one or multiple TCP/IP instances can be registered as the secondary router, depending on the level of the OSA-Express. (Refer to the description of the PRIROUTER and SECROUTER parameters on the DEVICE statement for MPCIPA in the *z/OS Communications Server: IP Configuration Reference* for details on how to designate a primary or secondary router.) If you define a TCP/IP instance as a primary or secondary router, you must also enable IP forwarding (using IPCONFIG DATAGRAMFWD in the TCP/IP profile).

When the device receives a datagram, OSA-Express routes the datagram as follows:

- If the datagram is destined for a registered IP address, OSA-Express routes the datagram to the TCP/IP instance that registered the IP address.
- If the datagram is destined for an unregistered IP address, OSA-Express routes the datagram to the TCP/IP instance that is registered as the primary router. If no TCP/IP instance is registered as the primary router, OSA-Express routes the datagram to one of the TCP/IP instances that is registered as the secondary router. If no TCP/IP instance is registered as a secondary router, OSA-Express discards the datagram.

Note: If a device becomes inactive for any reason, then OSA-Express unregisters any IP addresses that had been registered.

With QDIO, each TCP/IP instance using the device automatically registers its entire home list (including VIPAs) with OSA-Express, and OSA-Express automatically creates and initializes the OSA Address Table (OAT). There is no need for the customer to configure any home IP addresses in the OAT as the customer must do when using an OSA-2 in port sharing mode as an LCS device.

The QDIO primary and secondary routing function can provide fault tolerance without using virtual IP addresses (VIPA).

The following exampleFigure 55 illustrates how the QDIO primary and secondary routing works:

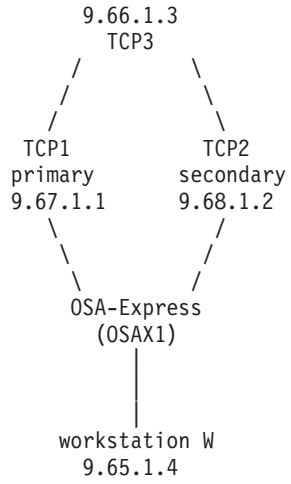


Figure 55. QDIO primary and secondary routing

In this example, both TCP1 and TCP2 activate OSA-Express device OSAX1. TCP1 is defined as the primary router and TCP2 is defined as a secondary router. TCP1 registers each of its home IP addresses (just 9.67.1.1 in this example) to OSA-Express when the device is started on TCP1. Similarly, TCP2 registers each of its home IP addresses (just 9.68.1.2 in this example) to OSA-Express when the device is started on TCP2.

If device OSAX1 receives an inbound datagram destined for 9.67.1.1, then OSA-Express will route the datagram to TCP1. If the device receives a datagram destined for 9.68.1.2, then OSA-Express will route the datagram to TCP2. If the device receives a datagram destined for any other (unregistered) IP address such as 9.66.1.3, then OSA-Express will route the datagram to TCP1 because TCP1 is the primary router. If TCP1 goes down or device OSAX1 becomes inactive on TCP1, then OSA-Express will route such datagrams to TCP2. If the device subsequently becomes active again on TCP1, then OSA-Express will once again route datagrams for unregistered IP addresses to TCP1.

IP traffic from workstation W destined for 9.66.1.3 will go to TCP3 through TCP1 (because the default router is the primary router TCP1). If device OSAX1 becomes inactive on TCP1 for any reason (or TCP1 goes down), then TCP2 (secondary router) becomes the default router and the IP traffic will go to TCP3 through TCP2. If device OSAX1 becomes active again on TCP1, then TCP1 once again becomes the default router and the IP traffic will now go through TCP1.

Note: If you are using static routing, then in order for the above to work, you need to ensure that you have static routes defined in both directions between TCP3 and the workstation (9.65.1.4) through both TCP1 and TCP2.

For more information on primary and secondary routing, refer to*zEnterprise 196, System z10, System z9 and eServer zSeries OSA-Express Customer's Guide and Reference*.

You can also use VIPA to provide fault tolerance. Refer to information about Virtual IP Addressing (VIPA) in the *z/OS Communications Server: IP Configuration Guide* for more details on VIPA.

Outbound priorities

For QDIO devices, z/OS Communications Server and OSA-Express provide a function that assigns a priority value to each outbound datagram and that attempts to provide preferential service to the higher-priority data.

z/OS Communications Server supports four priority values in the range 1–4 for outbound QDIO traffic (with 1 being the highest priority).

TCP/IP uses the first three bits of the Type of Service (ToS) byte in the IP header to determine the outbound priority value for a given datagram. The default mapping of ToS values to priorities is:

ToS	Priority
000	4
001	4
010	3
011	2
100	1
101	1
110	1
111	1

You can use the z/OS UNIX Service Policy Agent to override the default mapping of ToS values to priorities. (Refer to the description of the SetSubnetPrioTosMask statement in *z/OS Communications Server: IP Configuration Guide* for details on how to override the default priority for a given ToS setting.)

Note: You cannot use a virtual IP address (VIPA) as the SubnetAddr value on the SetSubnetPrioTosMask statement.

MTU

The maximum transmission unit (MTU) for QDIO gigabit Ethernet devices is 8992. The MTU for QDIO fast Ethernet devices is 1492.

Chapter 8. Sharing NCP peripheral nodes (CMC configurations)

When you plan to share NCP resources:

- Provide a data host to back up a communication management configuration (CMC) host.
- Code the PCCU definition statements in the NCP definitions filed at the CMC hosts.
- Code the HOST definition statements in the NCP definitions filed at the CMC hosts.

Figure 56 on page 222 shows a possible CMC. In this figure, NCP1 is channel-attached to HOST1, but it is in HOST2 domain. NCP1 is a channel-attached cross-domain NCP with respect to HOST1. The same relationship applies between NCP3 and HOST3. All three NCPs are controlled by HOST2, the CMC host, which means HOST1 and HOST3 can contact their channel-attached NCPs and communicate with resources attached to them without activating those NCPs, which is normally referred to as a data host connection. CMC HOST2 is responsible for activating all three NCPs. You must define the channel connection between the data hosts and NCPs in a channel-attachment major node. However, the VTAMs in the data hosts do not require the NCP load module, the NCP resource resolution table (RRT), or any NCP major node definitions in VTAMLST. All network problems are handled by HOST2. HOST1 and HOST3 (designated as data hosts) do not own any of the network resources. All LU-LU sessions to remote network resources are cross-domain sessions.

The channel connection between HOST2 and NCP1 is defined to HOST2 in a channel-attachment major node. The normal operating activation procedure for a CMC configuration consists of the following steps:

1. HOST2 activates NCP1.
2. HOST1 activates its channel-attachment major node to contact NCP1.

If CMC HOST2 fails and must be backed up by data HOST1, HOST1 activates NCP1. In this case, HOST1 needs a copy of the NCP source and RRT. Activating the NCP establishes an SSCP-PU session between HOST1 and NCP1; HOST1 becomes the new CMC host. HOST3 can provide backup for HOST2 in much the same way. However, for either data host to back up the CMC, VTAM must be capable of accessing the NCP major node definitions in VTAMLST and a copy of the NCP RRT. If either data host also needs to load or dump the NCPs in backup situations, VTAM needs access to the NCP load modules, service switching point (SSP), and dump data sets.

© Copyright IBM Corp. 1984, 2011

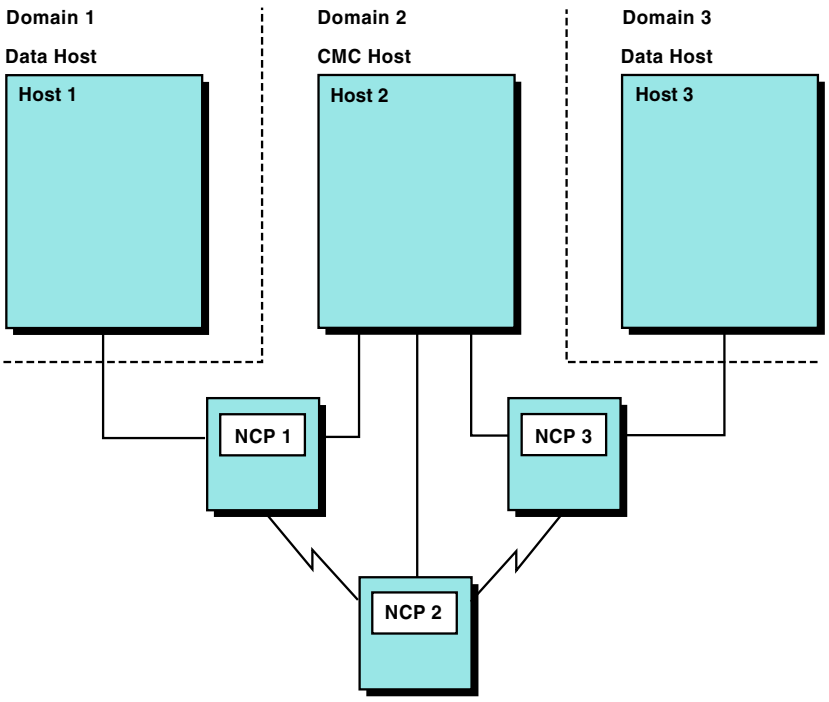


Figure 56. Example of communication management configuration

For further information on CMC configurations, see:

- "Channel-attached NCPs in a CMC configuration"
- "Link-attached NCPs in a CMC configuration" on page 224
- "Controlling resource ownership in a CMC configuration" on page 224
- "Problem determination in a CMC configuration" on page 225

Note: The NCP must be loaded and active if VTAM is to contact the NCP but not establish an SSCP-PU session with it.

If the NCP is deactivated with the VARY INACT command at the CMC and the channel-attached major node is active, sessions using the channel path to the data host continue uninterrupted.

Channel-attached NCPs in a CMC configuration

Several operands on the PCCU definition statement are used to define channel characteristics. If you do not define a channel in a channel-attachment major node, the values on the PCCU definition statement are used because the channel is implicitly defined and activated automatically when the NCP is activated.

However, when you define the channel in a channel-attachment major node, the definition statements in the channel-attachment major node contain operands similar to those on the PCCU definition statements. These operands are the following:

- CHANCON=***COND*** | UNCOND

  For a data host, the CHANCON operand on the PCCU definition statement is replaced by the CHANCON operand on the LINE definition statement in the channel-attachment major node.
- CUADDR=*channel device address*

For a data host, the CUADDR operand on the PCCU definition statement is replaced by the ADDRESS operand on the LINE definition statement in the channel-attachment major node.

- MAXDATA=*size*

  For a data host, the MAXDATA operand on the PCCU definition statement is replaced by the MAXDATA operand on the PU definition statement in the channel-attachment major node.

- DELAY=*time*

  For a data host, the DELAY operand on the PCCU definition statement is replaced by the DELAY operand on the PU definition statement in the channel-attachment major node.

Activating the channel-attachment major node to activate the channel causes VTAM to use the values defined in the channel-attachment major node. Later, if NCP1 is activated, these same operands, if they appear on NCP1 PCCU definition statements, are ignored. If you are defining the channel in a channel-attachment major node and want to avoid confusion concerning these operands, avoid coding duplicate values on the PCCU definition statement.

The RNAME operand on the PCCU definition statement can name a link station associated with a channel link defined in a channel-attachment major node. When the host activates each NCP, the link station and link defined in the channel-attachment major node representing the channel are activated automatically.

The values defined on the PU definition statement in the channel-attachment major node are used for the CUADDR, CHANCON, DELAY, and MAXDATA operands. The PCCU values for these operands are used only when the channel-link station is not defined in a channel-attachment major node.

A channel-attached NCP requires one HOST definition statement for each host that activates the NCP. The HOST definition statement in the NCP major node has a MAXBFRU operand for defining the number of buffers that the host can use for receiving data from the NCP. If you do not define the channel in a channel-attachment major node, the MAXBFRU value on the HOST definition statement is used because the channel is implicitly defined and activated automatically when the NCP is activated.

However, when you define the channel in a channel-attachment major node, the MAXBFRU operand on the LINE definition statement for that major node defines the number of buffers the host can use for receiving data. Figure 56 on page 222 shows an example of this. When HOST1 activates its channel-attachment major node, VTAM uses the value of the MAXBFRU operand on the LINE definition statement. Later, if NCP1 is activated by HOST1, the MAXBFRU value on the HOST definition statement is ignored.

When you define the channel in a channel-attachment major node, VTAM also ignores the MAXBFRU value on the HOST definition statement when the link station for the channel is activated automatically. This is because the value is determined by the RNAME operand when the NCP is activated.

Note: The MAXBFRU operands on the LINE definition statement of the channel-attachment major node should contain the same values as the MAXBFRU operand specified on the HOST definition statement of the NCP major node.

For more information on defining a channel-attachment connection between an NCP and a VTAM data host, refer to "Channel-attached NCP to a VTAM data host" on page 120.

Link-attached NCPs in a CMC configuration

In a multiple-domain network, you can define a switched major node in one domain and treat its logical units as cross-domain resources in other domains. You can also define a switched major node in each domain that owns an NCP and has a dial port for a terminal. If defined in more than one domain, NCPs can be dialed-in to or be dialed-out from each of the host processors in which the major node is defined.

No HOST definition statement should be defined for a link-attached NCP.

Controlling resource ownership in a CMC configuration

Using the OWNER operand on the PCCU definition statement enables you to easily identify and coordinate ownership of NCPs. You can also specify ownership of resources at the physical unit (PU) level for peripheral nodes attached to an SDLC link. You can share ownership of resources on a line and ownership of the lines. In addition, the use of the OWNER operand has implications in backup and recovery plans involving shared NCPs.

The acquisition and release of resources of a shared NCP allow more selectivity in specifying which resources are affected. The VARY ACQ and VARY REL operator commands, with the OWNER support command, enable you to selectively acquire and release resources. The VTAM operator can specify `VARY ACQ,OWNER=sscp name,ID=ncp name` to specify that the VARY ACQ command is to apply only to those resources defined as belonging to the specified owner. An equivalent VARY REL operator command allows the VTAM operator to selectively release ownership of a resource.

The TYPE=GIVEBACK operand on the VARY REL operator command permits the VTAM operator to release ownership of the resource without disrupting an LU-LU session. However, because this function operates only at the line level, all physical units attached to that line also have the ownership terminated without disrupting LU-LU sessions. If a host wants to relinquish ownership of the logical unit, it releases or deactivates the logical unit and its associated physical unit.

Note: You can release ownership of an SNA resource without disrupting an LU-LU session.

Every resource attached to the NCP is owned by a VTAM host. The values you code for each NCP major node determine which host owns which resources. The value of the OWNER operand on the PCCU definition statement is matched with the OWNER operands on the GROUP, LINE, and PU definition statements in the NCP.

More than one host can share ownership of a line or a PU type 4 link station. You specify this in the NCP by coding the appropriate names as values of the OWNER operand on the GROUP or LINE definition statements. In this situation, the first host to activate the physical unit becomes its owner. Other hosts cannot activate this physical unit unless ownership is relinquished. In addition, ownership of PUs can be specified by the OWNER operand on the PU definition statement.

Use the BACKUP operand on the PCCU definition statement to allow a backup host to take over NCP resources when there is a failure in the host that owns them. Code BACKUP=YES to identify all resources that this host can take over in case of a failure. When a failure occurs, an operator can issue a VARY ACQ operator command to acquire these resources without disrupting sessions.

If more than one VTAM host will activate the NCP, you can code a PCCU definition statement for each host. Each PCCU definition statement describes the functions performed by the host that it is associated with. For example, if you want to assign a second host to back up a communication management configuration (CMC) host, code a PCCU definition statement for the backup host. The definition statement is not used until a failure occurs, at which time the backup host activates the NCP.

Note: For an SSCP to successfully take over for another SSCP that has issued a DR MOVE (dynamic reconfiguration move), the takeover SSCP must also support DR MOVE. In other words, the takeover SSCP must support the dynamic reconfiguration MOVE facility. More importantly, the takeover SSCP must be configured.

Problem determination in a CMC configuration

If you configure your network to use a data host, you should be aware of the following problem determination restrictions associated with this type of attachment to the NCP:

- The VTAM I/O trace must be used to verify the transfer of data to and from VTAM over the channel-link station. The VTAM internal trace can be used if necessary. The line trace function cannot be used even though the NCP is defined on a LINE definition statement.
- The link level 2 test to the NCP is not supported.
- Any function requiring an SSCP-PU session from the data host cannot be supported (such as NCP storage display, line trace, link level 2 test for resources attached to the NCP or buffer, or TG or SIT traces for any channel-attached cross-domain NCP).

Chapter 9. Defining resources dynamically

You can reduce the number of definitions that must be coded by using the dynamic definition capabilities that VTAM provides. VTAM enables the dynamic definition of switched resources and channel-attached devices. It also enables the dynamic reconfiguration of existing resources and the dynamic change of operands on existing definitions.

The following lists shows where to find information about dynamically defining resources.

- "Defining switched resources dynamically"
- "Dynamic configuration of channel-attached devices" on page 230
- "Dynamic reconfiguration and change of operands" on page 235

Defining switched resources dynamically

You can dynamically define switched peripheral nodes (including LAN-attached nodes) using:

- Dynamic PU definition (DYNPU operand)
- The dynamic switched definition facility

The following sections describe this process.

Dynamic PU definition (DYNPU operand)

The DYNPU operand enables PUs for type 2.1 peripheral nodes to be created dynamically on the VTAM host accepting an incoming call. Code DYNPU=YES on the GROUP definition statement in an NCP, external communication adapter (XCA), or channel-attachment (CA) major node.

A switched PU is dynamically created when a calling PU cannot be identified by VTAM during a switched dial-in operation. VTAM uses the model PU definition in the model major node to define the characteristics of the PU.

Note: The DYNPU operand is *ignored* if a configuration services XID exit routine is active when a dynamic PU is needed. The exit routine determines whether a PU will be dynamically defined.

Dynamic switched definitions

The dynamic switched definition facility allows you to dynamically create definitions for types 1, 2, or 2.1 switched dial-in devices, and for leased devices that appear to VTAM to be switched (for example, devices connected by NTRI). You cannot dynamically define type 4 or type 5 devices. To create the dynamic definitions, this facility uses information provided by an installation-supplied exit routine and coded in a model major node.

The dynamic switched definition facility requires model definition statements and a configuration services XID exit routine. VTAM uses the model definitions to create the dynamic switched major node, from which definitions are taken to build dynamic switched devices.

© Copyright IBM Corp. 1984, 2011

A configuration services XID exit routine is invoked when an unknown device dials in to VTAM. The exit uses IDBLK, IDNUM and other parameters from the XID vector to gather resource definition information. The exit returns to VTAM the model statements, an optional dynamic switched major node name, and the names for the new PU and LUs.

If necessary, VTAM builds a dynamic switched major node. The new devices are added to the major node. If the exit does not specify a major node name, VTAM uses ISTDSWMN. Figure 57 illustrates the process of creating resources in a dynamically switched node.

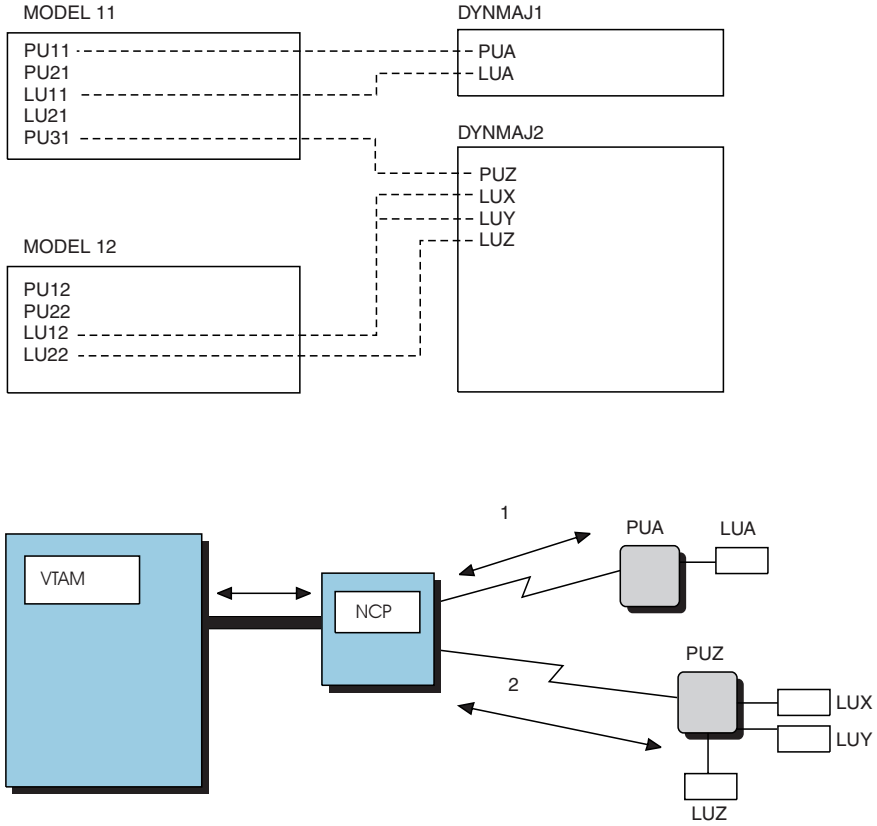


Figure 57. Creating resources in dynamic switched major node

In Figure 57, when device PUA dials in to VTAM, PU and LU definitions from MODEL 11 are used by a configuration services exit routine to create resources in an existing dynamic switched major node, DYNMAJ1. When device PUZ dials in to VTAM, PU and LU definitions in MODEL 11 and MODEL 12 are used by the exit routine to create resources in dynamic switched major node, DYNMAJ2. Because dynamic switched major node DYNMAJ2 did not already exist, it, along with its resources, was created by this process.

Configuration services exit routine

A sample configuration services XID exit routine (ISTEXCCS ASSEMBLE) is provided in the sample library (SYS1.SAMPLIB). You can write your own exit routine or use the sample. To code the exit routine to build dynamic resources, follow these steps:

1. Code the exit to supply names for:
   - The switched PU and LU resources
   - The model definition to be used for the PU or LU

- The dynamic switched major node where the dynamic PUs and LUs are to be created

  Note: If the exit does not return a major node name, the default major node ISTDSWMN is used. If the exit returns an unknown major node name, a new dynamic switched major node will be created.

2. Install the exit routine on VTAM to support dynamic switched devices.
3. Activate the routine using the MODIFY EXIT command; otherwise, no dynamic definitions are built, and the device request for connection is rejected.

For information on coding this exit routine, refer to *z/OS Communications Server: SNA Customization*.

Model major nodes and model statements

For additional dynamic PU definitions that are possible using the configuration services XID exit, refer to Defining a PU and an LU for the configuration services XID exit routine in *z/OS Communications Server: SNA Resource Definition Samples*.

You can code one or more model major nodes, or copy and modify the model major node provided on the sample library (ISTMODEL on SYS1.SAMPLIB). Model major nodes contain the parameters that VTAM uses to create dynamic switched resources.

Define a model major node with a VBUILD definition statement (TYPE=MODEL). The PU and LU definition statements in the model major node define the characteristics of switched peripheral nodes.

Model PU and LU statements are defined for the minor nodes. When the configuration services XID exit routine returns the names of the model resource and the previously undefined resource, VTAM uses these model statements to create the dynamic PUs and LUs. The exit can also identify the name of a dynamic switched major node. There is no relationship between these PU and LU definition statements in a model major node. The exit routine can provide the name of any model PU or LU definition that matches the device characteristics. Therefore, there is no sifting done from the PU definition to the LU definition. The GROUP definition is available only for sifting down to the LU definitions.

Sample model major node: In the following sample you do not need to code LU statements following a PU statement.

```
MODEL    VBUILD TYPE=MODEL             MODEL MAJOR NODE
SPUMOD02 PU     ADDR=02,               PU STATION ADDRESS
                MAXDATA=256,           MAX BYTES PU RECEIVES IN ONE PIU
                PUTYPE=2,              PU TYPE
                MAXOUT=1,              MAX PIUS WITH NO REPLY
                IRETRY=NO,             NCP RETRY
                DISCNT=YES,            DISCONNECT AT SESSION END
                ANS=CONTINUE           CONTINUE DURING NCP ANS
SPUMOD01 PU     ADDR=01,               PU STATION ADDRESS
                MAXDATA=256,
                PUTYPE=2,
                MAXOUT=1,
                IRETRY=NO,
                DISCNT=YES
GROUP1   GROUP PACING=(1,1),    PACING LU AND BOUNDARY NODE
                VPACING=2       PACING VTAM AND BOUNDARY NODE
SPUMOD   PU     ADDR=03,
                MAXDATA=256,
                PUTYPE=2,
                MAXOUT=1,
```

```
              IRETRY=NO,
              DISCNT=YES
SLUMOD01 LU   LOCADDR=1,        LU LOCAL ADDRESS
              MODETAB=MODETAB2  LOGON MODE TABLE NAME
SLUMOD03 LU   LOCADDR=3,        LU LOCAL ADDRESS
              MODETAB=MODETAB2
SLUMOD02 LU   LOCADDR=2         LU LOCAL ADDRESS
```

Note: The PACING and VPACING values coded on the GROUP statement do not sift down to SPUMOD. SPUMOD gets the default values for PACING and VPACING. PACING and VPACING sift down to the LUs from the GROUP, but they would not have sifted down from the PU if they had been specified there.

A PU on a switched line is identified by the device CPNAME or IDBLK and IDNUM. As soon as a dial-in line has been activated and placed in answer mode (during VTAM startup or by the VTAM operator), the physical unit can dial the line number and establish communication with VTAM.

When the dial-in connection is broken, the devices are deactivated and the PU and LU definitions in the dynamic switched major node are deleted. When the last PU within the dynamic switched major node is deleted, the switched major node is also deleted.

Coding DISCNT=YES causes the connection to be broken after all sessions using the line have ended. If you code DISCNT=YES on your model PU statements, consider the cost of the line and the number of session starts and ends this particular device will have.

Restriction: Dynamic definition of switched connections is for dial-in use only. VTAM has no knowledge of the path definitions. Therefore, should the line drop, even if the device is active, VTAM cannot use the definition for dial-out operations. The sole exception to this rule is for a model PU defined for Enterprise Extender with DYNTYPE=EE and DWINOP=YES coded. In this case, if the EE connection is dropped, VTAM will attempt to dial back out to the partner using the remote IP address and remote SAP sent inbound when the connection was established. The configuration restart files are not updated with dynamic definitions.

In environments with an owning and backup host, the backup host can also implement a configuration services XID exit, and dynamically create resource definitions during takeover of NCPs that have dial-in resources connected to them. The backup host exit is provided the same information during takeover that is provided to the exit during a dial-in operation.

Dynamic configuration of channel-attached devices

Dynamic configuration of channel-attached devices (dynamic I/O) enables you to modify your I/O configuration without disrupting your system. You can add, delete, or modify I/O components, such as devices and paths, without having to re-IML or re-IPL.

Use the following items to dynamically configure your channel-attached devices and dynamically create VTAMLST members for new devices:

- Hardware configuration definition (HCD)
- NetView Release 3 and subsequent releases, or a product with a similar interface and services

Note: NetView and TSO/E are necessary for dynamically creating VTAMLST members using the VTAM-supplied sample command lists. To modify the I/O configuration only, you need only MVS and HCD.

Figure 58 shows how a channel-attached device is added using dynamic I/O configuration support.

1. Physically attach a device.
2. Add the device to your configuration using hardware configuration definition (HCD), and activate the new configuration using the MVS ACTIVATE command.
3. MVS issues message IOS502I, which causes NetView to start the IOS502I command list.
4. The IOS502I command list calls the ISTDEFIN command list for each device added to the system.
5. The ISTDEFIN command list obtains a description of the new device using the ISTDINFO function module, which obtains the new device information and returns it to ISTDEFIN.
6. The ISTDEFIN command list builds a VTAMLST member for the device. Any information the device cannot provide can be found with IBM- or user-supplied defaults.
7. The ISTDEFIN command list issues a VARY ONLINE command for the device if it is not already online.
8. The ISTDEFIN command list issues a VARY ACT command to activate the device.

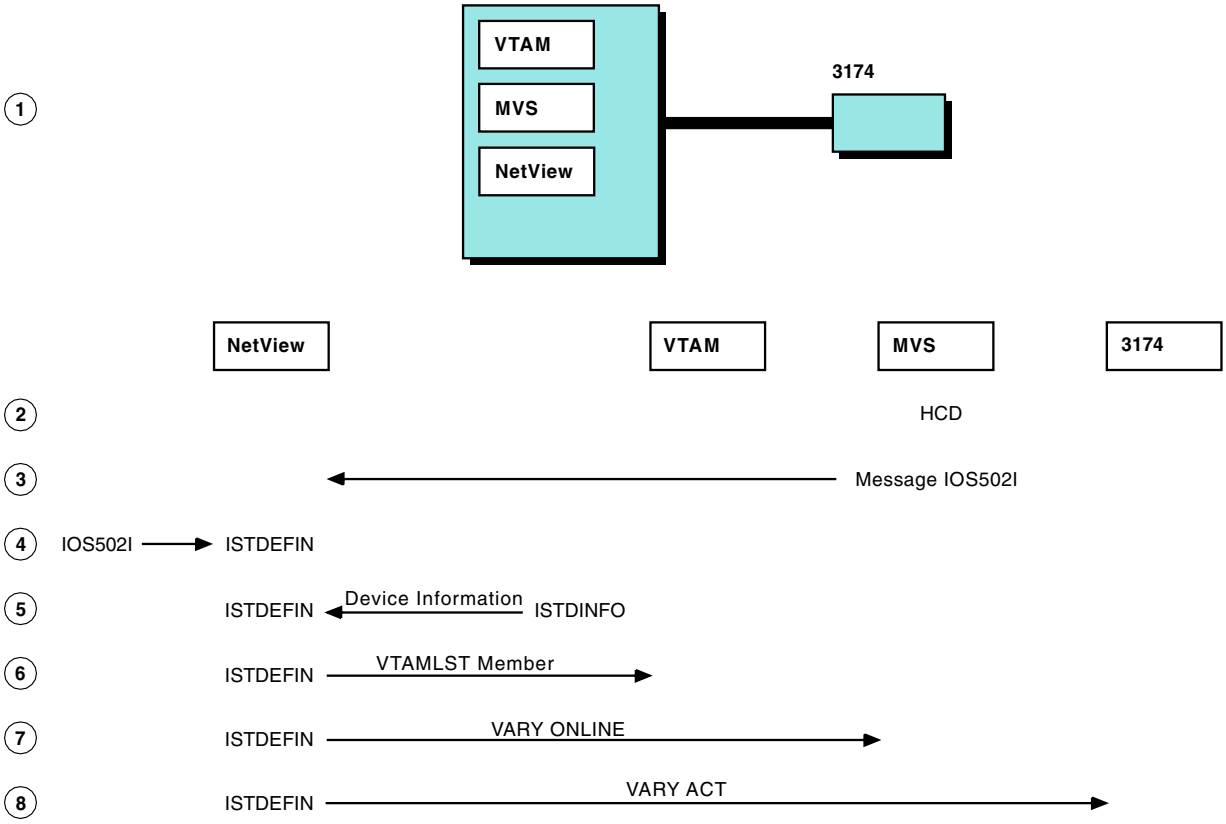


Figure 58. Dynamic configuration of channel-attached device

Installation and preparation

To dynamically configure your channel-attached devices, use the NetView program or another product providing a similar interface and services. If you are using NetView, do the following to prepare it for dynamic I/O configuration:

1. Modify the message automation table so that when MVS issues message IOS502I, NetView can start the IOS502I command list. The following sample addition to the message automation table logs messages issued by the ISTDEFIN command list, starts the IOS502I command list, routes the message to one operator and logs it:

```
IF MSGID="ISTDEFIN" THEN
   DISPLAY (N)
   NETLOG(Y);
IF MSGID="IOS502I" THEN
   EXEC (CMD("IOS502I") ROUTE (ONE OPER1))
   DISPLAY (N)
   NETLOG(Y);
```

2. Modify the NetView start procedure to point to the sample library (VTAM.SAMPLIB) where the IOS502I and ISTDEFIN command lists reside or move the command lists from the sample library to the library where your other command lists are stored.
3. Activate the message automation table that you changed using the NetView command AUTOMSG MEMBER=tablename.

Refer to the *z/OS Communications Server: New Function Summary* for information on data sets needed for dynamic I/O configuration.

Defining your configuration

After preparing NetView or a similar package, use the hardware configuration definition (HCD) to interactively define your I/O configuration changes. (Use of the HCD is described more fully in the MVS library.)

To attach a device to the system:

1. Physically attach the device.
2. Define the device to MVS using HCD.
3. Activate the new configuration using the MVS command ACTIVATE IODF.

MVS then issues message IOS502I, which causes NetView to start the IOS502I command list. The IOS502I command list calls the ISTDEFIN command list for each device added to the system.

Note: Message IOS502I does not display on the MVS console nor on the NetView operator console unless requested using the NetView ASSIGN command.

Building resource definitions

The ISTDEFIN command list obtains descriptions about the new devices using the ISTDINFO function module. The device-specific information that is returned by the ISTDINFO function module varies according to the device.

The ISTDEFIN command list builds channel attachment or local SNA major nodes (PU type 2, 2.1, and 4) for the devices.

The ISTDEFIN command uses IBM-supplied defaults to build the definition statements. Table 10 on page 233 shows the devices and the definitions that the ISTDEFIN command builds.

Table 10. Definitions for dynamically configured devices

Device	Definition
3705	Channel-attachment major node
3720, 3725	Channel-attachment major node
3745 (type 6 adapter)	Channel-attachment major node
3745 (type 7 adapter with NCP loaded)	Channel-attachment or Local SNA major node
3274	Local SNA major node
3174	Local SNA major node

For a 3745 device that is self-describing (3745 type 7 adapter), the ISTDEFIN command builds either a channel-attachment major node or a local SNA major node to define the device. The definition built is determined by the NCP generation already residing in the 3745. For example, an NCP level that does not support a type 7 adapter causes the ISTDEFIN command to build a channel-attachment major node. Otherwise, the major node built depends on whether the channel is defined as a type 2 (Local SNA major node) or type 4 (channel-attachment major node) in the NCP.

Information needed for defining the major nodes that is not provided by the ISTDINFO function module is defined in templates in the ISTDEFIN command list.

Notes:

1. If the device is self-describing, an NCP being defined must already be loaded.
2. The VTAM using dynamic I/O configuration can only be a data host to an NCP. VTAM does not alter existing PATH definition statements to create a route to the NCP. Do this before you try to dynamically add any devices. Otherwise, the activation of the channel attachment will fail.
3. A local SNA major node can be dynamically created to represent a type 2.1 channel-attached to an NCP. However, no cross-domain resources (CDRSC) definitions for the independent LUs defined under the type 2.1 channel are dynamically created. Also, default adjacent link station lists belonging to the independent LUs are not updated with the dynamically-created PU name.

Using the default naming convention

The IBM-supplied command lists generate names for resource definitions using either an IBM-supplied or user-written naming convention. Names are generated in the following manner:

- The first character is `N`, identifying the name as being generated by VTAM for dynamic I/O configuration.
- The second character is the subarea number expressed as a single character (0–9 or A–Z). For example, a communication controller in subarea 6 has a second character of 6. A communication controller in subarea 35 has a second character of Z. A communication controller in subarea 36 has a second character of 0, because subarea numbering begins with 1, not 0. The maximum number of subareas that can be defined using the IBM-supplied naming convention is 36.
- The next four characters are the 4-digit device address, represented in hexadecimal.
- In the definition of a single device, a number of resources must be named. The last characters depend on the resource being named:

L LINE

G GROUP

D Data-set name

xx 2-digit hexadecimal local address for the logical unit attached to the cluster controller (3x74)

For example, the name generated for the definition of a device at X'123' in subarea 6 is named `N60123D`. The LINE name for the device is `N60123L`.

Customizing the command lists

The IOS502I and ISTDEFIN command lists can run without any modification. However, you can modify the command lists to match the needs of your environment. The command lists are coded in REXX, and the subroutines and functions are documented in the code. For a description of the methods for calling the command lists and the information returned, see Appendix F, "Command lists: Dynamic configuration of channel-attached devices," on page 763.

Modifying the resource definition defaults

Information needed for defining the major nodes that is not provided by the ISTDINFO function module is defined in templates in the ISTDEFIN command list. The templates contain information on logon modes, interpret tables, resource names, the number of logical units (SNA cluster controllers only), and other information.

Some information, such as the number of buffers VTAM should allocate for receiving data from the NCP (provided on the MAXBFRU operand of the HOST definition statement), is not always available. You can modify the defaults in the templates to override the IBM-supplied defaults. Edit the ISTDEFIN command list, which shows you the templates with which you can modify the values in the REXX statements.

Note: When logical units are being defined, if the physical unit presents a *logical_terminals* string, only the first LU statement in the template is used.

Modifying the naming convention

You can change the naming convention by modifying the ISTDEFIN command list. A set of directions is provided in the ISTDEFIN command list for supplying a *seed* value to control the values for the first two characters of the device name. For example, if you want to have your device names begin with `MY`, change the following in the ISTDEFIN command list:

```
seed='MY'                /* user seed value */
```

Thus, the name generated for the definition of a device at X'123' is named `MY0123D`.

If you want to use the IBM defaults, do not change the *seed* value.

ISTDEFIN replaces the major node if the same name already exists in USER1.AUTO.VTAMLST, so the definition contains the latest device information.

Supplying your own naming convention

If you want to use your own naming convention, change the logic of the ISTDEFIN command list. The functions and subroutines that obtain the system parameters used to build the definitions (such as device number and subarea

number) are provided and documented in the command list. You can use these functions to modify the ISTDEFIN command list to use your own naming convention.

Writing your own command lists

The ISTDEFIN and IOS502I command lists are designed to work together. However, you can call these command lists with your own functions and routines.

For example, you can write a command list that is invoked when NetView or VTAM startup completes. This command list can determine all VTAM-supported devices not currently used by VTAM. This command list can check all possible devices and then call the ISTDEFIN command list to build the definitions and activate the devices.

Dynamic reconfiguration and change of operands

Dynamic reconfiguration is the process of adding, deleting, or moving resources within your network configuration without deactivating the affected major node. For example, you can change resources on an NCP without deactivating the NCP. Dynamic reconfiguration does not replace NCP generations, but supplements them to enable quick and efficient control of changes in the domain configuration. You can also use dynamic reconfiguration for local peripheral nodes, Enterprise Extender XCA major nodes, application major nodes, and CDRSC major nodes.

Dynamic change of operands enables you to modify certain operand values on resource definition statements without deactivating the major node.

Tip: LUs and PUs can be defined dynamically, in which case no system definition is required.

Dynamic reconfiguration and dynamic change of operands are supported for the following:

- PUs and LUs on nonswitched SDLC lines or ESCON® channels in an NCP major node
- PUs, LUs, and PATHs can be added and deleted in a switched major node
- LUs in a local SNA major node
- Type 1 PUs under NCP frame-relay lines, and their associated FRSESET definition statements
- TRLEs in a TRL major node
- GROUP definitions, LINE definitions, and PORT connection network definitions in an Enterprise Extender XCA major node
- CDRSCs in a CDRSC major node
- APPL definitions in an APPL major node

Dynamic change of operands is also supported for PUs and LUs in switched major nodes.

Dynamic reconfiguration of NCP resources requires NCP Version 4 Release 3 or Version 5 Release 3, or later (unless noted) and allows you to:

- Add and delete type 1, type 2, and type 2.1 physical units and their associated logical units on nonswitched SDLC lines
- Add and delete logical units under physical units on nonswitched SDLC lines
- Add and delete type 2.1 physical units on an NCP ESCON channel

- Add and delete independent logical units on an NCP ESCON channel

  Note: Independent LUs can be dynamically defined, in which case no system definition of the independent LU is required. Independent LUs can also be defined using CDRSC definition statements.
- Add and delete NCP frame-relay type 1 PUs and their associated FRSESET definition statements
- Move a physical unit and its associated logical units from its current line to another line within the same NCP
- Code and change the values of the following NCP operands in the NCP and switched major nodes:
  - COMRATE
  - DLCI
  - DYNWIND
  - HPRQLIM
  - INTFACE
  - IPQLIM
  - LADDR
  - T1TIMER
  - T2TIMER

  The values you specify for these NCP operands are passed to NCP during dynamic reconfiguration of the NCP or during switched PU connection.

  Note that VTAM registers only the values of these operands and does minimal validation of them before passing them to NCP. For more information on these operands, refer to the *NCP, SSP, and EP Resource Definition Reference*.
- Move a logical unit to another physical unit on nonswitched SDLC lines.
- Change the ADDR for a physical unit on nonswitched SDLC lines.
- Change the LOCADDR for a logical unit on nonswitched SDLC lines.
- Add TRLE entries to a TRL major node or replace existing TRLE entries that are not currently in use (that is, a TRLE entry that is currently in use is not deleted; new entries are still added). A TRLE entry that exists in another TRL major node is not replaced whether or not it is in use.

There are several ways to perform dynamic reconfiguration. See "Dynamic reconfiguration and dynamic change requirements" on page 237 to determine which method is valid for a given resource:

- Add, delete, or move resources in the VTAMLST definition file and issue a VARY ACT,UPDATE=ALL command. Changes made in this way are permanent changes, and do not require the major node to be deactivated.

  See "Using the VARY ACT,UPDATE technique" on page 238 for additional information.
- Code a DR file to add, delete, and move resources and activate it with the VARY DRDS command. Changes made in this way are temporary changes. If you want the changes to be permanent, a new NCP must be generated and reloaded to reflect the changes as soon as time permits.

  See "Using the VARY DRDS technique" on page 241 for additional information.
- Use the MODIFY DR command to move or delete resources. Changes made in this way are temporary changes. If you want the changes to be permanent, a new NCP must be generated and reloaded to reflect the changes as soon as time permits.

  See "Using the MODIFY DR technique" on page 243 for additional information.

- Add resources to the NCP VTAMLST definition file, and then reactivate the NCP (implicit dynamic reconfiguration). This procedure is disruptive because inactivation of an NCP causes all existing sessions to be terminated. As soon as possible, you must regenerate and reload the NCP to reflect the changes.

  See "Implicit dynamic reconfiguration" on page 246 for additional information.

For some examples of dynamic reconfiguration, see "Dynamic reconfiguration examples" on page 246.

Dynamic reconfiguration and dynamic change requirements

Table 11 shows which techniques of dynamic reconfiguration and dynamic change of operands are valid for a particular operation on a resource in a particular major node.

Table 11. Dynamic reconfiguration operations for valid major nodes

Operation/resource	NCP	Local SNA	Switched	TRL	CDRSC	EE	APPL
Add LU	2, 4	2	1	-	-	-	-
Delete LU	3	3	1	-	-	-	-
Move LU	1	1	-	-	-	-	-
Change LU operands	1	1	1	-	-	-	-
Add PU	2, 4	-	1	-	-	-	-
Delete PU	3	-	1	-	-	-	-
Move PU	3	-	-	-	-	-	-
Change PU operands	1	-	1	-	-	-	-
Add PATH	-	-	1	-	-	-	-
Delete PATH	-	-	1	-	-	-	-
Change PATH operands	-	-	1	-	-	-	-
Add TRLE	-	-	-	1	-	-	-
Delete TRLE	-	-	-	1	-	-	-
Change TRLE operands	-	-	-	1	-	-	-
Add CDRSC	-	-	-	-	1	-	-
Delete CDRSC	-	-	-	-	1	-	-
Change CDRSC operands	-	-	-	-	1	-	-
Add GROUP	-	-	-	-	-	1	-
Delete GROUP	-	-	-	-	-	1	-
Change GROUP operands	-	-	-	-	-	1	-
Add LINE	-	-	-	-	-	1	-
Delete LINE	-	-	-	-	-	1	-
Change LINE operands	-	-	-	-	-	1	-
Change PORT connection network values	-	-	-	-	-	1	-
Add APPL	-	-	-	-	-	-	1
Delete APPL	-	-	-	-	-	-	1

Table 11. Dynamic reconfiguration operations for valid major nodes (continued)

Operation/resource	NCP	Local SNA	Switched	TRL	CDRSC	EE	APPL
Change APPL operands	-	-	-	-	-	-	1

Legend:

1 V ACT,UPDATE only
2 Either V DRDS or V ACT,UPDATE
3 Either MODIFY DR, V DRDS, or V ACT,UPDATE
4 Implicit dynamic reconfiguration

Dynamic reconfiguration is not supported on a line that has type 4 or type 5 physical units on it (with the exception of type 2.1 PUs and independent LUs on NCP ESCON channels).

VTAM does not redrive changed PU ADDRs or LU LOCADDRs during the initial activation of an NCP (including activation of an NCP during a warm start), even when VARY ACT,UPDATE=ALL is used. However, if the PU or LU was also moved, VTAM will redrive.

VTAM support for dynamic reconfiguration requires that the PUDRPOOL and LUDRPOOL definition statements be coded at NCP generation. VTAM requests that the NCP assign a PU or LU address to a resource specified by the host. The addresses come from the address pools generated by the LUDRPOOL or PUDRPOOL definition statements. Be sure that LUDRPOOL and PUDRPOOL are large enough to accommodate all of the resources that you plan to add to your network using dynamic reconfiguration.

Similarly, be sure that the MAXPU keyword (on the LINE definition statement) and the MAXLU keyword (on the PU definition statement) are large enough to accommodate all of the resources that you plan to add using dynamic reconfiguration. MAXPU limits the number of PUs on a line, and MAXLU limits the number of LUs under a PU.

In addition, when a PU is added to a line, NCP adds the PU at the end of its service order table. If you want to have the PU added multiple times in the service order table, you must regenerate and reload the NCP.

Refer to *NCP, SSP, and EP Resource Definition Reference* and the *NCP, SSP, and EP Resource Definition Guide* for more information about PUDRPOOL, LUDRPOOL, MAXPU, MAXLU, and the service order table.

Using the VARY ACT,UPDATE technique

You can use the VARY ACT,UPDATE technique to dynamically change certain operand values on definition statements without deactivating the major node. Only the resources that you are changing must be inactive.

To change the values, edit the existing definitions in the VTAMLST file and then issue a `VARY ACT,UPDATE=ALL` command.

Some VTAM-only operands can be changed, depending on the state of the resource. All other operands can be changed only by deleting the resource and adding the resource. For more information on which operands support dynamic change, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

To use the VARY ACT,UPDATE technique for dynamic reconfiguration, change the VTAMLST source file for a major node and issue a VARY ACT,UPDATE=IMPLICIT|ADD|ALL command. The command causes VTAM to update the actual configuration to match the new definition in the VTAMLST source file. The major node can be active for ADD or ALL. Only the individual resources being added, deleted, or moved must be inactive.

Restrictions for changing PU and LU resources: When using the VARY ACT,UPDATE=ALL command to dynamically change PU and LU resources, the following additional restrictions apply:

- The PUTYPE value cannot be changed. Also, you cannot add SDDLU support to a PU.
- You can convert an APPN PU to a LEN PU, or a LEN PU to an APPN PU, by changing the value of the CONNTYPE operand. However, you cannot convert an APPN PU to a LEN PU if one of the following is true:
  - You coded the same NETID/CPNAME value on multiple PUs.
  - You coded an ADJCP definition statement for a PU to represent a partner CP and you coded the same NETID/CPNAME value on that PU.

Restrictions for changing Enterprise Extender resources: When using the VARY ACT,UPDATE=ALL command to dynamically change Enterprise Extender resources, the following additional restrictions apply:

- Changes to Enterprise Extender groups require the inactivation of the group and all subordinate resources within the group. This is done by using the VARY INACT,ID=*groupname* command.
- Changes to port connection network values require the inactivation of the group identified by the VNGROUP operand.

Restrictions for changing model CDRSCs: When using the VARY ACT,UPDATE=ALL command to dynamically change model CDRSCs in a CDRSC major node, the following additional restrictions apply:

- Existing clone CDRSCs are unaffected by changes to the model CDRSC.
- A model CDRSC cannot be deleted while it has clone CDRSCs.
- A conventional CDRSC cannot be added when there is a clone CDRSC of the same name. First, use the VARY INACT,DELETE=YES command to remove the clone CDRSC and then add the conventional CDRSC using the VARY ACT,UPDATE=ALL command.
- V ACT,UPDATE=ALL cannot be used to delete clone CDRSCs.

Restrictions for changing QDIO TRLE definitions: Stop all TCPIP devices that are using the TRLE and 2 minutes must pass before the QDIO TRLE can be updated.

Restrictions for changing applications: When using the VARY ACT,UPDATE=ALL command to dynamically change applications in an application major node, the following additional restrictions apply:

- An application cannot be added when there is a clone application of the same name. First, initiate close processing for the clone application. This causes it to be deleted. Then add the application using VARY ACT,UPDATE=ALL.
- An application cannot be added to the major node unless it can be added to the host name space as a resource or a shadow resource.
- The VARY ACT,UPDATE=ALL command cannot be used to delete clone applications.

- A model application cannot be deleted if any clone application created using the model exists.

VARY ACT,UPDATE=IMPLICIT (the default) has the same effect as implicit dynamic reconfiguration. For more information on implicit dynamic reconfiguration, see "Implicit dynamic reconfiguration" on page 246. For more information on the VARY ACT command, refer to *z/OS Communications Server: SNA Operation*.

With VARY ACT,UPDATE=ADD|ALL, dynamic reconfiguration changes stay in effect regardless of VTAM or NCP recycles (halt and restart) and upon takeover of an NCP. You do not need to redrive dynamic reconfiguration changes with VARY DRDS or MODIFY DR commands. It is recommended that you do *not* make changes to VTAMLST files during a VTAM or NCP recycle. Instead, make the changes dynamically while VTAM and NCP are running. To prepare for takeover of an NCP, distribute the new VTAMLST file to the backup host and issue the same VARY ACT,UPDATE=ADD|ALL command that you issued at the primary host.

With VARY ACT,UPDATE=ADD|ALL, you can delay the NCP generation for a longer time than you can with other methods of dynamic reconfiguration. The changes you have made in VTAMLST definition files cause more network flows between VTAM and NCP on a recycle, but you need not regenerate the NCP until performance during recycles becomes unacceptably slow. Regenerating and reloading the NCP eliminates the extra flows.

If you use the NCP/EP definition facility (NDF) to generate some definitions, make your changes to the NDF source input, run that source through NDF, and place the resulting NEWDEFN file into a VTAMLST member. Save the load module for future use. If you update the NEWDEFN file directly, your changes are lost the next time NDF is run.

Sifting of operands does take place on subordinate resources that have been added or moved by the VARY ACT,UPDATE method of dynamic reconfiguration.

Network management programs that rely on VTAMLST definitions for the network configuration have current information available to them about the configuration.

Note: If you use the VARY ACT, the UPDATE=ALL command might override the following commands:

- The VARY LOGON command to establish an automatic logon specification that is not coded on the LOGAPPL operand of a definition statement
- The VARY NOLOGON command to delete an automatic logon specification that is coded on the LOGAPPL operand of a definition statement
- The MODIFY DEFAULTS, DLOGMOD command to change the value of the DLOGMOD operand of a definition statement
- The MODIFY TABLE command to change VTAM table associations

For example, the LOGAPPL operand in the VTAMLST definition overrides the VARY LOGON and VARY NOLOGON commands. Even if no LOGAPPL is coded in the VTAMLST definition, a null value for LOGAPPL overrides the VARY LOGON value. To avoid having the VARY LOGON command overridden, code the LOGAPPL definition statement with the desired value. To avoid having the VARY NOLOGON command overridden, delete the LOGAPPL operand from the definition statement.

Coding dynamic reconfiguration changes

To add resources to a network, code the new definition statements in the desired location in the VTAMLST file for the major node. To delete resources, delete the appropriate definition statements from the VTAMLST file. To move resources, move the appropriate definition statements from one location to another location in the VTAMLST file.

You can move a physical unit (and its associated logical units) from a line under one major node to a line under a different major node under the same VTAM. To do this, delete the resources from one major node and add them to the other major node. You delete the resources with any method of dynamic reconfiguration. Make the additions in the VTAMLST file.

Some move or add operations can depend upon a delete operation completing first. For example, you could exchange resources between two major nodes at the same time (that is, move resources from the first major node to the second major node and also move resources from the second major node to the first). Or, you might delete a PU from a line and add a PU to the line with the same ADDR value as the deleted PU.

You can resolve this problem in either of these ways:

- Make deletions with any method of dynamic reconfiguration, and then make the additions by modifying VTAMLST and issuing the VARY ACT,UPDATE=ADD command.
- Process the source definitions twice (issue the same VARY ACT command again). The deletions will take place on the first run of the command, and the additions that failed on the first run will take place on the second run.

Errors in the VTAMLST definition file

If you attempt dynamic reconfiguration and the VTAMLST file contains errors, you might receive unexpected results. Errors in BUILD or PCCU statements or errors in required operands can cause the VARY ACT command to fail, in which case no dynamic reconfiguration takes place. Errors in definition statements can cause unplanned dynamic reconfiguration changes or unplanned dynamic changes in operand values. Depending on the severity of the error in the definition file, VTAM may skip over resources that have incorrect definitions and process the rest of the definition file. VTAM deletes resources that are skipped unless they are active.

Note: You should run the NCP/EP definition facility (NDF) or issue the VARY ACT,SCOPE=SYNTAX command to check for definition errors *before* you attempt dynamic reconfiguration changes.

Using the VARY DRDS technique

The VARY DRDS command activates a dynamic reconfiguration (DR) file. To use the VARY DRDS command:

1. Define the appropriate dynamic reconfiguration files.
2. Issue the VARY DRDS command to implement the changes.

You can use a dynamic reconfiguration file to:

- Move a physical unit from one nonswitched line to another within the same NCP
- Add physical units to nonswitched lines, or add logical units to physical units under nonswitched lines

- Change the link station address of physical units under nonswitched lines
- Delete physical units from nonswitched lines and delete logical units from physical units under nonswitched lines
- Delete NCP frame-relay PUs

The NCP to be reconfigured must be active, and the resource to be deleted or moved must be inactive. Also, the NCP cannot recognize, move, or add requests unless the physical unit was active at some time before inactivating it to issue a VARY DRDS command on it.

Dynamic reconfiguration with a DR file should be used only as a temporary method; a new NCP should be generated to reflect the changes as soon as time permits. The NCP should be reloaded to reflect the permanent changes.

If you are coding a dynamic reconfiguration file:

1. The first definition statement in the file must be VBUILD TYPE=DR. Include ADD, DELETE, or MOVE statements for the changes you want to make. For each statement, provide VTAM definitions for the resources you want to change.

   Code the proper value (LOCAL) for the DRTYPE operand on the ADD and DELETE definition statements to indicate that the dynamic reconfiguration operation is for a connection through the local peripheral node. The default (DRTYPE=NCP) does not work in this case.
2. File the dynamic reconfiguration file in the VTAM definition library.
3. Make sure that you have deactivated all resources that must be inactive.
4. Issue the VARY DRDS command to invoke dynamic reconfiguration using the dynamic reconfiguration file.
5. Issue the VARY ACT command to activate newly added resources and resources you deactivated for the reconfiguration.

   To automatically activate a resource added by dynamic reconfiguration, the status of the higher-level node should be ACTIVE, and ISTATUS=ACTIVE should be specified on the newly added resource.

   To automatically activate a moved physical unit and its logical units that have ISTATUS=ACTIVE coded, specify ACTIVATE=YES on the PU definition statement following a MOVE statement.
6. Check the results of the dynamic reconfiguration by issuing a DISPLAY command for the affected resources.

If the NODELST start option is in effect when you perform a dynamic reconfiguration, VTAM records the name of each dynamic reconfiguration file in the NODELST file. When you restart VTAM after a halt or a failure, you can automatically reapply the changes in the dynamic reconfiguration files by specifying the name of the NODELST file on the CONFIG start option.

If a physical unit was dynamically moved before VTAM failed, it cannot be moved during a warm start. The activation of resources precedes the processing of dynamic reconfiguration files during warm start. Because dynamic reconfiguration depends upon resources being inactive, dynamic reconfiguration move operations are not performed. Deactivate the PU and issue the VARY DRDS command again.

Note: VARY ACT,UPDATE enables resources to be dynamically reconfigured and activated on warm start.

Adding resources

To add resources, code the PU and LU definition statements as you would when defining these resources. If you do not code the MAXDATA operand on the PU definition statement, a value of 261 bytes is used for a PU type 1 and a value of 265 bytes is used for a PU type 2. Code the operands on the PU and LU definition statements in the same format as they appear in the NCP major node. For example, ensure that PASSLIM and MAXOUT are coded correctly. Refer to the *NCP, SSP, and EP Resource Definition Reference* for the coding format of NCP operands.

In a DR file, explicitly code all operands from the GROUP, LINE, or PU to which you are adding the PU or LU. VTAM does not process most NCP-only operands and might issue a warning message, but the GP3174 and RETRIES=(,t,n) NCP operands can be used when adding PUs on a nonswitched SDLC link.

Sifting takes place within the hierarchy of minor nodes being added dynamically. Values for operands coded in the original hierarchy above the hierarchy being added with the DRDS deck do not sift down to the added resources. Therefore, if you want the values coded in the original hierarchy, code them explicitly in the DR file for the resources you are adding.

The most efficient way to add a physical unit and its associated logical units with a DR file is to specify them on the same ADD statement. It is not necessary to use separate ADD statements for the physical unit and each logical unit. Figure 59 on page 247 shows how to code a DR file to add PUs and LUs.

Deleting resources

To delete resources, code only the PU or LU names in the DR file. You do not have to code individual PU or LU operands. When you delete a PU, you also delete its associated LUs. Do not code the associated LU definition statements. Figure 60 on page 248 shows how to code a DR file to delete PUs and LUs.

Moving resources

To move resources using a DR file, code only the names of the PUs you want to move. Do not code associated LUs. VTAM automatically includes the associated LUs in the move. For each PU, you can optionally specify a new SDLC link station address and whether you want the PU and its LUs automatically activated after the move. Figure 61 on page 249 shows how to code a DR file to move PUs and LUs.

To change the SDLC link station address for a physical unit on a certain line, use the MOVE statement and specify the new address for the ADDR operand. Keep the FROM and TO line the same.

The MOVE operation through a DR file is only for PUs (and their associated LUs). To move an individual LU to another PU, use DELETE and then a subsequent ADD. (For efficient storage management, code all DELETEs before their related ADDs.)

Note: You can move a physical unit only under the same NCP major node. The physical unit cannot be moved from a line attached on one NCP to a line associated with another NCP.

Using the MODIFY DR technique

The MODIFY DR command allows you to move a PU (and its LUs) from one line to another line, delete a PU (and its LUs) from a line, or delete an LU from a PU.

Figure 60 on page 248 shows how to delete PUs and LUs using the MODIFY DR command. Figure 61 on page 249 shows how to move a PU and its LUs using the MODIFY DR command. You can change the SDLC address for a PU by requesting a move in which the new line and old line are the same, but the address is different.

The NCP to be reconfigured must be active, and the resource to be deleted or moved must be inactive.

Dynamic reconfiguration with the MODIFY DR command should be used only as a temporary method; a new NCP must be generated to reflect the changes as soon as possible. The NCP must be reloaded to reflect the permanent changes.

Dynamic reconfiguration changes made by MODIFY DR are not recorded in NODELST. Therefore, they are not automatically applied and can cause errors because they are not used during a warm start.

Dynamic reconfiguration of independent LUs

You can dynamically add or move independent LUs with the VARY DRDS command or the VARY ACT command with the UPDATE operand. You can also dynamically change operand values in an independent LU with the VARY ACT,UPDATE=ALL command. For a dynamic reconfiguration add or move, or for dynamically changing operand values, the processing takes place as shown in Table 12. For further information, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Table 12. Rules for multiple definition of resources

Existing resource	Input resource	Major node after integration	Notes
Dynamic CDRSC	DR-defined independent LU	ISTPDILU	1, 2
Predefined independent LU	DR-defined independent LU	ISTPDILU	1, 2
Predefined CDRSC	DR-defined independent LU	ISTPDILU	1, 3

Notes:

1. If the existing resource has a known real NETID, it must match the NETID of the input resource. If the real NETID of the existing resource is not known, the NETID is inherited from the input resource.
2. All operands from the input resource apply to the existing resource.
3. Only LU operands from the input resource apply to the existing resource.

Dynamic reconfiguration of frame-relay resources

Frame-relay PUs, which are type 1 PUs, are used in FRSESET (frame-relay switching equipment set) definition statements to create routing paths within an NCP. Each FRSESET definition statement can contain two to four frame-relay PUs, which define the primary routing partners and their optional backups. In the following example, PU A and PU B are the primary partners, while PU C and PU D are the backups. PU C is the backup for PU A, and PU D is the backup for PU B.

```
frse3a3 FRSESET  SUBPORTS=(A,B,C,D)
```

If one of the primary partners is deactivated, routing can continue between the backup for the inactive primary PU and the remaining primary PU. Backup to backup routing is not permitted.

The VARY DRDS, MODIFY DR, and VARY ACT,UPDATE=ADD|ALL commands can be used to dynamically reconfigure frame-relay PUs and FRSESET definition statements. To delete a PU from a FRSESET, use VARY DRDS, MODIFY DR, or VARY ACT,UPDATE=ALL. To add a PU to a FRSESET, use VARY ACT,UPDATE=ADD|ALL. You cannot move frame-relay PUs, but you can delete a PU and add it back in the desired position to accomplish a move.

The following rules apply when using dynamic reconfiguration for frame-relay resources:

- Each FRSESET definition statement must have a domain-unique name.
- The LMI (local management interface) PU must be statically defined as the first PU under a frame-relay line. The LMI PU cannot be used in a FRSESET and cannot be added (or deleted) dynamically.
- Each PU in a FRSESET must be on a different frame-relay line.
- A PU cannot be concurrently defined in multiple FRSESET definition statements.
- Use UPDATE=ALL for initial activation. If VTAMLST is modified prior to initial activation, using UPDATE=IMPLICIT can cause NCP and VTAM to have inconsistent definitions. If this occurs, issue a VARY ACT,UPDATE=ALL command to correct the problem.
- To modify a statically defined FRSESET, all PUs in the FRSESET must first be deleted.

  Note: Statically defined PUs that are deleted and then added back (for example, when adding backups to a statically defined FRSESET) are regarded as dynamic PUs.
- A FRSESET cannot contain both static and dynamic PUs.
- To delete a PU from a FRSESET, you must first deactivate the PU. Otherwise, the deletion fails and the PU is still present.
- Do not delete either of the primary PUs from a dynamically defined FRSESET. Otherwise, remaining PUs in the FRSESET may be deactivated, and the FRSESET cannot be used until a primary PU is added.
- PUs must be deleted from the frame-relay line and the FRSESET simultaneously.
- When adding PUs to a FRSESET, PUs must be added to the frame-relay line and the FRSESET simultaneously.

If you do not follow these rules, you might receive messages IST322I, IST323I, and IST1350I. Do the following to correct the problem:

- If the FRSESET was statically defined, delete all PUs and the FRSESET, correct the error, and then add back the PUs and the FRSESET.
- If the FRSESET was dynamically defined, only the PU or FRSESET in error must be corrected.

If any PU in a dynamic FRSESET is deactivated, all other active or contact pending PUs in the FRSESET that have no potential routing partner become inoperative. The inoperative PUs go through recovery processing and are again active, although not currently usable for routing because they have no partner.

Data traffic flowing over the deactivated PU could be momentarily halted while the switch to the backup PU is performed. If the backup is not available, data

traffic ceases. Data traffic also ends if the other primary PU is not available; backup-to-backup routing is not permitted.

If the data is routed through a backup PU and the primary PU becomes available, routing automatically switches back to the primary PU.

Dynamic reconfiguration and change of operands

For further information on frame-relay PUs, refer to *z/OS Communications Server: SNA Operation* and *z/OS Communications Server: SNA Resource Definition Reference*. Refer to *NCP, SSP, and EP Resource Definition Reference* and the *NCP, SSP, and EP Resource Definition Guide* for more information.

Implicit dynamic reconfiguration

Implicit dynamic reconfiguration allows you to reconfigure an NCP by changing the VTAM definition of the NCP and then reactivating the NCP. This procedure is disruptive in that deactivation of an NCP causes all existing sessions to be terminated. As soon as possible, you must regenerate and reload the NCP to reflect the changes. To use implicit dynamic reconfiguration:

1. Modify the existing NCP generation definition and include new definition statements for the PUs and LUs you want to add. You can only add PUs and LUs. You cannot delete or move them.
2. Issue a VARY INACT command to deactivate the NCP.
3. Issue a VARY ACT,UPDATE=IMPLICIT (the default) command to reactivate the NCP, using the updated generation definition.

For an example of adding PUs and LUs using implicit dynamic reconfiguration, see Figure 62 on page 250.

Dynamic reconfiguration examples

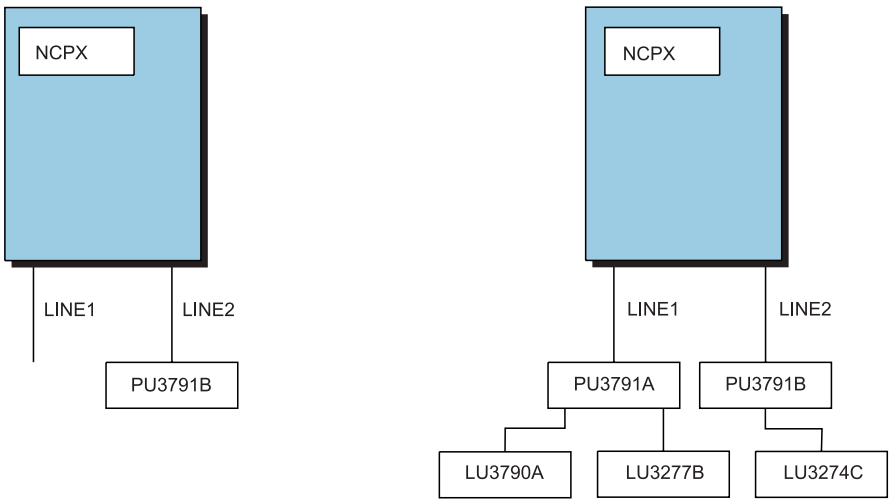
ADD: Figure 59 on page 247 shows how to add a resource using dynamic reconfiguration. You can modify the VTAMLST source for the desired major node and then use the VARY ACT,UPDATE=ALL|ADD command, including the member name of the data set containing the updated definition. Changes made in this way are permanent changes.

Alternatively, you can make temporary changes by creating a DR file and issuing the VARY DRDS command. These changes do not remain in effect when the NCP is regenerated and reloaded.

The MODIFY DR command cannot be used to add resources.

Original VTAMLST definition deck:

```
LINE1       LINE (Normally coded operands)
LINE2       LINE
PU3791B     PU
```



Using VARY ACT, UPDATE

Modified VTAMLST definition deck:

```
LINE1       LINE (Normally coded operands)
PU3791A     PU
LU3790A     LU
LU3277B     LU
LINE2       LINE
PU3791B     PU
LU3274C     LU
```

Operator command: VARY NET,ACT,UPDATE=ADD|ALL,ID=member name

Using VARY DRDS

DR File:

```
            VBUILD TYPE = DR
            ADD  TO = LINE1
PU3791A     PU     (Normally coded operands)
LU3790A     LU     (Normally coded operands)
LU3277B     LU     (Normally coded operands)
            ADD    TO=PU3791B
LU3274C     LU     (Normally coded operands)
```

Operator command: VARY NET,DRDS,ID=dr filename

Figure 59. DR ADD operation

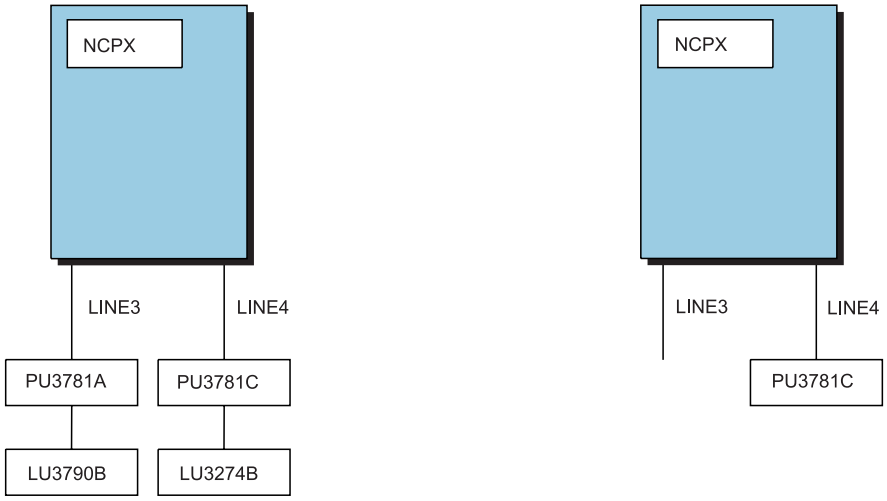
DELETE: Figure 60 on page 248 shows how to delete a resource using dynamic reconfiguration.

You can remove the PU and LUs from the VTAMLST definition and issue a VARY ACT,UPDATE=ALL command. Changes made in this way are permanent changes.

Alternatively, you can make temporary changes by creating a DR file and issuing the VARY DRDS command, or by using MODIFY DR commands. These changes do not remain in effect when the NCP is regenerated and reloaded.

Original VTAMLST definition deck:

```
        LINE3     LINE (Normally coded operands)
        PU3781A   PU
        LU3790B   LU
        LINE4     LINE
        PU3781C   PU
        LU3274B   LU
```



Using VARY ACT, UPDATE

Modified VTAMLST definition deck:

```
        LINE3     LINE (Normally coded operands)
        LINE4     LINE
        PU3781C   PU
```

Operator command: VARY NET,ACT,UPDATE=ALL,ID=member name

Using VARY DRDS

DR file:

```
        VBUILD TYPE  =DR
        DELETE       FROM=LINE3
PU3791A PU           (No operands required)
        DELETE       FROM=PU3781C
LU3274B LU           (No operands required)
```

Operator command: VARY NET,DRDS,ID=dr filename

Using MODIFY DR Operator Commands

```
MODIFY DR,TYPE=DELETE,FROM=LINE3,ID=PU3781A
MODIFY DR,TYPE=DELETE,FROM=PU3781C,ID=LU3274B
```

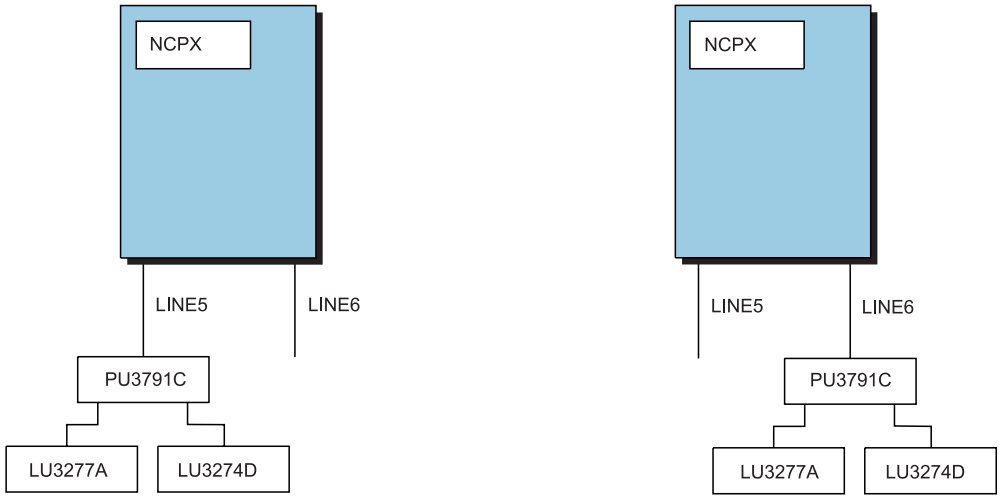
Figure 60. DR DELETE operation

MOVE: Figure 61 on page 249 shows how to move a resource using dynamic reconfiguration.

You can edit the VTAMLST definition to move the PU and LU definitions to the desired locations and issue a VARY ACT,UPDATE=ALL command. Changes made in this way are permanent changes.

Alternatively, you can make temporary changes by creating a DR file and issuing the VARY DRDS command, or by using MODIFY DR commands. These changes do not remain in effect when the NCP is regenerated and reloaded.

```
Original VTAMLST definition deck:
            LINE5       LINE (Normally coded operands)
            PU3791C     PU
            LU3277A     LU
            LU3274D     LU
            LINE6       LINE
```




```
Using VARY ACT, UPDATE
Modified VTAMLST definition deck:
            LINE5       LINE (Normally coded operands)
            LINE6       LINE
            PU3791C     PU
            LU3277A     LU
            LU3274D     LU
Operator command: VARY NET,ACT,UPDATE=ALL,ID=member name

Using VARY DRDS

DR file:
            VBUILD      TYPE=DR
            MOVE        TO=LINE6, FROM=LINE5
PU3791C     PU          (No operands required)

Operator command: VARY NET,DRDS,ID=dr filename

Using MODIFY DR  Operator Command

MODIFY DR,TYPE=MOVE,ID=PU3791C,FROM=LINE5,TO=LINE6
```

Figure 61. DR MOVE operation

ADD a PU and LU: Figure 62 on page 250 shows how to add a PU and LUs by implicit dynamic reconfiguration.

Note: IMPLICIT is the default for the UPDATE operand on the VARY ACT command.

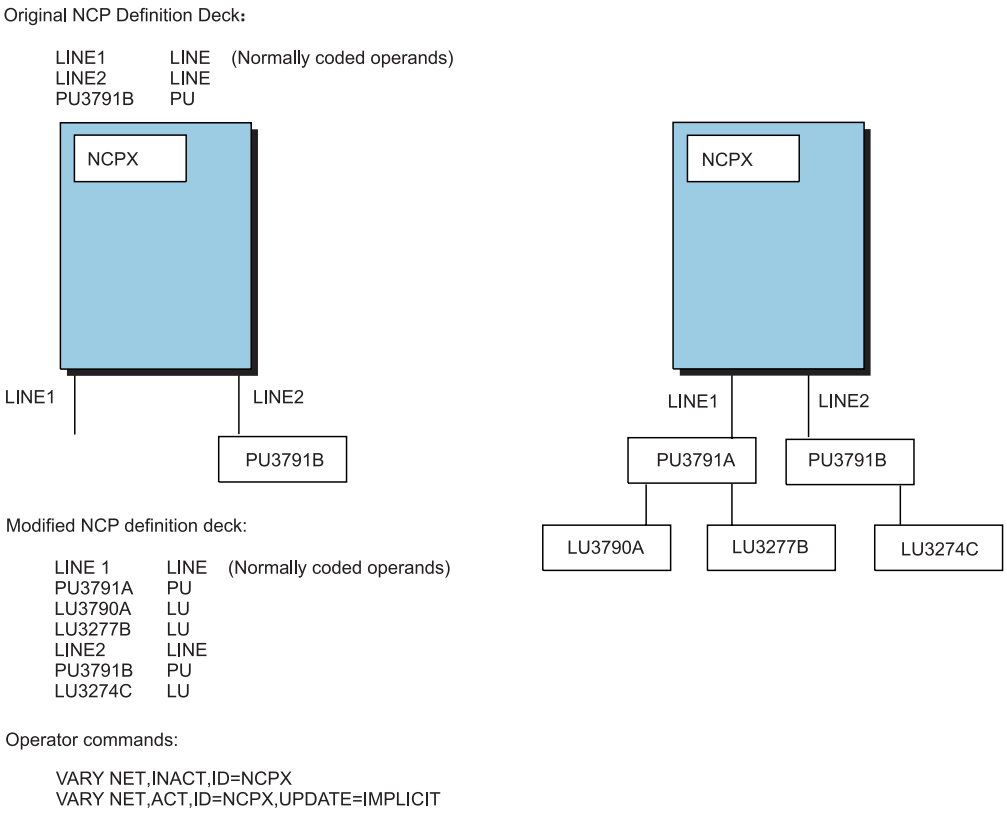


Figure 62. Sample implicit dynamic reconfiguration

Line backup through dynamic reconfiguration

Dynamic reconfiguration can be used to logically move physical and logical units (the station to be reconnected using switched network backup) to an alternate line. The alternate, physical, switched line is generated in the NCP as a nonswitched line. When a line to a multipoint station fails, the affected station can be deleted from its position as originally generated in the NCP and added to the switched link through dynamic reconfiguration or with the MODIFY DR command. It can then be physically connected by an operator dialing the remote modem. If several stations are attached by a remote "fan out" modem, all the physical units on that modem can be added to the same backup link with the DR ADD command.

The physical units and logical units can have the same names, so no changes to application programs are needed.

Figure 63 on page 251 illustrates dynamic reconfiguration without "fan out" modems.

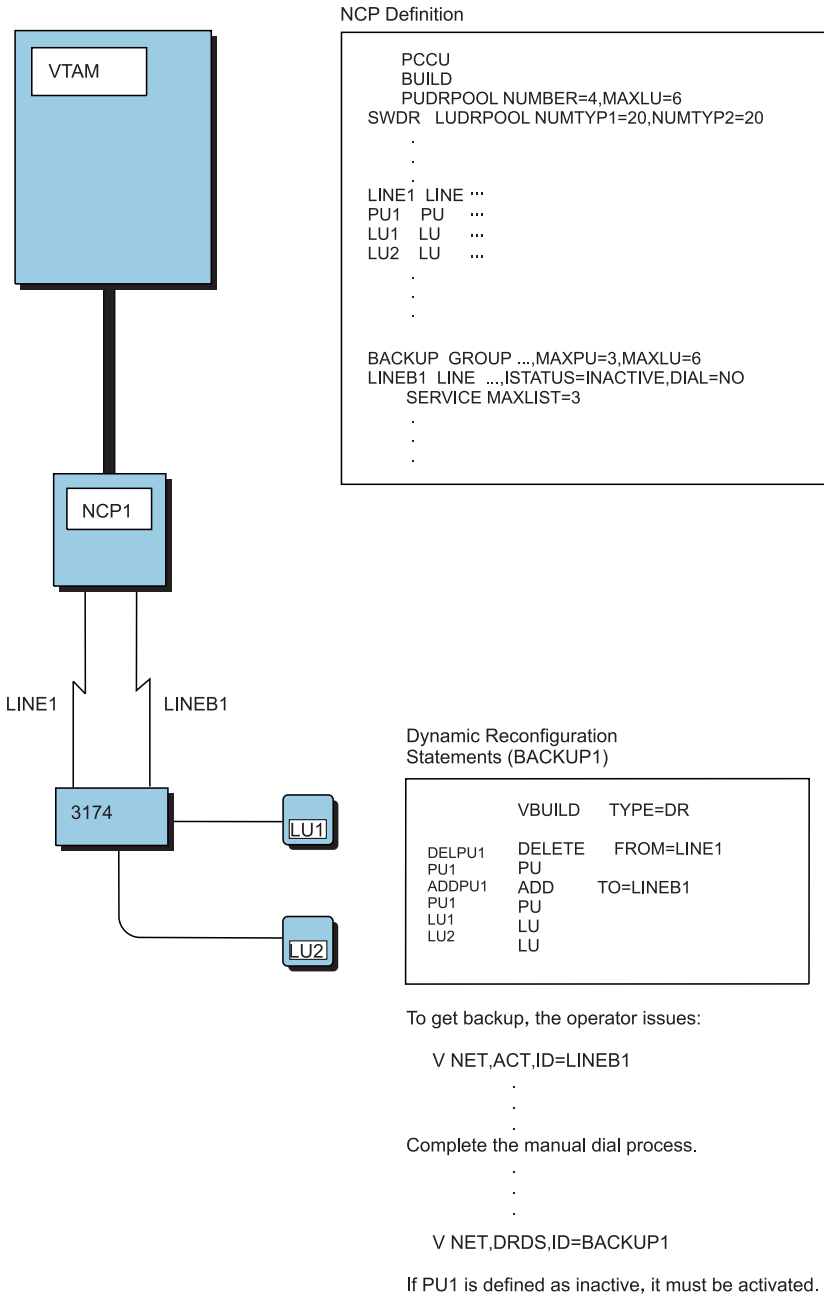


Figure 63. Line backup through dynamic reconfiguration

To define dynamic reconfiguration with both LPDA-1 modems and LPDA-2 DCEs:

1. Generate the NCP to include dynamic reconfiguration.
2. Generate alternate ports in the NCP as nonswitched lines.
   - For LPDA-1 modems, generate LPDA-1 lines.
   - For LPDA-2 DCEs, generate LPDA-2 lines.
3. Create the dynamic reconfiguration definitions and store them in the VTAM definition library. The file should include the following:
   - DELETE statements for the physical units being backed up
   - ADD statements to add these physical units to the backup link
4. Activate the backup line.

5. Enter a VARY DRDS command to invoke the dynamic reconfiguration. The backup switched line is now ready for use.

Chapter 10. Defining peripheral nodes

Peripheral nodes use local addresses for routing and require boundary function assistance from VTAM or the NCP. Logical units (LUs) are the ports through which users access the network. The type 1 and type 2 peripheral node architecture supports dependent LUs only. The type 2.1 peripheral node architecture supports independent and dependent LUs.

Defining type 2.1 peripheral nodes

For switched connections, PU definitions for type 2.1 peripheral nodes can be created dynamically on the VTAM host accepting the incoming call. For details, see Chapter 9, "Defining resources dynamically," on page 227.

If you code PU definition statements for type 2.1 peripheral nodes, consider the following.

- **Type 2.1 characteristics**

  Type 2.1 peripheral nodes can communicate with each other, as well as with application programs in VTAM. Communication between type 2.1 peripheral nodes can occur through the main communication path among the subarea nodes in the network. Type 2.1 peripheral nodes support link-level role negotiation, eliminate hierarchical control (no ACTPU or ACTLU unless PU requests), require fewer flows to establish sessions, and use dynamically assigned session identifiers rather than preassigned addresses, all of which require fewer system definitions.

  VTAM performs the directory services function for the subarea network and any APPN end nodes it serves. It locates the other session partner through a type 2.1 logical unit session request. VTAM is also involved in the session setup process when the type 2.1 session request traverses the subarea network, or when it is acting as a network node server for an APPN end node. VTAM or VTAM and NCP together appear and operate as a peer node to any type 2.1 peripheral node that attaches to the subarea network.

- **CPNAME operand**

  For a type 2.1 peripheral node on a switched or LAN connection, VTAM requires the coding of either the CPNAME operand or both the IDBLK and IDNUM operands on the PU definition statement in a switched major node. You can code all three. During activation of the node, VTAM uses the NETID (if specified) and the control point name (CPNAME) to find the PU and LU definition statements associated with the node. If CPNAME is not in the exchange ID (XID) or if CPNAME is not coded in VTAM, VTAM uses IDBLK and IDNUM instead. (To have VTAM first use IDBLK and IDNUM, use the SWNORDER=STATNID start option or specify the SWNORDER=STATNID operand on the GROUP or LINE definition statement.) VTAM also uses the control point name to locate switched node definitions, even if the boundary function does not provide type 2.1 peripheral node support.

- **XID services**

  For a type 2.1 peripheral node attached to the communication adapter using an SDLC link, the MODE operand on the LINE definition statement can be used to indicate whether VTAM performs the functions of the primary (PRI) or the secondary (SEC) link station. VTAM uses this PRI or SEC indication to determine its link-level role during the exchange identification (XID) processing

© Copyright IBM Corp. 1984, 2011

for the connection. Because VTAM does not negotiate the primary or secondary role and uses the MODE specification for the XID process, you need to know the link-level role capability of the other type 2.1 connection. For example, if you specify VTAM as primary (MODE=PRI) and the other peripheral node is also primary nonnegotiable, the connection fails.

If the SDLC line used is a switched link, the XMITDLY operand of the LINE definition statement can be used to specify the amount of time that VTAM should delay before performing XID processing after answering the incoming call. This delay allows the calling peripheral node to initiate XID transmission first and, therefore, to avoid VTAM initiating this process before the type 2.1 peripheral node is ready.

The XID=YES operand coded on the PU definition statement implies that the contact procedure used is for a type 2.1 peripheral node. VTAM determines whether the peripheral node is a type 2.1 or a type 2 by checking the XID value received from the node either during activation of the node or at CONTACT time for switched nodes.

- **XCA local area networks**

  Although VTAM can support type 2.1 peripheral nodes that are channel-attached through the external communication adapter (XCA), review the connectivity capabilities of each particular device to ensure it can be channel-attached as a type 2.1 peripheral node.

- **Defining a type 2.1 channel**

  If you are connecting a channel-attached type 2.1 PU to your host, code a VBUILD definition statement with TYPE=LOCAL specified.

```
LSNA3    VBUILD TYPE=LOCAL
LSNA3PU  PU    CUADDR=051,
               MAXBFRU=15,
               PUTYPE=2,
               SSCPFM=USSSCS,
               VPACING=0,
               XID=YES
```

Nonnative network type 2.1 connections

You can connect to type 2.1 PUs that are not in your network but instead are in a nonnative network. You can use the XNETALS start option or code the XNETALS operand on the GROUP, LINE, or PU statement, to network qualify any LUs that you connect to in a nonnative network.

If XNETALS is resolved to YES at the PU level, VTAM uses the NETID of the PU (rather than the VTAM network identifier) as the network qualifier of the LU name. For example, in Figure 64 on page 255, XNETALS is resolved to YES at the PU level and the LU in the attached type 2.1 peripheral node is recognized by VTAM as AS400.LU even though the network identifier of the subarea network is SANETX.

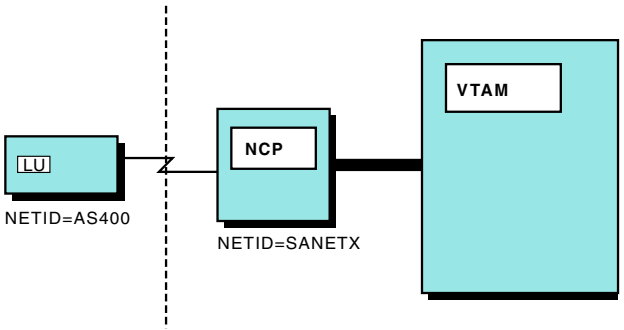


Figure 64. Nonnative network type 2.1 connection

Coding the NETID operand on the PU definition statement is optional (VTAM determines the network ID if not specified). However, if you specify XNETALS=YES, code the NETID operand if you want VTAM to perform dial-out operations to establish a session.

If XNETALS is resolved to NO at the PU level, and you do not code the NETID operand on the PU definition statement, VTAM assumes all connections are native unless the PU is a dependent LU server (DLUS) PU or a dependent LU requester (DLUR) PU. For information on DLUS and DLUR, see "Dependent LU server" on page 510.

If you code the NETID operand on the PU definition statement and XNETALS=NO is specified for the PU, the PU NETID must match the VTAM NETID.

Attaching peripheral nodes to VTAM

Peripheral nodes can be attached to VTAM with the following channels:
- Local non-SNA
- Local SNA
- Loop adapter attached
- External communication adapter

Local non-SNA connection

Each local non-SNA major node defines a set of channel-attached (local) non-SNA terminals. Each minor node represents a non-SNA terminal (such as a 3277).

You should always deactivate a local non-SNA major node before you take its terminals offline.

To define a local non-SNA peripheral node:

1. Code an LBUILD definition statement followed by a collection of VTAM definition statements. These definition statements should define each channel-attached non-SNA terminal as part of a logical set (group) of channel-attached non-SNA terminals.

   Remember the following:

   - You do not need to code a definition statement for the non-SNA cluster controller (3272 or compatible device) to which the terminal is attached. Non-SNA terminals connected to a single cluster controller do not have to be defined in a single major node. Different terminals on the same controller can be defined to VTAM in different local non-SNA major nodes.

- You cannot specify the channel device name when you are activating the logical unit for a channel-attached non-SNA device; the channel device name must be specified on the LOCAL definition statement for the device.
- Code one LBUILD definition statement for each logical group (major node) of channel-attached non-SNA terminals.
- Do not code a PU definition statement. The physical unit for a channel-attached non-SNA device is the VTAM host physical unit (ISTPUS), which is always active while VTAM is running. Therefore, only the local non-SNA major node and its logical units (representing the channel-attached devices) need to be defined and activated.

2. Code a LOCAL definition statement for each terminal (minor node) in the group. One or more LOCAL definition statements can be grouped with an LBUILD definition statement.

Following is an example of definition statements for a local non-SNA major node:

```
LC3270NS LBUILD                    LOCAL NON-SNA ATTACHMENT
LOC3277  LOCAL  CUADDR=00A,        CHANNEL UNIT ADDRESS
                TERM=3277,         TERMINAL TYPE

⋮
                DLOGMOD=S3270,     DEFAULT LOGON MODE TABLE ENTRY
                LOGAPPL=A50ACCTS   AUTOMATIC LOGON APPLICATION
```

Note: A local non-SNA terminal should not be defined to and activated by VTAM if its channel unit address is defined as an MVS console and allocated to console services. Activating a local non-SNA terminal whose channel unit address is in use by console services can cause VTAM, console services, or both, to abend.

Local SNA connection

Each local SNA major node defines a set of channel-attached (local) SNA cluster controllers. Each minor node represents an individual physical or logical unit.

Channel-attached SNA devices can be either statically or dynamically defined. For an overview of defining local SNA devices dynamically, see “Dynamic configuration of channel-attached devices” on page 230.

Defining a local SNA connection

To statically define a local SNA device:

1. Define a local SNA major node by coding a VBUILD definition statement for the major node and a separate PU or LU definition statement for each minor node.
2. Code a PU definition statement for each physical unit (such as a cluster controller) in the major node.

   Notes:

   a. A physical unit and all its logical units must be defined within a single major node.

   b. When activating a physical unit in a local SNA major node, supply the channel device name of the physical unit unless you have defined it in the PU definition statement. You can also specify the channel device name to override the value in the PU definition. To supply the channel device name, use the U operand of the VARY ACT command when activating the physical unit.

3. Code an LU definition statement for each logical unit placed after the associated PU definition statement. When you define a logical unit that logs on to VM, its name cannot be the same as any VM user ID in the system.

Sample local SNA connection

Figure 65 is a sample configuration with local SNA devices.

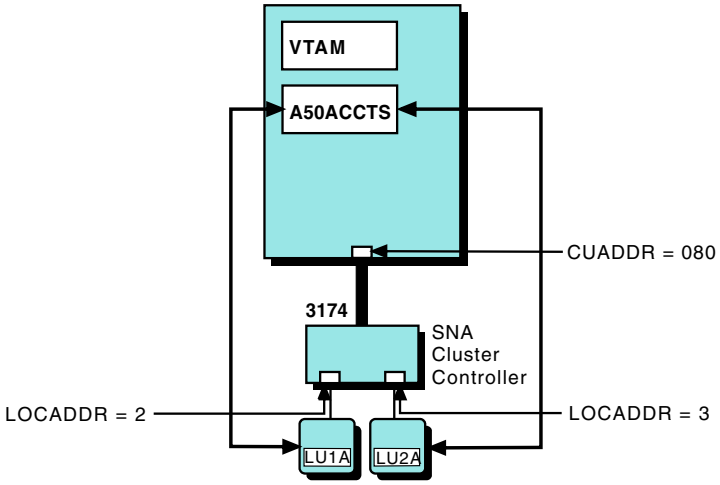


Figure 65. Local SNA devices

Following is an example of definition statements for Figure 65:

```
LOCALSNA        VBUILD  TYPE=LOCAL        LOCAL SNA ATTACHMENT
LPU1            PU      CUADDR=080,       CHANNEL UNIT ADDRESS
⋮
                        MODETAB=LOG3174   LOGON MODE TABLE
*
LU1A            LU      LOCADDR=2,        LOGICAL UNIT ADDRESS
⋮
                        MDLTAB=MDLTAB1,   MODEL NAME TABLE
                        MDLENT=ENTRY3,    MODEL NAME TABLE ENTRY
                        LOGAPPL=A50ACCTS  AUTOMATIC LOGON APPLICATION
*
LU2A            LU      LOCADDR=3,
⋮
                        MDLTAB=MDLTAB1,   MODEL NAME TABLE
                        MDLENT=ENTRY3,    MODEL NAME TABLE ENTRY
                        LOGAPPL=A50ACCTS  AUTOMATIC LOGON APPLICATION
```

If VTAM requires operator assistance to create a connection to a physical unit in a local SNA major node, VTAM displays a message indicating that a connection request was denied because the physical unit is offline. The connection request remains pending and the operator should either allow the connection request to complete by making the device available, or disallow the connection request by entering a VARY INOP command for the physical unit.

Enterprise Systems Connection (ESCON) channel attachment (ESA systems only)

The Enterprise Systems Connection (ESCON) channel is a high-bandwidth host attachment facility that can be used to connect SNA and non-SNA 3174 controllers, 3172 Nways interconnect controllers, 3746-900 controllers, and channel-attached hosts to VTAM running on MVS.

Loop-adapter-attached connection

Loop-adapter-attached devices are defined in a local SNA major node, and they communicate with VTAM application programs as if they were channel-attached SNA physical units and logical units.

External communication adapter (XCA) connections

This section describes the local area networks (LANs) that can connect to VTAM through an XCA. XCA can include an IBM 3172 Nways Interconnect Controller or an IBM S/390 or zSeries Open Systems Adapter.

VTAM and an XCA support the following types of LANs:

- Ethernet or Ethernet-type LAN
- Token ring
- Fiber Distributed Data Interface (FDDI)

This support allows SNA applications to access the supported LANs. TCP/IP and SNA applications can use the same physical attachment to an XCA at the same time.

VTAM connected through an XCA cannot load an NCP across a local area network.

Note: ATM networks accessed through LAN emulation appear to VTAM to be Ethernet or Ethernet-type LANs or token-ring networks and are defined to VTAM as such. ATM networks accessed through native ATM are defined to VTAM differently from those accessed through LAN emulation. See "ATM native connections" on page 69 for information about defining ATM native connections.

Defining an XCA configuration

To define an XCA configuration, code the following:

- One XCA major node to represent the PU in the XCA. This is not required, but is used for network management purposes. If you are running the NetView program, it is recommended that you code this definition statement. By defining this PU, you can have the XCA forward the same type of information that a 3174 cluster controller can forward. For example, any alerts detected by the XCA can be forwarded.
- One XCA major node for each LAN connected to the XCA.

  Note: If you have both subarea nodes (type 4 and type 5 nodes) and peripheral nodes (type 1, type 2, type 2.1, and subarea nodes that appear as type 2.1 nodes) attached to the LAN, you must code two GROUP definition statements in this XCA major node (one for the peripheral devices, and one for the subareas).
- Major nodes for any peripheral devices connected to the LAN.

Note: VTAM uses single-route broadcasts (rather than all-routes broadcasts) when attempting to connect to nodes on token-ring networks attached through the

IBM 3172 Interconnect Controller. When a node can be reached only through bridges, there must be a route from the IBM 3172 Interconnect Controller to the node, and that route must traverse bridges that are configured to route single-route broadcasts. Refer to the *Token Ring Network Architecture Reference* for more information on all-routes and single-route broadcasts.

Example XCA configuration

For example, Figure 66 illustrates an XCA configuration, including an IBM 3172 Nways Interconnect Controller:

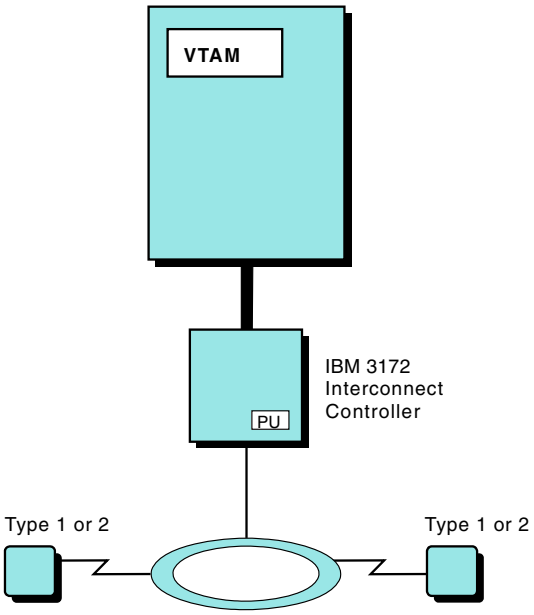


Figure 66. XCA connection in a single-domain environment

For the configuration in Figure 66, code the following:

1. An XCA major node to represent the PU in the IBM 3172 Nways Interconnect Controller:

```
XCACON1  VBUILD TYPE=XCA              XCA MAJOR NODE
PORT1    PORT  MEDIUM=BOXMGR,         3172 MANAGER
               CUADDR=500             CHANNEL UNIT ADDRESS
*
GROUP1   GROUP ISTATUS=ACTIVE         ACTIVATED AT GEN
LINE1    LINE
PU1      PU
```

2. Another XCA major node for the token ring:

```
XCACON2  VBUILD TYPE=XCA              XCA MAJOR NODE
PORT2    PORT  MEDIUM=RING,           TOKEN-RING
               SAPADDR=4,             SERVICE ACCESS POINT ADDRESS
               ADAPNO=1,              ADAPTER NUMBER
               CUADDR=BC0             CHANNEL UNIT ADDRESS
*
GROUP2A  GROUP DIAL=YES,              SWITCHED PERIPHERAL NODE
               CALL=INOUT,            CAN DIAL IN OR OUT
               ISTATUS=ACTIVE         ACTIVATED AT GEN
LINE2A   LINE  ANSWER=ON              PU CAN DIAL IN
PU2A     PU
GROUP2B  GROUP DIAL=YES,              SWITCHED PERIPHERAL NODE
               CALL=INOUT,            CAN DIAL IN OR OUT
               ISTATUS=ACTIVE         ACTIVATED AT GEN
LINE2B   LINE  ANSWER=ON              PU CAN DIAL IN
PU2B     PU
```

3. A switched major node for the two peripheral nodes attached to the LAN:

```
SWNODE1  VBUILD TYPE=SWNET
         GROUP
PUA      PU    ADDR=C1,
               DYNLU=NO,               NO DYNAMIC ALLOCATION OF CDRSCS
               PUTYPE=2,               PHYSICAL UNIT TYPE

⋮
               IDBLK=012,              BLOCK IDENTIFICATION
               IDNUM=00012             IDENTIFICATION NUMBER
*
LANPATHA PATH  GRPNM=GROUP2A,          DIAL-OUT PATH
               DIALNO=0104400000000013, PLACE HOLDER, SAP, AND MAC
               GID=2                   PATH GROUP
LUA      LU    LOCADDR=1               LOGICAL UNIT ADDRESS
*
PUB      PU    DYNLU=NO,               NO DYNAMIC ALLOCATION OF CDRSCS
               PUTYPE=2,               PHYSICAL UNIT TYPE

⋮
               CPNAME=NAME2            CPNAME IDENTIFICATION: TYPE 2.1 NODE
*
LANPATHB PATH  GRPNM=GROUP2B,          DIAL-OUT PATH
               DIALNO=0104400000000014, PLACE HOLDER, SAP, AND MAC
               GID=2                   PATH GROUP
LUB      LU    LOCADDR=2               DEPENDENT LOGICAL UNIT ADDRESS
*

⋮
```

Note: Switched PUs and LUs can be defined to VTAM using the model major node. See Chapter 9, "Defining resources dynamically," on page 227.

Automatically generating lines and physical units

To enable VTAM to automatically generate lines and PUs on an XCA major node, code the AUTOGEN operand on the GROUP definition statement in any external communication adapter major node where you have coded DIAL=YES. VTAM uses the values that you specify on the AUTOGEN operand to build the definitions for lines and PUs.

Names for the lines and PUs are generated using the naming convention `sssssnnn`, where:

sssss
Is the value specified by *line_seed_char* or *pu_seed_char* on the AUTOGEN operand.

- If you specify a single-character seed value, VTAM adds the channel unit address coded on the CUADDR operand on the PORT definition statement in the XCA major node to the single-character seed value, and then adds a sequential number to the end of the name.

  For ATM native connections: Four zeros are used in place of the CUA coded on the CUADDR operand.
- If you specify a seed value greater than a single character, VTAM uses the multiple-character seed value you specify, and then adds a sequential number to the end of the name.
- If you do not specify a seed character, VTAM uses the default (O for LINE definition statements; Q for PU definition statements).

nnn
Is a sequential hexadecimal number created by VTAM (X'0'–X'FFF').

For example, if you want to enable VTAM to generate 15 lines and physical units where each LINE name begins with L and each PU name begins with P, code the following:

```
PORT2    PORT  MEDIUM=RING,           TOKEN-RING
               SAPADDR=4,             SERVICE ACCESS POINT ADDRESS
               ADAPNO=1,              ADAPTER NUMBER
               CUADDR=BC0             CHANNEL UNIT ADDRESS
*
GROUP2A  GROUP DIAL=YES               SWITCHED PERIPHERAL NODE
               CALL=INOUT             CAN DIAL IN OR OUT
               ISTATUS=ACTIVE,        ACTIVATED AT GEN
               AUTOGEN=(15,L,P)       GENERATE LINES AND PUS
```

This generates 15 lines and 15 physical units. The lines will be named in the range L0BC0000–L0BC000F. The physical units will be named in the range P0BC0000–P0BC000F.

Attaching peripheral nodes to NCP

The following connections can be used to attach a peripheral node to NCP:
- Nonswitched SDLC lines
- Switched SDLC lines
- NCP/Token-Ring interconnection (NTRI)
- NCP Packet Switching Interface (NPSI)
- Binary synchronous (BSC) lines
- Start-stop (SS) lines

Notes:

1. Start-stop support is defined to VTAM as SNA physical and logical units. The start-stop protocol must be converted or enveloped in an SNA data stream using either a 3708 protocol converter or the NTO licensed program.
2. See *X.25 Network Control Program Packet Switching Interface Planning and Installation* for information on X.25 connections.

Defining type 2.1 peripheral nodes

You should consider the following when defining type 2.1 peripheral nodes.

PUDRPOOL definition statement
NCP must generate a pool of network addresses to establish switched connections to type 2.1 peripheral nodes. If you might add PUs using dynamic reconfiguration, you need to know the maximum number of concurrent type 2.1 peripheral node connections so that you can reserve the appropriate number of control blocks for PU addresses.

Note: The dynamic switched definition facility does not use addresses from the PUDRPOOL.

XMITDLY operand
You can specify XMITDLY on the LINE definition statement. This causes the NCP to delay the XID process for the specified period of time to avoid XID contention between the NCP and the type 2.1 peripheral node.

Nonswitched SDLC connection

You use the GROUP, LINE, PU, and LU definition statements in an NCP subarea node to define a peripheral node attached through a nonswitched SDLC connection. The primary VTAM-only operands that are used for this definition are associated with the following:

- Logon session parameters (DLOGMOD, MODETAB)
- Automatic logon usage (LOGAPPL)
- Unformatted session services facilities (USSTAB, SSCPFM)
- Model terminal support (MDLTAB, MDLENT, ASLTAB, ASLENT)
- Interpret table processing (LOGTAB)
- Pacing control (VPACING)
- Span of control (SPAN)

Following is a sample coding of a nonswitched SDLC connection between an NCP and a peripheral node:

```
G5S3276  GROUP ANS=CONTINUE,       ANS ACTION TOWARD PU
               DIAL=NO,            NONSWITCHED SDLC LINK
               LNCTL=SDLC,         LINE CONTROL DISCIPLINE
               MAXDATA=265,        MAX BYTES PU CAN RCV
               PACING=1,           NCP-LU PACING OPTIONS
               SPEED=9600,         9600bps LINE SPEED

⋮
               DLOGMOD=T3278M2,    (V) DEFAULT LOGMODE ENTRY
               LOGTAB=LOGONTAB,    (V) LOGON INTERPRET TABLE
               MODETAB=MT3274C,    (V) LOG MODE TABLE NAME
               SSCPFM=USSSCS,      (V) CHAR-CODED MSGS TO SSCP
               USSTAB=USS3270A,    (V) UNFMT SRVCS TABLE NAME
               VPACING=0           (V) VTAM-NCP PACING VALUE
*
L5032    LINE  ADDRESS=(32,HALF),  LINE ADDRESS — HALF DUPLEX

⋮
               ISTATUS=ACTIVE      (V) VTAM STATUS
*
         SERVICE ORDER=(C5032A)    SERVICE ORDER TABLE
*
P5032A   PU    ADDR=C1,            CLUSTER CONTROLLER ADDRESS
               PUTYPE=2,           PHYSICAL UNIT TYPE

⋮
               ISTATUS=ACTIVE      (V) VTAM STATUS
*
LU5032A1 LU    LOCADDR=2,

⋮
               ISTATUS=ACTIVE
```

Switched SDLC connection

A switched connection between NCP and a peripheral node can be any of the following:

- An SDLC switched line
- A switched virtual circuit (SVC) of an X.25 packet-switched data network
- A local area network (peripheral nodes only)

Define these connections within switched line groups in the NCP major node, and code either a switched major node or the dynamic switched definition facility to define switched resources to VTAM.

Virtual physical units and logical units with dial-in capability (such as NTO-controlled devices) can be dynamically defined using the configuration services XID exit routine. You can also use PU and LU definition statements in the switched or model major node. Virtual physical and logical units can be coded only for switched lines defined in an NCP major node.

Connecting to nonnative nodes

For a VTAM-initiated switched connection to nonnative nodes, the XNETALS=YES start option must be specified; otherwise, session establishment will fail.

Using error recovery activation

Code ANS=CONTINUE on the PU definition statement if you want the LU-LU sessions that traverse this NCP to be continued. (ANS=STOP is the default.) Coding the value of CONTINUE allows an LU-LU session on a switched connection to continue without disruption under either of the following conditions:

- VTAM fails
- The peripheral node is reactivated by VTAM

Sample switched SDLC connection to an NCP

Following is a sample switched major node. You can define multiple sets of switched devices. This enables the VTAM operator to use the VARY command to activate a selected subset of all the switched devices.

```
SWNODE1  VBUILD TYPE=SWNET,

:
P60SW1   PU    ADDR=C1,            SDLC ADDRESS OF STATION
               ANS=CONTINUE,       SESSION CONTINUES IF NCP IN SHUTDOWN
               IDNUM=08735,        IDENTIFICATION NUMBER
               IDBLK=017,          BLOCK IDENTIFICATION
               ISTATUS=ACTIVE,     VTAM INITIAL STATUS
               MAXDATA=256,        MAXIMUM DATA
               MAXOUT=7,           MAXIMUM FRAMES FOR LINK LEVEL RESPONSE
               MODETAB=MODTAB1,    LOGON MODE TABLE
               PACING=7,           BOUNDARY PACING
               PASSLIM=7,
               PUTYPE=2,           SNA CLUSTER CONTROLLER
               SSCPFM=FSS,         USS NOT INVOLVED IN LOGONS
               VPACING=14,

:
*
PATH01   PATH  DIALNO=1234567,     TELEPHONE NUMBER OF PN

:
               GID=5,              SWITCHED LINE GROUP IDENTIFIER
               GRPNM=G1SW111,      SWITCHED LINE GROUP NAME
               PID=1,              DIAL PATH IDENTIFIER
               REDIAL=3,           NUMBER OF DIAL RETRIES
               USE=YES             USABLE PATH
*
L60SLU01 LU    LOCADDR=1,          DEPENDENT LOGICAL UNIT ADDRESS

:
               ISTATUS=ACTIVE,     INITIAL STATUS ACTIVE
               LOGAPPL=A50PAYRL    CONTROLLING PRIMARY LU

:
               ISTATUS=ACTIVE,     INITIAL STATUS ACTIVE
               LOGAPPL=A50PAYRL    CONTROLLING PRIMARY LU
```

Controlling dial-in requests

A station identifier is used by VTAM to identify a physical unit dialing into VTAM on a switched line. The station identifier must be unique for each station in the network.

VTAM uses the station identifier to locate a PU by using:

1. The PU name. If this is a DLUR dial-in connection, PU name is constructed from V0E type F1 in REQACTPU.
2. The adjacent link station name, if the VERALSID=YES operand is specified on the PU definition statement. The ALS name is constructed from the V0E type F7 received in the XID.
3. The CPNAME and network ID specified on the CPNAME and NETID operands of the PU definition statement.
4. The CPNAME specified on the CPNAME operand of a PU definition statement with no network ID.
5. The station identification number for the PU (IDBLK and IDNUM operands on the PU definition statement). The station identification number is a 48-bit number that is used in the XID exchange during the dial procedure.

When locating a PU, to have VTAM use IDBLK and IDNUM before using CPNAME, code SWNORDER=STATNID on the LINE or GROUP definition statement or as a start option. To eliminate the possibility of using CPNAME as a search option for PU connection requests, define SWNORDER=(STATNID,ONLY) on the LINE or GROUP definition statement. DLUR served PUs can also be limited by specifying DLRORDER=(STATNID,ONLY) start option.

For APPN nodes, the TGN operand can also be specified on the PU definition statement. For details, see Chapter 17, “Implementing an APPN network,” on page 487.

If the peripheral node is a VTAM host defined as a type 2.1 physical unit, use the CPNAME operand; the value of the CPNAME operand must be the same as the value of the SSCPNAME start option in the attaching VTAM type 2.1 peripheral node. The IDBLK and IDNUM operands are ignored.

If the peripheral node is a type 1 or type 2 physical unit, use the IDBLK and IDNUM operands on the PU definition statement. VTAM uses the IDBLK and IDNUM values to build the station identifier.

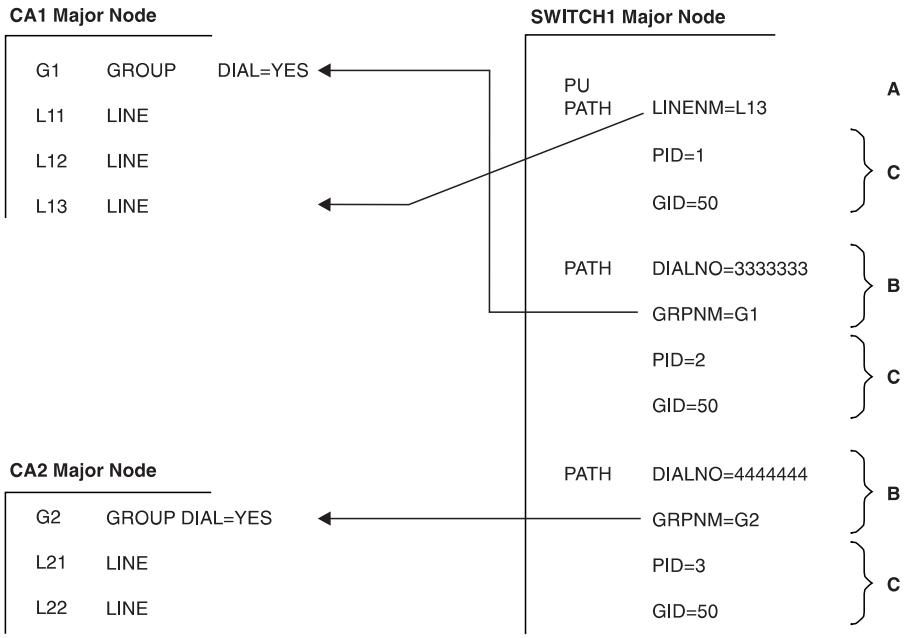
The value that you code for IDBLK is a 3-digit hexadecimal number. Each device type has its own device addressing scheme; you can find the IDBLK value in the component description manuals for the device. Note that this only applies to physical units; do not code IDBLK for logical units.

Because a subarea physical unit can have multiple connections, a connection identifier (IDNUM) is used to distinguish the multiple PU definition statements that might apply to a subarea physical unit.

As soon as a dial-in line has been activated and placed in answer mode (during VTAM startup or by the VTAM operator), the physical unit can dial that line number and establish communication with VTAM. A physical unit can be defined as call-in only, call-out only, or both.

Controlling dial-out requests

If you want VTAM to dial to a peripheral physical unit on a switched line, you need to give VTAM a path to contact the physical unit. You can identify the path using PATH definition statements. The PATH definition statement must be placed after the PU definition statement to define the paths over which that PU can be contacted. The PATH definition statement identifies to VTAM the line group and the dial characters to be used. Elements of a PATH are shown in Figure 67, parts A and B.



A These operands supply the essential information needed to create a direct call connection for a switched line.

B These operands supply the essential information needed to create a dial-out connection.

C These operands identify the path (PID) and indicate the path group (GID) with which it is associated.

Figure 67. Major elements of PATH definition statements for switched major node

VTAM searches the PATH definition statements for an available path in the order defined. If the line group contains more than one dial-out line, the first available line is tried. If contact cannot be established using that line group, VTAM tries again using the line group identified by the next PATH definition statement. If the line does not have automatic-call capability, the VTAM operator receives a message instructing the operator to manually dial the physical unit.

Two operands on the PATH definition statement, PID and GID, are used to identify each path and to assemble the paths into groups of paths (Figure 67, part C). Both operands are used by the VTAM operator. The PID operand identifies the path with respect to its associated physical unit. (The PID is the equivalent of the PATH definition statement label.) By citing the physical unit name and the PID, the VTAM operator can make that specific path usable or not usable.

The GID operand is used to associate the path with any other path or group of paths within the same switched major node. By naming the GID on the VARY PATH command, the VTAM operator can make that path and all similarly

identified paths usable or not usable. The criteria used to assign paths to particular groups are entirely up to you. One possibility is to group paths according to the type of call indicated by the dialing digits (for example, internal [extension] calls, local calls, tie-line or Wide Area Telephone Service [WATS] calls, and long-distance calls).

Dial-out operations for peripheral nodes (terminals) occur when a session is requested with a logical unit whose switched physical unit is not already connected to VTAM.

ID exchange

Access to LUs can be controlled through their identification. For SNA physical units on switched lines, use SNA ID verification. The IDs of SNA physical units on switched lines are built by VTAM from the PU definition statements. After a physical unit dials into or answers a dial-out request from an application program, an XID (exchange ID) command is sent to the physical unit. The physical unit responds with its station ID, which is sent to VTAM for verification against the IDs defined to VTAM. If the station ID is valid, VTAM sends its SSCP ID to the physical unit in the ACTPU command. If the station ID is not valid, VTAM disconnects the physical unit. When the identifications have been exchanged, normal physical connection procedures continue. This exchange of IDs provides a means for limiting access to the network.

Controlling disconnection

You can use the disconnect facility to specify whether VTAM will automatically terminate the SSCP-PU and SSCP-LU sessions when the last LU-LU session is terminated. This is most applicable when the connection is only temporary. VTAM also will terminate the SSCP-PU and SSCP-LU sessions if no LU-LU session is ever established within a given period of time.

You can also specify a period of time, beginning when the last LU-LU session is terminated, during which VTAM delays termination of the SSCP-PU and SSCP-LU sessions. Delayed disconnection is beneficial if there is usually a minimal time period between the ending of LU-LU sessions and the start of new LU-LU sessions under the same PU.

For a physical unit on a switched link, disconnection means that the dial connection is broken, and the SSCP-PU session is terminated. Therefore, the telephone is hung up, thus potentially saving unnecessary telephone charges. However, the PUs and LUs are not deactivated. They remain in connectable status awaiting another dial-in or dial-out operation, or for a channel-attached resource, awaiting the device to be powered-on. Sessions can be requested with the logical units, and the request simply causes the physical connection to be reestablished.

You can enable the disconnect facility by coding the DISCNT operand on the PU definition statement.

If you code DISCNT=YES for a peripheral node, all of the SSCP-LU sessions and the SSCP-PU session for that physical unit are terminated after all LU-LU sessions are terminated. If any of the logical units (or the physical unit) requests its own session termination, VTAM ignores the HOLD part of a character-coded logoff or the LAST-NOTLAST part of a field-formatted Terminate Self request. VTAM also rejects any attempt made by the physical unit to terminate its own SSCP session with a discontact request. If HOLD=YES is specified or defaulted on a LOGOFF request, DISCNT=YES is ignored.

If you specify DISCNT=DELAY for a peripheral node, all of the SSCP-LU sessions and the SSCP-PU session for that physical unit are terminated after all LU-LU sessions are terminated and after the delay period expires. In addition, you can specify a period of time VTAM delays termination of the SSCP-PU and SSCP-LU sessions if no LU-LU session is ever established. The delay period is determined using the DISCNTIM start option (default is 15 seconds), unless this global limit is overridden by a delay period specified for a particular PU on the DISCNT=DELAY operand or using the MODIFY RESOURCE command.

If you specify DISCNT=NO (the default) for a peripheral node, VTAM terminates the SSCP-LU and SSCP-PU sessions only when one of the following conditions is met:

- VTAM receives a normal discontact request from the physical unit, and all the LU-LU sessions associated with that physical unit have been terminated. All other existing LU-LU sessions are allowed to end normally.
- VTAM receives an immediate discontact request from the physical unit and immediately terminates any existing LU-LU sessions associated with that physical unit. This command overrides any previous normal discontact request from the physical unit.
- All LU-LU sessions have been terminated as a result of one of the following:
  - Character-coded logoff from the SLU with HOLD=NO
  - Terminate self request with LAST specified
  - VARY INACT or TERM command from the operator
  - Cross-domain takedown request
  - TERMSESS macro

  If any LU-LU sessions were terminated by any other means (for example, by an application program CLSDST macro), the SSCP-LU and SSCP-PU sessions are not terminated.

The type of processing VTAM performs depends not only on the type of connection but also on whether the type of node attached is peripheral or subarea. For more information on switched connections between subarea nodes, see Chapter 19, “Implementing a subarea network,” on page 529.

The disconnect facility also enables you to specify whether the “final-use” status is to be sent for the deactivation request of the physical unit. VTAM sends the “final-use” status in the deactivate physical unit request if DISCNT=YES,F is specified. If you code NF, “not-final-use” status is indicated. This “final-use” status does not apply when DISCNT=NO, nor does it have any effect on the VARY INACT command.

Defining switched resources used by multiple VTAMs

This section describes the coding methods that can be used when switched resources exist in a configuration that allows them to connect to a number of VTAMs in the network. There are two ways to define switched resources when the resource can be connected to multiple VTAMs.

- Define switched resources through the switched major node in each host.

  This method allows all VTAMs to activate the switched major node concurrently, causing the resource to be in CONCT state for each host. Once a connection is made through a particular host, the resource is in ACTIVE state for that host.

  A subsequent session request flowing through a host where the resource is in CONCT state causes a dial-out to the resource to fail. The host receiving the dial-out failure will reroute the request to another host. The request continues to

be routed until the resource is located and the session is established through the host where the resource is in ACTIVE state. The session request is rejected if the resource is not located in the network.

Note: For dial-out connections to an LU in another network, network ID (NETID) must be specified on the PU definition statement.

- Define switched resources through the switched major node and CDRSC definition statements.

  Using this method, all hosts contain both switched major node and CDRSC definition statements for the resource.

  – When the switched major node is activated, the associated PU and LUs are initially in a CONCT state.

  – When the CDRSC definitions are activated, the associated LUs are changed into a shadow resource and the CDRSC definitions become the real resource. The CDRSC definitions for the LUs are then put in a CONCT state.

  For additional information about shadow resources, see "Shadow resources" on page 549.

  During the activation processing started by a dial-in request from a PU, the PU is put in an ACTIVE state. In addition, VTAM changes the LU definitions to real and the CDRSC definitions to shadow. A result of this processing is that the owning VTAM recognizes the resource as real and all other VTAMs recognize the resource as a CDRSC.

  Other VTAMs in the network receiving a subsequent session request for the PU and its LUs recognize the resources as cross-domain resources. The session request is rerouted until it reaches the VTAM which has the real LU definition as ACTIVE. (Only the VTAM owning the PUs has the LUs as ACTIVE.)

  The disadvantage of this method is it requires a duplicate CDRSC to be defined for all LUs that might be the target of a search while the PU is connected to a VTAM.

NCP/Token Ring interconnection (NTRI)

Because the token-ring connection is defined to VTAM as though it was a switched SDLC link, code the following:

- NCP major node definitions
- VTAM switched major node

Also consider the effects of the DISCNT and HOLD operands as discussed in "Controlling disconnection" on page 266. HOLD=NO should not be specified for token-ring connections because it might cause errors in attached gateways that require operator intervention.

NCP major node definitions

In the NCP major node, define both physical and logical resources:

- The token-ring interface coupler (TIC) connection for NTRI support is defined as a physical connection, which is generated as a nonswitched duplex line.
- The peripheral nodes that are attached to the token ring are defined as logical connections, which are generated as switched half-duplex lines.

The following sample physical and logical definitions are used in an NCP subarea node to attach peripheral nodes to the token ring. Note that TRNPG1 is the LAN NTRI physical group, and TRNLG1 is the LAN NTRI logical group.

```
TRNPG1   GROUP ECLTYPE=(PHYSICAL,PERIPHERAL) LAN PHYSICAL GROUP
*
TRNL1    LINE  ADDRESS=(1088,FULL),          TIC ADDRESS
               LOCADD=400000003023,          LAN ADAPTER ADDRESS
               MAXTSL=1108,                  TRANSMIT MAX DATA (MAX)
               PORTADD=1,                    PHYSICAL PORT ADDRESS
               RCVBUFC=4095,                 RECEIVE MAX DATA
TRNPU1   PU
TRNLU1   LU    ISTATUS=INACTIVE
*
TRNLG1   GROUP ECLTYPE=(LOGICAL,PERIPHERAL),
               AUTOGEN=8,                    NUMBER DEVICES ON THE LAN
               ANS=CONTINUE,                 CONTINUE IF HOST LOST
               CALL=INOUT,                   CONNECTION OPTIONS
               PHYPORT=1                     ASSOCIATE WITH PHYSICAL PORT

⋮
               COMPOWN=YES
```

TIC address

The relative address of the TIC on the 3745 communication controller is identified by the ADDRESS operand of the LINE definition statement. The TIC address on the token ring is defined by the LOCADDR operand of the LINE definition statement and is a locally administered address in this example. Any peripheral node that is attached to the token ring and needs to communicate through the 3745 TIC must use this address.

Adapter type

The 3745 communication controller, with NCP Version 5 Release 3 and subsequent releases, supports a token-ring adapter type 2, which allows the token ring to operate at either four-megabyte or 16-megabyte speeds. For more detailed information on this support, refer to the *NCP, SSP, and EP Resource Definition Guide*.

Switched major node

The peripheral nodes and associated logical units that are attached to the token ring are defined in VTAM using a switched major node.

As shown in the following sample, the same coding that is normally used for an SDLC switched peripheral node is used for a token-ring connection.

```
TRNODE1  VBUILD TYPE=SWNET
P1088TR1 PU     ADDR=04,             SDLC ADDRESS OF STATION
                ANS=CONTINUE,        CONTINUES IF NCP IN SHUTDOWN (1)

⋮
                IDNUM=10881,         IDENTIFICATION NUMBER
                IDBLK=017,           BLOCK IDENTIFICATION
                ISTATUS=ACTIVE,      VTAM INITIAL STATUS
                MODETAB=MODTAB1      LOGON MODE TABLE
*
LU108802 LU     LOCADDR=2            DEPENDENT LOGICAL UNIT ADDRESS
*
LU108803 LU     LOCADDR=3            DEPENDENT LOGICAL UNIT ADDRESS
```

For information on the ANS operand and error recovery activation, see “Using error recovery activation” on page 263.

If VTAM is to establish the connection to the token-ring-attached peripheral node (for example, use an automatic logon [LOGAPPL] to an application), also code a PATH definition statement in the switched major node.

The DIALNO operand on the PATH definition statement for the callout operation can be used to specify the TIC number, the system access point address, and the token-ring address of the peripheral node.

In the following sample, the TIC number is 02, the system access point address is 04, and the locally administered token-ring address of the peripheral node is the last four digits of the user's telephone extension.

```
PATH01   PATH   DIALNO=0204400000008735,   PERIPHERAL NODE TOKEN-RING
                                           ADDRESS
⋮
                GRPNM=TRNLG1              TOKEN-RING LOGICAL GROUP NAME
```

You can also use the DLCADDR operand on the PATH definition statement for the callout operation to specify dial information. Use of DLCADDR allows specification of up to 250 bytes of dial information (versus the 32-character limit of the DIALNO operand). The information from the DIALNO operand above would be coded as follows using the DLCADDR operand. The DLCADDR operands specify the data link control type (TR for token ring), the port address, the DSAP, and the destination address, respectively.

```
PATH01    PATH   DLCADDR=(1,C,TR)
                 DLCADDR=(2,D,2)
                 DLCADDR=(3,D,4)
                 DLCADDR=(4,X,400000008735)
```

Note: DLCADDR information is used by the DLC provider, and VTAM verifies only the syntax of the DLCADDR operand.

Token ring security

You can restrict users that can dial in to your system. The name that you code for the GROUP definition statement in the NCP major node must match the GRPNM operand on the PATH definition statement in the switched major node. Therefore, the value that you code in the GRPNM operand on the PATH definition statement in the switched major node specifies the users in a group that can dial your system. Only the users in the group that you specify on the NCP major node can access your system.

BSC connection

A binary synchronous (BSC) connection for an NCP-attached peripheral node is defined by the following definition statements in the NCP major node:

- GROUP (defines BSC characteristics)
- LINE (defines the BSC link)
- CLUSTER (defines the BSC cluster controller)
- TERMINAL (defines each BSC terminal)

Note: Code PU=YES on the GROUP or LINE definition statement to inform VTAM that the cluster controller and the terminals attached to it are 3270 BSC devices that are to be treated as a physical unit and logical units, respectively.

The FEATUR2 operand describes machine features of the non-SNA device. In the following sample, FEATUR2=MODEL2 indicates that the default screen size for the BSC 3270 is 1920 characters.

```
GA03BSC2 GROUP PU=YES,
               CLOCKNG=DIRECT,
               CRITSIT=YES,
               DIAL=NO,
               DUPLEX=FULL,
```

```
              NEGPOLP=.2,
              REPLYTO=23,
              RETRIES=(2,1,2),
⋮
              TYPE=NCP,
              PAUSE=0,
              POLIMIT=(10,WAIT),
              POLLED=YES,
              LNCTL=BSC
*
LA03288  LINE  ADDRESS=(288,40-09)
CA032881 CLUSTER CUTYPE=3271,
              GPOLL=40407F7F
TA03288A TERMINAL ADDR=60604040,
              POLL=40404040,
              TERM=3277
TA03288B TERMINAL ADDR=6060C1C1,
              POLL=4040C1C1,
              TERM=3277
TA03288C TERMINAL ADDR=6060C2C2,
              POLL=4040C2C2,
              TERM=3277
TA03288D TERMINAL ADDR=6060C3C3,
              POLL=4040C3C3,
              TERM=3277
```

You can change the values of the PAUSE and POLIMIT operands, shown in the preceding sample code, using the MODIFY POLL and MODIFY NEGPOLL commands respectively. These values can be set as often as you want while the NCP is active. When the NCP is deactivated and later reactivated, the parameter values of the NCP definition are in effect.

If you define a configuration restart file for the NCP, the parameters for the PAUSE and POLIMIT operands are restored to their checkpointed values, regardless of whether you are using a WARM start for NCP reactivation.

Chapter 11. Defining LUs

Logical units (LUs) are the ports through which users access the network. Type 1 and type 2 peripheral node architecture support dependent LUs only. Type 2.1 peripheral node architecture supports independent and dependent LUs.

Independent LUs

Independent LUs are represented to VTAM as cross-domain resources (CDRSCs). Independent LUs can be coded with LOCADDR=0 in major nodes, coded as CDRSCs, or dynamically defined, but in all cases, VTAM represents them as CDRSCs. The independent LUs can access the network from multiple adjacent link stations, which can be in any domain. Even if you predefine independent LUs in device-type major nodes (NCP, local SNA, and switched major nodes), VTAM converts these definitions to CDRSCs when the major node is activated. Because independent LUs are represented as CDRSCs, when you use a DISPLAY command on an independent LU, the VTAM message responds with TYPE=CDRSC. However, message IST1131I shows a device type of `INDEPENDENT LU / CDRSC`.

Characteristics of independent LUs

Following are some characteristics of independent LUs.

Adjacent link stations

Adjacent link stations are type 2.1 physical units (or type 4 or type 5 PUs that appear as type 2.1 PUs) through which independent LUs can access the network. There is no hierarchical relationship between an adjacent link station (PU) and an independent LU. For more information about adjacent link stations and how independent LUs can use them to access the network, see "Using CDRSC definition statements for independent LUs" on page 278, and "Multiple connections between a type 2.1 node and a subarea node" on page 279.

Session capabilities

The session capabilities of an independent LU are significantly greater than those of dependent LUs. Independent LUs do not require SSCP-PU or SSCP-LU sessions. An independent LU can act as a primary LU or a secondary LU to initiate sessions. However, an independent LU must be the secondary logical unit if it is to be the destination LU (DLU) of a session.

Extended BIND

Independent LUs support the extended BIND, which contains information necessary to activate a session, to make use of features such as adaptive session pacing, and to identify a particular session. Extended BINDs always contain a fully qualified procedure-correlation identifier (a PCID qualified by the name and network identifier of the PCID generator). A BIND without this information is a nonextended BIND. Extended BINDs can contain network-qualified names if the subarea components in the session establishment path support them. An extended BIND is sent only if the boundary function (if applicable) and the destination LU support extended BIND.

Multiple sessions

An independent LU can also have multiple sessions with a single LU (parallel sessions) or with several different LUs. The independent LU can assume the role of primary or secondary for any one or all of the sessions.

© Copyright IBM Corp. 1984, 2011

If one or more sessions are required between LUs for application transaction program communication, the independent logical unit establishes the necessary session connections on behalf of those application transaction programs.

The BSBUF buffer pool is used to provide common storage for boundary type 2.1, type 2, and type 1 peripheral-node-session control blocks. Depending on how you code the buffer pool start option, the BSBUF buffer pool can expand to support more sessions and contract as less session control blocks are needed. Because independent LUs can have multiple sessions (including parallel sessions), you should determine the average and maximum number of sessions associated with independent LUs that are directly attached to that VTAM subarea node. You can then specify a value that prevents excessive buffer expansion and contraction.

Addressing

An independent LU in a type 2.1 peripheral node does not have a predefined address. Although VTAM definition statements or an NCP generation associates an independent LU with a network accessible unit, the address of that unit does not have to be coordinated with VTAM or NCP coding. For information on how NCP handles address assignments, see "Connecting independent LUs through NCPs" on page 281.

Alias name translation and independent LUs

Following are restrictions that apply to alias name translation and the independent LUs:

- If the PLU name received in the BIND from the LU is network-qualified, the SLU name must also be network-qualified.
- If the names are network qualified, they must be the real names (unless they are USERVAR names), or the session setup fails.
- Concurrent sessions to destination resources with the same name in different networks cannot be established when using alias-name translation. Concurrent sessions to destination resources with the same name in different networks can be established if the independent LU host is using NQNMODE=NQNAME to define the destination resources. For more information on NQNMODE=NQNAME, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.
- If the SLU name received in the BIND is network qualified, and the SLU is from a different network than the PLU, the SLU alias name in the PLU network will be the same as the SLU real name.

Dynamic reconfiguration and independent LUs

You can create and alter CDRSC major nodes as needed to effect the same level of support as dynamic reconfiguration provides. Because dynamic reconfiguration has meaning only if the independent LUs are predefined using LU definitions, it is no longer needed once you convert your independent LU definitions to CDRSCs.

Note: The ISTPDILU major node contains CDRSC definitions for independent LUs that are converted from LU definition statements by VTAM. The ISTPDILU major node is automatically activated by VTAM on initialization and cannot be activated or deactivated by operator commands.

When processing a dynamic reconfiguration definition that contains an independent LU definition to be added, VTAM converts that independent LU to a CDRSC and places it in the ISTPDILU major node. The PU named on the TO operand becomes an entry in the CDRSC default adjacent link station list.

If you specify a PU on the FROM operand of the dynamic reconfiguration DELETE command on an independent LU, VTAM deletes the network address associated with that independent LU that was predefined for that independent LU. Additionally, the PU is removed from the default adjacent link-station list if present.

To use the dynamic reconfiguration DELETE command on an independent LU, the independent LU must not have any sessions over the PU named on the FROM operand.

You cannot use the MODIFY DR command for independent LUs.

Defining independent LUs

You can define independent LUs in the following ways:

- You can have your independent LUs defined dynamically by VTAM during session establishment. See "Dynamic definition of independent LUs" on page 276.
- You can define CDRSC definition statements (or model CDRSC definition statements) to represent your independent LUs. See "Using CDRSC definition statements for independent LUs" on page 278.
- You can use standard LU definition statements with LOCADDR=0 coded. These LU definitions are converted into CDRSC definitions by VTAM when the major node is activated. This method is not recommended other than as a migration tool. Using this method, you cannot warm-start VTAM and have your independent LUs return to a before-failure condition. Independent LUs that are defined this way are converted to CDRSCs by VTAM and placed in the ISTPDILU major node. If a new resource is detected that matches an existing CDRSC, the new resource definition is integrated with the existing definition, and the resulting resource resides in the ISTPDILU major node. See "Automatic conversion of independent LU definition statements" on page 279.

For information on how VTAM handles multiple definitions of one independent LU resource, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Migration: It is recommended that you remove predefined independent LU definitions from your device-type major nodes and either let VTAM dynamically define CDRSC definitions for the independent LUs on a session-request basis or create CDRSC major nodes. If you decide to code CDRSC major nodes for your independent LUs and you previously had them defined to the NCP, you should regenerate your NCP to avoid any address mismatch problems.

One of the advantages of removing existing LU definition statements for independent LUs (and using either dynamic definition of independent LUs or coding CDRSC definition statements) is that predefined, NCP generated addresses are not used by the independent LU. Addresses are requested as needed and are returned when no longer needed.

You can use multiple connections and the dynamic selection of session connections without dynamically defining your independent LUs as CDRSCs.

Note: If you have session takeover operations that rely on VTAM creating substitute LUs, these takeover operations can fail.

Dynamic definition of independent LUs

Independent LUs attaching to VTAM through a type 2.1 PU can be dynamically defined as cross-domain resources. No system definition of the independent LU is required.

If the independent LU initiates the session, the adjacent link station (ALS), through which the independent LU contacts the network, does not have to be predefined to VTAM. For example, if the independent LU is attached through a token-ring network, the adjacent link station would be specified in the form of the medium access control address (MACADDR) of the host. VTAM discovers the resource during session startup and dynamically creates a CDRSC to represent the LU.

If the independent LU is the destination LU unit rather than the originating LU. and the destination LU is located on a LEN node, then you can dynamically define the independent LU using the adjacent link station (ALS) selection function of the session management exit routine. (An independent logical unit that is the destination LU on an APPN node has no such restriction.) Using this exit routine, you can provide VTAM with a method of contacting the independent LU. For information on the ALS selection function, see "Dynamic selection of session connections" on page 281. For information on writing a session management exit routine, refer to *z/OS Communications Server: SNA Customization*.

Note: The ALS selection function of the session management exit routine is not required if the destination LU is a LEN CP independent LU. Use the CPCDRSC=YES start option for this situation. If a session request is made to a LEN CP independent LU and CPCDRSC=YES is coded, VTAM builds a dynamic CDRSC to represent the LEN CP independent LU. As a result, other network resources will be able to establish sessions with the LEN CP independent LU before it has attempted to establish any sessions.

To have your independent LUs dynamically defined, perform the following steps:

1. Code DYNLU=YES in the start list, or use it on the START command when starting VTAM; code DYNLU=YES on the adjacent CP (ADJCP) definition statement for the adjacent APPN node; or code DYNLU=YES on all PU definition statements that represent links to the adjacent node. Coding DYNLU=YES enables VTAM to learn about the independent LU during session activation and dynamically define representations of the LU. If you code DYNLU=YES when starting VTAM (either in the start list or on the START command), you can restrict dynamically defined LUs from being used to connect to an adjacent node by coding DYNLU=NO on the ADJCP statement for the adjacent node or by coding DYNLU=NO on all PU definition statements that represent links to the adjacent node.

   Results: The DYNLU value is associated with an adjacent node when the first link to that adjacent node is activated. Once the DYNLU value is associated with the adjacent node, that value will be propagated to all other links when they are activated. See *z/OS Communications Server: SNA Resource Definition Reference* for further information on determining the source of the DYNLU value assigned to an adjacent CP and attached resources.
2. For virtual route transmission groups (VRTGs), the default for the CDRDYN start option is YES, enabling dynamic CDRSC definition for the host node. The CDRDYN operand can also be specified on the host CDRM definition statement, but it will be overridden by the CDRDYN start option. You must also code CDRSC=OPT on the CDRM definition statement for an adjacent node to allow dynamically defined CDRSCs for sessions with that node. If you code CDRDYN=YES on the host CDRM definition statement, or allow the CDRDYN

start option to default to YES, you can restrict dynamically defined CDRSCs from being used for sessions with an adjacent node by coding CDRSC=REQ (or allow it to default) on the CDRM definition statement for the adjacent node.

Results: The DYNLU value is set for an adjacent APPN node from the CDRSC keyword associated with the CDRM for that node when the first link that is activated to the adjacent node represents a virtual route transmission group (VRTG).

3. Set the CDRSCTI start option if you want to change the amount of time that the dynamic definitions of independent LUs are retained after session termination (default is 8 minutes).
4. Code the CPCDRSC=YES start option in a host that is connected to a LEN CP to allow dynamic definition of the LEN CP independent LU when it is the destination LU. DYNLU still controls dynamic definition of all independent LUs (including LEN CPs) when they are the originating LU.

When an independent LU is dynamically created by VTAM, the following defaults are used:

- ENCR=NONE
- RESSCB=0
- EAS=256
- MODETAB=value specified on the DYNMODTB start option, if coded (if the DYNMODTB start option is not coded, ISTINCLM is used)
- DLOGMOD=value specified on the DYNDLGMD start option, if coded

The PACING and VPACING operands specified on the PU definition statement associated with the session are sifted down to the CDRSC. The default logon mode table for dynamic CDRSCs is specified using the DYNMODTB start option or the DYNMODTB operand on the MODIFY VTAMOPTS command. You can also replace the existing table that is being used by dynamic CDRSCs by using the MODIFY TABLE,OPTION=LOAD command. If a default logon mode table for dynamic CDRSCs is not specified, ISTINCLM is used. The logon mode for each session is extracted from the BIND request that is passed to VTAM.

If the adjacent link station that the independent LU is using is being dynamically defined using dynamic switched definitions, you need to ensure that you have an entry for the mode you want to use in the logon mode table for the resource. If the mode entry is not found in the logon mode table for the resource or in the default logon mode table, the session fails. You can specify a default logon mode entry for dynamic CDRSCs using the DYNDLGMD start option or the DYNDLGMD operand on the MODIFY VTAMOPTS command. You can also override this value for a specific dynamic CDRSC using the MODIFY DEFAULTS or MODIFY RESOURCE commands.

Note: Using the MODIFY VTAMOPTS command to change the value of the DYNMODTB or DYNDLGMD start options does not affect dynamic CDRSCs that have already been created.

Dynamic CDRSCs are created and maintained in the ISTCDRDY major node. ISTCDRDY is activated automatically during VTAM initialization, and deactivated automatically during VTAM termination. The operator can deactivate ISTCDRDY with the VARY INACT command, in which case all dynamically defined CDRSCs are also deactivated and the dynamic CDRSC function is disabled. In addition, all sessions involving dynamically created CDRSCs (including CP-CP sessions with this host, if the partner CP was dynamically defined) are terminated.

For information on how VTAM handles multiple definitions of one independent LU resource, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

You can use dynamic definition of independent LU with other functions, such as multiple connections to a subarea, casual connection, or APPN.

Using CDRSC definition statements for independent LUs

You can code CDRSC definition statements (or model CDRSC definition statements) for your independent LUs and specify the adjacent link stations (PUs) that VTAM uses to contact the independent LU. You can specify the adjacent link stations either by using the ALSLIST operand on the CDRSC definition statement, or by using the adjacent link station selection function of the session management exit routine. For more information on using the exit routine, see "Dynamic selection of session connections" on page 281. For more information on defining model CDRSCs, see "Model definition of cross-domain resources" on page 538.

The following sample shows an independent LU defined using a CDRSC definition statement:

```
CDRSC    VBUILD TYPE=CDRSC
VRCDRSC  CDRSC ALSLIST=(SWPUAIO6,SWPUA107)   ADJACENT LINK STATION NAMES
               ALSREQ=YES,                   USE ALS LIST ONLY
               LOGAPPL=APPL1,                AUTOMATIC LOGON
               DLOGMOD=DSILGMOD,             LOGON MODE TABLE ENTRY
               RESSCB=3                      SESSION CONTROL BLOCKS
```

ALSLIST=*puname* indicates to VTAM that this resource is an independent LU defined as a CDRSC. The ALSLIST operand can list one or more adjacent link stations that VTAM uses. The ALSLIST operand can also include an ISTAPNPU entry, a reserved keyword that represents any APPN capable PU. This entry can be used instead of listing each APPN adjacent link station. For information on controlling connection to independent LUs that are accessible over both LEN and APPN connections, see "Establishing and controlling sessions" on page 519.

You can add and delete adjacent link stations in the list of adjacent link stations using the MODIFY ALSLIST command. For more information on this command, refer to *z/OS Communications Server: SNA Operation*.

You can use the adjacent link station selection function of the session management exit routine to dynamically add other adjacent link stations to the list and to perform other adjacent link station selection functions. For details, refer to *z/OS Communications Server: SNA Customization*.

You can use the ALSREQ operand or start option to control which adjacent link stations can be used for session traffic to independent LUs. The ALSREQ operand indicates whether this independent logical unit can establish sessions with an adjacent link station that is not defined by the ALSLIST operand. ALSREQ=YES means that a predefined adjacent link station is required for session establishment. If you code ALSREQ=NO, any adjacent link station can receive a session request for this LU.

Notes:

1. Even when ALSREQ=YES is specified, if any APPN adjacent link station name (or ISTAPNPU) exists in the ALS list for the CDRSC, any APPN adjacent link station can receive a session request for this LU.

2. Any operand normally coded on an LU definition statement can be coded on a CDRSC definition statement when defining an independent LU. These operands only have an effect when the CDRSC is an independent LU.

Automatic conversion of independent LU definition statements

The ISTPDILU major node contains CDRSC definitions for independent LUs that are converted by VTAM from LU definition statements. In converting the independent LU to a CDRSC, VTAM removes the implied hierarchy of the independent LU to a type 2.1 physical unit.

The converted CDRSCs can be used with multiple connections between a type 2.1 node and a subarea node. For this reason, the individual CDRSC minor nodes that are converted from LU definitions are not freed when the NCP (or other major node) in which they were originally defined is deactivated.

However, you can delete individual CDRSC minor nodes without deleting the entire major node. You can use the DELETE operand on the VARY INACT command on predefined CDRSC minor nodes.

When the LU definition statement is converted, the resource is represented as a CDRSC. This means that if the OWNER operand is specified on the LINE definition statement where the independent LU is predefined, it has no effect on the independent LU being converted to a CDRSC. The independent LU is converted to a CDRSC, but the PU under which it is predefined cannot be used as a session path until it is acquired by that host.

Multiple connections between a type 2.1 node and a subarea node

An independent LU can connect to the subarea network through only a type 2.1 PU. A type 2.1 PU can be connected to the subarea network through a type 4 or type 5 PU. The type 2.1 PU connections are called adjacent link stations. An independent LU can connect to multiple adjacent link stations. Thus, the independent LU can have multiple, simultaneous connections to the VTAM host.

If one connection fails, only the sessions using that connection must be reestablished; the other sessions are not affected. The independent LU can then use an alternative adjacent link station for any subsequent sessions. You can restart any sessions terminated by the first adjacent link-station failure and connect the independent LU through another adjacent link station.

Some of the independent LU session paths might involve type 2.1 nodes connected to a VTAM other than the VTAM of the desired session partner. For these sessions, the intermediate VTAM views the independent LU as a CDRSC. The CDRSC that represents the independent LU can also have other concurrent sessions that use PU type 2.1 nodes connected to the VTAM of the desired session partner.

If an independent LU is connected to VTAM by only LEN links and two application programs try to contact it, VTAM provides the first available adjacent link station specified on the ALSLIST operand for both application programs. Both application programs would then have parallel sessions through the same adjacent link station. If you want the application programs to use different adjacent link stations, use the adjacent link station selection function of the session management exit routine. For information on coding this exit routine, refer to *z/OS Communications Server: SNA Customization*.

Figure 68 shows how an independent LU can connect to VTAM with multiple connections.

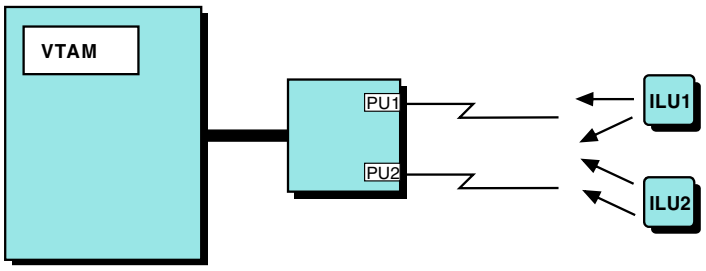


Figure 68. Independent LU with multiple connections to VTAM

Following are sample definition statements to define the configuration in Figure 68. Note that you can support independent LU as either destination logical units, originating LUs, or both:

- For independent LUs as originating LUs, code DYNLU=YES in your start list. VTAM determines the adjacent link stations dynamically.
- For independent LUs as destination LUs, code CDRSC minor nodes in a CDRSC major node. If the independent LUs are connected to VTAM by only LEN connections, include the ALSLIST operand.

Start list (ATCSTRxx)

If the independent LUs are originating LUs, code the start option for dynamic definition of independent LUs (DYNLU=YES).

NCP major node

Code the following (do not code LU definition statements):

```
⋮
         LUDRPOOL NUMILU=2        NUMBER OF INDEPENDENT LUS
LGROUP   GROUP DIAL=NO            LEASED LINE
LLINE    LINE
⋮
PU1      PU
⋮
SGROUP   GROUP DIAL=YES           SWITCHED LINE
SLINE    LINE
⋮
PU2      PU
⋮
```

Switched major node

Code a switched major node:

```
VBUILD   TYPE=SWNET
PUA      PU    ADDR=03,                  STATION ADDRESS
               CPNAME=A500N,             CONTROL POINT NAME (TYPE 2.1)
               DISCNT=YES,               DISCONNECT FACILITY
               PUTYPE=2                  PU TYPE 2.1
```

Note: You can code the DYNLU operand on the PU statement rather than in the start list.

CDRSC major node

If the independent LUs are destination LUs, define your independent LUs as CDRSCs:

```
VBUILD   TYPE=CDRSC
ILU1     CDRSC ALSLIST=(PU1,PU2),          ADJACENT LINK STATION NAMES
               ALSREQ=YES                  USE ALS LIST ONLY
ILU2     CDRSC ALSLIST=(PU1,PU2),          ADJACENT LINK STATION NAMES
               ALSREQ=YES                  USE ALS LIST ONLY
```

Dynamic selection of session connections

If an independent LU has an existing leased or switched LEN connection (and no APPN connection), VTAM can dynamically select session paths for outbound sessions from the boundary function to the independent LU. VTAM attempts to use leased connections first, and then tries active switched connections. If the first adjacent link station in the adjacent link station list is not available, VTAM tries to use an alternate connection from a default adjacent link station list that VTAM creates. When the first adjacent link station (the primary connection) does become available, VTAM uses that adjacent link station for any new sessions. However, sessions already allocated to the alternate adjacent link station do not change.

To control selection of an alternate session path when an independent LU is connected to VTAM by only LEN links, use the ALS selection function of the session management exit routine. VTAM then establishes the path between the independent LU and the boundary function; this continues to be the path for the remainder of that session. You can select a different adjacent link station for each session.

Using the ALSREQ start option, you can limit the adjacent link stations that an independent LU can use.

You can code the session management exit routine to choose a PU from the default list or another PU by overriding the default PU list. For information on writing a session management exit routine, refer to *z/OS Communications Server: SNA Customization*.

The default adjacent link station chosen must:
- Act as a type 2.1 PU
- Reside in a major node that supports independent LUs
- Be active, if it is nonswitched
- Be active, connectable, or pending dial if it is switched

Connecting independent LUs through NCPs

You should consider the following when connecting independent LUs through NCPs. For further information, refer to the *NCP, SSP, and EP Resource Definition Guide*.

LU address pool

NCP must generate a pool of network addresses to establish switched connections. The address pool is generated in NCP using the LUDRPOOL definition statement. The NUMILU operand on the LUDRPOOL definition statement specifies the number of independent LU switched connections. You need to know the maximum number of concurrent independent LU switched connections so that you can reserve the appropriate number of independent LU control blocks in the NCP.

Primary LU address pool

When an independent LU acts as the primary LU by initiating a session, VTAM and NCP must assign another network address to the PLU. The AUXADDR operand on the BUILD definition statement reserves this address pool. Because this is a predefined pool, you should know the maximum number of concurrent PLU

sessions that can be established by all independent LUs that are attached to that NCP. Otherwise, session establishment for an independent LU, acting as the primary LU, fails when the pool is exhausted.

Session information pool

You need to know the maximum number of sessions associated with all of the type 2.1 peripheral nodes attached to the NCP subarea node. NCP reserves a pool of control blocks for maintaining information about these sessions. The pool of control blocks is established during the NCP generation process using the ADDSESS operand on the BUILD definition statement. If this pool is exhausted in the NCP, session establishment fails because the pool is fixed and cannot expand dynamically.

Reserving session control blocks

You can also reserve session control blocks for each independent LU by specifying a value for the RESSCB operand on the LU definition statement.

Limiting sessions

To limit the number of sessions a single independent LU can establish per link station, you can use the MAXSESS operand on the following definition statements:

- The LU definition statement in an NCP major node, switched major node, or dynamic reconfiguration file
- The PU definition statement in a model major node
- The CDRSC definition statement in a cross-domain resource major node

This action prevents a single independent LU from using all of the session control blocks generated in the NCP.

Improving session establishment performance

Use of the EAS operand can improve the performance of VTAM in establishing independent LU sessions. The EAS operand on the LU or CDRSC definition statement estimates the number of concurrent sessions for the logical unit. For performance reasons, if you anticipate that a particular LU might have more than 256 concurrent sessions (the EAS default), increase this number to the nearest multiple of 256.

Authorized transmission priority for LEN connections

Normally it is desirable for a session between two independent LUs through a subarea network to use the same transmission priority for both type 2.1 LEN connections (entry and exit). By default, when VTAM receives a BIND over a type 2.1 LEN connection, the transmission priority field received on the BFINIT is passed back on the BFCINIT and through the subarea network on the BIND. This allows this same transmission priority to be used for the session by the type 2.1 LEN connection on exit from the subarea network.

To have VTAM ignore the transmission priority specified on the BFINIT, use the AUTHLEN=NO start option or the AUTHLEN=NO operand on a particular PU definition statement (or on a GROUP or LINE statement to sift down to the PU). VTAM then sets the transmission priority field in the BFCINIT to B'01' (medium priority), regardless of the transmission priority specified on the initial BFINIT.

Restrictions on using independent LUs

Following are some restrictions that apply to independent LUs:

- An independent logical unit cannot be the destination LU (DLU) of a session, unless it is the secondary LU (SLU). An independent LU cannot function as the

primary logical unit for an autologon (logappl) session. However, an independent LU can be the SLU of an autologon (logappl) session, provided that the independent LU resides on one of the following nodes:

- A peripheral (type 2.1) node that is APPN-attached to the same APPN network as the VTAM from which the independent LU autologon session will be initiated
- A peripheral (type 2.1) node that is LEN-attached directly to the VTAM (or an owned NCP) from which the independent LU autologon session will be initiated. Autologon is not supported for independent LUs residing on peripheral (type 2.1) nodes that are LEN-attached to non-VTAM/NCP nodes (for example, when independent LUs are migrated from being NCP-attached to being DLUR-attached). This is because the independent LU autologon session is initiated by VTAM when the adjacent link station to the LEN-attached node is activated; if a LEN-attached independent LU is not directly attached to VTAM (or an owned NCP), then VTAM has no knowledge of the activation of that link station, and therefore cannot reliably initiate the autologon session.

- An independent LU can initiate sessions as either a PLU or SLU.
- A dependent LU can have only one active LU-LU session at a time. Thus, there is no session with the controlling application program when the LU is in session with another application program. For independent LUs, this is not the case. VTAM initiates a session between an LU and its controlling application program, even though there are already active LU-LU sessions. Each time that a session with a PLU that is not the controlling application program ends, and the independent LU does not already have a session with the controlling application program, a session is established with the controlling application program. For more information on automatic logon with independent LUs, see "Automatic logons" on page 300.
- The name interpretation table function is not supported for independent LU sessions.
- The SSCP-PU session is optional and can be requested by the type 2.1 node during the XID exchange. However, the SSCP-PU session is one of the mechanisms that the type 2.1 node uses to transport network management device information (alerts) to the SSCP for delivery to the NetView program. An alternative mechanism used to report alerts is through a network management focal point function.

Dependent logical units

A dependent LU must be activated, controlled, and owned by a VTAM SSCP. Ownership of dependent LUs is determined when the SSCP establishes an SSCP-LU session. Dependent LUs have the following characteristics:

- They generally have unique, fixed, preassigned local addresses within the PU that are specified on the NCP or VTAM definition statements using the LOCADDR operand of the LU definition statement. With dynamic definition and with switched connections, the address is defined during the PU and LU activation process. For more information, see "Defining dependent LUs dynamically" on page 284.
- They are subordinate to a PU. If defined manually, they must be defined under a PU.
- They require an SSCP-LU session and an SSCP-PU session to be active to VTAM; these sessions must be established before any LU-LU session can be activated.
- They normally act as SLU only.

- They can have only a single session. Even an LU that supports LU 6.2 protocols can have only one session if it is defined as a dependent LU.

Defining dependent LUs dynamically

You can enable VTAM to define dependent LUs dynamically. One way to do this is to use the IBM-supplied selection of definitions for dependent LUs (SDDLU) exit routine. With this exit, VTAM can define LUs dynamically when it receives information from the PU that the device is powered on, rather than when the major node is activated. The dependent LUs must be attached to the host through PUs that support Reply/PSID NMVTs, request units that are used for dynamic definition of dependent logical units. For example, you can use a 3174 cluster controller because it supports these request units.

VTAM uses model LU definition statements to build LU definitions. Define model LU definition statements based on information the PU sends to VTAM about the logical unit device. The default information that the PU sends is the seven-character machine type and model number. When the PU is activated or when a device powers on, the PU sends a Reply/PSID NMVT request unit to VTAM, which can trigger VTAM to build or change LU definitions. This NMVT contains the local address of each LU, a power indicator (on or off), the machine type and model number of the device, and optionally other device-dependent information needed to define the LUs. VTAM uses this information to choose an appropriate model LU definition statement to build an LU definition.

LUGROUP operand: To enable this facility, code the LUGROUP operand on the PU definition statement for the PU, and code an LU group major node. The LUGROUP operand on the PU definition statement specifies the name of the model LU definition group that VTAM uses when dynamically defining LUs. The LU group major node contains the model definition statements. Dynamic definitions for LUs are built using the model LU definitions contained in this major node.

Note: Any operands that you code on the PU definition statement will not sift down to dynamically created LU definition statements.

After you enable this facility, VTAM uses the IBM-supplied selection of definitions for dependent LUs (SDDLU) exit routine to generate LU names for the dynamically built LU definition statements. The SDDLU exit routine controls the selection of the model LU and the LU name. If you prefer to use your own naming convention or method of selecting an LU model, you can write your own exit routine. For information on coding your own exit routine, refer to *z/OS Communications Server: SNA Customization*.

If you use the IBM-supplied SDDLU exit routine, code the LUSEED and LUGROUP operands on the PU definition statement. The LUSEED operand provides a pattern name that is used with the SDDLU exit routine to create a name for the dynamically created LU definition statements.

Figure 69 illustrates VTAM dynamically defining dependent LUs attached through a 3174 cluster controller, and the steps listed after the table explain the process.

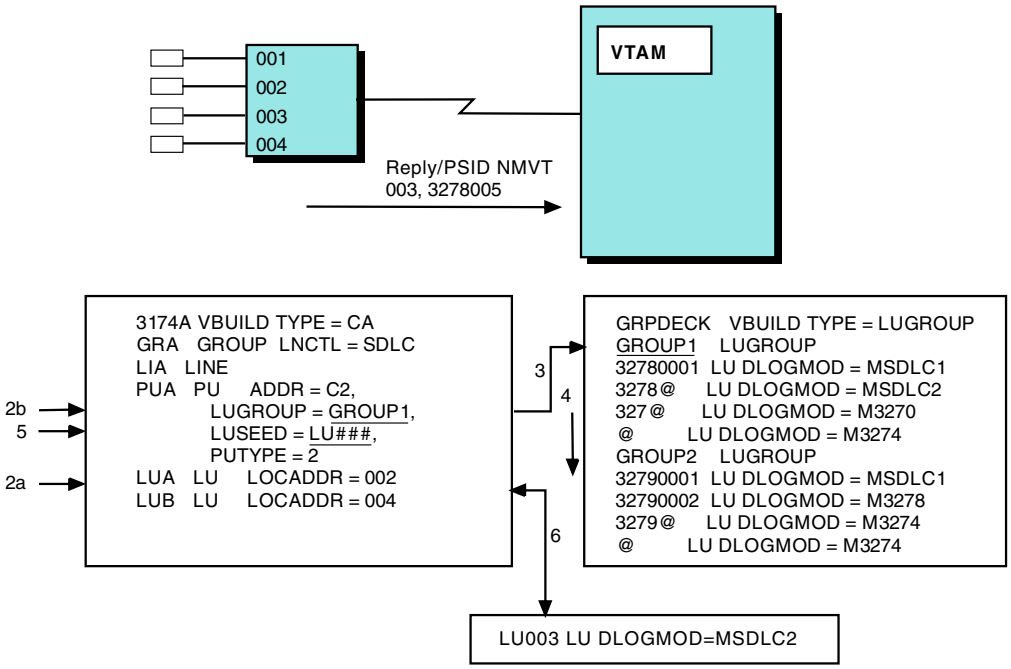


Figure 69. Definition building for dynamically defined dependent LUs

1. The LU at local address 003 powers on. The 3174 sends VTAM a Reply/PSID NMVT indicating a 3278 model 005 device has powered on at address 003.
2. VTAM uses the Reply/PSID NMVT to do the following:
   a. Check the 3174 PU definition statement for a definition of an LU at address 003
   b. Determine if the 3174 supports dynamic definition of dependent LUs (LUGROUP operand is coded), if the 3174 PU definition statement has no definition of an LU at address 003.
3. VTAM matches the LU group name specified in the 3174 PU definition statement with an LU group name in the LUGROUP definition statement.
4. The default SDDLU exit routine uses the seven-character machine type and model number as the model LU name. VTAM serially checks the LU group for a model LU statement that matches the model name supplied by the SDDLU exit routine. In this example, VTAM finds the second definition statement (3278@). The characteristics specified in that statement are used.
5. To create the LU name, the default SDDLU exit routine uses the value on the LUSEED operand. In this example, the three # characters specified on the LUSEED operand are replaced with the numeric local address. Therefore, the name of the LU is LU003. For more options, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.
6. VTAM builds a dynamic definition for the LU. The dynamic definition is rebuilt if that device powers off and a new device powers on at the same local address. Also, the dynamic definition is rebuilt if the device powers off and powers back on after its LU group major node has been deactivated and then reactivated.

Chapter 12. Establishing and controlling SNA sessions

Logical units (LUs) are defined to VTAM to provide access services to network resources. To tailor sessions, VTAM uses the unformatted system service (USS), interpret, logon mode, and Class of Service tables. VTAM also uses the logon-interpret routines and the virtual route selection and session management exit routines to authorize, establish, or terminate sessions. You can replace or modify these tables and routines. This, along with operands that can be specified in VTAM definitions, allows VTAM to adapt to your needs. The *z/OS Communications Server: SNA Resource Definition Reference* describes how to replace or modify the tables used in session establishment and termination and how to specify the network definition operands that relate to sessions. *z/OS Communications Server: SNA Programming* describes how VTAM initiates, controls, and terminates sessions. *z/OS Communications Server: SNA Customization* describes the exit routines.

Sessions between LUs can be initiated by the following:

- Session-initiation requests from a logical unit
- Automatic logon
- The application program, using API requests (for example, OPNDST, SIMLOGON, or REQSESS)
- The VTAM operator using, for example, VARY LOGON commands

Normally, sessions are terminated by one of the following:

- Session-termination requests from an LU
- The application program, using API requests (for example, CLSDST or TERMSESS)
- The VTAM operator, using VARY resource inactivation commands

z/OS Communications Server: SNA Programming describes the use of VTAM application programming interface (API) macros to establish and terminate sessions.

The following list shows where to find more information about establishing and controlling SNA sessions.

- "Multicultural support" on page 288
- "Model name table" on page 290
- "Associated LU table" on page 291
- "Selecting session parameters for the logon mode table" on page 292
- "Automatic logons" on page 300
- "Session management exits" on page 305
- "Session-level pacing" on page 307
- "Logon and logoff requests from dependent logical units" on page 321
- "Unformatted logon requests using mixed-case passwords" on page 322

© Copyright IBM Corp. 1984, 2011

Multicultural support

With multicultural support you can tailor messages and commands for users. Using the MVS Message Service (MMS) or the LANGTAB USS tables, you can define USS messages in several different languages.

Also, by default, the language passed to a VTAM application program comes from the logon mode entry specified by the USS LOGMODE parameter. With multicultural support, the language passed to the application program can be specified by the user, independent of the LOGMODE parameter.

For further information about multicultural support, see "Multicultural support for end-user USS messages and commands" and "Multicultural support for the language passed to application programs" on page 289.

Multicultural support for end-user USS messages and commands

Without multicultural support, USS messages (USSMSG00–USSMSG14) and commands (LOGON, LOGOFF, and IBMTEST) came from either a USS table that was statically defined to a logical unit, or the IBM-supplied default table, ISTINCDT. With multicultural support, users can choose a USS table by specifying the LANGTAB operand on any valid USS command. This enables users to enter USS commands and receive USS messages in the language of their choice. The USS table associated with the logical unit using the LANGTAB operand remains in effect until another valid USS command with the LANGTAB operand is processed.

You can also define USS messages using the MVS message service (MMS), which enables you to define each USS message in several different languages. Users can specify which language to use to retrieve USS messages from the MMS by using the LANG parameter on any valid USS command. The language for USS messages associated to the logical unit using the LANG parameter remains in effect until another valid USS command with the LANG parameter is processed.

For example, USSMSG10 can be defined in multiple languages, using the LANGTAB USS tables or MMS.

Remember that the language specified for the user USS commands (from the LANGTAB operand) and messages (from the LANG or LANGTAB operands) remain in effect for a logical unit (terminal) after the user logs off an application program. One logical unit (terminal) might be used by several users who speak different languages. Because LANG and LANGTAB can be entered with null values to cancel their effects, it might be wise to define LOGON commands with default null values for LANG and LANGTAB. For example, you can code the following as a default for the LOGON command:

```
LOGON    USSCMD   CMD=LOGON,FORMAT=PL1
         USSPARM  PARM=APPLID
         USSPARM  PARM=LANG,DEFAULT='()'
         USSPARM  PARM=LOGMODE
         USSPARM  PARM=DATA
         EJECT
```

Note: The coding for LANGTAB is the same as shown in the example for LANG.

Defining USS tables for use with the LANGTAB operand

A character-coded command that follows the USS command syntax rules (see the *z/OS Communications Server: SNA Resource Definition Reference*) is referred to as a USS command. A session-level USS table can be used to convert the USS command into a field-formatted SNA request. If a character-coded command violates these syntax rules, an interpret table must be used for the conversion.

The following restrictions apply when defining a USS table used for LANGTAB:

- The name of the USS tables you code must match the name the user specifies on the LANGTAB operand.
- The name of the USS table cannot be ISTINCNO or ISTCFCMM.
- FORMAT=DYNAMIC must be coded on the USSTAB macro to use the table for LANGTAB.
- If TABLE is coded on the USSTAB macro, it is ignored because the input translation table cannot be specified by the user.
- USS message text must comply with the rules detailed in the *z/OS Communications Server: SNA Resource Definition Reference*.

Defining USS messages to the MVS message service

To define USS messages to MMS:

1. Create message skeleton files for MMS. These files define the user messages (USSMSG00 – USSMSG14) in the languages required for your installation.
2. Define data sets to MVS.
3. Run the message compiler.
4. Update the parmlib.
5. Activate MMS.

For more information on message translation using MMS, see Appendix G, "Message translation using the MVS Message Service," on page 769. For information on performing these functions, refer to *z/OS MVS Planning: Operations*. For information on the format of message skeleton files used for language translation, refer to the *z/OS MVS Programming: Assembler Services Guide*.

When defining a USS message in a message skeleton file, use the substitute tokens &1., &2., and &3. for the first, second, and third variable data fields for the message. These tokens correspond to the %(1), %(2), and %(3) syntax of specifying variable data that is used for the TEXT operand of the USSMSG macro instruction.

Multicultural support for the language passed to application programs

By default, the language passed to a VTAM application program comes from the logon mode entry specified by the USS LOGMODE parameter. With multicultural support, the language passed to the application program can be specified by the user, independent of the LOGMODE parameter.

The user can enter a LOGON command to indicate which language is to be passed to the application program using:

- The language code parameter of the LANGTAB operand
- The LANG operand

Notes:

1. The language that is passed to the application program does not remain in effect across logons (as it does for USS messages). Each LOGON command must specify the LANG operand or the language code parameter of the LANGTAB operand to have a language passed to the application; otherwise, the default (MODEENT LANG operand value) is used.
2. If both operands indicate a language, the LANG operand has priority.
3. If the MMS is not currently active, the LANG operand is ignored.

For more information about the LANG and LANGTAB operands, refer to the *z/OS Communications Server: SNA Resource Definition Reference*. For specific information about how the application program receives the specified language, refer to *z/OS Communications Server: SNA Programming*.

Model name table

The model name table contains model names that can be passed to VTAM application programs in their LOGON exits. The purpose is to help VTAM application programs create dynamic definitions for their session-partner resources.

A model definition is normally used as the starting point for a dynamic resource definition. A typical application might maintain many different model definitions, one for each possible combination of resource characteristics. The model name passed by VTAM enables the application program to simply and accurately select the proper model definition to use.

Operands on an SLU resource definition can associate that SLU with the proper model name data. The MDLTAB operand specifies the model name table to be used, and the MDLENT operand specifies the proper entry within the table.

You create a model name table by specifying the following:
- A MDLTAB macro instruction
- One or more MDLENT macro instructions
- Optional MDLPLU macro instructions for each MDLENT

File these macros in the VTAM definition library. The name of the table is the name of the stored member or file. No assembly or link-edit is required to install the table. The table is built dynamically upon the first activation of any resource that has a defined association to the table.

IBM does not supply a default model name table.

An example of a model name table is shown in Table 13.

Table 13. Example of model name table

Name	Operation	Operands
MDLTAB1	MDLTAB	
ENTRY1	MDLENT	MODEL=MDLNAME1
ENTRY2	MDLENT	
	MDLPLU	PLU=APPL2, MODEL=MDLNAME2
	MDLPLU	PLU=APPL5, MODEL=MDLNAME3
ENTRY3	MDLENT	MODEL=MDLNAME3

Table 13. Example of model name table (continued)

Name	Operation	Operands
	MDLPLU	PLU=APPL5, MODEL=MDLNAME4
	MDLPLU	PLU=APPL7

Following are some examples of how the model name table in Table 13 on page 290 works:

- Assume that the SLU has the MDLTAB=MDLTAB1 macro coded in its definition. The entry name specified by the SLU and the name of the PLU involved in the session determines which model name is sent to the PLU during session initiation.
- Assume the SLU has the MDLENT=ENTRY1 macro coded in its definition. This macro specifies a default model name of MDLNAME1 for the entry. Because no MDLPLU macros are specified, model name MDLNAME1 is sent for any PLU.
- Assume the SLU has the MDLENT=ENTRY2 macro coded in its definition. This macro specifies no default model name. The first MDLPLU macro specifies that model name MDLNAME2 is sent when the PLU is APPL2. The second MDLPLU macro specifies that model name MDLNAME3 is sent when the PLU is APPL5. For any other PLU, no model name is sent.
- Assume the SLU has the MDLENT=ENTRY3 macro coded in its definition. This macro specifies a default model name of MDLNAME3 for the entry. The first MDLPLU macro specifies that model name MDLNAME4 is sent when the PLU is APPL5. The second MDLPLU macro specifies that no model name is sent when the PLU is APPL7. For any other PLU, model name MDLNAME3 is sent.

Associated LU table

An associated LU table contains associated LU names that can be passed to VTAM application programs in their LOGON exits. The purpose is to help VTAM application programs create dynamic definitions for their session-partner resources.

The associated LU names provide supplementary information unavailable from model definitions. The LU names specify primary and alternate printers logically related to the SLU. (For information on model definitions, see “Model name table” on page 290.)

Operands on an SLU resource definition can associate that SLU with the proper associated LU data. The ASLTAB operand specifies the associated LU table to be used, and the ASLENT operand specifies the proper entry within the table.

You create an associated LU table by specifying the following:
- An ASLTAB macro instruction
- One or more ASLENT macro instructions
- Optional ASLPLU macro instructions for each ASLENT

File these macros in the VTAM definition library. The name of the table is the name of the stored member or file. No assembly or link-edit is required to install the table. The table is dynamically built upon the first activation of any resource that has a defined association to the table.

IBM does not supply a default associated LU table.

Note: A printer must be in the same network as the terminal or must have a unique name that requires no translation across network boundaries. Violation of this restriction can cause failure of print-screen operations during cross-network sessions.

An example of an associated LU table is shown in Table 14.

Table 14. Example of associated LU table

Name	Operation	Operands
ASLTAB1	ASLTAB	
ENTRY1	ASLENT	PRINTER1=PTRA,PRINTER2=PTRB
ENTRY2	ASLENT	
	ASLPLU	PLU=APPL2, PRINTER1=PTRC,PRINTER2=PTRD
	ASLPLU	PLU=APPL5,PRINTER1=PTRA
ENTRY3	ASLENT	PRINTER1=PTRA
	ASLPLU	PLU=APPL2, PRINTER1=PTRC,PRINTER2=PTRD
	ASLPLU	PLU=APPL7,PRINTER2=PTRB

Assume that the SLU has the ASLTAB=ASLTAB1 macro coded in its definition. The entry name specified by the SLU and the name of the PLU involved in the session determines which printer names are sent to the PLU during session initiation. For example:

- Assume the SLU has the ASLENT=ENTRY1 macro coded in its definition. This macro specifies a default primary printer name of PTRA and a default alternate printer name of PTRB for the entry. Because no ASLPLU macros are specified, primary printer name PTRA and alternate printer name PTRB are sent for any PLU.
- Assume the SLU has the ASLENT=ENTRY2 macro coded in its definition. This macro specifies no default printer names for the entry. The first ASLPLU macro specifies that primary printer name PTRC and alternate printer name PTRD are sent when the PLU is APPL2. The second ASLPLU macro specifies that primary printer name PTRA is sent and no alternate printer name is sent when the PLU is APPL5. For any other PLU, no printer names are sent.
- Assume the SLU has the ASLENT=ENTRY3 macro coded in its definition. This macro specifies a default primary printer name of PTRA for the entry. The first ASLPLU macro specifies that primary printer name PTRC and alternate printer name PTRD are sent when the PLU is APPL2. The second ASLPLU macro specifies that no primary printer name is sent and alternate printer name PTRB is sent when the PLU is APPL7. For any other PLU, primary printer name PTRA is sent and no alternate printer name is sent.

Selecting session parameters for the logon mode table

When a logical unit requests a session with an application program, it uses a symbolic logon mode name, either directly or by default, to suggest the session protocols. Session protocols are a set of rules that describe how the session is conducted. For example, one protocol might specify the application program uses chaining for SNA requests. Another might require the logical unit does not send

end-bracket indicators to the application program. For a complete description of the session protocols available in VTAM, refer to *z/OS Communications Server: SNA Programming*.

Session protocols are expressed as a string of characters called session parameters, usually specified in the logon mode table. A set of session parameters for a session is also called its logon mode. VTAM contains an IBM-supplied logon mode table, ISTINCLM. This table contains a set of generally accepted session parameters for a basic list of IBM device types, but it might not completely meet your needs. You can create supplementary logon mode tables and associate them with device type logical units or application programs using the MODETAB operands on the definition statements defining them.

To create the logon mode table:

- Modify the IBM-supplied table.
- Create supplementary tables and associate them with device-type LUs or application programs using the MODETAB operands in the definition statements defining them. You can use the DLOGMOD operand on the appropriate definition statements to indicate the specific logon mode entry in the table that contains the session parameters to be used by default. The LOGMODE operand can also be specified on VARY LOGON commands to specify a logon mode table entry.
- Associate a supplementary table with dynamic CDRSCs using the DYNMODTB start option. You can also use the DYNDLGMD start option to indicate the default logon mode entry for dynamic CDRSCs. The value for DYNMODTB can be changed using the DYNMODTB operand on the MODIFY VTAMOPTS command, or by using the MODIFY TABLE,OPTION=LOAD command to replace the existing table that is being used by dynamic CDRSCs. You can change the default logon mode entry for dynamic CDRSCs using the DYNDLGMD operand on the MODIFY VTAMOPTS command. You can also override this value for a specific dynamic CDRSC using the MODIFY DEFAULTS or MODIFY RESOURCE commands.

  When VTAM builds a dynamic CDRSC, it uses the DYNMODTB value (if specified; otherwise ISTINCLM) to associate a logon mode table to the dynamic CDRSC. It also uses the DYNLOGMD value, if specified, to assign a default logon mode table entry to the dynamic CDRSC. Changing the value of these start options does not change the value for dynamic CDRSCs that have already been built.

Logon mode names are always resolved using the logon mode table associated with the secondary logical unit (SLU). For more information on which node performs the mode resolution, see "Resolving logon mode names to subarea and APPN Classes of Service" on page 525.

Logon mode tables are created or modified using MODETAB, MODEENT, and MODEEND macro instructions. The *z/OS Communications Server: SNA Resource Definition Reference* describes the contents of the IBM-supplied logon mode table. It also describes how TSO can query for the logon mode being used by a terminal in place of supplying the LOGMODE during the logon. Figure 70 on page 294 shows the sequence in which MODETAB, MODEENT, and MODEEND macros are coded to define a logon mode table.

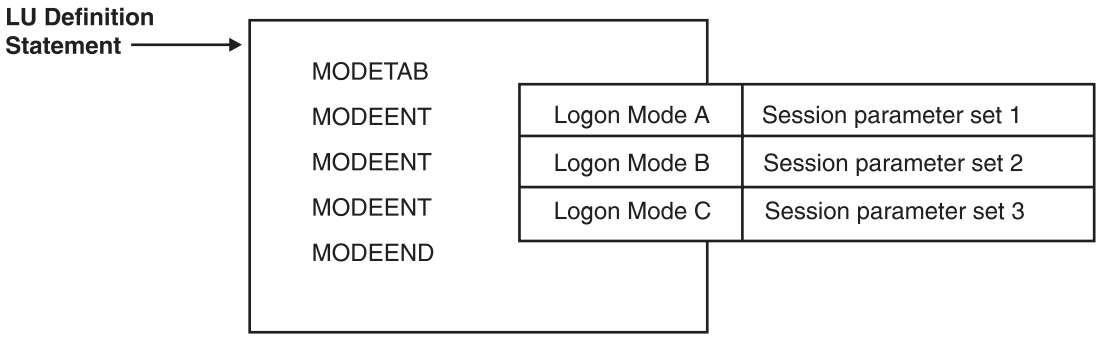


Figure 70. Macroinstructions for logon mode table

The various ways of specifying session parameters are illustrated in Table 15.

Table 15. How session parameters are identified

Request source	**Request form**	**How session parameters are identified**	
		Logon mode table entry	**Logon mode table**
Dependent Logical Unit	• Initiated Self request • Interpret table character string representing logon • Character-coded logon	• Field in Initiate Self request containing mode name used to search logon mode table • Parameter in a character-coded logon that translates into LOGMODE (name) • DLOGMOD operand on LU definition statement • Default entry ISTCOSDF, if the ISTCOSDF start option allows its use for dependent LUs • First entry in logon mode table named in MODETAB operand on LU definition statement • First entry in IBM-supplied logon mode table	• MODETAB operand on LU definition statement • IBM-supplied logon mode table

Table 15. How session parameters are identified (continued)

Request source	Request form	How session parameters are identified	
		Logon mode table entry	**Logon mode table**
Independent Logical Unit	• BIND • Cross-domain initiate	• Field in BIND or cross-domain initiate request containing mode name used to search logon mode table • DLOGMOD operand on CDRSC definition statement, or DYNDLGMD start option value (for dynamic CDRSCs) • Default entry ISTCOSDF, if the ISTCOSDF start option allows its use for independent LUs • First entry in logon mode table named in MODETAB operand on CDRSC definition statement or DYNMODTB start option value (for dynamic CDRSCs) • First entry in IBM-supplied logon mode table	• MODETAB operand on CDRSC definition statement or DYNMODTB start option value (for dynamic CDRSCs) • IBM-supplied logon mode table

Table 15. How session parameters are identified (continued)

Request source	Request form	How session parameters are identified	
		Logon mode table entry	**Logon mode table**
Application Program	Application program issuing VTAM macros, such as SIMLOGON or REQSESS	• Name of logon mode table entry specified in the NIB • DLOGMOD operand on LU definition statement • Default entry ISTCOSDF, if the ISTCOSDF start option allows its use for application programs • First entry in logon mode table named in MODETAB operand on LU definition statement • First entry in IBM-supplied logon mode table	
Automatic logon	Initiate Other request	• DLOGMOD operand on LU or CDRSC definition statement or DYNDLGMD start option value for dynamic CDRSCs • Default entry ISTCOSDF, if the ISTCOSDF start option allows its use • First entry in logon mode table named in MODETAB operand on LU or CDRSC definition statement, or DYNMODTB start option value for dynamic CDRSC • First entry in IBM-supplied logon mode table	

Table 15. How session parameters are identified (continued)

Request source	Request form	How session parameters are identified	
		Logon mode table entry	**Logon mode table**
VTAM operator	VARY LOGON command	• LOGMODE operand in VARY LOGON command • LOGMODE operand in previous VARY LOGON command • DLOGMOD operand on LU or CDRSC definition statement, or DYNDLGMD start option value for dynamic CDRSCs • Default entry ISTCOSDF, if the ISTCOSDF start option allows its use • First entry in logon mode table named in MODETAB operand on LU or CDRSC definition statement, or DYNDLGMD start option value for dynamic CDRSCs • First entry in IBM-supplied logon mode table	

Regardless of the source of the logon or what session parameters are associated with the logon, the application program decides which session parameters are used for the session. It can decide to use the session parameters associated with the pending logon, or it can choose a different set of parameters. When the application program issues a macro to initiate a session with a logical unit, the application program indicates the logon mode name or session parameters to be used during the session. The logon mode table from which the session parameters are selected is always the one associated with the SLU, or, if a table is not associated with the logical unit, it is the IBM-supplied default logon mode table in the VTAM that owns the SLU.

If the specified logon mode entry does not exist in the logon mode table associated with the secondary logical unit, the default logon mode (ISTCOSDF) is used if it is found in the table and its use is allowed as determined by the ISTCOSDF start option.

You can modify or replace the IBM-supplied logon mode table, provided that the modified or replacement table has the same name as the IBM-supplied table and that the IBM-supplied table is deleted. However, because the IBM-supplied table

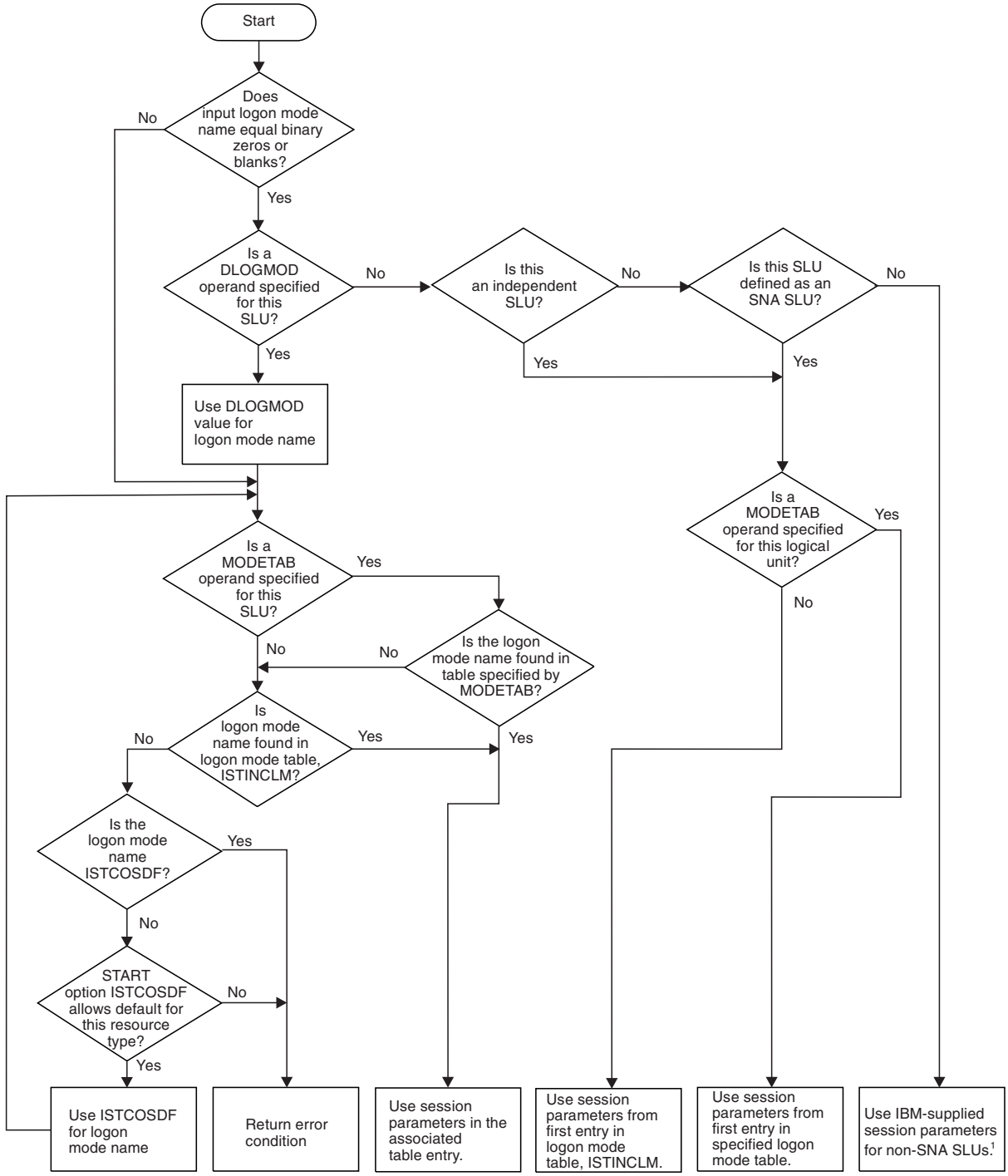
might be needed for problem determination, you should create supplementary tables instead of deleting the IBM-supplied table.

As part of the session-establishment procedures, an application program acting as the PLU can:

- Supply a logon mode name associated with the SLU. The session parameters associated with the logon mode name in the domain of the SLU are used.
- Directly supply the session parameters to be used.
- Use the session parameters associated with the pending logon.

z/OS Communications Server: SNA Programming describes in more detail how the application program handles session parameters.

Figure 71 on page 299 shows a summary of the algorithm VTAM uses to obtain session parameters.



[1] VTAM uses the following session parameters for the non-SNA SLU:
MODEENT LOGMODE=NONSNA, FMPROF=X'02', TSPROF=X'02', PRIPROT=X'71', SECPROT=X'40', COMPROT=X'2000'

Figure 71. How session parameters are obtained from a logon mode table

Automatic logons

You can enable your LUs to automatically log on to a particular application program. This is most useful when a designated terminal always accesses one application program. To enable automatic logons, you can either code the LOGAPPL operand on the LU definition statement, or you can use the VARY LOGON or VARY ACT,LOGON commands.

Automatic logon is useful for the following:

- Unattended terminals (terminals with no operator to issue a logon)
- Establishing a controlling application program to:
  - Manage allocation of terminals
  - Automatically reestablish sessions with terminals after failure and restart of a host or NCP

Coding for automatic logon

By using the LOGAPPL operand on the definition statement for an LU, the LU can automatically initiate a session with a specific application program whenever the LU is activated. The application program can either be the program with which the LU ultimately initiates a session or an intermediary application program to which the LU is temporarily assigned. For application programs in another network, the network-qualified name can be specified on the LOGAPPL operand.

The application program to which an LU is automatically logged on is called its controlling application program. If the LU is passed to another application program, VTAM automatically initiates a session with the controlling application program when that session is terminated. If the application program with which the LU is in session terminates that session and there is no controlling application program, the LU is available for session initiation.

Figure 72 shows a sample automatic logon session:

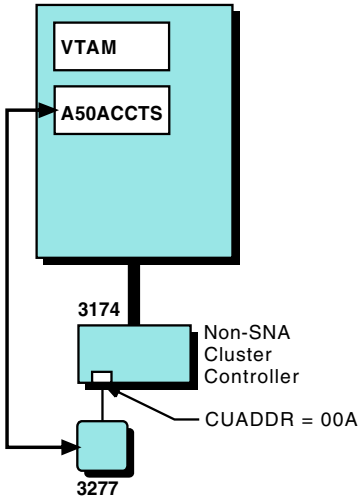


Figure 72. Automatic logon to A50ACCTS application program

Following is the sample coding for the A50ACCTS application program in Figure 72:

```
APPLNODE VBUILD TYPE=APPL
A50ACCTS APPL   ATNLOSS=ALL,      ENABLE ATTN EXIT FOR ALL LU 6.2 SESSIONS
                APPC=YES,         ENABLE LU 6.2 SUPPORT
                PARSESS=YES,      ENABLE PARALLEL SESSION FOR LU 6.2 SUPPORT
                SECACPT=ALREADYV, ACCEPT ALREADY VERIFIED LU 6.2 CONVERSATIONS
                ACBNAME=ACCOUNTS  NETWORK LU NAME=A50ACCTS
                                  VTAM APPLICATION APPLID=ACCOUNTS
```

⋮

Following is the sample coding for the 3277 display station in Figure 72 on page 300:

```
LC3270NS LBUILD                     LOCAL NON-SNA ATTACHMENT
LOC3277  LOCAL  CUADDR=00A,         CHANNEL UNIT ADDRESS
                TERM=3277,          TERMINAL TYPE

⋮
                DLOGMOD=S3270,      DEFAULT LOGON MODE TABLE ENTRY
                LOGAPPL=A50ACCTS    AUTOMATIC LOGON APPLICATION
```

Operator commands for automatic logon

You can establish an automatic logon session using the VARY LOGON command. After the automatic logon is established, it remains until one of the following occurs:

- The major node of one of the session partners is deactivated.
- The primary logical unit is changed using the VARY LOGON or VARY ACT,LOGON command. If the primary logical unit is a USERVAR name, the session setup requests can be rerouted to a different primary logical unit by using the MODIFY USERVAR command.
- The automatic logon is deleted using the VARY NOLOGON command.

Tips:

1. You can view the automatic logon relationship of an LU with the DISPLAY ID=*sluname* command.
2. You can view automatic logon relationships by controlling application name using the DISPLAY AUTOLOG,ID=*controlling_application name* command. This display shows only LUs that are session capable and not currently in session with their controlling application. This display also shows the events that can recover the controlling sessions.
3. The VARY AUTOLOG command can be used to attempt recovery of the controlling sessions.
4. Refer to *z/OS Communications Server: SNA Operation* for more details on using the DISPLAY AUTOLOG and VARY AUTOLOG operator commands.

Reallocation of autologon sessions

For dependent LUs and TN3270 applications using DEFAPPL, the VTAM that owns the secondary logical unit controls the automatic logon sessions. For independent LUs, the VTAM providing boundary function services for the secondary logical unit controls the automatic logon sessions. VTAM also provides autologon support over a CP-CP session for secondary LUs whose network node server does not have autologon capabilities.

By default, adjacent CDRM activation redrives any autologon sessions that are not established between resources of the two VTAMs. Adjacent CP activation also redrives any such autologon sessions, if the adjacent CP supports automatic logon.

To specify which, if any, adjacent node activations are to result in trying pending autologon requests again, use the AUTORTRY start option. In addition to the default for AUTORTRY described above, you can use AUTORTRY to limit session retries to only after adjacent CDRM activation (AUTORTRY=CDRM), to suppress all retries for adjacent node activations (AUTORTRY=NONE), or to specify that all adjacent node activations are to result in session retry (AUTORTRY=ALL). For AUTORTRY values other than NONE, CDRM deactivation causes redrives of any autologon sessions that relied on the lost CDRM. CP deactivation does not cause redrives.

Use the AUTOTI start option to control how often pending autologon requests are retried. Time values are specified in seconds, minutes, hours, or days. If AUTOTI is nonzero, redrives occur at timer expiration for all autologon sessions that are not established, regardless of the value of AUTORTRY. Use AUTOTI to ensure session setup in the following situations:

- Multihop CP-CP network
- Network node server that has not yet received the controlling primary LU registration
- CLSDST PASS that was sent to an unavailable application
- USERVAR value that changed while a node is inactive, and the node is then reactivated
- Controlling PLU that was a clone application that had not yet opened its ACB
- Controlling PLU was a clone application that has moved to a new host (for example, CLOSE ACB on HOST A and then OPEN ACB on Host B)

Notes:

1. When using search reduction support (SRCHRED=YES, and SRTIMER, SRCOUNT, or both are nonzero) in conjunction with a nonzero value for AUTOTI, remember that the two functions are designed to produce opposite effects. Search reduction is used to limit searches for resources that have been found to be unreachable. AUTOTI is used to intermittently search the network at a specified interval for a previously unreachable controlling PLU. After the first search for a controlling PLU fails, attempts are made to redrive unestablished autologon sessions whenever AUTOTI expires. However, if AUTOTI is less than SRTIMER, these requests for the PLU are limited by the search reduction entry until it also expires. If AUTOTI=x and SRCOUNT=z, such requests for the PLU are limited for (x*z) seconds. For information on search reduction, see “Improving VTAM performance using start options” on page 31.
2. The VARY AUTOLOG operator command can be used to manually reallocate autologon sessions.

Sample automatic logon reallocation

In Figure 73 on page 303, assume that secondary logical unit LU1A is in session with an application program APPL1, and that LU1A has LOGAPPL=A50ACCTS coded. When the session with APPL1 ends, VTAM attempts to establish a session between LU1A and A50ACCTS.

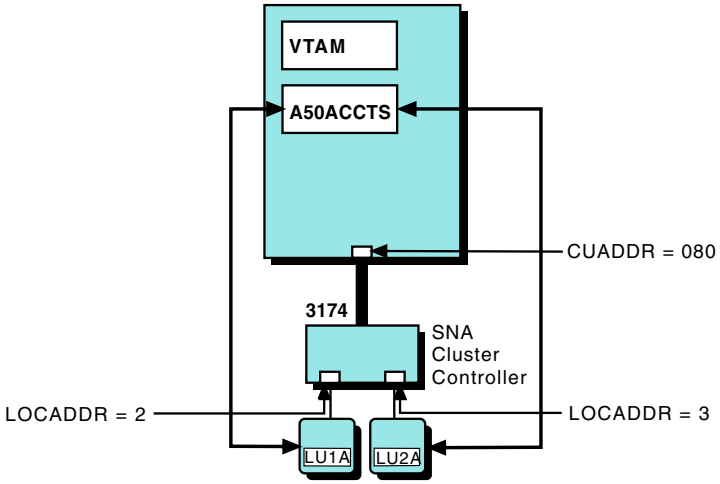


Figure 73. Automatic logon reallocation

If A50ACCTS is inactive, the automatic logon session fails. If the A50ACCTS major node is active to VTAM but it has not opened its ACB yet, the automatic logon session fails, but is initiated when A50ACCTS opens its ACB.

If you use the VARY INACT command to deactivate A50ACCTS, you can only regain the automatic logon sessions with LU1A and LU2A by issuing a VARY ACT,LOGON command for A50ACCTS or reactivating A50ACCTs and then reactivating LU1A and LU2A.

Conditions for reallocation

A session is reallocated as a result of one of the following conditions:

- If a secondary logical unit goes through a state change from disabled or inhibited to enabled, the owning VTAM attempts to set up a session with the controlling application program for the secondary logical unit. The session state is determined by the device and is indicated to VTAM by an ACTLU response from the logical unit or a NOTIFY request from the device.
- If a session is terminated, VTAM checks to see whether an automatic logon relationship exists and, if so, reallocates that session if the session being ended is not with the controlling application program.
- When an active session between a secondary logical unit and its controlling application program terminates, the VTAM that owns the secondary logical unit decides whether to reallocate the controlling session based on the type of termination request or UNBIND that is received. The controlling session is not reallocated if the termination request or UNBIND type indicates normal termination. If the session was terminated using the VARY TERM command with either TYPE=UNCOND or TYPE=FORCE, the controlling session is reallocated if a session path exists at that time between the two logical units. To avoid redriving the session, issue VARY TERM TYPE=COND instead. For further information, refer to *z/OS Communications Server: SNA Operation*.

How primary logical unit status affects reallocation

VTAM attempts to start a session with the controlling application program (the primary logical unit). Following is an explanation of how VTAM reacts to the primary logical unit status:

- If the owning host of the secondary logical unit cannot find the primary logical unit (the major node is not active) or is inactive, the automatic logon session

fails. VTAM does not automatically reallocate the failing session to the primary logical unit; instead, the owning host of the secondary logical unit creates a pending autologon request. VTAM retries pending autologon requests periodically based on the settings of the AUTOTI and AUTORTRY start options.

- If the primary logical unit is in CONCT status (active to VTAM but OPEN ACB has not been issued), the automatic logon fails with sense code 088A0003, and VTAM tries to set up the session again when the primary logical unit issues OPEN ACB and SETLOGON OPTCD=START macroinstructions.
- If the primary logical unit is deactivated through VTAM by an operator command and becomes inactive, the secondary logical unit must reattempt the automatic logon session using a secondary logical unit state change, or the operator can issue a VARY ACT command for the secondary logical unit after the primary logical unit is at least in CONCT status.

  Note: If an SLU is in session with its controlling application at the time of the PLU deactivation, VTAM retries the automatic logon session. If VTAM retries to establish a session and fails, a pending autologon request is created in the owning host of the SLU. VTAM retries pending autologon requests periodically based on the settings of the AUTOTI and AUTORTRY start options.

Determining automatic logon relationships

To determine an automatic logon relationship, use the DISPLAY ID=(*SLUname*),E command. Use this command to determine whether a logical unit or TN3270 application has a controlling primary logical unit. Message IST1131I contains the name of the primary logical unit. Message IST635I contains the name of the session partner, the status of the session, and other session information. The status of the session will have the last two characters overlaid with */C* if this session is due to an automatic logon relationship.

The DISPLAY AUTOLOG command displays controlling applications for which there are one or more LUs in this host that are session capable but that are not currently in session with their controlling application. The command can also be used to list SLUs that are in this state. The PLU name is displayed in messages that include the events that automatically attempt to re-establish sessions between the controlling application and the SLUs that have pending automatic logon requests.

Sample displays received during reallocation

If the primary logical unit is in CONCT status when the automatic logon session setup fails, you will receive the following messages:

```
IST663I INIT OTHER OR CDINIT REQUEST FAILED, SENSE=088A0003
IST664I REAL OLU= <SLU-NAME>  REAL DLU= <PLU-NAME>
IST889I SID= <SESSION ID>
IST890I AUTOLOGON SESSION SETUP FAILED
IST896I AUTOLOGON WILL BE RETRIED WHEN CONTROLLING PLU IS AVAILABLE
IST314I END
```

Note: Besides normal message suppression, the number of IST663I and IST664I messages for the 088A0003 sense codes can be suppressed using the PLUALMSG and SLUALMSG start options. For more information, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

If the primary logical unit was in CONCT status when the automatic logon failed and then issues an OPEN ACB, you will receive the following messages:

```
IST899I RETRY OF AUTOLOGON(S) TO <PLU-NAME> IN PROGRESS
```

If the primary logical unit is not active when the automatic logon fails, you will receive the following messages:

```
IST663I INIT OTHER OR CDINIT REQUEST FAILED, SENSE=08570002
IST664I REAL OLU= <SLU-NAME>  REAL DLU= <PLU-NAME>
IST889I SID=.....................
IST890I AUTOLOGON SESSION SETUP FAILED
IST1138I REQUIRED RESOURCE <PLU-NAME> NOT ACTIVE
IST314I END
```

Note: The sense code in message IST663I, and the text of message IST1138I might be different.

If you had automatic logon sessions fail on a primary logical unit that was in CONCT status and then you deactivate it before it opens its ACB, you will receive the following messages:

```
IST097I VARY ACCEPTED
IST899I RETRY OF AUTOLOGON(S) TO <PLU-NAME> WILL NOT OCCUR
IST1133I <PLU-NAME> NODE NOW INACTIVE
```

If the VARY AUTOLOG command is used to attempt reallocation of automatic logon sessions, you will receive the following messages:

```
IST097I VARY AUTOLOG ACCEPTED
IST2093I AUTOLOGON SEARCH INITIATED FOR <NUMBER> APPLICATIONS
IST314I END
```

At this point, searches are generated to find and determine the status of one or more controlling applications. After the search for each controlling application completes, the following messages are issued if sessions cannot be established with the controlling application.

- If the search is unable to locate the controlling application, the following messages are issued:

  ```
  IST2096I AUTOLOGON SEARCH COMPLETED FOR APPLICATION <PLU-NAME>
  IST2097I STATUS - CANNOT BE LOCATED
  IST314I END
  ```

- If the search locates the controlling application and finds the status of the application is not session capable, the following messages are issued:

  ```
  IST2096I AUTOLOGON SEARCH COMPLETED FOR APPLICATION <PLU-NAME>
  IST2098I STATUS - UNABLE TO ACCEPT LOGONS
  IST314I END
  ```

- If the search finds the controlling application capable of accepting sessions, the following messages are issued:

  ```
  IST2096I AUTOLOGON SEARCH COMPLETED FOR APPLICATION <PLU-NAME>
  IST2099I STATUS - AUTOLOGON SESSIONS INITIATED FOR <NUMBER> LUS
  IST314I END
  ```

Session management exits

The session management exit routine can be used to:

- Check or restrict the use of a logical unit (session authorization)
- Gather accounting information (session accounting)
- Select adjacent SSCP for LU-LU session and resource status requests
- Process a session takeover by the alternate XRF application program

Code a session management exit routine if you want to use these functions. There is no default session management exit routine. If you do not code a session management exit routine, it is assumed that all session requests are authorized and no accounting data is collected by VTAM.

VTAM calls the session management exit routine whenever a session between logical units is established, ended, taken over, or recovered. The session management exit routine is invoked by every VTAM in the session setup path. It is also called for session accounting when an LU-LU session ends.

Note: Improper use of the session management exit routine can impact VTAM operation. For example, you should consider the following:

- A program error can affect VTAM, the operating system, and other programs. Although abnormal end recovery is provided for errors within the exit, the exit code can accidentally corrupt VTAM storage.
- Lengthy processing time could degrade VTAM performance. While these routines are running, VTAM cannot process any new VTAM operator requests to initiate or end sessions. To reduce the effect on VTAM performance, any analysis of the information collected should be done later by another program.

In a multiple domain environment, the session management exit routine can be invoked in the domains in which each of the session partners resides and can be used to select adjacent SSCPs for LU-LU session and resource status requests.

The adjacent SSCP selection function in the session management exit routine can be used to control the SSCP to which a session request is sent. The exit allows you to shorten or reorder the list of adjacent SSCPs, or to not route at all. You can use this function with the dynamic adjacent SSCP table facility so that you can avoid defining adjacent SSCP tables while maintaining control of session request routing.

VTAM supports session authorization exit routines, which can be used to control session initiation. These routines can accept or reject the session based on various information that is passed to them. VTAM also supports session accounting exit routines that can be used to perform accounting functions for established sessions.

The session authorization and session accounting exit routines provide some of the same functions as the session management exit routine. These exits are invoked prior to the corresponding session management function. However, these exit routines do not provide all of the functions that are available in the session management exit routine, and they are not recommended.

For further information on the session management, session authorization, and session accounting exit routines, including a detailed description of the vector lists and the information provided to the routines, refer to *z/OS Communications Server: SNA Customization*.

Session authorization

When SSCP routing is in process for cross-domain sessions, full information about the destination logical unit might not be available when the session management exit routine is first invoked. To enable proper authorization to be done for cross-domain sessions, two separate calls for authorization can be made for these sessions. The first call is made early in the session setup process. The session management exit routine can accept or reject the session at this time, or it can defer its decision until later. If the decision is deferred, the routine is called again for a secondary authorization after full information is known about the destination logical unit.

You might code the routine to contain a table of valid sessions against which the session-establishment request can be compared. For example, an application

program can be designed to establish a session with any logical unit using the OPNDST macro instructions with the ACCEPT option in its LOGON exit routine. The authorization exit routine can compare the identity of any logical unit that attempts to establish a session with the application program to entries in such a table to determine whether authorization can be granted for that logical unit.

Both primary and secondary authorization exits are also driven during LU takeover processing if the sessions being taken over were established using extended BIND protocols.

Session accounting

With the capability of peripheral LUs serving as primary LUs (PLUs), accounting application programs relying on the primary LU to be located only in the host might not be sufficient.

At LU-LU session setup time, a session awareness PIU is sent to the NetView Performance Monitor (NPM) as unsolicited information. Byte and PIU counts are updated every time there is traffic on the session. When the counts surpass the user-specified threshold, the counters are sent to NPM. At the end of the session, session awareness data and the last set of counters for the session are sent to NPM. The user-specified options, as well as thresholds, can be changed using NPM commands. In addition, transport of accounting data to NPM can be suspended and resumed using NPM commands.

If selecting session accounting, the user can specify the following during NCP generation:

- Whether session accounting is required for PLUs, SLUs, or all LUs
- Whether accounting collection should begin immediately or be deferred
- The session accounting byte and PIU thresholds
- Whether backup NPM sessions are defined
- Whether gateway session accounting is to be used for cross-network sessions
- The number of half-sessions for which session accounting will be done
- The number of session accounting extension blocks to define
- The PIU distribution ranges (if any) to be used

Especially important to a multiple-network environment is the ability of the user to specify the following during NCP generation:

- Whether gateway session accounting is to be used for cross-network sessions
- The number of half-sessions for which session accounting is done

Session-level pacing

Session pacing can be used to influence performance between session partners. Using high-pacing window sizes or no-session pacing for interactive sessions and using lower-session pacing window sizes for batch sessions is a widely used approach to begin tuning an SNA network. For example, a VTAM application driving a low-speed printer needs to use small session pacing window sizes to keep one session partner from overwhelming the other with data. Using higher (or no) pacing window sizes for interactive sessions than for batch sessions favors the interactive users in the network containing both batch and interactive traffic. For batch, if the session partners and network can support it, higher-pacing window sizes can be used to improve throughput.

Session-level pacing involves each session-layer component in a session path. Session pacing occurs in session stages. Each stage is independently paced, allowing different pacing window sizes to be used for each stage.

There are two types of session pacing:

Fixed With fixed session-level pacing, the pacing window size is constant. The amount of data that can be transmitted or received between session partners is maintained as a constant window size.

Adaptive
With adaptive session-level pacing, the window size can vary depending on the resources available at each session endpoint. After each window of data is transmitted, the receiver informs the transmitter how large the next window can be.

Adaptive pacing will be used whenever possible. A session path between two LUs may have a mixture of pacing types along the route.

Refer to the *SNA Technical Overview* for additional information on session-level pacing concepts.

Fixed session-level pacing

The pacing window size between two endpoints of a pacing stage is constant when fixed session-level pacing is being used. The window size is passed in the BIND request during session activation. The window size in the BIND response is the actual window size used. The number of pacing stages can be the same for both directions (primary-secondary and secondary-primary) in a session, or it can be different. In Figure 74, APPL1 is in session with a type 2.0 node. Two-stage pacing occurs from APPL1 (primary logical unit) to the type 2.0 node (secondary logical unit), and one-stage pacing occurs from the secondary logical unit to APPL1. In this case, the NCP provides the boundary function for the LU.

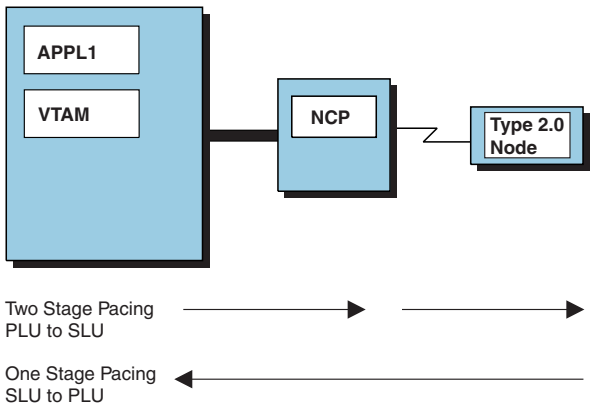


Figure 74. Fixed-session pacing (one- and two-stage)

For fixed session pacing, the pacing response is sent in either of two ways:
- In the response header of a message unit
- In a stand-alone message called an isolated pacing response (IPR)

Adaptive session-level pacing

Adaptive session pacing allows the pacing window size to dynamically:
- Increase to accommodate increased traffic

- Decrease when congestion is in the receiving node

The pacing window sizes are adapted to buffer availability and demand on a session-by-session basis. The session stage endpoints exchange explicit pacing windows that can vary in size during the course of a session. The mechanism thus prevents any single session from monopolizing or exhausting the buffer resources of a session stage endpoint.

Adaptive pacing occurs between each pacing stage endpoint and can vary for each direction. In Figure 75, APPL1 is in session with a type 2.1 node. Two-stage pacing occurs for data flowing in both directions between the session partners.

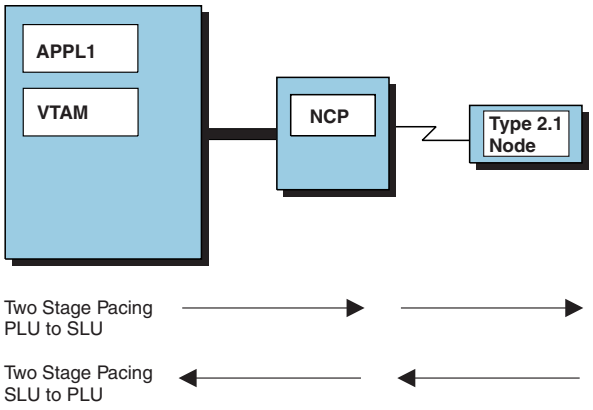


Figure 75. Adaptive session pacing

If adaptive pacing is performed, the value determined during session setup for the pacing window size represents the minimum pacing window size to be used during normal operation.

Note: Prior to VTAM Version 4 Release 4, VTAM did not dynamically vary the session partner send window size. VTAM always returns the same window size that was set during session establishment. However, VTAM honors the pacing windows sent by its pacing stage partners for adaptively paced sessions. The next window size used for data transmission is whatever value was received from the session partner in the pacing response.

An isolated pacing message (IPM) indicates whether the pacing window size should be changed. Pacing window sizes are changed by IPMs. The IPM, like the IPR used in fixed session pacing, is used to authorize the sending of the next window of messages units, but it also specifies whether the window size is to be changed or remain the same.

The following scenario describes how adaptive pacing works.

1. The transmitter requests a pacing response from the receiver when sending the first PIU in the current send window.
2. The receiver returns a pacing response that contains the next window size it considers appropriate for the next transmission. If the transmitter sends the entire window of PIUs before the pacing window response is received, the transmitter must wait until the pacing response is received.
3. The transmitter can request a larger window size by setting an indicator in the next transmission.

4. The receiver can ignore the transmitter request for a larger window or honor it when generating the pacing response.

   The transmitter is obligated not to exceed the window size specified by the receiver.

Note: You must have NCP Version 5 Release 2, or higher, to support two-stage and adaptive-session pacing involving a 3745 communication controller, or Version 4 Release 3, or higher, for a 3725 communication controller.

Setting initial pacing values

The type of session-level pacing and pacing window sizes are determined by the BIND request when a session is established. When an extended BIND is used, adaptive session-level pacing is requested. If adaptive-session pacing is not supported or desired by the session partner, the BIND response will be sent indicating fixed-session pacing. (XBSI is the extended BIND indicator in the session initiation request. XBSI=ON means that extended BIND is used to establish the session.) The actual values used are those in the BIND response.

The pacing values are carried in the BIND request flowing from the primary LU to the boundary function (if one exists for this session), and then to the secondary LU. In this way, all the participants in the session are informed of the pacing values to be used.

With this in mind, the possible pacing windows for a session are the following:

PS Primary send
SR Secondary receive
SS Secondary send
PR Primary receive

The direction of the data being sent determines which pacing window values are used.

- When data is sent from the primary LU to the secondary LU, the PS and SR values are used.

  – The primary send count controls the flow of requests from the primary logical unit to the boundary function,
  – The secondary receive count controls the flow from the boundary function to the secondary logical unit.
- When data is sent from the secondary LU to the primary LU, the SS and PR values are used.

  – The secondary send count controls the flow of requests from the secondary logical unit to the boundary function.

– The primary receive count controls the flow from the boundary function to the primary logical unit.

The values for the four pacing windows are derived from:

- The following operands on the MODEENT macro instruction for the logon mode entry:
  – PSNDPAC for primary send
  – SRCVPAC for secondary receive
  – SSNDPAC for secondary send
- The following operands on the LU or CDRSC definition statement:
  – PACING
  – VPACING
- The following operands on the APPL definition statement:
  – AUTH=NVPACE|VPACE
  – VPACING

Note: An application program acting as a primary end of a session can obtain information about pacing counts using the INQUIRE macro instruction or the LOGON exit routine. For more information, refer to *z/OS Communications Server: SNA Programming*.

The sources used to determine pacing-window values are dependent on the configuration and the secondary LU (SLU). Figure 76 on page 312 and Figure 77 on page 312 describe the environment, and Table 16 identifies the sources used in determining the initial pacing window value when the primary LU (PLU) is an application program. For example, if the SLU is an application program, the pacing sources, regardless of whether an extended BIND is sent, are described in Table 16 under:

- Letter I for primary to secondary flow
- Letter D for secondary to primary flow

Notes:

1. For fixed-session pacing, the pacing window values derived are the actual values used.
2. For adaptive-session pacing, the pacing window values derived represent the minimum pacing window size to be used.

Table 16. Correspondence of methods to letters

Letters	Methods used by VTAM
A	If SRCVPAC is not 0, use it **else** use SLU VPACING (default is 7)
B	If SRCVPAC is not 0, use it **else** use SLU PACING (default is 1)
C	Use SRCVPAC
D	If SSNDPAC is 0, use it **else** use PLU VPACING (default 7)
E	Use SSNDPAC
F	If PLU AUTH=NVPACE, use 0 **else** if PSNDPAC is not 0, use it **else** use SLU VPACING (default is 2)
G	If PLU AUTH=NVPACE, use X'5F' **else** if PSNDPAC is not 0, use it **else** use SLU VPACING (default 1)

Table 16. Correspondence of methods to letters (continued)

Letters	Methods used by VTAM
H	Use PSNDPAC
I	If PLU AUTH=NVPACE, use 0 **else** if SRCVPAC is not 0, use it **else** use SLU VPACING
J	Use PLU VPACING (default 7)

	If SNA SLU Is:															
	An Application				An NCP LU				A Same-Domain Local LU				A Cross-Domain Local LU			
And the Pacing Direction Is:	P→S		S→P		P→S		S→P		P→S		S→P		P→S		S→P	
And the Windows Are:	PS	SR	SS	PR	PS	SR	SS	PR	PS	SR	SS	PR	PS	SR	SS	PR
Windows Are Set Using Method:	I		D		F	B	D		A		D		G	B	E	J

Figure 76. Pacing windows for SNA LUs. Coding is from the perspective of the owning SSCP acting as the PLU.

	If Non-SNA SLU Is:							
	A Same-Domain LU or an NCP Terminal		A Cross-Domain 370 CA Terminal			A Cross-Domain Local LU		
And the Pacing Direction Is:	P→S	S→P	P→S		S→P	P→S	S→P	
And the Windows Are:	PS=SR	SS=PR	PS	SR	SS=PR	PS=SR	SS=PR	
Windows Are Set Using Method:	C	E	H	C	E	C	E	

Figure 77. Pacing windows for non-SNA LUs. Coding is from the perspective of the owning SSCP acting as the PLU.

Overriding fixed pacing counts

The pacing counts specified on the LU definition statement apply to all sessions in which the LU participates. Values stated in the APPL definition statement override the values in the logon mode table entry and the LU definition statement. When you specify AUTH=NVPACE, the value for the primary send that goes out in the BIND is 0. Therefore, there is no pacing between the application program and the boundary function.

AUTH=VPACE prevents overloading buffers in the communication controller with outbound messages from VTAM. You can specify NVPACE when the application program sends a limited amount of data and then waits for a user response.

The primary application program can override the primary receive value for a specific session by specifying a nonzero value in the primary receive field of the session parameters it specifies at OPNDST. If 0 is specified, the primary receive value present in the session parameters in the pending logon is used. Only values less than or equal to the value in the pending logon should be specified by the application program.

Note: NVPACE can also be specified when an application program is sending chains of RUs to a secondary logical unit and the application program cannot send more BIUs than can be held in the queue. The number of elements in each chain must be no larger than the results of the following formula, where *n* is the smallest value in the VPACING operands of the definition statements for the LUs:

number of elements ≤ $2n - 1$

If the application program is in session with LUs whose VPACING operands are different from each other, the smallest VPACING values should be used in the formula.

See "Setting initial pacing values" on page 310 for information on how VTAM determines the initial pacing-window size.

Independent logical units and adaptive-session pacing

Although some type 2.1 peripheral nodes do not support adaptive-session pacing, an independent logical unit in that node can still act as the primary LU and therefore sends the BIND. The primary pacing values in the BIND are established by the independent logical unit (primary). However, the value of the primary send count might be so large that it can impact the subarea node to which it is attached. For example, a PS/2 attached to an NCP using a token ring (NTRI) can specify a value of 0 (no pacing) as the primary send count, which if not altered, can cause the NCP to enter slowdown. VTAM, in conjunction with NCP, allows you to control the primary pacing values used by an independent logical unit for which adaptive-session pacing is not supported. If a type 2.1 peripheral node that supports adaptive session pacing is attached directly to VTAM, VTAM honors the pacing-window-size value included in the pacing response from the peripheral node but continues to use the appropriate receive count for its fixed-window-size value.

The primary send count specified in the logon mode entry of the primary logical unit is used by VTAM to control the rate at which a peripheral node acting as a primary logical unit sends data to an NCP. This value is then used by NCP or VTAM for a directly attached type 2.1 peripheral node to control the primary send pacing. If a negotiable BIND is used, the smaller value (other than 0) is used. If the BIND is nonnegotiable and the primary send value in the original BIND (other than 0) is larger than the value VTAM obtained from the logon mode entry or PACING specification, VTAM or NCP may reject the session setup request. For a directly attached type 2.1 peripheral node, VTAM performs the same type of operation.

Therefore, you must specify proper pacing values in VTAM for any type 2.1 independent logical unit. If the type 2.1 peripheral node supports adaptive-session pacing, this procedure is not necessary.

Application program pacing

This section describes how to use the APPL definition statement to specify pacing for application programs.

Inbound session pacing: Application programs can act as either primary or secondary LUs. If the application program is the primary logical unit, you define pacing with two values: a nonzero value in SSNDPAC on the secondary LU-mode table entry and the pacing value on the VPACING operand of the APPL definition statement. In this way, you can pace normal-flow requests from another logical unit.

If the application program is a secondary logical unit, you define pacing with either value: a nonzero value in PSNDPAC on the secondary LU-mode table entry, or the pacing value on the VPACING operand of the secondary logical unit APPL definition statement. If you code both values, the nonzero value in PSNDPAC overrides the VPACING value.

Outbound session pacing: When the application program is acting as the primary logical unit, you can specify AUTH=NVPACE to nullify pacing from that application program. VTAM does not wait for a pacing response before sending data to the NCP on behalf of this session. VPACE, the default, is normally used to prevent overloading buffers in the communication controller with outbound messages from VTAM. However, you can specify NVPACE when either of the following situations occur:

- The application program sends only single-element request units (RUs) to any one logical unit and, after sending each RU, waits for a response before sending the next RU.
- The application program sends chains of RUs containing a limited number of elements and does either of the following:
  - After sending one chain, the application waits for a response before it starts sending the next chain.
  - The application sends the change direction indicator in the last element of the chain.

Local flow control is a part of network flow control and can affect application program design. For primary or secondary application programs, macros that cause data to be sent on a session (SEND, SESSIONC, OPNDST, or CLSDST) are not posted complete when a virtual route (VR) is blocked due to congestion. The application program is posted when the VR is again opened. The data to be sent is left in application program buffers.

An application program that issues a synchronous SEND request does not receive control until the request is processed, even if the route is not blocked. For an asynchronous request, the program receives control as soon as the request is accepted by VTAM. However, further processing of the SEND RPLs for this session is suspended as long as the VR is blocked.

Sample configurations

The following examples show the relationship between VTAM pacing parameters and the initial pacing window sizes. Various network configurations are provided to illustrate a variety of pacing stages.

Same domain application program-to-application program session

For same domain application program-to-application program session pacing, there is one-stage adaptive pacing. Figure 78 shows the pacing stage for this example. See Table 17 and Table 18 for examples of the values used by VTAM to set the pacing-window values.

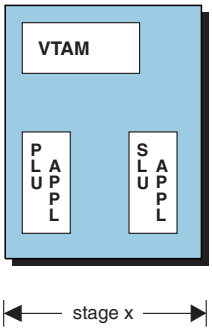


Figure 78. Same domain application program-to-application program session

Table 17 shows sample definition statement coding and corresponding initial send windows when data flows from the PLU to the SLU.

Table 17. Same domain application program-to-application program session—PLU to SLU flow

SRCVPAC MODEENT macro	**VPACING APPL definition statement (SLU)**	**AUTH APPL definition statement (PLU)**	**Send window stage x**
6	n/a	PACE	6
0	500	PACE	500
n/a	n/a	NVPACE	32767

Table 18 shows sample definition statement coding and corresponding initial send windows when data flows from the SLU to the PLU.

Table 18. Same domain application program-to-application program Session—SLU to PLU Flow

SSNDPAC MODEENT macro	**VPACING APPL definition statement (PLU)**	**Send window stage x**
0	n/a	32767
50	7 (default)	7
600	0	32767

Same domain application program-to-local device session

For same domain application program-to-local device session pacing there are two pacing stages. One stage, which is adaptive, exists between the application program and VTAM boundary function. Another pacing stage, which can be either fixed or adaptive, exists between the boundary function and the device LU. Figure 79 on page 316 shows the pacing stage for this example. See Table 19 on page 316 and Table 20 on page 316 for examples of the values used by VTAM to set the pacing-window values.

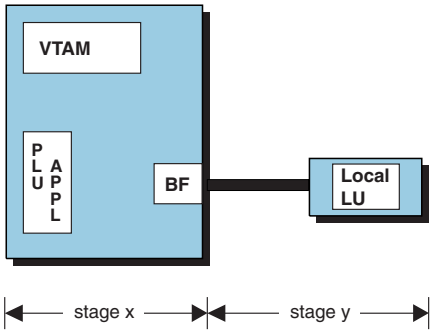


Figure 79. Same domain application program-to-local device session

Table 19 shows sample definition statement coding and corresponding initial send windows when data flows from the PLU to the SLU.

Table 19. Same domain application program-to-local device session—PLU to SLU flow

SRCVPAC MODEENT macro	VPACING definition statement (SLU)	Send window stage x	Send window stage y
6	n/a	6	6
0	500	500	500

Table 20 shows sample definition statement coding and corresponding corresponding initial send windows when data flows from the SLU to the PLU.

Table 20. Same domain application program-to-local device session—SLU to PLU flow

SSNDPAC MODEENT macro	VPACING APPL definition statement (PLU)	Send window stage x	Send Window Stage y
0	n/a	32767	32767
50	7 (default)	7	7
600	0	32767	32767

Application program-to-application program over APPN host-to-host connection

There are three pacing stages for application program-to-application program sessions over an APPN host-to-host connection. See Table 21 on page 317 and Table 22 on page 317 for examples of the values used by VTAM to set the pacing-window values. Figure 80 on page 317 show the pacing stages for this example.

Table 21 on page 317 shows sample definition statement coding and corresponding initial send windows when data flows from the PLU to the SLU.

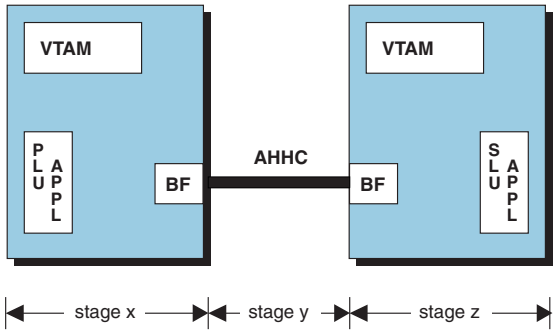


Figure 80. Application program-to-application program over APPN host-to-host connection

Table 21. Application program-to-application program over APPN host-to-host connection—PLU to SLU flow

SRCVPAC MODEENT macro	VPACING APPL definition statement (SLU)	Send window stage x	Send window stage y	Send window stage z
0	0	7	128	32767
0	7	7	7	7
50	n/a	50	50	50

Table 22 shows sample definition statement coding and corresponding windows when data flows from the SLU to the PLU.

Table 22. Application program-to-application program over APPN host-to-host connection—SLU to PLU flow

SSNDPAC MODEENT macro	VPACING APPL definition statement (PLU)	Send window stage x	Send window stage y	Send window stage z
0	n/a	32767	128	7
50	7	7	7	7
50	0	32767	128	7

Application program-to-application program over CTCA connection

There is one pacing stage, which is adaptive, for application program-to-application program session over a CTCA connection. Figure 81 on page 318 shows the pacing stage for this example. See Table 23 on page 318 and Table 24 on page 318 for examples of the values used by VTAM to set the pacing-window values.

Table 23 on page 318 shows sample definition statement coding and corresponding windows when data flows from the PLU to the SLU.

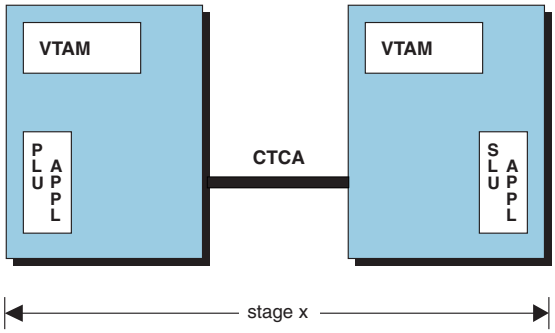


Figure 81. Application program-to-application program over CTCA connection

Table 23. Application program-to-application program over CTCA connection—PLU to SLU flow

SRCVPAC MODEENT macro	VPACING APPL definition statement (SLU)	AUTH operand (PLU)	Send window stage x
6	n/a	PACE	6
0	50	PACE	50
n/a	n/a	NVPACE	32767

Table 24 shows sample definition statement coding and corresponding windows when data flows from the SLU to the PLU.

Table 24. Application program-to-application program over CTCA connection—SLU to PLU flow

SSNDPAC MODEENT macro	VPACING APPL definition statement (PLU)	Send window stage x
0	n/a	32767
50	7 (default)	7
50	0	32767

Application program-to-local SNA over CTCA connection

There are two pacing stages for an application program-to-local SNA device session over a CTCA connection. Stage x uses adaptive pacing and stage y can use either adaptive or fixed pacing. Figure 82 on page 319 shows the pacing stages for this example. See Table 25 on page 319 and Table 26 on page 319 for examples of the values used by VTAM to set the pacing window values.

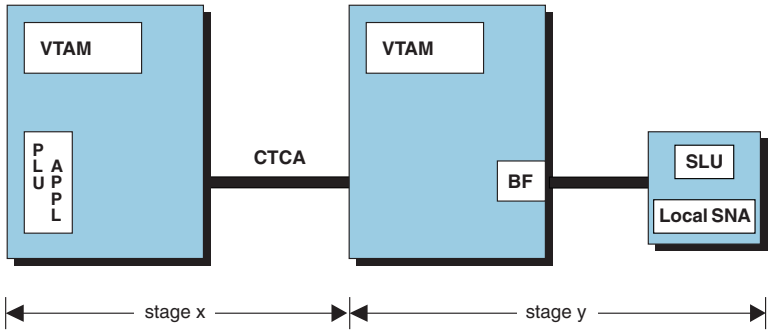


Figure 82. Application program-to-local SNA device over CTCA connection

Table 25 shows sample definition statement coding and corresponding windows when data flows from the PLU to the SLU.

Table 25. Application program to local SNA device over CTCA connection—PLU to SLU flow

SRCVPAC MODEENT macro	PSNDPAC MODEENT macro	PACING operand (SLU)	VPACING operand (SLU)	AUTH APPL definition statement (PLU)	Send window stage x	Send window stage y
0	n/a	n/a	4	NVPACE	95	4
0	15	n/a	4	PACE	15	4
5	0	6	n/a	PACE	6	5

Table 26 shows sample definition statement coding and corresponding windows when data flows from the SLU to the PLU.

Table 26. Application program-to-local SNA device over CTCA connection—SLU to PLU flow

SSNDPAC MODEENT macro	VPACING APPL definition statement (PLU)	Send window stage x	Send window stage y
9	2	2	9
0	7 (default)	7	32767
9	0	32767	9

Application program-to-local SNA device over AHHC connection

There are three pacing stages for application program-to-local SNA device session over an AHHC connection. Stage x and stage y use adaptive pacing and stage z uses either adaptive or fixed pacing. Figure 83 on page 320 shows the pacing stages for this example. See Table 27 on page 320 and Table 28 on page 320 for examples of the values used by VTAM to set the pacing window values.

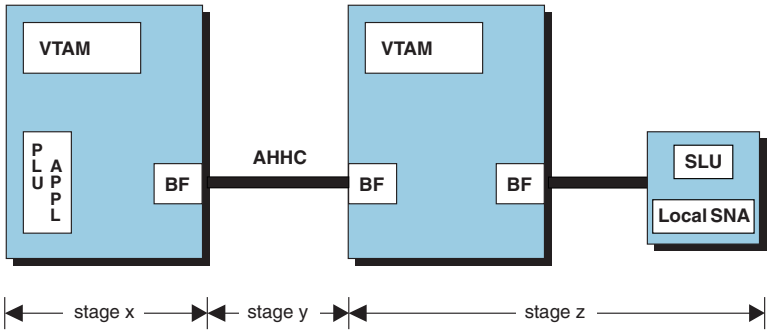


Figure 83. Application program-to-local SNA device over AHHC connection

Table 27 shows sample definition statement coding and corresponding windows when data flows from the PLU to the SLU.

Table 27. Application program-to-local SNA device over AHHC connection—PLU to the SLU

SRCVPAC MODEENT macro	PSNDPAC MODEENT macro	VPACING operand (PLU)	PACING operand (PLU)	Send window stage x	Send window stage y	Send window stage z
5	0	6	n/a	6	5	5
5	0	6	n/a	6	5	5
0	15	n/a	4	15	4	4

Table 28 shows sample definition statement coding and corresponding windows when data flows from the SLU to the PLU.

Table 28. Application program-to-local SNA device over AHHC connection—SLU to PLU flow

AUTH APPL definition statement (PLU)	SSNDPAC MODEENT macro	VPACING APPL definition statement (PLU)	Send window stage x	Send window stage y	Send window stage z
PACE	0	7 (default)	7	128	32767
NVPACE	100	n/a	32767	128	100
PACE	9	8	8	9	9

Application program-to-application program with VR from intermediate host-to-SLU host

There are three pacing stages for application program-to-application program with VR from an intermediate host to the SLU host connection. All three pacing stages use adaptive pacing. Figure 84 on page 321 shows the pacing stages for this example. Stage x and stage y use adaptive pacing and stage z uses either adaptive or fixed pacing. Figure 84 on page 321 shows the pacing stages for this example. See Table 29 on page 321 and Table 30 on page 321 for examples of the values used by VTAM to set the pacing window values. Figure 84 on page 321 shows the pacing stages for this example.

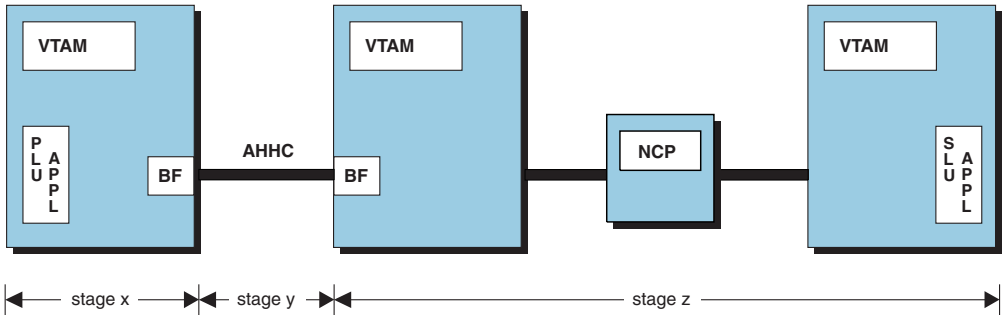


Figure 84. Application program-to-application program with VR from intermediate host-to-SLU host

Table 29 shows sample definition statement coding and corresponding windows when data flows from the PLU to the SLU.

Table 29. Application program-to-application program with VR from intermediate host-to-SLU host—PLU to SLU flow

SRCVPAC MODEENT macro	VPACING APPL definition statement (SLU)	Send window stage x	Send window stage y	Send window stage z
0	0	7	128	32767
0	7	7	7	7
50	n/a	50	50	50

Table 30 shows sample definition statement coding and corresponding windows when data flows from the SLU to the PLU.

Table 30. Application program-to-application program with VR from intermediate host-to-SLU host—SLU to PLU flow

SSNDPAC MODEENT macro	VPACING APPL definition statement (PLU)	VR window VRPWSxx	Send window stage x	Send window stage y	Send window stage z
0	n/a	(128,255)	32767	128	255
50	7	n/a	7	7	7
50	0	(128,255)	32767	128	255

Logon and logoff requests from dependent logical units

To request a session with an application program, an LU sends a logon request to VTAM specifying the application program name and, optionally, a logon mode name and some additional information. Logon and logoff requests can be field-formatted SNA Initiate and Terminate requests, which are built by the logical unit that sent them, or they can be (unformatted) character-coded commands, which are translated by VTAM into a field-formatted SNA request. VTAM uses session-level USS tables and interpret tables to do the translation.

A session-level USS table can be used to convert the USS command into a field-formatted SNA request. If a character-coded command violates the USS command syntax rules (refer to the *z/OS Communications Server: SNA Resource Definition Reference*), an interpret table must be used for the conversion.

You can customize VTAM to accept other character-coded commands by coding an interpret table. The following forms of requests make use of the interpret table to determine the name of the application program to which the request is being made:

- A field-formatted SNA request received directly from a logical unit
- An unformatted character-coded command, converted to a formatted SNA request

For example, any of the above forms of a request can indicate a logon to APPL01, and the interpret table can change APPL01 into TESTAPL1.

You associate an interpret table to a logical unit using the LOGTAB operand on the LU definition statement. You associate a USS table to a logical unit using the USSTAB operand on the LU definition statement.

For a complete description of operation- and session-level USS tables, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Unformatted logon requests using mixed-case passwords

Mixed case passwords can be used by applications if an SAF-compliant security product (such as RACF®) has enabled this support. In some cases, the USS LOGON DATA parameter is used to send the password to the application. If a terminal user enters a mixed case password on the USS LOGON command and it is translated to uppercase by the translation table, the logon will fail if the target application expects to receive the password in mixed case.

The USS LOGON command is displayed on the terminal as it is typed. Therefore, the password is displayed. For additional security, you can inform the terminal user to discontinue entering the password as part of the USS LOGON process. Instead, the application should prompt the terminal user for the password in a non-displayed field. If mixed case passwords are used and the terminal user continues to enter the password as part of the USS LOGON command, the logon will fail when using TRANSLATE=YES (the default) on the USSPARM because the password has been translated to uppercase.

To continue to allow the terminal user to enter the password on the LOGON command, use one of the following methods to support mixed case passwords. With any of these methods, if the user ID is entered with the password, you must first verify whether the application supports translating the user ID to uppercase. A simple test is to enter the DATA portion of the USS LOGON within single quotes with the user ID specified in lowercase. USS will not translate data within single quotes and the quotes are removed before the data is passed to the application. If the logon fails, the application does not support translating the user ID to uppercase and a terminal user must enter the user ID in uppercase and the password in mixed case for the methods suggested.

- If the current USS translate table is used to set all characters to uppercase, add the TRANSLATE=NO operand to the USSPARM macro for the corresponding USSCMD to prevent translation of the specified DATA USSPARM containing the password to uppercase. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for details.
- Because USS will not process data within single quotes against the translate table (the quotation marks are removed before the data is passed to the application), instruct terminal users to enter the DATA portion of the USS LOGON within single quotes.

- If the terminal user is specifying only APPLID and DATA on the USS LOGON command, an interpret table can be used. The interpret function passes all entered data to the application in untranslated format. Use REMOVE=Y on the LOGCHAR macro to remove the first non-blank string from the entered data. An interpret table should be set up to look for both an upper and a lowercase APPLID since the terminal user could enter the APPLID in lowercase.

Chapter 13. Network routing

Routing within a network is different for a subarea and an APPN network. In a subarea network, network paths are designed and coded by the user. In an APPN network, routing is a dynamic function between nodes.

Network routing and resource location for APPN nodes

VTAM uses topology and routing services (TRS) and directory services (DS) to monitor and maintain information on the topology and resources of the network.

TRS maintains information about nodes, transmission groups (TGs), and classes of service so that appropriate routes through an APPN network can be calculated.

TRS creates a network topology database, in which it maintains information about the current topology of the intermediate routing portion of an APPN network (network nodes and the transmission groups interconnecting them). This network topology database is replicated at every network node in an APPN network. Whenever a new network node or TG is added, or whenever the characteristics of an old resource change, a topology database update (TDU) that describes the current characteristics of the resource is generated and propagated to all network nodes.

When a network node first joins an APPN network (that is, establishes CP-CP sessions with other nodes), the other nodes send the joining node a copy of the network topology database. This is known as topology exchange. (A topology exchange that occurs when a network node first joins an APPN network is also called a full topology exchange or an initial topology exchange.) If the joining node later experiences planned or unplanned down time, it can remonitor the network topology by receiving all the topology updates that were exchanged since its topology database was last updated.

Directory services maintains a database that contains information on the location of LUs in the network. Entries are added to this database by resource definition, resource registration, and dynamically through searches. This means even if a session is requested to an LU whose location is not currently stored in the database, the location of this LU can be monitored, stored in the database, and session establishment will occur. Also, if a session is requested to an LU that cannot be located, that information is stored in the database if search reduction is enabled (see “Improving VTAM performance using start options” on page 31).

When a session is requested with an LU, VTAM uses its search capability and the directory services database to determine what node the resource is on. The search results give the location of the resource but not the route. VTAM dynamically determines the best route for the session using the APPN Class of Service requested for the session and the characteristics of the intermediate nodes and links stored in the topology database.

The defaults for the definition of the node and link characteristics and the VTAM ability to dynamically update the contents of the databases allow sessions to be established without requiring any definitions for resources outside the domain of that VTAM.

© Copyright IBM Corp. 1984, 2011

There are techniques you can use to improve the efficiency of resource location and session setup. These techniques include coding a central directory server and registering resources to a central directory server or a network node server. Resource registration occurs by default for some resources, but you can modify which resources are registered.

You can also checkpoint the topology and directory databases to ensure that information is retained during system outages. Checkpointing the directory database increases the chances that a resource can be located with a directed search instead of a broadcast search. Directed and broadcast searches are described in "Types of searches." Checkpointing the topology database reduces the number of topology database updates that must flow at startup.

You may also need to adapt the Class of Service definitions, link and node characteristics to your network, such as coding some links as secure and requesting a secure Class of Service for certain sessions.

The following sections give information on searches, checkpointing, and APPN Class of Service.

Types of searches

To establish a session, the location of the partner resource must be known to VTAM. If the location of the partner resource is not statically defined, it can be monitored dynamically as a result of a directed, broadcast, or cross-subnetwork search. A directed search is used when VTAM has location information for the resource as the result of a previous search or if there is a central directory server in the network. A broadcast search is used when VTAM has no information on the location of the resource. A cross-subnetwork search is used to determine the location of a partner resource in another subnetwork.

A VTAM network operator can also initiate a search for a resource. Unlike searches that are performed for session initiation requests, an operator-initiated search does not update the directory services database when a resource is found.

Directed search

What is a directed search?: A directed search uses information stored in the directory services database of a network node server to direct the search to the location of the requested LU. A directed search is sent to the node recorded as the owner of the requested LU to verify the information. A directed search can be sent to a network node server from an end node and to an end node from a network node server. A directed search can also be sent from a network node to a central directory server when the network node does not have information on the location of the destination logical unit or following a failed search.

Who originates a directed search?: A directed search is originated by a network node server for the LUs it serves, by a central directory server as a result of a search request from a network node, or by an end node to its network node server.

When is a directed search needed?: A directed search is needed when a network node server or a central directory server receives a request for an LU for which it has a location stored in its database. The search is used to verify the information. A directed search is also used by a network node server to send a search request to a central directory server if the network node server has no information on the

location of a requested resource. A directed search is used by an end node to request location information about a resource from the end node network node server.

Broadcast search

What is a broadcast search?: A broadcast search does not use database information about the location of a requested LU to propagate the search, because a directed search using database information has failed or there is no database information for the requested resource. Each network node receiving the search request sends the search to each of its adjacent network nodes; this allows the whole network to be searched. (VTAM end nodes do not allow searches for resources that are not registered. With other types of end nodes, broadcast searches include all domain end nodes that have requested that they be searched for the specific resource type.)

When the search reaches the network node serving the destination resource, that node sends back a positive reply to the first search request it receives. All subsequent search requests received by that network node as part of the same search are answered with a negative reply.

Who originates a broadcast search?: A broadcast search is initiated only by a network node server for the LUs it serves if there is no central directory server defined in the network. If there is a central directory server defined, only the central directory server can issue a broadcast search. If a central directory server is active in your network, a network node server sends a directed search to the central directory server instead of issuing a broadcast search. Only if the directed search is unable to reach the central directory server does the network node server issue its own broadcast search.

When is a broadcast search needed?: A broadcast search is needed when there is no directory information giving the location of a requested LU or a directed search using existing directory information fails.

- **Broadcast searches in a network without a central directory server**

  In a network without a central directory server, broadcast searches are performed by any of the network nodes. These network nodes may be acting on behalf of LUs attached directly to them, LEN-attached LUs, or (when the network node is acting as a network node server) on behalf of LUs attached to end nodes that they serve. When the network node receives a request from an LU for a session with another LU, it checks for information in the database about the location of the partner LU. If it does not find the location information for the partner LU, the network node server of the requesting logical unit initiates a broadcast search.

  The location information of the resource is not communicated to the other network nodes. If another network node server receives a request for the same resource and does not have the location information, it issues a broadcast search.

- **Broadcast searches in a network with a central directory server**

  In a network with a central directory server, broadcast searches are performed only by the central directory server. If a network node server does not know the location of a partner LU, it sends a directed search to the central directory server instead of issuing a broadcast search. The central directory server checks its database and issues a broadcast search only if it does not have the information. When the requested LU is located, the central directory server updates its directory and returns the information to the requesting network node. The network node then updates its database with the information and initiates the

session with the partner LU. After a resource location is stored in the central directory database, no further broadcast searches are required, until the resource moves or is deactivated.

Cross-subnetwork search

What is a cross-subnetwork search?: A cross-subnetwork search is designed to find an LU outside the local APPN subnetwork. Because other APPN searches are sent between nodes that are in the same APPN subnetwork, this search type is employed at border nodes to forward searches for a requested LU to other APPN subnetworks. The target nodes of a cross-subnetwork search may be found in the directory services database, monitored dynamically, or explicitly defined by definition statements.

Who originates a cross-subnetwork search?: The first border node encountered as the search leaves the requester is responsible for originating a cross-subnetwork search. The border node forwards the request over locally managed subnetwork boundaries. Other nodes in nonnative subnetworks then perform their search logic to find the resource in their local subnetwork or across their subnetwork boundaries. These searches are performed sequentially until the target resource is located.

Is a cross-subnetwork search simply a broadcast search sent over subnetwork boundaries?: No. Cross-subnetwork search logic finds resources across subnetwork boundaries with minimum disturbance to other subnetworks. The sequential fashion in which the searches are sent allows the search to terminate as soon as the resource is located, without having to search every connected subnetwork. In addition, routing information can be both defined or monitored dynamically, which directs the search toward the subnetwork that is most likely to own the resource.

Operator-initiated search

What is an operator-initiated search?: An operator-initiated search performs all valid searches for the node, including the following searches:

- Locally-attached subarea
- Subarea networks attached through other interchange nodes
- Broadcast
- Cross-subnetwork
- Local directory (directed search)

The operator-initiated search is started when a DISPLAY DIRECTRY, ID=*resource_name*, SCOPE=NSEARCH command is issued at a network node. The search returns all instances of the resource in the network. Unlike other searches, information returned by an operator-initiated search is not recorded in the directory services database.

When is an operator-initiated search needed?: The following are some examples of when you would use the operator-initiated search:

- During resource name administration, to verify that names chosen are unique within the network.
- To determine the owning control point of a resource.
- To determine whether duplicate names exist in the network that might result in session failures. If the information returned indicates that a naming conflict exists in the network, you can use the MODIFY RESOURCE command to change the registration of one of the resources.

Minimizing broadcast searches

You can control searches in the following ways:

- Define a central directory server to minimize the number of broadcast searches.
- Register resources to a directory database.
- Manage duplicate resource definition.

Search reduction also limits broadcast searches for unreachable viable resources in the network. See "Improving VTAM performance using start options" on page 31 for additional information on search reduction.

Also, an installation-wide directory services management exit provides control over the extent to which VTAM conducts search requests in the network in which the directory services management exit is loaded. The directory services management exit is the interface between VTAM directory services and user-written code. VTAM conducts searches based on steps authorized by the user code. For more information about the directory services management exit, refer to *z/OS Communications Server: SNA Customization*.

Central directory servers

A central directory server is a focal point for broadcast searches in the network. It uses broadcast and directed searches to locate resources. When a central directory server is active in the network, all network node servers issue directed searches to the central directory server instead of issuing a broadcast search.

What are the benefits of a central directory server?: A central directory server reduces the number of broadcast searches in your network. When the central directory server performs a broadcast search for a network node, it stores the location of the requested LU in its database. If another network node server sends a directed search for the same resource to the central directory server, the central directory server does not have to issue another broadcast search because it already has the resource in its database.

Because a central directory server can receive search requests from many network nodes, its database grows faster and contains more resources than that of a network node server, which only receives the search requests of its served LUs. This centralization increases the chance that a resource will be found in the database and a broadcast search will not be required.

How does a central directory server receive a search request?: When a network node receives a search request, it checks its database for the resource. If it does not find the resource in its database, it sends the request to a central directory server if one exists in the network. If there is more than one central directory server in the network, the network node selects a central directory server based on one of the following:

- If the CDS selection function of the directory services management exit is active, the network node selects a central directory server from the list arranged by the exit. For information about how to use this function, refer to *z/OS Communications Server: SNA Customization*.
- If the CDSREFER start option value is 1 (the default), the network node sends the search request to the nearest (minimal weight route) central directory server in the network. For more information about the CDSREFER start option, refer to the *z/OS Communications Server: SNA Resource Definition Reference*. If the CDS selection function of the directory services management exit is active, then the CDSREFER start option value is used only for ordering the list of central directory servers passed to the exit.

How does a central directory server handle a search request?: When the central directory server receives a search request, it checks its database for the resource. If it does not find the resource in its database and there are other central directory servers in the network, it sends the search to the other central directory servers only. The selection of alternate central directory servers can be customized by using the alternate CDS selection function of the directory services management exit. If the central directory server receives a positive reply from any of the other central directory servers, it verifies the information, updates its own database with the information, and notifies the originating network node of the location of the requested LU. If the central directory server receives negative replies from all the other central directory servers, it initiates a broadcast search. If the central directory server that received a search request locates the resource in its own database, it verifies the information with a directed search and sends a reply to the originating network node server with the location of the requested LU.

If the central directory server does not find the resource in its database and there are no other central directory servers in the network, it initiates a broadcast search.

What do you need to define to take advantage of a central directory server?: Use the CDSERVR start option to define VTAM as a central directory server. You do not need to define anything on the other network nodes. The other network nodes will find out about the existence of the central directory server through normal topology exchanges.

Registering resources

Resource registration places information about the location of resources in a directory services database. This registration reduces broadcast searches by ensuring that a resource will be found in the directory services database. Resources can be registered to a directory database on a network node server or to a central directory server, or both. By registering resources to a network node server, you ensure that the resources will be found during a search. By registering resources to a central directory server, you ensure that the resources will be found without requiring a broadcast search.

Resources are registered by using the REGISTER operand on the APPL, CDRSC, LOCAL, and LU definition statements. This operand can also specify that a resource is not registered. The defaults for registration change depending on the type of resource and whether the resource is located on an end node or a network node. Table 31 shows the default registration values for resources.

Table 31. Default registration values for resources

Major node	Definition statement	Default registration
Application program	APPL	Central directory server (REGISTER=CDSERVR)
CDRSC	CDRSC	No registration (REGISTER=NO)
Local Non-SNA	LOCAL	Network node server (REGISTER=NETSRVR)
LUGROUP	LU	Network node server (REGISTER=NETSRVR)
Local SNA Model NCP Switched	LU	Network node server for dependent LUs, LOCADDR > 0 (REGISTER=NETSRVR) No registration for independent LUs, LOCADDR = 0 (REGISTER=NO)

Dynamic registration for a resource can be initiated while VTAM is running by using the MODIFY RESOURCE command. For more information, refer to *z/OS Communications Server: SNA Operation*.

Because the characteristics of an independent LU and of the node on which it resides are not recorded by VTAM, it is difficult to determine the appropriate level of registration for all cases. Therefore, independent LUs are not registered by default.

VTAM end nodes register their dependent LUs with their network node server by default. This includes all resources defined by LOCAL definition statements and LU definitions with a nonzero LOCADDR value. A resource on a VTAM end node that is not registered to the network node server will not be found during a search unless that resource has initiated a request through the network node server. Registering all end node resources ensures that they will be located by a search. You can also specify that these resources be registered to a central directory server.

Applications are registered to a central directory server by default, with the exception of TSO applications. For more information on default registration of TSO applications, see "Resource registration in an APPN network" on page 719.

Note: If a resource is never going to be a destination resource, there is no need to register it.

Resources on a network node can also be registered to a central directory server if one exists. By default, network nodes register only their applications to a central directory server.

Controlling which central directory server receives registration requests: If more than one central directory server exists in the network and the value for the CDSREFER start option is 1 (the default), the network node sends all central resource registration requests to the nearest (minimal-weight route) central directory server. You can change this value if you want to disperse directory information to other central directory servers in the network. For more information about this start option, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

You can alternatively specify which central directory servers receive registration requests by using the central resource registration CDS selection function of the directory services management exit. For information about how to use this function, refer to the *z/OS Communications Server: SNA Customization*.

Controlling the timeliness and size of the directory database

By using the start options DIRTIME and DIRSIZE, you can control how long a resource remains in the directory services database and how many dynamic entries are maintained.

The DIRTIME start option specifies the amount of time a dynamic resource remains in the directory database without being used before it is deleted. The default value for this start option is eight days.

The DIRSIZE start option defines the maximum number of dynamically added resources that are stored in the database. After the number of dynamic resources

reaches the maximum, new resources are added to the database in place of the resources that have gone the longest without being referenced.

Network search overhead caused by duplicate resource definitions

In an effort to facilitate recovery procedures, you might choose to define the same application programs or dependent LUs on more that one VTAM simultaneously. Because SNA does not support duplicate resource definition, your search logic must be able to locate the active instance of the resource first. If a nonactive instance is located first, the session usually fails to set up. However, when an APPN broadcast search is performed, you have little or no control over which instance is located first, and this can result in intermittent session failures.

To allow continued use of duplicate resource definitions, the APPN search logic in VTAM was modified to recognize nonactive resource search replies. The search will continue in the APPN network for the active instance of the specified resource. If an active instance is located, the reply from that instance is used and session setup continues. If an active instance is not located, the reply from a nonactive instance is used instead.

Although this additional search logic allows the continued use of duplicate resource definitions in VTAM, it does this by increasing the amount of network search traffic generated by VTAM. If your application program or dependent LUs are not in the active state, you might incur this overhead, even if you are not using duplicate resource definitions. To minimize or avoid this additional searching, you can use DUPDEFS start option. This start option lets you specify what types of resources (application programs, dependent LUs, or both) have duplicate definitions on this and other VTAMs.

Specifying DUPDEFS=ALL means that application programs and dependent LUs may be duplicated on this and other VTAMs. DUPDEFS=APPL means that only application program definitions may be duplicated. DUPDEFS=DEPLU means that only dependent LU definitions may be duplicated. In all cases, if the target resource is one that might have duplicate definitions and the resource is not in the ACTIVE state, VTAM returns a reply indicating that searching should continue to find an active instance of the resource. If the target resource is one that does not have duplicate definitions, the search reply returned by VTAM will indicate that searching should not continue (regardless of the current state of the resource). DUPDEFS=NONE means that no duplicate definitions exist on this and other VTAMs, which prevents the additional search logic from being performed.

Avoiding congestion

When search requests and search replies cannot be delivered immediately to a node, VTAM queues the data for delivery when possible. In situations where a node is not responding and searches are accumulating rapidly, VTAM storage can be excessively used. To prevent this overuse of VTAM storage, the MAXLOCAT start option can be specified.

The value of the MAXLOCAT operand is compared to the product of the number of queued searches and the amount of time between responses from the node. If the product is greater than the MAXLOCAT value, VTAM issues message IST1601I, which indicates that routing to this node is suspended and VTAM has stopped sending searches to the node until the congestion is reduced.

Even after VTAM suspends sending APPN search requests to an adjacent CP, VTAM continues to queue APPN search replies to the adjacent CP. If the number of queued requests plus the number of queued replies reaches 200% of MAXLOCAT and a minimum time interval has been exceeded, VTAM issues message IST2243I indicating that VTAM will initiate termination of the CP-CP session to the adjacent CP. VTAM ends the CP-CP session with sense code 80030004 and retries the CP-CP session. This allows the storage related to the queued APPN searches to be cleaned up.

The minimum time interval is based on how long it has been since VTAM issued message IST1601I indicating routing to this node has been suspended. If IOPURGE is less than the default of 180 seconds, then the minimum time interval is 180 seconds. Otherwise, the minimum time interval is the current IOPURGE value.

When coding a value for MAXLOCAT, consider the amount of VTAM storage available and the response time of nodes. For example, set the value high if there is plenty of available storage and a slow node exists.

Checkpointing of the TRS database and the directory database

When a network node is added to an APPN network, it must complete a full topology exchange with the other nodes in the network. In large APPN networks, full topology exchanges can involve the transfer of large amounts of data, affecting performance by slowing down the CP-CP session activation process.

When a node that has just joined the APPN network attempts to establish sessions, it has to monitor the location of all resources that it wishes to establish sessions with by issuing a broadcast search or a directed search to a central directory server.

By checkpointing the TRS and directory services databases, the topology and resource location information can be saved and used if VTAM is restarted.

Checkpointing saves the databases to the checkpoint data sets at the time either of the following occurs:

- The VTAM operator issues a MODIFY CHKPT command.
- The VTAM operator issues a HALT or HALT QUICK command to terminate the VTAM network node.

This process of saving the databases is also called database hardening.

Note: The databases are not saved if the VTAM operator does the following:

- Issues a HALT CANCEL command to terminate the VTAM network node
- Cancels VTAM by canceling VTAM partition

When a database is checkpointed, it is written to a checkpoint data set. This data set is read by VTAM when it is restarted and the information is placed in the databases. If checkpointing is not used, both the topology and directory services databases have to be completely rebuilt. When checkpointing is used, only those changes to resources and topology since the last checkpoint need to be monitored.

To specify which of the checkpointed databases are to be loaded at startup time, you can use the INITDB start option. By default, both the topology and directory services databases are loaded. You can also specify topology only, directory services only, or neither.

APPN Class of Service

What Is APPN Class of Service?

APPN Class of Service defines the characteristics of a route for a session. Unlike subarea Class of Service, which defines the actual routes to be used for a session, APPN Class of Service defines the characteristics of the nodes and transmission groups (TGs) to be used for a session. VTAM chooses an APPN route by matching the required characteristics defined in the COS definition with the actual characteristics of the nodes and TGs in the topology database.

What information is specified in the APPN Class of Service?

The APPN Class of Service indicates the desired characteristics of the nodes and lines in a route and the transmission priority of sessions. The possible line characteristics that can be specified include the security, speed (line capacity), propagation delay, and cost (both per unit of time and per byte). There are also three user-defined parameters that can be used to specify additional characteristics. The node characteristics that can be specified are the level of congestion allowed when selecting a node for routing and the desirability of the node for intermediate routing. The desirability of using a node for intermediate routing is called route resistance.

The characteristics are specified in ranges, with a low weight assigned to the most desirable set of ranges and a high weight assigned to the least desirable set of ranges. The first row in the default Class of Service has a very low weight and has the most restrictive set of ranges. The next set of ranges is less restrictive but has a higher weight. (These sets of ranges are called LINEROWs and NODEROWs.) The last set of ranges has the widest range of acceptable characteristics but has a very high weight. By coding the rows in this manner, the best resources (lowest weight and best characteristics) are chosen when determining the route.

Do you have to code anything to make APPN Class of Service work?

IBM supplies seven default APPN Class of Service definitions. The TG and node characteristics also default. For APPN-only sessions, VTAM locates an available route without any additional coding.

What are the IBM-supplied default classes of service?

Descriptions of the seven IBM-supplied APPN Classes of Service follow. These classes define seven generic types of session traffic. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for samples.

#BATCH
: An APPN COS for LU-LU sessions that specifies a general batch-oriented COS that uses low transmission priority, and for which high bandwidth and low cost are considered more important than short delay.

#BATCHSC
: An APPN COS for LU-LU sessions that specifies a general batch-oriented COS that uses low transmission priority, and for which high bandwidth and low cost are considered more important than short delay. A minimum security level is required.

#CONNECT
: An APPN COS for LU-LU sessions that provides connectivity at medium transmission priority.

CPSVCMG
An APPN COS for CP-CP sessions that is used for network flows. It provides connectivity at network transmission priority.

#INTER
An APPN COS for LU-LU sessions that specifies a general, interactive COS that uses high-transmission priority, and for which short delay is considered more important than high bandwidth and lost cost.

#INTERSC
An APPN COS for an LU-LU session that specifies a general, interactive COS that uses high-transmission priority, and for which short delay is considered more important than high bandwidth and lost cost. A minimal security level is required.

SNASVCMG
An APPN COS for LU-LU CNOS sessions that provides connectivity at network transmission priority.

Three sets of IBM-supplied classes of service are available. Each set contains the seven classes described previously. The three sets are:

- COSAPPN

  The definitions in the COSAPPN definition set are made up of 8-row LINEROW entries and 8-row NODEROW entries for all classes of service and are appropriate for most sessions. The LINEROW values for the #CONNECT class of service in COSAPPN are shown in Table 32.

Table 32. COSAPPN #CONNECT class of service LINEROW values

LINEROW	CAPACITY	COSTBYTE	COSTTIME	PDELAY	SECURITY	UPARM1 UPARM2 UPARM3	WEIGHT
1	4M MAXIMUM	0 0	0 0	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	30
2	56000 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	60
3	19200 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	90
4	9600 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	120
5	19200 MAXIMUM	0 0	0 0	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	150
6	9600 MAXIMUM	0 128	0 128	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	180
7	4800 MAXIMUM	0 196	0 196	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	210
8	MINIMUM MAXIMUM	0 255	0 255	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	240

.

- ISTACST2

  The definitions in the ISTACST2 definition set are made up of 12-row LINEROW entries and 8-row NODEROW entries for all classes of service except CPSVCMG and SNASVCMG, which are made up of 8-row LINEROW entries and 8-row NODEROW entries. These 12-row LINEROW entries enable z/OS Communication Server to select an optimal route for a session. The greater

number of entries provides optimal results when multiple connections with different TG characteristics (for example, channel-to-channel, token-ring network, FDDI LAN, and ATM) are used in the network.

The LINEROW values for the #CONNECT class of service in ISTACST2 are shown in Table 33.

Table 33. ISTACST2 #CONNECT class of service LINEROW values

LINEROW	CAPACITY	COSTBYTE	COSTTIME	PDELAY	SECURITY	UPARM1 UPARM2 UPARM3	WEIGHT
1	100M MAXIMUM	0 0	0 64	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	20
2	10M MAXIMUM	0 0	0 64	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	40
3	4M MAXIMUM	0 0	0 64	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	60
4	10M MAXIMUM	0 0	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	80
5	56000 MAXIMUM	0 0	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	100
6	29K MAXIMUM	0 0	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	120
7	100M MAXIMUM	0 128	0 128	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	140
8	10M MAXIMUM	0 128	0 128	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	160
9	56000 MAXIMUM	0 128	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	180
10	29K MAXIMUM	0 128	0 128	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	200
11	9600 MAXIMUM	0 0	0 64	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	220
12	MINIMUM MAXIMUM	0 255	0 255	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	240

- ISTACST3

  The definitions in the ISTACST3 definition set are made up of 12-row LINEROW entries and 8-row NODEROW entries for all seven classes of service. The definitions enable z/OS Communications Server to select an optimal route for a session when connections in the network include connections with high speed link characteristics such as FICON®, Gigabit Ethernet, and HiperSockets.

  The LINEROW values for the #CONNECT class of service in ISTACST3 are shown in Table 34.

Table 34. ISTACST3 #CONNECT class of service LINEROW values

LINEROW	CAPACITY	COSTBYTE	COSTTIME	PDELAY	SECURITY	UPARM1 UPARM2 UPARM3	WEIGHT
1	10G MAXIMUM	0 0	0 64	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	20
2	1G MAXIMUM	0 0	0 64	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	40

Table 34. ISTACST3 #CONNECT class of service LINEROW values (continued)

LINEROW	CAPACITY	COSTBYTE	COSTTIME	PDELAY	SECURITY	UPARM1 UPARM2 UPARM3	WEIGHT
3	100M MAXIMUM	0 0	0 64	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	60
4	1G MAXIMUM	0 0	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	80
5	25M MAXIMUM	0 0	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	100
6	1M MAXIMUM	0 0	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	120
7	10G MAXIMUM	0 128	0 128	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	140
8	1G MAXIMUM	0 128	0 128	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	160
9	16M MAXIMUM	0 128	0 64	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	180
10	1M MAXIMUM	0 128	0 128	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	200
11	64K MAXIMUM	0 0	0 64	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	220
12	MINIMUM MAXIMUM	0 255	0 255	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	240

The definition sets COSAPPN, ISTACST2, and ISTACST3 are shipped in SYS1.ASAMPLIB. To use one of these sets, you must copy it into SYS1.VTAMLST when you install z/OS Communication Server, and then activate the member in which the definitions reside. You can copy multiple sets of definitions into SYS1.VTAMLST, but you can have only one set active at any time.

The COSAPPN definition set is automatically activated when z/OS Communication Server is initialized. If you choose to use ISTACST2 or ISTACST3, use the VARY ACT command to activate it, or place the ISTACST2 or ISTACST3 member in the configuration list to automatically activate it when z/OS Communication Server is initialized. You can rename the IBM-supplied sets of definitions so that ISTACST2 or ISTACST3 is named COSAPPN, and COSAPPN is either not used or is renamed to something else. Renaming the definition sets in this way causes the set of definitions with 12-row LINEROW entries to be automatically activated when z/OS Communication Server is initialized.

Guideline: If you activate a set of definitions with 12-row LINEROW entries, you should have the same set of definitions with 12-row LINEROW entries activated on each network node in the network for optimal routing in networks that include ATM native or high-speed connections.

Not all HPR APPN products support COS definitions with 12-row LINEROW entries. This could affect your ability to optimally use native ATM connections or high speed connections among the nodes in your network. Consult technical representatives for the HPR APPN products in your network to determine if those products support COS definitions with 12-row LINEROW entries.

If you use COS definitions with 12-row LINEROW entries, routes selected for sessions could be different than those selected when you use COS definitions with 8-row LINEROW entries.

What do you get if you accept all the defaults?

The name of the APPN Class of Service definition associated with the mode is specified by using the APPNCOS operand on the MODEENT macroinstruction. Values are supplied for the APPNCOS operands in the IBM-supplied logon mode table (ISTINCLM). The value for APPNCOS in the IBM-supplied logon mode table is coded with the APPN Class of Service that most closely matches the type of session the logon mode entry is intended for. For example, if the logon mode name is coded as BATCH, BAT13790, BAT23790, or #BATCH, the APPNCOS operand value is #BATCH [3]. For the complete list of logon mode names and APPNCOS values, see Table 35.

Table 35. ISTINCLM APPNCOS values based on LOGMODE operand values

IBM-supplied mode name	**APPNCOS value**
CPSVCMG	CPSVCMG
CPSVRMGR	SNASVCMG
SNASVCMG	SNASVCMG
#BATCH BATCH BAT13790 BAT23790	#BATCH
#BATCHSC	#BATCHSC
#INTER INTERACT INTRACT INTRUSER	#INTER
#INTERSC	#INTERSC
DSILGMOD DSC4K D32901 EMUDPCX SCS xxx3270x xxx3277x xxx3278x xxx3790x etc.	#CONNECT
ISTCOSDF	#CONNECT
IBMRDB	#CONNECT

What is the APPN Class of Service default in a user-coded logon mode table?

If the APPNCOS operand is not coded on the MODEENT macroinstruction, VTAM checks the COS operand value.

- If the COS operand value is coded, VTAM checks for an APPNCOS definition by the same name as the COS name. If VTAM does not find an APPNCOS definition by the same name as the COS, it fails the session request, unless the APPNCOS start option is defined with a substitute APPN Class of Service name.
- If the COS operand is not coded, VTAM defaults APPNCOS to #CONNECT, regardless of the setting for the APPNCOS start option.

3. # represents X'7B'. You should use whichever character on your system equates to X'7B'.

How do I add APPN Class of Service to my logon mode table?

You can add APPN Class of Service to your user-coded logon mode table in one of three ways:

- You can create APPNTOSA and SATOAPPN tables to allow VTAM to map Class of Service between APPN and subarea networks, without having to change existing logon mode tables.
- You can code the APPNCOS operand for each logon mode entry in the table.
- You can create new APPNCOS definitions with the same name as the subarea COS.

To create APPNCOS definitions with the same name as the subarea COS, you can copy the default definition for APPN Class of Service that best fits that logon mode and change the name to match the COS operand coded on that logon mode.

How does z/OS Communication Server use the Class of Service to choose a route?

z/OS Communication Server chooses a route by comparing the actual characteristics of the available nodes and TGs to the allowed characteristic ranges specified in the requested COS. For each APPN COS entry, there are 1–12 LINEROW operands and 1–8 NODEROW operands. These operands give up to 12 acceptable sets of characteristics for the lines and up to 8 for the nodes in each COS. The WEIGHT parameter on the NODEROW and LINEROW operands is coded to indicate the desirability of that set of characteristics. The lower the value of the WEIGHT parameter, the higher the desirability of a node or TG that fits that set of characteristics.

Tip: The IBM-supplied COS definitions in COSAPPN are made up of 8-row LINEROW and NODEROW entries for all classes of service. The IBM-supplied COS definitions in ISTACST2 are made up of 12-row LINEROW entries for all classes of service except CPSVCMG and SNASVCMG, (which are made up of 8-row LINEROW entries), and 8-row NODEROW entries for all classes of service. The IBM-supplied COS definitions in ISTACST3 are made up of 12-row LINEROW entries and 8-row NODEROW entries for all classes of service. See “Class of Service” on page 335 for more information.

For each of the LINEROW and NODEROW operands, an acceptable range is coded. For each potential line or node in a route, z/OS Communication Server compares the node or TG characteristics to the LINEROW or NODEROW operand values in the requested Class of Service. For an entire route to be acceptable for a given Class of Service, all nodes and TGs for that route must be acceptable for the requested Class of Service.

If more than one route fits the required characteristics of the Class of Service, z/OS Communication Server chooses the one with the least total weight, and therefore the most desirable characteristics. If no route matches the required characteristics of the Class of Service, the session fails (for example, if the Class of Service requires a secure route and VTAM cannot find a complete route using only secure TGs).

Influencing session routing

The RESUSAGE start option is used to affect the number of times a given resource is used in creating a route. RESUSAGE specifies the number of times a resource can be used before it is considered overused. When a resource is overused, any routing trees using it are reconstructed in an attempt to use alternate resources.

Example of weight computation

The following example illustrates how a class of service is used to assign a weight to an element of a route. This example illustrates what weight is assigned to a TG with default characteristics using the #CONNECT class of service. Default characteristics of a TG are shown in Table 36. Default LINEROW values for the #CONNECT Class of Service are shown in Table 37.

Table 36. Default TG characteristics

TG number	TG1
CAPACITY	8K
COSTBYTE	0
COSTTIME	0
PDELAY	TERRESTR
SECURITY	UNSECURE
UPARM1	128
UPARM2	128
UPARM3	128

Table 37. #CONNECT Class of Service LINEROW values

LINEROW	CAPACITY	COSTBYTE	COSTTIME	PDELAY	SECURITY	UPARM1 UPARM2 UPARM3	Weight
1	4M MAXIMUM	0 0	0 0	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	30
2	56000 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	60
3	19200 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	90
4	9600 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	120
5	19200 MAXIMUM	0 0	0 0	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	150
6	9600 MAXIMUM	0 128	0 128	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	180
7	4800 MAXIMUM	0 196	0 196	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	210
8	MINIMUM MAXIMUM	0 255	0 255	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	240

The values shown in Table 37 for capacity, cost per byte, cost per unit of time, propagation delay, security level, and the user-defined characteristics (UPARM1, UPARM2, and UPARM3) represent ranges, with the top value in a LINEROW representing the minimum value and the bottom value in a LINEROW representing the maximum value. In comparing these values with the TG characteristics shown in Table 36, the first match occurs at row 7.

The cost per byte, cost per unit of time, security, and user-defined characteristics are acceptable for all rows. However, the possible LINEROW values for capacity (number of bits per second that the link can transmit data) are MINIMUM, 600,

1200, 2400, 4800, 9600, 14400, 19200, 48000, 56000, 64000, 4M, 16M, and MAXIMUM, and effective capacity is not within the range for rows 1 through 6. The possible values of propagation delay (from lowest to highest) are:

- MINIMUM
- NEGLIGIB
- TERRESTR
- PACKET
- LONG
- MAXIMUM

Propagation delay is not within the range for the first row.

So, the default values for a TG match both rows 7 and 8 of the #CONNECT Class of Service. Row 7 has a weight of 210 and this weight is assigned to this TG for the route being calculated. Row 8 has a weight of 240, but even if it had a lesser weight it would not be used. (It is recommended that you code your rows from least weight to highest weight to ensure that the least weight row is used in the event of multiple matches.)

This TG is used if no other TG is available. If another TG is available, it is used if its weight is less than 210. In the event two or more TGs are equally weighted, one is chosen at random.

If this TG is being considered for part of a route that involves nodes and other TGs, the weight of this TG would be added to the weights of the other TGs and nodes, to produce a total weight for the route.

If the origin or destination of this TG is a virtual node, the weight of the TG is divided by two and the calculated value is used in route calculation.

For an example of how to use the node and TG characteristics to force a route, see Appendix H, “Forcing an APPN route in a VTAM network,” on page 773.

APPN network routing through a composite network node (CNN)

In a configuration containing a composite network node (CNN), it is possible that the optimal route computation does not occur. Route computation is performed by the network node serving the origin logical unit (OLU). For example, in Figure 85 on page 342, a route from LU to APPL is computed that uses TG1-TG4. TG1-TG3 would be a better route.

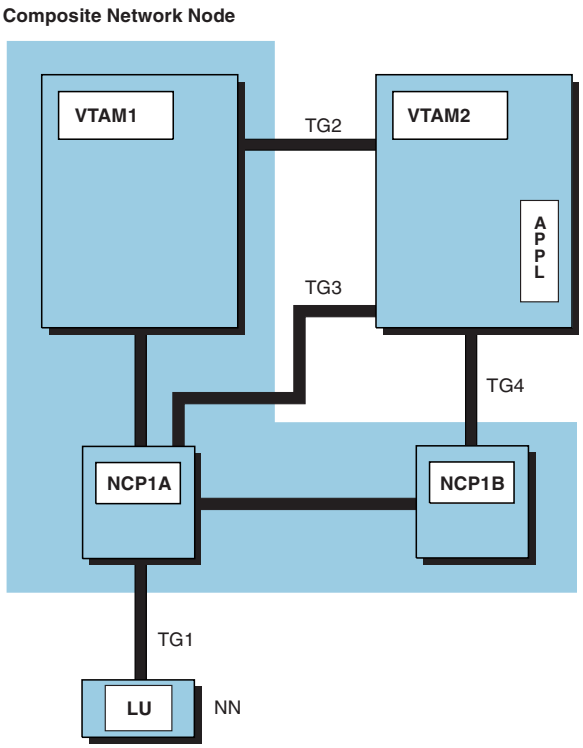


Figure 85. Routing example through a CNN node

VTAM in the composite network node (CNN) node affects APPN routing by:

- Selecting an optimal route through the CNN

  When computing a route and choosing among TGs of equal weight, VTAM searches for matching subarea numbers for the entry and exit TGs to a CNN (TG1 and TG3 in this example, since they share NCP1A). If TGs with matching subarea numbers are found, VTAM selects that route. In situations where no match is found (for example, if TG3 did not exist), the original route is used (either TG1–TG4 or TG1–TG2).
- Changing the route during BIND processing

  The CNN node affects routing during BIND processing. When a BIND reaches the CNN node, the next hop in the RSCV contained in the BIND may be changed by VTAM in the CNN node when the entry and exit TGs use two different subareas. VTAM changes the route during BIND processing when the following conditions are met:
  - The subarea number of the new exit TG is the same as the subarea number of the entry TG.
  - TG characteristics are the same as the original TG.

Figure 86 on page 343 shows an example of VTAM changing the selection of a route. In Figure 86 on page 343 mesh connectivity exists between VTAMA, VTAMB, NCPA, NCPB, and NCPC. In selecting a path between EN1 and NN2, VTAMA determines that using TG2-TG3 is the optimal route. Because NN2 does not use subarea numbers in route calculation, it might choose TG2-TG4 when selecting a route from NN2 to EN1.

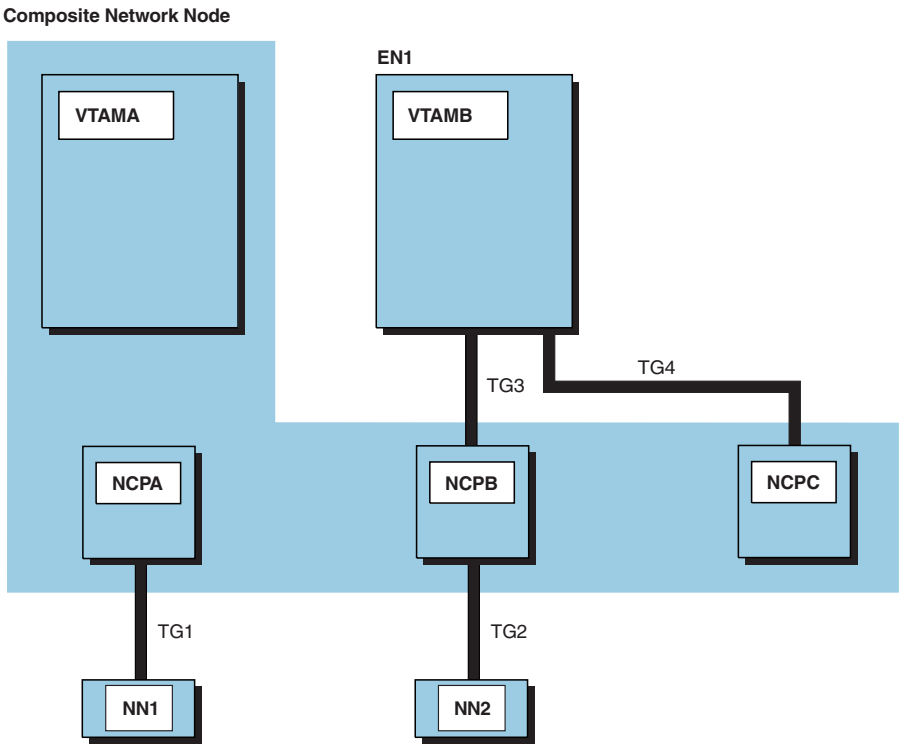


Figure 86. CNN route calculation example

Figure 87 shows another example of route calculation through a CNN node. VTAMA, the dependent LU server for ENA, calculates a route of TG5-TG6.

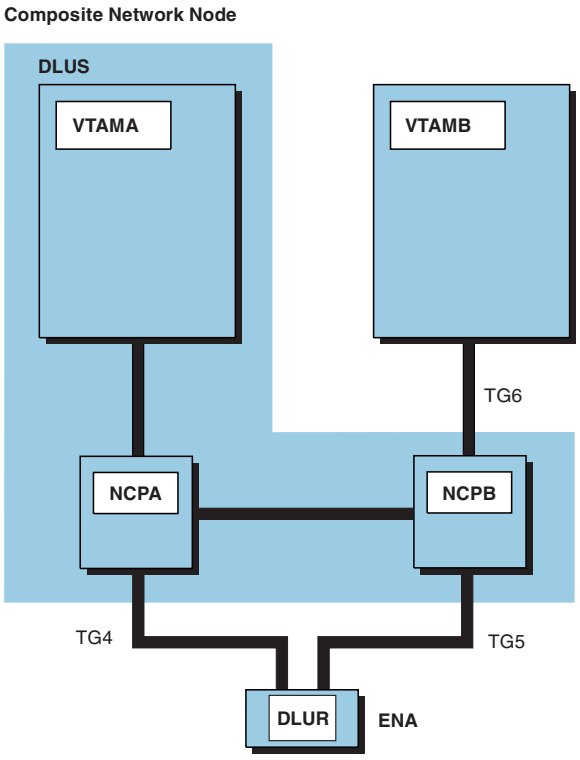


Figure 87. Composite network node route calculation example

Figure 88 is an example of the CNN node changing the route when receiving a BIND request. A BIND request sent from an LU located on NN1 to an LU on NN2 contains TG1-TG2 in the RSCV. VTAMA changes the RSVC to use TG1-TG3. By rerouting the BIND to a TG with the same subarea as the entry TG, VTAM prevents unnecessary hops within the CNN.

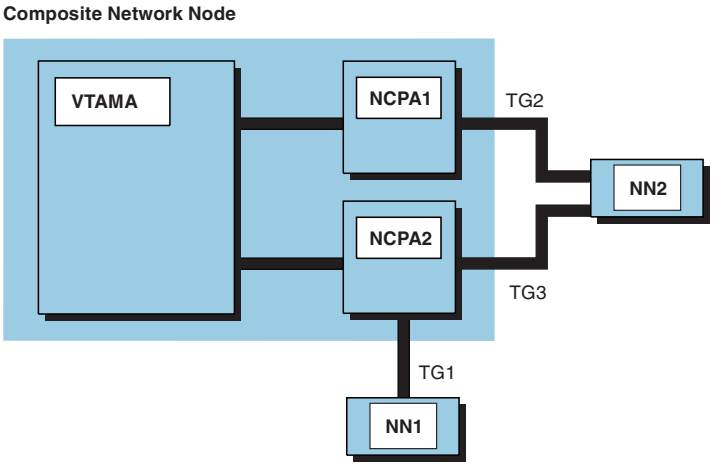


Figure 88. Composite network node route BIND reroute example

Non-optimal routes might result because the topology of the CNN is not known by the APPN topology and route selection process, or because the route was calculated by a non-VTAM node that does not support the use of subarea numbers in route calculation. For example, referring to Figure 88, if NN2 calculates TG2-TG1 as the route to go from NN2 to NN1, when the BIND arrives at VTAMA it cannot be changed to use TG3. It is too late in the process for the CNN node to change the route. However, if CNNRTMSG=NOSUPP is in effect, VTAMA issues the IST1774I message group to indicate that an optimal route through the CNN does exist but was not chosen during session activation.

Note that all the optimizations concern *equally* weighted TGs. For this reason, making the characteristics of TGs between two identical nodes allows VTAM more flexibility in optimizing paths through CNNs.

Using the SAMAP table

The subarea mapping (SAMAP) table is used to supply additional NCP connectivity information to APPN topology and routing services (TRS), which can then use that information in an effort to optimize routes through the CNN.

In configurations with remote NCP, the relationship between the local and remote NCP is not known to APPN TRS. Therefore, the subarea number mapping process (described in the previous section) is not adequate. The SAMAP table can be used to specify local-to-remote NCP connectivity for use by APPN TRS during route computation.

The SAMAP table can also be used to map the subarea number for a gateway NCP to the Net ID for the SNI-connected network that will be accessed through that gateway NCP.

Subarea number mapping

While APPN topology and routing services is not normally aware of the subarea connectivity within the CNN, the SAMAP table can be used to provide NCP-to-NCP mapping information that can be used to improve the intra-CNN routing in many cases.

Figure 89 illustrates a typical session path scenario.

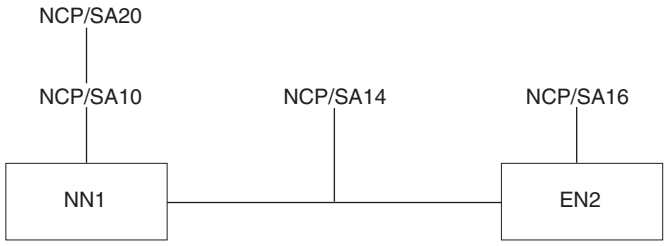


Figure 89. Typical CNN session path

In the above scenario, each of the three NCPs (subarea 10, 14, and 16) are owned by the NN1 composite network node and connect to each of the two hosts (NN1 and EN2). Remote NCP SA20 is connected only to NCP/SA10. When LUs connected to SA20 attempt to log on to applications on EN2, the best route (through the link to NCP/SA10) is not always selected. Figure 90 shows an APPN view of the configuration shown in Figure 89.

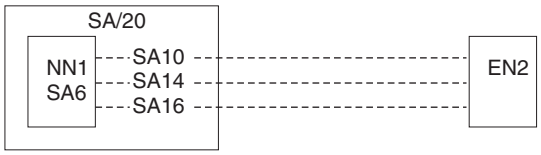


Figure 90. APPN view of CNN session path

In Figure 90, three TGs are show between NN1 and EN2, and the DLU is owned by SA20. If the SA10 TG is selected when VTAM calculates the route, the session is set up. If either 14 or 16 is selected, the resulting path passing through two NCPs is less efficient. This route might fail if there is no VR defined from NCP/SA14 or NCP/SA16 to NCP/SA20.

The SAMAP table provides a way to define to APPN TRS the relationship between the subarea components of the CNN. In this way, the TRS can use that information to determine the best links to use for sessions entering or leaving the CNN. The two subarea numbers coded in the SAMAP table represent two subarea components in the CNN that are connected to each other. The definition of this relationship is most important for CNNs that have remote NCPs.

Figure 91 on page 346 illustrates a configuration where the SAMAP table would assist in determining optimal CNN routes.

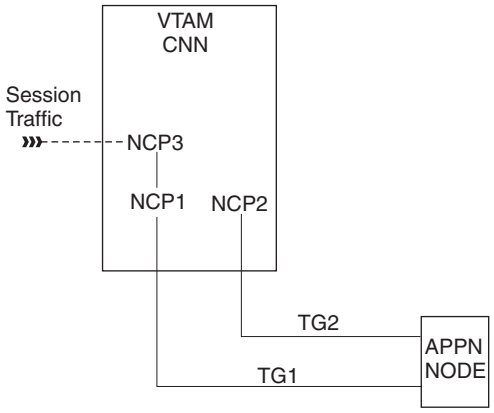


Figure 91. SAMAP session routing concept

In Figure 91, the VTAM CNN owns NCP1, NCP2, and NCP3. TG1 and TG2 have identical route weights based on their TG characteristics and the APPN COS being used. However, NCP3 is connected directly to NCP1 (not NCP2). At session setup, it is better to select TG1 instead of TG2. In this way, session traffic will traverse NCP3–NCP1 within the CNN. If the session was set up with TG2, session traffic would have to traverse the path NCP3–VTAM-NCP2 within the CNN to reach the next APPN node.

You will need to code SA03 MAPSTO SUBAREA=SA01 on the SAMAP table to indicate a direct connection exists between subarea 3 and subarea 1. For sessions to LUs attached to NCP3, TRS will select TG1 based on the SAMAP entry (assuming both TG1 and TG2 are operational and have the exact same route weight). If you do not use the SAMAP entry, TRS randomly selects either TG during session setup.

Note: The MAPSTO statement specifies that connectivity exists between the two subareas and is not directional in nature. In other words, SA03 MAPSTO SUBAREA=SA01 has the exact same meaning as SA01 MAPSTO SUBREA=SA03. Only one of those statements needs to be included in the table.

The SAMAP table reflects the NCP-to-NCP connectivity within a CNN environment. Within a single SAMAP table, it is permissible to specify the NCP-to-NCP connectivity for all of the CNNs within the network, as long as the subarea numbers are all unique across all of the CNNs within the APPN subnetwork.

Net ID mapping

When computing a route to a partner in an SNI-connected network, APPN TRS might not always be able to choose the optimal NCPs. This is because the APPN portion of the route might be calculated prior to the address assignment for the gateway NCP. You can use the SAMAP table to inform TRS that a particular gateway NCP should be assumed when computing routes to a given Net ID.

To provide this information, use the Net ID form of the SAMAP entry:

```
SA03 MAPSTO NETID=NETX
```

This informs TRS that when computing routes to a partner in NETX, the gateway NCP with subarea number 03 will be used. Then, TRS can use that information during TG selection for the CNN.

Note: The SAMAP table on a given host can contain both types of SAMAP entries, and this is often appropriate for VTAMs that own SNI connections.

So, a SAMAP table can contain:

- SAxx MAPSTO SUBAREA=SAyy definitions to map all of the remote NCP to the local NCPs, and the gateway NCPs to local NCPs that have APPN connections.
- SAxx MAPSTO NETID=netid definitions to map the gateway NCP subarea numbers to the Net IDs of their SNI-partners.

Consider the example shown in Figure 92.

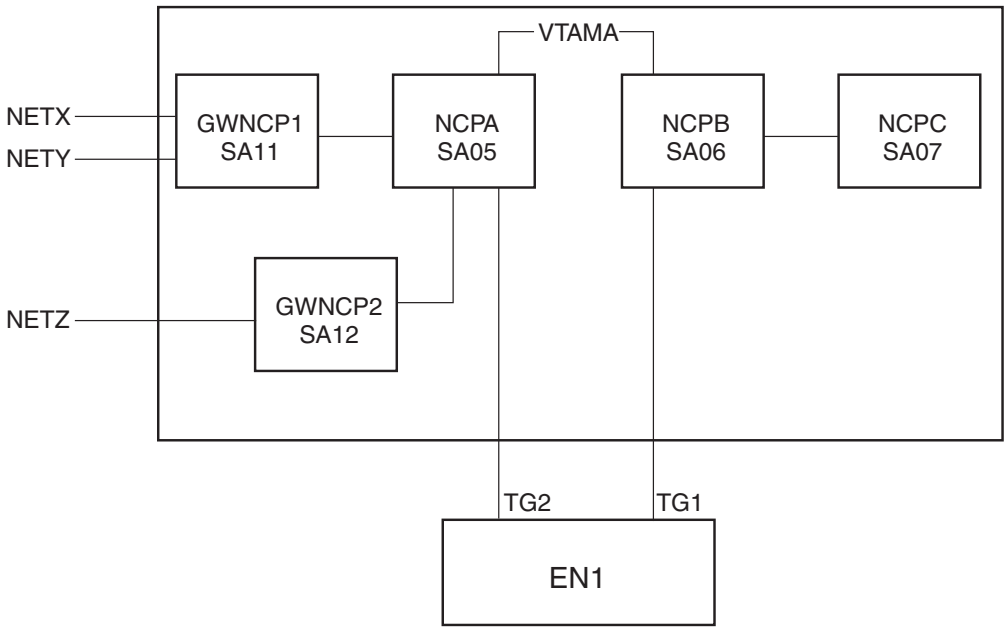


Figure 92. SAMAP example

In this case, the SAMAP table should reflect the following:

- If the partner is in NETX or NETY, GWNCP1 (subarea 11) will be used.
- If the partner is in NETZ, GWNCP2 (subarea 12) will be used.
- NCPA (subarea 05) is connected to GWNCP1 and GWNCP2. This information allows TRS to prefer TG2 (assuming TG1 and TG2 are equally weighted) for sessions going through GWNCP1 and GWNCP2.
- NCPC (subarea 07) is connected to NCPB (subarea 06). This information allows TRS to prefer TG1 for sessions to dependent LUs attached to NCPC.

The corresponding SAMAP table would be as follows:

```
SAMAP1 VBUILD TYPE=SAMAP
SA11   MAPSTO SUBAREA=SA05
SA12   MAPSTO SUBAREA=SA05
SA07   MAPSTO SUBAREA=SA06
SA11   MAPSTO NETID=NETX
SA12   MAPSTO NETID=NETZ
SA11   MAPSTO NETID=NETY
```

The preceding example shows the subarea-to-subarea mappings together in a group, preceding the Net ID mappings. However, this sequence is only illustrative, and it is not required.

Network routing for subarea nodes

You need to design routes between:

- VTAM and any NCPs it activates
- NCPs communicating over a subarea connection (which includes NCPs within a composite network node)
- VTAMs communicating over a subarea connection

Communication between two network accessible units (NAUs) in different subareas requires a definition of at least one route connecting the subareas. This definition includes a physical and logical path between the two.

Physical paths

The physical path between two subarea nodes is an explicit route.

Explicit route (ER)
An explicit route is an ordered set of subarea nodes and transmission groups along a path between communicating subarea nodes, including:

- The endpoint subareas
- Any subareas between the endpoint subareas
- The transmission group used to connect each subarea pair along the route

In connecting one endpoint subarea to another, an explicit route cannot include the same subarea more than once.

Forward explicit route
Explicit routes originating in the host are forward explicit routes and are numbered 0 through 15 (0 through 7 with releases prior to NCP Version 4 Release 3.1 or Version 5 Release 2.1 or if ERLIMIT is set to 8).

Reverse explicit route
Reverse explicit routes terminate in the host and must use the same set of subarea nodes and transmission groups as their corresponding forward explicit route. They are also numbered 0 through 15 (0 through 7 with releases prior to NCP Version 4 Release 3.1 or Version 5 Release 2.1 or if ERLIMIT is set to 8), but they do not have to have the same explicit route number as the corresponding forward explicit route.

Transmission group (TG)
A transmission group consists of a single SDLC link or data channel between two subarea nodes, or of a logical combination of up to eight parallel SDLC links between communication controller subarea nodes. Transmission groups are numbered 1 through 255.

Transmission groups appear as one link to the path control network, even if the transmission group contains parallel SDLC links. Data flows across a specific transmission group during a session, but not across a specific link in a transmission group. The specific link used within the transmission group is determined dynamically depending on which link is available at the time.

Transmission groups can also exist in parallel, creating multiple explicit routes between two subareas. For instance, parallel SDLC links between adjacent communication controller subarea nodes can be divided into multiple transmission groups. A maximum of 16 parallel transmission groups are allowed between adjacent subarea nodes.

Parallel links in one or more transmission groups can be active and carry data simultaneously. Each parallel link is controlled (activated and deactivated) independently of the others.

Routing tables

An entire explicit route is not defined to each subarea node along a path; instead, each subarea node has its own routing table that contains only the information that path control needs to forward data to the next subarea along the route (the adjacent subarea). Given the explicit route number being used and the destination subarea, the table provides the adjacent subarea to which the data is forwarded and the transmission group over which the data is forwarded.

Peripheral link

A peripheral link is the portion of the route between a subarea node and a peripheral node. A peripheral node uses local addresses for routing and requires boundary function assistance from an adjacent subarea node to communicate with a nonadjacent subarea node.

NCP/Token-Ring Interconnection (NTRI) Nondisruptive Route Switching (NDRS)

Token-Ring networks often contain bridges that allow multiple physical paths between two nodes. NCP Version 5 Release 3 and later provides the NDRS function that attempts to find an alternate route through the token ring network. NCP invokes this function automatically when timeout conditions occur and all retries have been exhausted for a connection.

VTAM and NCP Version 7 Release 8 and later provide the MODIFY NCP operator command. This command can be used to force NCP to initiate NDRS for a 3745 token-ring attached subarea connection. This function is useful in order to switch back to the original route (once recovered) after NCP has automatically invoked NDRS to find a backup route. It can also be used at any time to search for a faster route through the token-ring network. NCP Version 7 Release 7 and later will generate a "Token-Ring Path Switch Notification" Generic Alert upon successful completion of NDRS. Refer to *z/OS Communications Server: SNA Operation* for further information.

In the sample network configuration shown in Figure 93 on page 350, HOSTA and peripheral node T1 can communicate over two explicit routes, one in each direction. Both explicit routes have been numbered 0 and include HOSTA, NCPA1, NCPA2, and transmission groups 1 and 14. A data channel assigned to TG1 connects HOSTA with NCPA1. An SDLC link assigned to TG14 connects NCPA1 and NCPA2. Each subarea node contains a routing table to route data between subareas along ER0.

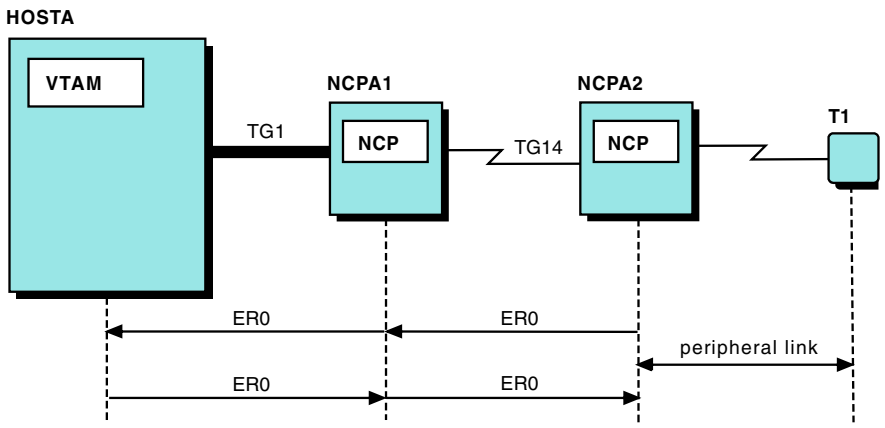


Figure 93. One explicit route in each direction

In the sample network configuration in Figure 94, parallel SDLC links connect NCPA1 and NCPA2. Two different explicit routes between HOSTA and NCPA2 are now available in either direction. All routes take TG1 between HOSTA and NCPA1, but data flows between NCPA1 and NCPA2 over TG14 for ER0 or TG15 for ER1. The routing tables have been updated to reflect the additional explicit routes.

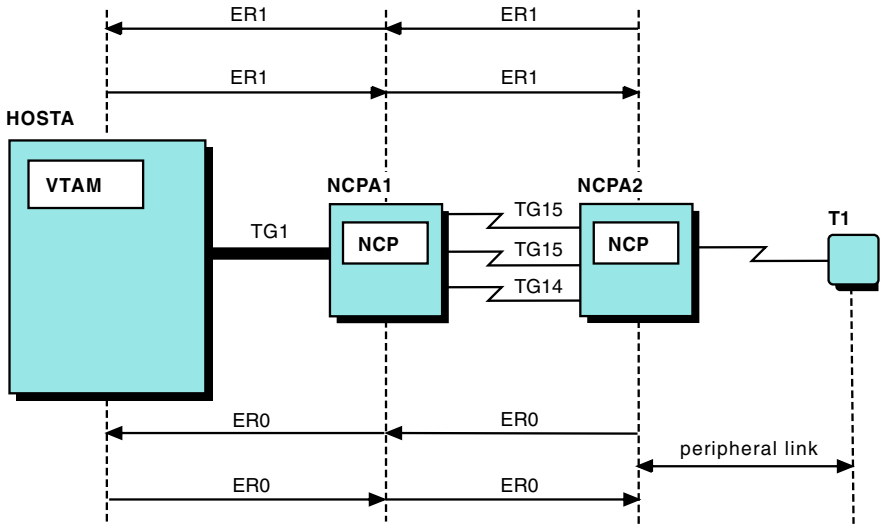


Figure 94. Two explicit routes in each direction

Logical paths

The logical path between two subarea nodes is a virtual route.

Virtual route (VR)

A virtual route is a bidirectional logical connection between two subarea nodes. At least one end of a virtual route must be in a subarea node that activates virtual routes. All hosts can activate virtual routes. NCP Version 4 Release 2 and higher releases can activate virtual routes.

Eight virtual routes numbered 0 to 7 can be defined between two subarea nodes.

One or more virtual routes must be defined for each forward-reverse explicit route pair. A virtual route places a transmission priority on data traffic using the underlying explicit routes.

You can modify the list of virtual routes and their associated transmission priorities using the virtual route selection function of the session management exit routine, or using the virtual route selection exit routine. For information on using these, refer to *z/OS Communications Server: SNA Customization*.

Transmission priority (TP)
The transmission priority identifies the priority of message units flowing over an explicit route during a session. The three possible levels of transmission priority are: 0 (lowest), 1, or 2 (highest).

In general, high-priority messages are routed before low-priority messages. Within a specific transmission priority, messages are routed on a first-in-first-out (FIFO) basis.

The eight virtual routes and three levels of transmission priority provide for the possibility of 24 virtual route and transmission priority pairs between two subarea nodes. In other words, each subarea pair can have virtual routes numbered 0 through 7 between them, and each virtual route can have up to three different transmission priorities.

Route extension
A route extension is a logical connection between a subarea node and a peripheral node. A peripheral node uses local addresses for routing and requires boundary function assistance from an adjacent subarea node to communicate with a nonadjacent subarea node.

As shown in Figure 95 on page 352, a virtual route must be defined for each explicit route pair. For example, between HOSTA and NCPA1, VR0 is associated with ER0 and reverse ER0, and VR1 is associated with ER1 and reverse ER1. Between HOSTA and NCPA2, VR0 is associated with ER0 and reverse ER0, and VR2 is associated with ER1 and reverse ER1. Data flows at a designated transmission priority over a virtual route. In this arrangement, data can flow over any of the virtual routes at transmission priority 0,1, or 2.

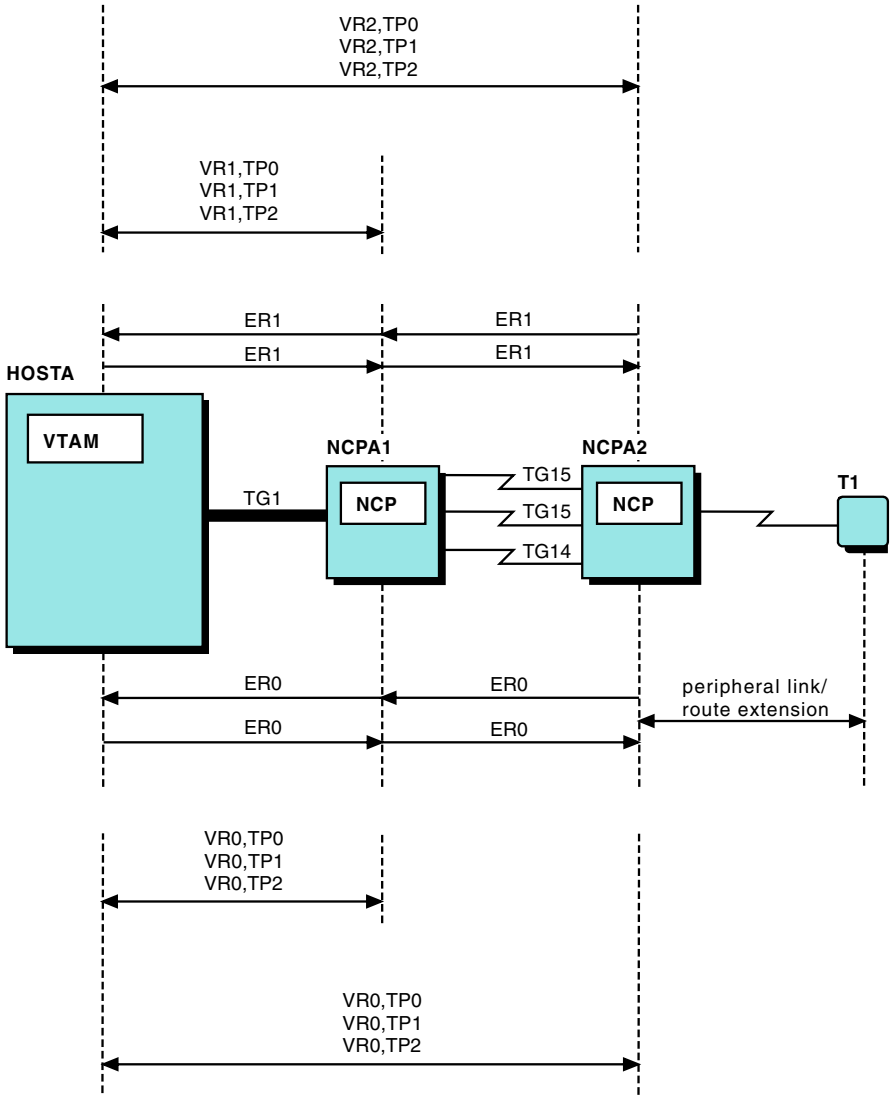


Figure 95. Virtual route and explicit route associations and transmission priority

How session traffic is assigned to a specific route

In a subarea network, data traffic for a session is assigned to a virtual route. The virtual route then maps to an explicit route over which data traffic flows at a designated priority for the session. Following are details of how session traffic is assigned to a specific route.

Logon mode table

Secondary logical units specify a logon mode entry in an LU-LU session-initiation request. Classes of service that you want to be used for an LU-LU session are specified in logon mode table entries using class of service (COS) table names similar to the following:

```
LOGMODE1 MODEENT COS=BATCH,...
LOGMODE2 MODEENT COS=INTERACT,...
⋮
```

If both session endpoints reside in the same subarea, the Class of Service requested by the secondary logical unit is ignored because data flowing

between the two endpoints never enters the network. Otherwise, VTAM obtains the requested Class of Service from the specified MODEENT macroinstruction in the logon mode table associated with the secondary logical unit. VTAM then uses the requested Class of Service to map to the Class of Service table.

Notes:

1. If the specified logon mode entry does not exist in the logon mode table associated with the secondary logical unit, the default logon mode (ISTCOSDF) is used if it is found in the table and its use is allowed as determined by the ISTCOSDF start option.
2. If no logon mode table has been designated for a logical unit, VTAM uses the IBM-supplied logon mode table, ISTINCLM. For details on ISTINCLM, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.
3. If a logon mode table entry specifies a Class of Service name that is not defined in the Class of Service table, VTAM rejects the session-initiation request.

Class of service (COS) table

A class of service specifies a set of performance characteristics used in routing data between two subareas. Different classes of service can be defined in a network for different types of users and data. For example, interactive sessions can be assigned to a fast route, batch jobs to a high-bandwidth route, and sessions involving the transmission of sensitive data to a secure route. At least one class of service can also be specified for SSCP sessions.

Different classes of service are specified in a COS table. Each class of service is specified by a symbolic name (for example, INTERACT for interactive sessions or BATCH for batch sessions). The COS table lists, in order of preference, the virtual route and transmission priority pairs that are to be used for each named class of service. For example:

```
INTERACT COS VR=(1,1),(0,1),...
BATCH    COS VR=(0,0),(1,0),...
⋮
```

When the logon mode table entry for an LU-LU session does not contain a COS name, VTAM uses the unnamed COS entry. The unnamed COS entry is one of two special entries that you should include when you code a COS table. The other is ISTVTCOS.

Unnamed entry

An unnamed entry (the entry name consists of eight blanks) can be placed in the COS table and is used when either of the following is true:

- No class of service name is obtained from the logon mode entry for an LU-LU session.
- No ISTVTCOS entry exists in the COS table, and an SSCP session has been requested.

If the unnamed class of service entry is requested but has not been included in the COS table, VTAM uses its own Class of Service defaults.

ISTVTCOS

ISTVTCOS is the special COS entry you use to specify the routes you want to use for SSCP sessions (SSCP-SSCP, SSCP-PU, and SSCP-LU).

VTAM looks for and uses the ISTVTCOS entry any time there is an SSCP session request for another subarea. If the ISTVTCOS entry is not present, the unnamed COS entry is used.

Note: Downstream PUs and associated LUs are assigned the same VR as the boundary node for SSCP-PU and SSCP-LU sessions. For example, when an SSCP-PU session is established for a PU connected to an NCP, the VR that was chosen for the NCP SSCP-PU session will also be used for the downstream SSCP-PU session.

You need not define a COS table if the only COS names to be used are ISTVTCOS and the unnamed class of service; VTAM uses its own class of service defaults.

IBM-specified class of service defaults

VTAM has a default list of virtual route and transmission priority pairs that is used when either of the following is true:

- No COS table has been created.
- A COS table has been created, but:
  - No class of service name has been obtained from the logon mode entry associated with an LU-LU session request, and no unnamed entry has been included in the COS table.
  - An SSCP session has been requested, and no ISTVTCOS entry or unnamed entry has been found in the COS table.

The defaults consist of all 24 possible virtual route and transmission priority pairs in the following order:

```
VR0,TP0;  VR1,TP0;  VR2,TP0;  ...  VR7,TP0;
VR0,TP1;  VR1,TP1;  VR2,TP1;  ...  VR7,TP1;
VR0,TP2;  VR1,TP2;  VR2,TP2;  ...  VR7,TP2;
```

This default list might not be the best for your needs, and its use might not provide optimal network performance. You can replace the default list by creating a COS table with an unnamed COS entry containing the new list. This new default list is then used if no COS entry is named in an LU-LU session request or if an SSCP session is requested and no ISTVTCOS entry exists in the COS table.

Substituting class of service parameters

Usually, if VTAM receives a user session request using a COS name that is unknown to this VTAM, the session is rejected. If you want the user session to be established despite the unknown COS name, you can enable VTAM to accept the unknown COS name and use substitute parameters that are coded on another COS name. To do this, code SUBSTUT=YES on the COS name that will provide the substitute parameters.

For example, in Figure 96 on page 355 assume that you are trying to set up a session between APPLA and APPLB, and APPLB is the primary LU. VTAMA specifies COSA as the COS to be used for the session. VTAMB does not have COSA coded, so VTAMB uses the parameters from COSB to set up the session because COSB has SUBSTUT=YES coded. If you display the session from either VTAMA or VTAMB, the display will indicate that COSA is being used, even though the parameters for COSB are being used.

The substitution is transparent to VTAMA. However, you can monitor the substitution in VTAMB and reject session establishments that you do not want to use the substitute parameters by coding a session management exit routine. For information about coding the exit routine, refer to *z/OS Communications Server: SNA Customization*.

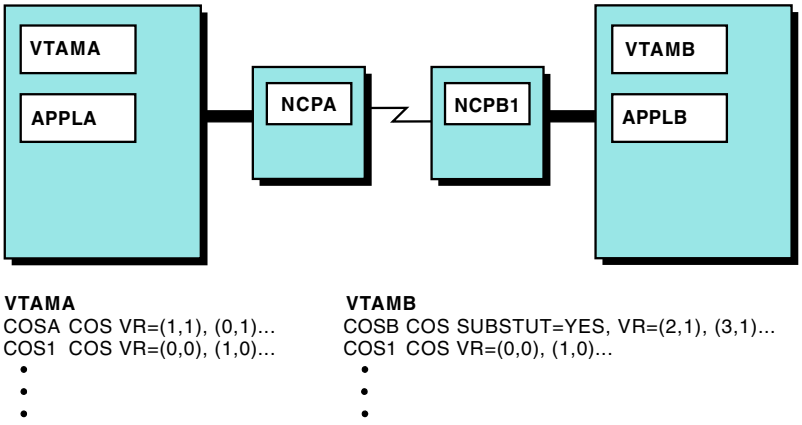
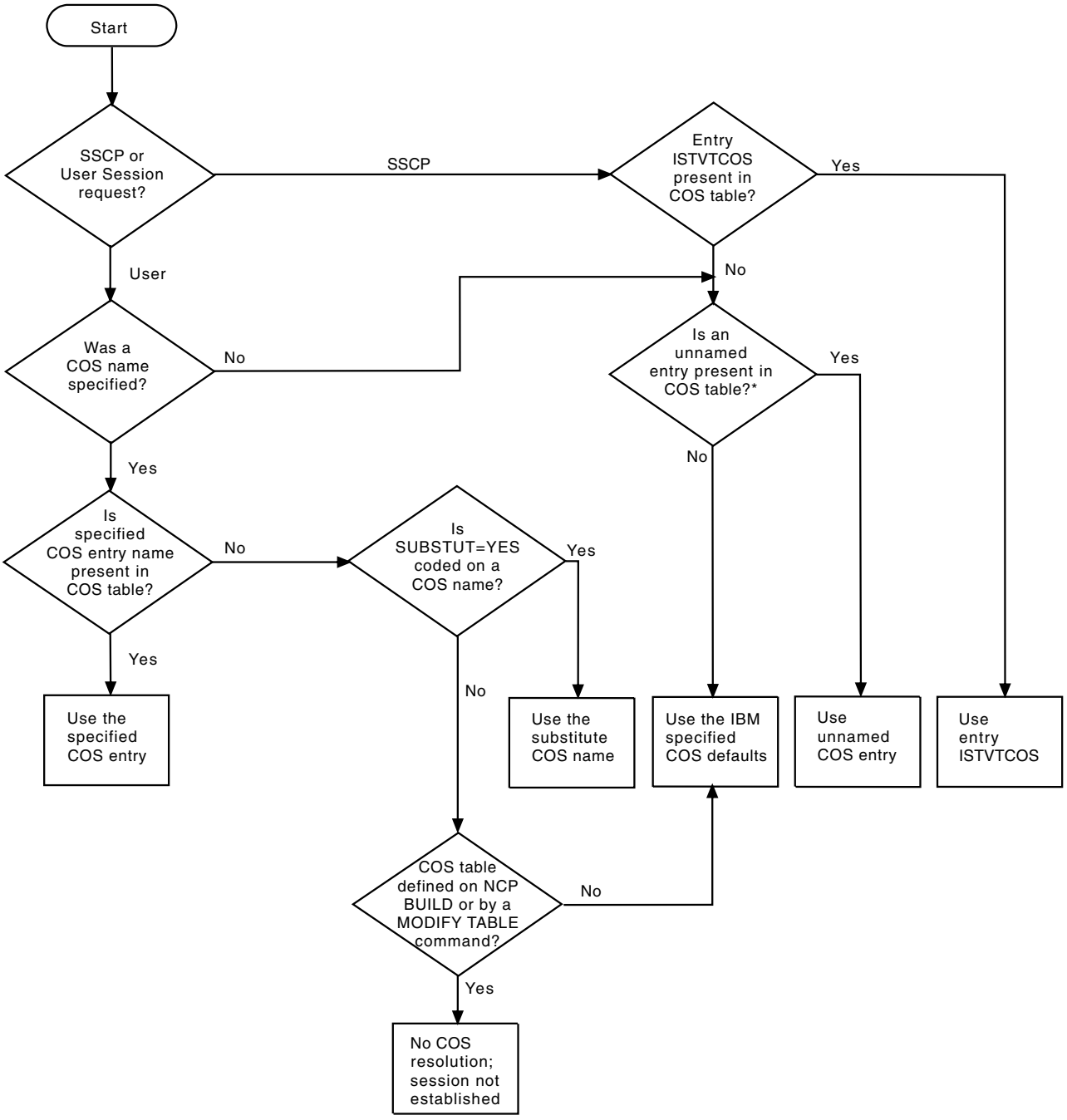


Figure 96. Class of Service substitution

Substitute class of service parameters are not passed between VTAMs. If a request is forwarded to an adjacent SSCP, the COS name that was originally requested is the name passed. The substitute COS name is not passed.

The class of service hierarchy in Figure 97 on page 356 summarizes the steps VTAM takes to determine the appropriate COS table entry to use for SSCP and user sessions (or to determine to use the IBM-specified class of service defaults).

The SSCP selects the first virtual route and transmission priority pair listed for the requested class of service (or the first pair listed in the IBM-specified class of service defaults) that has been defined to the subarea of the session partner. An attempt is made to activate the virtual route.



*This is also referred to as a "blank entry."

Figure 97. Class of Service hierarchy

The virtual route maps to an explicit route over which data traffic flows at the designated priority for the session. If all the network resources along the explicit route are operative, the virtual route is activated and available for session use.

If the explicit route is not operative, the next virtual route and transmission priority pair for the Class of Service requested is selected, and an attempt is made to activate that virtual route. If all virtual route and transmission priority pairs for the Class of Service requested map to inoperative explicit routes, the request to

establish a session is either queued or rejected. Requests for SSCP sessions are queued (if queuing is allowed), but all others are rejected.

How to plan routes in your network

Designing routes through a network ranges in complexity depending on the number of subareas, connections between subareas, and types of user sessions that your network supports. Also, decisions must be made to achieve some combination of the following objectives, which involve the following compromises:

- Maximize quantity of data transmitted
- Maximize data security
- Maximize route availability
- Minimize transmission time
- Minimize cost
- Minimize data loss or the necessity to retransmit data
- Minimize traffic congestion

NETDA/2 is a tool that can help plan for routes in your network and other network design tasks, such as:

- Designing a network
- Producing reports on designs so that you can evaluate them on performance
- Generating route statements

The following steps can also help you make the decisions necessary to plan the routes in your network:

1. Determine the Classes of Service that will exist in your network. The following questions are the types of questions you need to answer in determining the classes of service you need.
   - Will you have one Class of Service for all interactive sessions?
   - Do you want a Class of Service that provides quick response times suitable for high-priority interactive sessions and a Class of Service that provides response times suitable for low-priority interactive sessions?
   - Do you want a Class of Service that provides routes that have the best availability?
   - Do you want a Class of Service that is suitable for batch processing?
   - Do you want a Class of Service that is suitable for high-security transmissions?
2. Identify the possible explicit routes available in your network, including the possible ways parallel links between communication controller subarea nodes could be divided among transmission groups.

   Dividing parallel links into multiple transmission groups creates multiple explicit routes between two nodes. Multiple explicit routes between two nodes provide the opportunity to physically separate the data traffic for different Classes of Service in your network.

   Combining parallel links between communication controller subarea nodes into a single transmission group increases the probability that explicit routes using the transmission group will be operational. A transmission group that consists of parallel links will still be available even if one of the links fails, and data traffic can continue to flow on the remaining links. Only links that have similar transmission characteristics (for example, high-transmission speed) are normally placed in the same transmission group.
3. Analyze the physical characteristics of each identifiable explicit route, such as:
   - Length
   - Transmission time

- Lowest link capacity in the route
- Lowest transmission group capacity in the route
- Lowest security level in the route
- Operational probability

4. Match the Classes of Service you selected to explicit routes having appropriate characteristics. Remember to consider the characteristics of a route between each individual subarea pair. For example, for a Class of Service requiring a fast route, choose the fastest route between each subarea pair along the route.

   Also, be sure to consider alternate routes for each Class of Service. If a subarea node or transmission group fails, all routes passing through that node or transmission group become inoperative and existing sessions are disrupted. An alternate route is a route that is available if the route being used becomes inoperable. If you want to ensure that you always have a route available, you should choose alternate routes that do not pass through the same subarea nodes or transmission groups as the routes that they back up.
5. Identify a virtual route for each selected explicit route.
6. Create an ordered set of virtual route and transmission priority pairs for each Class of Service.

   In creating this ordered set, you assign a transmission priority to a virtual route which, together with the physical characteristics of the explicit route, determines the suitability of a particular virtual route for a designated Class of Service. Remember that multiple sessions can use the same virtual route (with the same or a different Class of Service).

   For example, for a Class of Service requiring faster data transmission and predictable response times (a Class of Service suitable for interactive sessions), you can list virtual routes assigned to explicit routes that have fewer physical elements before listing virtual routes assigned to longer explicit routes. The shorter routes would be used first, and the longer routes would be used only for backup purposes.

   However, you might have interactive users with differing Class of Service requirements. The virtual routes for each of these Classes of Service could be identical to those above, and the Class of Service would be realized through the transmission priority. For the interactive users requiring faster data transmission and predictable response times (for example, an inquiry-response Class of Service), you would list virtual routes having a higher transmission priority than the virtual routes for interactive users where slower data flow is acceptable (for example, a data-collection Class of Service). In other words, sessions for different kinds of applications can be in progress over a given explicit route, and transmission priorities determine the order of transmission.

   Suppose you have two Classes of Service for low priority and higher priority batch sessions. The higher-priority batch sessions would be assigned to virtual routes with a higher transmission priority than the virtual routes for the lower-priority batch sessions. For both batch Classes of Service, you would probably list high-bandwidth explicit routes before listing the explicit routes used for interactive sessions. The shorter routes used for interactive sessions could be used as alternate routes for batch sessions. The batch sessions would likely be assigned a transmission priority lower than the transmission priority of the interactive sessions using the same virtual route.

   Note: In migration environments, you should include VR 0, TP 0 in each COS table entry as an identifier for default routes containing nonextended addressing nodes.

7. For details on coding a COS table to hold the ordered virtual route and transmission priority pairs for each class of service, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.
8. As discussed in "How session traffic is assigned to a specific route" on page 352, VTAM processes the virtual route and transmission priority pairs in a COS table entry sequentially, starting with the first entry. If you require a more precise selection algorithm, use the session management exit routine or virtual route selection exit routine to change the virtual route selection process for LU-LU sessions.

   For example, if you provide a virtual route selection exit routine, your routine receives the ordered virtual route list as a parameter from VTAM. Your routine can modify the list. VTAM then uses your reordered list of virtual routes to select a virtual route for the session.

   If you have provided a virtual route selection exit routine, VTAM calls it whenever a session between a primary logical unit in the VTAM subarea and a logical unit in another subarea is about to be established. The virtual route selection exit routine is not called if both logical units are in the same VTAM subarea.

   For more information about coding a virtual route selection exit routine or using the session management exit routine to change the virtual route selection process for LU-LU sessions, refer to *z/OS Communications Server: SNA Customization*.
9. For each subarea node, use PATH definition statements to create the path definitions necessary for defining the explicit routes and virtual routes that exist between the subarea node and other subareas. For example, the PATH statements needed in HOSTA of Figure 95 on page 352 are:

```
PATH1   PATH DESTSA=NCPA1,
             ER0=(NCPA1,1),
             ER1=(NCPA1,1),
             VR0=0,
             VR1=1
PATH2   PATH DESTSA=NCPA2,
             ER0=(NCPA1,1),
             ER1=(NCPA1,1),
             VR0=0,
             VR2=1
```

   From path definitions like these, VTAM builds a routing table for a subarea node similar to the table for HOSTA shown in Figure 94 on page 350. In creating your path definitions, list virtual routes only in those subareas that are going to activate them. For NCP subareas that cannot activate virtual routes, the virtual routes ending in those subareas are dynamically defined to those subareas as part of route activation.

   Coding paths for a multiple-domain environment is similar to coding paths in a single-domain environment. Communication between domains requires paths to be defined between VTAM subareas. The PATH definition statements that are used to define routes between VTAM and NCP subareas in a network are also used to define routes between the VTAM domains. For each subarea, the PATH definition statements define the explicit and virtual routes that connect the various VTAM domains.

   VTAM can function as an intermediate routing node. VTAM receives the data from the adjacent subarea and transmits it to the next subarea on the network route as defined by the PATH definition statements. The IRNSTRGE start option is used to define the maximum size of the virtual area in VTAM storage that can save host intermediate routing node (IRN) transmissions. If IRNSTRGE=0 (the default) is used, the amount of storage is unlimited.

For further details on PATH statements or the IRNSTRGE start option, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

10. One or more path definition sets must be activated for VTAM to know how to route cross-subarea session traffic. A path definition set is activated with a VARY ACT command. When a file containing a set of PATH definition statements is activated, it defines explicit routes and virtual routes to VTAM. At that point, the routes are defined but not yet active.

    You can activate additional path definition sets to add or modify route definitions when necessary. You can define new routes and redefine or delete inactive routes. Existing route definitions not affected by a given path definition set activation are unmodified and remain in effect. VTAM rejects any attempt to modify the definition of any previously defined route that is not in the inactive state.

    A path definition set is not a major node and cannot be deactivated.

11. In addition to the activation of path definitions, all physical elements included in an explicit route (subarea nodes and transmission groups) must be activated before an explicit route is ready for activation.

    An explicit route is operational when physical connectivity between the endpoints is established. Path information is exchanged between each subarea when the connection between the subareas is activated. A virtual route becomes available for activation after its corresponding explicit route is active.

    VTAM automatically attempts to activate explicit and virtual routes when they are first needed for a session. A virtual route remains active until all sessions assigned to it have ended, at which time it is automatically deactivated. The deactivation of a virtual route has no effect on the status of its associated explicit route. Once active, an explicit route is never deactivated, but it can become inoperative due to hardware failure or deactivation of physical elements within the route.

12. You can use the DISPLAY ROUTE command to display information about the explicit routes and virtual routes between the host subarea and a given destination subarea. You can also use this command to test whether one or more explicit routes to a given destination are operative (see "Displaying and testing routes" on page 617).

13. Remember that the requested Class of Service for a session is obtained from the logon mode table associated with the secondary LU. For details on creating a logon mode table and associating LUs with your table, refer to *z/OS Communications Server: SNA Resource Definition Reference*. If you do not create a logon mode table and also make the association between an LU and that table, VTAM obtains the LU requested Class of Service from the IBM-supplied logon mode table, ISTINCLM.

How VTAM handles network and subarea addressing

For VTAM to control routing in a subarea network or within a composite network node, VTAM must know the location of its resources, such as logical units, physical units, and other SSCPs. VTAM uses element addresses, in conjunction with the subarea address, to identify the location of resources (also known as network addressable units). The subarea address indicates in which subarea the resource is located; the element address indicates the unique address within the subarea.

Minor nodes, such as an application program or logical unit, require at least one element address. Some minor nodes, such as local non-SNA devices and

application programs that use parallel sessions, require more than one element address. This can increase the number of element addresses that is used in a subarea.

The different types of addressing structures are:

Preextended network addressing
Used by releases prior to VTAM Version 3 Release 1. Enables you to define 255 subareas and 254 elements.

Extended network addressing (ENA)
Used by VTAM Version 3 Release 1 and subsequent releases prior to VTAM Version 3 Release 2. ENA supports addressing to 255 subareas and extends element addressing to 32 768 elements. Beginning with VTAM Version 4 Release 1, host element addressing was extended to 65 535 elements.

Extended subarea addressing
Used by VTAM Version 3 Release 2 (with compatibility PTF) and subsequent releases. Extended subarea addressing increased the size of the subarea address to 65 535 and the number of explicit routes for each destination subarea to 16.

Note: Applications are assigned a low-order primary LU address upon activation. This low-order address may be used in cases where a high-order address could have been used.

Compatibility between nonextended and extended network addressing nodes

You can use the MAXSUBA start option or the MAXSUBA operand in NCP to enable subareas with different addressing structures to communicate. For a network containing nonextended addressing nodes, the MAXSUBA start option specifies the highest subarea value that can be used throughout the network. A MAXSUBA of 63, for example, defines a network with up to 63 subareas and 1024 elements in each subarea. Code the MAXSUBA start option in the start option list if you want VTAM to communicate with nonextended addressing nodes, or if the MAXSUBA operand is coded in an NCP that VTAM will communicate with.

Extended network addressing is a network addressing structure that increases the size of the subarea address up to 255 in conjunction with up to 32 768 elements. VTAM extends host element addressing to 65 535 elements.

Nonextended addressing nodes can continue to use this structure. They can, however, participate in an extended network addressing network only when the following two conditions are met:

- A compatibility program temporary fix (PTF) must be installed on all nonextended addressing nodes that communicate with extended network addressing nodes.
- All nodes that communicate with nonextended addressing nodes must define a MAXSUBA. Extended network addressing nodes need this information to decode 16-bit network addresses. The value specified for MAXSUBA must be the same at all nodes.

In addition, the following restrictions apply when extended and nonextended addressing nodes coexist in the same network:

- VTAMs that are nonextended addressing nodes cannot communicate with any logical unit whose element address exceeds the maximum element number defined by the subarea/element split.
- VTAMs that are nonextended addressing nodes cannot communicate with any node whose subarea is greater than the MAXSUBA.

Figure 98 illustrates an element address incompatibility. Domain 1 contains nodes that use extended network addressing. Domain 2 contains nodes that use nonextended addressing node addresses. All nodes have defined a MAXSUBA of 63. Although all of the subareas are within the MAXSUBA, some of the element addresses exceed the maximum, which is 1023. Following are two address constraints on this network:

- In Domain 1, the terminals T1 and T3 can communicate with any subarea because their element addresses (100) are less than 1024. T2 and T4, however, cannot communicate with host A40M because their element addresses (1500 and 2000) are greater than 1023.
- In Domain 2, T5 and T6 can communicate with either host. However, they cannot communicate with any application program in host A60M whose element address is greater than 1023.

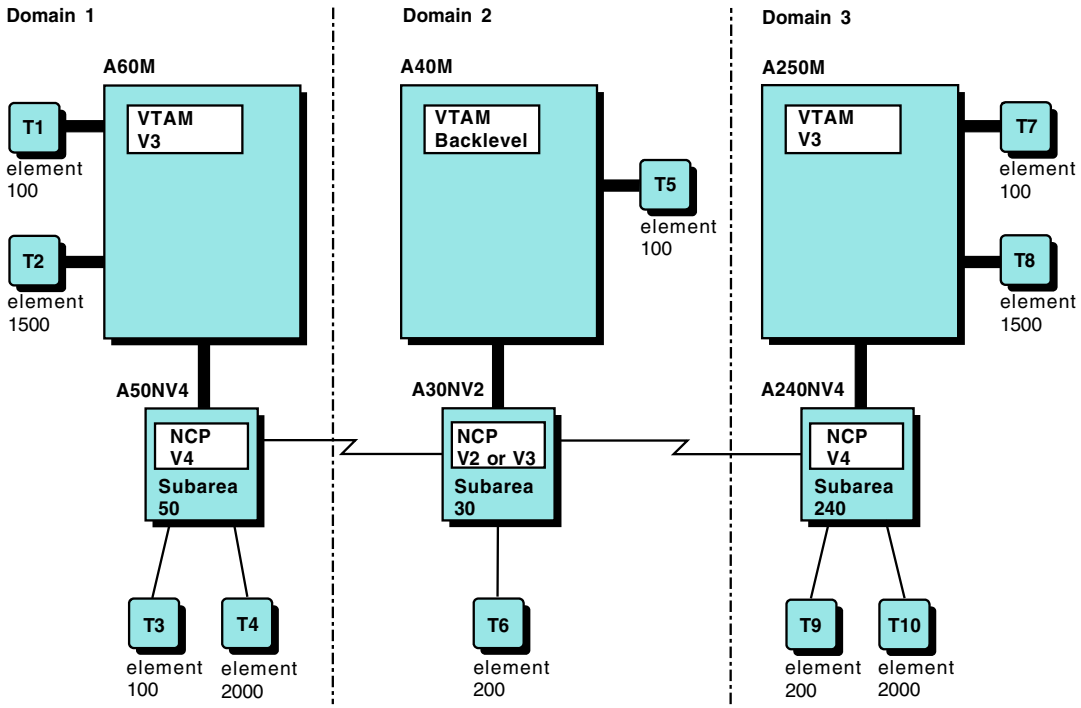


Figure 98. Element and subarea address incompatibility in multiple-domain environment

These limitations can be overcome through SNA network interconnection (SNI), which can be used to place nonextended addressing nodes and extended network addressing nodes into separate networks. See Chapter 20, "Connecting multiple subarea networks," on page 551 for information about implementing SNI.

Although it is not recommended, a network containing nonextended addressing nodes can have subareas whose subarea number is greater than the MAXSUBA. The nonextended addressing nodes are unable to communicate with these subareas. In this situation, the following constraints apply:

- A nonextended addressing node VTAM cannot appear in any route whose endpoints have a subarea address greater than MAXSUBA.
- A nonextended addressing node VTAM cannot be adjacent to a node whose subarea is greater than MAXSUBA.
- A nonextended addressing node NCP can be adjacent to a node whose subarea is greater than MAXSUBA, but only if the NCP is used as an intermediate routing node (IRN). This requires a PTF.

Table 38 and Table 39 show the adjacent and endpoint node subarea requirements.

Table 38. Adjacent node subarea requirements for multiple-domain environment

	Pre-ENA VTAM	ENA VTAM subarea ≤ MAXSUBA	ENA VTAM subarea > MAXSUBA
Pre-ENA NCP	Yes	Yes	Yes[3]
ENA VTAM Subarea < MAXSUBA	Yes	Yes	Yes
ENA VTAM Subarea > MAXSUBA	No	Yes	Yes

Notes:

1. Yes: These two nodes can be endpoints.
2. No: These two nodes cannot be endpoints.
3. Pre-ENA is used as an IRN only.

Table 39. Endpoint node subarea requirements for multiple-domain environment

	Pre-ENA VTAM	ENA VTAM subarea ≤ MAXSUBA	ENA VTAM subarea > MAXSUBA
Pre-ENA NCP	Yes	Yes	No
ENA VTAM Subarea < MAXSUBA	Yes	Yes	Yes
ENA VTAM Subarea > MAXSUBA	No	Yes	Yes

Notes:

1. Yes: These two nodes can be endpoints.
2. No: These two nodes cannot be endpoints.

Figure 98 on page 362 illustrates these constraints. The third domain contains extended network addressing nodes whose subarea addresses are greater than MAXSUBA.

The following communication restrictions apply to the network in Figure 98 on page 362:

- None of the Domain 2 terminals can have a session with host A250M. Conversely, the Domain 3 terminals cannot have a session with host A40M.
- All of the terminals in Domain 3 can communicate with any application program in Domain 1, and all of the terminals in Domain 1 can communicate with any application program in Domain 3.
- NCP A30NV2 is a nonextended addressing node, but can still act as an IRN between Domains 1 and 3. However, the NCP terminal, T6, cannot communicate with Domain 3.

As with the element address problem, subarea address incompatibilities can be avoided by using SNI to place nodes in separate networks. See Chapter 20, “Connecting multiple subarea networks,” on page 551 for information about implementing SNI.

Compatibility between extended network and extended subarea addressing nodes

Whereas extended network addressing supports a maximum subarea address of 255, extended subarea addressing allows you to specify subarea addresses higher than 255. Subarea addresses can be as high as 65 535.

However, defining networks with large numbers of subareas can increase the storage requirements for VTAM and NCP. To control the maximum number of subareas in a network, the MXSUBNUM start option and the NCP operand SALIMIT allow you to specify the maximum subarea size of the network. You can define VTAMs and NCPs in the network with varying values for the maximum subarea limit. However, the actual subarea addresses used must be within a common address range for the subarea nodes to communicate. Processors and communication controllers with levels of VTAM and NCP that do not support extended subarea addressing can exist in the same network with other processors and communication controllers that do. For subarea node communication to be supported, the actual subarea addresses used must be within 255 or the MAXSUBA value, whichever is smaller.

Migration can be simplified by installing levels of software that support extended addressing on all subareas in the network, with the maximum subarea limit specified to accommodate planned growth, before adding any subareas with a higher subarea address value. Failure to do so results in subareas that no longer have connectivity to other parts of the network.

The maximum subarea limit should, in general, be constant throughout the network. If the subarea limit cannot be constant, you can try different values with no loss of connectivity until a subarea address is used that is higher than the lowest subarea limit in the network. When moving to a lower subarea limit value, you should lower all subarea numbers before changing the subarea limit value.

The following subarea addressing restrictions apply to a multiple-domain network unless you use SNA network interconnection (SNI) to circumvent the addressing constraints:

- Adjacent subareas can support different maximum subarea values and still communicate provided their actual subarea addresses fall within their common range. For example, a nonextended subarea addressing VTAM can communicate with an extended subarea addressing NCP that supports an SALIMIT value greater than 255 if the NCP subarea address is no greater than 255.
- A subarea whose address is above the addressing supported by other adjacent subareas on a route cannot communicate to or through those subareas.
- Subareas can communicate through other subareas with an address higher than they support provided there is an intermediate subarea with an address acceptable to both. For example, a nonextended subarea addressing VTAM could communicate through an NCP with a subarea address of 300. There would, however, have to be an intermediate VTAM or NCP supporting extended subarea addressing with an address no greater than 255 and an SALIMIT of at least 300.
- An SSCP cannot activate any subareas with addresses greater than its addressing capabilities.

- For an extended subarea addressing NCP to successfully send a CONTACT request unit to an adjacent extended subarea addressing NCP with an address above 255, the link station must be owned by an extended subarea addressing VTAM.

With extended subarea addressing, VTAM and NCP support the use of 16 explicit routes. You can define up to 16 explicit routes for each destination subarea. However, for an explicit route to have a number greater than 7, all subareas on the route must support extended subarea addressing, and all NCPs must have ERLIMIT=16 coded on the BUILD definition statement.

Virtual route pacing

The term virtual route pacing applies to the monitoring and flow control mechanisms that are used for virtual routes. On a network-wide basis, subarea nodes (VTAM and NCP) continuously monitor the amount of traffic in the network. If congestion occurs, these subarea nodes can limit the amount of data they send over the affected virtual routes until the congestion clears. The monitoring of data flow and limiting of data occurs automatically, requiring no action by users or domain operators.

Activation of a virtual route includes initializing the pacing count. This count, the virtual route window size, indicates the number of path information units (PIUs) that are allowed to flow on a virtual route before the subarea node receiving the PIUs authorizes the sending of more data. The set of PIUs that can be sent is called a window. This value fluctuates between a minimum and a maximum value depending on network congestion. (Network congestion is determined by how many messages are queued.) You can affect how VTAM does virtual route pacing through the PATH definition statement or the VR pacing window size calculation module. The minimum and maximum values can be specified on the NCP PATH definition statements and the VTAM PATH definition statements that define the virtual routes.

Degrees of congestion

The network detects two degrees of congestion: minor and severe. Minor congestion is detected when the amount of data queued for a specific transmission priority to a particular transmission group exceeds a user-specified value. Severe congestion is detected when the aggregate amount of data for all transmission priorities exceeds the total user-specified value for a transmission group. At the first, less severe level, the number of PIUs allowed to be sent in a window is decreased by one until congestion clears. At the severe level, the window size is immediately decreased to its minimum value. These two levels can be specified with NCP transmission group flow control threshold parameters. These threshold parameters can be coded on the ER operand on the NCP PATH definition statement.

VTAM detects only severe congestion. The reset window indicator is set in the PIU if VTAM cannot immediately get an IO buffer, either due to a shortage or because XPANPT is not high enough to allow expansion to complete before all the remaining buffers are allocated.

Local flow control and severe congestion

Virtual route pacing is supplemented by a local flow control function that limits the amount of data that users within a subarea can present to the network for transmission. Figure 99 on page 366 illustrates local flow control.

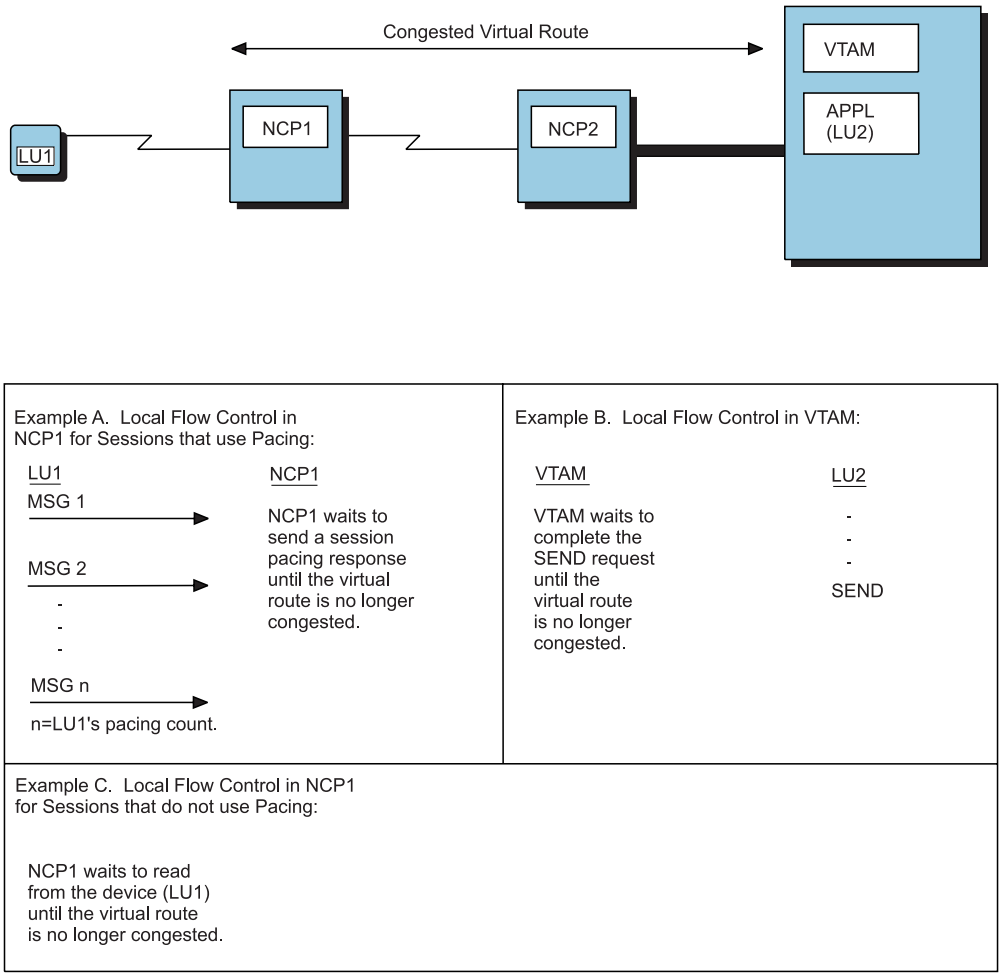


Figure 99. Local flow control

Suppose, for example, that the virtual route between NCP1 and VTAM is congested and that VTAM has been notified of the congestion. If NCP1 does not give VTAM permission to send another window of data, the virtual route is held. VTAM then keeps the windows of data in IOBUF buffer space until the data can be passed.

If LU1 is an SNA device and session pacing is being used on the LU-LU session, NCP1 withholds the session pacing response going to LU1 until the virtual route is no longer congested. (See Example A in Figure 99.) Because virtual route pacing controls only normal-flow requests, LU1 can still send responses to previously received requests as well as expedited flow requests. Because those RUs can also congest the virtual route, NCP1 holds them until the congestion clears.

If LU1 is a non-SNA device (or an SNA device that is not using session pacing on the LU-LU session), NCP1 does not read any input from the device until the virtual route is no longer congested. (See Example C in Figure 99.)

When the application program (LU2) in Host1 sends data to a device attached to NCP1, VTAM holds the data in IOBUF buffer pool space until it is informed that the virtual route is no longer congested. After the virtual route is blocked and VTAM holds the windows of data, VTAM queues the application program SEND request and does not read the data from the user buffer. (See Example B in

Figure 99 on page 366.) The direction of data flow is paced independently. One direction might be congested while the other direction is not. Figure 99 on page 366 shows a virtual route that is congested in both directions.

Note: If VTAM determines that sessions with one of its local devices are contributing to congestion over a route, it shuts off data transmission from the device. Therefore, if there are a number of SNA devices attached to a channel-attached cluster controller, sessions for all the devices are shut down, regardless of whether their sessions are being conducted over the congested route or another route. To avoid this, you might want to specify inbound session pacing for all channel-attached SNA devices so that they are each paced separately. See "Session-level pacing" on page 307.

Window sizing

The IBM-supplied algorithm for window size calculation is designed to work with the route pacing algorithm used in the network. It is appropriate for most installations and configurations. VTAM calculates the minimum and maximum sizes of virtual route pacing windows based on the link protocol and the explicit route length (that is, the number of transmission groups in the explicit route used by the virtual route). While the virtual route is being used to transmit data, adjacent subarea nodes on the route automatically adjust the window sizes within the minimum and maximum limits according to traffic conditions along the route.

However, after tuning VTAM and analyzing traffic patterns and resource capabilities, you might want to choose your own bounds or code a replacement routine that sets the window sizes to different values than the ones supplied by IBM. This exit routine is appropriate for systems where the number of resources could vary considerably from one day to the next. For example, you might find one or more resources whose capacities are not consistently used. This might warrant increasing the window sizes. Decreasing the window sizes is less likely to be useful because network flow control protocols are designed to prevent congestion, and setting window sizes too small could reduce traffic flow considerably.

If you use the Information Management System (IMS), you might want to code a replacement virtual route-pacing window-size calculation exit routine to reduce the number of virtual route pacing responses that VTAM processes for every IMS transaction. For more information about coding the exit routine for IMS, refer to *z/OS Communications Server: SNA Customization*.

You can also specify default minimum and maximum window sizes on the VRPWS*nn* operand of the PATH definition statement when defining a virtual route. This is the simplest way to choose the default and is sufficient for most needs. If you do not specify VRPWS*nn*, VTAM uses the default algorithm described previously.

The virtual route-pacing window-size calculation exit routine can be used in addition to VRPWS*nn* operands (coded on the VTAM and the NCP PATH definition statements) to alter the VTAM virtual route pacing window sizes. This exit routine is used to specify the bounds for virtual route pacing windows. A virtual route pacing window represents the quantity of path information units (PIUs) that can be transmitted on a virtual route before a virtual route pacing response is received. This response indicates that the virtual route receiver is ready to receive more PIUs on the route. The exit routine is called when a virtual route is activated. It returns the minimum and maximum values for the window of the virtual route.

Parallel sessions using parallel transmission groups

If the application program supports parallel sessions, each session can use a different class-of-service entry and can map to a route that uses a different channel path.

For example, in Figure 100, because the type 2.1 peripheral node is connected to an NCP with parallel channel-link transmission groups (TG1 and TG2), the independent logical unit (APPL21) can establish parallel sessions with CICS in the host, and each session can use a different route. However, each session must use a different Class of Service entry (such as COS1 and COS2). These different Class of Service entries map to different virtual routes (such as VR1 and VR2) using the same or different transmission priorities (in this case, the same, TP1). As shown in Figure 100, the different virtual routes map to the explicit routes for each session.

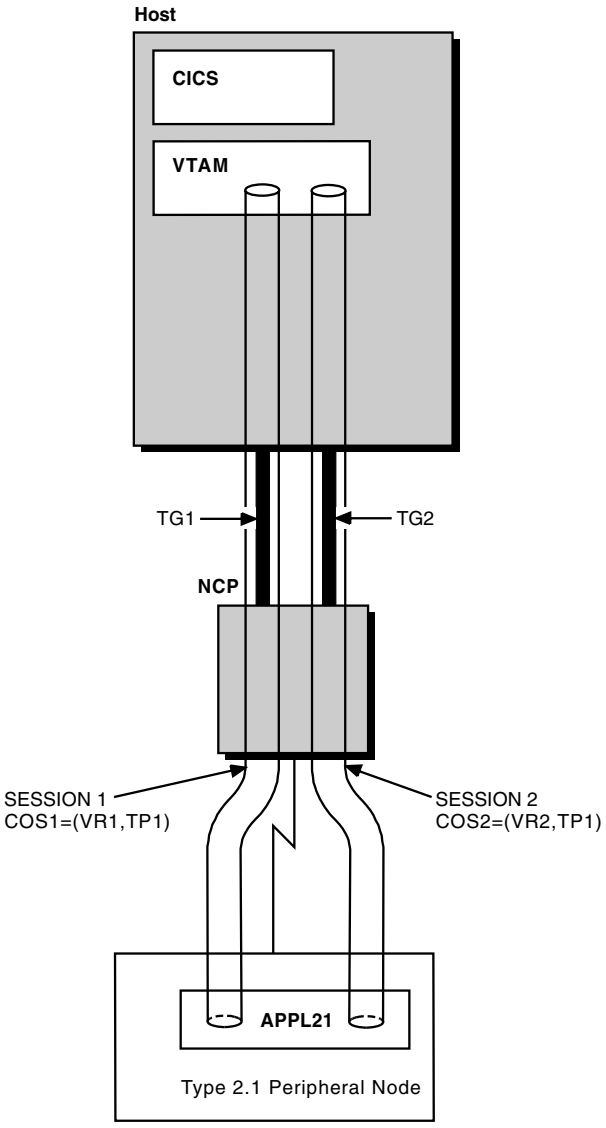


Figure 100. Parallel sessions using parallel transmission groups

Dynamic path update

Dynamic path update enables the VTAM operator to add NCP path definitions dynamically without regenerating and reloading the NCP. Using the same operator facilities, you can also dynamically add VTAM path definitions. The explicit route definitions in VTAM and NCP can be replaced or deleted only if the explicit route is inoperative (a status of INOP).

Note: If you are not attempting to modify an ER mapping to a TG, the ER state is not checked and the ER can be either operative or inoperative.

Dynamic path update can also be used to change inactive (a status of INACT) or undefined VRs, path definition values for virtual route pacing window size on inoperative routes, and transmission group thresholds on operative routes.

To use dynamic path update, at least one route must be defined and operative to establish the necessary SSCP-PU session between VTAM and each NCP. The dynamic path update changes are transmitted to NCP on this session. Dynamic path update members are filed in the VTAM definition library. A dynamic path update member can contain path specifications that are added, replaced, or deleted from one or more NCP or VTAM subareas. A set of path update specifications that is targeted to a specific subarea node within a dynamic path update member is called a dynamic path specification set. A dynamic path update member can contain one or more dynamic path specification sets.

The dynamic path update member in the following sample can be used to add NCP3 to the network shown in Figure 101 on page 370.

```
*  Dynamic Path Specification Set for VTAM 1 (subarea 1)
VTAM1    VPATH NETID=NETA
         PATH  DESTSA=3,
               ER0=(2,1),
               VR0=0
*  Dynamic Path Specification Set for NCP 2  (subarea 2)
NCP2     NCPPATH NETID=NETA
         PATH  DESTSA=3,
               ER0=(3,10)
```

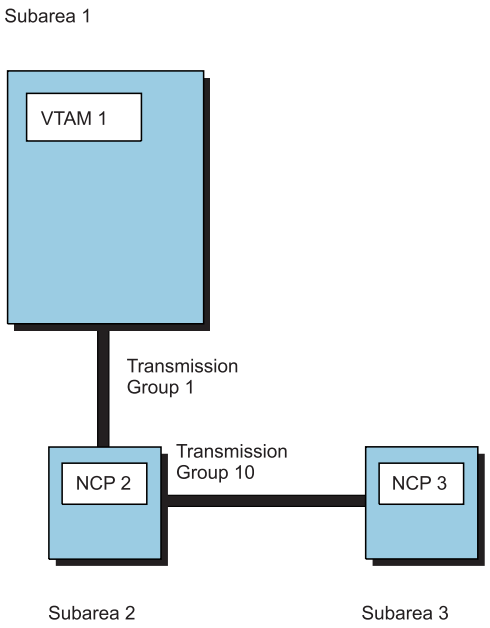


Figure 101. Sample single-domain network

Note that the name on the NCPPATH definition statement identifies the NCP for which the PATH statements are to be processed.

A dynamic path update can be triggered by any of the following methods:

- Activating a dynamic path update member using the VTAM configuration list or the VARY operator command. An active SSCP-PU session must exist between VTAM and the NCP whose routes are being dynamically updated, and the explicit routes being changed must be inoperative.
- Specifying the NEWPATH operand when activating an NCP with an operator command. This causes VTAM to activate the dynamic path update member immediately following NCP activation.
- Activating an NCP that has the NEWPATH operand specified on the PCCU definition statement for this host. This causes VTAM to activate the dynamic path update member immediately following NCP activation.

Note: Because dynamic path update changes do not alter VTAM or the NCP load module, you should use a method that ensures that these changes are reinstated if VTAM is restarted or if NCP is reloaded.

The dynamic path update member in the preceding example could be filed in the VTAM1 definition library. When the operator activates the dynamic path update member, VTAM1 identifies the dynamic path update member because the name in the VPATH definition statement matches its SSCPNAME (VTAM1). VTAM1 transmits the dynamic path update member to NCP2 using the active session path (SSCP-PU session).

Note: You should file the dynamic path update member for each subarea in separate dynamic path update members to avoid possible performance deterioration.

The following restrictions are placed on coding dynamic path update members and dynamic path specification sets:

- A dynamic path update member can include path definitions for only one VTAM host. If an incorrect definition statement or VPATH or NCPPATH definition statement is encountered, VTAM discards all PATH definition statements that follow the definition statement that is not valid up to the next recognizable VPATH or NCPPATH definition statement and issues a warning message. PATH definition statements that were already processed are not affected.
- When a dynamic path update member is activated using the VTAM configuration list or the VARY operator command, an NCPPATH definition statement is considered to be not valid if the NCP name on the NCPPATH definition statement does not match the resource name of an active NCP.
- When a dynamic path update member is activated by specifying the NEWPATH operand, an NCPPATH definition statement is considered to be not valid if the NCP name on the NCPPATH definition statement does not match the resource name of the NCP being activated.
- A dynamic path update member can include the DELETER operand if it applies to the host activating the dynamic path update member.

VTAM saves the name of the dynamic path update member in the NODELST data set (the VTAM configuration restart data set). The actual dynamic path specification sets are not saved in a VSAM configuration restart file.

Note: Extreme caution must be used when deleting or replacing a route for a node that is an intermediate routing node for other routes.

The NCPPATH definition statement identifies the NCP and the network to which the dynamic path update member applies in a dynamic path update member.

Chapter 14. Application programs

Each application program must be defined within an application program major node. Each application program represents a minor node.

You define an application program major node with one VBUILD definition statement, and you define application program minor nodes with one APPL definition statement for each application program in the major node. Following is a sample of definition statements for an application program major node. Each APPL definition statement represents a single logical unit to VTAM.

```
APPLNODE VBUILD TYPE=APPL
A50PAYRL APPL   AUTH=ACQ,          PERMIT APPLICATION TO ACQUIRE LUS
                EAS=1000,          CONCURRENT APPLICATION SESSIONS
                MODETAB=COMTAB,    LOGON MODE TABLE—APPLICATION IS SLU
                APPC=YES,          ENABLE LU 6.2 SUPPORT
                PARSESS=YES,       ENABLE PARALLEL SESSION FOR LU 6.2 SUPPORT
                ACBNAME=PAYROLL,   NETWORK LU NAME=A50PAYRL
                                   VTAM APPLICATION APPLID=PAYROLL

⋮
A50ACCTS APPL   ATNLOSS=ALL,       ENABLE ATTN EXIT FOR ALL LU 6.2 SESSIONS
                APPC=YES,          ENABLE LU 6.2 SUPPORT
                PARSESS=YES,       ENABLE PARALLEL SESSION FOR LU 6.2 SUPPORT
                SECACPT=ALREADYV,  ACCEPT ALREADY VERIFIED LU 6.2 CONVERSATIONS
                ACBNAME=ACCOUNTS   NETWORK LU NAME=A50ACCTS
                                   VTAM APPLICATION APPLID=ACCOUNTS

⋮
```

You can also code a model application program definition, which can be used as the definition for one or more VTAM application programs. For some application programs, such as Telnet or TSO/VTAM, using model application program definitions can save on resources (such as host network addresses) and is recommended.

Naming an application program

You can specify two different names for an application program, one required and one optional. The required application program name is the one that you specify in the label field of the APPL definition statement. The optional application program name is the one that you specify on the optional ACBNAME operand of the APPL definition statement.

You can use wildcard characters in the name you specify in the label field and on the ACBNAME operand. Using wildcard characters enables you to define a model application program definition, which can be used as the definition for one or more VTAM application programs. For more information on defining model application program definitions, see “Model application program definitions” on page 374.

If an application program name is not specified on the APPLID operand of the ACB macroinstruction, VTAM uses a name supplied by your operating system. If the name supplied by your operating system does not match a name on the APPL definition statement, the application program cannot open its access method control block (ACB) and is therefore not identified to VTAM.

© Copyright IBM Corp. 1984, 2011

If the application program is started by a job step that is a procedure invocation, the name used by the operating system is the procedure step name in the job control language (JCL) for the application program. Otherwise, it is the job step name in the JCL.

Note: TSO/VTAM and logon manager are VTAM application programs and must be defined to VTAM. For the specific coding requirements for the APPL definition statement that defines TSO/VTAM, see Appendix A, "TSO/VTAM," on page 715. For the specific coding requirements for the APPL definition statement that defines logon manager, see Appendix D, "Logon manager," on page 745.

If you are using an application program that needs information about the environment in which it is running, you can code macroinstructions in that application program to obtain information about the following:

- VTAM version and release levels
- VTAM component and feature identifiers
- Application program functions provided by your release of VTAM
- Application program network name
- Application program ACB name
- HOSTPU name
- Network ID for your network

For information on application program pacing, see "Application program pacing" on page 314.

Model application program definitions

You can code a model application program definition, which can be used as the definition for one or more VTAM application programs. You code a model application program definition by placing wildcard characters in the name field of an APPL definition statement that defines characteristics for one or more application programs.

If VTAM cannot find an active definition for the application program associated with the ACB specified on an OPEN macroinstruction, it searches a list of model definitions to find a name that best matches the name of the application program associated with the ACB. If it finds a match, it uses that definition statement to define the application program. If it does not find a match, it returns an error condition indicator to the application program.

An application program that is built from a model application program definition is called a *dynamic application program*, a *clone application program*, or a *replicated application program*.

Model application program definitions enable you to reduce the number of application program definitions you must code in VTAMLST.

You can use model application program definitions in nonsysplex and sysplex environments. The value of using them, however, is most apparent in sysplex environments.

In a sysplex environment, model application program definitions can be contained in a single VTAMLST accessible to all VTAMs in the sysplex. An application

program can open its ACB on any VTAM in the sysplex. In addition, an application program can be moved to any VTAM in the sysplex by closing its ACB on one VTAM and opening it on another.

Note: It is recommended that model application program definitions be used for application programs that intend to use the multinode persistent session (MNPS) function on an end node (EN), and it is required that model definitions be used for such applications on a network node (NN). Activating a nonmodel image of the MNPS application at an EN or NN, when the same application is active elsewhere (on another EN or NN), can lead to APPN search failures.

Overview

To code a model application program definition, simply code an APPL definition statement to define application program characteristics that you expect to be used by one or more VTAM application programs. Use wildcard characters in the name field of the APPL definition statement. You can use the following characters:

Asterisk (*) Represents 0 or more unspecified characters.

Question mark (?)
Represents a single unspecified character.

When placing wildcard characters in the name fields of model application program definitions, you should have some idea of which dynamic application programs might be built from those model definitions. You will probably want to use a naming scheme that ensures that dynamic application programs are not accidentally built from model application program definitions; that is, that an application program does not open its ACB using a model definition that you did not intend for it to use. See Table 40 on page 376 for examples of model application program definition names and the names of the dynamic application programs that could be built from them.

Coding guidelines

Coding guidelines for model application program names depend on whether you define only the required name (in the label field of the APPL definition statement) or both the required name and the optional name (on the ACBNAME operand of the APPL definition statement).

When only the required name is defined

A model application program name, including wildcard characters, can be a maximum of eight characters in length. A question mark (?) can be used anywhere in the model application program name. An asterisk (*) can be used in the second to eighth characters of the model application program name. Model application program names must be unique across all resources known to VTAM (including model CDRSCs).

There is no defined limit on the number of dynamic application programs that can be created from one application program model definition, nor is there a defined limit on the number of model definitions that you can define.

Model definitions can be defined in any number of application program major nodes. Those model definitions can appear in an application major node along with conventionally defined APPL definition statements. The model definitions and conventionally defined APPL definition statements can appear in any order.

Table 40 shows examples of model application program names.

Table 40. Sample model application program names

Model application program name	Names that could match
APPL*	Any name that begins with "APPL," followed by zero to four additional valid characters in length. Examples: APPL APPL1 APPL1A APPL01A APPL1234
?*APPL	Any name that begins with at least one valid character, plus zero to three additional valid characters in length, followed by "APPL." Examples: A1APPL A1AAPPL A01AAPPL A123APPL
AP?L	Any four-character name that begins with "AP," followed by one valid character in length, and ends with "L." Examples: AP1L APPL AP$L AP0L
APPL?	Any five-character name that begins with "APPL," followed by one valid character in length. Examples: APPL1 APPLS APPL3 APPL0
A*APPL	Any name that begins with "A," followed by zero to three valid characters in length, and ends with "APPL." Examples: AAPPL A1APPL A01APPL A$12APPL A123APPL

Note: You cannot specify a wildcard name that begins with an asterisk (*) because VTAM interprets this as a comment line. To achieve the effect of a wildcard name that begins with an asterisk, code two APPL definition statements:

- One whose name begins with ?* and ends with zero to six valid characters in length
- One whose name is the same as the zero to six characters

For example, *APPL would match any name that ended with APPL and began with zero to four valid characters in length. ?*APPL matches any name that begins with one to four valid characters in length, but it does not match a name that begins with zero to four characters in length, namely APPL. To allow for the possibility that an application program with a name ending in APPL and beginning with zero to four valid characters in length could open its ACB, you would have to code the following two APPL definition statements:

```
?*APPL  APPL  AUTH=(PASS,ACQ),EAS=5,...

APPL    APPL  AUTH=(PASS,ACQ),EAS=5,...
```

When the required name and the optional name (ACBNAME) are defined

When you use wildcard characters in the optional name specified on the ACBNAME operand, observe the following additional coding guidelines:

- At most, only one asterisk (*) can be used in the name specified on the ACBNAME operand. Up to eight question marks (?) can be used.
- The number of asterisks and question marks used in the name specified on the ACBNAME operand must match the number of asterisks and question marks used in the name specified in the label field of the same APPL definition statement.

  Note that the asterisks and question marks do not need to be in the same positions in the required and optional names.

All other coding guidelines that apply when only the required name is defined also apply when both the required name and the optional name are defined.

When VTAM finds a match on an ACBNAME operand, the characters on the APPLID operand on the ACB macroinstruction matching the wildcard characters in the optional name specified on the ACBNAME operand are substituted in the required name, in left-to-right order, by wildcard type. That is, the character that matches the first question mark specified on the ACBNAME operand is substituted for the first question mark in the required name. The character that matches the second question mark is substituted for the second question mark in the required name. All the characters (including none) in the name specified on the APPLID operand on the ACB macroinstruction that match the asterisk in the optional name specified on the ACBNAME operand are substituted for the asterisk in the required name.

How VTAM finds the best match

If VTAM finds more than one model application program definition that could match the name of an application program requesting to open its ACB, it uses the model definition whose name best matches the name of the application program requesting to open its ACB. (Similarly, if VTAM finds more than one model CDRSC definition that matches the name of a cross-domain resource it encounters, it uses the model definition whose name best matches the name of the cross-domain resource.) VTAM determines the best match following the rules for wildcard support described in *z/OS MVS Setting Up a Sysplex*. This section provides an overview of these rules (which apply to model CDRSCs, as well as to model applications).

VTAM compares a model application program name with the name of the application program trying to open its ACB. It compares each name, character for character, scanning from left to right. For each character, VTAM looks for a match in the following order:

1. Most specific match: The actual character (A–Z, 0–9, @, #, or $)
2. Next most specific match: A single-character wildcard character (?)
3. Least specific match: A multiple-character wildcard character (*)

This order implies the following:

- The model application program name that contains no wildcard characters will always be the most specific.

For example, if an application program named ABCDEFG requests to open its ACB, VTAM will search VTAMLST for a definition of ABCDEFG. If there is a definition statement named ABCDEFG, VTAM will use that statement as the definition of ABCDEFG.

- The model application program name with the most actual characters appearing before a wildcard character is more specific.

  For example, if an application program named ABCDEFG requests to open its ACB, VTAM will search VTAMLST for a definition of ABCDEFG. If there is no definition statement named ABCDEFG, but there are two model definitions named ABC?EFG and ABCD*, VTAM will use ABCD* as the definition of ABCDEFG. As shown below, VTAM scans from left to right, looking first for character-to-character matches, which it finds for the first three characters in ABC?EFG. However, as it continues scanning, it finds that ABCD* has a character-to-character match for the first four characters, which gives it priority over ABC?EFG. Even though the question mark in ABC?EFG is in an earlier position (fourth) than the asterisk in ABCD* (fifth), because ABCD* matches ABCDEFG character-to-character for the first four positions, and because VTAM searches first for character-to-character matches, ABCD* is the more specific match.

```
A B C D E F G
| | |
A B C ? E F G


A B C D E F G
| | | |
A B C D *
```

Other examples are:

ABC?DE is more specific than

ABC*DE because the single-character wildcard character (?) is more specific than the multiple-character wildcard character (*).

ABC?DE is more specific than

ABC??E because the actual character (D) is more specific than the single-character wildcard character (?).

ABC?DE is more specific than

ABC?D* because the actual character (E) is more specific than the multiple-character wildcard character (*).

Tip: Use the DISPLAY MODELS command with the APPL operand to verify that the model definition that you intend to use for your application name is the one that VTAM selects.

Example of using model application program definitions

The following examples show how model application program definitions are used. One example shows how definitions are used when wildcard characters are used only in the required name. The other example shows how definitions are used when the ACBNAME operand is coded and wildcard characters are used in the optional name defined on the ACBNAME operand.

When only the required name is defined

The following model application program definitions are contained in VTAMLST:

```
APPL?   APPL  AUTH=(PASS,ACQ),EAS=5,...

APPL*   APPL  AUTH=(PASS,ACQ),EAS=1,...

A*APPL  APPL  AUTH=(PASS,ACQ),EAS=6,...

AP?L    APPL  AUTH=(PASS,ACQ),EAS=3,...
```

The OPEN macroinstruction, OPENXYZ, specifies that the application program associated with ACB123 requests to open its ACB, as follows:

```
OPENXYZ  OPEN  ACB123

ACB123   ACB   AM=VTAM,APPLID=APPL1A,...
```

The ACB macroinstruction, ACB123, requests to be associated with the definition of application program, APPL1A. VTAM cannot find a definition for APPL1A in VTAMLST, but it does find a name that best matches APPL1A, and that is APPL*. The asterisk in APPL* indicates that it can be used as the definition for any application program whose name begins with APPL, followed by zero to four valid characters in length. APPL? is not a valid match in this example because the question mark indicates that it can be used as the definition statement for any application program whose name begins with APPL, followed by only one valid character in length.

When the required name and the optional name (ACBNAME) are defined

The following model application program definitions contain an optional name defined on the ACBNAME operand:

```
NET?NAME  APPL  ACBNAME=WILD?,AUTH=(PASS,ACQ),EAS=5,...

NET*NM    APPL  ACBNAME=WILD*,AUTH=(PASS,ACQ),EAS=1,...

NET?NAM?  APPL  ACBNAME=WILD??,AUTH=(PASS,ACQ),EAS=6,...
```

The OPEN macroinstruction, OPENXYZ, specifies that the application program associated with ACB123 requests to open its ACB, as follows:

```
OPENXYZ  OPEN  ACB123
ACB123   ACB   AM=VTAM,APPLID=WILD1,...
```

The ACB macroinstruction, ACB123, requests to be associated with the definition of application program, WILD1. VTAM cannot find a definition for WILD1 in VTAMLST, but it does find a name, specified on the ACBNAME operand, that best matches WILD1, and that is WILD?. The ? in WILD? is replaced with 1, as is the ? in NET?NAME. Therefore, the optional name of the application program is WILD1 and the required name is NET1NAME.

If the name specified on the APPLID operand on the ACB macroinstruction was WILDONE, VTAM would use WILD* as the model application program definition. The optional name would be WILDONE and the required name would be NETONENM.

If the name specified on the APPLID operand on the ACB macroinstruction was WILD1S, VTAM would use WILD?? as the model application program definition. The optional name would be WILD1S and the required name would be NET1NAMS.

Resource state requirements

A model application program must be active before it can be used to build dynamic application programs. You can activate a model application program as you activate other application programs by doing one of the following:

- Issue a VARY ACT command.
- Include the major node in which the model application program is defined in the configuration list that VTAM uses when it is initialized.

By activating the model application program, you ensure that the state of the model application program is connectable, thus making it available to build dynamic application programs.

You cannot activate dynamic application programs with the VARY ACT command. Dynamic application programs are activated only by opening their ACBs. For a dynamic application program to successfully open its ACB, a model application must exist to build the dynamic application program, and that model application program must be connectable.

Authorizing application facilities

You can use the following facilities to enhance your application program environment:

- Passing and validating logon requests
- Overriding dial number digits
- Acquiring LU sessions
- Enabling parallel sessions
- Authorizing privileged paths

Passing and validating logon requests

You can write a logon exit routine that passes VTAM session requests to another application program. This allows the application program to validate the logon before accepting it. Passing an LU session request to another application program queues the LU session request to that application program. A session is established after the original session has ended. The AUTH=PASS operand on the APPL definition statement permits the passing of session requests.

Overriding dial number digits for dial or token-ring connections

In a dial-out operation, an application program can be authorized to override the dial number digits and other parameters specified in the switched major node for the device being contacted. The device can be PU types 1, 2, or 2.1. Normally, if these parameters do not match in XID (exchange identification), the connection is terminated. However, you can authorize an application to override this checking. Coding the AUTH=ASDP operand on the APPL definition statement allows the application to supply dial parameters to the model switched major node during session initiation.

Notes:

1. This operation can cause a security exposure.
2. If an active APPN connection exists, VTAM may attempt to use it rather than dialing a new connection.

Acquiring LU sessions

An application program acquires LU sessions by issuing either an OPNDST macroinstruction with OPTCD=ACQUIRE or a SIMLOGON macroinstruction followed by an OPNDST macroinstruction with OPTCD=ACCEPT specified.

You can enable the acquiring of sessions by coding the AUTH=ACQ operand on the APPL definition statement.

Enabling parallel sessions

Certain session protocols can prevent an application program from performing more than one transaction at a time on a single session. Using the parallel sessions function, you can enable an application program to engage in several sessions at once with the same logical unit. Thus, the application program can perform multiple transactions at the same time.

If you want an application program to have parallel sessions, you must authorize it by coding PARSESS=YES on the APPL definition statement. APPC=YES overrides PARSESS=NO to allow parallel sessions. The application program can distinguish sessions between the same LUs through the proper use of the communication identifier (CID).

Authorizing privileged paths

Authorized path is a VTAM facility that enables an application program to specify that a data transfer operates in a privileged manner. Authorized path enables an authorized application program to run entirely under a system request block (SRB) after the application program ACB has been opened. In most cases, this means a shorter path length through VTAM for RPL-related macroinstructions. An application program can be authorized by using the authorized program facility (APF) running in the supervisor state or by running in a key of 0 through 7. For a full description of authorized path, refer to *z/OS Communications Server: SNA Programming*.

Data compression

When using application programs in a multiple-domain environment, consider taking advantage of the data compression facility. This facility enables VTAM to compress the data for messages on selected LU-LU sessions. Data compression is transparent to application programs.

When using application programs in a single-domain environment, between application-to-application sessions in the same host, there is no benefit to compressing data. In fact, extra processing cycles are expended. If the SLU is an application in the same host, VTAM builds the BIND without data compression even if data compression is indicated. Data compression is not allowed in these single-domain conditions.

Also, you should not use the data compression facility for VTAM in conjunction with any application program data compression. After the application program has compressed the data, the amount of additional compression would be minimal.

You can implement data compression for logical units in either an extended BIND (types 2.1, 4, or 5 physical unit) or nonextended BIND environment (type 1 or type 2 physical unit). You cannot compress data for sessions with LU type 4 or LU type 7 in a nonextended BIND environment.

Data compression uses data storage.

Types of compression

The use of data compression can be negotiated for each session through the BIND and BIND(RSP) request units (RUs). The session partners can negotiate whether to use data compression. If compression is to be used, the session partners can negotiate the type of compression. VTAM supports two different compression algorithms:

- Run length encoding (RLE) compression

  This type of compression replaces strings of identical bytes with shorter encoded strings.
- Dynamic dictionary-based compression (called adaptive compression in this document)

  This type of compression is an adaptive dictionary-based compression algorithm similar to Lempel-Ziv. It is applicable to a wide range of data types. It uses tables that adapt dynamically to match the data being sent or received. Adaptive compression replaces the original data with a set of compression codes. Each set represents one or more bytes. Following are the variations of adaptive compression:

  Small table This type uses 9-bit compression codes.

  Medium table This type uses 10-bit compression codes.

  Large table This type uses 12-bit compression codes.

  With adaptive compression, each partner begins a session with an identical set of tables. The senders of the RU update their send tables as data is compressed. The receivers of the RU makes identical updates to their receive tables as data is decompressed. This method keeps both ends of the session identical without exchanging table data between nodes.

  By default, the send and receive tables are continually updated. You can speed up adaptive compression processing on outbound messages, through software only or with the help of hardware compression, by using the CMPMIPS start option. By specifying a value from 1 to 99 (default value is 100), VTAM suspends and resumes updating of the send and receive tables whenever compression effectiveness exceeds or falls below a threshold corresponding to the CMPMIPS start option value.

  Thus, the CMPMIPS option allows you to balance the number of machine cycles needed with the effectiveness of compression for outbound messages. Higher values for CMPMIPS will likely increase both compression effectiveness and cycle usage, while lower CMPMIPS values will likely lower both compression effectiveness and cycle usage.

Implementing data compression

If you choose to use data compression, you can adjust the use of compression for your system by using a combination of compression levels specified for VTAM and for application programs. The following levels of data compression are available:

0 No compression
1 RLE
2 Adaptive small table
3 Adaptive medium table
4 Adaptive large table

- The overall compression limit for a VTAM host can be:

  - Initialized through the CMPVTAM start option. For more information on the CMPVTAM start option, refer to *z/OS Communications Server: SNA Resource Definition Reference*.
  - Changed by the MODIFY COMPRESS command. (The compression levels of active sessions are not affected.) For more information on this command, refer to *z/OS Communications Server: SNA Operation*.
  - Displayed through the DISPLAY VTAMOPTS command. For more information on this command, refer to *z/OS Communications Server: SNA Operation*.
- Input and output compression limits for an application program can be:
  - Initialized through the CMPAPPLI and CMPAPPLO operands on the APPL definition statement.
  - Changed by the MODIFY COMPRESS command. (The compression levels of active sessions are not affected.)
  - Displayed through the DISPLAY ID command.
- COMPRES is the compression-override operand on the MODEENT macroinstruction that specifies how the negotiation of compression should be handled. You can use COMPRES to override the BIND negotiation process and either require or prohibit compression. This operand allows data compression specifications to be associated with a logon mode name. For more information on this operand, refer to *z/OS Communications Server: SNA Resource Definition Reference*.
- The DISPLAY SESSIONS command can indicate the compression levels in use for a given session and the average message length reductions achieved in both directions. For more information on this command, refer to *z/OS Communications Server: SNA Operation*.

Compression level negotiation

If the value specified on the COMPRES operand of the MODEENT macroinstruction allows negotiation, the compression levels for a session are negotiated through the BIND and BIND(RSP) RUs.

The data compression information is carried on a control vector X'66'. Control vector X'66' is used only for compression-level negotiation. The control vector is comprised of these subvectors:

X'80' is passed on a CDCINIT and a CINIT.
X'81' is passed on a BIND to carry the requested compression levels.
X'82' is passed on a BIND(RSP) to carry the actual compression levels.

The SLU first requests a compression level on the CDCINIT. The PLU responds with requested levels specified on the BIND. The SLU responds with actual levels on the BIND(RSP).

Note: There are limited negotiation capabilities for sessions using a nonextended BIND. Bits 6 and 7 in byte 25 of the BIND are reserved for data compression, and are used to negotiate whether compression is acceptable or not. No level negotiation is possible. In the PLU-to-SLU direction, level 2 is the highest possible compression level (If VTAM owns the PLU, setting CMPVTAM = 1 further limits the PLU-to-SLU level to 1). In the SLU-to-PLU direction, level 1 is the highest possible compression level.

Negotiation when VTAM owns the PLU

When VTAM owns the PLU for the session, VTAM has the option to request compression. VTAM does not request compression if any of the following conditions occur:

- The CMPVTAM start option is set to 0, which indicates no compression.
- The CMPAPPLI and CMPAPPLO operands on the APPL definition statement are set to 0, which indicates no compression.
- The SLU owner prohibits compression through bits in the control vector X'66'. Whether the control vector indicates that compression is prohibited or required is determined by the setting of the COMPRES operand of the MODEENT macroinstruction.

If none of the previous conditions occur, the PLU requests compression levels. The PLU-to-SLU compression level is the lower of the values set on the CMPVTAM start option or the CMPAPPLO operand on the APPL definition statement. The SLU-to-PLU compression level is the lower of the values set on the CMPVTAM start option or the CMPAPPLI operand on the APPL definition statement.

Negotiation when VTAM owns the SLU

When VTAM owns the SLU for the session, the SLU can begin negotiation when it sends the control vector X'66'. The control vector indicates whether compression is prohibited or required. If compression is desired, the COMPRES operand on the MODEENT macroinstruction must indicate that compression is required.

When sending the BIND(RSP), the SLU negotiates no compression if any of the following occur:

- The BIND does not request compression.
- PROHIB or SYSTEM was specified in the control vector X'66'.
- The CMPVTAM start option is set to 0, which indicates no compression.

Note: When compression is negotiated from the SLU side, the CMPAPPLI and CMPAPPLO operands are not used.

If none of the previous conditions occur, the SLU requests compression levels. The PLU-to-SLU compression level is the lower of the values set on the CMPVTAM start option or the requested PLU-to-SLU compression level passed on the BIND. The SLU-to-PLU compression level is the lower of the values set on the CMPVTAM start option or the requested SLU-to-PLU compression level passed on the BIND.

Examples of data compression

The following are examples of how the values set for CMPVTAM, CMPAPPLI, and CMPAPPLO work together to determine the compression levels for a session. Figure 102 on page 385 shows a sample configuration.

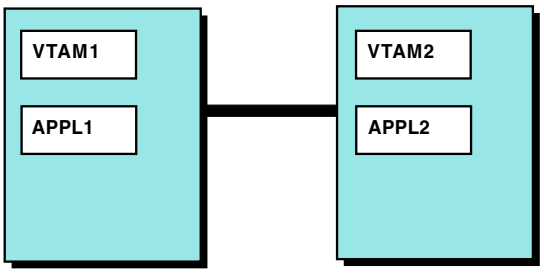


Figure 102. Data compression yield

For the examples, assume the values shown in Table 41. Also assume that COMPRES=REQD is specified on the logon modes used for the sessions.

Table 41. Compression values for example of data compression yield

VTAM1	APPL1	APPL2	VTAM2
CMPVTAM = 4	CMPAPPLI = 4	CMPAPPLI = 4	CMPVTAM = 3
	CMPAPPLO = 1	CMPAPPLO = 2	

In the first example, APPL1 is the PLU and APPL2 is the SLU.

1. VTAM determines that the requested PLU-to-SLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM1 or the CMPAPPLO operand for APPL1) is 1.
2. VTAM determines that the requested SLU-to-PLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM1 or the CMPAPPLI operand for APPL1) is 4.
3. APPL1 passes the requested levels to APPL2 on the BIND.
4. VTAM compares the requested levels to the VTAM2 CMPVTAM level. APPL2 determines that the actual PLU-to-SLU level for this session is 1 and the actual SLU-to-PLU level for this session is 3.
5. APPL2 returns the actual levels on the BIND(RSP).

In the second example, APPL2 is the PLU and APPL1 is the SLU.

1. VTAM determines that the requested PLU-to-SLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM2 or the CMPAPPLO operand for APPL2) is 2.
2. VTAM determines that the requested SLU-to-PLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM2 or the CMPAPPLI operand for APPL2) is 3.
3. APPL2 passes the requested levels to APPL1 on the BIND.
4. VTAM compares the requested levels to the VTAM1 CMPVTAM level. APPL1 determines that the actual PLU-to-SLU level for this session is 2 and the actual SLU-to-PLU level for this session is 3.
5. APPL1 returns the actual levels on the BIND(RSP).

Summary of data compression

The following summarizes data compression.

- Data compression is established between users of a session through the BIND and RSP(BIND) flows. VTAM can handle compression settings at one end or both ends on behalf of VTAM applications.

- Compression is disabled on sessions with both ends in the same VTAM.
- VTAM defines compression levels 0 to 4 used in VTAM start option CMPVTAM and PLU-APPLication definition options CMPAPPLI (incoming flow) and CMPAPPLO (outgoing flow). The actual compression level settles on the lower of the CMPAPPLI value or the CMPAPPLO value, and the level acceptable for each of the hosts (CMPAPPL value for VTAM).
- The CMPMIPS option can be used to balance CPU cycles versus LZ-like compression efficiency for outgoing data only. The value (0–100) can be changed dynamically using the command

  ```
  F NET,VTAMOPTS,CMPMIPS=value
  ```

  When CMPMIPS=0, data on all sessions is sent uncompressed. When CMPMIPS=100, most CPU cycles are used for best compression efficiency. A recommended value of 50 should save CPU cycles while maintaining high compression efficiency.
- RLE compression is allowed with its normal CPU utilization anytime CMPMIPS is not 0.
- Compression efficiency may be displayed on selected sessions using the command

  ```
  D NET, SESSIONS,SID=sessionid
  ```

  Efficiency is shown as a percentage reduction value on the original length of the uncompressed data.
- Some VTAM compression level statistics may be displayed using the command

  ```
  D NET,STATS,TYPE=COMPRESS
  ```
- The logmode table option COMPRES= SYSTEM | PROHIB | REQD can be used for the logmode table entry for the sessions over which compression is requested.
  - The logmode table applies to the SLU-controlling VTAM.
  - The default value SYSTEM may be OK when the remote station is smart enough to accept, reject or update the compression levels requested by VTAM on the BIND CV66. This may be the case for a CM/2 acting as node 2.1.
  - SYSTEM=REQD is required for VTAM APPL to VTAM APPL sessions. For a work station with compression capability but no compression negotiation capability on CV66, set COMPRES=REQD. The PLU VTAM compression levels (resolved from CCMPVTAM, CMPAPPLI, or CMPAPPLO) must be acceptable to the work station.

Security features

VTAM provides a variety of security features that can be enabled for an application program, including the following:
- Cryptography facility
- Message authentication
- SLU authentication
- VTAM application security
- Confidential data

For information on LU 6.2 security features, see “LU 6.2 security” on page 417.

Cryptography facility

The cryptography facility protects the confidentiality of data transmitted between network resources by enciphering and deciphering session data. Cryptography is

available for both LU 6.2 and non-LU 6.2 sessions. Support is available for both switched and nonswitched LUs. However, support is not available for binary synchronous communication (BSC) or local non-SNA devices.

The facility establishes cryptographic sessions for application programs and peripheral node LUs that require cryptographic services. For an LU to have a cryptographic session, the host processor must support cryptography.

The encryption facility provides two levels of cryptographic sessions:

Selective
Each end of the session selects the data to be enciphered before transmission. The selection is based on the capability of the session partner and the availability of cryptographic services.

Required
All outbound data request units are enciphered and all inbound data request units are deciphered.

The encryption facility uses services provided by the z/OS Integrated Cryptographic Service Facility (ICSF) and S/390 or zSeries Cryptographic Co-Processor. ICSF is a licensed program that runs under MVS and provides access to the hardware cryptographic feature for programming applications. The combination of the hardware cryptographic feature and ICSF provides secure high-speed cryptographic services.

Tip: You might be able to use cryptographic products other than ICSF if the cryptographic product runs in one of the supported modes of operation. The following terms are used to refer to cryptographic products that support one of these modes of operation:

- PCF/CUSP - Refers to any cryptographic product that is compatible with PCF/CUSP.
- CCA - Refers to any cryptographic product that is compatible with Common Cryptographic Architecture (CCA).

Requirement: Triple-DES 24–byte encryption requires the use of Common Cryptographic Architecture (CCA). CCA defines a set of cryptographic functions, external interfaces, and a set of key management rules that provide a consistent, end-to-end cryptographic architecture across different IBM platforms.

The cryptographic facilities provide services that include handling requests that VTAM receives to generate a cryptographic key. The cryptographic key is used to encipher and decipher session data.

VTAM support of dynamic cryptographic keys allows you to do the following:

- Define both unique and alternate key-encrypting key names for LUs and CP/SSCPs.
- Switch between cryptographic products while VTAM is running.

  Note: Switching to PCF/CUSP will terminate any sessions using triple-DES.
- Establish “clear” sessions (without encryption) if ENCR=COND and when either session partner does not support cryptography, or when cryptographic services are temporarily unavailable.

Implementing cryptography for LU-LU session data

To use the cryptography feature:

1. Install and activate the cryptographic product. The ENCRYPTN start option enables you to start VTAM before activating the cryptographic product. (ENCRYPTN=CCA is required if triple-DES encryption will be used by LUs or applications owned by this node.)
2. File cryptographic keys. File the cryptographic key data set at the host processor before activating any LUs that are used in cryptographic sessions.

   For information about filing cryptographic keys, see Appendix E, "Cryptographic keys," on page 755.
3. Specify the cryptographic requirements. VTAM uses this information to identify the cryptographic session keys and to establish the cryptographic session.
   - Code the ENCR operand and the ENCRTYPE keyword (DES or TDES24) on the LU and APPL definition statements to define the cryptographic capabilities of LUs and application programs.
   - Code the ENCR operand on the MODEENT macroinstruction to specify cryptographic session requirements for an LU.
   - Code the ENCRTYP operand on the MODEENT macroinstruction to specify encryption type for an LU (TDES24 is the only available value).
   - Identify the name of the cryptographic key that will be used to establish cryptographic sessions for the LU. Code the CKEYNAME operand on the LU definition statement or use the default, which is the LU name.
   - Specify whether you want to use the alternate cryptographic key name during session activation. You can either use the CKEY operand on the MODEENT macroinstruction or issue a modify security command to switch to the alternate CKEY.
4. Initiate the cryptographic session.

Note: For information about the options you can specify on the ENCR, ENCRTYPE, CKEYNAME, and CKEY operands, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

Establishing cryptographic sessions

Cryptographic LU-LU sessions are established based on the requirements and capabilities of the session partners.

As shown in Figure 103 on page 389, the application programs can establish the following sessions with the peripheral node LUs:

APPL1A Can establish a nonencrypted session with LU1A.

Can establish a cryptographic session with LU1B *if* a cryptographic product is currently available. If it is not available, a session without encryption is started.

APPL1B Can establish a cryptographic session with LU1B *if* a cryptographic product is currently available; otherwise the session fails.

Cannot establish a session with LU1A; the session fails. APPL1B requires a cryptographic session that LU1A does not support.

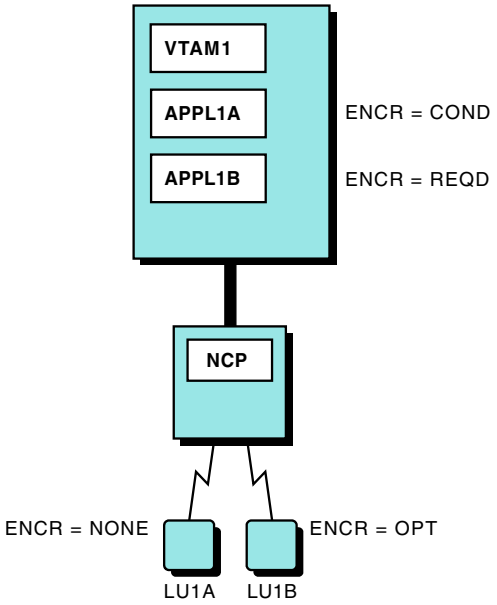


Figure 103. Encryption facility specifications

The MODIFY ENCR or MODIFY SECURITY can be used to change the cryptographic capability of a logical unit, or change the capability to a higher level of cryptography than was previously specified. Only MODIFY SECURITY can be used to change the minimum encryption type (ENCRTYPE).

When VTAM tracing is in effect for a cryptographic session, the protected data in the VTAM buffers is neither displayed nor printed.

When sessions require encryption, a session could fail if a cryptographic support is not currently available. You can define some sessions to establish a clear session if cryptographic services are not available at session start. The ENCR=COND operand on the APPL definition statement specifies whether to establish a clear session if a cryptographic facility is not available. A clear session is established when a cryptographic facility is not available and one of the following conditions exist:

- Both session partners have ENCR=COND coded.
- When ENCR=COND is coded by one session partner and ENCR=OPT is coded for the other session partner.

If either end of a session specifies a higher level of encryption than ENCR=COND (ENCR=SEL or ENCR=REQ), an encrypted session is required and the session request fails. The following shows examples of the interactions of the ENCR and ENCRTYPE keywords in determining the encryption level.

Table 42. DES-TDES24 encryption options

ENCR		ENCRTYPE		Results
SLU	PLU	SLU	PLU	
REQD	REQD	DES	DES	8-byte DES encryption is used.

Table 42. DES-TDES24 encryption options (continued)

ENCR		ENCRTYPE		Results
SLU	**PLU**	**SLU**	**PLU**	
REQD	REQD	TDES24	DES	TDES24 encryption is used if PLU hardware is capable of supporting triple-DES. The PLU side can be upgraded to triple-DES. Encryption fails if PLU hardware is not capable of supporting triple-DES.
REQD	REQD	DES	TDES24	Encryption fails; the SLU has obtained too small a key (8 bytes) while the PLU requires a 24-byte key. The SLU VTAM cannot be upgraded; it obtains the session key.
OPT/REQD	REQD	DES*	TDES24	(* — SLU also sets &ENCRTYP=TDES24) TDES24 encryption used, assuming both SLU and PLU hardware is capable of supporting triple-DES.
OPT	COND	DES	TDES24	Session established in the clear. The ENCRTYPE mismatch cannot be resolved, but encryption is not required.

Information needed to encipher and decipher session information is included in session establishment commands. When a session that will use cryptography is being initiated, the cryptographic facility of the SLU SSCP/CP enciphers the cryptography key. From the cryptography key two enciphered keys are created:

1. One copy is enciphered under the SLU master key

   The enciphered session key is placed in the BIND image to be used later. For 24–byte keys, the extra 16 bytes are transported as control vector data and not in the BIND image itself.

2. Another copy is enciphered under a cross key.

   The enciphered cross key is inserted in the CDINIT or CDCINIT command (or corresponding APPN command) and is used by nodes along the session path. Depending on the configuration, this key is deciphered and reenciphered at every node along the session path or only by the session endpoint.

See Appendix E, “Cryptographic keys,” on page 755 for a description of a cross key as well as how these keys are entered into the various cryptographic products.

Note: The actual session data that is enciphered under the session key is enciphered and deciphered only at the session endpoints.

Cryptographic session initiation

The following only applies to the session key *enciphered under a cross key* as it is used during session initiation.

VTAM supports both end-to-end and host-by-host encryption. The method used is dependent on the types of nodes in the configuration and the coding of the cryptographic key data sets (CKDS).

End-to-end
Session key is deciphered and or reenciphered only at the session endpoints or APPN endpoints during session initiation.

Host-by-host
Session key is deciphered and or reenciphered at every VTAM along the path during session initiation.

The installation actually determines which method VTAM will use by placing cross keys in the appropriate cryptographic facility data sets and by the capability of the nodes involved.

- If VTAM is defined as an APPN node and, if during session setup VTAM finds the name of the CP or network node server (NNS) of the target, that VTAM can provide end-to-end cryptography support for subarea, APPN, and mixed subarea and APPN networks.
- If VTAM is defined as a subarea only node, the session cryptography key usually must be encrypted on a host-by-host basis.

  However, if the VTAM node that is defined as a subarea only is connected to an NN, the first host-by-host decipher and reencipher will be done from the subarea only VTAM to the NN. When the NN (by definition an APPN-capable node) chooses the cross key for the next leg of the session initiation, it may find the name of the CP of the target so this host-by-host decipher and reencipher will be altered to an end-to-end decipher and reencipher.

During session initiation VTAM interrogates the encryption facility to determine whether cross key has been defined for a particular name.

The following describes the order in which VTAM will choose a name by which to interrogate the cryptographic product:

1. If the name is present, the cross key associated with that name will be used to encipher the session key for the next, or only, hop.
2. If the name is not present, the next favored choice of name will be selected and the interrogation will again be attempted.
3. If the entire list is attempted and no name is found, the attempt to initiate a cryptographic session will fail.

When preparing to send a session initiation request into an APPN network or when sending a subarea CDINIT (request or response):

1. VTAM chooses the name of the owning CP of the PLU.
2. If step one fails, and the information is known to VTAM, VTAM chooses the name of the NNS of the PLU. This choice is not possible for a subarea CDINIT because there is no network node server (NNS) involved in a subarea-only configuration.
3. If that fails, VTAM chooses the name of the adjacent node, but only if the adjacent node is either another VTAM using an SSCP-SSCP control session or an NNS for the session. Note that cryptographic processing cannot be done by intermediate APPN nodes because they do not parse the session initiation request.

When sending cryptographic information on a CDCINIT into a subarea network, VTAM always follows the earlier host-by-host algorithm — encrypt the key in the cross domain key of the adjacent node. Also, if cryptographic processing was done on CDINIT, it will not be done again on CDCINIT.

The first key found using the above search will be the key used. VTAM also includes the partner name in a control vector so the other VTAMs along the path either ignore the cryptographic fields when the name included is not theirs, or decipher the cross key. If VTAM deciphers the cross key, VTAM then acts upon the cross key by saving the key if this is the endpoint, or reenciphering the key, changing the name in the control vector, and then forwarding the session initiation.

End-to-end cryptography

End-to-end cryptography enables an LU-LU session to use encryption as long as all intermediate VTAMs are Version 3 Release 2 or later and both endpoint VTAMs are Version 4 Release 1 or later. Instead of filing cryptographic key data sets at each host that an LU-LU session initiation request traverses, the installation defines the data sets at only the endpoint hosts. These data sets contain the cross keys for the other endpoint and not the adjacent node. Defining cryptographic key data sets between only endpoint hosts eliminates the need to share cryptographic key information with other hosts. This can be important if, for example, the intermediate hosts are owned by a different company. Using end-to-end cryptography in Figure 105 on page 394, cryptographic key data sets would need to be filed in only VTAM1 and VTAM3.

Defining cryptographic keys between only endpoint hosts can still become burdensome, since many endpoints can exist and the amount of key definition can get very large. Therefore, in an APPN network, cryptographic key data sets can be defined in two ways:

- Between a network node server and its end nodes
- Between network node servers

Although more key definition is required at a network node server using this strategy, the overall number of cryptographic key data sets declines. End nodes are required to define only one key to the network node server (rather than a key for each node the end node is to communicate with). In addition to the keys for its served end nodes, network node servers are required to define only keys for other network node servers (rather than a key for each node its end nodes are to communicate with).

Both session endpoints do not have to use their network node servers for cryptography. That is, one end node can define a key to its server and the server can define a key for the destination end node (instead of defining a key for the destination end node network node server).

An example of defining cryptographic keys using the network node server is shown in Figure 104 on page 393. If VTAM1 files a cryptographic key data set with VTAM2 and VTAM2 files cryptographic key data sets with VTAM4, applications and LUs at VTAM1 can have cryptographic sessions with applications or LUs at VTAM4 without the need for VTAM1 and VTAM4 to exchange keys.

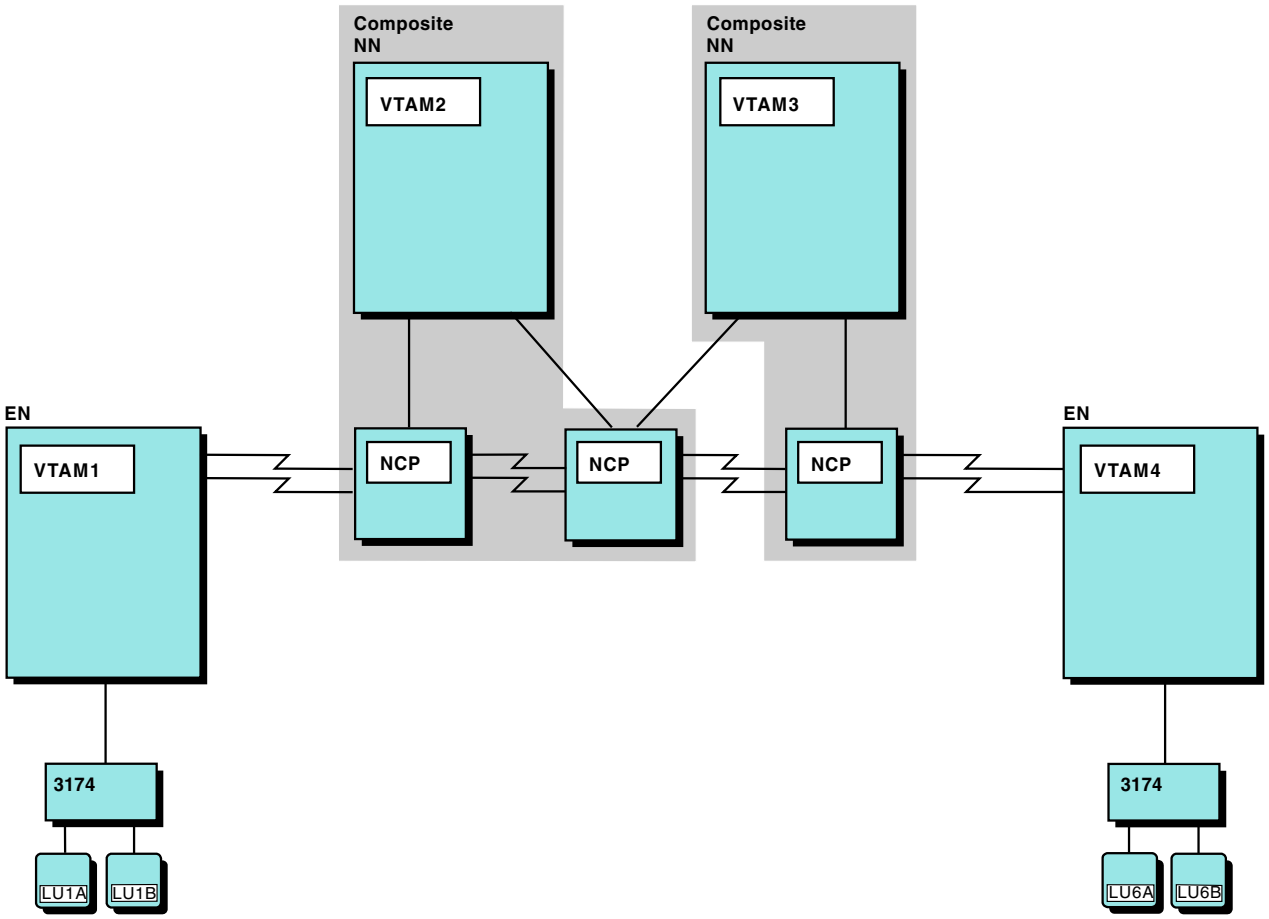


Figure 104. Encryption facility in an APPN environment

Host-by-host cryptography

With encryption on a host-by-host basis, each host must support cryptography, and cryptographic key data sets must be filed at each host involved in the LU-LU session initiation.

In Figure 105 on page 394, for cryptographic sessions to exist between resources in VTAM1 and VTAM3, cryptographic key data sets must be filed at VTAM1, VTAM2, and VTAM3.

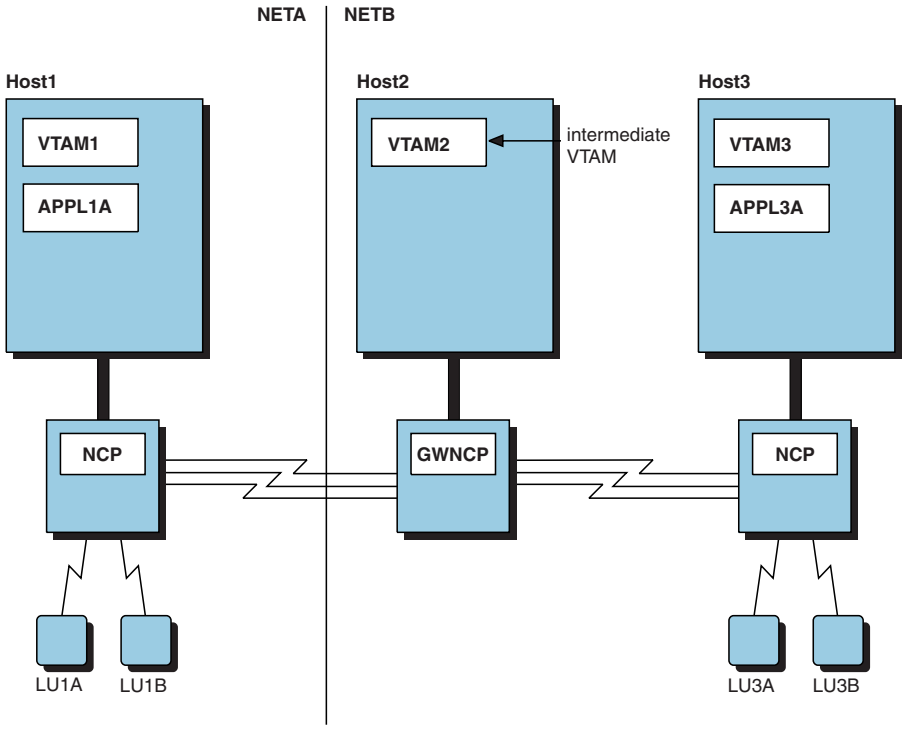


Figure 105. Encryption facility in multiple-network environment

Changing cryptographic keys dynamically

You can change LU and CP/SSCP resource keys for a session while VTAM is running. This enables you to avoid synchronization problems when changing keys, and helps reduce down time when changing static definitions.

By coding the appropriate operands and issuing the appropriate commands, you can:

- Use alternate LU key-encrypting keys while you change the LU master key in the CKDS at the control point (CP/SSCP).
- Use alternate CP/SSCP import and export key-encrypting keys (KEKs) while you change the corresponding import/export values in the CKDS at each control point.

Using alternate LU key-encrypting key names: To encipher the session key that is sent to the LU when a session is established, the control point (CP or SSCP) that owns the LU uses the LU master key in the cryptographic key data set (CKDS). When changing the LU master key, a similar change must also be made in the CKDS at the control point. Otherwise, the LU would have the new key and the CP would have the old key.

By specifying the CKEY operand VTAM uses an alternate key-encrypting-key (KEK) name that matches the LU master key while the primary master key is being changed in the CKDS. The CKEY operand can be specified on the MODEENT macro in the logon mode table or on the MODIFY SECURITY command. The KEK specified by the CKEY operand will be used during session activation until it is changed by the MODIFY SECURITY command or the LU is deactivated.

Use of the CKEY operand eases the administration process involved when changing the LU master key. Without the CKEY, both the master key at the LU and in the CKDS must be done simultaneously. This in turn reduces the likeliness of a session failure because the LU master key and CKDS key are not the same. The CKEY allows for continued use of the alternate master key until both the CKDS and LU have been updated with the new primary master key.

For VTAM to use the alternate LU KEK name while you change the LU master key, follow this general procedure:

1. Notify user of a specific time when the value of the default KEK for the LU will be changed.
2. At the specified time, change the default and alternate KEK in the CKDS to the appropriate values.

   Note: These changes only apply to subsequent sessions with the LU. Currently active or pending sessions use the old key values.

Refer to *z/OS Communications Server: SNA Resource Definition Reference* for details on the CKEY operand, and to *z/OS Communications Server: SNA Operation* for full details on the use of the MODIFY SECURITY command.

Specifying key encrypting key (KEK) names for LUs: You can specify a cryptographic key name to specify a unique cryptographic key name for an LU by specifying the CKEYNAME operand. If you do not specify CKEYNAME, the cryptographic key name defaults to the LU name. You can code the same value for multiple LUs on the CKEYNAME operand, which reduces the number of definitions needed in the cryptographic key data set.

The CKEYNAME operand can be coded on the LU definition statement in the following major nodes:

- Local SNA
- LU group
- Model
- NCP
- Switched

Using import and export CP/SSCP KEK names: By using unique import and export KEKs, VTAM may support other cryptographic products that implement CCA and provide you with a choice between ICSF and these other cryptographic products. VTAM appends a unique prefix and suffix to the CP/SSCP name that is used to reference import and export KEKs.

To change the export and import KEKs in the CKDS of a host, you must also change these keys in the other host. Instead of bringing down sessions to change the master keys, you can force VTAM to temporarily use an alternate name that matches the new LU master key name. The user should update the LU master key in the CKDS at the CP/SSCP as soon as possible after being notified that the keys are changed.

Note: For migration purposes, VTAM retries the CP/SSCP name without the suffix if a request fails because the KEK could not be found for the CP/SSCP name with the suffix. This alleviates having to change the CKDS.

Follow these steps when using alternate KEK names for CPs and SSCPs:

1. File an alternate set of export and import keys in each CKDS, in addition to the original export and import key-encrypting key names. You must file keys in the

CKDS of each host on each end of a cross-domain session or each end of an APPN session (for example, HOST1, HOST2):

HOST1

- IMPORTER.CP2
- EXPORTER.CP2
- IMPORTER.CP2.ALT
- EXPORTER.CP2.ALT

HOST2

- IMPORTER.CP1
- EXPORTER.CP1
- IMPORTER.CP1.ALT
- EXPORTER.CP1.ALT

2. Delete export keys in each host.

   HOST1

   EXPORTER.CP2

   HOST2

   EXPORTER.CP1

3. Issue the MODIFY SECURITY command to specify that VTAM use the .ALT KEKs.
4. Define new import and export values for the KEK fields in the CKDS of each host. VTAM starts all new sessions using the new values. Keep the alternate names in the CKDS until you are sure that all pending sessions using the alternate names have completed.

Message authentication

Message authentication is another form of security. Similar to data encryption to ensure data confidentiality, the message authentication data security feature:

- Provides services to insure the integrity of data for selected LU-LU sessions.
- Provides end-to-end protection of data, which does not require support from intermediate nodes.

Message authentication allows VTAM to determine if a message has been altered in transmission between the session partners. A code is attached to each message by the sender and verified by the session partner.

There are two methods for producing the message authentication code:

- Data encryption standard (DES) product that requires a cryptographic product to be active. Using this method, both cryptography and message authentication can be performed concurrently. Although the keyword is DES, if the session is setup to use triple-DES encryption, TDES24 will be used. The use of the term DES here does not mean only DES encryption can be used.
- Cyclic redundancy check (CRC), which creates a message authentication code using an internal VTAM algorithm. Using this method does not require a cryptography product to be active.

The APPL definition statement and MODEENT macroinstruction provide operands that allow you to define the message authentication support to be provided for a session. Code the following operands for each end of the session:

MAC Specifies whether authentication of data sent and received by the LU is required, conditional, or not supported.

MACLNTH Specifies the minimum length of the message authentication code attached to the message.

MACTYPE Specifies the type of message authentication checking (DES or CRC) to be used for the session.

Refer to *z/OS Communications Server: SNA Resource Definition Reference* for information on how to code the preceding operands.

Modifying and displaying MAC specifications

The MODIFY SECURITY and the DISPLAY ID commands allow you to display or change the message authentication specifications for an LU.

MODIFY SECURITY
Allows you to increase the security specifications as defined in the applications' resource definition. Any change you make must increase the level of the message authentication required for the application.

DISPLAY ID Allows you to display the current security data for the application. The DISPLAY ID has session status modifiers to show what a given session level is using for the level of encryption (DES = **D** or Triple-DES = **T**) encryption. Refer to *z/OS Communications Server: IP and SNA Codes*.

Refer to *z/OS Communications Server: SNA Operation* for details on using the preceding commands.

SLU authentication

During establishment of a cryptographic session, the secondary LU (SLU) verifies that the primary LU (PLU) is using the same cryptographic key. However, if the SLU does not reject a session when the partners are using different session keys, subsequent data flowing over the session is lost because proper deciphering of the data cannot be performed by the SLU. By specifying the CERTIFY=YES operand, the verification that both partners are using the same session key is performed by both the PLU and the SLU. If the session keys are not the same, the session is not established.

The CERTIFY operand can be coded for a resource in the following major nodes:
- Application program
- Local SNA
- Logical unit (LU) group
- Model
- Network Control Program
- Switched

In addition, the CERTIFY operand can be specified on the MODEENT macro in the logon mode table. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for additional information on coding the CERTIFY operand.

Notes:

1. SLU authentication is performed if CERTIFY=YES is specified for either session partner.
2. CERTIFY=YES is supported by both the 3174 and 3276 cryptographic features. To determine support for other devices, refer to the specific device documentation.

VTAM application security

You can specify a password on the APPL definition statement and require the application program to specify both its APPL definition statement name and its

password when it opens its access method control block (ACB). The authority of the application program to gain access to the network can be verified by comparing the password in the OPEN ACB macroinstruction of the program to the password specified on the PRTCT operand of the APPL definition statement.

The VTAM application OPEN ACB security facility provides additional resource access control. To perform security processing, VTAM invokes a security management product (such as RACF) through the security authorization facility for any application program that is not APF-authorized. The security management product determines whether an application program access to the network is approved.

To enable the VTAM application OPEN ACB security facility, register the application program with a class of VTAMAPPL (CLASS=VTAMAPPL) in a security management product that is capable of controlling authorization for VTAM application program execution (such as RACF Version 1 Release 9). VTAM bypasses any password checking if the security management product provides the resource access control. The application program is authorized to access the network based on either the approval of the security management product or the user-specified password check.

The authority of the LU to initiate a session with an application program can also be verified by requiring that the LU include a password in its logon. This password can be verified by the session authorization function of a session management exit routine, a session authorization exit routine, the application logon exit routine of the program, or the application program itself.

Confidential data

When data is transmitted between an application program and a logical unit, it passes through VTAM buffers. These buffers are allocated from common page-fixed buffer pools when I/O is being performed and from pageable buffers in user-protected storage when data is queued. An application program that is transmitting or receiving confidential data can have fixed buffers cleared if PROC=CONFTXT is specified in the appropriate node initialization block (NIB). If you use this option, only the name of the application program, the name of the logical unit, and the direction of the data flow are included in the trace records; confidential data is not included. A buffer trace of nonconfidential data includes the data.

Following are two ways of protecting application program data:

- The application program can ensure that confidential data in VTAM buffers within the host of the application program is cleared after the data is sent to the NCP or is moved into the application program address space. The application program does this by using the CONFTXT option in the node initialization block (NIB) associated with the session.
- An LU 6.2 application program, which provides a LOGON or SCIP exit routine to accept LU 6.2 sessions, can also use a confidential test option. You can code the CONFTXT=YES operand on the APPCCMD macroinstruction used to accept an LU 6.2 session (`APPCCMD CONTROL=OPRCNTL,QUALIFY=ACTSESS`). If you code CONFTXT=YES, VTAM clears its buffers that have been used to hold application program data prior to returning them to VTAM buffer pools for reuse.

Note: TSO also provides a CONFTXT option that allows you to protect confidential data. For information on CONFTXT and TSO, see “Security” on page 726.

Logon mode parameters

If you have an application program that acts as the secondary end of the session, you might need to create new entries or supplementary tables, as follows:

- If the application program is the secondary logical unit, you might want to specify session protocols. You can direct VTAM to the proper session parameters by specifying the logon mode table entry (through the DLOGMOD operand) and the logon mode table (through the MODETAB operand) on the APPL definition statement of the program.
- If the application program is intended to be in session with specific devices or application programs and it requires certain protocols, you might have to create new logon mode table entries associated with the devices.

Both an APPN Class of Service and a subarea Class of Service, specified in the logon mode entry, can be associated with each session. The APPN Class of Service determines the route that is used for APPN links, and the subarea Class of Service determines the virtual route that is used for a session. Therefore, you should understand the Class of Service requirements for your application programs. Considerations such as the importance of the function provided by the application program, the nature of the data exchanged with the logical units, and the expected response times contribute to determining the Class of Service needed for sessions with the application program. The Class of Service names must then be specified in the logon mode table entries used for LU-LU sessions. For more information about logon mode entries, see "Selecting session parameters for the logon mode table" on page 292.

Using user variables (USERVAR)

The VTAM user variable (USERVAR) maps a generic application program name used in a terminal logon to the name of a specific application program, based on the value of the USERVAR. This function is used by IMS and CICS XRF to map user logons to the IMS or CICS application program that is currently active.

You can also use USERVARs in other ways, such as to facilitate migration from one application program release to another. The VTAM user variable:

- Enables your installation to use the USERVAR function without having to rely upon NetView command lists or manual procedures to propagate USERVAR updates. VTAM communicates USERVAR values across domains in the same or different networks.
- Enables your application programs to use USERVAR names across VTAM application programming interface (API) in place of logical unit names without requiring code changes.
- Enables your terminals to take advantage of USERVAR translation without your having to code interpret tables for them.

The VTAM application programming interface (API) allows application programs to use USERVAR names across the API in place of LU names. Because USERVAR can be used across the API, many application programs that participate in sessions with an XRF-capable application program (such as IMS or CICS) can remain unchanged when USERVARs are used.

Also, because VTAM determines whether a name refers to an LU or to a USERVAR and performs the appropriate translation automatically, you do not have to code an interpret table to specify that a name used in a particular logon is actually a USERVAR.

Note: The generic name used by USERVAR should not be used by ALIAS. An alias CDRSC should not exist in a host where USERVAR mapping is used unless the host is in the same network as the real resources.

Application workload balancing with USERVAR

A group of applications can be given a common or generic name. VTAM uses this generic name, known as a USERVAR, to associate a logon request with the currently active member of this group. This capability can be used for workload balancing and preparing for the planned takedown of a host. USERVAR values can be managed by the user as well as by VTAM.

For example, assume that the USERVAR, IMS, has been defined, and that there are two IMS application programs: IMS1 and IMS2. The generic name (USERVAR) can be assigned the value of either IMS1 or IMS2, based on which system is active. If the USERVAR for the IMS application programs is set to IMS1, for instance, then any session requested with IMS is routed to IMS1. If IMS is set to IMS2, then new sessions are routed to IMS2. Therefore, if the workload going to IMS1 is very heavy, the IMS USERVAR can be set to IMS2 and subsequent sessions are routed to IMS2.

The following commands display and modify the VTAM application names associated with USERVARs:
- DISPLAY USERVAR
- MODIFY USERVAR
- DISPLAY ID
- DISPLAY SESSIONS

For further information on these commands, see *z/OS Communications Server: SNA Operation*.

Classes of USERVARs

VTAM supports the following two classes of USERVARs:
- User-managed USERVAR
- Automatic (or VTAM-managed) USERVAR

User-managed USERVAR

This class is explicitly set using the MODIFY USERVAR operator command. The MODIFY USERVAR command can be issued either manually by a human operator or automatically by a NetView command list or a VTAM application program (as is done by IMS or CICS for XRF).

VTAM does not attempt to change the value of a USERVAR that has been set explicitly using the MODIFY USERVAR operator command, so existing command lists and procedures for propagating USERVAR changes across domains in a network should continue to work as is. However, a MODIFY USERVAR command can be used to delete a user-managed USERVAR to subsequently allow that USERVAR name to be managed by VTAM.

At least one host in the network must have a user-managed USERVAR from which other hosts can obtain the USERVAR value and type. In XRF situations, IMS and CICS both set the USERVAR value at the site of the XRF active application program.

Notes:

1. For XRF application programs, VTAM deletes user-managed USERVARs when the application program terminates or enters a specific state, such as CONCT or INACT.

2. In a subarea-only network, the user-managed USERVAR can be defined on the VTAM that owns the real destination resource, on the VTAM where the session originates, or on any VTAM along the session path. However, because there might be attempts from many different VTAMs to establish sessions with the destination resource using the USERVAR name, the best results are achieved when the user-managed USERVAR is defined on the VTAM that owns the real resource, or on a VTAM that is as close to the owning VTAM as possible.
3. In an APPN network, or in a mixed APPN and subarea network, the location of the user-managed USERVAR is more restrictive than in other networks. If the real resource resides on a VTAM end node (EN), a network node (NN), or a migration data host (MDH), the user-managed USERVAR must be defined on that VTAM (the owning VTAM). Therefore, when the real resource is on an EN, define the USERVAR on the EN but do not define it on its network node server (NNS). If the real resource resides on an interchange node (ICN), or on a VTAM located in or through a subarea network attached to an ICN, the user-managed USERVAR must be defined on the ICN or on one of the VTAMs along the subarea portion of the session path.

   When the MODIFY USERVAR command is issued on an EN, the USERVAR name is registered to the NNS of the EN unless the value UVEXIT=NO is coded (or is the default value) and the real resource name is either a dynamic CDRSC or is not defined.

Automatic (or VTAM-managed) USERVAR
This class is created automatically by VTAM in one domain as a copy of a user-managed USERVAR of the same name in another domain. You do not have to do anything to allow VTAM to create automatic USERVARs. When a VTAM SSCP sends a cross-domain session-initiation request to another domain and receives a reply indicating that the name specified in the original session request is really a USERVAR with a particular value and not the name of a logical unit, the SSCP creates an automatic (VTAM-managed) USERVAR to save the information. The cross-domain session request is then resent using the real LU name, which was passed back in the original reply.

Types of USERVARs

Changes to the value of a USERVAR are not propagated by VTAM at the time of the change. Instead, when a domain using a USERVAR to establish a session encounters a specific USERVAR for the first time, it uses an adjacent SSCP table to perform a cross-domain search to obtain the USERVAR value. Subsequent attempts to use the same USERVAR can result in a repeat of the cross-domain search to update the USERVAR value, depending on the value of the TYPE operand. The TYPE operand is specified on the MODIFY USERVAR command when the USERVAR is created. The TYPE operand is copied, along with the USERVAR value, when an automatic USERVAR is created in another domain. Following are the types of USERVARs:

Static USERVAR
The value is assumed not to change. It is saved as an automatic USERVAR upon initial use but is not updated automatically using the cross-domain search process (unless the existing copy of the automatic USERVAR is explicitly deleted by the operator).

Dynamic USERVAR
The value is expected to remain relatively stable but change occasionally

(for example, when an XRF takeover occurs). To establish a cross-domain session to an application program identified by a dynamic USERVAR, VTAM must recheck the USERVAR value after every abnormal session termination involving the application program that it references.

DYNAMIC is the default value for the MODIFY USERVAR command and is recommended for use within XRF complexes.

Volatile USERVAR
The value is expected to change often. Rather than creating an automatic copy of the USERVAR, a remote VTAM repeats the cross-domain search to reestablish the USERVAR current value each time it is referenced to establish a session. Due to the additional search flow that is necessary for each session establishment, use of volatile USERVARs might have a substantial impact on the time required for session establishments.

Note: If a USERVAR value is allowed to vary over multiple active application programs, LU 6.2 communication works only for a recovery environment. See "LU 6.2 in an XRF Environment" on page 408.

Processing USERVARs

Assume that an XRF application program, such as CICS, runs on Systems A and B of the sample network shown in Figure 106. CICSA is the specific VTAM APPLID of the CICS running on System A, and CICSB is the APPLID of the CICS running on System B. The terminal network is owned by the VTAM in System C (this is a communication management configuration, or CMC).

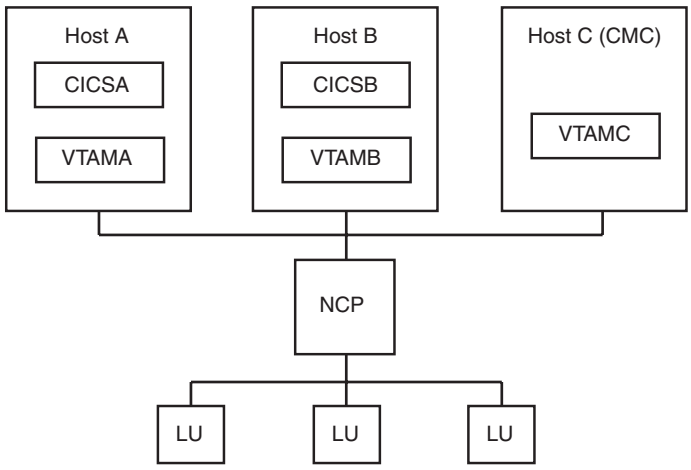


Figure 106. Sample extended recovery facility network

1. When CICSA is started as an active CICS application program, it issues an `F NET,USERVAR,ID=CICS,VALUE=CICSA` command, requesting that VTAMA create a dynamic (by default), user-managed USERVAR with a value of CICSA. Because CICSB is started as an XRF alternate, it does not request that VTAMB create a USERVAR.
2. Logon requests from a terminal are processed by VTAMC, which is the controlling VTAM for the terminal network. The first time a terminal attempts to log on to generic application name CICS, VTAMC processes the logon request as follows:

- Because VTAMC does not recognize the name CICS yet, it treats the request like any other cross-domain logon request and sends a cross-domain session request to other VTAMs listed in an ADJSSCP table looking for CICS. At this point, VTAMC does not know whether CICS is the name of a USERVAR or a real VTAM APPLID.
- When VTAMA receives the session request, CICS is identified as the name of one of its own user-managed USERVARs. It replies to VTAMC, indicating that CICS is really a dynamic USERVAR whose value is currently CICSA.
- VTAMC uses the information it received from VTAMA to establish a cross-domain session with the application program by sending another session request using that specific application name, CICSA. In addition, VTAMC creates an automatic USERVAR to save the USERVAR information for future use.

3. For subsequent terminal logons to CICS, VTAMC uses the information it saved previously in the automatic USERVAR to translate the requests immediately to CICSA. VTAMC continues to translate CICS requests directly to CICSA without repeating the cross-domain search process because CICS is a dynamic USERVAR. This process continues until either an XRF takeover occurs or VTAMC discovers that it is otherwise unable to establish sessions with CICSA.
4. If CICSA fails and CICSB takes over and becomes the new active application program, CICSB issues an `F NET,USERVAR,ID=CICS,VALUE=CICSB` command, requesting that VTAMB create a new user-managed USERVAR with the value CICSB. This means that VTAMB is now the VTAM to identify session requests searching for CICS. VTAMB then responds, indicating CICSB is the current value of the USERVAR.

   Each VTAM that was notified that its XRF sessions were switched deletes any automatic USERVARs that were used to create those sessions. This means that the next logon for CICS does not find the USERVAR and goes through the cross-domain search process (step 2 on page 402) to determine the new value of CICS. In addition, the VTAM associated with the failing application program (VTAMA) deletes its user-managed USERVAR for CICS and prevents any further logons to CICSA to ensure that logons do not get directed to the failing application program.

   Note: If a resource submits a network-qualified session request with a generic resource name, USERVAR processing occurs in the domain of the resource if one of the following is true:
   - The NETID specified on the session request is equal to the NETID of the USERVAR value.
   - The NETID specified on the session request is equal to the NETID of the host and the USERVAR value is not network-qualified.

USERVAR propagation and routing

NetView command lists are not required to propagate USERVAR modifications to a host in which VTAM can manage the USERVAR values; however, existing NetView USERVAR propagation command lists continue to work as they did before.

Note: VTAM can manage USERVAR values at a given host only if it and all hosts on the session establishment path have VTAM Version 3 Release 2 or later and have installed the appropriate PTF for VTAM Version 3 Release 2 that provides this USERVAR support.

VTAM searches for the value of a USERVAR through ADJSSCP table routing for a cross-domain session request. Therefore, if you have not already done so, you

might need to define adjacent SSCP tables for searches to find the VTAM having the user-managed USERVAR. For information about this, see “Adjacent SSCPs” on page 541.

Generally, VTAM does not change user-managed USERVARs; they are viewed as being the responsibility of the user (or application program) that created them. However, in XRF situations, VTAM reduces the problem of session requests going to a failing XRF-active application program by deleting the user-managed USERVAR that references the failing application. VTAM deletes the user-managed USERVAR only at the site of the failing application program. If you have created a user-managed USERVAR at another host in the network, you need to make sure that USERVAR is deleted or that it continues to direct session requests to the failing application program.

For the VTAM having a user-managed USERVAR to respond to a session initiation request using that USERVAR, VTAM must know certain information about the LU that its value references (for example, the name of the SSCP that owns the LU). The necessary information is available if the referenced LU is owned by the host having the user-managed USERVAR, or if a CDRSC exists for the LU on that host. Otherwise, VTAM responds to the session initiation request as though the name were not found to allow the ADJSSCP routing to continue.

Defining your network with USERVARs

When you define your network resources for XRF, you should consider the following:

- If you allow dynamic CDRSCs and trial and error rerouting, you do not need to predefine a CDRSC for the USERVAR name or CDRSCs for the active and backup application program.

  Note: Referencing dynamic CDRSCs with USERVARs can cause situations in which session initiations fail intermittently, because dynamically defined CDRSCs can be deactivated and deleted by VTAM, depending on the dynamic CDRSC timer value (CDRSCTI) and whether there are any active sessions.
- If you do not allow dynamic CDRSCs to be created in the network, you need to predefine a CDRSC for the USERVAR name, a CDRSC for the active application program, and a CDRSC for the backup application program. If you do predefine a CDRSC with the USERVAR name, you should not specify the cross-domain resource manager (CDRM) name because the USERVAR may not always be managed from the same SSCP. However, you should define the CDRSC under the network in which the USERVAR resides to reduce the amount of time for the ADJSSCP search process.

The following applies regardless of whether you use USERVARs with XRF:

- You can use USS tables, interpret tables, or both with USERVAR translation; the USS and interpret processing is done first and USERVAR translation is done using the output of those processes.
- You are not required to define interpret table entries for terminals to use the USERVAR function. If you have already done so, however, you do not have to remove them.
- If the application is cross-domain, the USERVAR name must be defined as a CDRSC using either static or dynamic definitions and must be unique within the network.

- The OLU host of a session initiation as well as the host that is resolving the USERVAR name must support name translation. For VTAM 4.1 and earlier, the TRANSLAT start option must include USERVAR. In VTAM 3.4.2 and later, RACALIAS must include USERVAR in ISTRACON.

Dynamic USERVAR update session failure

Whenever an abnormal session termination occurs for a session that was initiated using a USERVAR, VTAM deletes the USERVAR if it is VTAM-managed and dynamic. When the next session request arrives at any host other than the XRF primary host, VTAM repeats the search for the USERVAR and saves the new current value in an automatic USERVAR.

The following kinds of session termination are considered normal because they do not cause the resetting of a dynamic USERVAR:

- Requests for conditional or unconditional termination, such as the following:
  - Operator issued the VARY NET,TERM,TYPE=COND|UNCOND command.
  - SLU application program issued the TERMSESS OPTCD=COND|UNCOND command.
  - Terminal issued (using USS) the LOGOFF TYPE=COND|UNCOND command.
- Session setup failures due to the following sense codes:
  - 08210002
  - 088A0003
  - 088A0004
- LU-LU session terminations (UNBIND) for the following reasons:
  - Normal end of session (TYPE=X'01')
  - CLSDST PASS (TYPE=X'02')
  - Session parameters that are not valid (TYPE=X'06')
  - Other UNBINDs between X'01' and X'06'
  - Format or protocol error (TYPE=X'FE')

A session termination for any reason other than those reasons described previously is handled as an abnormal session termination and causes the deletion of any VTAM-managed dynamic USERVAR associated with that session. Following are examples of abnormal terminations:

- Requests for forced termination which, like requests for conditional or unconditional termination, can come from operators, SLU application programs, or terminals.
- LU-LU session terminations, including the following:
  - VR INOP (TYPE=X'07')
  - Route extension INOP (TYPE=X'08')
  - LU failure (TYPE=X'0C')
  - Cleanup termination, such as V NET,INACT (TYPE=X'0F')
  - Gateway node cleanup (TYPE=X'11')
- Session setup failures other than those previously described (such as failure to find the destination LU after using a dynamic USERVAR to determine the application program name). In such cases, VTAM deletes the dynamic USERVAR and retries the session initiation using the USERVAR name to determine the current value of the user-managed USERVAR.

Whenever the active application program in an XRF complex fails, notification of the session failures is sent to the SSCPs of the session partners, thereby causing them to reset their dynamic USERVARs. In the host of the XRF active application program, VTAM deletes all USERVARs that reference the failing application

program to prevent future USERVAR searches from using a USERVAR that references the failing application program.

If a backup session already exists, an alternate application program sends a switch request to the NCP to takeover the sessions. The XRF takeover can be initiated before VTAM even knows that the active application program is failing. The VTAM first awareness of such a takeover is the receipt of a session ended notification having a specific session termination reason code (X'13'), which indicates that the primary session terminated due to an XRF switch. At a remote SSCP, such session terminations are handled as abnormal session terminations. At the SSCP of the failing active application program, however, the SSCP also does the following:

1. Marks the application program as "not enabled for sessions" to prevent any further session establishments
2. Deletes the USERVAR that is associated with the session that is ending

Generic resources function

Whereas a USERVAR is assigned to only one resource at a time, the generic resources function allows the assignment of a generic resource name to a group of active application programs that all provide the same function. The generic resource name is assigned to multiple active application programs simultaneously, and VTAM automatically distributes sessions among these application programs rather than assigning all sessions to a single resource. Thus, session workloads in the network are balanced. Session distribution is transparent to users; an LU initiates a logon request using the generic resource name and need not be aware of which particular application program is actually providing the function.

The generic resources function also increases application program availability, as each active application program that uses a given generic resource name (a generic resource member) can backup other generic resource members. Thus, no single application program is critical to resource availability. When a generic resource member fails, an LU can reinitiate its session using the same generic resource name. VTAM resolves the session initiation to one of the other generic resource members. Because the user is unaware of which generic resource member is providing the function, the user is less affected by the failure of any single generic resource member.

In the same way, an application program can be added as a generic resource member and accessed using the same generic resource name. Because the user is unaware of which generic resource member is providing the function, an application program can be added as a generic resource member with little or no impact to the application user. At the same time, the additional application program provides an immediate improvement in performance and availability since the workload is now shared by an additional application program.

Use of generic resources requires a coupling facility structure. See "Generic resources" on page 448 for additional information concerning generic resource requirements.

High availability using extended recovery facility

You can identify an application program as capable of operating in an extended recovery facility (XRF) environment. Two IBM application programs, Information Management System (IMS) and Customer Information Control System (CICS), support an XRF environment.

XRF provides an alternate application program when certain host components fail. It reduces the impact of planned outages or outages caused by failures in VTAM, MVS, IMS, CICS, or the host processor on selected users of IMS, CICS, or other application programs that use XRF functions. The types of failures that cause the takeover of an XRF session are determined by how IMS, CICS, or the application program using XRF is defined. XRF might not reduce the impact of failures in the NCP or in telecommunication lines, depending on which NCP or line fails. However, the XRF function in CICS or IMS does not automatically perform a switch to the alternate application program under these conditions. Therefore, either the operator must switch from the active to the alternate system, or you can use NetView automation to perform the switch.

To enable high availability, use the HAVAIL operand on the APPL definition statement.

VTAM, in conjunction with the application program, establishes a primary session from the currently active application program and a backup session from the alternate application program to the terminal. The NCP for a terminal in an XRF environment maintains two sessions through two VTAMs: one to a primary XRF application program and one backup session to the alternate XRF application program. The XRF application programs, in conjunction with VTAM and NCP, switch to the backup session when some component fails. The active and alternate application programs must be in the same network. However, a terminal participating in an XRF session can be in a different network than either of the application programs.

One application program acts as the active application program, and the other acts as the alternate in case the active application program or another component in the active host fails. Both application programs are identified by a single name that is associated with the currently active application program.

Although one VTAM in a single-processor environment can support both a primary and a backup session, this configuration does not provide the availability characteristics usually associated with an XRF environment.

XRF is also supported for application programs that use LU 6.2 sessions to communicate with other applications or logical units in peripheral nodes attached to an NCP. For further information, see “LU 6.2 in an XRF Environment” on page 408.

Security features in an XRF environment

An XRF application program can establish cryptographic sessions and message authentication sessions with other LUs. Cryptography protects data passing over lines by permitting enciphering and deciphering of data for LU-LU sessions. Message authentication provides a message authentication code used to validate the contents of the data.

The XRF application program must reside in a host that has the IBM Integrated Cryptographic Service Facility/MVS (ICSF/MVS) or a compatible cryptographic product installed and active.

The following references are used with compatible cryptographic products::

PCF/CUSP
Refers to any cryptographic product that is compatible with PCF/CUSP.

CCA Refers to any product that is compatible with Common Cryptographic Architecture (CCA).

Many PCF/CUSP compatible cryptographic products must be started before starting VTAM. They also must be started before you activate an external CDRM or CP for which CROSS statements have been defined in the cryptographic key data sets. This is not necessary for the VTAM Integrated Cryptographic Service Facility.

The XRF application program can specify that cryptography is either selective (specified by the session end) or required (session ends must support cryptography).

To enable this facility, code the following:

- ENCR operand on the APPL definition statement for the XRF application program.
- ENCR operand on the LU definition statement for any logical units that will be session ends.
- ENCRTYPE keyword can be coded on both the APPL and LU. To enable TDES24 encryption, ENCRTYPE=TDES24 must be coded; otherwise the ENCRTYPE will default to DES.

  Note: A corequisite of NCP Version 7 Release 8 is required for XCF/Crypto with triple-DES operation.
- Cryptographic keys in the cryptographic key data set.

For more information about coding the ENCR operand, see "Cryptography facility" on page 386. For more information about coding cryptographic keys, see Appendix E, "Cryptographic keys," on page 755.

LU 6.2 in an XRF Environment

USERVAR process of using a generic name to represent a service does not work in an active LU 6.2 environment because LU 6.2 cannot tolerate LU name substitutions. However, support for LU 6.2 in an XRF environment allows generic name substitution, but only in a recovery environment.

The LU 6.2 application uses a USERVAR ID during recovery to transfer its users from one application program to a backup application program. This provides faster backup for failing applications.

To use this support, you need to identify the USERVAR to be used for LU 6.2 communications.

Be aware of the following restrictions:

- You can assign only one USERVAR ID to an LU 6.2 application program.

- You can have only one LU 6.2 application program as the primary application program to provide a service.
- The operator can modify the USERVAR ID for LU 6.2, but the command must be issued from the host system of the application program providing the service.
- If an incorrect USERVAR ID is assigned to an active LU 6.2 application program, the ACB must be closed and then reopened with the correct USERVAR ID.
- Even sessions that request a bind using the real LU name must still carry the USERVAR ID.

Persistent LU-LU sessions

Application programs supporting persistent LU-LU sessions can reestablish LU-LU sessions at the level before the failure without the session establishment flow. There are two levels of support for persistent LU-LU sessions.

- Single node persistent sessions (SNPS)

  If an application program fails or is brought down, VTAM can retain active sessions, allowing the same or another application program to reconnect to the sessions, avoiding the need to reestablish the sessions. If an application program fails, it can reconnect to the retained sessions when it recovers. Also, another application program can take over the sessions. SNPS takeover processing allows an application program to take over the sessions of an active application, although the active application is able to indicate, through SETLOGON PERSIST, that such requests are rejected by VTAM.
- Multinode persistent sessions (MNPS)

  Multinode persistent LU-LU sessions (MNPS) extends the single node persistent session support to application, VTAM, operating system, and hardware failures. Applications can be restarted on an alternate VTAM and their sessions restored after a VTAM failure. Applications can also restart on the same VTAM after VTAM has been restarted.

  MNPS planned takeover processing differs from SNPS takeover processing in that the application is required to fail, or issue CLOSE ACB, while enabled for persistence in order to be eligible to move to another VTAM.

  An alternative to MNPS planned takeover, called MNPS forced takeover, works closer to SNPS takeover processing, in that the target application does not have to fail or issue a persistent CLOSE ACB prior to the MNPS takeover attempt being made. However, unlike SNPS takeover, MNPS forced takeover requires that the taking over application indicate on the ACB macroinstruction that any OPEN ACB attempted using this ACB is an MNPS forced takeover attempt; likewise, the target application must indicate on SETLOGON OPTCD=PERSIST that it will accept MNPS forced takeovers. SNPS takeover processing does not require any special communication from the application, but the capability does exist for the application to indicate that SNPS takeover is not permitted.

When an application program is persistent enabled, VTAM saves data sent and received by the application program and other status information. VTAM uses this information to reestablish the session after a failure has occurred, or as part of planned or forced takeover processing.

For an application to be capable of persistence, code PERSIST=YES on the ACB macroinstruction. To enable persistence, the application program must specify OPTCD=PERSIST on the SETLOGON macroinstruction. The application program can also disable persistence by issuing SETLOGON OPTCD=NPERSIST at any time. An application that wants to use multinode persistent session support must have PERSIST=MULTI coded on its application definition statement.

For an OPEN ACB to be considered for MNPS forced takeover processing, include FORCETKO=YES on the ACB macroinstruction (along with PERSIST=YES).

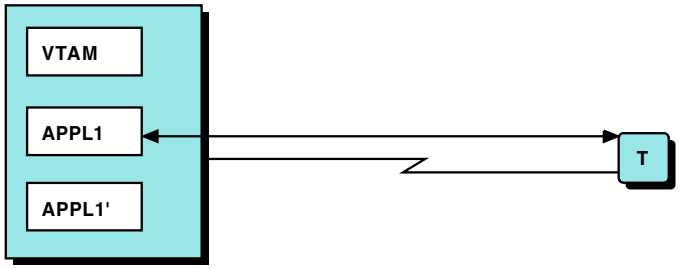
The application is permitted to specify the following types of forced takeover requests it will accept from other application images of the same name:

- To indicate that the application will accept MNPS forced takeover requests from other instances of the application, code PARMS=(FORCETKO=ALL) or PARMS=(FORCETKO=MULTI) on the SETLOGON OPTCD=PERSIST macroinstruction.
- To indicate that the application will accept SNPS forced takeover requests from other instances of the application, code PARMS=(FORCETKO=ALL) or PARMS=(FORCETKO=SINGLE) on the SETLOGON OPTCD=PERSIST macroinstructions.
- To prevent any forced takeover from being processed for the application, code PARMS=(FORCETKO=NONE) on the SETLOGON OPTCD=PERSIST macroinstruction.

Rule: Issuing SETLOGON OPTCD=NPERSIST does not affect the setting of the FORCETKO capability of the application.

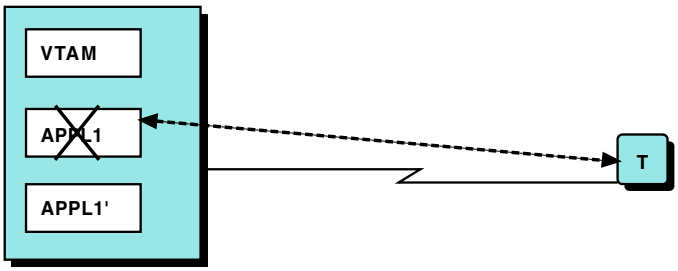
Single node persistent sessions

When a persistence-enabled application program fails, VTAM retains the sessions, saves the allocated resources and control blocks, and shields the network from knowledge of the application program failure. VTAM stores the incoming data so that the network views the session as active but not currently responding. When the failed application program restarts or another application program takes over, VTAM reconnects the sessions. The backup of a persistence-enabled application program is shown in Figure 107.



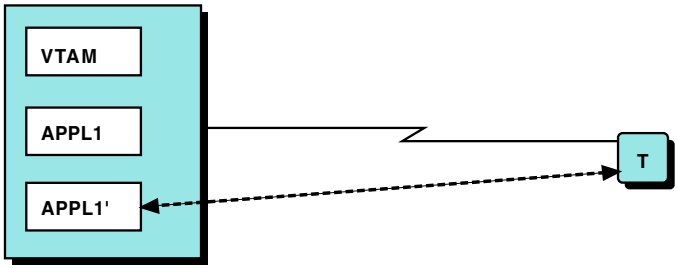
1. APPL1 has initiated a session with terminal T. The session is persistent capable and enabled. APPL1 specified persistence on both the OPEN ACB (PERSIST = YES) and the SETLOGON (SETLOGON OPTCD = PERSIST) macroinstructions.

Figure 107. Application program backup using persistent LU-LU sessions (Part 1 of 3)



2. APPL1 fails. The session is pending recovery. If no timer is specified (PSTIMER), the session can remain pending recovery indefinitely. However, the retained session can be terminated by a VARY NET INACT command.

Figure 107. Application program backup using persistent LU-LU sessions (Part 2 of 3)



3. The operator or program operator detects that APPL1 has failed and uses APPL1' as a backup. APPL1' issues an OPEN macroinstruction with the same ACB name APPL1 issued. It then restores the session. Takeover is transparent to terminal T.

Figure 107. Application program backup using persistent LU-LU sessions (Part 3 of 3)

For more information on enabling an application program to support persistent LU-LU sessions, refer to the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* and *z/OS Communications Server: SNA Programming*.

The HALT and VARY INACT,TYPE=FORCE commands override single node persistent sessions.

VTAM common network services

When you use VTAM common network services (VCNS) with the ES/9000 series communication adapter to send data across an X.25 packet switched data network, specify the VCNS=YES operand on the APPL definition statement. This allows sharing the physical connectivity of the communication adapter between SNA and VCNS data traffic.

You can use VCNS with the IBM 3172 Nways interconnect controller to send data to a local area network. Specify VCNS=YES on the APPL definition statement to allow sharing the physical connectivity of the communication adapter between SNA and VCNS data traffic.

Cross-memory application programming interface (API)

Sessions can be established by an application program running in an address space other than that which opened the application ACB of the program. This enables the application program to be represented to VTAM by one ACB (that is, APPL definition statement) but span multiple address spaces. The application program can also issue VTAM requests to send and receive data while running in a different address space than the one in which the session was originally established.

For information on enabling an application program to support cross-memory API, refer to *z/OS Communications Server: SNA Programming*.

Allocating private storage

All incoming data is queued in a VTAM-owned ESA data space, not in private storage. The MAXPVT operand therefore has no effect on data queued for an application.

Communicating with start-stop devices

When communicating with a start-stop device that is supported by the Network Terminal Option (NTO) or the X.25 NCP packet switching interface (NPSI), special data stream characteristics of the non-SNA terminal might need to be specified for use by an application program. You indicate these characteristics by specifying the appropriate value on the TERM operand of the PU or LU definition statement that defines the virtual physical or logical unit. The application program can get information about these characteristics by using the INQUIRE OPTCD=DEVCHAR macroinstruction. It can also get it from the CINIT RU, which is queued when a logon is received from one of these devices.

LU 6.2 application programs

LU 6.2 enables a VTAM application program to communicate with nonhost application programs on an advanced program-to-program communication (APPC) level. Without LU 6.2, the nonhost program usually must emulate a terminal when communicating with a VTAM application program, using protocols that are not well-suited for application programs and that diminish the nonhost application program ability to handle errors and to initiate and end communications.

LU 6.2 protocols are designed to optimize use of sessions and improve the efficiency of data transfer through the network. LU 6.2 can also simplify the program logic needed to establish communications with other application programs and to send and receive data. Though VTAM support simplifies the task of writing application programs that use LU 6.2, be aware that enabling VTAM support requires some LU 6.2 functions to be implemented by an application program.

You can write or modify host application programs to use LU 6.2 protocols throughout the network.

Enabling LU 6.2 support

You enable LU 6.2 support for an application program by specifying the APPC=YES operand on the APPL definition statement. The application program must use the APPCCMD macroinstruction to send and receive data on LU 6.2 sessions and must allow VTAM to manage LU 6.2 session establishment and

termination. This means that an application program cannot use non-LU 6.2 VTAM macroinstructions such as OPNDST, SEND, or RECEIVE on LU 6.2 sessions; however, the same application program can use non-LU 6.2 macroinstructions to communicate on non-LU 6.2 sessions. For information on macroinstructions for LU 6.2 sessions, see the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Existing application programs that have implemented LU 6.2 without VTAM APPCCMD support (and consequently use the restricted non-LU 6.2 macroinstructions) should not have APPC=YES on their APPL definition statement. They do not work if it is included. To take advantage of VTAM support, they must be migrated to use the APPCCMD macroinstruction.

LU 6.2 sessions

Following is information about how LU 6.2 sessions operate.

Session establishment and termination

VTAM handles the tasks of session establishment and termination for LU 6.2 sessions for those application programs that use LU 6.2 support. Application programs explicitly issue conversation establishment and termination requests. VTAM automatically sets up and takes down sessions in response to these requests, according to the session limits the application programs have requested and negotiated. The application program's main involvement with session management is in establishing limits on the number of sessions that it can have with another LU in the network.

Conversations

Sessions between LU 6.2 session partners are used by transaction programs that operate in each partner LU to perform data exchange. Rather than sessions, the basic communication link between logical units using LU 6.2 protocols is a conversation. A conversation is the exclusive use of a session between two partner LUs for some unit of work. Sessions between LU 6.2 session partners are serially reusable. When one conversation ends, another conversation can use that same session between the partner LUs.

Sending and receiving data

If a session is full-duplex, conversations on that session can be full-duplex or half-duplex. If a session is half-duplex, conversations on that session must be half-duplex. In full-duplex conversations, an LU can send and receive data simultaneously. In half-duplex conversations, only one LU can send data at a given time. One transaction program in the conversation must be designated as the sender and the other as the receiver. Sessions between logical unit partners are designated as contention-winner or contention-loser sessions. If the two LU partners attempt to use a session simultaneously, the designated contention winner for the session has priority for the use of the session. An application program can allocate a conversation over a session for which it is designated the contention loser, but the logical unit partner must give the application program permission.

Session characteristics

Session characteristics are determined by the mode name associated with the session. Each session between LU 6.2 partners is related to a specific mode name. The mode name represents a set of characteristics associated with that session. All other sessions between the LU 6.2 partners that use that mode name are assumed to have the same set of session characteristics.

Session initiation using resource verification reduction

In an APPN network, LU-LU session initiation requires that the control point serving the requesting LU obtain partner LU location and route information to establish the session. For example, when the control point in the end node of the requesting LU receives the session initiation request, it forwards a Locate request to its network node server to locate the partner LU. If the resource is known to the network node server, the network node server sends a verification search directly to the resource. If the resource is not known to the network node server (that is, if the resource information has not been previously cached), the network node server conducts a network-wide search for the resource.

After information about a resource has been cached, the resource verification reduction function can be used to eliminate the verification search for the partner LU. Either the network node server of the requesting LU or the network node server of the partner LU may use the cached information to build a reply on behalf of the destination LU when resource verification reduction is requested. As long as the cache information is accurate and complete, session setup time is reduced. The maximum amount of time cached information will be used for resource verification reduction is controlled by the VFYREDTI start option. If the information is older than the value of the VFYREDTI start option value, resource verification will be performed upon receiving the next request for the resource. For more information on the VFYREDTI start option, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Cached information that is not accurate may cause the session setup to fail. When this occurs, VTAM will redrive the session initiation request, forcing verification of the resource.

If the requesting application has a LOGON exit, it is driven for a second time when the second initiation request is successful. However, the CINIT information for the second request contains different session information than that of the initial LOGON exit invocation. Thus, an application program with a LOGON exit must be able to have its LOGON exit driven multiple times as a result of one logical request (for example, an ALLOCATE). Application programs that support having their LOGON exits driven multiple times must indicate this support, by coding the application capabilities vector with the multiple logon exit support indicator on, to take advantage of resource verification reduction. For more information on how to code the application capabilities vector, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Use of the resource verification reduction function is limited to application programs using the VTAM LU 6.2 programming interface. In addition, the application program must use network-qualified names for session requests. Because some session requests require that the destination LU be contacted, verification reduction cannot be performed for sessions using:

- Application supplied dial parameters (ASDP)
- Cryptography
- Generic resources
- Secure Class of Service
- USERVARs

The benefit of resource verification reduction is limited for session requests traversing subnetwork boundaries (border nodes and interchange nodes). Resource verification reduction can eliminate a verification search across only one of the traversed subnetworks.

Controlling the use of resource verification reduction

The use of resource verification reduction is controlled by the VFYRED and VFYREDTI start options.

- Use of the VFYRED start option determines whether resource verification reduction is allowed for all LU 6.2 session initiation requests using the APPCCMD application programming interface.
- Use of the VFYREDTI start option indicates the maximum amount of time that cache information may be used before a resource location must be verified.

Resource verification reduction is suppressed for application programs with a LOGON exit unless the application program indicates support of multiple logon exit through the application capabilities control vector. Table 43 shows when the resource verification reduction function is allowed and requested by the requesting application.

Table 43. Resource verification reduction matrix

VFYRED start option value	LOGON exit included	Multiple logon indicator set on	Verification reduction allowed
YES	YES	YES	YES
YES	YES	NO	NO
YES	NO	N/A	YES
NO	N/A	N/A	NO

LU 6.2 session limits

VTAM maintains a set of session limits for partner logical units for each mode name used. These session limit values are used if a partner LU attempts to negotiate a change in the current number of sessions. If VTAM receives a request from the other partner LU to change the number of sessions for its LU 6.2 logical unit, VTAM uses the session limit values to determine whether to accept or reject the requested change. Note that the partner LU may be unable to accept increases in the session limits. Therefore, the end result of any change requests can still mean that the session limits do not increase.

If a session is available for use by a conversation, the process of starting and ending a conversation requires fewer system resources than starting and ending a session. Ideally, therefore, application programs should set up their session limits so that VTAM can establish a number of sessions with other LUs that will be relatively long-lived. Applications can then treat these sessions as resources that support conversations.

Requesting session limit changes: Application

An LU 6.2 logical unit requests that its partner LU alter the number of sessions for a given mode name by sending a change number of sessions (CNOS) request. A VTAM application program can request that its partner LU change the number of sessions. The application program does this by issuing an APPCCMD macroinstruction with the CONTROL=OPRCNTL and QUALIFY=CNOS operands specified.

Requesting session limit changes: Operator

You can enable the operator to modify CNOS parameters by specifying OPERCNOS=ALLOW on the APPL definition statement. The parameters can then be changed by a MODIFY CNOS command, which functions the same as an APPCCMD macroinstruction with the CONTROL=OPRCNTL and

QUALIFY=CNOS operands specified. The operator can then also use the MODIFY DEFINE command, which functions the same as an APPCCMD macroinstruction with the CONTROL=OPRCNTL and QUALIFY=DEFINE operands specified. The application is notified when the operator-issued commands have completed.

Coding session limits

These session limit values can be set for a VTAM application program by coding specific operands on the APPL definition statement. The VTAM application program can also establish these session limit values and override the specifications on the APPL definition statement by issuing an APPCCMD macroinstruction with the QUALIFY=DEFINE operand specified. Following are the unique session limit values that are used:

- The overall maximum number of sessions between two partner LUs for a given mode name, which you specify on the DSESLIM operand of the APPL definition statement
- The minimum number of sessions for a given mode name for which the application program is the contention winner, which you specify on the DMINWNL operand of the APPL definition statement
- The minimum number of sessions for a given mode name for which the application program might be the contention loser (the partner LU is the contention winner), which you specify on the DMINWNR operand of the APPL definition statement

The sum of the DMINWNL and DMINWNR values must always be less than or equal to the maximum number of sessions specified on the DSESLIM operand. The AUTOSES operand on the APPL definition statement can be used to have VTAM automatically activate a specific number of contention-winner sessions for a given mode name for the application program. VTAM establishes these sessions when the first CNOS request is issued by the application program for a partner LU using a given mode name.

Session limits and deactivation

Using CNOS, an LU 6.2 session partner can request that the session limits be set to 0. Then, the sessions that are active and not used for a conversation are deactivated. The LU partner that is responsible for deactivating the sessions for a given mode name is determined by an indicator in the CNOS request. The DRESPL operand on the APPL definition statement can be used to indicate if the VTAM application program accepts the responsibility for deactivating the sessions (ALLOW) or if VTAM assigns that responsibility to the partner LU when the CNOS request is sent (NALLOW).

When sessions between partner LUs are to be deactivated, the existing conversations are not disrupted, but subsequent allocation requests for conversations can be queued. Prior to deactivating the sessions for a given mode name, the partner LUs can indicate whether conversations that are queued for a session are to be processed or discarded. The DDRAINL operand on the APPL definition statement indicates whether the VTAM application drains the queued allocation requests when deactivating sessions. If you code DDRAINL=ALLOW, the conversations that are queued for a session for the given mode name are processed before deactivating the sessions. If you allow conversations to be drained, new conversations are allocated a session if the allocation request occurs prior to the deactivation of all of the sessions for the given mode name. If you code DDRAINL=NALLOW on the APPL definition statement, VTAM does not allow the application program to drain queued session requests and the conversation requests are rejected.

Managing LU 6.2 sessions with operator commands

Session characteristics, such as session limits, LU capability, and LU responsibilities, must be managed for every conversation involving an LU 6.2 application program. These characteristics can be controlled by the application program and through operator commands.

Operator commands manage conversation resources by establishing or modifying the following:

- New session limits between an application and partner LU (MODIFY CNOS).
- Session limit values VTAM uses to negotiate CNOS requests initiated by a partner LU for a specific application (MODIFY DEFINE). These values are stored in the logon mode table.

You can establish session limits using the MODIFY CNOS operator command. After session limits are established, actual sessions can be started.

Session limits must also be established for a mode name before conversations that use that mode can be allocated. When a request for a session is made, the requested session limits can be accepted by the partner LU or the session limits can be changed to new values. The process for accepting or changing proposed session limits is called negotiation. Logon mode table values are used in session limits negotiation. Negotiation is handled for the application program by the MODIFY DEFINE operator command. For more information on session negotiation, see the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LU 6.2 security

VTAM LU 6.2 support provides the following security functions for your network:

- Encryption facility
- LU 6.2 conversation-level verification (user ID verification)
- LU 6.2 session-level LU-LU verification

The encryption facility protects data passing over lines between network resources by permitting enciphering and deciphering of data for LU-LU sessions. For an application program or peripheral node logical unit to have cryptographic sessions, the host processor must support cryptography. For more information on the encryption facility refer to "Cryptography facility" on page 386.

LU 6.2 user ID verification is a conversation-level security protocol, taking place at the time a conversation is started. For more information on LU 6.2 conversation-level security, refer to the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LU 6.2 session-level LU-LU verification is a session-level security protocol that is used to verify the identity of each logical unit at the time the session is activated. LU 6.2 session-level LU-LU verification provides the ability to verify the identity of an application program partner LUs during the activation of sessions between type 6.2 LUs. VTAM-generated random data is encrypted using one of the data encryption standard algorithms. The encryption key is a password associated with each LU-LU pair. The encrypted data is carried on both the session activation request and the response so that each LU partner can verify the other partner for the session. The passwords used for session-level LU-LU verification are not coded on any VTAM definition statements but are implemented through an external security management product, such as RACF. If you plan to use LU 6.2 session-level LU-LU verification, a security management product (such as RACF

1.9 or later) must be installed and active. In addition, a profile for the LU needs to be in the security management database. With RACF 1.9, the APPCLU class needs to be active and a profile of the LU needs to be in the APPCLU class. For an example of the appropriate RACF coding, refer to the *z/OS Security Server RACF Security Administrator's Guide*.

During activation of LU 6.2 sessions involving control points, the VERIFYCP start option specifies whether VTAM performs session-level LU-LU verification.

The VERIFY and SECLVL operands on the APPL definition statement identify the level of partner-LU security verification.

If the application program is the PLU and an LU-LU password is defined for the partner LU, VTAM requests that LU-LU verification be performed during session activation. If a password is not defined, LU-LU verification is not requested.

If the application program is the secondary logical unit, one of the following occurs:

- If the session activation request specifies LU-LU verification and the LU-LU password is defined, verification is performed.
- If the session activation request does not specify LU-LU verification and no LU-LU password is defined, session activation continues without verification.
- If neither of the previous cases applies, VTAM rejects the session activation request.

If you are using LU 6.2 session-level LU-LU verification, you must create the RACF profile using either three-part or four-part names. Create the profile using a four part name if one of the following is true.

- The profile you are creating is for the VTAM control point. That is, you are trying to verify the identity of the partner CP for CPCP sessions.
- The local LU supports network-qualified names. The local LU supports network-qualified names if it specified NQNAMES=YES on the OPEN ACB.

Create a RACF profile using a three part name if none of the above situations are true. Refer to the following examples of a three part name definition followed by a four part name.

- RDEFINE APPCLU localnetid.localLU.remoteLU UACC(NONE) ...
- RDEFINE APPCLU localnetid.localLU.remotenetid.remoteLU UACC(NONE) ...

Note: Specifying VERIFY=OPTIONAL does not restrict the ability of a logical unit without a corresponding LU-LU profile to establish sessions with this application program.

If using RACF as your external security management product, the MODIFY PROFILES command enables you to reload an active application program set of existing defined RACF profiles. However, you cannot change the RACF profile with MODIFY PROFILES, only refresh it. The RACF profile contains the LU-LU password and only someone with RACF security clearance can change it. This can be helpful when the password for an LU-LU pair has been changed or when session activation errors are occurring. The profile changes affect only those sessions that are started after using the command; active sessions are not affected.

LU 6.2 sync point services

LU 6.2 sync point services is a session-level protocol. The maximum service level parameters are exchanged at session activation. However, when the conversation is allocated, the service level can be lower than the parameters exchanged at session activation. This can be caused by the inability of the other logical unit to support a higher service level. Synchronization of protected resources is the application program's responsibility. It is a means of recovery from errors or failures that allows the data network to divide its transmissions into logical units of work and commit or back out the information in these divisions of work.

The sync point function is divided between VTAM and a VTAM application program. LU 6.2 provides the facilities to allow the application program to implement a sync point facility by supporting the flow of information between application programs and the ability to control a session by rejecting, holding, or releasing a session based on a session identifier.

Persistent LU-LU session support does not allow LU 6.2 sync point sessions to be retained. Instead, they end when an application fails and they cannot be recovered.

The SYNCLVL operand on the APPL definition statement indicates the level of synchronization for this application program. If you do not code SYNCLVL, confirmation requests and sync point processing are not supported.

On the APPL definition statement, you can also specify (in conjunction with sync point support) when the ATTN exit for an LU 6.2 application program is to be scheduled. This exit routine can be invoked when the last session (ATNLOSS=LAST) is deactivated between an LU and MODE (LU-mode) pair. The ATTN exit can also be invoked for all session deactivations by specifying a value of ALL on the ATNLOSS operand. The LU 6.2 application program can use this operand without implementing sync point support.

For more details on sync point support, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Selective termination of idle LU 6.2 sessions

You can limit the use of some network connections, such as lines. Defining a limited network connection enables you to limit the use of the connection and keep only active sessions on that connection. Sessions that are not active are terminated. If all sessions are terminated, then the connection is terminated. The best network connections to define as limited are lines and PUs whose cost is determined by the length of time a connection exists. Defining these as limited can help reduce switched line connect charges.

When a session traverses a limited connection, the session can be terminated if no conversation is detected for a set period of time. You must specify the timeout on the LIMQSINT operand of the APPL definition statement. LIMQSINT specifies how long (in seconds) unused limited resource sessions will remain on the queue before they are terminated.

Note: Only LU 6.2 conversations are affected by limited resource definition. Non-LU 6.2 sessions are unaffected and cannot be limited.

To use LU 6.2 limited resource management:

1. Determine which network connections you want to define as limited.

2. Define the network connections to VTAM by coding LIMRES=YES on the GROUP, LINE, or PU statement. LIMRES is valid for PU types 1, 2, and 2.1 only.
3. Determine the time interval for VTAM to search the queue and define the interval to VTAM using the LIMQSINT operand on the APPL definition statement.

You can specify a line, a group of lines, or a PU as a limited resource for the following major nodes:
- External communication adapter
- Local SNA (PU only)
- MODEL (PU only)
- NCP
- Switched (PU only)

After you define the resource to be limited, set the time interval for the queue search. As a starting point, you should specify a time interval at least 1 second less than half of the shortest line time cost interval. For example, if the line time cost interval is 120 seconds, you should specify 59 seconds on the LIMQSINT operand. You might need to adjust the time interval for different applications.

If you are using VTAM CMIP services, be aware that the selective termination of idle LU 6.2 sessions can affect CMIP services associations. An association is a cooperative relationship between application entities for data exchange. In this case, an application entity can be another CMIP services or CMIP application program.

VTAM CMIP services ends idle associations when either of the following occurs:
- Selective termination of idle LU 6.2 sessions is in effect and the timeout value has expired.
- Selective termination is *not* in effect and VTAM CMIP services automatically ends the association because it is idle.

In either case, all idle associations are ended. Associations that were specifically requested by an application program (associations for which an application program issued an ACF.Associate request) are not affected by selective termination at the system where the ACF.Associate request was initiated, but can be terminated automatically on the system that was the target of the association.

Selective termination of idle network management sessions

Network management is the process of planning, organizing, monitoring, and controlling a communication-oriented data processing or information system. The architecture provided to assist in network management of SNA systems is called *management services* and is implemented as a set of functions and services designed to capture and use the information needed for effective management.

VTAM can exchange management services information with other systems that provide management services transport, or multiple-domain support, such as AS/400, Communication Manager/2, or other VTAM systems. This type of information flows between architected transaction programs in the respective systems over SNA sessions. The network management sessions that carry this exchange use either the CPSVCMG or SNASVCMG mode names. For further information on management services concepts and architecture, refer to *Systems Network Architecture Management Services Reference*.

The limited resource function is also available on some control point sessions used for network management flows, though for these sessions, it differs from the way it operates for noncontrol point sessions. For network management flows using the CP SNASVCMG session, a session is deactivated only when there are not any management services (MS) transport transactions outstanding, in addition to the session being inactive.

By coding the LIMINTCP start option, you specify the interval to retain a free CP SNASVCMG session for a resource that is defined as limited (with the LIMRES operand). The CP SNASVCMG session is used for some network management flows.

Note: The limited resource function for the control point is supported only on CP sessions for network management flows using the CP SNASVCMG session. The function is not available on APPN CP sessions or dependent LU server sessions.

To define limited resource management for network management flows, perform the following steps:

1. Determine which network connections you want to define as limited. These are the lines whose costs are determined by the length of connect time.
2. Define the network connections to VTAM by coding LIMRES=YES on the GROUP, LINE, or PU definition statement. LIMRES is valid only for PU type 2.1.

   You can specify a line, a group of lines, or a PU as a limited resource for the following major nodes:
   - External communication adapter
   - Local SNA (PU only)
   - MODEL (PU only)
   - NCP
   - Switched (PU only)
3. Determine the interval for VTAM to search the queue.

   As a starting point, the interval should be at least 1 second less than half the shortest line time cost interval. For example, if the shortest time cost interval is 1 minute, the interval should be 29 seconds.

   Specify the interval to VTAM with the LIMINTCP start option. If you do not code LIMINTCP, the limited resource function will not be supported for network management flows. Refer to *z/OS Communications Server: SNA Resource Definition Reference* for information on how to specify the LIMINTCP start option.

APING support

APING support provides the ability to test for connectivity between LU 6.2 resources, and to display pertinent routing and performance information. APING support consists of two transaction programs, APING and APINGD. The APING transaction program represents the client side of a transaction, while the APINGD transaction program represents the server side of a transaction. Once a conversation is established, the client transaction program sends packets of data to the server transaction program. When the client is finished, it indicates to the server transaction program either to send the data back to the client or to send a confirmation back to the client indicating that the data has been received. The client continues the transaction until complete.

The VTAM APING transaction program is started when a DISPLAY APING command is issued. The APINGD transaction program is started when another system in the network attempts to verify its connectivity and performance with VTAM using the APING transaction program.

The DISPLAY APING command enables the network operator to do the following:

- Verify connectivity with any LU 6.2 resource in the network
- Verify that another VTAM 4.3 or later node is operational
- Check the performance of the network using a particular logon mode
- Display routing information to the destination node, if a new session is established for the APING transaction

When you issue a DISPLAY APING command, the following operands determine the amount of data that is sent by the client transaction program to the server transaction program:

SIZE Specifies the size (in bytes) of the packets to be sent.

CONSEC
Specifies the number of consecutive packets to be sent.

Note: Depending on the maximum allowed RU size on the session path, the packets might be transmitted in a number of flows different than what you specified for CONSEC.

ITER Specifies the number of times that consecutive packets are sent to the server transaction program and returned.

ECHO Specifies whether data is returned to the client transaction program.

For example, if you issue the command `D NET,APING,ID=dluname,CONSEC=3,ECHO=YES,ITER=2,SIZE=500`, as shown in Figure 108 on page 423, three consecutive packets of 500 bytes are sent to the server transaction program two times, and the server transaction program returns three packets of 500 bytes two times.

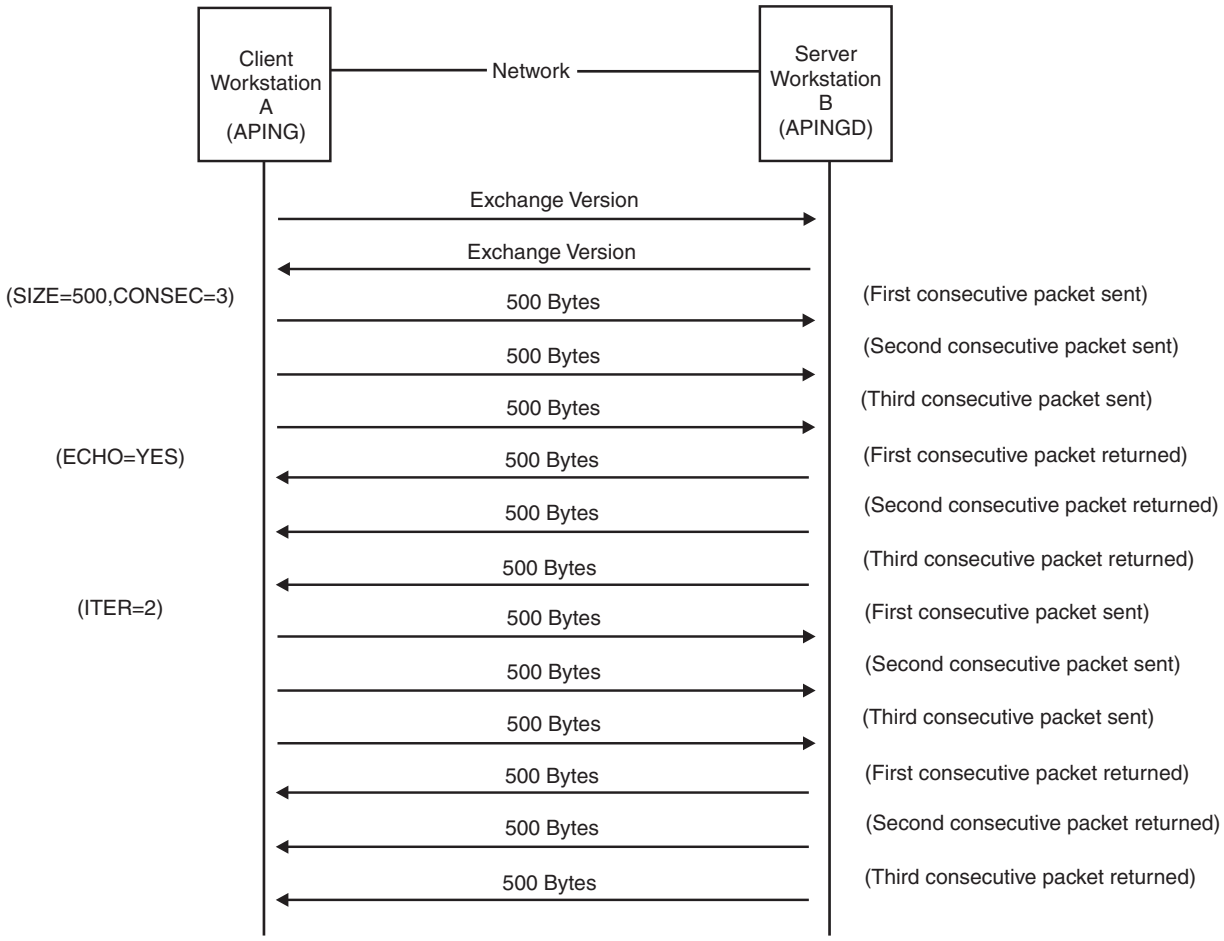


Figure 108. Example of flows between client and server for DISPLAY APING command

Guideline: The maximum value for CONSEC, ITER, and SIZE is 32763. By coding 32763 for each of these operands with ECHO=YES, 32763 packets of 32763 bytes of data are sent to the partner transaction program 32763 times and the partner transaction program returns 32763 packets of 32763 bytes of data 32763 times. This could cause a significant impact to the performance of your network.

The number of concurrent APING or APINGD transaction program instances allowed can be displayed with the DISPLAY APINGTP and DISPLAY APINGDTP operator commands, and can be adjusted using the MODIFY APINGTP and MODIFY APINGDTP commands. The default instance limit for the APING and APINGD transaction programs is 10. If you need to cancel APING transaction processing, see "Canceling APING sessions." The command verification exit, ISTCMMND, can be used to restrict the operands specified on the DISPLAY APING command.

Canceling APING sessions

If you want to terminate the sessions used by APING transaction processing, you can issue a VARY TERM command with the session identifier (SID) of the session being used for the data.

Canceling APING processing: If a session is allocated for the DISPLAY APING command, the SID of the session is reported in message IST1490I of the routing information message group, or as output from a DISPLAY APINGTP,LIST=ALL command.

If a session is not allocated for the DISPLAY APING command, the routing information message group is not issued. The performance information message group is not issued until the transaction processing is complete. You can determine which session to cancel in the following way:

1. Issue a `D NET,APINGTP,LIST=ALL` command.
2. Find the session with the LUNAME specified on the DISPLAY APING command.
3. Note the session identifier.
4. Issue a `V NET,TERM,SID=session_id` command to terminate the session.

Canceling APINGD processing: If this VTAM is the server for an APING transaction and you want to terminate processing, you can determine which session to cancel in the following way:

1. Issue a `D NET,APINGDTP,LIST=ALL` command.
2. Note the session identifiers.
3. Issue a `V NET,TERM,SID=session_id` command for each session you want to terminate.

High-performance data transfer (HPDT)

High-performance data transfer (HPDT) optimizes the performance of large data transfers for VTAM LU 6.2 applications. The performance benefits provided by HPDT services are achieved by more efficient use of critical CPU resources (CPU cycles, cache, memory bus, and the channel). Specifically, system resources are conserved by the following methods:

Reduction of VTAM internal data copies:
The movement of large pieces of data can result in significant processing costs on the send and receive execution paths. This cost is due to the effective path length of the MVCL instruction. Additionally, data movement flushes the CPU cache and consumes the memory bus bandwidth.

Reduced path length:
The processing of data transfers is streamlined through the use of the communications storage manager (CSM), which reduces the increase in path length that normally occurs as API crossing and PIU size increase. CSM is described in Appendix C, "Communications storage manager," on page 741.

Use of HPDT MPC:
HPDT services take advantage of HPDT MPC, which is described in "Multipath channel connections" on page 51.

HPDT includes a service that reduces the number of times data is moved for an APPCCMD send or receive request. No application change is required to receive performance benefits. However, an HPDT interface is also provided to enable applications to obtain performance benefits even greater than those provided by HPDT services. For more information about the HPDT interface, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Applications using the HPDT interface can specify CSM data space or CSM extended common service area (ECSA) for inbound data. When CSM data space is specified, the storage, when fixed, tends to be backed by real storage on or above the 2-gigabyte bar. VTAM receives data over the channel in the storage type specified by the STORAGE operand on the TRLE definition statement. If a mismatch exists between the VTAM TRLE definition and the application specification, an extra data move occurs. You should consider the CSM storage type used by your LU 6.2 applications when setting the STORAGE parameter for the TRLE definition. For more information about the TRLE definition, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

Communications storage manager (CSM)

The communications storage manager (CSM) function allows authorized host applications to share data with other CSM users without having to physically move the data. A CSM user can be any system-authorized application program or product. Users of CSM obtain and return storage in the form of CSM buffer pools by using the IVTCSM macroinstruction.

VTAM uses CSM to perform channel I/O over a HPDT MPC connection and to provide the high-performance data transfer (HPDT) function for host LU 6.2 applications. (HPDT MPC connections are explained in "Multipath channel connections" on page 51.) Host LU 6.2 applications use the IVTCSM and APPCCMD macroinstructions to reduce the use of system resources for large data transfers. For more information about the HPDT function, see "High-performance data transfer (HPDT)" on page 424.

The application programming interface for CSM is described in *z/OS Communications Server: CSM Guide*.

For information about how to install, configure, monitor, and diagnose CSM storage, see Appendix C, "Communications storage manager," on page 741.

Chapter 15. CMIP application programs

VTAM Common Management Information Protocol (CMIP) services is an implementation of the OSI standards for network and system management known as CMIP. VTAM CMIP services provides an API that enables application programmers to write application programs for network management.

CMIP application programs are defined to VTAM in an application program major node. For information about creating CMIP application programs, refer to the *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

VTAM topology agent CMIP application program

The manager application program for topology can be the NetView program (Version 3 Release 1) or a manager application program you write yourself. The VTAM topology agent is an agent application program for topology. It is a CMIP application program that resides on the VTAM host and runs internally to VTAM.

The manager application program for topology sends requests for the collection of different types of topology data to the VTAM topology agent. The VTAM topology agent collects topology information to send to a manager application program. Manager and agent application programs can reside on the same VTAM host or on different hosts.

CMIP services is used by both the topology manager application program and the topology agent application program (as well as other CMIP services application programs) to provide communication between the two application programs. Communication between the manager and agent application programs on different hosts is done by CMIP services-to-CMIP services communication using VTAM management services (MS) transport over LU 6.2 sessions.

VTAM CMIP services can communicate with CMIP services on platforms other than VTAM host platforms. For example, VTAM CMIP services can communicate with Communication Manager/2 (CM/2) CMIP services to allow host-based manager application programs to communicate with CM/2 agent application programs at workstations.

Implementing CMIP services

In VTAM, CMIP services is made available by specifying the OSIMGMT=YES start option at each host that is to use CMIP services. This start option gives you access to the following:

- VTAM topology agent
- CMIP services

The following steps describe, in general, how to enable CMIP services and the VTAM topology agent to communicate with a topology manager application program:

1. Use the estimating storage worksheets in *z/OS Communications Server: New Function Summary* to determine the total amount of CSA/ECSA storage you need.

© Copyright IBM Corp. 1984, 2011

2. Create a directory definition file that indicates which remote CMIP services have access to the local CMIP services. You do this by updating the IBM-supplied directory definition file. The IBM-supplied file allows unrestricted access to CMIP services. If you want to control access, you must update the IBM-supplied file.
3. Include the job control statements that load the directory definition file and the other CMIP services data sets in your VTAM start procedure.
4. Start VTAM with the start option OSIMGMT=YES or issue the MODIFY VTAMOPTS command with the OSIMGMT=YES start option to enable CMIP services and to start the VTAM topology agent.
5. Determine whether the default values for the OSITOPO, OSIEVENT, and UPDDELAY start options are acceptable, and code these start options if desired. For further information on these start options and keywords, refer to the *z/OS Communications Server: SNA Resource Definition Reference*. Selective reporting is also possible for logical lines and switched PUs and their major node through the VTAMTOPO keyword on the resource definitions.
6. Code an APPL definition statement for the application program, if you are writing your own application program.

For more information about the directory definition file, see "Associations and using the directory definition file for CMIP services" on page 429.

To learn how to write your own manager or agent application program, refer to the *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

The manager application program is installed, started, and maintained separately from VTAM. Refer to *Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information on the manager function.

What the topology agent does

At the request of the topology manager, the VTAM topology agent gathers information about the network and sends it to the topology manager. The topology agent can send the following information:

- Information about the local topology

  This includes information about:
  - The local agent host
  - Resources owned by the agent host
  - Resources defined to NCPs owned by the agent host
  - TGs, lines, and PUs supporting those connections
  - Names of contacted APPN and subarea nodes, and partner link-station names when available
- Information about the network topology

  This includes information about the APPN network nodes and subarea CDRMs known to this VTAM and the TGs and VRs that connect them.
- Information about LUs that includes:
  - Cross-domain resources
  - USERVARs
  - Generic resources
  - Application programs
  - Local non-SNA terminals

  - Owned LUs (those activated or defined to the VTAM, but not local to this VTAM)
- Information about a particular resource, by name
- Event notifications

The VTAM topology agent sends this information by way of CMIP snapshot actions, get requests, and event notifications. Snapshot actions send an initial view of certain groups of resources, depending on the type of snapshot. Ongoing snapshots keep reporting the status changes for the resources in respective groups, until the snapshots are stopped. Get requests gather information about a particular resource at a single point in time.

An event notification is a report that indicates when a resource changes, in contrast to snapshot actions, which give information on resource connectivity as well.

How data flows between the topology manager and the topology agent

The topology manager and the topology agent communicate data and data requests through CMIP services. For flows between CMIP services application programs on different hosts, the requests and responses flow over APPC sessions. The topology agent supplies information about local topology, network topology, and LU data in response to requests from the topology manager.

The topology agent forwards topology and status information upon request to the topology manager. The topology manager can then store this information and graphically display it.

The topology manager works with one or more topology agents to gather topology from the network. Topology agents can be on subarea nodes, APPN nodes, and combined subarea and APPN nodes in the network. Subarea nodes and APPN network nodes provide network and local topology; APPN end nodes provide local topology.

To start monitoring topology, the topology manager sends a request to the agent. The topology agent obtains the requested network or local topology data and sends the data to the topology manager.

Associations and using the directory definition file for CMIP services

To implement CMIP services and the VTAM topology agent, you must have a directory definition file. The directory definition file controls:

- The associations that CMIP services on this host has with other CMIP services and with itself
- Whether security is to be used on each association
- Name mapping
- Address mapping

The directory definition file is read when CMIP services is started. Before an association is started, CMIP services checks the file to determine whether the association is allowed. The directory definition file is checked only at the time the association is established. It is not checked at every exchange between partners in an association.

If the directory definition file indicates that the association is allowed, it continues to be allowed until the association ends in one of the following ways:
- Timing out
- Being ended by the application program
- CMIP services ending
- VTAM ending

If you allow an association and then decide not to allow that association, you cannot stop the association until you restart CMIP services [4]. Even if you change the directory definition file to indicate that the association is no longer allowed, issuing the MODIFY TABLE command does not stop the association.

Controlling associations

An *association* is a connection between CMIP services on this host and CMIP services on another node or between CMIP services on this host and itself. An association between CMIP services on this host and itself is called a local association. An association between CMIP services on this host and CMIP services on another node is called a remote association.

If the directory definition file allows an association, the association is created by CMIP services when a CMIP application program sends a CMIP request.

If CMIP services on NETA.SSCP1 has an association with CMIP services on NETB.SSCP2, CMIP services on each system can exchange CMIP requests and actions with each other. If no association exists between the two, the two CMIP services cannot exchange CMIP requests and actions.

The IBM-supplied directory definition file shipped with VTAM allows unlimited associations with other CMIP services. If you want to control which CMIP services on other systems can have associations with CMIP services on this host, you must edit the data set that contains the IBM-supplied file. The IBM-supplied directory definition file is in the ACYDDF member of the partitioned data set, SYS1.SISTCMIP. ISTCMIP is the DD name used in the VTAM start procedure, as described in *VTAM Installation and Migration Guide*.

If you use the IBM-supplied directory definition file without changing it, CMIP services on any node in any network can send data to and receive data from CMIP services on this host.

The IBM-supplied directory definition file is adequate for a controlled test environment. When you start communicating with other networks, you might want to update the directory definition file to control the associations that CMIP services on this host has with CMIP services on other nodes.

Determining security for associations

For each association, you can choose one of the following security options:
- No security on associations
- Data encryption standard (DES)-based security on associations
- Application program-to-application program security on associations
- No associations

4. You can also stop an association by writing a CMIP application to do so. For further information, refer to the *z/OS Communications Server: CMIP Services and Topology Agent Guide* .

The value of the associationKey attribute, which is specified on each entry in the directory definition file, determines whether an association is allowed, and if it is allowed, the type of security required. For a description of the associationKey attribute, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

No security on associations

If you do not want security on associations, specify the class, name, and associationKey attributes. Refer to *z/OS Communications Server: SNA Resource Definition Reference* for a description of these attributes.

In the following example, the directory entry indicates that CMIP services on this host can have associations with CMIP services on any node in any network, without any security.

```
class aetitle
name '*'
associationKey '*'
```

The asterisk (*) specified as the value of the name attribute means that the directory entry applies to any name not specifically listed in another directory entry in the file.

DES-based security on associations

To specify DES-based security, define the same encryption key for each CMIP services in an association. The encryption key is specified on the associationKey attribute of the directory definition file. The key is 16 hexadecimal characters in length, and must be defined on the entries that represent each of the instances of CMIP services that form the association.

Be careful in the distribution of encryption keys, because they allow access to VTAM similar to passwords. At the least, access to the directory definition file should be protected by a system security facility such as RACF.

You can also synchronize the time-of-day clocks on the two systems involved in the association. In the directory entry for the CMIP services on the other system, you specify the timeSync attribute to indicate the maximum difference allowed between the two time-of-day clocks.

CMIP services from products other than VTAM might not implement DES-based security. If you want to allow associations to CMIP services on another product by using DES-based security, verify that the CMIP services from the other product supports DES-based security before you define the directory definition file. If CMIP services from the other product does not support DES-based security and you have specified DES security in the directory definition file, no association to the CMIP services on that product is created.

In the following example, some instances of CMIP services on NETB can exchange requests and actions with the CMIP services on this host.

```
class aetitle
name '1.3.18.0.2.4.6=NETB'
associationKey 'a0b1c2d3e4f50011'
timeSync '20'
```

In the directory definition file for each instance of CMIP services on NETB, the same encryption key must be specified for the association with the CMIP services on this host. The timeSync attribute is required only if you do not choose to use the default value of 300 seconds.

Application program-to-application program security on associations

If you want to use security, but do not want DES-based security, the CMIP application programs can define application program-to-application program security. The VTAM topology agent and the NetView program do not implement application program-to-application program security. For information about defining this type of security, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

In the following example the associationKey value indicates that the CMIP application programs control access to themselves, rather than CMIP services controlling access to the application programs.

```
class aetitle
name '*'
associationKey '.'
```

No associations

In the following example the associationKey value indicates that no instance of CMIP services can establish an association with CMIP services on this host.

```
class aetitle
name '*'
associationKey '-'
```

Name mapping and address mapping

Name mapping allows an AE-title to be specified to override the default AE-title. Address mapping allows a network ID and node to be specified, rather than allowing CMIP services to extract them from the AE-title.

For more information, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

Updating the directory definition file

To update the directory definition file, you can edit the data set that is used to construct the directory definition file. You can edit the file while VTAM is running, but CMIP services reads the updated file by using one of the following methods:

- If CMIP services is active, edit the directory definition file and then load it by issuing the MODIFY TABLE command:

  ```
  MODIFY proc,TABLE,OPT=LOAD,TYPE=CMIPDDF
  ```

  If the new directory definition file is not syntactically correct, the old table continues to be used.

  Note that associations that were allowed with the previous directory definition file can remain in effect until CMIP services is restarted. Even if you update the directory definition file to indicate that a particular association is no longer allowed, that association can remain in effect until CMIP services is restarted. The MODIFY TABLE command does not cause previous associations to come down, even if they are not allowed in the updated directory definition file.
- If CMIP services is active, stop CMIP services by issuing the MODIFY VTAMOPTS command with the OSIMGMT=NO start option and then restart CMIP services by issuing the MODIFY VTAMOPTS command with the OSIMGMT=YES start option
- Restart VTAM with the OSIMGMT=YES start option.

When CMIP services is started (either when VTAM is initialized with the OSIMGMT=YES start option or when the MODIFY VTAMOPTS command is

issued with the OSIMGMT=YES start option), CMIP services reads the directory definition file from member ACYDDF. The SYS1.SISTCMIP data set that contains member ACYDDF is read and processed when CMIP services is started.

The directory definition file cannot be displayed with the DISPLAY TABLE command. You cannot see what the directory definition file contains. The directory definition file is not like other user-defined tables. You can specify the member name of other user-defined tables on the MODIFY TABLE command, but for the directory definition file, the member name is always ACYDDF. To change the directory definition file, edit ACYDDF and replace its current contents.

Chapter 16. Functions provided by VTAM in a sysplex

A sysplex is a set of MVS systems communicating and cooperating with each other, through certain multisystem hardware components and software services, to process customer workloads. The inclusion of a coupling facility within the sysplex allows for high performance data sharing. For more information about sysplex, refer to *z/OS MVS Setting Up a Sysplex*.

VTAM supports the attachment of a sysplex to a network. Sessions into the sysplex can be established from either subarea nodes or APPN nodes. There are several VTAM functions that are available only in a sysplex environment. If a coupling facility and an APPN or mixed APPN and subarea environment exists in the sysplex, the user can take advantage of the following VTAM functions:

Generic resources
: Allows an user to easily connect to any one of a number of duplicate application programs on different systems. The user uses a generic name and VTAM determines the actual application program for the session based on workload and other performance criteria.

Multinode persistent sessions
: Allows for the restoring of an application program session during a planned or forced takeover, or after hardware, operating system, or VTAM failures.

TSO generic resources
: Allows for multiple TSO/VTAM application programs to have a common name. During session establishment, VTAM determines the actual application program to be used for the session.

Dynamic definition of VTAM-to-VTAM connections
: Allows for dynamic establishment of XCF connections between VTAM nodes in a sysplex when VTAM is initialized with the XCFINIT start parameter, or at a later time when the operator determines connections can be established (by activating the XCF Local SNA major node, ISTLSXCF).

VTAM also provides support for TCP/IP functions that need access to a coupling facility. If a coupling facility exists in the sysplex, the user can take advantage of the following TCP/IP functions:

Sysplexports
: Enables multiple TCP/IP stacks in the sysplex to collaborate on the assignment of ephemeral port numbers for a distributed dynamic virtual IP Address (DRVIPA) that is shared by a number of applications. This function uses a structure in the coupling facility to track the ephemeral port assignments across the sysplex.

Sysplex-wide security associations
: Allows for IPSec security associations (SA) and their workloads to be distributed to target TCP/IP stacks within the sysplex. It also allows the SAs to be automatically restarted on another processor in the sysplex when a dynamic routable virtual IP Address (DRVIPA) takeover occurs. This takeover capability is provided for the following cases:

  - Takeover occurs because the VIPA owning stack or its host went down (unplanned takeover)

© Copyright IBM Corp. 1984, 2011

- Takeover or giveback occurs when the stack and its host stay up but a VIPA ownership change between hosts is desired (planned takeover)

Setting up the sysplex environment for VTAM and TCP/IP functions

The following sections apply to VTAM functions.

Sysplex subplexing

VTAM participates in the sysplex environment through two sysplex groups:

- The XCF group is used for dynamic definition of VTAM-to-VTAM connections.
- The CFS group is used for VTAM coupling facility services.

You partition VTAMs in the sysplex into subsets (subplexes) by modifying the names of the two sysplex groups that the sysplex joins. Use subplexing to separate and isolate the XCF connectivity of sets of VTAM nodes in the sysplex. VTAMs using the same XCF group name and the same CFS group name are in one subplex, and VTAMs with a different XCF group name and CFS group name are in a different subplex.

Modify the group names that VTAM uses with the XCFGRPID start option. VTAM joins the XCF group with a name in the format ISTXCF*vv* and it joins the CFS group with a name in the format ISTCFS*vv*, where *vv* is the numeric 2-digit value you supply on the XCFGRPID start option. The value specified for *vv* must be in the range 02-31. If you specify a single digit in the range 2-9, the value is padded on the left with 0. If you do not supply an XCFGRPID value, the XCF group name is ISTXCF and the CFS group name is ISTCFS01.

TCP/IP stacks can also be partitioned into subsets (subplexes) within the sysplex by specifying the XCFGRPID parameter on the GLOBALCONFIG statement in the TCP/IP profile. Because the TCP/IP stack relies on VTAM support for its XCF connectivity, TCP/IP subplexes cannot span VTAM subplexes. That is, two TCP/IP stacks in the same subplex cannot be in different VTAM subplexes. To enforce this, the sysplex group that TCP/IP joins in the sysplex has a name in the format EZBT*vvtt*, where *vv* is the VTAM group ID suffix you specified on the XCFGRPID start option, and *tt* is the TCP/IP group ID suffix value you specified on the XCFGRPID parameter on the GLOBALCONFIG statement. For example, if VTAM is started with a value of XCFGRPID=11, and a TCP/IP stack is started with a GLOBALCONFIG statement specifying XCFGRPID=21, the corresponding TCP/IP sysplex group name is EZBT1121.

If VTAM is stopped and restarted with a different value for the XCFGRPID parameter, the TCP/IP stacks that use that VTAM must also be stopped and restarted to pick up the new VTAM subplex suffix.

Because subplexing separates XCF connectivity, the VTAM and TCP/IP coupling facility structures that are accessed must be separated by subplexing.

- For the VTAM structures (for generic resources and MNPS), VTAM appends the VTAM XCFGRPID suffix to the end of the name specified on the STRGR and STRMNPS start options (or on ISTGENERIC and ISTMNPS, if the names are the default values). Thus, using the previous example, if you specify STRGR=ISTMYGR and STRMNPS=ISTMYMNPS as start options, and you also specify XCFGRPID=11, VTAM would attempt to connect to the structures ISTMYGR11 and ISTMYMNPS11. The fully suffixed MNPS and generic resources structure names can be displayed by issuing the D NET,ID=VTAM command.

- For the TCP/IP structures (for SWSA and sysplexports), VTAM appends the VTAM XCFGRPID suffix, followed by the TCP/IP XCFGRPID suffix to the end of the base structure names to create the separation. The base structure name for SWSA is EZBDVIPA and the base structure name for sysplexports is EZBEPORT. Again, using the previous example, TCP establishes connectivity to structures EZBDVIPA1121 and EZBEPORT1121.

Note: Any system in the sysplex that has access to the coupling facilities can be picked as a monitor for sysplex processing, and this has operational implications affecting how certain messages are processed. It is possible that the XCF monitor processing could be issuing messages on a system in 'subplex a' for a system that resides in 'subplex b'.

References to the base group names, ISTXCF, ISTCFS01, EZBTCPCS, and the base structure names, ISTGENERIC, ISTMNPS, EZBDVIPA, and EZBEPORT reference the fully suffixed names, if subplexing is used.

Setting up a subplex

Setting up TCP/IP and VTAM subplexes within a sysplex requires some preparation. Following is a procedure for preparing your sysplex for subplexing:

1. If there are TCP/IP stacks that use the subplexed VTAM nodes, see setting up a sysplex in *z/OS Communications Server: IP Configuration Guide*.
2. If the TCP/IP stack subplex configuration (determined in the previous step) requires the use of HiperSockets VLAN IDs (using the IQDVLANID parameter on the GLOBALCONFIG TCP/IP profile statement), ensure that all VTAM nodes in the sysplex are at z/OS V1R8 or later.
3. If VTAM coupling facility structures for generic resources or MNPS are to be used, ensure that all the structures have been defined to z/OS and are specified in the active CFRM policy using their fully suffixed names. For example, if the generic resources function is to be used with the base structure name of ISTGENERIC, and two VTAM subplexes are to be configured, one subplex with a suffix of 11 and the other with a suffix of 12, then ensure that the ISTGENERIC11 and ISTGENERIC12 structures are defined to MVS and specified in the active CFRM policy.
4. For each VTAM that needs to specify an XCFGRPID value or supports TCP/IP stacks that need to be changed to specify an XCFGRPID value or IQDVLANID value on their GLOBALCONFIG statements, do the following:
   a. If this VTAM requires a new XCFGRPID value, stop VTAM
   b. For each TCP/IP stack supported by this VTAM that requires a new XCFGRPID or IQDVLANID value:
      1) Stop the TCP/IP stack.
      2) Update the GLOBALCONFIG statement in the TCP/IP profile for this stack with the new XCFGRPID and IQDVLANID values.
      3) Restart the TCP/IP stack.
   c. If VTAM was stopped, update the VTAM start options to add the new XCFGRPID option with the group id value for the new subplex (or specify the new XCFGRPID start option when prompted, after starting VTAM).
   d. Restart the VTAM node.

Both the XCF and CFS groups must be considered when planning sysplex coupled data sets. Specifically for the XCF group of VTAM, the MAXMSG parameter for PATHIN and PATHOUT should be adequately tuned to minimize the number of

no buffer conditions. Inadequate tuning of MAXMSG can lead to unexpected XCF outages (INOPs). Refer to *z/OS MVS Setting Up a Sysplex*, for additional information.

The following information applies to all VTAM and TCP/IP functions that require a coupling facility structure in an MVS sysplex. If you plan to use the generic resource function, the multinode persistent session function, the sysplexports function, or the Sysplex Wide Security Associations function, a coupling facility structure must exist for each function.

- An MVS coupling facility provides the hardware and software that supports high-speed shared storage across multiple MVS systems and provides multiple copies or images of the MVS operating system to process work concurrently.
- There must be an active coupling facility resource management (CFRM) policy for the sysplex environment.

  A CFRM policy defines how an installation will manage its coupling facility resources, which can be shared among multiple subsystems. For more information about how to create, activate, and make changes to the CFRM policy, refer to *z/OS MVS Setting Up a Sysplex*.

  The CFRM policy information describes the coupling facilities that are used in the sysplex and specifies the requirements for structure size and allocation within each coupling facility.
- Each coupling facility structure to be used must be defined in the active CFRM policy for the sysplex. If subplexing is used, the structure names specified in the CFRM policy must include the subplex suffixes. You might need to define (using the CFRM policy) more than one structure of each type as being in the sysplex. See "Sample CFRM coding" on page 441 for an example of CFRM coding.
- When defining the coupling facility structure in a CFRM policy, specify the amount of coupling facility storage which should be allocated on behalf of this structure. For information on determining the amount of storage necessary, see "Determining the size of the coupling facility structure" on page 441.
- All VTAM and TCP/IP functions using a coupling facility are designed to maintain data integrity when any single type of failure occurs, but do not reliably handle concurrent failures of different types. For instance, if a VTAM node and the coupling facility structure fail concurrently, data is lost. Thus, it is advisable to isolate the coupling facilities from the VTAM nodes (for example, they should not share the same power supply).

  In the same way, multiple coupling facilities in the same configuration should be isolated from each other. If more than one coupling facility is present in a configuration, a new version of the structure can be created if the coupling facility containing the active version of the structure fails. Also, if a link failure to the structure occurs, the structure might be moved to a new coupling facility where it could be accessed by all VTAM nodes in the configuration.

Coupling facility structure attributes

The attributes defined by VTAM describe the various coupling facility structure characteristics. For the structures used by VTAM and TCP/IP functions, VTAM supplies all attributes to MVS listed in Table 44 on page 439 and Table 45 on page 440 with the exception of the maximum list counts attribute. The list-set-entry count or list-set-element count attribute is used to determine the size specified in the policy.

When the structure size has been determined, the corresponding CFRM policy can be defined following the directions in *z/OS MVS Setting Up a Sysplex*. For the

sysplex to recognize a structure, a CFRM policy must be defined and active. This definition indicates that the coupling facility will contain the named structure.

Access to the structure occurs when VTAM connects to a coupling facility structure.

Table 44. Structure attributes used to compute structure size

Attribute	Generic resources	Multinode persistent sessions	Sysplex Wide Security Associations	Sysplexports
Adjunct	Yes	Yes	Yes	Yes
List entry reference	Names	Keys	Keys	Keys
List element characteristic	2	0	0	0
List count	4	256 [1]	2048	1024
Number of lock entries	3	0	0	1024
Maximum number of data elements	16	255	255	32
Initial entry-to-element ratio [2]	List entry portion – 50 Data object portion – 1	List entry portion – 1 Data object portion – 2	List entry portion – 1 Data object portion – 10	List entry portion – 1 Data object portion – 32
Maximum list set element count	N/A	See "Determining the size of the coupling facility structure" on page 441	N/A	N/A
Maximum list set entry count	See "Determining the size of the coupling facility structure" on page 441	N/A	See "Determining the size of the coupling facility structure" on page 441	See "Determining the size of the coupling facility structure" on page 441
Lock table entry characteristic	0	0	0	0

Notes:

1. The list count number can increase across a structure rebuild. Issue DISPLAY STATS,TYPE=CFS to get the current number of lists allocated.
2. During the rebuild process this ratio might be altered to better fit the actual use of list entries and elements. VTAM can also use the dynamic structure alter to change this ratio.

Table 45 on page 440 contains structure attributes used to determine the coupling facility preference and exclusion list for a structure.

Table 45. Structure attributes defined by VTAM

Attribute	Generic resources	Multinode persistent sessions	Sysplex Wide Security Associations	Sysplexports
Structure type	List	List	List	List
Structure rebuilds allowed	Yes	Yes	Yes	Yes
Structure connection disposition	KEEP	KEEP	DELETE	DELETE
Structure disposition	DELETE	KEEP	DELETE	DELETE

Structure type

The coupling facility supports three structure types: cache, list, and lock. The coupling facility structure type used by VTAM is list. List structures allow list format data sharing. The type attribute is defined when the first VTAM node connects to the coupling facility structure.

Structure rebuilds

Structure rebuild allows a connector to a structure to allocate another structure with the same name and reconstruct data in the newly allocated structure. Rebuilding allows connectors to change the location or attributes of the structure without having to disconnect.

VTAM sets the size of the rebuilt structure equal to the current size of the original structure, unless there is a structure storage shortage or the policy size information is being changed across the rebuild. VTAM considers a shortage to exist if more than 80 percent of total structure storage is in use. If there is a shortage, and the current size is less than the maximum size, the rebuild structure size will be set to the maximum size. If the CFRM policy size information is being changed across the rebuild, the rebuild structure size will be set as follows:

- If INITSIZE is not specified, the rebuild structure size is the value of SIZE in the CFRM policy definition.
- If INITSIZE is specified and:
  - The current structure size is greater than or equal to the INITSIZE value and less than or equal to the SIZE value, the current size is retained.
  - The current structure size is less than the INITSIZE value, the INITSIZE value is used.
  - The current structure size is greater than the SIZE value, the SIZE value is used.

See “Structure rebuild” on page 443 for more information on structure rebuilds.

Structure connection disposition

The structure connection disposition attribute indicates how the connection is handled in the event of abnormal disconnects from the structure.

DELETE Indicates that if a VTAM node fails or is forced to disconnect due to some type of failure, the node's connection definition is removed.

KEEP Indicates that if a VTAM node fails or is forced to disconnect due to some type of failure, the node's connection remains defined but

is in a failed-persistent state. Failed-persistent state indicates that if a VTAM node fails or is forced to disconnect due to some type of failure, the node's connection remains defined but is in a failed-persistent state. The failed-persistent state also indicates that recoverable data is located in the coupling facility structure.

Note: A network operator might see VTAM in a failed-persistent state because of a VTAM failure or because VTAM disconnects from the coupling facility. To ensure that data needed to recover is available, the operator should not issue the MVS SETXCF FORCE command.

Structure disposition

Structure disposition indicates what happens to the structure when there are no defined connections to a structure.

DELETE Indicates that the structure will be deallocated after the last connection to that structure is deleted.

KEEP Indicates that the structure is never deallocated. The only way to deallocate the structure is through the MVS SETXCF FORCE command.

Determining the size of the coupling facility structure

The Coupling Facility Structure Sizer Tool (CFSizer) is used to determine the values for the INITSIZE and SIZE parameters specified in the CFRM policy for a VTAM or TCP/IP structure. The CFSizer tool can be found at the following Web address: http://www.ibm.com/servers/eserver/zseries/cfsizer

Sample CFRM coding

The following example shows the CFRM policy definition for including the generic resources, MNPS, SWSA and sysplexports structures in a sysplex that is partitioned into two VTAM subplexes, 11 and 12. Within VTAM subplex 11, the sysplex is partitioned into two TCP/IP subplexes, 21 and 22. Within VTAM subplex 12, the sysplex is partitioned into two TCP/IP subplexes, 22 and 23. Note that TCP/IP subplex 22 within VTAM subplex 11 is separate from TCP/IP subplex 22 within VTAM subplex 12. TCP/IP subplexes cannot span VTAM subplexes.

```
//SYSIN   DD   *
      DATA TYPE(CFRM) REPORT(YES)

      DEFINE POLICY NAME(VTAMSAMP) REPLACE(YES)

         CF NAME(RALNSCF1)
                TYPE(009672)
                MFG(IBM)
                PLANT(02)
                SEQUENCE(000000041266)
                PARTITION(1)
                CPCID(00)
                DUMPSPACE(2000)

         CF NAME(RALNSCF2)
                TYPE(009672)
                MFG(IBM)
                PLANT(02)
                SEQUENCE(000000041266)
                PARTITION(2)
                CPCID(00)
                DUMPSPACE(2000)
```

```
STRUCTURE NAME(ISTGENERIC11)
       SIZE(10000)
       INITSIZE(7000)
       PREFLIST(RALNSCF1,RALNSCF2)

STRUCTURE NAME(ISTGENERIC12)
       SIZE(10000)
       INITSIZE(7000)
       PREFLIST(RALNSCF1,RALNSCF2)

STRUCTURE NAME(ISTMNPS11)
       SIZE(25600)
       INITSIZE(12288)
       REBUILDPERCENT(30)
       PREFLIST(RALNSCF1,RALNSCF2)

STRUCTURE NAME(ISTMNPS12)
       SIZE(25600)
       INITSIZE(12288)
       REBUILDPERCENT(30)
       PREFLIST(RALNSCF1,RALNSCF2)


STRUCTURE NAME(EZBDVIPA1121)
       SIZE(50000)
       INITSIZE(15000)
       REBUILDPERCENT(20)
       PREFLIST(RALNSCF2,RALNSCF1)

STRUCTURE NAME(EZBDVIPA1122)
       SIZE(50000)
       INITSIZE(15000)
       REBUILDPERCENT(20)
       PREFLIST(RALNSCF2,RALNSCF1)

STRUCTURE NAME(EZBDVIPA1222)
       SIZE(50000)
       INITSIZE(15000)
       REBUILDPERCENT(20)
       PREFLIST(RALNSCF2,RALNSCF1)

STRUCTURE NAME(EZBDVIPA1223)
       SIZE(50000)
       INITSIZE(15000)
       REBUILDPERCENT(20)
       PREFLIST(RALNSCF2,RALNSCF1)


STRUCTURE NAME(EZBEPORT1121)
       SIZE(10000)
       INITSIZE(5000)
       REBUILDPERCENT(20)
       PREFLIST(RALNSCF2,RALNSCF1)

STRUCTURE NAME(EZBEPORT1122)
       SIZE(10000)
       INITSIZE(5000)
       REBUILDPERCENT(20)
       PREFLIST(RALNSCF2,RALNSCF1)

STRUCTURE NAME(EZBEPORT1222)
       SIZE(10000)
       INITSIZE(5000)
       REBUILDPERCENT(20)
       PREFLIST(RALNSCF2,RALNSCF1)

STRUCTURE NAME(EZBEPORT1223)
```

```
SIZE(10000)
INITSIZE(5000)
REBUILDPERCENT(20)
PREFLIST(RALNSCF2,RALNSCF1)
```

Connecting to and allocating storage for coupling facility structures

When a coupling facility resource management (CFRM) policy with a VTAM structure is activated, VTAM is notified and connects to the structure. A structure policy can be changed or activated before or after VTAM is active. VTAM attempts to connect to a VTAM coupling facility structure when it detects that a structure has been defined. For the TCP/IP functions, VTAM does not attempt to connect to the structure associated with that function until a TCP/IP stack requests access to the structure. Before a VTAM node attempts to connect to the coupling facility structure, it first checks to see that it is running on a version of MVS that supports the coupling facility and that the coupling facility structure is defined to the active CFRM policy. The first VTAM node to connect to a structure causes the allocation of storage for the structure within a coupling facility and also defines the structure attributes associated with the structure.

Structure rebuild

The structure rebuild process is used to create a new copy of a coupling facility structure, either in the same coupling facility or in a different coupling facility, and to store in this new copy of the structure all of the data from the old copy of the structure.

Notes:

1. For generic resources only, data is copied one of two ways during the structure rebuild process for generic resource structure. Data is either copied by a VTAM node in the generic resource configuration from the old version of the structure into the new version of the structure, or all VTAM nodes in the generic resource configuration each supply the new version of the structure with local data.
2. For multinode persistent sessions only, use the MVS FORCE command specifying the ARM parameter to cancel VTAM when a rebuild is in progress. This allows VTAM to correctly disconnect from the structure.

Initiating a structure rebuild

Note: VTAM can specify a connection specific reason for starting a rebuild. This reason is displayed in MVS message IXC526I. Refer to *z/OS Communications Server: IP and SNA Codes* for a list of VTAM connection-specific reasons for starting a rebuild.

There are several reasons why a structure rebuild might be initiated:

- An MVS operator can initiate a structure rebuild
- VTAM will honor the REBUILDPERCENT value coded in the CFRM policy, if there is an active SFM policy. When a SFM is active, VTAM will allow MVS to initiate the rebuild based on the REBUILDPERCENT value. If no SFM policy is active, VTAM will initiate a structure rebuild when it loses access to the structure. For more information about the REBUILDPERCENT parameter and the SFM policy, refer to *z/OS MVS Setting Up a Sysplex*. MVS can initiate a rebuild if a VTAM node loses access to the structure and the REBUILDPERCENT parameter is coded in the active CFRM policy and an SFM

policy is active. For more information about the REBUILDPERCENT parameter and the SFM policy, refer to the *z/OS MVS Setting Up a Sysplex*.

- A VTAM node can initiate a structure rebuild when notified that the structure has failed.

Stopping a structure rebuild

Note: VTAM can specify a connection-specific reason for stopping a rebuild. This reason is displayed in MVS message IXC527I. Refer to the *z/OS Communications Server: IP and SNA Codes* for a list of VTAM connection-specific reasons for starting a rebuild.

When a structure rebuild has been initiated, it can be stopped for any one of several reasons.

- An MVS operator can stop a structure rebuild for structure.
- MVS automatically stops a rebuild if the new structure being created fails or becomes inaccessible.
- A VTAM node in a generic resource configuration can stop a structure rebuild if it loses access to the new version of the structure. A VTAM node never stops a structure rebuild once the old version of the structure has failed.
- A VTAM node in a generic resource configuration can stop a structure rebuild when it determines that all of the data in the old version of the structure might not be available for creating the new version of the structure.

Once a structure rebuild is stopped, the old version of the structure continues to be used (if possible) and the new version of the structure, if allocated, will be deallocated.

If MVS or an operator initiates the rebuild, VTAM will not stop the rebuild.

Coupling facility duplexing

All z/OS Communications Server coupling facility structures support system-managed duplexing. This allows two copies of the structure to be created for backup purposes. Updates to the structure are automatically kept in sync so if access to one of the structures is lost, the other copy is used without any disruptions. To create a duplexed copy of the structure, use the SETXCF START,REBUILD,DUPLEX command. To stop duplexing and return to a single allocated structure, use the SETXCF STOP,REBUILD,DUPLEX command. For more information on the SETXCF command, refer to *z/OS MVS System Commands*.

Coupling facility storage shortages

VTAM monitors the total amount of storage used by a coupling facility structure, the number of entries in use, and the number of elements in use. If VTAM detects storage 80 percent used (or greater), VTAM will issue message IST1439I, which indicates which type of storage shortage has occurred. The shortage situation is not ended until the constrained resource reaches 70 percent utilization.

To correct constrained resources:

- For total storage problems, change the SIZE parameter on the structure definition in the CFRM policy and issue a structure rebuild.
- For entries and element problems, VTAM performs a structure rebuild to adjust the entry-to-element ratio at 100 percent utilization.

Dynamic altering of structures

If you are using Level 1 Coupling Facility, VTAM provides nondisruptive reconfiguration of the coupling facility size and entry-to-element ratio attributes when VTAM detects a coupling facility storage problem.

Note: The MVS operator can use the MVS SETXCF ALTER command to initiate a structure alter process.

Dynamic changes to coupling facility size

For VTAM to dynamically increase the coupling facility storage size, specify the following on the structure policy statement:

SIZE Specify the maximum allowed size for the structure.

INITSIZE Specify the initial size of the structure. See "Determining the size of the coupling facility structure" on page 441 for information on calculating the structure size.

Note: For VTAM to dynamically alter the structure size, the SIZE value must be larger than the value of INITSIZE. The structure alter process is initiated to change the structure size when the current in-use percentage of total storage reaches 90 percent and the maximum size is greater than the current size. Once the size has reached the value of SIZE attribute, the policy must be changed and an alter or rebuild command issued to incorporate the changes. Set the ALLOWAUTOALT parameter on the CFRM policy for VTAM or TCP/IP structures to ALLOWAUTOALT(NO) (or allow the ALLOWAUTOALT parameter to default to NO) to prevent MVS attempts to resize the structure from conflicting with VTAM re-sizing events.

Dynamic changes to entry-to-element ratio

The structure alter process is initiated to change the entry-to-element ratio when the current in-use percentage of either entries or elements reaches 90 percent.

Dynamic definition of VTAM-to-VTAM connections

Dynamic definition of VTAM-to-VTAM connections depends on the functions of XCF in a sysplex. The MVS system programmer must establish the XCF environment. VTAM requires that a value for the MVS symbol &SYSCLONE be provided during MVS initialization. VTAM joins the XCF group named ISTXCF. No action is required to specify this group.

VTAM uses the XCF facilities only if they exist. If XCF facilities are not available or if subarea routing is used exclusively, VTAM-VTAM communication within a sysplex requires explicit definition.

Whether VTAM attempts to establish connectivity through XCF with other nodes in the sysplex during VTAM initialization is determined by the value specified for the start option XCFINIT.

- If XCFINIT=YES is specified (for an APPN node), VTAM joins the ISTXCF group at VTAM initialization. As it becomes aware of each other member node of the Sysplex, VTAM builds a TRLE and APPN PU definition for that node and begins the activation of the APPN PU. XCFINIT=YES is the default for APPN nodes. XCFINIT=YES is not allowed for pure subarea nodes.
- If XCFINIT=DEFINE is specified, VTAM joins the ISTXCF group at VTAM initialization. If the node is a pure subarea node, as it becomes aware of each other member node of the Sysplex, VTAM builds a TRLE definition for that

node, but does not activate it. While this option does not enable SNA communications over XCF for pure subarea nodes, it does allow TCP/IP communications through XCF to be enabled. If the node supports APPN, as it becomes aware of each other member node of the sysplex, VTAM builds a TRLE and APPN PU definition for that node, but does not activate the APPN PU. This option is useful in APPN configurations where XCF connectivity is desirable for TCP/IP communications but not for APPN communications.

- If XCFINIT=NO is specified, VTAM does not join the ISTXCF group at VTAM initialization. The operator can cause VTAM to join the ISTXCF group at a later time by activating the ISTLSXCF major node (using the V ACT,ID=ISTLSXCF command).

Activation of the APPN PU for VTAM connectivity through XCF requires that HPR support in the node be specified as (or defaulted to) HPR=RTP. [If HPR=(RTP,ANR) is specified, VTAM will still treat XCF links as having RTP-level HPR support, not just ANR-level support.]

When VTAM builds a dynamic APPN PU definition to support XCF connectivity to another VTAM node in the sysplex, it adds dynamic PU definitions to the ISTLSXCF major node as each VTAM joins the XCF group. The first four characters of the names of the PUs defined in this manner default to ISTP. The next two characters are the &SYSCLONE value of this VTAM. The last two characters are the &SYSCLONE value of the partner VTAM.

When VTAM builds a dynamic TRLE entry to describe the connectivity characteristics used for XCF connections, it adds the entry to the ISTTRL major node. The first four characters of the names of the TRLEs defined in the TRL major node default to ISTT. The next two characters are the &SYSCLONE value of this VTAM. The last two characters are the &SYSCLONE value of the partner VTAM.

Requirement: The XCF major node (ISTLSXCF) and the PU representing the XCF connection to the VTAM on the other side of the XCF connection must each be active for SNA connectivity between the two nodes to be established.

Example: As an example, consider a sysplex with two nodes, Node A and Node B. Each can be started as a pure subarea node or as an APPN node. The APPN node can have XCFINIT=YES, XCFINIT=NO, or XCFINIT=DEFINE. The pure subarea node can have XCFINIT=DEFINE or XCFINIT=NO. Each can be started at the z/OS V1R7 level or above or with a prior level of z/OS code (prior levels cannot have XCFINIT=DEFINE). Table 46 on page 447 shows the XCF connectivity between the two nodes for each of these combinations. To make the table more readable, the following points apply:

- If either node (or both nodes) specify XCFINIT=NO, no XCF connectivity is established. The node with XCFINIT=NO will not join the ISTXCF group and other VTAM nodes in the group will not be aware of that node's presence in the Sysplex and will not build support for XCF connectivity to that node.
- Nodes that specify XCFINIT=NO when started can subsequently be allowed to participate in the Sysplex group by activating the ISTLSXCF major node. At that point, pure subarea nodes behave as if XCFINIT=DEFINE had been specified, and APPN nodes behave as if XCFINIT=YES had been specified.
- Pure subarea nodes at the pre-V1R7 level default to XCFINIT=NO. No other XCFINIT value can be specified.

Therefore, in Table 46 on page 447, each node can be one of the following:

- A V1R7 or later pure subarea node specifying XCFINIT=DEFINE (or defaulted to DEFINE)
- A V1R7 or later APPN node specifying XCFINIT=DEFINE
- A V1R7 or later APPN node specifying XCFINIT=YES (or defaulted to YES)
- A pre-V1R7 APPN node specifying XCFINIT=YES (or defaulted to YES)

Table 46. VTAM to VTAM connection example

		Node B			
		V1R7 pure subarea XCFINIT= DEFINE	**V1R7 APPN XCFINIT= DEFINE**	**V1R7 APPN XCFINIT= YES**	**Pre-V1R7 APPN XCFINIT= YES**
Node A	**V1R7 pure subarea XCFINIT= DEFINE**	Result 1	Result 1	Result 1	Result 2
	V1R7 APPN XCFINIT= DEFINE	Result 1	Result 6	Result 6	Result 5
	V1R7 APPN XCFINIT= YES	Result 1	Result 6	Result 3	Result 3
	Pre-V1R7 APPN XCFINIT= YES	Result 7	Result 4	Result 3	Result 3

Results:

1. Node A and Node B have XCF TRLEs defined, but no XCF APPN PUs are defined. SNA connectivity is not available. TCP/IP stacks on the two nodes can establish connectivity.
2. Node A and Node B have XCF TRLEs defined. Node B has the XCF APPN PU defined and activated. The PU remains in Pending Request Contacted (PREQC) state until manually deactivated by the operator. SNA connectivity is not available. TCP/IP stacks on the two nodes can establish connectivity.
3. Node A and Node B have XCF TRLEs and XCF APPN PUs defined and activated. Each PU activation will complete. SNA connectivity will be established. TCP/IP stacks on the two nodes can establish connectivity.
4. Node A and Node B have XCF TRLEs and XCF APPN PUs defined. Node A has activated the XCF PU. The PU remains in Pending Request Contacted (PREQC) state until the operator either manually deactivates the PU on Node A or manually activates the PU on Node B. TCP/IP stacks on the two nodes can establish connectivity. If the PU on Node B is activated, SNA connectivity is established.
5. Node A and Node B have XCF TRLEs and XCF APPN PUs defined. Node B has activated the XCF PU. The PU remains in Pending Request Contacted (PREQC) state until the operator either manually deactivates the PU on Node B or manually activates the PU on Node A. TCP/IP stacks on the two nodes can establish connectivity. If the PU on Node A is activated, SNA connectivity is established.

6. Node A and Node B have XCF TRLEs and XCF APPN PUs defined, but the PU is not activated by either node. TCP/IP stacks on the two nodes can establish connectivity. If the XCF PU is activated on either Node A or Node B, SNA connectivity is established.
7. Node A and Node B have XCF TRLEs defined. Node A has the XCF APPN PU defined and activated. The PU remains in Pending Request Contacted (PREQC) state until manually deactivated by the operator. SNA connectivity is not available. TCP/IP stacks on the two nodes can establish connectivity.

You can define a model for dynamic XCF local SNA PUs in a model major node to override the XCF default values for the operands of the PU definition statement (for example, CPCP or TGP). Coding a TRLE operand signifies that the model is for XCF connections. Do not specify CONNTYPE=LEN on the XCF model definition. Only one model definition is in effect at a time. The first model activated is the one used for the dynamic PUs. DYNTYPE=XCF can be coded in a model major node definition.

You can change the default values for the first four characters of the dynamic PU names added to ISTLSXCF by specifying up to four characters as the name of the model PU. The default values for the first four characters of the dynamic TRLE names added to the ISTTRL major node can be changed by specifying up to four characters on the TRLE operand of the model PU.

Generic resources

A generic resource is a method of allowing multiple application programs to be known by a common name. For additional description of generic resources, see "Generic resources function" on page 406.

Generic resources requirements

Any VTAM application program running on a VTAM APPN node that is connected to an MVS coupling facility structure identified by the STRGR start option can be known by a generic resource name. All VTAM nodes having access to the common generic resource coupling facility structure, along with the structure itself, make up a generic resources configuration.

- A generic resources configuration must be part of a sysplex environment.
- All VTAMs in a generic resource configuration must be connected to an MVS coupling facility in which the generic resource structure can be allocated. VTAM connects to the generic resource structure when it is defined in the active CFRM policy.

  The generic resource coupling facility structure stores the data needed by VTAM to implement the generic resources function. This data includes mappings that show:

  – For each generic resource name, a list that identifies the generic resource members currently using that name. This is called a generic resource mapping.
  – For each LU in session with one or more generic resource members, the application program network name and the generic resource name for each generic resource member. This is called a partner LU mapping.

  In defining generic resource structure in a CFRM policy, you specify the amount of coupling facility storage which should be allocated on behalf of this structure. For information on determining the amount of storage necessary, see "Determining the size of the coupling facility structure" on page 441.

- VTAM Version 4 Release 2 or later is required for all VTAMs in a generic resources configuration, and all VTAMs must be defined as network nodes or end nodes. If an application program is both a generic resource and uses multinode persistent sessions, all VTAMs in the sysplex must be Version 4 Release 4 or later.
- If you are using subplexing (that is you have specified the XCFGRPID start option), all resources that share a generic resource name must reside within the same subplex. That is, they must be running on VTAMs that have been started with the same value for the XCFGRPID start option. In addition, the generic resource coupling facility structure within each subplex must have a unique name. To ensure this, the structure name specified by the STRGR start option is suffixed with the 2-character XCFGRPID start option value. You must ensure that each of these subplex structure names is defined in the active CFRM policy. For example, if you have two subplexes within your sysplex, one with XCFGRPID=11 and one with XCFGRPID=02, and your STRGR option specifies the structure name ISTMYGR, you must ensure that the CFRM policy defines both an ISTMYGR11 and an ISTMYGR02 generic resource structure. You can display the fully suffixed generic resource structure name accessed on a VTAM node by issuing the D NET,ID=VTAM command.
- If instances of a generic resource exist on an end node, CP-CP sessions must exist between the end node and its network node server.
- If instances of a generic resource exist in the domain of multiple network nodes, CP-CP sessions should exist between each pair of network nodes to ensure correct session setup.
- The OLU host of a session initiation to a generic resource must support name translation. The TRANSLAT start option must include USERVAR.
- A network node server is required to provide generic resource resolution for resources on end nodes within the generic resource configuration. The recommended APPN generic resource configuration consists of a network node and served end nodes connected to the same generic resource structure with a backup network node connected to the same structure. But a backup network node server that is not connected to the same generic resource structure can be used. This requires that the end nodes allow broadcast searches, by coding ENBCAST=YES on the NETSRVR list. Refer to the ENBCAST operand on the NETSRVR statement for the VTAMLST VBUILD TYPE=NETSRVR list in *z/OS Communications Server: SNA Resource Definition Reference*.

  If the network node server currently in use by the end nodes resides outside the sysplex, session setups to generic resources residing on these end nodes can fail if the NETSRVR list is not defined correctly. Symptoms of the failure would be similar to the following, where TSOXX is a generic resource name:

  ```
  IST663I  CD DSRLST  REQUEST TO   CDRMX  FAILED, SENSE=087D0001
  IST664I  REAL  OLU=netid.luname       ALIAS DLU=netid.TSOXX
  IST889I  SID = EC7F2F218E97F0CF
  IST1705I  SORDER = ADJSSCP  FROM START OPTION
  IST1705I  SSCPORD = PRIORITY  FROM START OPTION
  IST894I ADJSSCPS TRIED,FAILURE SENSE,ADJSSCPS TRIED,FAIL SENSE
  IST895I    ISTAPNCP       08400007     CDRMA         087D0001
  IST895I    CDRMB          087D000A     CDRMX         087D000A
  IST314I  END
  ```

- VTAM uses default settings to control generic resource resolution. To modify the default settings for all or specific generic resources you must define a generic resource preference table using GRPREF operands on the GRPREF statement for the VTAMLST VBUILD TYPE=GRPREFS statement. For more information see *z/OS Communications Server: SNA Resource Definition Reference*. In addition, refer to "Initiating sessions using the generic resource name" on page 454.

Generic resource mapping

A map of the application programs that are members of each generic resource is kept in a generic resource structure. When an LU initiates a session using a generic resource name, VTAM establishes the session with one of the members that is mapped to that name.

Figure 109 shows how the mapping identifies the application programs that are using the different generic resource names. In this example, APPLA, APPLB, and APPLC are represented to the network as JOHN. APPLD is using the generic resource name SEAN.

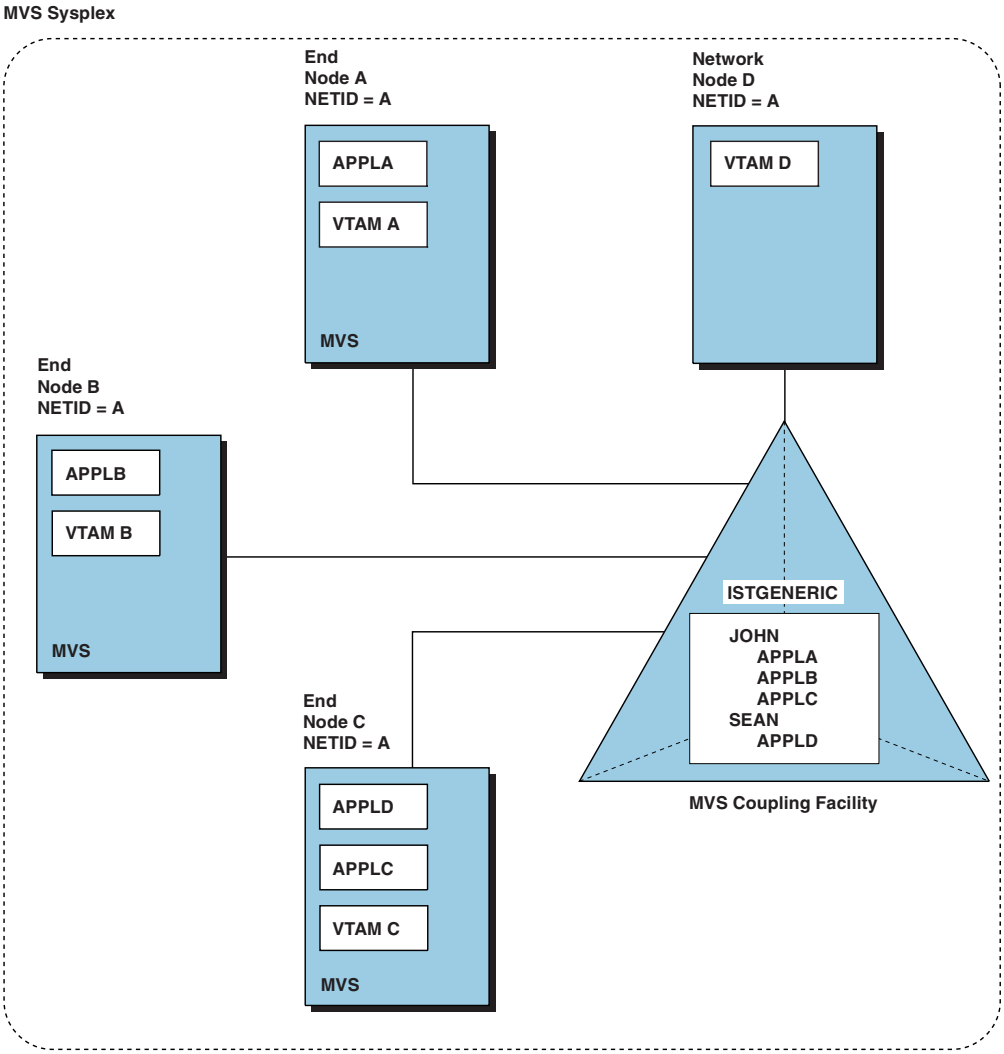


Figure 109. Generic resource mapping

Partner LU mapping

The generic resource structure also contains a mapping that identifies the partner LUs for each generic resource member. When the first session is established between an LU and a generic resource member, VTAM creates a mapping for that LU. This mapping, or affinity, identifies the name of the LU, the application program network name of the generic resource member, and the generic resource

name. When subsequent parallel sessions are established using the same generic resource name, VTAM uses this mapping to ensure the session is established with the same generic resource member.

Figure 110 on page 452 shows partner LU mapping information. In this example, LU5 is in session with APPLB using the generic resource name JOHN and is in session with APPLD using the generic resource name SEAN. LU3 is in session with APPLA using the name JOHN.

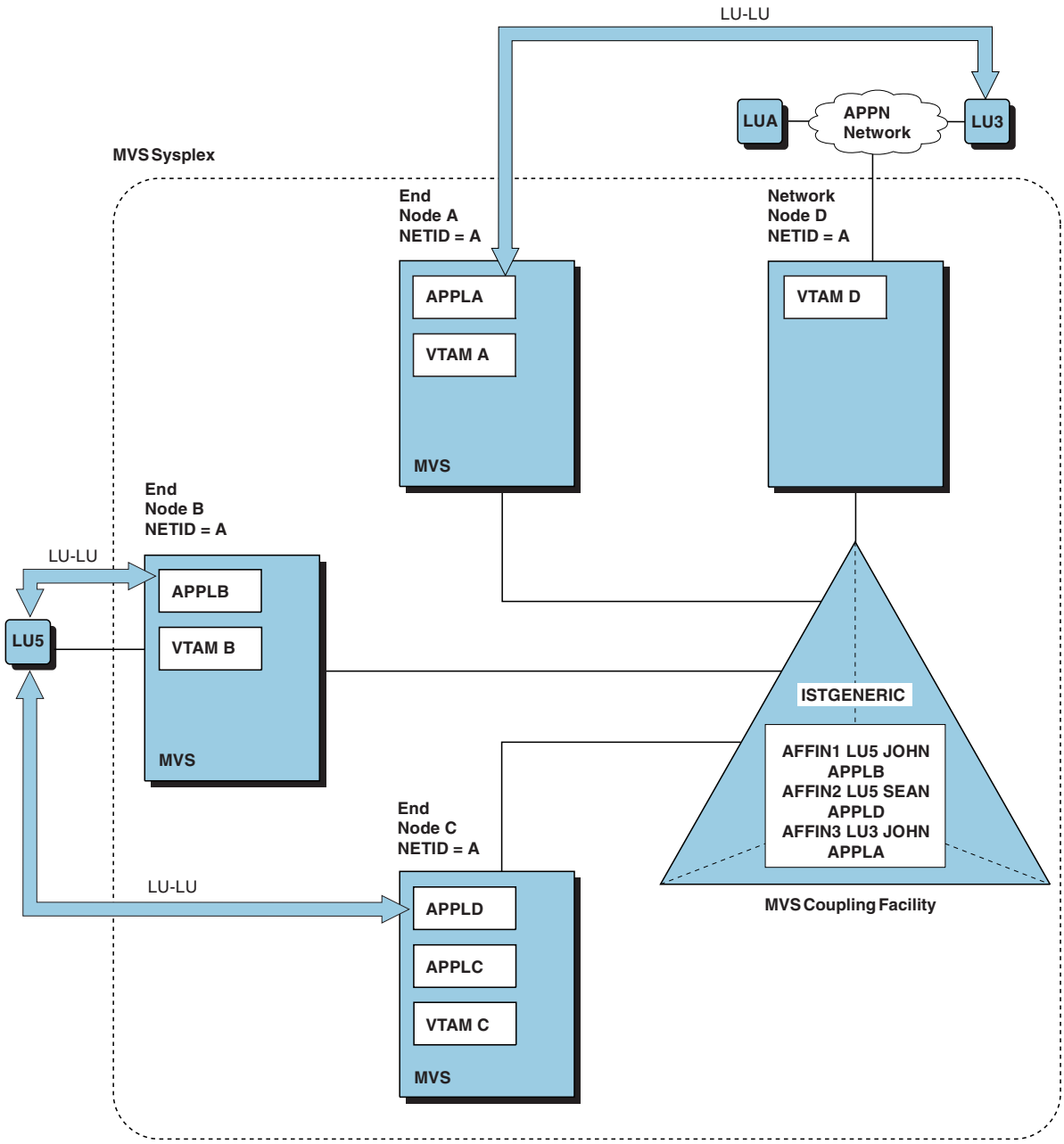


Figure 110. Partner LU mapping

When LUA initiates a session with JOHN, VTAM determines whether LUA has an affinity with a generic resource member for JOHN. Because LUA does not have an affinity with any generic resource member using the name JOHN, VTAM establishes the session with the generic resource member having the least number of LU-LU sessions (in this example, APPLC).

Figure 111 on page 453 shows the affinity that is created after LUA establishes a session with JOHN. VTAM resolves the session to APPLC; LUA is only aware of

having established a session with JOHN. All subsequent parallel sessions between LUA and JOHN resolve to the same generic resource member, APPLC, regardless of session count.

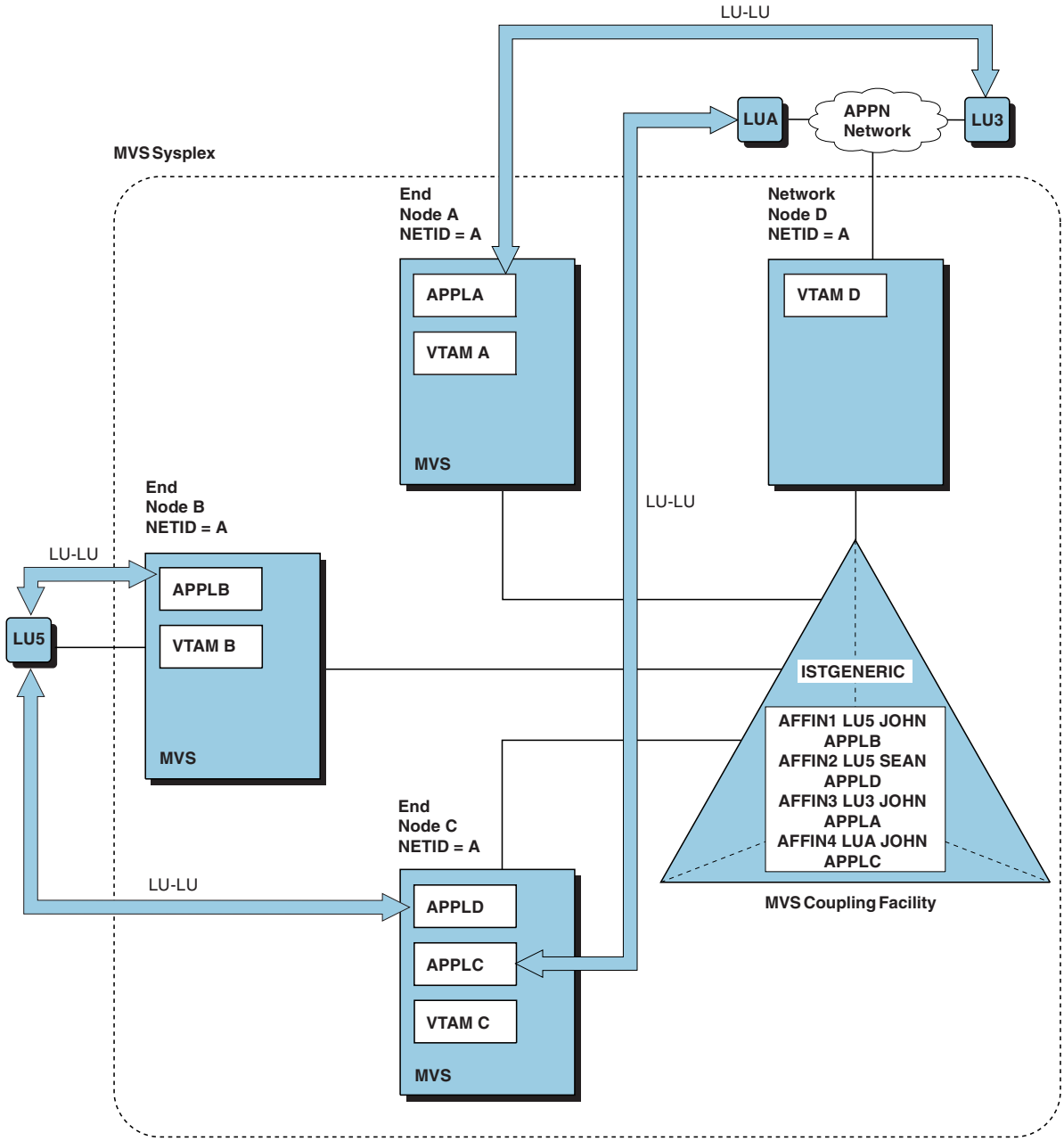


Figure 111. Session establishment with generic resource members

Affinities are deleted by their owner. Unless overridden by LUAFFIN=NOTAPPL specified for OPNDST, OPNSEC, or APPCCMD (CNOS, PREALLOC, or ALLOC) when a session is established, the application owns all affinities for sessions using:

- LU 6.2 synch-point services
- LU 6.2 limited resource support
- LU 6.1 protocols

The application program is also owner of all affinities when AFFIN=APPL is specified for the SETLOGON OPTCD=GNAMEADD issued by the application,

provided this is not overridden by LUAFFIN=NOTAPPL specified for OPNDST, OPNSEC, or APPCCMD (CNOS, PREALLOC, or ALLOC) when a session is established. The application program owns the affinity due to a particular session when LUAFFIN=APPL is specified for OPNDST, OPNSEC, or APPCCMD (CNOS, PREALLOC, or ALLOC) when a session is established.

The application program can delete an affinity by using the CHANGE OPTCD=ENDAFFIN macroinstruction, after all sessions between the application and the LU have terminated. The application program can assign the affinity ownership to VTAM, even if there are still sessions with the LU, by issuing the CHANGE OPTCD=ENDAFFNF macroinstruction. VTAM will delete VTAM owned affinities when the last session between the LU and the application has terminated. LU 6.2 sessions, whose affinities are not owned by the application, are retained for 10 minutes after the last session, or for some other user-specified period (specified by start option AFFDELAY). For further information, see *z/OS Communications Server: SNA Programming*.

Initiating sessions with generic resource members

Using generic resources, VTAM allows an application program to be known by a generic resource name. An application specifies its generic resource name on the GNAME operand of the NIB macroinstruction. The application establishes its association with the generic resource name on the SETLOGON OPTCD=GNAMEADD macroinstruction. VTAM uses RACF class VTAMAPPL to validate the generic resource name.

For any particular generic resource name, an LU can initiate a session using the generic resource name or the application program network name of the generic resource member, as shown in the following sections. For more information about how VTAM establishes sessions using the generic resources function, see *z/OS Communications Server: SNA Programming*.

Initiating sessions using the application program network name

If an LU accesses a generic resource member by the application program network name, VTAM establishes the session with that member, regardless of the current workload of that member. In addition, when the first session between an LU and a generic resource member is started using the generic resource member application program name, any subsequent session initiation attempt from that LU to the generic resource name results in a session with one of the other generic resource members. For example, in Figure 111 on page 453, if LUA initially used the application program name APPLC at logon, any subsequent session from LUA using the generic resource name JOHN is established with APPLA or APPLB.

Initiating sessions using the generic resource name

When an LU initiates a session using a generic resource name, VTAM performs generic resource resolution to identify a specific application name. VTAM determines whether the LU has an affinity, or mapping, with a generic resource member. If an affinity exists, VTAM establishes the session with the same generic resource member, regardless of current workload. (A session can be started with one of the other generic resource members using its application program network name.) If an affinity does not exist, VTAM uses the default or defined generic resource preferences from the generic resources preferences table to determine how to resolve the generic name. You can define a default generic resource preference or a generic resource preference for specific generic resources using the generic resources preference table (VBUILD type GRPREFS). See *z/OS Communications Server: SNA Resource Definition Reference* for the generic resource preference table.

The default generic resource resolution preferences for selecting a generic resource (unless overridden by defining a nameless entry in the GRPREFS table) are as follows:

- If a local application initiates a session to a generic resource, VTAM prefers generic resource members on the same VTAM node over those on other VTAM nodes. If no local generic resource members are active, then all generic resource members are eligible for resolution. This preference avoids the overhead associated with routing a session to other nodes. In many situations, you might want this local generic resource preference, but it can create a temporary or minor imbalance of session distribution. However, if the origin logical unit (OLU) is a session manager application or a TN3270 server through which most of the sessions in your network are started, VTAM always selects a local instance of a generic resource; this can significantly overload a single instance of a generic resource. By setting the LOCAPPL generic resource preference in the GRPREFS table, you can control the preference for picking local instances of a generic resource.
- If a local LU that is part of a local SNA or local non-SNA major node initiates a session to a generic resource, VTAM prefers generic resource members on the same VTAM node over those on other VTAM nodes. If no local generic resource members are active, then all generic resource members are eligible for resolution. This preference avoids the overhead associated with routing a session to other nodes. In many situations you might want this local generic resource preference, but it can create a temporary or minor imbalance in session distribution. However, if you have a large number of local LUs on the same host as the destination generic resource, VTAM always selects a local instance of a generic resource; this can significantly overload a single instance of a generic resource. By setting the LOCLU generic resource preference in the GRPREFS table, you can control the preference for picking local instances of a generic resource.
- If a generic resource resolution is from an LU that is not a local application or a local LU, then all generic resource members are eligible for resolution. However in the case of a third-party-initiated session (CLSDST-PASS) it might be preferable to select from generic resource members at the OLU CP for the session. If, for example, workload balancing had previously established the OLU of the session (for example, TN3270 using sysplex distributor), but the session setup path includes a CLSDST-PASS to a generic resource by a session manager on a non-OLU CP, you might prefer that the generic resource be resolved to a generic resource member on the OLU CP. This would avoid the overhead associated with routing the final session path to other nodes. By setting the PASSOLU generic resource preference in the GRPREFS table, you can control the preference for resolving to generic resource instances at the OLU CP of the session during CLSDST-PASS.
- VTAM calls the Workload Manager (WLM) to select an instance of the generic resource from all generic resource members that were previously determined to be eligible for this session. In most conditions, the workload at the time of the session initiation is the best way to determine how to resolve a generic name. However, you can set the WLM generic resource preference in the GRPREFS table to avoid calling the Workload Manager and instead resolve to a generic resource based on the fewest total number of active and pending sessions.
- IBM provides a default installation-wide generic resource exit routine (ISTEXCGR) that you can modify to select the member based on user-specified criteria. By default VTAM does not call the generic resource exit to select an instance of the generic resource. In most conditions, you can set generic resource preferences to customize generic resource resolution. However, you can cause the generic resource exit to be called using the GREXIT generic resource preference in the GRPREFS table.

The generic resource exit can use the information passed in the generic resource exit parameter list to identify an instance of the generic resource. This information includes the following:

- OLU name
- Generic resource name
- Eligible generic resource instances
- Session counts
- WLM's best generic resource instance
- Best session load balanced generic resource instance

The exit is called only if no affinity exists and after the other generic resource preferences have been used to identify eligible generic resource instances. Only the previously determined eligible generic resource instances are passed to the exit. For more information, see the generic resource resolution exit routine in *z/OS Communications Server: SNA Customization*.

Implementation considerations

When implementing the generic resources function, the following considerations apply to the generic resources configuration:

- All generic resource members with the same generic resource name must reside within the same generic resource configuration. This means that all hosts where a generic resource member resides, and any resource selector nodes for that generic resource, must be in the same sysplex and be connected to the same coupling facility structure.
- LUs that initiate sessions using a generic resource name can be located outside the generic resource configuration that supports that generic resource name.
- Using two network node servers maintains generic resource resolution capability should one of the network nodes fail. A backup network node server outside the sysplex can be used. Refer to the ENBCAST operand on the NETSRVR statement for the VTAMLST VBUILD TYPE=NETSRVR list in *z/OS Communications Server: SNA Resource Definition Reference*.
- Performance can be enhanced by placing generic resource members at end nodes in the generic resource configuration. This frees up network nodes to provide routing services.
- An application program can be known by only one generic resource name.
- Generic resource members using the same generic resource name must have the same NETID.
- Generic resource names must be unique within a single network. A generic resource name cannot be identical to:
  - A USERVAR
  - An alias name
  - A real LU name
  - An ACBNAME
- The generic resources function supports multiple copies of the same application program; however, VTAM does not verify that each application program known by a generic resource name provides the same function.
- If your network has a security manager product installed, the security level for the generic resource name must be the same as the security level for all application programs that are using that generic resource name. The security level of the generic resource name is determined by the first generic resource member that associates itself with that generic resource name. If there is no

security product, any application program residing on a VTAM that is connected to an MVS coupling facility can use any generic resource name.

- Using the generic resources function, VTAM distributes sessions across generic resource members rather than evenly across hosts. VTAM evaluates the current usage of each application program, regardless of the owning host. When activating generic resource members, consider the host capabilities. If VTAM A has twice the capability of VTAM B, VTAM A should support twice as many generic resource members.
- The MVS workload manager can also be used to balance allocation of sessions across hosts.

Coupling facility failures for generic resource configuration

There are three types of failures that can cause generic resource data to be lost:

- Failure of a VTAM node; all of the local data of the node is lost.
- Failure of a link between a VTAM node and generic resource structure; the VTAM node involved is unable to access data in the structure.
- Failure of generic resource structure; all of the data in the structure is lost.

Failure of a VTAM node

All local data in a VTAM node that needs to persist through a VTAM failure is replicated in the generic resource structure. If other VTAMs are present, they provide the necessary cleanup, and generic resources on the failing node can be restarted on another VTAM node. Affinities that should persist through a VTAM failure include LU 6.1 sessions, LU 6.2 sessions with SYNCPT synchronization support, multinode persistent sessions, and affinities claimed by the application through SETLOGON AFFIN=APPL, OPNDST/OPNSEC LUAFFIN, or APPCCMD LUAFFIN. If a failing VTAM has no local persistent data, a peer VTAM performing cleanup can delete the failed VTAM connection.

Note: If a VTAM connection to the generic resource structure is in failed-persistent state, an operator should not delete it.

Also, the generic resource structure remains allocated until all connections to it are deleted. Because there are no VTAMs to perform cleanup for the last VTAM to disconnect, you do need to delete the last connection to the generic resource structure to deallocate the structure.

Failure of a link between a VTAM node and generic resource structure

When a VTAM node loses its link to the generic resource structure, it initiates a structure rebuild to attempt to rebuild the generic resource structure on another coupling facility to which all VTAM nodes in the generic resource configuration have access. The rebuild structure process can be stopped by a peer node that is either unable to connect to, or loses its connection to, the new version of the generic resource structure.

If the generic resource structure is rebuilt to a coupling facility that is not accessible by all nodes, the VTAM that has no link to the current version of the generic resource structure disconnects from the structure. Other VTAMs provide the necessary cleanup, and once the failing link is repaired the disconnected VTAM attempts to reconnect to the generic resource structure.

Failure of the generic resource structure

When generic resource structure fails, each VTAM in a generic resource configuration is notified and attempts to initiate a structure rebuild to create a new version of the generic resource structure. The new version of the structure is replenished from the local data of each VTAM node in the generic resource configuration.

Removing a generic resource

After a generic resource name is defined, it can prevent other resources on VTAMs in the same sysplex from being defined with the same name. Therefore, there might be a need to delete a generic resource name in order to assign the name to another resource. One reason this might occur is because of the relationship between the USERVAR function and Generic Resources. USERVAR performs a similar function to generic resources and often customers migrate from USERVAR to generic resources. However, in preparation for migrating to use generic resources, a customer might choose to temporarily test a generic resource name before returning to using the identical USERVAR name. This would require the deletion of the generic resource name before the identical USERVAR name could be used.

When an application registers as a generic resource, several types of information related to the generic resource are created and maintained within VTAMs in the sysplex and within the generic resource coupling facility structure. This information includes a GENERIC USERVAR name, a generic mapping, and affinities.

- A GENERIC USERVAR name represents the generic name and is created on every VTAM in the sysplex that contains an application instance of the generic name, or has resolved the generic name.
- A generic mapping also represents the generic name, along with every application instance associated with the generic name. It is created in the generic resource coupling facility structure.
- Affinities are created when sessions are started to the generic resource. They relate specific LUs to specific application instances of the generic resource. There are two different types of affinities: application owned and VTAM owned. Application-owned affinities require action on the part of the generic resource application before they are terminated, while VTAM-owned affinities end when the corresponding session ends.

Deletion of a generic resource name requires the deletion of all of this information.

Steps for removing a generic resource

Perform the following steps to remove a generic resource. This example deletes a generic resource named neta.grappl, which has three real instances.

1. Identify all instances of the generic resource.

   Issue `DISPLAY NET,ID=grappl,IDTYPE=GENERIC` at every host in the sysplex. The following sample output is generated:

```
d net,id=grappl,idtype=generic
IST097I DISPLAY ACCEPTED
IST075I NAME = GRAPPL, TYPE = GENERIC RESOURCE
IST1359I MEMBER NAME        OWNING CP   SELECTABLE  APPC
IST1360I NETA.NETAPPL1      SSCP2A         YES       NO
IST1360I NETA.APPL1         SSCP1A         NO        NO
IST1360I NETA.APPLAA1       SSCPAA         DEL       NO
IST1393I GENERIC RESOURCE NAME RESOLUTION EXIT IS ISTEXCGR
IST314I END
```

This display shows three real instances, each on a different VTAM, associated with the grappl generic resource.

- The NETA.NETAPPL1 instance has issued an OPEN ACB and is selectable as a generic resource from any host in the sysplex.
- The NETA.APPL1 instance has issued an OPEN ACB but SSCP1A does not have CP-CP sessions with a network node server, and therefore, NETA.APPL1 is not selectable from any host in the sysplex. However, it is still registered as a generic resource and is selectable for sessions that originate on SSCP1A.
- The NETA.APPLAA1 instance has either CLOSED its ACB, or remained OPEN and issued the VTAM API command to make itself not selectable.

---

2. Terminate all sessions with all instances of the generic resource.

   Session information can be displayed using the VTAM commands `DISPLAY SESSIONS,LU1=`*`real_instance_name`*`,SCOPE=ALL` or `DISPLAY NET,ID=`*`real_instance_name`*`,SCOPE=ALL`.

   Note: `DISPLAY NET,ID=`*`real_instance_name`* output indicates that the application is active (message IST486I), its generic name (message IST1363I), and its session partners (message IST635I).

```
d net,id=appl1,scope=all
 IST097I DISPLAY ACCEPTED
 IST075I NAME = NETA.APPL1, TYPE = APPL
 IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
 IST1447I REGISTRATION TYPE = CDSERVR
 IST1363I GENERIC RESOURCE NAME GRAPPL REPRESENTS NETA.APPL1
 IST977I MDLTAB=***NA*** ASLTAB=***NA***
 IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
 IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
 IST1632I VPACING =  7
 IST1938I APPC = NO
 IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
 IST231I APPL MAJOR NODE = APPL1A
 IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
 IST1500I STATE TRACE = OFF
 IST271I JOBNAME = ECHO, STEPNAME = ECHO, DSPNAME = IST3C391
 IST228I ENCRYPTION = OPTIONAL , TYPE = DES
 IST1563I CKEYNAME = APPL1 CKEY = PRIMARY CERTIFY = NO
 IST1552I MAC = NONE MACTYPE = NONE
 IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
 IST1633I ASRCVLM = 1000000
 IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 0
 IST171I ACTIVE SESSIONS = 0000000001, SESSION REQUESTS = 0000000000
 IST206I SESSIONS:
 IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
 IST635I LAA3270A ACTIV-S    EAABEE18F0E34F0F 0002 0001       NETA
 IST314I END

d net,id=netappl1,scope=all
 IST097I DISPLAY ACCEPTED
 IST075I NAME = NETA.NETAPPL1, TYPE = APPL
 IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
 IST1447I REGISTRATION TYPE = CDSERVR
 IST1363I GENERIC RESOURCE NAME GRAPPL REPRESENTS NETA.NETAPPL1
 IST977I MDLTAB=***NA*** ASLTAB=***NA***
 IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
 IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
 IST1632I VPACING =  7
 IST1938I APPC = NO
 IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
 IST231I APPL MAJOR NODE = APPL2A
 IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
```

```
 IST1500I STATE TRACE = OFF
 IST271I JOBNAME = ECHO, STEPNAME = ECHO, DSPNAME = IST36CF1
 IST228I ENCRYPTION = OPTIONAL , TYPE = DES
 IST1563I CKEYNAME = NETAPPL1 CKEY = PRIMARY CERTIFY = NO
 IST1552I MAC = NONE MACTYPE = NONE
 IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
 IST1633I ASRCVLM = 1000000
 IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 0
 IST171I ACTIVE SESSIONS = 0000000002, SESSION REQUESTS = 0000000000
 IST206I SESSIONS:
 IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
 IST635I L2A3270A ACTIV-S    F6ABEEC3A58FC5CF 0001 0000       NETA
 IST635I L1A3270A ACTIV-S    EAABEEC3A28FC3B2 0001 0000       NETA
 IST314I END


d net,id=applaa1,scope=all
 IST097I DISPLAY ACCEPTED
 IST075I NAME = NETA.APPLAA1, TYPE = APPL
 IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
 IST1447I REGISTRATION TYPE = CDSERVR
 IST1363I GENERIC RESOURCE NAME GRAPPL REPRESENTS NETA.APPLAA1
 IST977I MDLTAB=***NA*** ASLTAB=***NA***
 IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
 IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
 IST1632I VPACING =  7
 IST1938I APPC = NO
 IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
 IST231I APPL MAJOR NODE = APPLAA
 IST212I ACBNAME = FIRSTONE
 IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
 IST1500I STATE TRACE = OFF
 IST271I JOBNAME = ECHO, STEPNAME = ECHO, DSPNAME = ISTC5DC7
 IST228I ENCRYPTION = OPTIONAL , TYPE = DES
 IST1563I CKEYNAME = APPLAA1 CKEY = PRIMARY CERTIFY = NO
 IST1552I MAC = NONE MACTYPE = NONE
 IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
 IST1633I ASRCVLM = 1000000
 IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 0
 IST171I ACTIVE SESSIONS = 0000000001, SESSION REQUESTS = 0000000000
 IST206I SESSIONS:
 IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
 IST635I LAA3270B ACTIV-S    EAABEE18F0E34F11 0001 0000       NETA
 IST314I END
```

The DISPLAY NET,SESSIONS command can also show the session partners (message IST874I).

```
d net,sessions,lu1=netappl1,scope=all,list=all
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = SESSIONS
 IST873I        PLU               SLU              SID        STATUS
 IST874I NETA.NETAPPL1     NETA.L2A3270A     F6ABEEC3A58FC5CF ACTIV
 IST874I NETA.NETAPPL1     NETA.L1A3270A     EAABEEC3A28FC3B2 ACTIV
 IST924I -------------------------------------------------------------
 IST878I NUMBER OF PENDING SESSIONS =          0
 IST924I -------------------------------------------------------------
 IST878I NUMBER OF ACTIVE  SESSIONS =          2
 IST924I -------------------------------------------------------------
 IST878I NUMBER OF QUEUED  SESSIONS =          0
 IST924I -------------------------------------------------------------
 IST878I NUMBER OF TOTAL   SESSIONS =          2
 IST314I END
```

---

3. Delete all affinities associated with all real instances of the generic resource.

Affinities associate a session partner to a generic resource and a specific member of the generic resource. They are created when the first session from a session partner is started to a generic resource. As long as the affinity exists, all subsequent sessions between the session partner and the generic resource will be directed to the specific member maintained in the affinity.

Affinities can be displayed using the `DISPLAY GRAFFIN,LU=*.*,GNAME=grappl` command.

```
d net,graffin,lu=*.*,gname=grappl
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = GENERIC AFFINITY
IST1706I PARTNER NAME       GENERIC RESOURCE    MEMBER     ATTRIBUTES
IST1707I NETA.LAA3270A      NETA.GRAPPL         APPL1      -VG--W--
IST1707I NETA.L1A3270A      NETA.GRAPPL         NETAPPL1   -VG--W--
IST1707I NETA.LAA3270B      NETA.GRAPPL         APPL1AA1   -AG6-W--
IST1454I 3 AFFINITIES DISPLAYED FOR LU=*.*
IST314I END
```

This display indicates the following about the affinities associated with grappl:

- The affinities for LAA3270A and L1A3270A are owned by VTAM (V).
- The affinity for LAA3270B is application owned (A) and persistent (6).
- All affinities indicate the associated sessions were started using the generic name (G).
- All affinities were created based on input from the MVS Workload manager (W).

All sessions between the session partner and the generic resource member must end before any associated affinities can be deleted. The different types of affinities would then be terminated using differing rules:

- An affinity that is VTAM owned is deleted when the last session between the session partner and the generic resource ends.
- An affinity that is application owned and persistent can be deleted only when the application notifies VTAM to delete the affinity. An affinity that is application owned, but not persistent, is deleted if the application closes its ACB or notifies VTAM to delete the affinity. Deleting persistent or application-owned affinities requires the application to be active and invocation of an application-unique command, or application termination. Refer to application-specific documentation to determine how to delete application-owned and persistent affinities.

  Sessions can be terminated by application users, application operator commands, or the VTAM command `VARY NET,TERM,LU1=`*`real_instance_name`*.

  ```
  v net,term,lu1=laa3270a
   IST097I VARY ACCEPTED
   IST457I POSITIVE TERM COMMAND RESPONSE
   IST455I LU1=NETA.LAA3270A SESSIONS ENDED
  ```

  The following display shows that the VTAM-owned affinity for LAA3270A is deleted after the session is terminated.

  ```
  d net,graffin,lu=*.*,gname=grappl
  IST097I DISPLAY ACCEPTED
  IST350I DISPLAY TYPE = GENERIC AFFINITY
  IST1706I PARTNER NAME       GENERIC RESOURCE    MEMBER     ATTRIBUTES
  IST1707I NETA.L1A3270A      NETA.GRAPPL         NETAPPL1   -VG--W--
  IST1707I NETA.LAA3270B      NETA.GRAPPL         APPL1AA1   -AG6-W--
  IST1454I 2 AFFINITIES DISPLAYED FOR LU=*.*
  IST314I END
  ```

---

4. Close the VTAM ACB for every real instance of the generic resource.

In some cases, VTAM retains information about the generic resource capability of the application as long as the application exists, so the application must perform CLOSE ACB processing. The CLOSE ACB can be triggered by an application-unique command or the VTAM command VARY INACT,ID=*real_instance_name* can be used.

```
v net,inact,id=applaa1
 IST097I VARY ACCEPTED
 IST105I APPLAA1 NODE NOW INACTIVE
```

The command DISPLAY ID=*real_instance_name* can be used to verify that every instance of the generic resource has closed its ACB. This command needs to be issued at the owning host of each real instance to verify that the CLOSE ACB has been performed.

```
d net,id=applaa1
 IST097I DISPLAY ACCEPTED
 IST075I NAME = NETA.APPLAA1, TYPE = APPL
 IST486I STATUS= INACT, DESIRED STATE= INACT
 IST1447I REGISTRATION TYPE = CDSERVR
 IST977I MDLTAB=***NA*** ASLTAB=***NA***
 IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
 IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
 IST1632I VPACING =  7
 IST1938I APPC = NO
 IST597I CAPABILITY-PLU INHIBITED,SLU INHIBITED,SESSION LIMIT NONE
 IST231I APPL MAJOR NODE = APPLAA
 IST212I ACBNAME = FIRSTONE
 IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
 IST1500I STATE TRACE = OFF
 IST271I JOBNAME = ***NA***, STEPNAME = ***NA***, DSPNAME = ***NA***
 IST228I ENCRYPTION = OPTIONAL , TYPE = DES
 IST1563I CKEYNAME = APPLAA1 CKEY = PRIMARY CERTIFY = NO
 IST1552I MAC = NONE MACTYPE = NONE
 IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
 IST1633I ASRCVLM = 1000000
 IST1634I DATA SPACE USAGE: CURRENT = ***NA*** MAXIMUM = ***NA***
 IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
 IST314I END
```

Note: Message IST486I indicates that the application is inactive.

---

5. Identify all hosts in the sysplex that contain a GENERIC USERVAR.

   GENERIC USERVAR mappings can exist on a VTAM because a generic name was once resolved on this host. To identify whether a GENERIC USERVAR exists, issue DISPLAY RSCLIST,ID=grappl,IDTYPE=GENERIC on every host in the sysplex.

```
d net,rsclist,id=grappl,idtype=generic
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = RSCLIST
 IST1417I NETID    NAME     STATUS     TYPE              MAJNODE
 IST1418I NETA     GRAPPL   ACTIV      GENERIC RESOURCE  **NA**
 IST1418I NETA     GRAPPL   *NA*       GENERIC USERVAR   **NA**
 IST314I END
```

   This display indicates that the resource name GRAPPL with resource type GENERIC RESOURCE exists in the generic resource coupling facility structure. It also indicates that this VTAM has defined a type GENERIC USERVAR for GRAPPL.

---

6. Delete the generic resource information from the sysplex.

   When all sessions have been terminated for all real instances, all affinities are deleted, and all real instances have performed CLOSE ACB processing, the

generic resource grappl can be successfully deleted. Issue the command MODIFY GR,OPTION=DELETE,GNAME=netid.grappl at every host in the sysplex that has a GENERIC USERVAR for the generic resource. This should include every host where a real instance resided, but could also include other hosts in the sysplex.

All members of the generic resource are not selectable.

```
d net,id=grappl
 IST097I DISPLAY ACCEPTED
 IST075I NAME = GRAPPL, TYPE = GENERIC RESOURCE
 IST1359I MEMBER NAME        OWNING CP   SELECTABLE  APPC
 IST1360I NETA.APPL1         SSCP1A         DEL       NO
 IST1360I NETA.NETAPPL1      SSCP2A         DEL       NO
 IST1360I NETA.APPLAA1       SSCPAA         DEL       NO
 IST1393I GENERIC RESOURCE NAME RESOLUTION EXIT IS ISTEXCGR
 IST314I END
```

All affinities have been deleted.

```
d net,graffin,lu=*.*,gname=grappl
JOB    2  IST097I DISPLAY ACCEPTED
JOB    2  IST350I DISPLAY TYPE = GENERIC AFFINITY
 IST1358I NO QUALIFYING MATCHES
 IST1454I 0 AFFINITIES DISPLAYED
 IST314I END
```

The command to delete the generic resource must be issued at every host in the sysplex that has a GENERIC USERVAR for the generic resource.

```
modify vtam,gr,gname=neta.grappl,option=delete
 IST097I MODIFY ACCEPTED
 IST223I MODIFY GR COMMAND COMPLETED
```

If a MODIFY GR,DELETE command is issued at a host that has temporarily lost connectivity to the generic resource coupling facility structure, then generic resource data local to the host can still be deleted. However the MODIFY GR DELETE command must complete successfully from at least one host connected to the generic resource coupling facility structure in order to delete the generic resource data in the structure.

When the generic resource name has been deleted, another resource such as an application or USERVAR can be defined with the same name. For instance, the MODIFY USERVAR command can be used to create a USERVAR for the name grappl.

```
modify vtam,uservar,id=grappl,option=update,value=appl1
 IST097I MODIFY ACCEPTED
 IST825I USERVAR DEFINED - NAME = GRAPPL, VALUE = APPL1
 IST314I END

d net,id=grappl
 IST097I DISPLAY ACCEPTED
 IST075I NAME = APPL1, TYPE = APPL
 IST113I GRAPPL IS A USERVAR WITH VALUE APPL1 IN NETWORK NETA
 IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
 IST1447I REGISTRATION TYPE = CDSERVR
 IST599I REAL NAME = NETA.APPL1
 IST977I MDLTAB=***NA*** ASLTAB=***NA***
 IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
 IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
 IST1632I VPACING =  7
 IST1938I APPC = NO
 IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
 IST231I APPL MAJOR NODE = APPL1A
 IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
 IST1500I STATE TRACE = OFF
 IST271I JOBNAME = ECHO, STEPNAME = ECHO, DSPNAME = ISTB972A
 IST228I ENCRYPTION = OPTIONAL , TYPE = DES
 IST1563I CKEYNAME = APPL1 CKEY = PRIMARY CERTIFY = NO
 IST1552I MAC = NONE MACTYPE = NONE
```

```
IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
IST1633I ASRCVLM = 1000000
IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 0
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
```

Note: Message IST113I indicates that GRAPPL is a USERVAR for APPL1.

In general, you should use generic resources because they have a greater degree of availability and workload distribution. However, if you use the MODIFY GR,DELETE command to consolidate a group of generic resource applications to a single application represented by a USERVAR, then caution should be taken to ensure that the single USERVAR application can handle the consolidated workload.

Routine maintenance for VTAM nodes

Before bringing a VTAM down for routine maintenance, it is recommended that any existing affinities be ended. VTAM automatically ends most affinities when halted or during a failure. However, for LU 6.1 and synch point-capable sessions, and application-owned affinities, VTAM maintains affinities between LUs and their generic session partners at both of the following:

- The node of the generic session partner
- The coupling facility

To end these affinities a CHANGE ENDAFFIN must be issued. Refer to *z/OS Communications Server: SNA Programming* for information on the CHANGE ENDAFFIN macroinstruction. The consequences of not ending the affinities are the following:

- A session attempting to use one of the affinities while the application's ACB is closed will fail.
- If the ISTGENERIC structure fails, the affinities will be lost.

Multinode persistent sessions

VTAM uses the coupling facility in the MVS sysplex to maintain session and HPR connection information for all multinode persistent session (MNPS) application programs. This allows VTAM to restore application sessions after instances of system or application failure, or as part of takeover of the application.

- For application program failures, an application can be restarted on the same VTAM on which it failed, through the single node persistent session (SNPS) support, or on a different VTAM through MNPS planned takeover or MNPS forced takeover.
- For planned or forced takeover or for operating system, hardware, or VTAM failures, a multinode persistent session application program can be restarted on any VTAM node that supports multinode persistent sessions in the sysplex, with the following considerations:
  - If a multinode persistent application program moves to the VTAM where one of its session partners is located, the sessions with that partner will not be restored. The sessions are terminated and must be restarted. This is also true if the multinode persistent application program moves to the VTAM that is the partner endpoint of the HPR connections associated with the application session.
  - If both session partners are located on the same VTAM when the failure occurs, the sessions between them cannot be restored. The sessions are terminated and must be restarted.

– If the multinode persistent application moves to a VTAM node that is not connected to the multinode persistent session coupling facility structure, no sessions are restored.

Applications capable of utilizing triple-DES encryption lose the triple-DES sessions if the application is moved to a node running a level of Communications Server for OS/390® that does not support triple-DES sessions. Also, if the application requires triple-DES encryption (ENCRTYPE = TDES24 was coded on the definition statement), the application is not allowed to successfully issue OPEN ACB on a node running a level of Communications Server for OS/390 that does not support triple-DES.

Until the application program is restarted, incoming data is maintained at the other end of the HPR connection. When the application is restarted and before the sessions are restored, VTAM stores the incoming data.

VTAM uses Control Vector 29 to preserve the session state. This includes the last sequence number, RH, and the first five bytes of the RU for each expedited and normal PIU sent outbound and inbound. At restart, VTAM passes that information to the application to resynchronize with the session partner. It is recommended that you check your application to verify that it makes use of this information.

The backup of a multinode persistence-enabled application program is shown in Figure 112 on page 466.

1. **APPL1 is in session with APPL2. The session is multinode persistent session (MNPS) capable and enabled. APPL1 has specified persistent on the APPL definition statement. Session data is being stored in the MNPS coupling facility structure.**

2. **When VTAM fails, another VTAM in the sysplex detects the error. The application is marked pending recovery. Each VTAM connected to the MNPS structure starts a timer (PSTIMER value). If the timer expires before recovery is successful, one of the VTAMs in the sysplex performs cleanup of the application program's session data in the MNPS structure.**

3. **Either through an operator or the automatic restart manager (ARM), APPL1 can be restarted on another multi-node persistent session capable VTAM.**

   In this case, APPL1 has been restarted on VTAM3.

   The sessions are then restored using the data saved in the coupling facility structure.

Figure 112. Application program backup using multinode persistent sessions

Multinode persistent session support is enabled if single node persistence has been enabled, PERSIST=MULTI has been specified, and the configuration requirements are met. The PERSIST operand on the APPL definition statement allows you to make an application program multinode persistence-capable (however, single node persistence is always available).

To change an application program from multinode persistence to single node persistence (and vice versa) you must first close and deactivate the application, modify the APPL definition statement (PERSIST operand), reactivate the

application program major node, and start the application program. When operating on a network node, use a model definition for your MNPS applications, because PERSIST=MULTI cannot be coded on a nonmodel application definition if VTAM is a network node.

MNPS forced takeover processing requires some additional settings at the taking over application to indicate that this is a forced takeover attempt. In particular, PARMS=(FORCETKO=YES) must be coded on the ACB macroinstruction. Similarly, additional action is required at the application being taken over to indicate that forced takeovers are acceptable. In particular, PARMS=(FORCETKO=ALL) or PARMS=(FORCETKO=MULTI) must be coded on a SETLOGON OPTCD=PERSIST macroinstruction.

Multinode persistent session configuration requirements

Note: HALT and HALT QUICK override multinode persistent session capability of all application programs on the VTAM where the command is issued. HALT CANCEL will *not* override multinode persistent session capability. VARY INACT,TYPE=FORCE overrides multinode persistent session capability for a specific application program.

Following are the requirements for implementing multinode persistent sessions:

- A persistent application program must run on a VTAM node that is part of a sysplex environment.
- VTAM nodes in the sysplex wanting to support multinode persistent application programs must be running under a release of MVS that supports coupling facility services. The coupling facility must be using, at a minimum, coupling facility control code (CFCC) level 1.
- All VTAMs in the multinode persistent session (MNPS) configuration must be connected to an MVS coupling facility in which the MNPS structure can be allocated. The MNPS coupling facility structure stores the data and session information needed by VTAM to recover a persistent session.

  For any MNPS using cryptography (and assuming that ICSF is active on all nodes where the MNPS application can operate), each z/OS Communications Server node that can recover an MNPS application should have the same host master key defined so a session key enciphered at one host is valid at the recovery host.
- In defining the structure in a CFRM policy, you specify the coupling facility storage that should be allocated on behalf of this structure. For information on determining the amount of storage necessary, see “Determining the size of the coupling facility structure” on page 441.
- All VTAM nodes providing multinode persistent session support must be defined as Rapid Transit Protocol (RTP) level nodes (HPR=RTP start option).
- z/OS Communications Server V1R6 or higher is required to implement MNPS forced takeover.
- If you are using subplexing (that is, you have specified the XCFGRPID start option), all MNPS application recoveries must be within the same subplex. That is, you must specify your ARM policy such that, when recovering a failed MNPS application, the new MNPS application instance is started within the same subplex (on a VTAM node that specified the same XCFGRPID start option value) as the original MNPS application. In addition, the MNPS coupling facility structure within each subplex must have a unique name. To ensure this, the structure name specified by the STRMNPS start option is suffixed with the 2-character XCFGRPID start option value. You must ensure that each of these

subplex structure names is defined in the active CFRM policy. For example, if you have two subplexes within your sysplex, one with XCFGRPID=11 and one with XCFGRPID=02, and your STRMNPS option specifies the structure name ISTMYMNPS, you must ensure that the CFRM policy defines both an ISTMYMNPS11 and an ISTMYMNPS02 MNPS structure. You can display the fully suffixed MNPS structure name accessed on a VTAM node by issuing the D NET,ID=VTAM command.

- If using APPC/MVS as a persistent application, you must specify the PERSIST keyword. PERSIST has been added to the LUADD statement in the APPCPMxx parmlib member. When you code this keyword, APPC/MVS issues a persistent close when LUDEL is issued for one of its LUs.
- To receive the full benefit of multinode persistent session support, meshed (meaning every node has a one-hop connection to every other node in the sysplex) connectivity should exist between nodes in the sysplex. Also, all sysplex network nodes, even if no multinode persistent session applications plan to operate on the network node, should support RTP tower to guarantee recoverability of sessions.

  In addition, if you plan to operate MNPS applications on the network nodes in the sysplex, you need to extend HPR into the network beyond the sysplex. In particular, all nodes physically adjacent to the network node or nodes where the application can operate should be configured to perform RTP tower function; otherwise, sessions which use paths through the non-RTP tower nodes will not be recoverable, because there will be no underlying HPR connection.

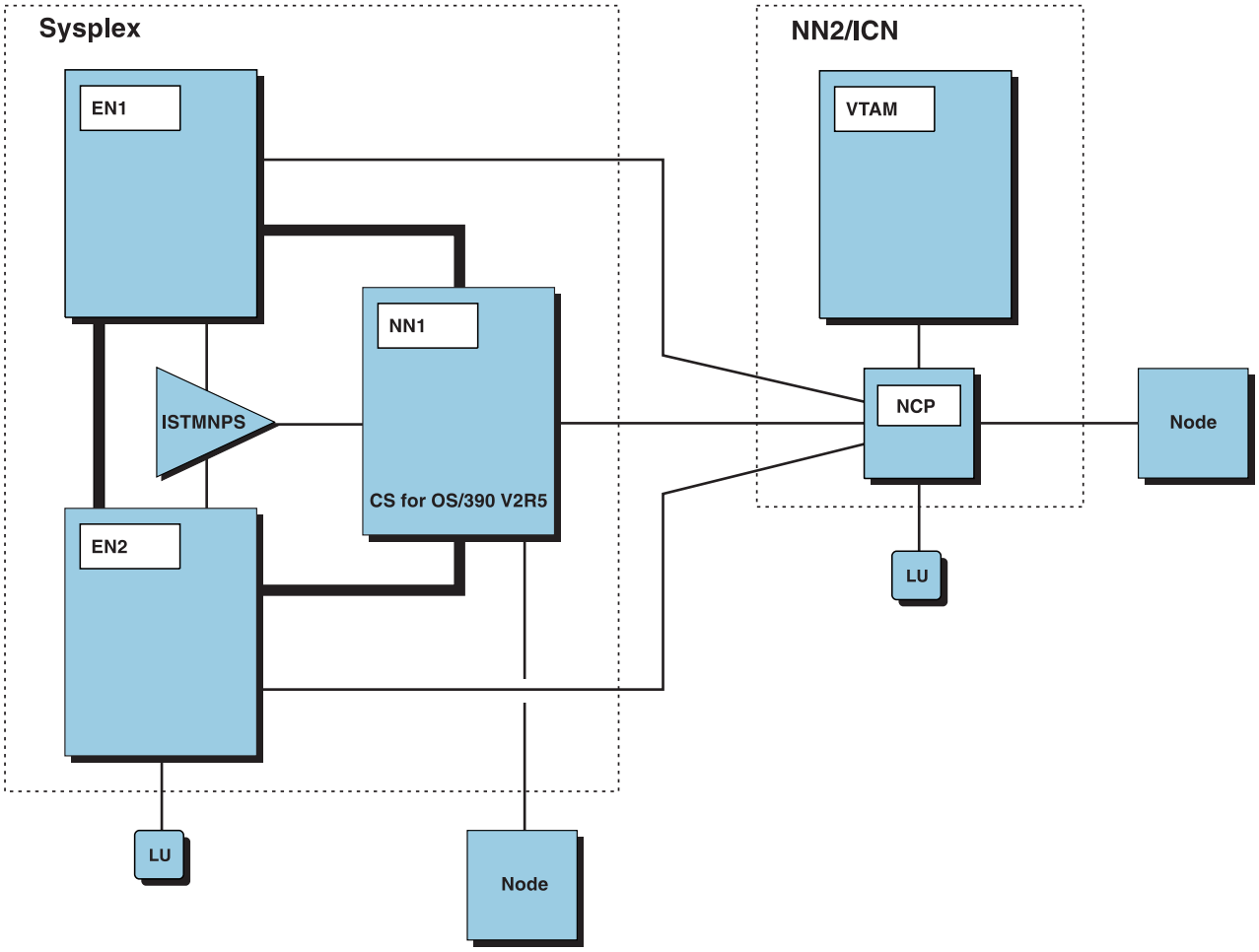


Figure 113. Multinode persistent session network example

Figure 113 shows an example network configuration supporting multinode persistent sessions. EN1 and EN2 are both connected to the coupling facility. They are using the default name for the multinode persistent sessions structure, ISTMNPS. Also, in the sysplex is NN1, which is the network node server for the two end nodes. Outside of the sysplex is an interchange node, NN2, which has connections to all of the VTAMs in the sysplex. EN1, EN2, and NN1 are all RTP-capable nodes. It is optional for NN2 to be RTP capable; however, because NN1 is connected to the ISTMNPS structure and MNPS applications are anticipated to operate on NN1, NN2 should be configured to be RTP-capable as well. This will allow HPR connections to be established from NN1 to NN2 for sessions involving MNPS applications on the network node.

Note: If you plan to use VR-based TG connectivity in the sysplex, it is recommended that you also define at least one non-VR-based TG connection between any end node supporting MNPS application programs and its adjacent nodes. This will allow MNPS processing to calculate session paths to the end node that uses the non-VR-based TG and increase the probability that an HPR connection is used. See "Session route setup at an end node" on page 471 for more information.

Using multiple coupling facility structures for multinode persistent sessions

Multiple coupling facility structures can be defined to increase the number of multinode persistent sessions supported in the sysplex. Multiple structures can be defined on a single coupling facility or across multiple coupling facilities. However, defining multiple structures on one coupling facility increases coupling facility overhead and is not advised.

To implement multiple multinode persistent session structures, define the structures in the MVS policy for each associated coupling facility. VTAM attempts to connect to a base structure and any alternate structures when they are defined in the CFRM policy. See "Setting up the sysplex environment for VTAM and TCP/IP functions" on page 436 for information on defining structures used for multinode persistent sessions.

The structure name defined by the STRMNPS start option is the base structure. All other structures used for multinode persistent support are called alternate structures. The name of any alternate structure must begin with the name of the base structure followed by any number of unique and valid characters up to a total of 16 characters. For example, if you specify STRMNPS=ISTMNPS, an alternate structure name could be ISTMNPS01. Care should be taken when using subplexing to avoid generating an MNPS structure name for use in one subplex that might be mistaken as an alternate MNPS structure for a VTAM in another subplex. For example, if a VTAM is started in subplex 11 (that is, start option XCFGRPID=11 is specified), and STRMNPS=ISTMYMNPS is specified, and another VTAM in the sysplex is started with the default subplex (that is, start option XCFGRPID is not specified) and STRMNPS=ISTMYMNPS is specified, then the structure name generated for the MNPS structure in subplex 11 will be ISTMYMNPS11, which will appear to the second VTAM to be an alternate structure to its MNPS structure, ISTMYMNPS. Refer to *z/OS Communications Server: SNA Resource Definition Reference* for additional information concerning the STRMNPS start option.

You can use the DISPLAY STATS command to view information about the base structure and its associated alternate structures.

Each structure on a coupling facility must have a unique name. If you are using multiple structures, the number of characters used to specify the name of the base structure (STRMNPS start option) must not be too large (to allow unique names for all the alternate structures).

Establishing multinode persistent sessions

When the application opens its ACB, it must indicate that persistent sessions are supported (PARMS=(PERSIST=YES)). The application must also include PERSIST=MULTI on its application definition. An MNPS application must be defined using a model application definition if the application operates at a network node. Upon completion of the OPEN ACB, the following information is added to the multinode persistent session structure:

- Name of the owning VTAM
- Application program in persistence disabled state
- Capabilities of the application (for example: NQN, APPC, and MACRF)

The application program then issues SETLOGON OPTCD=PERSIST to enable persistent LU-LU session support. If an application program wants to disable support for multinode persistent LU-LU sessions, SETLOGON OPTCD=NPERSIST

should be issued. In both cases, the application program state in the coupling facility is updated to show “persistence enabled” or “persistence disabled,” respectively.

You can issue the DISPLAY ID command to determine if application program sessions are recoverable.

- If the command is issued from the owning VTAM, the /M status modifier in message IST635I indicates that the session might be recoverable by multinode persistent session (MNPS) support. The current MNPS state is also displayed.

  ```
  IST1550I MNPS STATE = ENABLED
  .
  .
  .
  IST635I APPLAA1 ACTIV/M-P EAABEE1819963497 0000 0000 0 0 NETA
  ```

- If the command is issued from another VTAM connected to the coupling facility structure in the sysplex message, IST1549I is issued. Message IST1549I contains MNPS STATE=ENABLED if the application program is recoverable by MNPS support.

  ```
  IST1549I OWNER = NETA.SSCP1A MNPS STATE = ENABLED
  ```

Because sessions established for multinode persistent application programs use separate HPR connections from sessions not involving MNPS application programs, you might notice an increase in the number of HPR connections.

Session route setup at an end node

During session establishment, the end node provides a list of TG tail vectors to the network node server used in the computation of the session route. Because HPR is required for the recovery of a multinode persistent session, the EN tries to ensure the session route will transverse an HPR connection, by only providing to the NN server the endpoint TGs that connect the EN to RTP-capable nodes. This guarantees that the portion of the session path ending at the end node is HPR-capable and, therefore, the session is potentially recoverable. This might result in indirect routes when routing outside the sysplex. For example, in Figure 114 on page 472, assume that APPL1 wishes to establish a session with APPL2.

- Without multinode persistent session enabled, the session path chosen is directly between MDH1 and VTAM1.
- With multinode persistent sessions enabled, the session path chosen is MDH1 to NN1 to VTAM1. So, the route for the session between APPL1 and APPL2 is a two-hop route.

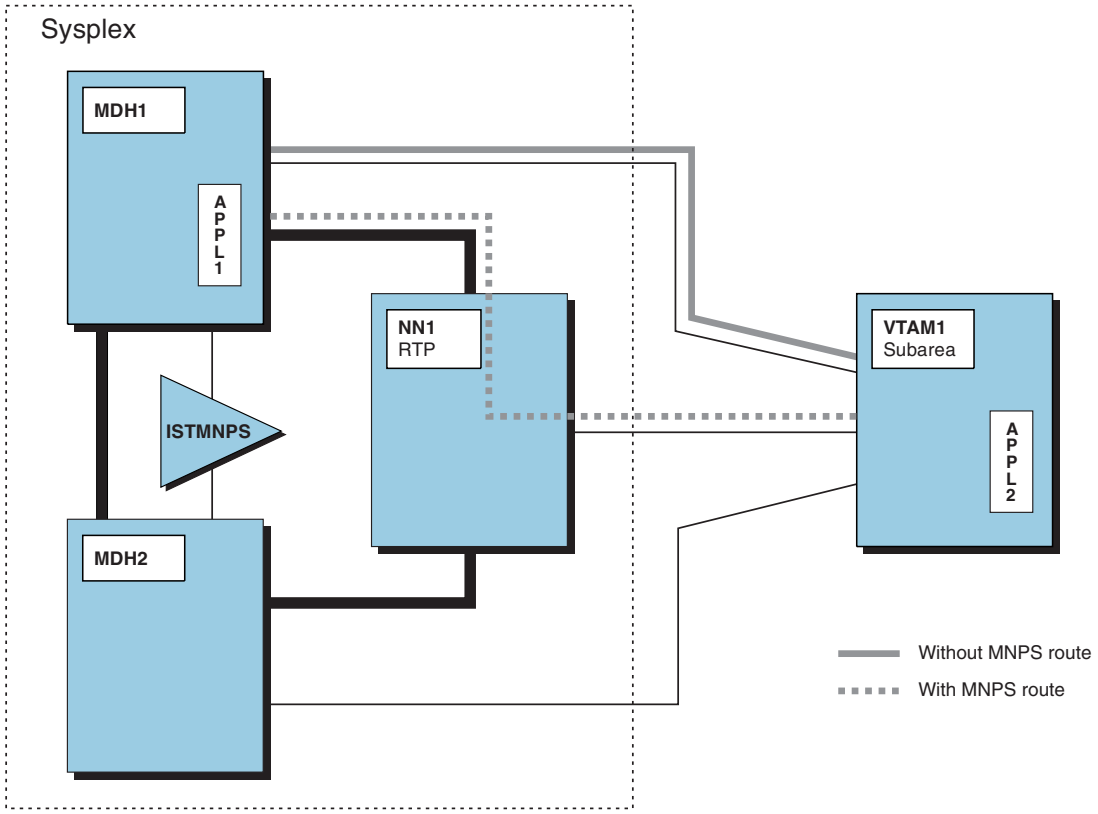


Figure 114. Session routes

Note: If APPL2 requests a session with APPL1 and VTAM1 sends the request directly to MDH1 using subarea protocols, the session is established without an HPR connection and is not recoverable. You might want to modify your search order to force the session requests for multinode persistent application programs to go through APPN over an HPR connection. See "Controlling searches" on page 519 for additional information.

If during the APPN search no route is found, VTAM will establish the session using subarea routes if available. In this instance, the session is not recoverable even if the application program is multinode persistent-enabled.

After the session path has been established, VTAM can force a path switch to select a better session route. This is necessary because during the setup for the original path, the EN reports only HPR-capable links to the network node server. During session establishment:

- Any TG that connects an RTP-capable node to an ANR-capable node is omitted; however, these TGs might provide a better session route.
- VR-based TGs are not included in the subset of tail vectors if there are any other TGs that can be used. Recall that a VR-based TG can represent connections to either VTAM, NCP, or both. Also, note that at this time, the path to the partner LU is not known. If a VR-based TG is selected for the session path and if the partner LU is connected to the NCP in a composite network node, an HPR connection cannot be established because an NCP cannot be an RTP endpoint. Because the end node is trying to guarantee an HPR connection, the VRTG is not included due to the possibility of NCP being an endpoint.

  For the better session route calculation, however, the VR-based TG can be included, because the other endpoint of the connection is now known and the

network node server will only consider HPR-capable connections to that endpoint while trying to obtain a better session route.

A path switch is performed if an HPR-capable route is found and a different end node TG is selected for the route. After the path is determined and the session is established, VTAM places the appropriate information in the multinode persistent session coupling facility structure.

For example, in Figure 115 the session is established between APPL1 and APPL2 using EN1-NN1-NN3-EN3. The TGs considered when establishing the route were TGA, TGC, and TGD, because they guaranteed an RTP connection.

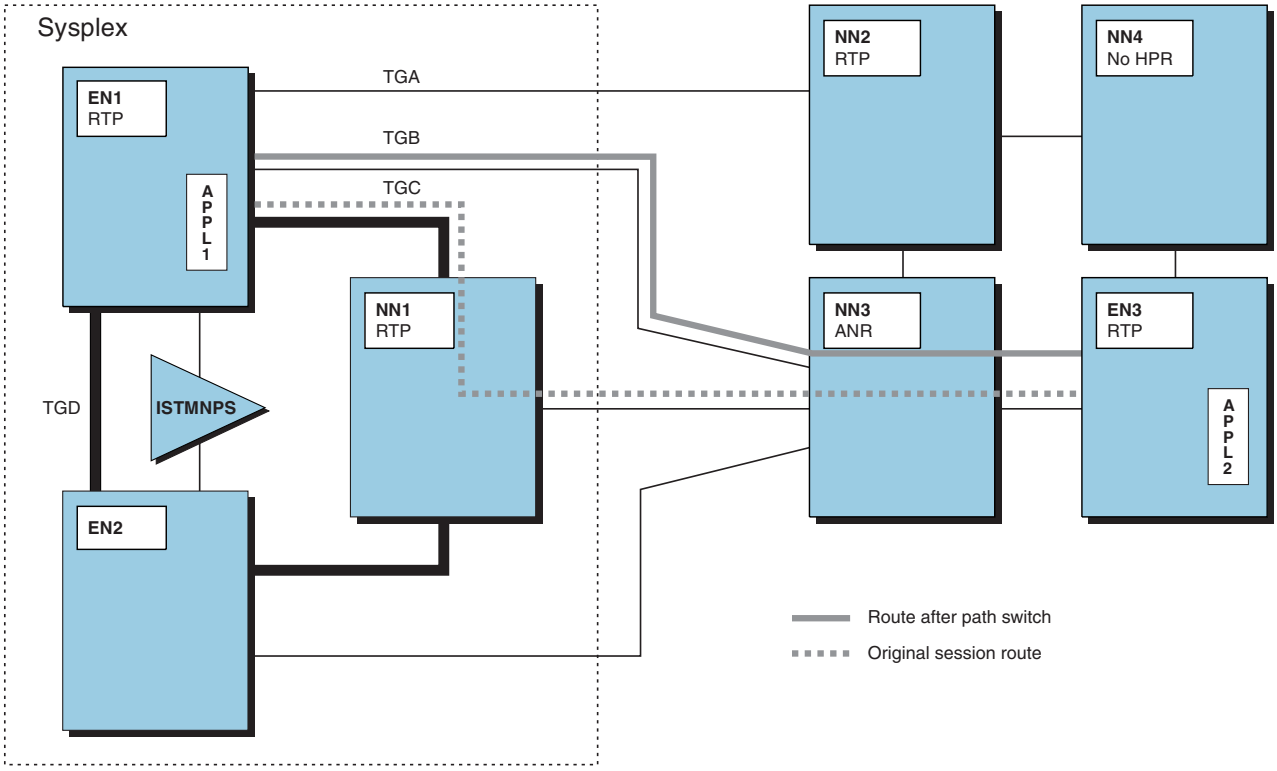


Figure 115. Path switch processing

After the session is established, EN1 attempts to find a better session route by initiating path switch processing. This time, TGB is included as a possible transmission group, and as a result the session route EN1-NN3-EN3 is selected. Because a different TG is used than in the original session route calculated, the data path for the HPR connection is switched to the better route.

If a better session path exists, and the data (or physical) path of the connection is switched to that path, the nonmultinode persistent partner is not told of the new path. The nonmultinode persistent partner remains unaware of the new path so that subsequent sessions will reuse the established HPR connection, since those sessions will, like the first, have their routes calculated using the subset of connections to RTP-capable nodes.

Because the nonmultinode persistent partner is not aware of the real data path, when the second session is set up, there appears to be an HPR connection already established for the route (EN1-NN1-NN3-EN3). Therefore, the connection is reused. The MNPS EN, which is aware of the different physical path being used, will

compare new session routes against the computed session path, and not the physical path, and will also be able to reuse existing HPR connections that traverse the same computed session path.

The DISPLAY ID=RTP_PU_resource command can be used to display the session path information known at a given node:

- A DISPLAY ID=RTP_PU_resource command, when issued at the multinode persistent node, provides both the computed session path and the physical path (if different) for the HPR connections used by multinode persistent application programs.
- A DISPLAY ID=RTP_PU_resource command provides the computed session path or actual path (being used) at the nonmultinode persistent session application program endpoint (the physical path).

Session route setup at a network node

Session establishment of multinode persistent sessions when the application is operating on a network node is less complicated than when the application is operating on an end node. The reason for this is that there is no concept of a list of TG tail vectors to report for a network node. All the transmission groups associated with a network node are known throughout the network, so any APPN network node in the network can calculate a session route to the MNPS NN using the network topology information. Because this information is known everywhere, there is no mechanism at the network node (such as TG tail vector lists) to force session paths to be calculated that will result in sessions that can be recovered by multinode persistent session processing.

To guarantee that session paths to the MNPS NN use HPR connections, it is required that the network configuration be managed in such a way that all nodes adjacent to the MNPS NN are RTP-tower capable. Additionally, all transmission groups connecting the MNPS NN to the adjacent RTP-tower capable nodes should be configured to support HPR connections. In such a configuration, regardless of the session path calculated, the portion of the path that terminates at the MNPS NN will consist of an HPR-capable connection to an RTP-tower capable node. This allows an HPR connection be established for at least that portion of the session path. The presence of the HPR connection allows the sessions using that connection to be recovered by multinode persistent session processing.

Coupling facility failures for multinode persistent session configuration

There are three types of failures which can cause multinode persistent data to be lost:

- Failure of a VTAM node; all local data for the node is lost.
- Failure of a link between a VTAM node and coupling facility structure; the VTAM node involved is unable to access or update data in the structure.
- Failure of a multinode persistent session (MNPS) structure; all of the data in the structure is lost.

Failure of a VTAM node

All other VTAMs connected to the MNPS structure will detect the VTAM failure. One of these VTAMs will process any data in the structure owned by the failing VTAM. For all MNPS application programs owned by the failing VTAM that are not enabled for recovery, all application data will be removed from the structure. For all MNPS application programs owned by the failing VTAM that are enabled for recovery, the application program will be marked recovery pending. This will

allow the application program to recover its sessions on a different VTAM node. See “Failure recovery processing” for information about data recovery when a VTAM node fails.

Failure of a link between a VTAM node and multinode persistent structure

When a VTAM node loses its link to an MNPS structure and no system failure management (SFM) policy is in effect, it initiates a structure rebuild to attempt to rebuild the MNPS structure on another coupling facility to which all VTAM nodes have access. If an SFM policy is in effect, VTAM will be directed by MVS to either disconnect or start a rebuild. If the rebuild was started by VTAM because no SFM policy was in effect, the rebuild can be stopped by a peer node that is either unable to connect to, or loses its connection to, the new version of the MNPS structure. A rebuild started by MVS due to SFM policy is never stopped by VTAM.

If the MNPS structure is rebuilt to a coupling facility that is not accessible by all nodes, any VTAM that has no connectivity to the current version of the MNPS structure disconnects from the structure. Other VTAMs will mark all applications owned by the disconnected VTAM as suspect. This indicates that the MNPS data is not up to date. Once the failed link is repaired, the disconnected VTAM attempts to reconnect to the MNPS structure. If the connection is successful, the reconnecting VTAM will refresh all MNPS data for applications it owns from local data and reset the state to indicate the current persistent state of the application program as enabled or disabled.

Note: You should avoid performing planned takeover when a rebuild is underway, because the application will be viewed as suspect by the takeover VTAM, and the sessions will not be moved successfully.

Failure of an MNPS structure

When an MNPS structure fails, each VTAM connected to the structure is notified and attempts to initiate a structure rebuild to create a new version of the structure. The new version of the structure is replenished from the local data of each VTAM. Any data for applications that were recovery pending will be lost.

During a structure rebuild for an MNPS structure, data for the new version of the structure is always replenished from local data. In the case that an MNPS application program owning VTAM has failed and the state is recovery pending, if a rebuild occurs before the application program data is recovered, the data will be copied to the new version of the structure only if at least one VTAM has connectivity to both the original version of the structure and the new version of the structure. If this is not the case, the data is lost and the sessions cannot be recovered.

Failure recovery processing

When a VTAM connects to the multinode persistent session coupling facility structure in the sysplex, VTAM indicates that it wants to be notified when other connections to the structure end. The following describes the recovery processing for a persistent-enabled application program.

1. First, the other VTAMs are notified that there is a node failure. One of the other VTAMs that is connected to the multinode persistent session structure marks the failing VTAM persistent-enabled applications as recovery pending.
2. The other VTAMs then clean up the failing VTAM persistent-disabled application programs from the top of the coupling facility. Once the coupling facility structure is cleaned up, no recovery can occur.

Note: From the perspective of the nonfailing partner, the sessions are still active. The sessions will remain active until the HPR connection is terminated.

3. For all persistent-enabled application programs, a timer is started with the PSTIMER value specified by the application program. There is a timer for each MNPS application program residing in the failed VTAM. The PSTIMER indicates the amount of time an application can remain in recovery pending state. The recovering application must successfully issue an OPEN ACB before the timer expires on one of the VTAMs or VTAM cleans up the application program session information in the MNPS coupling facility structure.

   Note: Affinities associated with generic resource application programs that are multinode persistent session-capable remain in the generic resource coupling facility structure until the timer expires. If the generic resource application program is not recovered, affinities are not deleted until the application program is restarted.

For recovery to occur, the application program must be persistent-enabled and the sessions must have traversed an HPR connection. If these conditions are met and there is a VTAM in the sysplex that is connected to the MNPS coupling facility structure, recovery can take place. The following describes the recovery process:

1. An application is started either through an operator command or the automatic restart manager (ARM). The application opens an ACB with the same application program name as the application program being recovered. Recovery can occur on the same VTAM that the application program resided on (if that VTAM has been restarted) or on a different VTAM.
2. The recovering VTAM obtains the capabilities of the application being recovered from the coupling facility structure. The capabilities of the recovering application program must match those of the application program being recovered for recovery to continue.

   Note: The use of model definitions to define the application program reduces the chances of capabilities not matching. The use of model definitions for MNPS applications is required at a network node.

   Refer to *z/OS Communications Server: SNA Programming* for details on the actual capabilities that must match.
3. To indicate that the application has been recovered, the information in the coupling facility structure is updated to reflect the new owning VTAM and the application program is marked in persistence-ENABLED state.
4. The new owning VTAM determines whether it has a CDRSC with the same name as the new application program. If a CDRSC definition exists and VTAM determines that the CDRSC represents the recovering application program when it was active on its previous owning VTAM, the sessions associated with the CDRSC are terminated and the CDRSC becomes a shadow resource. This must be done because an application program of the same name now exists.

   If recovery occurs at a VTAM DLUS node, any sessions between the application and the DLUR-dependent LUs served by the VTAM DLUS network node are the exceptions to this processing. If the DLUS is intermediate or absent on the HPR connection path, the DLUS has only minimal session awareness of these sessions. These sessions are maintained and associated with the HPR connection that is to be rebuilt during MNPS recovery processing. Although the sessions are maintained, they are disconnected from the CDRSC representation

of the application at this point in the processing and associated with the APPL RDTE that now represents the application. This allows OPEN ACB processing to continue.

5. The new owning VTAM reads in all the application program data from the coupling facility into a VTAM data space.
6. The new owning VTAM performs a path switch to reestablish a route for the session. During the path switch processing, the recovering side of the session will wait 90 seconds (or 4 minutes if both application programs are MNPS). If the connection is not reestablished, recovery is not successful.

   The other end of the HPR connection is cleaned up when its path switch timer pops. For HPR connections being used by multinode persistent sessions, the minimum value for the path switch timer is four minutes. However, if a higher value is specified for the path switch timer (for VTAM, this is the HPRPST start option) at either session partner, it will be used.

   Note: The values of the PSTIMER and HPRPST should be coordinated to synchronize effective use.
7. The new owning VTAM re-creates the necessary session control blocks from the information in the coupling facility. The application program must restore the session, using the OPNDST RESTORE command, before session data traffic can resume completely. Refer to the *z/OS Communications Server: SNA Programming* for additional information. Some applications might reset the sessions before resuming data traffic and if they do that, they are able to reduce the performance impact of MNPS by using NIBNTRCK on OPNSEC and OPNDST when establishing sessions.

   Note: Applications using APPCCMD API will use the APPCCMD RESTORE command.

During recovery, you might notice a peak in storage usage on the new owning VTAM. When recovery is complete, storage usage should go back down.

A VARY INACT of an HPR connection being recovered because of multinode persistence processing will not be processed until recovery processing has completed.

Note: The OPEN will fail for an application program recovering for a failed generic resource application program if the recovering VTAM does not support generic resources. It will also fail if the recovery VTAM is attached to a generic resource structure name different from that of the MNPS application (the associated structure name is saved in the coupling facility).

MNPS planned and forced takeover processing

The two types of MNPS takeover processing that can be performed are:

- MNPS planned takeover is a more orderly takeover activity, forcing the target application to first enter a state, such as SNPS recovery pending, which indicates that recovery of some sort is necessary.
- MNPS forced takeover is a more dynamic takeover activity, in which the takeover application requests that the application to be taken over be shut down even in cases where the target application is currently active and operational.

During either type of takeover attempt, you might notice a peak in storage usage on the new owning VTAM. Once takeover is complete, storage usage should go

back down. A VARY INACT of an HPR connection being moved because of multinode persistence recovery processing will not be processed until takeover processing has completed.

Restriction: The OPEN will fail for an application program recovering for a failed generic resource application program if the recovering VTAM does not support generic resources or does not connect to the same generic resource structure.

MNPS planned takeover

For an MNPS planned takeover to occur, the application program must be in pending SNPS recovery state. An application enters this state when it has enabled persistence and then issues CLOSE ACB or fails. The state is maintained by the owning VTAM in the multinode persistent sessions coupling facility structure, so that other VTAMs in the sysplex are aware that the application is eligible for planned takeover processing.

Note: The D NET,ID command can be used to verify the state of any MNPS application within the sysplex.

If using APPC/MVS as a persistent application, you must specify the PERSIST keyword to support planned takeover. PERSIST has been added to the LUADD statement in the APPCPMxx parmlib member. When you code this keyword, APPC/MVS issues a persistent close when LUDEL is issued for one of its LUs.

The following describes the planned takeover process:

1. Either through an operator command or the automatic restart manager, an application opens an ACB with the same application program name as the application program that failed or closed its ACB.
2. The recovering VTAM obtains the capabilities of the application being recovered from the coupling facility structure. The capabilities of the recovering application program must match those of the application program being recovered for takeover to continue. Refer to *z/OS Communications Server: SNA Programming* for details on the actual capabilities that must match.
3. The new owning VTAM determines if it has a CDRSC with the same name as the new application program. If a CDRSC definition exists and VTAM determines that the CDRSC represents the recovering application program when it was active on its previous owning VTAM, the sessions associated with the CDRSC are terminated and the CDRSC becomes a shadow resource. This must be done because an application program of the same name is attempting to move to this node.

   If recovery occurs at a VTAM DLUS node, any sessions between the application and DLUR-dependent LUs served by the VTAM DLUS network node are the exceptions to this processing. If the DLUS is intermediate or absent on the HPR connection path, the DLUS has only minimal session awareness of these sessions. These sessions will be maintained and associated with the HPR connection that is to be rebuilt during MNPS recovery processing. Although the sessions are maintained, they will be disconnected from the CDRSC representation of the application at this point in the processing and associated with the APPL RDTE that now represents the application. This allows OPEN ACB processing to continue.
4. OPEN ACB processing is performed for the application at the takeover VTAM, but no response is posted to the application. VTAM must first obtain ownership of the application from the current owning VTAM before the OPEN ACB can be considered to be a success.

5. The VTAM attempting to acquire ownership sends a takeover request to the current VTAM (by XCF). If accepted, the current owning VTAM updates the multinode persistent session structure to show the takeover VTAM as the application owner, and then sends a positive takeover reply.
6. When the takeover VTAM receives the positive takeover reply, VTAM will claim the structure storage lists associated with the application and will mark the application as being enabled for persistence. The response to the OPEN ACB is then posted to the application, because this VTAM has now acquired ownership of the application.
7. After transferring ownership, the previous owning VTAM cleans up its local representations of the sessions, using care to prevent any modifications to the coupling facility data. The HPR connections associated with the sessions are also cleaned up locally, although the other endpoint is unaware of the processing as VTAM will not send, or handle any received, information on these connections. The image of the application is also cleaned up at the previous owning VTAM. Any sessions not involving HPR pipes are terminated at this point by the previous owning VTAM.

   If the previous owning VTAM is a DLUS network node, minimal awareness of sessions between the application and DLUR-dependent LUs served by this VTAM DLUS will be maintained after the local session representation is cleaned up. This minimal session awareness must be kept to provide accurate session limit management for the dependent LUs. To maintain this session awareness, a CDRSC representation of the application must either be reused from the shadow set of resources (if a predefined CDRSC had existed for the application prior to OPEN ACB having been performed), or a CDRSC must be created dynamically. If no CDRSC can be reused or built (for example, CDRDYN=NO was specified as a start option), then session awareness cannot be maintained, and the session is terminated.
8. The takeover VTAM performs path switches to make this VTAM the new endpoint for the HPR connections associated with the application. During the path switch processing, the recovering side of the session will wait 90 seconds for the other endpoint to restart the HPR connection. If no response is received, recovery does not complete.

   The other end of the HPR connection is cleaned up if its path switch timer pops. For HPR connections being used by multinode persistent sessions, the minimum value for the path switch timer (for VTAM, this is the HPRPST start option) is four minutes. However, if a higher value is specified on the HPRPST start option, it will be used.
9. The new owning VTAM re-creates the necessary session control blocks from the information in the coupling facility. The application program must restore the session, using the OPNDST RESTORE command or APPCCMD RESTORE (if using the APPCCMD API), before session data traffic can resume completely.

Note: The OPEN will fail for an application program recovering for a failed generic resource application program if the recovering VTAM does not support generic resources and connect to the same generic resource structure.

MNPS forced takeover

MNPS forced takeover processing can be performed when the application program is in a wider range of states, the common state characteristic being that the application is active and has enabled persistence. The application state is maintained by the owning VTAM in the multinode persistent sessions coupling facility structure. Other VTAMs in the Sysplex currently use the state information

when processing OPEN ACBs or VTAM failure conditions, and now will use it to determine if the application is eligible for forced takeover processing.

In addition to state requirements, the application being taken over must indicate that it will accept forced takeover requests. This capability is expressed by coding PARMS=(FORCETKO=ALL) or PARMS=(FORCETKO=MULTI) on the SETLOGON OPTCD=PERSIST macroinstruction used to enable persistence in the first place. This capability is saved into the MNPS structure, much like the state of the application.

The D NET,ID command can be used to verify the current takeover capability of the application, both at the owning VTAM and at remote VTAMs connected to the same coupling facility structure. Any of the following messages can be generated as part of the D NET,ID command:

```
IST2061I NO FORCED TAKEOVER REQUESTS ARE ACCEPTABLE
IST2062I type FORCED TAKEOVER REQUESTS ARE ACCEPTABLE (where type can be MNPS or SNPS)
IST2063I ALL FORCED TAKEOVER REQUESTS ARE ACCEPTABLE
```

Requirement: Both the forced takeover and owning VTAMs must be z/OS Communications Server V1R6 level or higher.

The mechanics of an MNPS forced takeover are very similar to an MNPS planned takeover. The highlights of the forced takeover processing are:

1. An application opens an ACB with the same application program name as the target application program. The ACB macroinstruction must specify PARMS=(FORCETKO=YES,PERSIST=YES) in order for the OPEN ACB processing to treat this request as one that is permitted to initiate an MNPS forced takeover sequence. The application being taken over must also have indicated that MNPS forced takeovers are permitted, using the SETLOGON macroinstruction.
2. The taking over VTAM obtains the capabilities of the application being taken over from the coupling facility structure. The capabilities of the taking over application program must match those of the application program being taken over for forced takeover to continue. Refer to *z/OS Communications Server: SNA Programming* for details on the actual capabilities that must match. In addition, for forced takeover, the state of the application being taken over must be one in which forced takeover is permitted, and the application being taken over must indicate that it will accept forced takeover requests.
3. The same CDRSC and DLUR processing is performed for forced takeover as is performed for planned takeover.
4. OPEN ACB processing is performed for the application at the taking over VTAM, but no response is posted to the application. VTAM must first obtain ownership of the application from the current owning VTAM before the OPEN ACB can be considered successful.
5. The VTAM attempting to acquire ownership sends a takeover request to the current VTAM (by XCF). This signal will carry a new indicator to say this is a forced takeover request. The target node, if the application status is unchanged or if an earlier forced takeover attempt from a different node has not already been accepted, will initiate CLOSE ACB processing for the application. This is the same persistent CLOSE ACB processing that was required prior to MNPS planned takeover request processing, but now, for forced takeover, the CLOSE ACB is driven internally by VTAM. This persistent CLOSE processing is still required, even for forced takeover, in order to get a stable representation of the application sessions and underlying HPR pipes for takeover purposes.

The application is notified of the persistent CLOSE processing using the TPEND exit. Once the persistent CLOSE ACB completes, the current owning VTAM, just as in planned takeover, updates the multinode persistent session structure to show the taking over VTAM as the application owner, and then sends a positive takeover reply.

Note: Because the CLOSE ACB is now performed at the current owning VTAM as part of forced takeover processing, the user should anticipate a slightly longer recovery time for forced takeover as compared to planned takeover.

Recommendation: An application willing to accept MNPS forced takeover requests must consider carefully the PSTIMER setting chosen (if one is used at all). A persistent CLOSE ACB for the application is driven by VTAM as part of the forced takeover logic and after that has completed successfully, a non-persistent CLOSE ACB is driven. The application PSTIMER value should be set to a value at least long enough to allow the internal VTAM processing to occur between the completion of the persistent CLOSE and the start of the non-persistent CLOSE. An interval of 30 seconds or more is recommended for this situation.

6. When the taking over VTAM receives the positive takeover reply, VTAM will claim the structure storage lists associated with the application and will mark the application as being enabled for persistence. The response to the OPEN ACB is then posted to the application, because this VTAM has now acquired ownership of the application. Takeover processing proceeds along the same lines as planned takeover at this point.
7. After transferring ownership, the previous owning VTAM cleans up its local representations of the sessions and underlying HPR pipes in the exact same manner as it did for planned takeover processing. The local copy of the application is discarded once the sessions and pipes are eliminated.
8. As with planned takeover, once the new owning VTAM has re-created the necessary session control blocks from the information in the coupling facility, the application program must restore the session, using the OPNDST RESTORE command or APPCCMD RESTORE (if using the APPCCMD API), before session data traffic can resume completely.

What to do if recovery does not occur or complete

The D NET,ID command can be used to display the sessions associated with an MNPS application. As discussed earlier, the session that is eligible for MNPS recovery is denoted with a /M status modifier; however, the presence of the /M status modifier does not guarantee that the session will recover if the application were to move or to fail. Factors, such as whether the application is enabled for persistence and whether the session partner is a DLUR-related dependent LU, for example, can affect the actual recovery of the session.

If the display information indicates that a session is not recoverable and if a message is displayed indicating that a session is not recoverable and you expected it to be recoverable, check the following:

- Verify that PERSIST=MULTI is specified on the APPL definition statement. If model application definition statements are being used to define your application program, verify that the application program is active and PERSIST=MULTI is coded on the application model.
- Verify that PARMS=(PERSIST=YES) is specified on the OPEN ACB for the application program.
- Verify that the coupling facility structure name matches the STRMNPS start option for all VTAMs.

- Verify that HPR=RTP start option is coded and not overridden by the definitions for the PU or TG.
- For end nodes, verify that at least one adjacent node has HPR=RTP start option specified.
- For network nodes, verify that the path chosen for the session is one that traverses an adjacent node with HPR=RTP coded (or the configuration option that applies to the adjacent node if not a VTAM).
- Verify that the adjacent CDRM (for migration data host node) does not favor subarea routing over APPN (SSCPDYN start option or SSCPORD and SORDER).

  Note: SSCPORD and SORDER can be specified as start options or specified separately for each ADJSSCP table.

If you have a multinode persistent session-capable application program and the sessions are not being recovered when you expect them to be recovered, check the following:

- Verify that the recovering application program is using the same name as the failed application program.
- Verify that an MVS system symbol is used as part of the value for LUAPFX. If model application definition statements are being used to define your MNPS application program and LUAPFX is specified, use an MVS system symbol as part of the value for LUAPFX to reduce collision of LUALIAS values.
- Verify that the sessions do not involve LUs active where the recovery is taking place or LUs active at the node where the MNPS application was previously active.
- Verify that the PSTIMER has not expired or that it is not too short. Recall that when the PSTIMER expires, the session information is cleaned up. If the timer value is too short, there might not be sufficient time to recover the sessions.
- Verify that the partner node still has sessions showing active. It could be that the MNPS node shows the sessions as PRECOVRY, but the HPR connection has timed out and sessions are cleaned up at the partner node.
- Verify that the sessions are not in PRECOVRY state. If they are, the recovery might be occurring but is slow to complete.
- Verify that the application program being recovered has PERSIST=MULTI specified on its APPL definition statement and PARMS=(PERSIST=YES) on its ACB statement.
- Verify that the new VTAM (where the application is being recovered) has HPR connectivity and (if VTAM is an end node) has CP-CP sessions to an NNS (which might be needed to calculate the new route to the partner endpoint). Use the DISPLAY NET,STATS,TYPE=CFS command to verify that the new structure has connectivity to the multinode persistent session coupling facility structure.
- Verify that if the attempt is a forced takeover, that the takeover and the current owning VTAM are at the correct level, which is at a minimum z/OS Communications Server V1R6. Verify that the ACB macroinstruction of the application issuing the forced takeover request correctly specified PARMS=(FORCETKO=YES). Verify that the application being taken over has correctly indicated the ability to accept forced takeover requests by specifying PARMS=(FORCETKO=ALL) or PARMS=(FORCETKO=MULTI) on SETLOGON OPTCD=PERSIST (and that no subsequent SETLOGON OPTCD=PERSIST specified PARMS=(FORCETKO=NONE) or PARMS=(FORCETKO=SINGLE)). Finally, if the application being taken over was once owned by a pre-z/OS Communications Server V1R6 node, then ensure that SETLOGON OPTCD=PERSIST with PARMS=(FORCETKO=ALL) or

PARMS=(FORCETKO=MULTI) was performed after the application moved to a z/OS Communication Server V1R6 node.

- Verify that a CDRSC representation of the MNPS application could be created, or reused, at the VTAM DLUS node when moving the MNPS application to another VTAM node in the sysplex.

TSO generic resources

When TSO is part of the sysplex environment and it exists on multiple sysplex members, the use of TSO through VTAM can be extended to allow a generic name to be assigned to all TSO/VTAM application programs. A TSO/VTAM on one MVS system can be known by the same generic name as a TSO/VTAM on any other MVS system. All TSO/VTAM application programs sharing a particular generic name can be concurrently active.

Using a generic name allows for increased availability and workload balancing because all TSO/VTAM application programs can be accessed by the same generic name. For example, the MVS workload manager can be used to make efficient use of system resources by selecting a session partner based on balanced load distribution from among TSO/VTAM application programs with the same generic name.

An enhanced TSO reconnect function is also available; this function is dependent on a Job Entry Subsystem (JES) complex. A JES complex can be either a JES2 Multiaccess spool (JES2 MAS) or a JES3 processor complex. Either of these is a collection of host systems that share spool and other system facilities. For example, one piece of shared information is the identity and location of every TSO user address space. In a TSO/GR configuration, JES is asked for the system location of a disconnected TSO user when a reconnect logon request is submitted for the TSO generic name. The scope of visibility for JES is the JES complex. In this way, reconnect works in a predictable way for TSO/GR only when all members of the GR set are contained within one JES complex. This does not mean that you cannot have two, distinct TSO/GR sets with two, distinct JES complexes, but that arrangement is not recommended. Management is easier if you have only one JES complex for a parallel sysplex. Also, the TSO/GR set does not, and cannot, extend beyond the boundaries of a parallel sysplex.

When a generic name is used for TSO, unique names must be given to each of the TCAS, terminal control address space, APPL definition statements. The name *TSO* should not be used as the name on the TCAS APPL definition statement in any of the sysplex members. *TSO* is the TCAS ACBNAME in each sysplex member. Using *TSO* as the name on a TCAS APPL definition statement and as a generic resource member name can result in a local TCAS being selected rather than the generic member, whenever generic name resolution selects the member name *TSO*. The result is an unexpected distribution of TSO sessions in the generic resource environment.

To assign a generic name to a TSO/VTAM application, specify the GNAME parameter. The GNAME parameter can be:

- Coded in the TSOKEYxx parmlib member
- Specified on the START command

Sysplex-wide security associations

TCP/IP uses the coupling facility in the MVS sysplex to provide the ability to recover IPSec security associations for a distributed DVIPA on another node in the sysplex, either when a TCP/IP stack fails or when a planned takeover is initiated. The configuration and use of this function is described in the *z/OS Communications Server: IP Configuration Guide*.

Coupling facility failures for sysplex-wide security associations

There are four types of events that can impact the sysplex wide security associations function:

- Failure of a TCP/IP stack
- Failure of a VTAM node
- Rebuild of the sysplex-wide security associations structure EZBDVIPA
- Disconnect from the EZBDVIPA structure

Failure of a TCP/IP stack

If a TCP/IP stack (which is the distributor for a DRVIPA involved in IPSec connections) fails, ownership of the entries in the EZBDVIPA coupling facility structure associated with that DRVIPA (owned by the failing TCP/IP stack) is taken over. The ownership is taken over by the TCP/IP stack designated as the backup for the DRVIPA, if a backup stack exists. When the failing stack is restarted, ownership of those entries is given back to it.

Failure of a VTAM node

When a VTAM node fails, all TCP/IP stacks using that VTAM node to access the EZBDVIPA structure lose access to the structure, and the VTAM node is disconnected from the EZBDVIPA structure. When VTAM is restarted, the TCP/IP stacks that were using that VTAM node re-register as EZBDVIPA users with this VTAM and VTAM attempts to reconnect to the EZBDVIPA structure. When the reconnect is complete, VTAM notifies the TCP/IP users, and the TCP/IP stacks repopulate the EZBDVIPA structure with updates to the IPSec security associations that have changed while VTAM was down.

Rebuild of the sysplex-wide security associations structure (EZBDVIPA)

Rebuild can occur for several reasons:

- Operator initiated
- Loss of connectivity
- Structure failure

After a rebuild, the EZBDVIPA structure is repopulated using local data from each of the TCP/IP stacks. Each TCP/IP stack is notified that a rebuild has occurred, and the structure must be repopulated. Each TCP/IP stack sends updated information about the security associations for DRVIPAs that it owns to VTAM to be written to the EZBDVIPA structure.

During a rebuild, TCP/IP is not able to access the coupling facility. This creates a problem with obtaining sequence numbers. If TCP/IP cannot access the coupling facility, it cannot get the next available sequence numbers. Therefore, during rebuild, packets might be dropped and normal TCP/IP retransmit protocol will

resend the data. It is assumed that rebuild takes on the order of 10-15 seconds, and the data is sent with the correct sequence number once the rebuild and repopulate have completed, so the connection is not broken. However, there are many factors which can affect the time for rebuild to complete, such as system load and number of structures being rebuilt. It is possible that connections might be dropped during such a rebuild. The situation is not likely, and it is rare that such a massive rebuild is required.

Disconnect from the EZBDVIPA structure

The following are cases in which VTAM might disconnect from the structure:

- A VARY NET,CFS,ACTION=DISCONNECT operator command.
- A rebuild fails and the original structure is not accessible.
- An internal error within VTAM occurs.

Disconnect is viewed as a much longer loss of coupling facility access than rebuild. It is more likely that connections will be dropped during the first case. However, VTAM has automatic reconnect logic for cases 2 and 3 above, which might occur before the connections timeout.

When VTAM reconnects, it notifies each TCP/IP stack using the EZBDVIPA structure that a reconnect has occurred, and the structure must be repopulated. Each TCP/IP stack sends updated information about the security associations for DRVIPAs that it owns to VTAM to be written to the EZBDVIPA structure.

Sysplexports

TCP/IP uses the coupling facility in the MVS sysplex to provide coordinated sharing of ephemeral port assignments among TCP/IP stacks in the sysplex, for Distributed DVIPAs. The configuration and use of this function is described in the *z/OS Communications Server: IP Configuration Guide*.

Coupling facility failures for Sysplexports

There are four types of events that can impact the Sysplexports function:

- Failure of a TCP/IP stack
- Failure of a VTAM node
- Rebuild of the Sysplexports structure EZBEPORT
- Disconnect from the EZBEPORT structure

Failure of a TCP/IP stack

If a TCP/IP stack fails, and that stack is a target stack for a DRVIPA using the Sysplexports function, the distributor stack for that DRVIPA returns all ephemeral ports allocated to the failing stack back to the available pool in the EZBEPORT structure. If the distributor stack for a DRVIPA fails, and has no backup stack designated, the target stacks for that DRVIPA return all ephemeral ports assigned to each target stack back to the available pool. The ephemeral ports assigned to a failing distributor stack (with no backup) are recovered when the distributor stack is restarted and requests an ephemeral port for the first time.

Failure of a VTAM node

When a VTAM node fails, all TCP/IP stacks using that VTAM node to access the EZBEPORT structure lose access to the structure, and the VTAM node is disconnected from the EZBEPORT structure. When VTAM is restarted, the TCP/IP

stacks that were using that VTAM node re-register as EZBEPORT users with this VTAM, and VTAM attempts to reconnect to the EZBEPORT structure. When the reconnect is complete, VTAM notifies the TCP/IP users, and the TCP/IP stacks repopulate the EZBEPORT structure with the list of ephemeral ports each TCP/IP stack has currently allocated.

Rebuild of the Sysplexports structure (EZBEPORT)

Rebuild can occur for several reasons:

- Operator initiated
- Loss of connectivity
- Structure failure

After a rebuild, the EZBEPORT structure is repopulated using local data from each of the TCP/IP stacks. Each TCP/IP stack is notified that a rebuild has occurred, and the structure must be repopulated. Each TCP/IP stack sends updated information about the ephemeral ports VTAM allocated to it to VTAM to be written to the EZBEPORT structure. During a rebuild, TCP/IP is not able to access the coupling facility. This creates a problem with obtaining ephemeral port numbers for new connections. Therefore, requests for connections involving new ephemeral port numbers fail during a rebuild. It is assumed that rebuild takes on the order of 10-15 seconds, and the requests for new ephemeral port numbers are satisfied once the rebuild and repopulate has completed. However, there are many factors which can affect the time for rebuild to complete, such as system load and number of structures being rebuilt. This situation is not likely, and it is rare that such a massive rebuild is required.

Disconnect from the EZBEPORT structure

The following are cases in which VTAM might disconnect from the structure :

- A VARY NET,CFS,ACTION=DISCONNECT operator command.
- A rebuild fails and the original structure is not accessible.
- An internal error within VTAM occurs.

Disconnect is viewed as a much longer loss of coupling facility access than rebuild. When VTAM reconnects, it notifies each TCP/IP stack using the EZBEPORT structure that a reconnect has occurred, and the structure must be repopulated. Each TCP/IP stack sends updated information about the ephemeral ports VTAM has allocated to it to VTAM to be written to the EZBEPORT structure.

Chapter 17. Implementing an APPN network

An APPN network contains network nodes and end nodes connected by APPN connections. VTAM can be coded as either a network node or an end node. Configuring VTAM for APPN is optional. To configure VTAM as an APPN node, select the characteristics of the node and define the resources of that node to VTAM.

The HOSTSA, NODETYPE, and SACONNS start options are used to determine what type of node VTAM will be. Table 47 shows how these start options are defined for the different types of nodes.

Table 47. Start options and node type relationship

Node type	NODETYPE	HOSTSA	SACONNS
Subarea node	not coded	*subarea number*	YES
Interchange node	NN	*subarea number*	YES
Migration data host	EN	*subarea number*	YES
Network node	NN	*subarea number**	NO
End node	EN	*subarea number**	NO
*HOSTSA is optional. If not specified, VTAM will start as a EN or NN. If HOSTSA is specified, then SACONNS=NO must be specified to start VTAM as an EN or NN.			

By coding the NODETYPE start option, you specify a node as an APPN node. After NODETYPE is coded, the following start options default to the values needed to enable APPN connections.

CPCP The CPCP start option defaults to LEASED and enables CP-CP control sessions across APPN leased links. The other possible values for CPCP are SWITCHED, YES (for all links), and NO (for no links). For information on modifying CP-CP session enablement, see "Enabling control sessions" on page 489.

CONNTYPE
The CONNTYPE start option defaults to the value APPN and enables full APPN function across a link. The other connection type value is LEN. There is also a CONNTYPE operand on the PU definition statement that can be used to override the value of the start option for specific links. For more information on the CONNTYPE start option and operand, see Chapter 18, "Implementing a combined APPN and subarea network," on page 505.

Coding considerations for APPN resources

LU definitions are not needed if you use dynamic independent LUs. See "Defining independent LUs" on page 275.

Coding examples for each of the connection types are included in this section following the general requirements. See "Channel connections between APPN nodes" on page 51, "Leased connections between APPN nodes" on page 59, "IBM

© Copyright IBM Corp. 1984, 2011

3172 Nways Interconnect Controller connections between APPN nodes" on page 60, and "IBM Open Systems Adapter connections between APPN nodes" on page 68.

If you take advantage of the CPCP and CONNTYPE start option defaults, you only need to consider the following requirements for an APPN connection:

- To enable CP-CP sessions over a switched APPN connection, the CPCP start option should be defined as YES or SWITCHED.
- The PU definition for the NCP in a composite network node or for the VTAM in a network node or end node needs to indicate exchange identification support (XID=YES). (For a switched connection, XID cannot be coded but is automatically set to YES.)
- Use different names for the CP name and the PU name of a node.
- If you choose to allow dynamic definition of your independent logical units by defining the DYNLU start option or operand as YES, you only need to define the independent LUs on their owning CP.
- If you choose not to allow dynamic definition of your independent logical units, each network node that needs to connect to the independent logical unit needs a CDRSC definition (or a model CDRSC definition) for the resource. If you chose to code the ALSLIST operand on the CDRSC statement, the ALSLIST operand should specify either the name of an adjacent link station that can be used to connect to the LU or to ISTAPNPU. ISTAPNPU is a generic representation for an APPN adjacent link station. If ISTAPNPU is included in the ALSLIST, VTAM uses any available APPN link station when calculating a route to the LU.

  If you chose not to code ALSLIST, VTAM searches for the LU in the APPN network and, if found, adds ISTAPNPU to the ALSLIST dynamically.

  Also, unlike a LEN adjacent link station, an APPN adjacent link station is not required to have a NETID that matches the NETID of the independent LU at session establishment.
- To establish a statically defined switched APPN connection, either the CPNAME operand or the IDBLK and IDNUM operands must be defined for each switched PU definition. If the CPNAME operand is defined, its value must be the same as the CP name of the node (the SSCPNAME start option for VTAM nodes) that owns the PU. If the values are not the same, VTAM cannot locate the PU and LU definitions. There may be more than one PU related to that same CP.

  To establish a statically defined switched LEN connection, either the CPNAME operand or the IDBLK and IDNUM operands must be defined for each switched PU definition. If the CPNAME operand is defined, its value specifically identifies a specific PU and cannot be duplicated.
- To establish a dynamically defined switched connection, you can allow the DYNADJCP start option to default to YES and code the DYNPU operand on the GROUP definition statement in the XCA or NCP major node.
- You need to know the maximum number of sessions that will be established through each NCP in a composite network node. For NCP type 2.1 connections, this number is defined during NCP generation and cannot dynamically expand. However, for a direct VTAM connection, the session control blocks are held in the BSBUF buffer pool that can expand and contract as necessary.
- For an NCP connection, you should also know the maximum number of sessions that any particular type 2.1 node can establish per link station. If you do not specify this limit, a single independent LU in a type 2.1 node can exhaust all of the session control blocks generated in the NCP. For information on specifying this limit, see "Limiting sessions" on page 282.

There is no corresponding control for type 2.1 nodes directly attached to a VTAM. However, because this is a buffer pool, it can expand as needed to accommodate more sessions.

- When an independent LU in a type 2.1 node acts as the primary logical unit, NCP must assign another network address to the PLU. This address pool is generated in NCP. Because this is a predefined pool, you should know the maximum number of PLU sessions that all type 2.1 node independent LUs attached to that NCP can establish concurrently. For a direct VTAM connection, the addresses are assigned from the element pool that is managed by VTAM.
- Depending upon the specific type of APPN connection and the type of node at each endpoint, High-Performance Routing (HPR) can be used. If desired, HPR support can be disabled on a node-wide basis using the HPR=NONE start option, or limited using the HPR=ANR start option. Within a node providing HPR support, HPR can be disabled on individual links by using the HPR operand on the GROUP, LINE, PU or CDRM definition statements. See "Limiting HPR support" on page 495 for additional information on the effects of the HPR start option. For further information on HPR, see "High-Performance Routing (HPR)" on page 491.

Maximum APPN Locate size considerations

Although VTAM supports sending and receiving APPN Locate searches with a size up to a maximum of 16 KB, other products that act as APPN network nodes or end nodes might support much smaller maximum APPN Locate sizes (as small as 1 KB). This can cause problems for ENs that have many links (described by TG Vectors) to other nodes in the network, since these TG Vectors often must be included on the APPN Locate search in order to compute the best session path.

If any such APPN products are present along the path of an APPN Locate search, and the APPN Locate search that is to be sent exceeds the maximum Locate size allowed by these products, then some of these TG Vectors will be removed from the APPN Locate search in order to prevent the maximum Locate size from being exceeded. (VTAM typically copies the TG Vectors for active links to the APPN Locate search in the order these TGs were activated, which means the most recently activated TGs are removed first.) When this occurs, the network node server responsible for computing the session path might not choose the best possible path or the route computation could fail entirely (if the TG Vector describing the best or only path was removed). To avoid this problem, whenever possible, minimize the number of links (TGs) that end nodes have to other nodes. Using a connection network to reduce the number of EN TG Vectors can also be helpful.

Enabling control sessions

The CPCP start option specifies whether an APPN node supports CP-CP sessions with an adjacent node. The CPCP start option is valid only for nodes at which the NODETYPE start option is sqecified. APPN nodes use CP-CP sessions to communicate topology, routing, directory, and session information to adjacent control points. If you use the default, CP-CP sessions are supported on all APPN leased links. Because CP-CP sessions carry network control information, they should be active as long as the node is active. Using the default allows you to control whether your APPN switched links support CP-CP sessions on a link-by-link basis by defining the CPCP operand.

You can use the CPCP start option to specify support for CP-CP sessions on leased links only, on switched links only, on both leased and switched, or on neither leased nor switched.

The CPCP operand can be coded on the PU definition statement for the local SNA, model, NCP, and switched major nodes.

CP-CP sessions between two VTAM nodes

CP-CP sessions between two VTAM nodes require both CDRSC and ADJCP definitions at each VTAM to represent the partner CP. The CDRSC definition represents the partner LU, and the ADJCP definition represents the adjacent CP at the connection level. ADJCP definitions can be created dynamically or predefined. When needed, CDRSC definitions are dynamically created for adjacent CPs, regardless of the value coded for DYNLU.

Defining adjacent APPN nodes

The DYNADJCP start option controls whether adjacent CP minor nodes can be created dynamically. An adjacent CP major node, ISTADJCP, is automatically created when a VTAM node is activated. An adjacent CP minor node describes the CP name, the network identifier, node type (end node or network node), and whether dynamic definition is allowed for LUs owned by this adjacent node.

When the NODETYPE start option is specified, the default for the DYNADJCP start option is DYNADJCP=YES. If you allow the DYNADJCP start option to default to YES, adjacent CP minor nodes are created as needed to provide control and management of connections to adjacent APPN nodes.

Note: Unless CDRSCs or appropriate model CDRSCs are predefined for adjacent CPs, CDRDYN=YES is also required for the dynamic creation of adjacent CP minor nodes.

If you define the DYNADJCP start option as NO, you need to define every potential adjacent CP within adjacent CP major and minor nodes. Connections are established with only those nodes you specify. If you do not allow dynamic ADJCPs, and you do not define any adjacent CPs with the ADJCP definition statement, all connection attempts will fail.

You can choose to allow dynamic ADJCP minor nodes to be created for some links and not allow them for others. Be aware that for each link where you restrict dynamic definition, you need to define each CP to which you will allow connections with an ADJCP definition statement.

If you want dynamic definitions on most links, you can define the DYNADJCP start option as YES, and override it by defining the DYNADJCP operand as NO for the links on which you want to restrict dynamic definitions.

If you want limited dynamic definitions, you can code the DYNADJCP start option as NO and override it by coding the DYNADJCP operand as YES for links on which you want dynamic definitions.

Defining the logon mode for CP-CP sessions

CP-CP sessions always use the CPSVCMG logon mode. This logon mode is defined in the IBM-supplied default logon mode table, ISTINCLM. Logon mode CPSVCMG must be included in ISTINCLM in order to establish CP-CP sessions.

If no CPSVCMG logon mode is defined, CP-CP session activation will fail, and VTAM will issue the following message:

```
IST264I REQUIRED LOGMODE NAME CPSVCMG UNDEFINED.
```

In addition, message IST663I will display the error sense code X'08210002' (not valid mode name at CP).

High-Performance Routing (HPR)

Migrating NCP connections to APPN connections results in the creation of additional NCP control blocks, which can require additional storage space and cycles in the NCP. In addition, when a node or link along a route fails, sessions using that route fail.

High-Performance Routing (HPR) allows you to migrate NCP connections to APPN connections without incurring the associated increase in storage and cycles. HPR uses a rapid transport protocol (RTP) connection to transport session traffic between session endpoints. HPR routes can also traverse an existing subarea network, as HPR support provides for the mapping of HPR routes over VR-based TGs.

In addition, if a node or link on an HPR route fails and an alternate HPR route exists between HPR session endpoints, HPR automatically switches routes and sessions are not disrupted. A method is also available to allow the operator to force an HPR path switch.

To use HPR over an NCP you must have, at minimum, NCP Version 7 Release 3 (and 37xx controllers supported by NCP Version 7 Release 3). To use HPR over a subnetwork boundary where one or both of the border nodes are composite network nodes, you must have, at minimum, NCP Version 7 Release 5.

What is High-Performance Routing?

High-Performance Routing is a set of enhancements for APPN that meets the following requirements:

- **Improved APPN data routing:** HPR transports data at very high speeds by using low-level intermediate routing and by minimizing the number of flows over the links for error recovery and flow control protocols. The flows are minimized by performing these functions at the endpoints rather than at each hop (link) along the path.
- **Improved APPN reliability:** HPR switches paths within the HPR portion of the network to bypass link and node failures if an acceptable alternate path is available. This occurs transparently to the sessions; in other words, the session is not disrupted.
- **Functional equivalence:** HPR maintains functional equivalence with APPN. To do so, HPR continues to support priority routing, connection networks and multiple network connectivity. Priority routing allows the capability for higher priority traffic to pass lower-priority traffic in intermediate nodes within the HPR portions of the network. HPR also routes across connection networks or subnetwork boundaries in much the same way as APPN.

  HPR routes are not given preferential treatment by the APPN routing algorithm. Existing non-HPR APPN routes will also be used if they meet the requirements of the APPN Class of Service.

- **Seamless migration:** HPR is designed for *drop-in* migration. A given APPN node can be upgraded to the HPR level of function without taking down the network, without configuring new parameters at its adjacent nodes, and without any logistical complications or coordination.

What is Rapid Transport Protocol?

Rapid Transport Protocol (RTP) is a transport layer protocol that provides the following:

- Flow control.
- End-to-end error recovery.
- Selective retransmission of lost packets.
- Nondisruptive rerouting of sessions.
- Segmenting and reassembly of packets.
- Translation of APPN to HPR protocols (boundary function) for sessions that have both HPR and non-HPR portions.
- Translation of subarea to HPR protocols for sessions that have subarea and HPR portions. This translation is limited to routes that have a:
  - Non-HPR VR-based TG adjacent to an HPR route.
  - Subarea portion adjacent to a one-hop HPR route where the HPR endpoints are an interchange node and an adjacent APPN node. (Some restrictions apply; for details, see "Sessions traversing subarea and APPN networks" on page 501.)

RTP uses an adaptive rate-based (ARB) congestion control algorithm that makes efficient use of network resources by providing a congestion avoidance and control mechanism. The basic approach used in this algorithm is to regulate the input traffic during changing network conditions. When the algorithm detects that the network is approaching congestion, it reduces the input traffic rate until these indications go away. The algorithm also allows more traffic to enter the network without exceeding the rate the receiver can handle.

At LU-LU session establishment, if a node providing the RTP function determines that at least some portion of the session route traverses one or more HPR-capable links and eventually terminates in another RTP node, it will establish an *RTP connection* between the two RTP nodes. Subsequent sessions can reuse existing RTP connections if the HPR portion of the route, the APPN Class of Service, and the network connection endpoints (NCEs) are the same. CP-CP sessions over APPN node-to-node connections and ATM native connections also use RTP connections.

Tip: You can use the HPRSESLM start option or the MODIFY VTAMOPTS,HPRSESLM= command (if HPRSESLM is not currently set to DISABLED) to limit the number of LU-LU sessions assigned to each RTP connection. Limiting the number of sessions on a single RTP connection can result in performance improvement by allowing concurrent traffic on multiple RTP connections.

When a VTAM establishes a logical connection between itself and another RTP node, VTAM creates a dynamic PU to represent the connection. These PUs are easily identified since their names are always in the form CNRxxxxx (unless the DYNHPPFX start option has been used to change this default prefix).

Note: A model PU definition can be created to customize the characteristics of dynamically created PUs. Use the DYNTYPE=RTP operand for the model PU definition in the model major node.

RTP transmits data in a form known as a network layer packet (NLP). The NLP begins with a header that specifies the automatic network routing (ANR) information necessary for routing the packet through the network.

Note: The maximum data transmission size referenced by VTAM to set the maximum size of data transmitted is calculated from the maximum packet size returned by the Route_Setup signal exchanged during RTP initialization.

To prevent excessive data segmenting within VTAM's data transmission processing, carefully evaluate the TCP/IP MTU size (for Enterprise Extender) and the maximum received data size for VTAM and each node in the network.

For Enterprise Extender, VTAMs maximum BTU length the sender can receive is calculated as follows:

```
Max_Receive_Size=
MAX((MIN(TCP/IP MTU SIZE -3),32767),768)
```

Because excessive segmenting can occur in the effort to conform to maximum data size requirements, you should optimize the maximum received data size. Failure to do so can result in the following:

- Inefficient TIPAC (I/O) buffer allocation
- Excessive TIPAC buffer pool expand operations
- Degraded data transmission performance

What is automatic network routing?

Automatic network routing is a low-level routing mechanism that minimizes cycles and storage requirements for routing packets through intermediate nodes. Unlike APPN, intermediate ANR nodes are not aware of SNA sessions or RTP connections passing through the node. All an ANR node must do is read the header in a network layer packet and forward the information to the next node on the path. The ANR information is learned by RTP during establishment of the RTP connection by sending a “Route Setup” message which flows through all nodes on the prospective HPR path.

Tip: On nodes configured as border nodes (BN=YES is specified) with HPR fully enabled (HPR=RTP is specified), code RTPONLY=YES on ADJCP definition statements in adjacent CP major nodes if you want the border node to maintain awareness of all sessions established to, from or through the adjacent node being defined. The RTPONLY operand is valid only when the activating node is configured as a border node, and RTPONLY=YES is meaningful only when the adjacent node being defined is a nonnative border node or nonnative network node. By coding RTPONLY=YES, you are instructing VTAM to disallow use of the ANR function for any new RTPs that are established or path switched to, from, or through the adjacent nonnative node being defined. (Allowing the use of ANR could result in RTPs being established through this border node, thereby preventing the border node from maintaining awareness of any sessions that use these RTPs.) Instead of allowing RTPs to be established through this border node, coding RTPONLY=YES will result in VTAM forcing these RTPs to terminate on this border node or forcing the use of intermediate session routing (ISR) instead of HPR (or a combination of the two).

Restrictions:

1. Use of RTPONLY=YES at any APPN subnetwork boundary on a session setup path prevents the use of global VRNs (GVRNs) for intersubnetwork connectivity, since using GVRNs could result in sessions being established across this subnetwork boundary without this border node maintaining awareness of these sessions.
2. Use of RTPONLY=YES can result in an increase in network traffic in the form of additional Route_Setup flows used for RTP establishment. These additional Route_Setup flows will occur only during the establishment of sessions that cross subnetwork boundaries defined with RTPONLY=YES.
3. Use of RTPONLY=YES can result in an increase in storage and CPU utilization, due to VTAM maintaining awareness of these sessions and performing ISR instead of HPR/ANR for these sessions.

How does HPR switch paths?

When a link or node outage occurs in the path of the RTP connection, the RTP endpoint attempts to find an alternate route to maintain the connection. If an acceptable route is not immediately found, RTP will continue to try to find one until the path switch timer expires. (The path switch timer duration is modifiable. Refer to the HPRPST start option in *z/OS Communications Server: SNA Resource Definition Reference* for more details on the path switch timer.) While the path switch attempt is in progress, data transmission is suspended (due to the outage in the original path) until the path switch completes.

Additionally, a path switch attempt can be forced by the operator by invoking the MODIFY RTP command. This will force the specified RTP to make one attempt at finding a better route (as specified by the requirements of the Class of Service) for the RTP connection. If no better route is available, the original route for the RTP connection will be unchanged.

The PSRETRY start option can be used to periodically force VTAM to attempt to find a better route for existing HPR connections. Like MODIFY RTP processing, if a better route is not available, the original route for the connection is unchanged. For both timer-driven and operator-initiated path switch processing, a path switch can be performed to what appears to be an identical path if the path uses a VR-based transmission group. The reason is that VTAM will perform virtual route selection whenever the first hop in the path is a VR-based transmission group, and this underlying virtual route might change from one usage of the VR-based transmission group to the next. VTAM will not verify that a different underlying virtual route is selected, but instead will just perform the path switch in case a change occurred. For more information, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

The HPRPSMSG start option can be useful for large installations that have hundreds or thousands of RTP endpoints on z/OS Communications Server. This start option controls the HPR path switch message reduction function. In the event of a large network outage, this function can prevent VTAM from flooding the system console log with HPR path switch messages. Although this function reduces the amount of path switch information issued to the console log, the summary information attempts to supply the operator with enough information to understand the size and scope of the outage. For more information about the HPRPSMSG start option, see *z/OS Communications Server: SNA Resource Definition Reference*.

HPR implementation overview

The default level of HPR support available is dependent upon:
- Node type (NODETYPE and HOSTSA start options)
- Connection type

Levels of HPR support

There are three levels of HPR support for a VTAM node:

Rapid Transport Protocol (RTP)
An RTP connection can exist between two VTAM nodes with APPN capability (NODETYPE is coded); these two VTAMs can be endpoints of an HPR route.

Automatic network routing (ANR)
A VTAM network node (NODETYPE=NN) can also provide ANR-level support as an intermediate node on an HPR route.

No HPR support
HPR support can be disabled for a particular VTAM network node or end node using the HPR=NONE start option. If HPR support is disabled, that VTAM cannot be an endpoint or intermediate node for an HPR route. A VTAM subarea node (NODETYPE not coded) also cannot provide HPR support.

Limiting HPR support

HPR support provided by a VTAM APPN node can be limited using the following start options:

HPR=ANR
You can use the HPR=ANR start option to limit HPR support provided by a network node. Ordinarily, network nodes can provide both RTP-level and ANR-level support, but use of HPR=ANR precludes a network node from providing RTP-level support. If you use HPR=ANR, coding HPR=YES on a CDRM or PU definition statement (or allowing it to default to YES) results in ANR-level HPR support over that connection.

HPR=(RTP,ANR)
Using HPR=(RTP,ANR) indicates this VTAM provides RTP-level HPR support only if HPR=YES is coded for a particular CDRM or PU definition statement. If you let the HPR operand default on a CDRM or PU definition statement, you only get ANR-level HPR support over that connection. This way you can easily select exactly which links serve as RTP endpoints, while allowing others to provide ANR-level HPR support by default.

Note: The HPR=(RTP,ANR) start option is recommended only for migration purposes. It can cause unexpected results in HPR route selection. The ANR part of this option means that the nodes connected to a TG can act only as intermediate nodes for an HPR connection if HPR=YES is not coded on the PU definition statement for that connection or the PU is not activated with HPR=YES on the VARY ACT command. They cannot act as endpoints for the connection.

If HPR=(RTP,ANR) is used, it is important to ensure that a TG is defined with the same capabilities on both sides of the link. To use the TG with ANR capabilities for the origin and destination nodes of that TG, ANR must be defaulted for the PUs defining the TG on both the origin and destination nodes, because there is no other way

to specify ANR for a PU. To use a TG with RTP capabilities for the origin and destination nodes of the TG, with HPR=(RTP,ANR), HPR=YES must be specified on the PU statement that represents the link station, or the PU must be activated with HPR=YES on the VARY ACT command.

In the following example, TG15 and TG17 cannot be the first hop or the last hop of an HPR connection, but TG16 can. The combination of definitions for TG15, with HPR=YES on one side and ANR defaulted on the other, is not recommended.

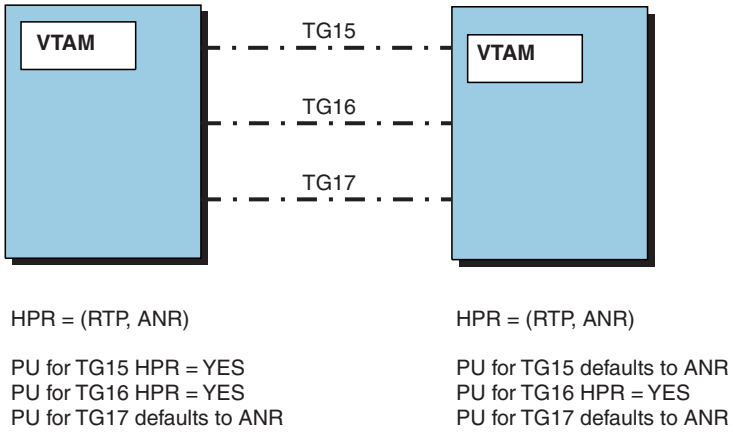


Figure 116. HPR=(RTP,ANR) and TG capabilities

Matching HPR characteristics is especially critical when defining TGs across a border node connection (ISL) and when defining connections to an end node (endpoint TGs). With the current APPN architecture, the HPR characteristics are known in only one direction, so the node doing HPR route calculation cannot ensure that the nodes on both sides of the TG have the necessary HPR capabilities. In these cases, the HPR capabilities must be the same on both sides of the TG.

In summary, unless you adhere to the guidelines above, the establishment of HPR connections will not work correctly across border node TGs (ISLs) or TGs to an end node (endpoint TGs). In addition, TGs used for enterprise extender or ATM connections must be RTP capable on both sides.

HPR=(RTP,NONE)
Specifying HPR=(RTP,NONE) indicates this VTAM provides RTP-level HPR support only if HPR=YES is coded on the CDRM or PU definition statement. If you let the HPR operand default on a CDRM or PU definition statement, you get no HPR support over that connection.

HPR=(ANR,NONE)
Specifying HPR=(ANR,NONE) indicates this VTAM provides ANR-level HPR support only if HPR=YES is coded on the CDRM or PU definition statement. If you let the HPR operand default on a CDRM or PU definition statement, you get no HPR support over that connection. This way you can get ANR support on selected links without automatically getting it on all links.

HPR support can also be disabled by coding HPR=NO on a particular GROUP, LINE, or PU definition statement, or on the CDRM definition statement for VR-based TGs.

Sample HPR configurations

HPR over composite network nodes: In Figure 117, HOSTB and HOSTC are composite network nodes connected by a VR-based TG (TG255). In this configuration, an LU-LU session between HOSTA and HOSTD can use an RTP connection that traverses the composite network nodes. HOSTB and HOSTC have no awareness of the session between HOSTA and HOSTD, as they are providing ANR-level support only. However, for sessions where HOSTB or HOSTC constitute one end of a session, HOSTB and HOSTC can provide RTP-level support (see "Sessions through NCP in a composite network node providing RTP support" on page 503 for further details). HOSTA, as a network node, can also provide ANR-level support for sessions between HOSTB, HOSTC, or HOSTD and some other unpictured node. HOSTD, as an end node, cannot provide ANR-level support.

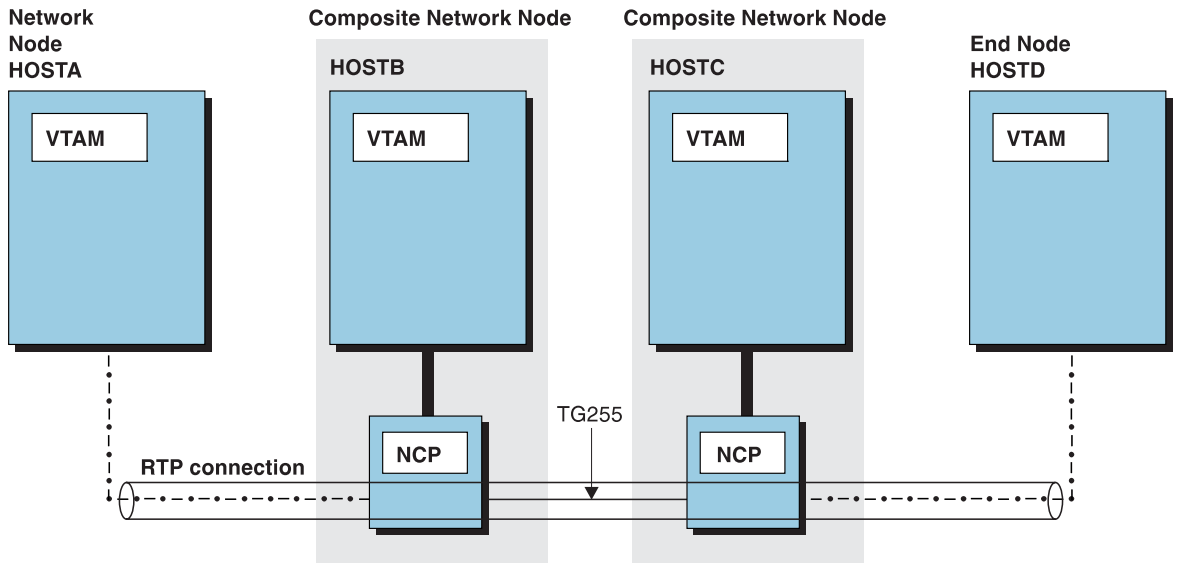


Figure 117. HPR over composite network nodes

HPR over APPN host-to-host channel connections: In Figure 118 on page 498, an LU-LU session between HOSTA and HOSTC can use an RTP connection that traverses the APPN host-to-host channel connections between HOSTA and HOSTB and between HOSTB and HOSTC. For sessions between HOSTA and HOSTC, HOSTB provides only ANR-level support. However, for sessions between HOSTB and HOSTC, HOSTB can provide RTP-level support as the endpoint of an HPR route. Similarly, though not pictured, HOSTB can provide RTP-level support for sessions between HOSTB and HOSTA.

Note: Connections between two hosts that both provide RTP-level support can use HPDT MPC. See "Multipath channel connections" on page 51 for information about the different levels of MPC provided by VTAM.

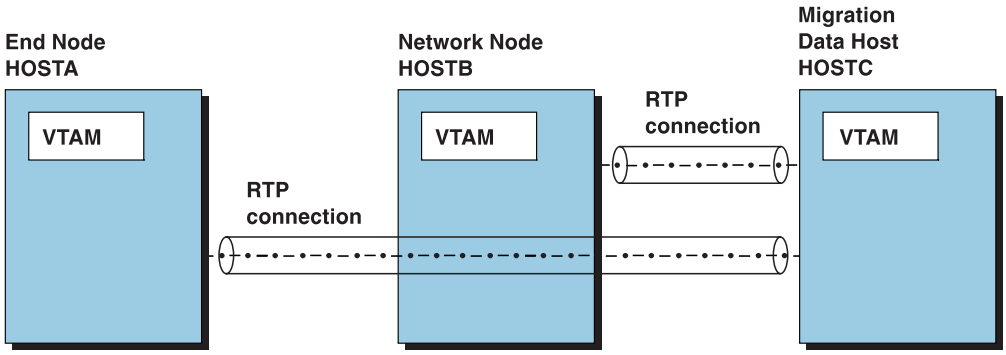


Figure 118. HPR Over APPN host-to-host channel connections

Multiple HPR routes between HPR session endpoints: In Figure 119 on page 499, like Figure 118, none of the hosts have NCPs and all have APPN capability. In this configuration, an LU-LU session between HOSTA and HOSTD can use an RTP connection that traverses the APPN host-to-host channel connections between HOSTA and HOSTB and between HOSTB and HOSTD. For sessions between HOSTA and HOSTD, HOSTB provides only ANR-level support. However, for sessions between HOSTB and HOSTA or HOSTB and HOSTD, HOSTB can provide RTP-level support as the endpoint of an HPR route.

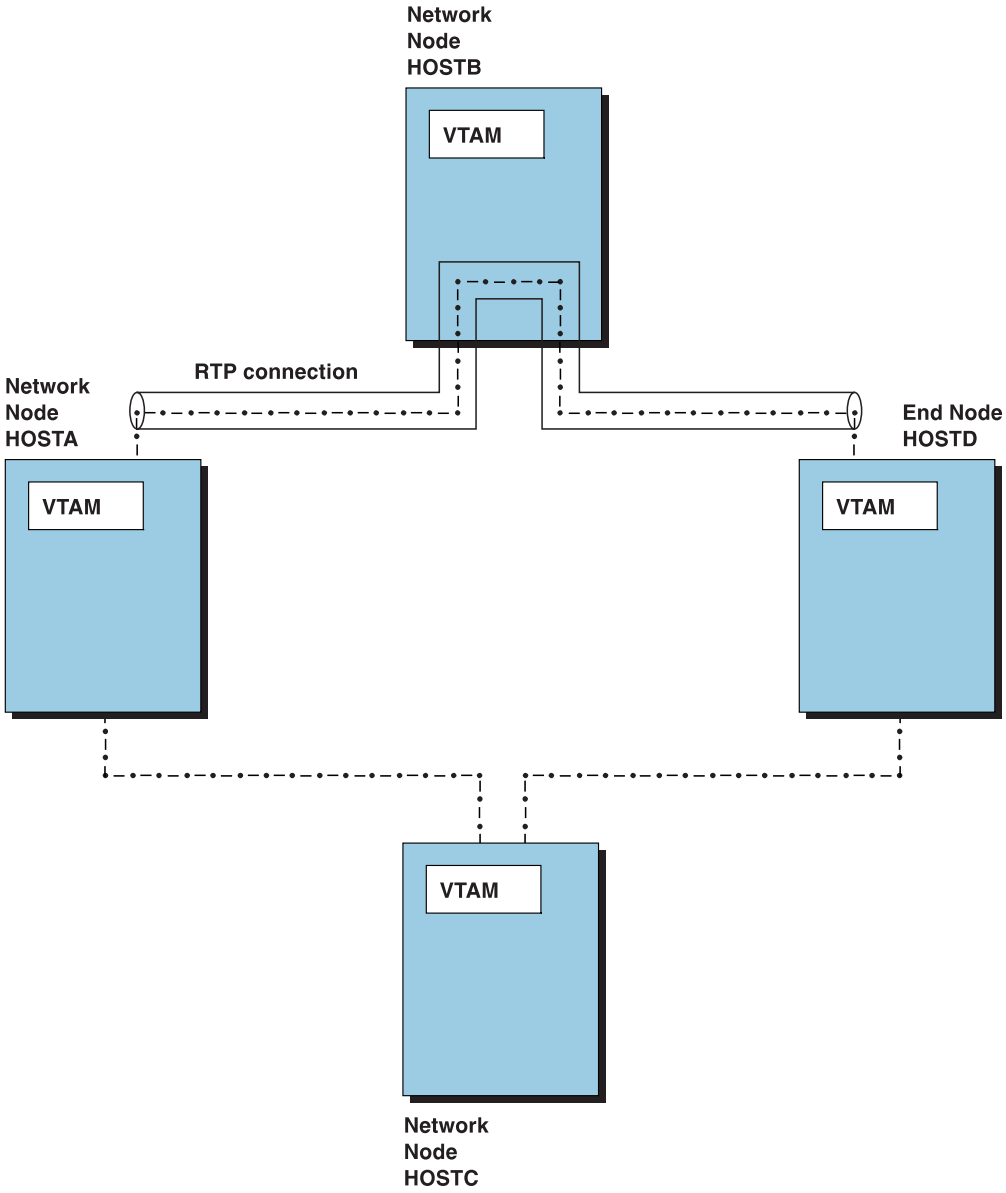


Figure 119. Multiple HPR routes between HPR session endpoints

Also in Figure 119, note that an alternate HPR route exists between HOSTA and HOSTD through HOSTC. When multiple paths exist between HPR session endpoints, HPR support provides for automatic, nondisruptive path switching when a node or link fails on the path in use. For example, if the APPN host-to-host channel connection between HOSTA and HOSTB fails, HPR support provides for the movement of the RTP connection to the alternate path between HOSTA and HOSTD through HOSTC. LU-LU sessions continue uninterrupted.

Multiple RTP connections over the same HPR route: Multiple RTP connections can exist over the same HPR route. For example, in Figure 119, multiple RTP connections can exist between HOSTA and HOSTD through HOSTB. Furthermore, it is possible that more than one of these connections are for sessions desiring the same Class of Service. In this case, to determine which connection a session is traversing, use the DISPLAY SESSIONS command at one of the RTP endpoints. First use DISPLAY SESSIONS to find the session identifier (SID) for the session in question. Then issue DISPLAY SESSIONS and specify the SID on the SID operand.

The resulting display shows the RTP PU (CNRnnnnn, unless the DYNHPPFX start option has been used to change the default prefix) as the ALSNAME, if the session is over an HPR route.

Using VR-based TGs for non-HPR endpoints: As discussed previously, HPR allows you to migrate NCP connections to APPN connections without incurring the associated increase in storage and cycles. However, this is not always possible. In Figure 120, for instance, for a session between the terminal and HOSTB (an interchange node), HPR cannot be used (see "Sessions through NCP in a composite network node providing RTP support" on page 503 for additional information). If APPN storage demands are unacceptable in such a situation, add a VR-based TG to connect the terminal and HOSTB through the subarea network, which allows them to act as APPN peers. Meanwhile, for sessions between HPR-capable endpoints, HPR routing can continue over the existing APPN connection.

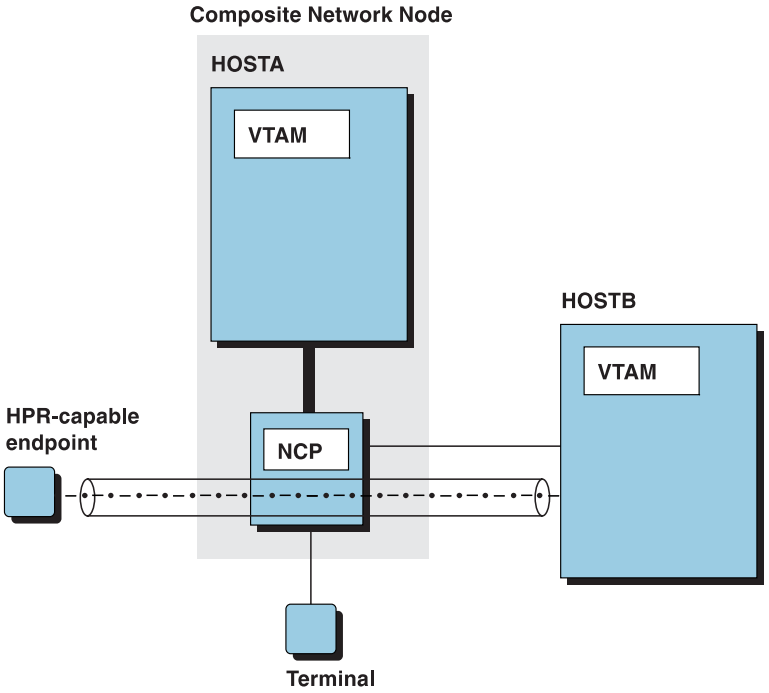


Figure 120. Using VR-based TGs for non-HPR endpoints

Sessions involving HPR and APPN routes: Some sessions might have routes consisting of both APPN segments and RTP segments. In Figure 121 on page 501, for example, the VTAM on HOSTC might not have been defined to provide HPR support. In this case, for a session between HOSTB and HOSTE, HPR is used where possible (between HOSTB and HOSTA and between HOSTD and HOSTE). Otherwise, APPN routing is used (between HOSTA and HOSTC and between HOSTC and HOSTD).

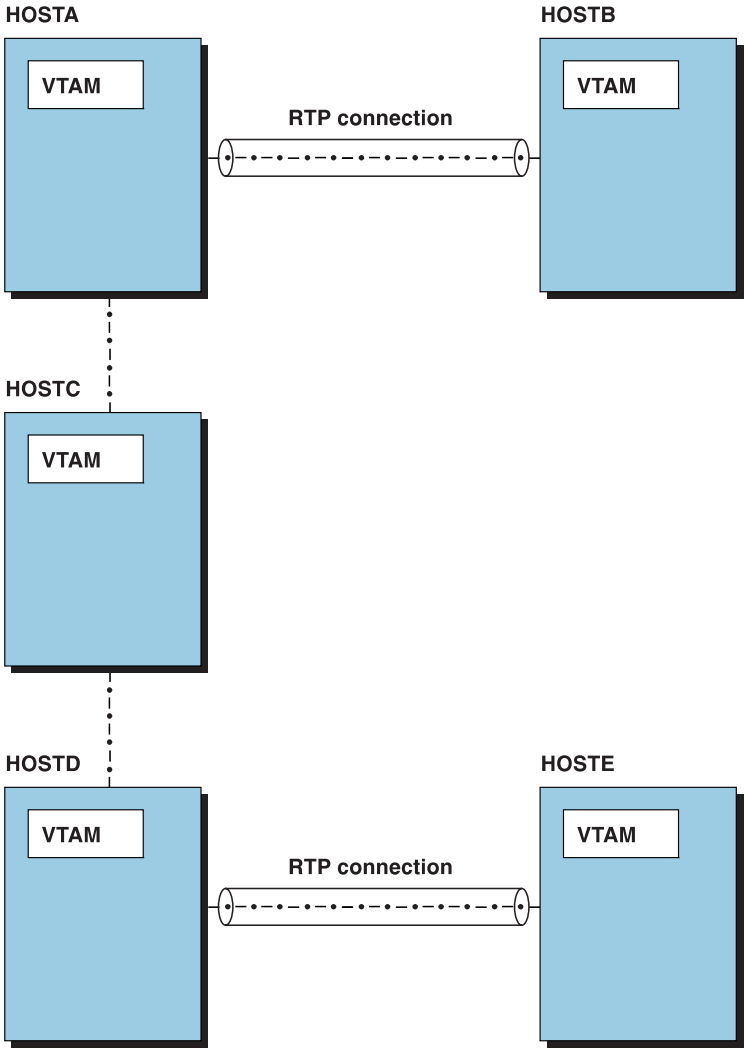


Figure 121. Session involving HPR and APPN routes

Sessions traversing subarea and APPN networks: When one end of a session is in a subarea network and the other in an APPN network, HPR routing can be used at the interchange node in the session path, but only under certain conditions.

APPN session endpoint is one hop from the interchange node: If the session is from a subarea node through an interchange node to an end node that is one hop away from the interchange node, translation of subarea to HPR can occur. In Figure 122 on page 502, for example, a session can exist between LUA and LUB, and HOSTB will translate subarea data received from HOSTA to HPR and route it across the RTP connection to HOSTC.

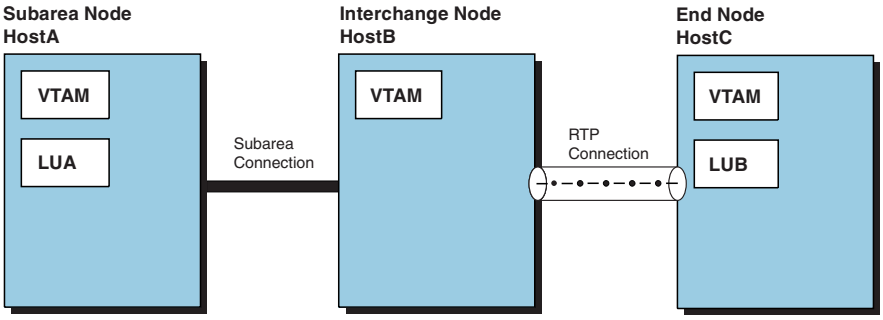


Figure 122. Interchange node utilizing HPR routing between subarea and APPN

Similarly, if the session is from a subarea node through an interchange node to a network node that is one hop away from the interchange node, translation of subarea to HPR can occur if the interchange node is IBM Communications Server for OS/390 V2R10 or later and the network node is one of the following:

- A network node with IBM Communications Server for OS/390 V2R10 or later installed
- A non-z/OS Communications Server network node

For instance, in Figure 122, if HOSTC were a network node, HOSTB will translate subarea data received from HOSTA to HPR and route it across the RTP connection to HOSTC, provided that HOSTB and HOSTC meet the above criteria.

APPN session endpoint is more than one hop from the interchange node: If the session endpoint in the APPN network is an end node or network node that is more than one hop away from the interchange node, translation of subarea to HPR can occur if the interchange node is IBM Communications Server for OS/390 V2R10 or later and the adjacent network node is one of the following:

- A network node with IBM Communications Server for OS/390 V2R10 or later installed
- A non-z/OS Communications Server network node

Note, however, that the RTP connection will only extend from the interchange node to the adjacent network node. An additional, separate, RTP connection can be established for the remainder of the APPN session path (depending on the HPR and RTP capabilities of the nodes on the path), but a single end-to-end RTP connection between the interchange node and the APPN session endpoint will not be used. In Figure 123 on page 503, for example, HOSTD is part of a subarea network connected to an APPN network through HOSTC. In the case of a session between LUD and LUA, one RTP connection can be established between the HPR-capable endpoints HOSTC and HOSTB, and a second RTP connection can be established between HOSTB and HOSTA. Of course, for sessions between LUC and LUA, a single end-to-end RTP connection can be used between HOSTC and HOSTA (through HOSTB).

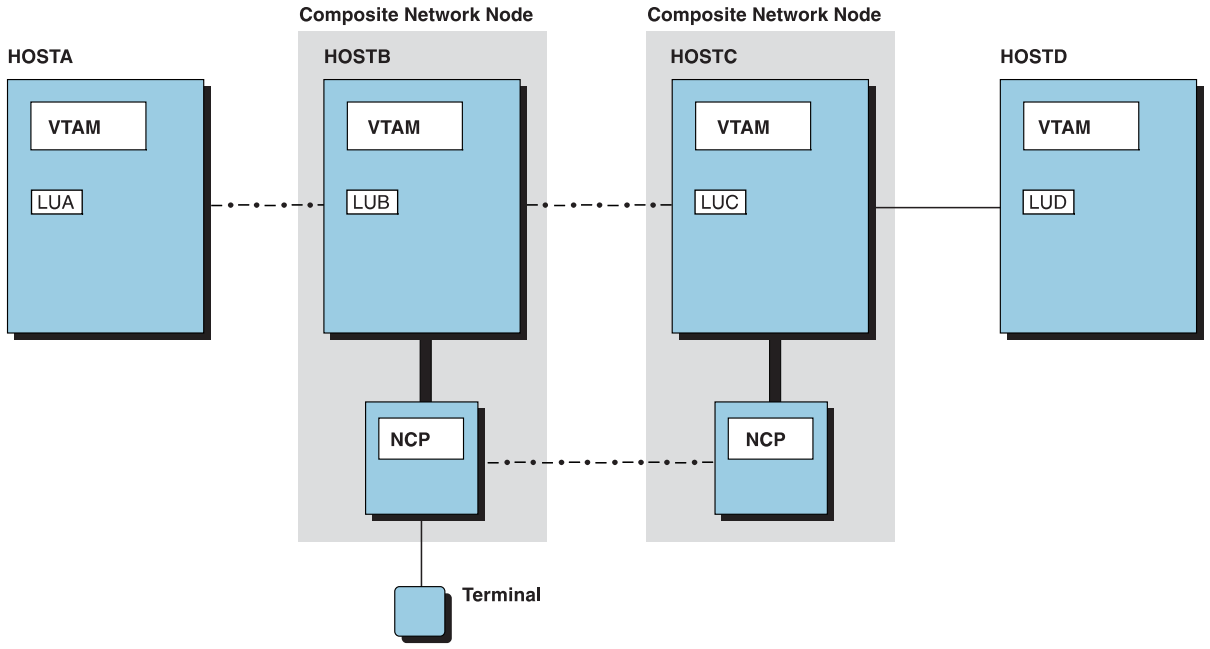


Figure 123. Sessions traversing APPN and subarea networks

Session endpoints are in separate subarea networks: When the session endpoints are in separate subarea networks connected by one or more APPN networks, translation of subarea to HPR can occur if the interchange nodes are IBM Communications Server for OS/390 V2R10 or later. For instance, in Figure 123, if HOSTA and HOSTD are both subarea nodes and there is a session between LUA and LUD through HOSTB and HOSTC, HOSTB and HOSTC will translate subarea data received to HPR and route it across the RTP connection between them. If the interchange nodes are separated by other APPN nodes, the RTP connection from each interchange node is limited to its adjacent node. The RTP connection between intermediate nodes can include multiple nodes.

Sessions through NCP in a composite network node providing RTP support: When a composite network node is providing RTP-level support as a session endpoint, a session must enter or exit the composite network node through VTAM to use HPR. HPR routing is not used when a session traverses such a composite network node through NCP (in through NCP, out through NCP). In Figure 123, for example, HPR is not used for a session between LUC and the terminal that traverses the connection between the two NCPs. However, HPR is used between VTAMs for the same session if it traverses the connection between the two VTAMs. HPR is also used between VTAMs for a session between LUA and the terminal.

Setting session paths using HPRNCPBF

HPRNCPBF is a VTAM start option that allows you to determine the path used for session setup for LUs whose entry point to the HPR network is by way of NCP/FID4. This lets you determine whether HPR should be used in cases where it will cause session data to travel through an NCP twice.

In this case, a more optimal HPR TG between the VTAM nodes exists. When the LU requests a session with *appla*, the session is usually set up directly through the NCP to *appla* on VTAM2. While this path might offer better performance, it cannot take advantage of HPR features, such as dynamic rerouting or the high availability offered by MNPS. HPRNCPBF can be set to either YES or NO (the default). If you

select NO, the session continues to be set up over the direct path. If you select YES, the session will be set up over the HPR pipe between the two VTAM nodes. Specifically, it will use the VR-TG path. In this case, in a connection failure between the NCP and VTAM, the session is redirected over an alternate route without disruption.

Note: If you select YES, the session requests from the NCP-attached LU are routed through the NCP twice.

For more information, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

Chapter 18. Implementing a combined APPN and subarea network

A sample communication management configuration (CMC) is shown in Figure 124. This configuration is not intended to be typical or recommended, but is used here to demonstrate converting to APPN.

Before using the APPN capabilities of VTAM, any communication controller that is to support APPN links (all of them in Figure 124) must be a 3745 with NCP Version 6 Release 2 or later. Communication controllers that do not support APPN links do not require NCP Version 6 Release 2 or later.

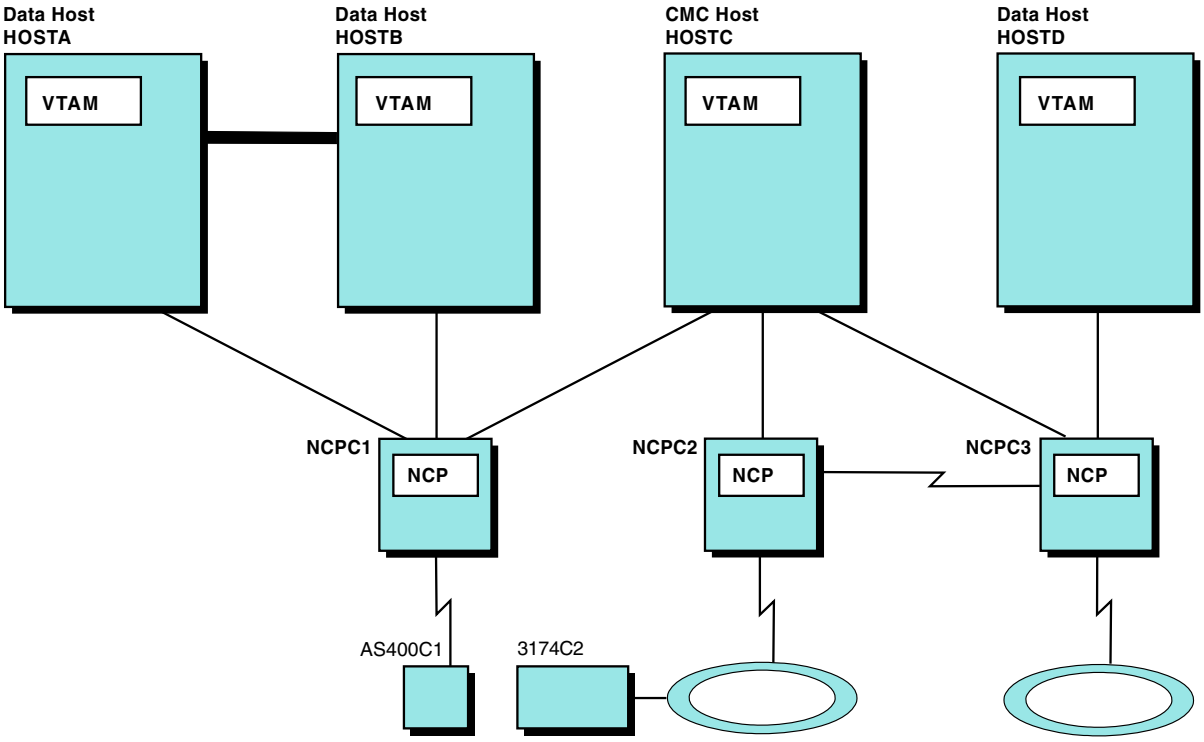


Figure 124. Example of communication management configuration

Use the following steps to begin to convert this configuration to APPN:

1. Determine which connections to type 2.1 nodes you want to be LEN connections, and which connections you want to be APPN connections.

   All connections between the NCPs and type 2.1 nodes are automatically tried as APPN connections when APPN support is specified, unless you explicitly designate them as LEN connections, using the CONNTYPE=LEN start option or the CONNTYPE=LEN operand on the GROUP, LINE, or PU definition statement. By specifying CONNTYPE=LEN initially, connections can later be converted to APPN in phases.

   Notes:

   a. Prior versions of VTAM did not enforce CPNAME uniqueness for LEN nodes, except when CPNAME is coded on a switched PU definition statement. It is possible that your network has PUs with duplicate

© Copyright IBM Corp. 1984, 2011

CPNAMEs. This CPNAME duplication should be resolved prior to a node becoming an APPN node. Specifying CONNTYPE=LEN for a connection causes VTAM to avoid checking for duplicate CPNAMEs, except when CPNAME is coded on a switched PU definition statement.

b. When converting a connection from LEN to APPN, ensure that the adjacent link station name (VTAM-defined PU name) for an adjacent node is unique from the CP name of that node.

If you want most or all connections to type 2.1 nodes to initially be LEN connections, code the CONNTYPE=LEN start option at the CMC host. Then, if you do not code CONNTYPE=APPN on any GROUP, LINE, or PU definition statement, all connections to type 2.1 nodes are LEN connections.

If you want only particular connections to type 2.1 nodes to initially be LEN connections, do not use the CONNTYPE=LEN start option. Instead, code CONNTYPE=LEN on the appropriate GROUP, LINE, or PU definition statements.

2. Code the NODETYPE=NN start option at the CMC host.

   The NODETYPE start option is required to specify APPN support. Assuming that the HOSTSA start option is also coded for HOSTC, coding NODETYPE=NN makes this VTAM an interchange network node. An interchange network node supports subarea and APPN protocols, and is capable of transforming subarea protocols to APPN protocols, and vice versa.

   To APPN, the CMC host and all owned NCPs (NCPC1, NCPC2, and NCPC3) become one composite network node, as shown in Figure 125.

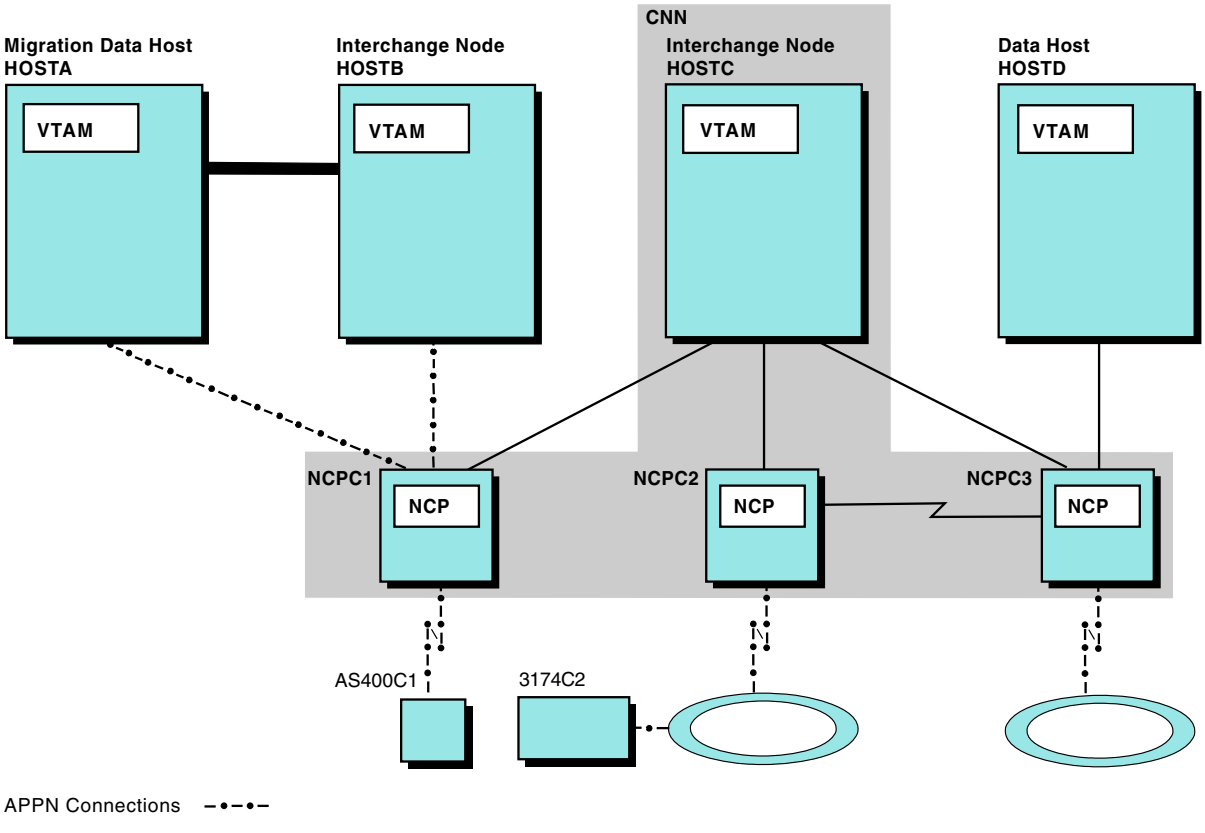


Figure 125. Communication management configuration after conversion

At this point, however, provided that all connections to type 2.1 nodes were specified as LEN connections in step 1 on page 505 using CONNTYPE=LEN, VTAM and NCP perform only their traditional subarea functions. If VTAM is started in HOSTC, no CP-CP sessions would be started between the composite network node and the adjacent nodes because connections to type 2.1 nodes are LEN connections and the adjacent VTAMs do not have APPN support.

3. Depending on your conversion strategy determined in step 1, convert LEN connections to APPN connections as desired by doing one of the following:
   - Changing CONNTYPE=LEN to CONNTYPE=APPN (or simply deleting CONNTYPE=LEN) on the appropriate GROUP, LINE, or PU definition statements (and inactivating and reactivating the appropriate links), or restarting VTAM with the CONNTYPE=APPN start option.
   - Specifying any other desired APPN operands on the GROUP, LINE, or PU definition statement (for example, NN, CPCP, or TGP).

     Note: If the NN operand is not specified, the node type (NN or EN) of the adjacent type 2.1 node is determined when the connection is activated. If specified incorrectly, the connection will fail.
   - Defining the connection as APPN at the adjacent type 2.1 node.

   In the example in Figure 125 on page 506, all LEN connections are converted to APPN connections. With APPN connections, HOSTC can be used as the network node server, and independent LUs can now set up LU-LU sessions with the composite network node using APPN searches and flows. In addition, if a new independent LU is added to a type 2.1 node that is adjacent to the composite network node, the independent LU must now be defined only to that adjacent type 2.1 node. VTAM requires no definition for the added LU.
4. Code the NODETYPE=NN start option at data host B. Code a local SNA major node (VBUILD TYPE=LOCAL) that defines the type 2.1 channel attachment between data host B and NCPC1. Code PUTYPE=2 on the PU definition statement in the local SNA major node.

   Assuming that the HOSTSA start option is also coded at HOSTB, this VTAM is now an interchange network node as shown in Figure 125 on page 506.

   To the composite network node, HOSTB appears as a network node. The connection between HOSTB and NCPC1 is now a type 2.1 connection. A CP-CP session is now possible between HOSTB and the composite network node. Thus, LUs connected to the composite network node through one of the NCPs can now set up LU-LU sessions with any application in HOSTB using APPN searches and flows. New applications added in HOSTB need not be defined to the composite network node.

   The change of the link between HOSTB and NCPC1 to an APPN link results in the creation of additional NCP control blocks, which could require additional storage space. Whether you see a net decrease or increase in NCP storage is dependent upon the decrease in path table sizes in relation to any increase in storage due to the number of cross-domain sessions that are now boundary function sessions over APPN connections.
5. Code the NODETYPE=EN start option at data host A. Again, use a local SNA major node (VBUILD TYPE=LOCAL) to define the type 2.1 channel attachment between data host A and NCPC1, and code PUTYPE=2 on the PU definition statement in the local SNA major node.

   Assuming that the HOSTSA start option is also coded at HOSTA, this VTAM is now a migration data host as shown in Figure 125 on page 506. A migration data host node supports subarea and APPN protocols, but does not act as an intermediate node. A migration data host cannot activate any NCPs.

To the composite network node, HOSTA appears as an end node. The connection between HOSTA and NCPC1 is now a type 2.1 connection. A CP-CP session is now possible between HOSTA and the composite network node. Thus, LUs connected to the composite network node through one of the NCPs can now set up LU-LU sessions with any application in HOSTA using APPN searches and flows. New applications added in HOSTA need not be defined to the composite network node.

The change of the link between HOSTA and NCPC1 to an APPN link results in the creation of additional NCP control blocks, which could require additional storage space.

A migration data host can also still participate in SSCP-SSCP sessions with another VTAM (for example, HOSTB). If the channel-to-channel (CTC) connection between HOSTA and HOSTB is not also converted to APPN, you can continue to use subarea CTC support over that connection.

In this example, the NODETYPE start option is not coded at HOSTD. HOSTD can still communicate with resources in the HOSTC domain using the subarea connection to NCPC3. HOSTD can also still communicate with resources in the HOSTA and HOSTB domains using the interchange node function of HOSTC.

As a result of the above conversion steps, the following definitions are no longer required. You can remove them, but they do not have to be removed.

Note: If VR-based TGs are going to be used, you need to maintain definitions for routes and CDRMs. See "Virtual-route-based transmission groups" on page 95 for additional information.

- All PATH definitions for routes between HOSTA or HOSTB and other subareas, except PATHs for the CTC connection between HOSTA and HOSTB
- In HOSTA and HOSTB, the CDRM definitions for HOSTC and HOSTD
- In HOSTC and HOSTD, the CDRM definitions for HOSTA and HOSTB
- CDRSC definitions in HOSTA and HOSTB for LUs in HOSTC and HOSTD, if present
- CDRSC definitions in HOSTC and HOSTD for LUs in HOSTA and HOSTB, if present
- Wildcard definitions in 3174C2 and AS400C1 that allow requests for unknown resources to be forwarded to VTAM over LEN links (assuming the links to 3174C2 and AS400C1 are now APPN links)
- ALS selection logic in the session management exit that allows requests to be forwarded from VTAM to independent LUs over LEN links that have been converted to APPN
- APPN-capable PUs included on the ALSLIST operand on the CDRSC definition statement (They can be optionally replaced by a single ISTAPNPU entry, a reserved keyword that represents any APPN capable PU.)

Start options defining a combined subarea and APPN environment

In addition to the CONNTYPE, NODETYPE, and HOSTSA start options, some other start options used in a combined subarea and APPN environment are associated with locating resources. All new and changed start options default to optimal values for an APPN network. During conversion, however, these defaults can be modified based on your particular conversion strategy and configuration.

SORDER start option

SORDER controls the order in which the APPN and subarea portions of the network are searched when a search request is processed by an interchange node or migration data host. SORDER can be specified as a start option or coded on individual ADJSSCP table definitions (or both). If SORDER is not coded for a given ADJSSCP table, the current value of the SORDER start option value is used each time that ADJSSCP table is selected for use.

Note: If SSCPORD=PRIORITY is specified and previous searches for the target resource have been performed, the search order might be modified based on the results of these previous searches.

For further information about SORDER, see "Controlling searches" on page 519.

SSEARCH start option

The setting of the SSEARCH start option determines whether the subarea network is searched when search requests from the APPN network arrive at an interchange node. By default, the subarea network is searched.

Note: Resources in the domain of the interchange node are found even when SSEARCH=NO is specified.

For more information on the SSEARCH start option, see "Controlling searches" on page 519.

CDRSCTI start option

The setting of the CDRSCTI start option controls the amount of time that a dynamic CDRSC is retained by VTAM after its last session ends.

However, dynamic CDRSCs representing APPN resources are freed by an interchange node immediately after the last session ends, regardless of the specified CDRSCTI value. This is because APPN directory services in the interchange node has an entry for each APPN resource.

Dynamic CDRSCs representing subarea resources are retained for eight minutes (by default). You can increase the CDRSCTI start option value to minimize the potential for unnecessary broadcasts of the APPN network for subarea resources.

For further information on the CDRSCTI start option, see "Controlling searches" on page 519.

IOPURGE start option

The setting of the IOPURGE start option specifies an interval after which outstanding I/O requests are purged. The types of I/O requests that are checked are:

- CDINIT requests
- Direct search list requests
- APPN search requests
- HPR route setup requests

For CDINIT requests, after the specified time interval, if a response has not been received, VTAM continues its search through the adjacent SSCP table until it either

finds the resource or the table has been exhausted. For the other requests, if IOPURGE expires, VTAM handles the request as if a negative response to the request has occurred.

If IOPURGE is set too low, outstanding searches will be terminated before normal searching has completed. This can result in failed searches and possibly increased network traffic due to retries. This can occur if IOPURGE on an end node or migration data host is set to a value that does not allow the network node server to complete all search tasks and return a reply.

If IOPURGE is set to 0, outstanding I/O requests can remain outstanding indefinitely.

Note: Do not use IOPURGE for congestion control. If you have APPN congestion control problems, use a lower MAXLOCAT value. IOPURGE was intended for situations where a node drops a request and never responds and not for congestion control.

Dependent LUs

VTAM extends support for dependent (non-LU 6.2) LUs to APPN networks and combined subarea and APPN networks. Dependent LUs are owned by VTAM and rely on the SSCP to start sessions. There are SSCP-PU and SSCP-LU sessions for dependent LUs. Dependent LUs can start a session when the primary LU supports SLU-initiated sessions (for example, as does a VTAM application program). Dependent LUs can be the secondary LU in sessions with VTAM application programs in APPN, subarea, and combined APPN and subarea networks.

Note: If bisynchronous (BSC) 3270 sessions are required, traditional subarea paths must be retained for routing these sessions. These same subarea paths can be shared by APPN sessions using VR-based TG, permitting coexistence of APPN and BSC sessions.

Dependent LU server

The dependent LU server (DLUS) function facilitates conversion from a subarea environment to an APPN environment, allowing you to maintain central management of remote dependent LUs while benefiting from APPN throughout a network.

Note: The DLUR and DLUS architecture supports dependent secondary LUs (SLUs) only. The AS/400 SPLS function is not supported by DLUR and DLUS.

Two LU 6.2 sessions (one inbound, one outbound) are established between a DLUS and a dependent LU requester (DLUR), and these LU 6.2 sessions are collectively known as the CPSVRMGR pipe. SSCP-PU and SSCP-LU session flows use the CPSVRMGR pipe. An SSCP-PU session is established between a VTAM network node and the PU that owns the dependent LU, and an SSCP-LU session is established between VTAM and the dependent LU. Session initiation flows for the dependent LU are sent over the SSCP-LU session, and VTAM can use subarea or APPN flows to initiate a session with the PLU. BIND and session data are then routed directly between the PLU and the dependent LU.

For example, in Figure 126 on page 511, VTAMB is a dependent LU server for LU4 and LU5. SSCP-PU sessions are established between VTAMB and PU4 and

between VTAMB and PU5, and SSCP-LU sessions are established between VTAMB and the dependent LUs (LU4 and LU5). Session initiation flows for the dependent LUs are sent over the SSCP-LU sessions. VTAMB uses APPN flows to initiate a session with a PLU that VTAM can communicate with through an APPN network (for example, LU2 and LU3), and subarea flows to initiate a session with a PLU reachable through a subarea network (for example, APPLA1). The BIND and session data are then routed directly between the PLU and the dependent LUs.

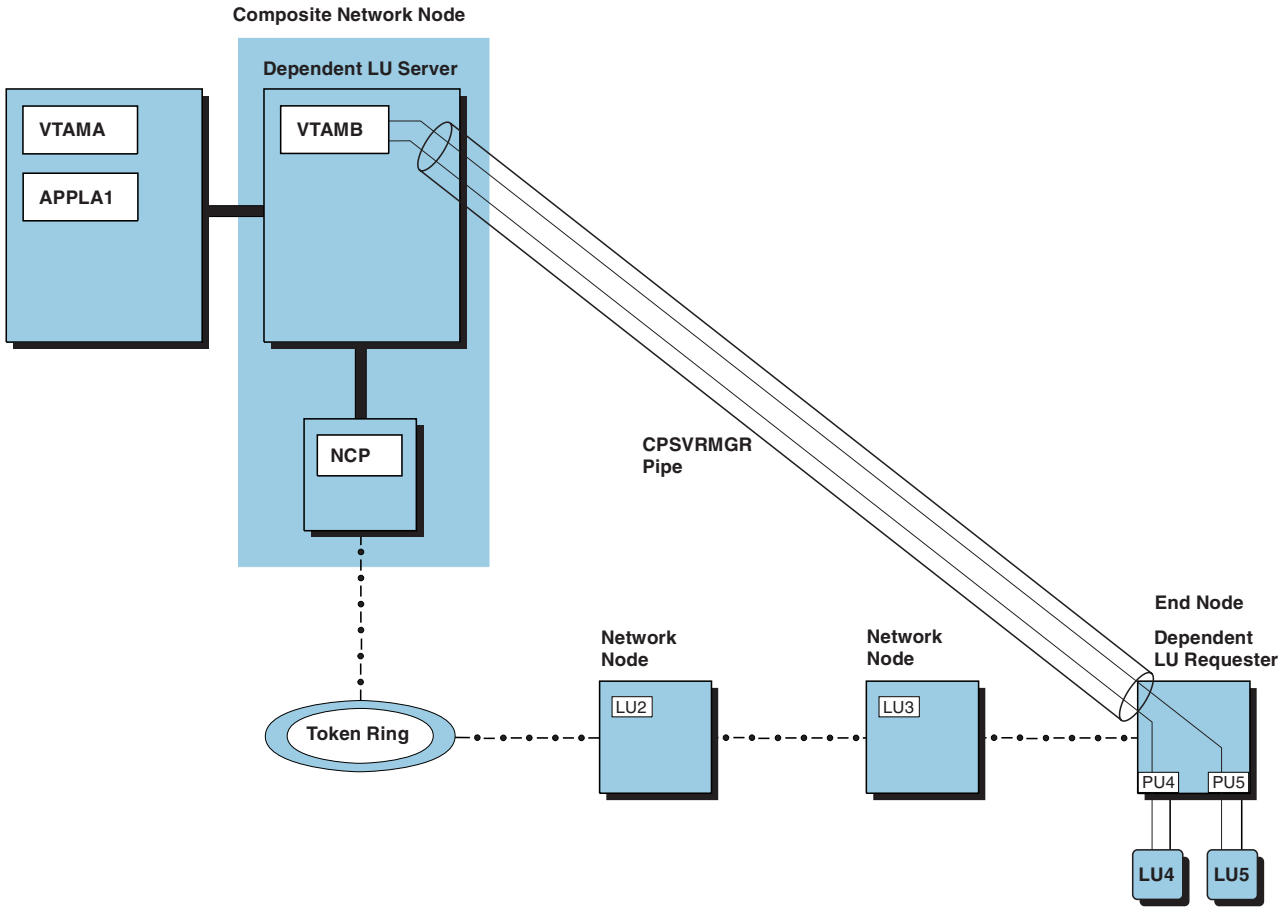


Figure 126. VTAM functioning as a dependent LU server

To implement the dependent LU server function, consider the following:

- NODETYPE=NN must be specified for a VTAM DLUS.
- For DLUR-initiated PU activation, no system definition is required. The dynamic switched definition facility can be used to dynamically define the PU. If the DLUR is adjacent, a PU name must be added to the REQACTPU by the DLUR. When the DLUR is adjacent, a CPNAME is sent in the REQACTPU and used to search for a PU. The previously defined CPNAME is found, and the REQACTPU is rejected. For information on the dynamic switched definition facility, see Chapter 9, “Defining resources dynamically,” on page 227 and “Dynamic switched definitions” on page 227.
- Dynamically defined PUs can be displayed using the DISPLAY ID command, specifying the DLUR CP name on the ID operand. This shows all PUs that currently have an SSCP-PU session (both dynamic and predefined PUs). You can also determine which LUs currently have an SSCP-LU session by displaying a specific PU.

- For VTAM to initiate activation of a PU, define the dependent LU requester by including the DLURNAME and DLCADDR operands on the PATH definition statement in the switched major node (regardless of whether its connection to the DLUR is switched or leased). DLURNAME specifies the CP name of the DLUR that owns the PU, and DLCADDR includes data link control (DLC) information used by the DLUR to locate the PU.
- In some cases the DLUS and the network node server of the DLUR are different VTAM nodes. In this case, there must be a searchable APPN path from the DLUS to the NNS of the DLUR. This path can go through multiple nodes if necessary.

  Note: If the DLURNAME is not fully qualified, the NETID of the DLUS is used.

  A sample switched major node to match Figure 126 on page 511 is shown in Figure 127. DLUR1 is used in the switched major node as the name of the DLUR shown in Figure 126 on page 511.
- The DLUS and DLUR must be connected over an APPN path that supports the CPSVRMGR Class of Service.
- Takeover and giveback is supported for DLUR-attached PUs. The PU must be defined in VTAM with ANS=CONT, and the DLUR must be capable of handling ANS=CONT support. Giveback processing for DLUR-supported PUs occurs when an operator command is issued from the DLUS to relinquish control of the PU (for example, V NET,INACT,ID=dlur_pu,G), or an outage occurs for the CPSVRMGR pipe. Takeover processing can occur automatically for redial or can be initiated by the operator with a VARY DIAL command, for PUs previously released by giveback processing.

```
SAMPSWMN VBUILD TYPE=SWNET
PU4      PU    ADDR=01,                                          X
               IDBLK=05D,                                        X
               IDNUM=00001,                                      X
               USSTAB=AUSSTAB,                                   X
               MODETAB=AMODETAB,                                 X
               ISTATUS=ACTIVE                                    X
PATHU4   PATH  PID=1,                                            X
               DLURNAME=DLUR1,                                   X
               DLCADDR=(1,C,INTPU),                              X
               DLCADDR=(2,X,05D00001)
LU4      LU    LOCADDR=1,                                        X
               PACING=(1,1),                                     X
               VPACING=2*
PU5      PU    ADDR=02,                                          X
               IDBLK=05D,                                        X
               IDNUM=00002,                                      X
               USSTAB=AUSSTAB,                                   X
               ISTATUS=ACTIVE                                    X
PATHU5   PATH  PID=1,                                            X
               DLURNAME=DLUR1,                                   X
               DLCADDR=(1,C,INTPU),                              X
               DLCADDR=(2,X,05D00002)
LU5      LU    LOCADDR=1,                                        X
               PACING=(1,1),                                     X
               VPACING=2,                                        X
               MODETAB=AMODETAB
```

Figure 127. Switched major node for a dependent LU server

Notes:

1. If using NCP, NCP Version 6 Release 2 or later is required.

2. The VARY INACT,TYPE=GIVEBACK command is used to release a DLUR-attached switched PU (and its subordinate LUs) from SSCP ownership. To use this function, the DLUR must support ANS=CONT.
3. A DLUS can serve multiple DLURs simultaneously.
4. A DLUR can be served by multiple DLUS VTAMs simultaneously.
5. Multiple DLURs can support the same PU (one at a time). The SSCP-PU session is through only one DLUR at any given time.
6. DLUS function cannot be used with XRF or cryptography in an XRF environment.
7. Redial occurs as follows for DLUS supported PUs:
   - When a DLUS initiates the activation of a PU but receives a negative response, VTAM attempts to redial over each valid path statement for the PU until successful or until all valid paths have been tried. Redial does not occur if the negative response indicates that the PU is already active, or that the fully-qualified PCID is not unique. If the fully-qualified PCID is not unique, the DLUS attempts to redial the PU over the same path with a newly generated PCID.
   - Redial does not occur if PU activation was initiated by the DLUR.
   - When a protocol violation, TDU error, or CPSVRMGR session outage signal is received for a particular DLUR, VTAM attempts to redial every active or pending-active PU served by that DLUR for which a valid PATH statement is found. If the PU is already active, VTAM performs the following:
     – If the PU was defined with ANS=CONT and the DLUR supports this function, giveback processing is performed prior to attempting redial.
     – If the PU was defined with ANS=STOP or the DLUR does not support ANS=CONT, the PU is deactivated prior to attempting redial.
8. When a CP-CP session between a DLUS and DLUR fails, the CPSVRMGR session is deactivated, enabling reactivation of the CP-CP session.
9. If the DLUS is located in a different APPN subnetwork from that of the DLUR or the PLU, APPN multiple network connectivity support is required (see "Dependent LU server support across subnetwork boundaries" on page 514). If CPSVRMGR pipe initiation is attempted across an APPN subnetwork boundary and the DLUR does not support this function, a protocol violation is detected and the pipe is terminated.
10. Information regarding DLUR-attached resources can be processed by the configuration services XID exit. This exit routine can be coded so that VTAM either processes or denies requests for contact from known switched devices, or so that VTAM processes or denies requests for PU activation from a DLUR. For details, refer to*z/OS Communications Server: SNA Customization*.
11. The IDBLK/IDNUM specified on the PU definition statement must match the value specified on the DLCADDR operand (when DLCADDR information is provided for IDBLK/IDNUM) on the associated PATH statement.
12. When a DLUR-attached dependent LU starts a session, or reports an active session as part of DLUR takeover, VTAM (as the DLUS) receives session information on the SESSST and passes that information to the network logical data manager (NLDM). However, if the session uses HPR and a path switch occurs, VTAM will not be aware of the change unless VTAM is at one end of the HPR route or if DLURSAW=YES is used. To supply NLDM with the updated path switch data between the DLUR and the other end of the session, even when VTAM is not at one end of the HPR route, use the DLURSAW=YES (this is the default) start option. The DLUR must be capable of reporting this information and be configured to do so.

13. For information on the interaction between the DLURSAW and SAVERSCV start options, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Dependent LU server support across subnetwork boundaries

To use dependent LU server support across subnetwork boundaries, APPN multiple network connectivity support is required. NCP Version 7 Release 1 is also required for DLUS support across subnetwork boundaries.

If a DLUS is located in a different APPN subnetwork than a DLUR, the subnetwork of the DLUS must enable APPN multiple network connectivity support, as shown in Figure 128.

Note: The node on the border of the DLUR subnetwork (in Figure 128, the network node in subnetwork B) can be a network node, peripheral border node, or extended border node.

Also shown in Figure 128, if a DLUS is located in a different APPN subnetwork than the primary LU, both the subnetwork of the DLUS and the subnetwork of the primary LU must enable APPN multiple network connectivity support.

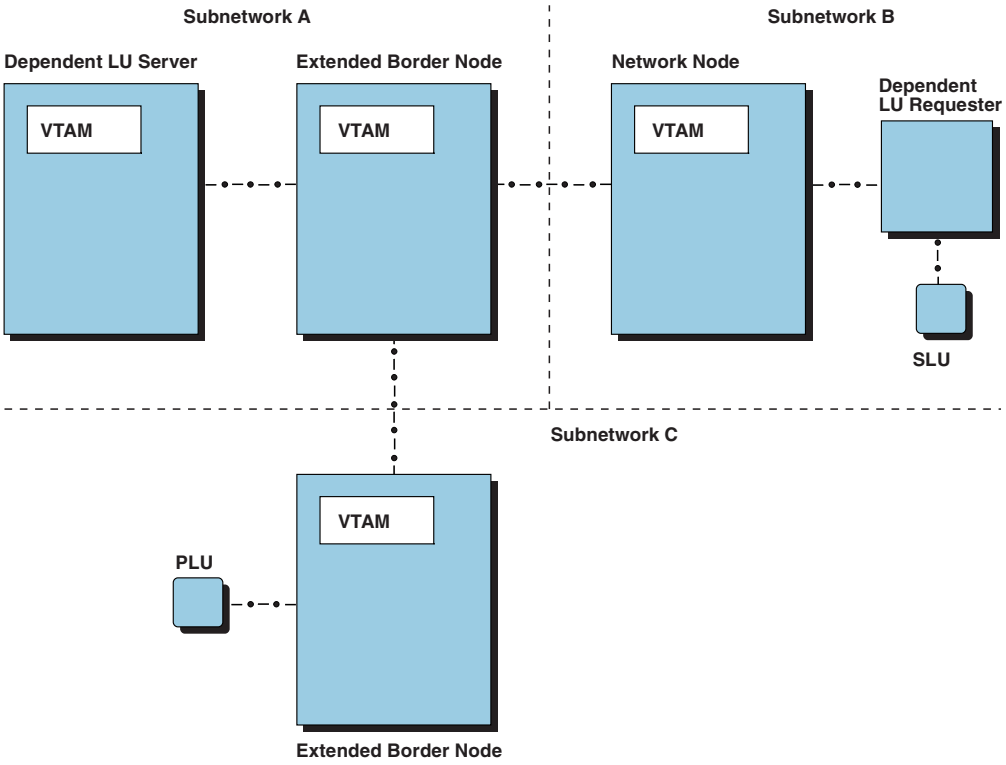


Figure 128. DLUS located in different APPN subnetwork than DLUR or PLU

If a primary LU exists in or through a subarea network, as shown in Figure 129 on page 515, all other APPN subnetworks through which the primary LU might be reached that use interchange nodes for connectivity to or through the subarea network must enable APPN multiple network connectivity support.

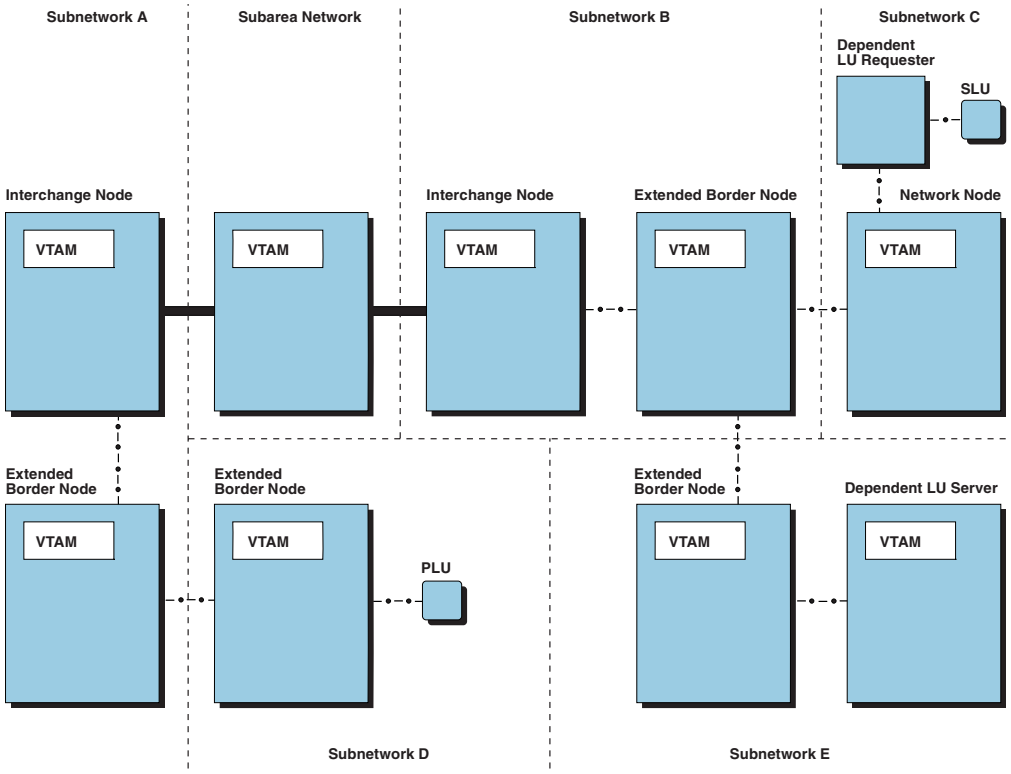


Figure 129. PLU exists in or through a subarea network

For further information on APPN multiple network connectivity support, see "APPN multiple network connectivity" on page 89.

Defining CDRSCs

When a resource is found in or through an APPN network, the interchange node through which the APPN network is entered becomes the owning CDRM of the resource to other VTAMs in the subarea network, unless the resource was predefined as a CDRSC with an owning CDRM name coded. If there are multiple interchange nodes, the owning CDRM can change, depending on the best search route at the time. If cross-domain resources (CDRSCs) are defined with the CDRM operand, they might have to be modified to specify an interchange node as the owning CDRM.

Shared NCPs

In a combined APPN and subarea network, VTAM provides for shared ownership of a local or remote NCP by two VTAMs.

If two VTAMs share an NCP, they both must have a subarea connection to that NCP and thus a subarea path between them. These two VTAMs can communicate through an SSCP-SSCP session. The topology of the APPN network connected to one VTAM is not shared with the APPN network connected to the other VTAM, unless CP-CP sessions also exist between the two VTAMs. If CP-CP and SSCP-SSCP sessions exist between the two VTAMs, the decision about which control session to use for a specific LU-LU session setup depends on resource locations, start options, and tables.

SSCP takeover

In a combined APPN and subarea network, VTAM provides full takeover capability.

A connection that is established as an APPN or LEN connection remains such until it is deactivated. However, an SSCP taking over an APPN connection views it as an LEN connection if the takeover SSCP has not implemented APPN. No CP-CP sessions can be established over the connection until the original SSCP regains control.

For example, in Figure 130, if HOSTA fails and HOSTB takes over, the connection from the NCP to the adjacent CP is viewed by HOSTB as an LEN connection if HOSTB has not implemented APPN. No CP-CP sessions can be established over the connection until HOSTA regains control.

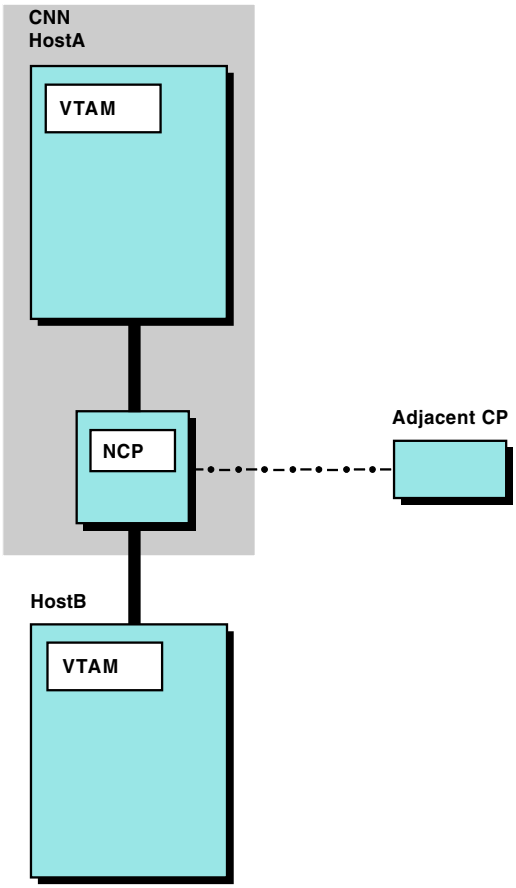


Figure 130. SSCP takeover when adjacent CP does not support CP name change

If the takeover SSCP has implemented APPN, a connection is still treated as an LEN connection if both of the following are true:

- The takeover SSCP is not the original SSCP, in which case the adjacent CP has a connection to a different SSCP after takeover (CP name change).
- The adjacent CP does not support the changing of CP names.

Although the adjacent CP still views the connection as an APPN connection with the original SSCP, the takeover SSCP views the connection as an LEN connection. Therefore, no CP-CP sessions can be established over the connection until the original SSCP regains control.

For example, in Figure 130 on page 516, if HOSTA fails and HOSTB takes over, the CP name (from the perspective of the adjacent CP) changes from HOSTA to HOSTB. If the adjacent CP does not support the changing of CP names, HOSTB treats the connection to the adjacent CP as an LEN connection (even if HOSTB has implemented APPN) and no CP-CP sessions can be established over the connection until HOSTA regains control. If the adjacent CP does support the changing of CP names, CP-CP sessions can be established when HOSTB takes over.

If the adjacent CP is located over a subnetwork boundary, the following conditions must also be met to allow a new CP-CP session with the takeover SSCP:

- The takeover SSCP must have APPN multiple network connectivity support.
- The definitions in the takeover SSCP must be consistent with those in the original SSCP (for example, a subnetwork boundary cannot be changed to a native connection during takeover).

If these conditions are not met, the takeover SSCP views the connection as an LEN connection. For information on APPN multiple network connectivity support, see "APPN multiple network connectivity" on page 89.

If the adjacent control point is another composite network node as shown in Figure 131 on page 518, SSCP takeover with APPN connectivity occurs if the takeover SSCP has implemented APPN. For example, in Figure 131 on page 518, if HOSTB has implemented APPN and HOSTA fails, HOSTB can take over, and a CP-CP session between the composite network nodes (HOSTC/NCP2 and HOSTB/NCP1) is established. The APPN connection between the NCPs is viewed as an LEN connection by the adjacent control point after takeover if the takeover SSCP has not implemented APPN.

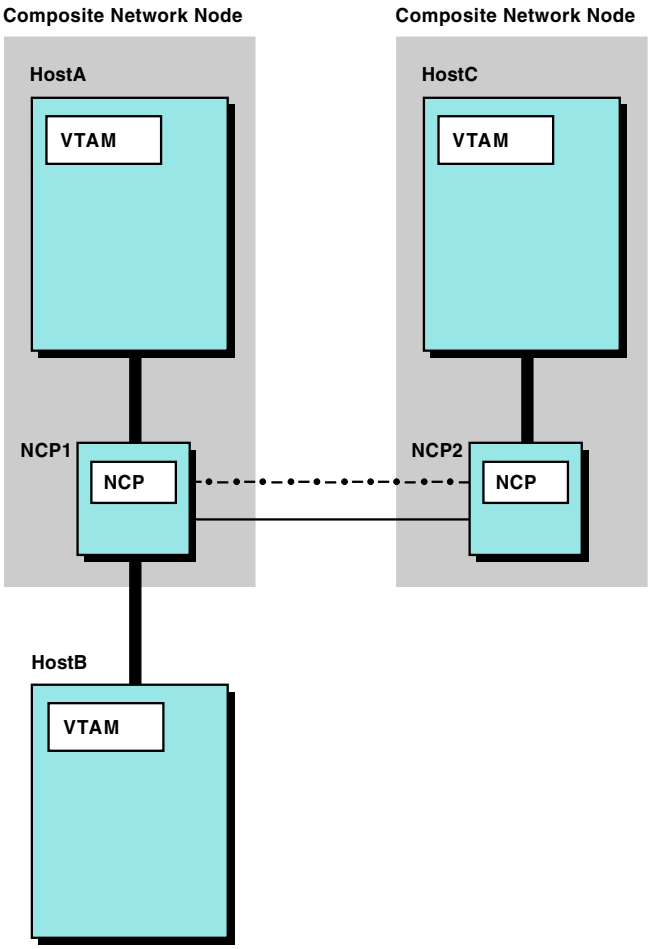


Figure 131. SSCP takeover when adjacent CP is another composite network node

Notes:

1. In Figure 131, if HOSTA takes over for HOSTC or HOSTC takes over for HOSTA, APPN and LEN connectivity is not available after takeover (assuming a CP-CP session originally existed between only the composite network nodes shown) because the takeover SSCP cannot have a connection with itself (that is, after takeover, the takeover SSCP and the NCPs now form a composite network node). APPN and LEN connectivity is not available until the failing host regains control or another host (for example, HOSTB) takes control of the NCP originally owned by the failing host.
2. In Figure 131, a subarea connection and an APPN connection exist between the NCPs. If both the subarea connection and the APPN connection are active concurrently, the route used for communication between the NCPs is unpredictable. To control which link is used, you can choose to activate only one connection. For SSCP takeover, however, the subarea connection must be activated if not already active.

Establishing and controlling sessions

In a combined APPN and subarea environment, an independent LU can be accessible over APPN and LEN connections. In this case, VTAM always gives preference to APPN connections, if the connections are known and available. APPN connections are known when an APPN-capable PU name or ISTAPNPU is included in the ALSLIST for the resource. If no APPN connection is available, an LEN connection is used (if available).

To cause VTAM to sometimes use an LEN connection or sometimes use an APPN connection, use the adjacent link station selection function of the session management exit routine. For information on coding this exit routine, refer to *z/OS Communications Server: SNA Customization*.

To force VTAM to use particular LEN connections and not allow a search of the APPN network, include only those LEN adjacent link stations on the ALSLIST operand and also code ALSREQ=YES for the resource. Session setup will continue using an LEN adjacent link station from the list even if a route exists through the APPN network.

Controlling searches

When trying to locate a DLU, an interchange node or migration data host can search the APPN network, the subarea network, or both. The order that VTAM searches is based on the defined ADJSSCP tables, start option values, and the origin of the search request. Using the ADJSSCP tables and start options, you can concentrate the search on the part of the network containing the most frequently sought resources. See "Adjacent SSCPs" on page 541 for information on using ADJSSCP tables.

For further information on searching an APPN network, see "Network routing and resource location for APPN nodes" on page 325.

For further information on searching a subarea network, review the appropriate sections of Chapter 19, "Implementing a subarea network," on page 529, beginning with "Start options defining other domains" on page 530.

Using SORDER to control network search order

SORDER controls the order in which the APPN and subarea portions of the network are searched when adjacent SSCP routing is performed for a search request received by an interchange node or migration data host. SORDER can be specified as a start option or coded on individual ADJSSCP table definitions (or both). In general, the SORDER start option value should be chosen based on where the most frequently sought after resources are located (in the APPN network or in the subarea network). Although this approach provides the best search performance for MOST resources in the network, a single SORDER value rarely results in optimal search performance for ALL resources. For these cases, the search order can be specified on a more granular level by coding SORDER individually on each ADJSSCP table.

The default value for the SORDER start option (SORDER=APPN) will cause VTAM to prefer searching the APPN network before searching the subarea network (adjacent CDRMs). However, if the CDRM that owns the target resource is known (defined on the cross-domain resource definition or learned from previous searches), new searches will be sent to this CDRM first, prior to searching the

APPN network. If the APPN network should always be searched first even before the owning CDRM (for example, to maximize the use of High Performance Routing), SORDER=APPNFRST should be used.

If most target resources are still located in the subarea portion of the network, SORDER=SUBAREA should be used. In this case, the APPN network will be searched only after all adjacent CDRMs have been searched. If VTAM has been enabled for APPN but there are currently no resources located in the APPN network (that is, no APPN searching should be performed), SORDER=ADJSSCP should be used. With SORDER=ADJSSCP, the APPN network is only searched if an ISTAPNCP ADJCDRM entry has been explicitly coded in the ADJSSCP table that is chosen for the search.

To specify the search order for individual adjacent SSCP tables, code the SORDER operand on the NETWORK or CDRM statements that define each adjacent SSCP table. If SORDER is coded on a NETWORK statement, that value will sift down to all subordinate adjacent SSCP tables until the next NETWORK (or ADJLIST) statement is encountered. To override the SORDER value on the preceding NETWORK statement, code the SORDER operand on the CDRM statement that defines the subordinate adjacent SSCP table. To override the SORDER value on the preceding NETWORK statement with the current SORDER start option value, code SORDER=STARTOPT on the CDRM statement. (SORDER=STARTOPT is the default value for NETWORK statements.)

Notes:

1. If a search request is received by VTAM from the APPN network, SORDER is ignored. The search order used for searches received from the APPN network will depend on what information (if any) exists in the Directory Services database. The DISPLAY DIRECTRY command can be used to display information in the Directory Services database.
2. If SSCPORD=PRIORITY is specified and previous searches for the target resource have been performed, the search order can be modified based on the results of these previous searches. Use the DISPLAY ADJSSCPS command with the CDRSC operand to display the search order that will be used by VTAM on the next search for this resource.
3. If SORDER=APPNFRST, APPN or SUBAREA is specified for an adjacent SSCP table, any ISTAPNCP ADJCDRM entries coded within that table are ignored, because VTAM will automatically place the ISTAPNCP ADJCDRM entry in the appropriate position in the table. This is true regardless of where the SORDER value was obtained.
4. If SORDER=ADJSSCP is specified for an adjacent SSCP table and an ISTAPNCP ADJCDRM entry is not coded within that table, the APPN network will not be searched (unless SSCPORD=PRIORITY is specified and a previous search for this resource determined that it could be found in the APPN network).

Using SSEARCH to limit subarea network searches

The SSEARCH start option controls whether the subarea network is searched when a network search request is received from the APPN network. The default value (SSEARCH=YES) means that subarea network searching is allowed. In cases where both the APPN network and the subarea network provide connectivity to (or toward) the target resource, use SSEARCH=APPNFRST to force the APPN network to be used whenever possible (for example, to maximize the use of High-Performance Routing). SSEARCH=APPNFRST works like SSEARCH=YES, except that the subarea network is only searched after all APPN searching has been performed.

If APPN searches should only be propagated into the subarea network when the target resource has been defined (as a cross-domain resource) or has been previously found in the subarea network, SSEARCH=CACHE should be specified. To prevent all APPN searches received by this VTAM from entering the subarea network, use SSEARCH=NO.

Using the CDRSCTI start option to reduce broadcast searches of APPN

The setting of the CDRSCTI start option controls the amount of time that a dynamic CDRSC is retained by VTAM after its last session ends. If a dynamic CDRSC has been freed by VTAM in a combined environment, search requests originating in the subarea network or in the domain of the interchange node itself can result in unnecessary broadcasts to locate a subarea resource in the APPN network.

Therefore, to reduce broadcast searches of an APPN network that does not have a central directory server, you can increase the CDRSCTI value from the default of eight minutes to as much as seven days at each interchange node. This will minimize the potential for unnecessary broadcasts of the APPN network, because dynamic CDRSCs are retained for a longer period of time after the last session ends and VTAM has more of an opportunity to take advantage of priority routing to locate the resource in the subarea network.

Using the DISJOINT operand to define disjoint subarea networks

The DISJOINT operand on the CDRM definition statement in the cross-domain resource manager major node can also be used to control session request searches. At an interchange node, DISJOINT indicates whether an adjacent CDRM is disjoint from other SSCPs with the same NETID as the adjacent CDRM, and the only connection between the disjoint SSCPs is through the APPN network.

For example, in Figure 132 on page 522, HOSTA and HOSTB both have the same network identifier (NETA), but their only connection to each other is through adjacent network NETB. When a session request is received by HOSTC from HOSTA, HOSTC might decide to search the attached APPN network. When the search is forwarded into the APPN network, HOSTC includes information about the adjacent SSCP from which the request was received; namely, the HOSTA network identifier (NETA) and an indication of whether HOSTA is disjoint from other SSCPs within NETA (as defined by HOSTC using the DISJOINT operand in the CDRM definition statement for HOSTA).

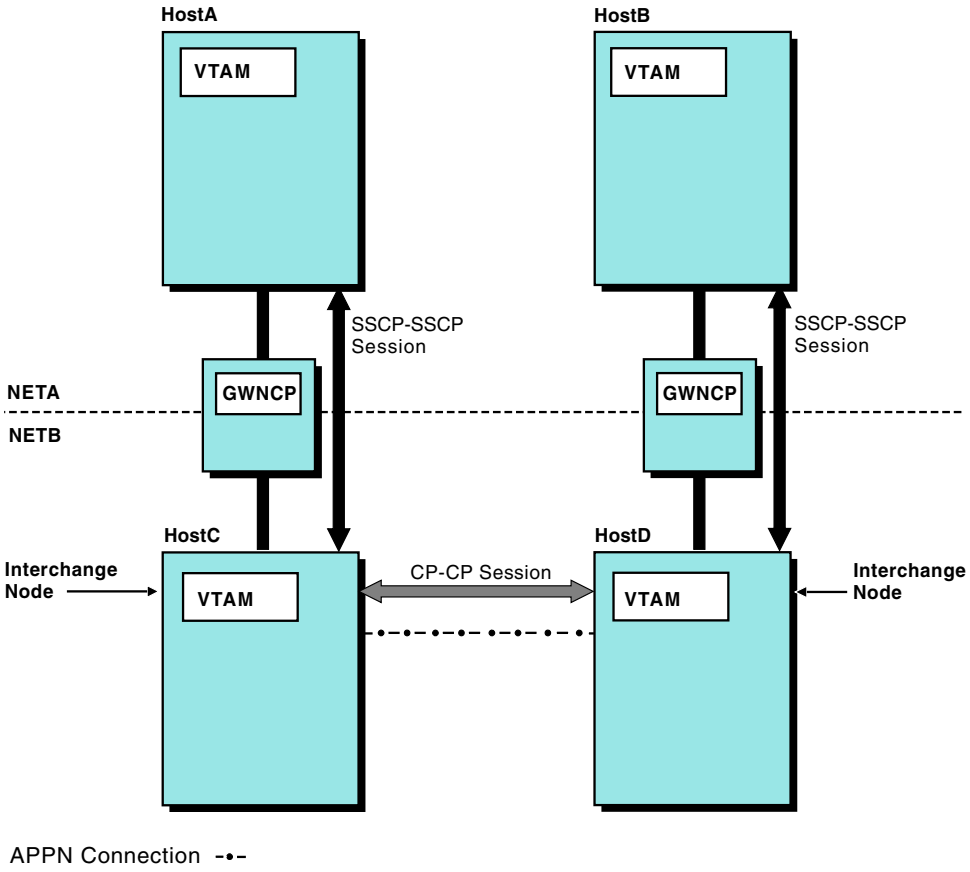


Figure 132. Disjoint hosts

When the APPN search request is received by HOSTD, HOSTD uses this information to determine the SSCPs to which the request should be routed. HOSTD routes the request to other SSCPs that are not in NETA (the network identifier provided by HOSTC, the interchange node serving as the APPN entry point).

However, when deciding whether to route the request to SSCPs within NETA (the network identifier provided by HOSTC), HOSTD uses the DISJOINT network indicator provided by HOSTC to make the decision. If DISJOINT=YES was coded in HOSTC (and provided to HOSTD on the APPN search request), HOSTD routes the search request to other SSCPs within NETA, because it is assumed that the APPN network can provide the only connectivity between the two disjoint parts of network NETA. If DISJOINT=NO was coded (or allowed to default) in HOSTC, HOSTD does not route the request to other SSCPs within NETA, because it is assumed that other connectivity exists between HOSTA and HOSTB (see Figure 133 on page 523). When HOSTD and HOSTC fail to locate the target LU, HOSTA continues to try other SSCPs and find the direct path to HOSTB.

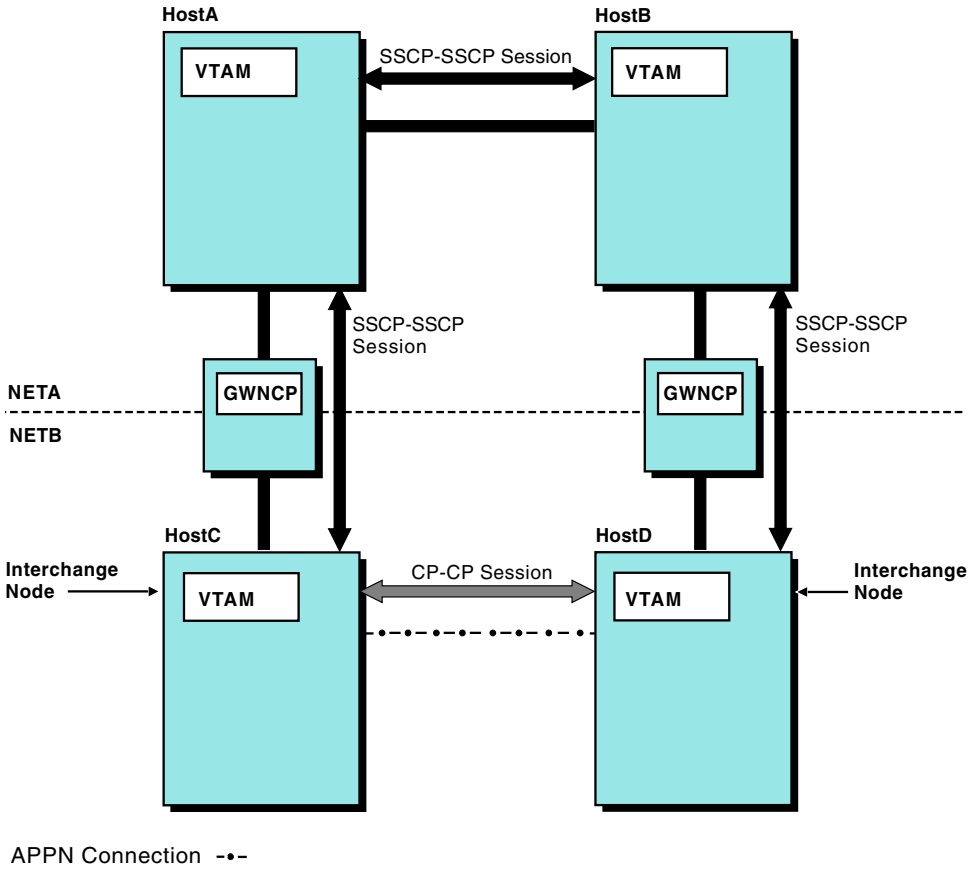


Figure 133. Hosts with subarea connection

Note: The preceding discussion of the DISJOINT operand also applies when the net IDs of all four hosts in Figure 133 and Figure 134 are the same (NETA).

APPN and subarea Class of Service resolution

In a subarea network, the LOGMODE resolves to a subarea COS at the SLU SSCP. In Figure 134, the following occurs in a subarea environment:

- For PLU-initiated sessions, the COS is resolved at VTAM2 when the CDINIT is received from VTAM1 before returning the CDINIT response.
- For SLU-initiated sessions, the COS is resolved at VTAM2 before the CDINIT is sent to VTAM1.

Figure 134. LOGMODE resolution example

In an APPN network, the LOGMODE is always resolved to an APPN COS at the CP(OLU) or NNS(OLU). Using Figure 134 on page 523, the following occurs in an APPN environment:

- For PLU-initiated sessions, the APPN COS is resolved at VTAM1.
- For SLU-initiated sessions, the APPN COS is resolved at VTAM2.

This difference between the architectures (subarea and APPN) creates new considerations in the implementation of mixed APPN and subarea networks, especially when you consider the fact that VTAM allows for different LOGMODE tables to be defined for various LUs. For example, consider a PLU-initiated session in both a subarea and an APPN network configuration.

- **In a subarea network**

  A CDINIT flows from VTAM1 to VTAM2. VTAM2 (the owner of the SLU) uses the SLU definition to resolve the LOGMODE to a subarea COS. The SLU definition can specify a specific LOGMODE table.

- **In an APPN network**

  The APPN COS must be determined before sending an APPN Locate from VTAM1 to VTAM2. VTAM1 uses the LOGMODE table associated with the SLU to resolve the LOGMODE to an APPN COS. The SLU is defined to VTAM1 as a CDRSC. If the CDRSC has been statically defined, the logon mode table to be used can be identified on the MODETAB operand. If the CDRSC has been dynamically created, the default logon mode table (ISTINCLM) is used.

If every possible SLU is predefined as a CDRSC in every VTAM that has a PLU that might wish to start a session with these SLUs, the MODETAB operand can be added to these CDRSCs to make sure the correct LOGMODE table (and, therefore, APPNCOS) is selected. However, requiring SLU definitions at every possible PLU VTAM is both undesirable and unreasonable in an APPN environment. Using dynamic CDRSCs for these SLUs is easier to maintain. Without any definition changes, the default logon mode table (ISTINCLM) is used for dynamic CDRSCs.

In addition, to ensure desired logmode to COS resolution, all SLUs need to be defined at any VTAM that is an intermediate network node on the BIND path. For example, if a composite network node existed between VTAM1 and VTAM2, the CNN would have to be able to resolve the LOGMODE name to a subarea COS so that an appropriate ER and VR can be chosen through the CNN. Because neither the PLU nor SLU has been predefined to the CNN, dynamic CDRSCs are created for both and the SLU would inherit ISTINCLM as it logon mode table.

To ensure that the desired COS is being used when LU-LU session requests use both APPN links and subarea VRs, there are several choices available for specifying the appropriate APPN and subarea COS for a given logon mode name. The following choices are provided in order of the preference used by VTAM for COS selection.

1. You can define APPN-to-subarea (APPNTOSA) and subarea-to-APPN (SATOAPPN) Class of Service mapping tables to define the mappings between APPN and subarea Classes of Service.
2. You can modify your customer-defined logon mode tables to specify the appropriate APPN and subarea Class of Service names for each logon mode table entry. (You can also associate one of your customer-defined logon mode tables with all dynamic CDRSCs using the DYNMODTB start option.)
3. You can modify the IBM-supplied default logon mode table, ISTINCLM, to specify (or change) the APPN and subarea Class of Service names for the

IBM-supplied logon modes. (This might be required if you allow VTAM to create dynamic CDRSCs and you have not specified a logon mode table name on the DYNMODTB start option.)

Resolving logon mode names to subarea and APPN Classes of Service

VTAM uses the logon mode table for the secondary LU (SLU) to initially resolve the logon mode name to a subarea COS. This happens at the VTAM that owns the SLU or at the VTAM node that must send a session request into the APPN network.

For sessions that will use APPN links, one of two methods is used to determine the APPN COS for the session. If a subarea-to-APPN (SATOAPPN) Class of Service mapping table has been defined at the VTAM that must choose the APPN Class of Service, it will be used to map the resolved subarea COS to an appropriate APPN COS. If a SATOAPPN Class of Service mapping table has not been defined, or a table is found with no matching entry, VTAM uses the logon mode table for the SLU (as defined in that VTAM node) to resolve the logon mode name to an appropriate APPN COS.

The same is true for session requests (or replies) received by VTAM from an APPN network that will also use subarea virtual routes. One of two methods is used to determine the appropriate subarea COS for the session. If an APPN-to-subarea (APPNTOSA) Class of Service mapping table is defined at the VTAM that must choose the subarea COS, it will be used to map the APPN COS received on the APPN session request (or reply) to an appropriate subarea COS. If an APPNTOSA Class of Service mapping table is not defined, or a table is found with no matching entry, VTAM uses the logon mode table for the SLU (as defined in that VTAM node) to resolve the logon mode name to an appropriate subarea COS.

Note: VTAM always uses the logon mode table for the secondary LU (SLU) to resolve the logon mode name to an APPN or subarea Class of Service, when needed. But at interchange nodes and for other node roles, the SLU might be a dynamic CDRSC. In this case, the IBM-supplied default logon mode table (ISTINCLM) would normally be used. If you want to have a different table used for these resources, you can use the DYNMODTB start option to specify a different logon mode table to be used for all dynamically created CDRSCs.

Resolving the logon mode name to session parameters and subarea COS

Session parameters and subarea Class of Service are determined in the SSCP that is closest to the secondary logical unit (SLU). This includes cases where this VTAM does the following:

- Owns the SLU (the SLU is an application or a dependent LU in this VTAM domain), like HostD owns LUD in Figure 135 on page 526
- Owns the boundary function in the direction of an LEN-attached independent SLU, like HostD does for LUD in Figure 135 on page 526
- Owns the boundary function in the direction of an APPN-attached independent SLU, like HostA does for LUD in Figure 135 on page 526. This VTAM might:
  - Own the primary logical unit (PLU), and the SLU is elsewhere in (or through) the APPN network.
  - Be an interchange node, with the PLU in (or through) the subarea network, and the SLU elsewhere in (or through) the APPN network.

– Be part of a composite network node that is an intermediate node on the BIND path of the session. In this case, the Class of Service is needed to determine the subarea route through the composite network node.

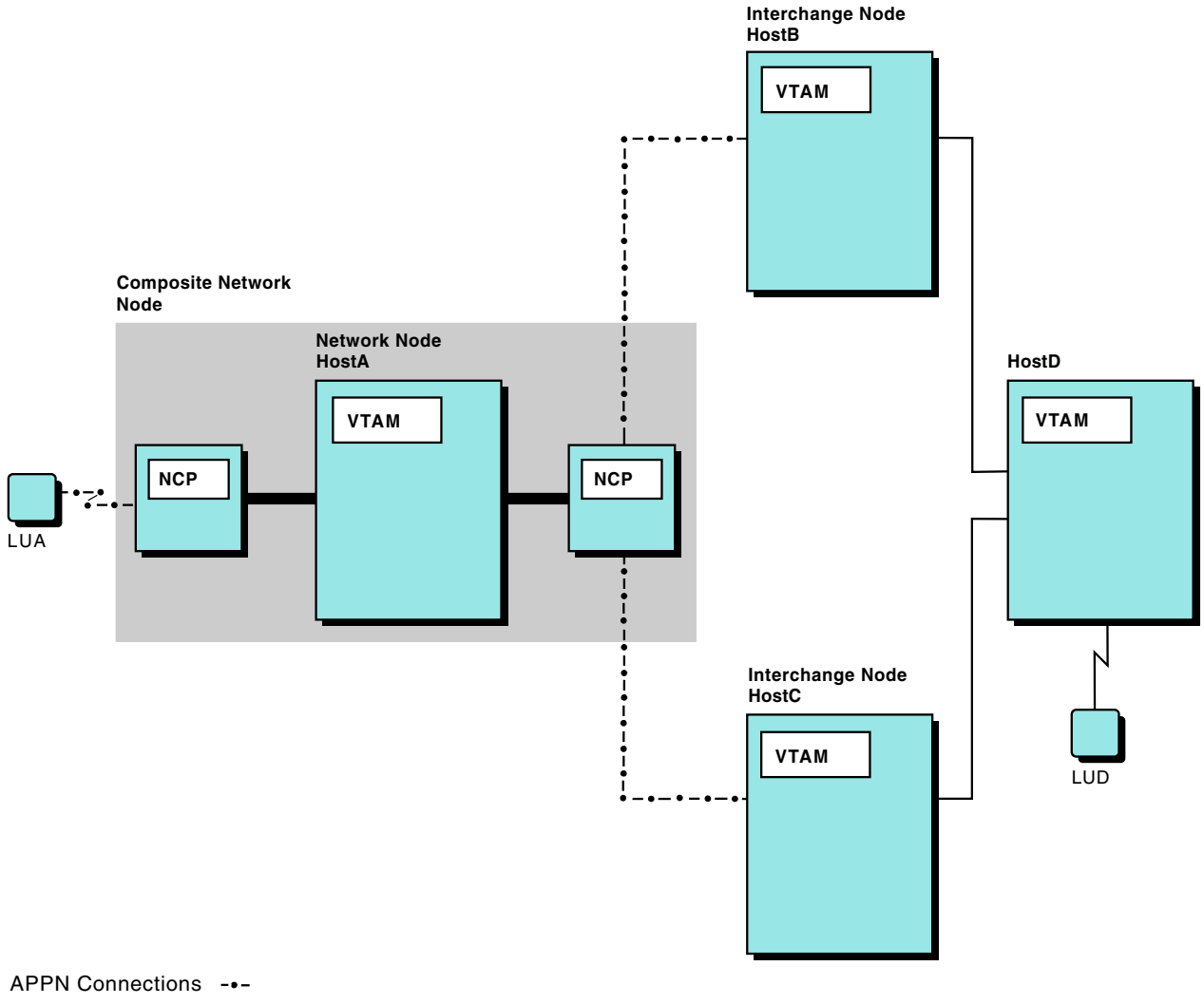


Figure 135. Class of Service resolution at multiple nodes

Resolving the logon mode name to an APPN Class of Service

The APPN Class of Service is determined in the APPN control point (CP), closest to the originating logical unit (OLU) for the session, that has the ability to perform the determination. For VTAM, this includes cases where this VTAM:

- Has the OLU in its domain and the destination logical unit (DLU) is elsewhere in (or through) the APPN network, like HostA in Figure 135
- Is the network node server for an end node (EN) that owns the OLU but is not capable of determining the APPN Class of Service itself, and the DLU is owned by this network node server or is elsewhere in (or through) the APPN network (like HostA in Figure 135)
- Is an interchange node, with the OLU in (or through) the subarea network and the DLU elsewhere in (or through) the APPN network, like HostB or HostC in Figure 135

Defining APPN and subarea Classes of Service in logon mode tables

For each logon mode table associated with LUs that might be involved in a session with both APPN links and subarea virtual routes, specify an APPN Class of Service by adding the APPNCOS operand and a subarea Class of Service using the COS operand on the appropriate MODEENT macroinstructions. The subarea virtual routes use the Class of Service name coded on the COS operand, and APPN links use the Class of Service name coded on the APPNCOS operand. Using both the COS and APPNCOS operands enables you to use two different Class of Service names to represent the same Class of Service within the two subnetworks. This can be very useful as you migrate to APPN.

If neither the COS operand nor the APPNCOS operand is coded, the default Class of Service names (the unnamed Class of Service entry for subarea and the #CONNECT entry for APPN) are used.

If the COS operand is coded and the APPNCOS operand is not coded, the Class of Service name specified on the COS operand is used for both the subarea and APPN sessions. Therefore, when you want to use the same Class of Service name to represent identical Classes of Service in both the APPN and subarea networks, you do not need to code the APPNCOS operand when the COS operand is already coded. However, make sure that a corresponding APPN Class of Service definition exists in all network nodes. It is recommended that the architected standard APPN COS names be used.

If you code the APPNCOS operand and the COS operand is not coded, the unnamed Class of Service is used for the subarea session, and the Class of Service designated on the APPNCOS operand is used for the APPN session.

Defining APPNTOSA and SATOAPPN class of service mapping tables

Rather than resolving the logon mode name to a subarea or APPN class of service at each APPN or subarea network boundary, you can choose to map subarea classes of service to APPN classes of service, and vice versa. This is done by coding a subarea-to-APPN COS mapping table (SATOAPPN) and an APPN-to-subarea (APPNTOSA) COS mapping table. These tables are used at any VTAM that must choose an APPN or subarea Class of Service when the corresponding Class of Service (subarea or APPN) is already known.

Note: If subarea-to-APPN or APPN-to-subarea COS mapping tables are defined, these tables are used instead of resolving the logon mode name to an appropriate APPN or subarea COS using the logon mode table for the secondary LU. If COS mapping tables are not defined, or a matching entry is not found, the logon mode table for the SLU is used to resolve the logon mode name to an appropriate APPN or subarea Class of Service.

Adding and moving nodes

To add or move workstations or applications in an APPN network, no coordinated LU definitions are required. As workstations and applications first become available or are later moved, APPN directory functions can locate them dynamically. If new hosts are added, only the resources on that host must be defined to that host.

If an APPN network node is to be moved from one APPN subnetwork to another, purge its topology database before activating CP-CP sessions in the new APPN subnetwork. This prevents broadcasting of network topology from the original subnetwork to the new subnetwork, information that is of no use in the new subnetwork. For a description of APPN subnetworks, see "APPN multiple network connectivity" on page 89.

No definitions are necessary in the subarea network for resources added or moved in the APPN network. APPN directory functions can locate them dynamically.

Chapter 19. Implementing a subarea network

This topic contains information needed when more than one VTAM subarea domain exists in the network. Chapter 3, “Implementing a VTAM network,” on page 25 contained basic concepts and issues requiring consideration for each VTAM domain. This topic expands on Chapter 3, “Implementing a VTAM network,” on page 25 to include multiple domain issues.

A multiple-domain subarea network contains more than one system services control point (SSCP). Control of the resources in the network is divided among the SSCPs. Defining your VTAM network involves identifying your domains and defining the resources in those domains to the SSCPs.

Note: A VTAM host contains a host CP, even if you have defined your VTAM as a subarea node and not as an APPN node. This host CP is used for management services transport and is not used for session setup or routing.

Figure 136 is the same as the configuration in Figure 4 on page 27, except that this is a multiple-domain network because it includes another host. Note that NCP12 is now NCP21. This is for naming convention purposes.

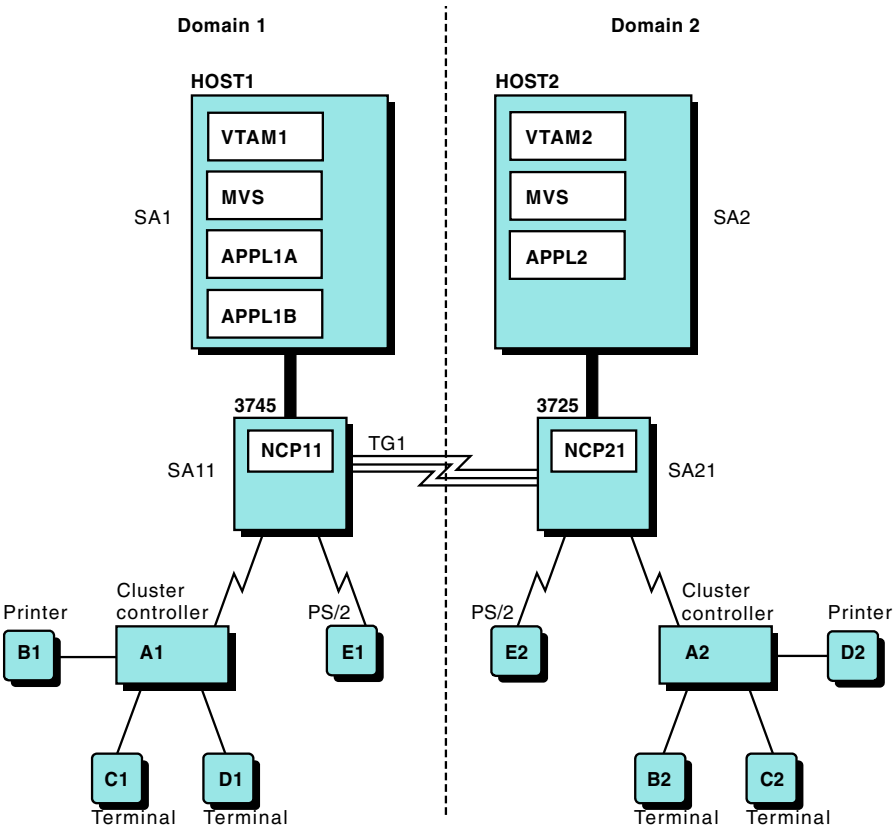


Figure 136. Multiple-domain network

© Copyright IBM Corp. 1984, 2011

Code everything that you would code to define the network in Figure 4 on page 27, except that you need to put the definitions for NCP21 and its peripheral nodes in HOST2 rather than in HOST1. You also need to code at least the following in HOST2:

- Start options describing HOST2 (NETID, HOSTSA, SSCPID, and SSCPNAME).
- One application program minor node defining APPL2.
- One NCP major node defining NCP21. This includes the NCP definitions. In NCP21 major node, also code another PATH definition statement to attach to subarea 2, and change the PATH definition statement for NCP21 in NCP11 major node to DESTSA=21.
- VTAM PATH definition statements defining the explicit and virtual routes used for data flow throughout the network.
- One cross-domain resource manager major node containing two CDRM statements (one for HOST1 and one for HOST2).
- One cross-domain resource major node (or indicate the use of dynamic definition of cross-domain resources in the CDRM major node).
- One adjacent SSCP table (or indicate dynamic definition of adjacent SSCPs using start options [SSCPDYN and DYNASSCP]).

Also code at least the following in HOST1:

- The HOSTSA start option
- A cross-domain resource manager major node containing two CDRM statements (one for HOST1 and one for HOST2)
- A cross-domain resource major node (or indicate use of dynamic definition of cross-domain resources in the CDRM major node)
- An adjacent SSCP table (or indicate dynamic definition of adjacent SSCPs using start options [SSCPDYN and DYNASSCP])

Start options defining other domains

In a multiple-domain subarea network, you can define the location of cross-domain resource managers and also let VTAM define cross-domain resource managers dynamically.

Defining the location of cross-domain resource managers by coding adjacent SSCP tables

In a multiple-domain network, session setup involves VTAM locating resources (session partners) in the network. Resources are located by routing a session initiation request to VTAMs in other domains. You can define a list of cross-domain resource managers (CDRMs) to which a session setup request can be routed. The list of CDRMs that you code for the search (referred to as adjacent SSCPs) can be supplemented dynamically by your VTAM domain. The search order, as well as the number of VTAMs to which the session request is routed, can affect session setup time. For more information, see “Adjacent SSCPs” on page 541.

VTAM provides two start options, SSCPDYN and SSCPORD, that control how VTAM processes adjacent SSCP tables. SSCPORD can be specified as a start option or coded on individual ADJSSCP table definitions (or both).

SSCPDYN start option
SSCPDYN determines whether VTAM should add entries dynamically to the adjacent SSCP table (that is, supplement the list of adjacent SSCPs that

you have coded with other SSCPs that it discovers dynamically). The addition of these entries to the list can increase the time needed to search for a cross-domain resource.

If you code SSCPDYN=NO, you might have to define additional adjacent SSCP tables. When an application program passes a session (issues a CLSDST PASS macro instruction) during session setup, VTAM does not build an adjacent SSCP table in the application program network. When SSCPDYN=YES is specified in the application program network, VTAM builds an entry in the table automatically.

SSCPORD start option

SSCPORD determines the order in which VTAM processes the entries in the table to route session requests to adjacent SSCPs. SSCPORD can be specified as a start option, modified using MODIFY VTAMOPTS, or coded separately for each ADJSSCP table (or all three). If SSCPORD is not coded for a given ADJSSCP table, the current value of the SSCPORD start option value is used each time that ADJSSCP table is selected for use. SSCPORD indicates whether VTAM is to process the list in the order in which it is defined (DEFINED) or whether it is to route the session request to the last successful entry (PRIORITY). VTAM remembers successful and unsuccessful adjacent SSCP entries for routing session initiation requests. If you specify SSCPORD=PRIORITY, VTAM always uses the last successful adjacent SSCP entry to route the session setup request.

One exception to order of search, regardless of the SSCPORD value, is that VTAM always routes the session-initiation request directly to the VTAM that owns the resource if it has an SSCP-SSCP session with that owner.

Defining the location of cross-domain resource managers dynamically

If you use the start option DYNASSCP=YES, you do not have to code an adjacent SSCP table. VTAM dynamically builds the list in the order in which the SSCP-SSCP sessions are activated. VTAM then uses this dynamically constructed list to route session initiation requests to the CDRMs, if an appropriate list is not defined.

Specifying timeout values for locating cross-domain resources

The following start options also affect VTAM processing in a multiple-domain subarea network:

CDRSCTI start option

In a multiple-domain environment, you define resources that are controlled by a VTAM in another domain using a cross-domain resource (CDRSC) definition statement. However, if you do not code the CDRSC definition statement, you can specify that a CDRSC definition be dynamically created by using the CDRSC=OPT operand on the cross-domain resource manager (CDRM) definition statement that defines that VTAM. If your VTAM dynamically defines a cross-domain resource, that dynamic definition can be retained by VTAM for a user-specified period of time or deleted immediately after the last session for that cross-domain resource terminates. The cross-domain resource timeout value (CDRSCTI) start option determines the processing VTAM uses for dynamic cross-domain resources. The value of this start option specifies the amount of time that VTAM is to retain the control blocks for a dynamic cross-domain resource

after all sessions with that resource are terminated. For more information, see “Dynamic definition of cross-domain resources” on page 534.

IOPURGE start option

During cross-domain session initiation, a CDINIT is sent to the adjacent node that is most likely to be the resource owner. If a positive response is received, session setup continues with that adjacent SSCP. If a negative response is received indicating the resource was not found, the next adjacent SSCP that potentially owns the resource is obtained from the adjacent SSCP table and a CDINIT is sent to it. This continues until the resource is either found or the adjacent SSCP table is exhausted.

A situation could arise where the adjacent SSCP that was queried has a critical failure and cannot respond. In this case, the node that is trying to initiate the session waits forever for a reply (IOPURGE=0, the default), possibly blocking any future attempt to find the resource. Because the failing SSCP may not be the resource owner, the resource may be available and could be found if the search was allowed to continue through the adjacent SSCP table.

The IOPURGE start option can be used to specify how long VTAM will wait for a response before continuing a search. After a specified time interval, if a response has not been received, VTAM continues its search through the adjacent SSCP table until it either finds the resource or the table has been exhausted.

Identifying VTAMs in other domains (CDRMs)

A cross-domain resource manager (CDRM) is the part of an SSCP that supports cross-domain session setup and takedown. Before logical units in one domain can have cross-domain sessions with logical units in another domain, an SSCP-SSCP session must be established between the SSCPs of the two domains.

For an SSCP-SSCP session to exist, VTAM must know about all cross-domain resource managers with which it will communicate. You must define to VTAM its own cross-domain resource manager and all other cross-domain resource managers in the network. The cross-domain resource manager that represents the SSCP in your domain is called the host cross-domain resource manager. The cross-domain resource managers that represent the SSCPs in other domains are called external cross-domain resource managers.

Thus, to have a session between two SSCPs, define two cross-domain resource managers to each VTAM: one for the host and one for the external cross-domain resource manager. You file these definitions in a CDRM major node. Each cross-domain resource manager is a minor node.

One or more major nodes can be used to define cross domain resource managers. Each CDRM major node is defined with a VBUILD definition statement, and each minor node is defined with a CDRM definition statement.

Figure 137 on page 533 shows two types of major and minor nodes in a multiple-domain network.

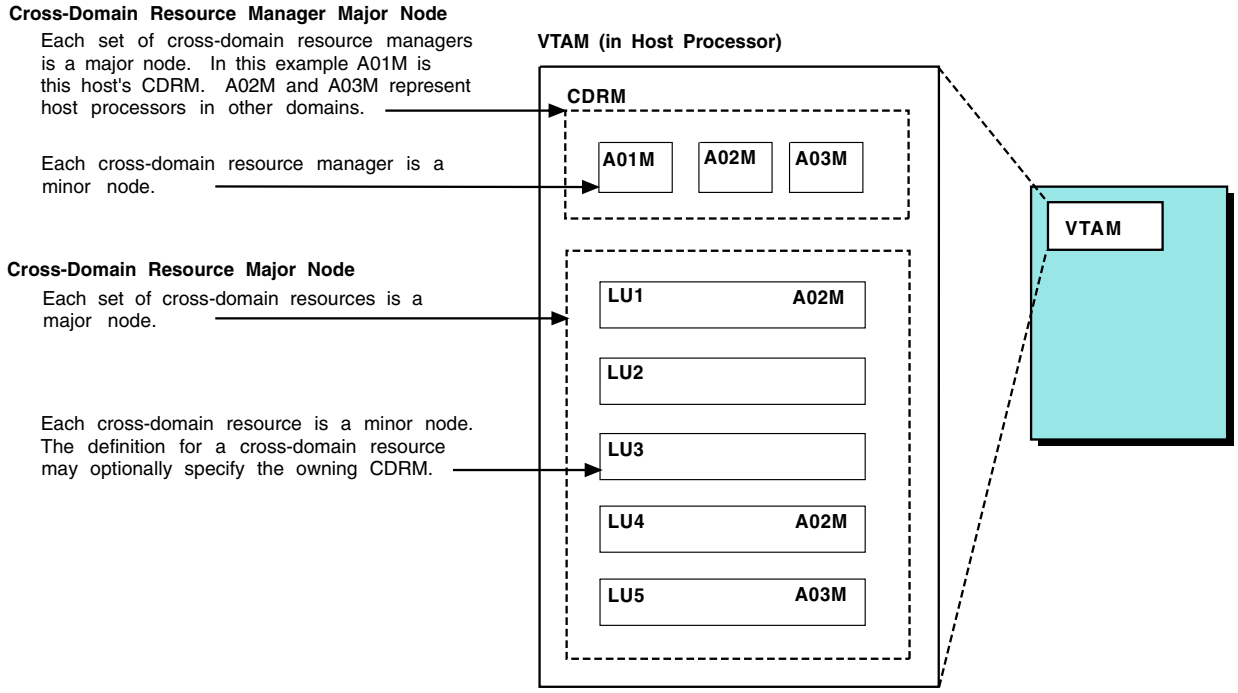


Figure 137. Major and minor nodes in multiple-domain environment

Following is an example of major node CDRM for host A01M. The same definition set can be used at A02M and A03M. In this case, the definition set is at A01M. Thus, A01M is the host cross-domain resource manager and A02M and A03M are external cross-domain resource managers.

```
        VBUILD TYPE=CDRM
A01M    CDRM  CDRDYN=YES,        AUTHORIZE DYNAMIC CDRSC
              CDRSC=OPT,         ALLOW DYNAMIC CDRSCs
              ELEMENT=1,         CDRM ELEMENT NUMBER
              ISTATUS=ACTIVE,    VTAM INITIAL STATUS
              SUBAREA=1,         NETWORK UNIQUE SUBAREA ADDRESS
              VPACING=63         PACING BETWEEN CDRMS
*
A02M    CDRM  CDRDYN=YES,        AUTHORIZE DYNAMIC CDRSC
              CDRSC=OPT,         ALLOW DYNAMIC CDRSCs
              ELEMENT=1,         CDRM ELEMENT ADDRESS
              ISTATUS=ACTIVE,    VTAM INITIAL STATUS
              SUBAREA=2,         NETWORK UNIQUE SUBAREA ADDRESS
              VPACING=63         PACING BETWEEN CDRMS
*
A03M    CDRM  CDRDYN=NO,         DYNAMIC CDRSC NOT AUTHORIZED
              CDRSC=REQ,         REQUIRE PREDEFINED CDRSCs
              ELEMENT=1,         CDRM ELEMENT ADDRESS
              ISTATUS=ACTIVE,    VTAM INITIAL STATUS
              SUBAREA=3,         NETWORK UNIQUE SUBAREA ADDRESS
              VPACING=63         PACING BETWEEN CDRMS
```

Each CDRM minor node definition includes its network address and its initial status (active by default, or inactive). CDRM minor nodes are assigned to the subarea address of the appropriate host.

You can use the same set of CDRM definition statements for the entire network. Each VTAM uses the subarea specified on the CDRM definition statements to determine which definition statement defines its own host cross-domain resource manager. If the subarea on a CDRM definition statement matches the host subarea,

VTAM uses the operands on this CDRM definition statement that apply to the host cross-domain resource managers and ignores operands that apply to external cross-domain resource managers. If the subarea on a CDRM definition statement does not match its host subarea, VTAM assumes that this CDRM definition statement applies to an external cross-domain resource manager. VTAM then uses only the operands that apply to external cross-domain resource managers.

When defining the host cross-domain resource manager, you can specify whether the host cross-domain resource manager is authorized to dynamically define cross-domain resources. If the host cross-domain resource manager and an external cross-domain resource manager are authorized in a VTAM domain, the other-domain SSCP can request a session setup for a logical unit for which no cross-domain resource (CDRSC) was defined. The receiving SSCP handles the request by dynamically creating a temporary CDRSC definition for the logical unit. Using dynamic definition of cross-domain resources can greatly reduce the number of definitions required in a multiple-domain network.

The default for the CDRDYN start option is YES, enabling dynamic definition of cross-domain resources. The CDRDYN operand can be coded on the host CDRM definition statement, but the CDRDYN start option overrides it. Also code CDRSC=OPT on external CDRM definition statements whose resources you want VTAM to dynamically define. The CDRSC operand is meaningful only on CDRM definition statements for external CDRMs.

The preceding example, which illustrates a CDRM major node, defines CDRMs that use different dynamic CDRSC options. The resulting capabilities of and restrictions on dynamic cross-domain resource definition are as follows:

- A01M can initiate session requests for any undefined cross-domain resources; however, it can route such requests only to A02M because that is the only other cross-domain resource manager with CDRSC=OPT. A01M can accept session requests from undefined cross-domain resources owned only by A02M because that is the only other cross-domain resource manager with CDRSC=OPT.
- A02M can initiate session requests for any undefined cross-domain resources; however, it can route such requests only to A01M. A02M can accept session requests from undefined cross-domain resources owned by A01M only.
- A03M cannot initiate or accept any session requests for undefined cross-domain resources.

Identifying resources in other domains

This section describes the following:
- Dynamic definition of cross-domain resources
- Static definition of cross-domain resources
- Model definition of cross-domain resources

Resources controlled by a VTAM in another domain are called cross-domain resources (CDRSCs).

Note: CDRSC definition statements are also used to define independent logical units in the same domain and in nonnative networks. See "Independent LUs" on page 273.

Dynamic definition of cross-domain resources

You do not have to define resources controlled by VTAMs in other domains. VTAM can dynamically create the definition statements to represent resources that

reside in other domains. The operands on the CDRM definition statement for the host CDRM and the external CDRM determine whether VTAM can dynamically create CDRSC definition statements.

Notes:

1. You can also dynamically define independent logical units within the same domain using this function, and independent logical units in nonnative networks. See "Dynamic definition of independent LUs" on page 276.
2. To define two or more cross-domain destination LUs that have the same name but are in different networks, use the NQNMODE=NQNAME start option. Otherwise, name conflicts cause session requests involving these LUs to fail.

To have resources in other domains dynamically defined to VTAM:
1. Code your host CDRM with CDRDYN=YES.
2. Code your external CDRM with CDRSC=OPT.

Now all unidentified resources residing under that external CDRM will be dynamically defined to your VTAM (host CDRM). Following is a sample coding for dynamic definition of resources in another domain:

```
         VBUILD TYPE=CDRM
MYHOST   CDRM  CDRDYN=YES,                                    X
               SUBAREA=1
HOST2    CDRM  CDRSC=OPT,                                     X
               SUBAREA=2
```

Dynamic definition can occur in the following two directions:

- When VTAM receives a session request from an undefined cross-domain resource, the cross-domain resource is the originating logical unit for this session. For example, a terminal in another domain attempts to log on to an application in your domain. The external cross-domain resource manager initiates the session, sending a session request to your VTAM. If the terminal is not defined to your VTAM as a cross-domain resource, VTAM does not accept the session request unless the host CDRM definition is coded with CDRDYN=YES and the external CDRM is coded in your VTAM with CDRSC=OPT.
- When VTAM sends a session request for an undefined cross-domain resource, the cross-domain resource is the destination logical unit for this session. For example, a local terminal attempts to log on to an application in another domain. If the application is not defined as a cross-domain resource, VTAM can still attempt to establish the session by sending a session request to adjacent SSCPs. VTAM sends the session request only if the host CDRM is coded with CDRDYN=YES and the adjacent CDRMs to which VTAM routes the session request are coded with CDRSC=OPT.

When VTAM creates dynamic CDRSCs for destination logical units, it must then dynamically locate the domain in which the resource resides. For this type of dynamic cross-domain resource, VTAM uses a search mechanism to route a cross-domain session request to other adjacent VTAMs. The mechanism is called an adjacent SSCP table and is discussed in "Static definition of cross-domain resources" on page 536.

Dynamically defined cross-domain resources are collected in a CDRSC major node named ISTCDRDY. ISTCDRDY is activated automatically during VTAM initialization, and deactivated automatically during VTAM termination.

The dynamic cross-domain resource definition occurs when you have defined the host to allow dynamic definition of CDRSCs.

In general, you have the same control of dynamically defined cross-domain resources as predefined cross-domain resources. You can display and deactivate dynamically defined cross-domain resources using operator commands.

You also generally have the same control of the CDRSC major node ISTCDRDY as other CDRSC major nodes. The DISPLAY ID, MODIFY CDRM, VARY ACT, and VARY INACT commands can all be used for the ISTCDRDY major node.

While the host CDRM is active, the operator can deactivate ISTCDRDY with the VARY INACT command, in which case all dynamically defined CDRSCs are also deactivated and the dynamic CDRSC definition function is disabled. The dynamic CDRSC definition function remains disabled until ISTCDRDY is activated again. You can activate ISTCDRDY again by issuing a VARY ACT command naming ISTCDRDY directly, or by activating the host CDRM (even if the host CDRM is already active). If the host CDRM is inactive, or if the host CDRM is not defined to allow the dynamic CDRSC definition function, activating ISTCDRDY does not provide the dynamic CDRSC definition function.

Dynamically defined CDRSCs are deactivated and deleted by VTAM on a periodic basis if they are not in use, based on the setting of the timer specified in the CDRSCTI start option. A dynamically defined CDRSC is also deleted if the host that owns that resource fails and the shadow resource for the CDRSC becomes the active definition. For more information on shadow resources, see “Shadow resources” on page 549.

Static definition of cross-domain resources

Cross-domain resources are logical units (application programs, peripheral nodes, and terminals) that are controlled by another VTAM domain. You define cross-domain resources in one or more cross-domain resource major nodes. Each cross-domain resource represents a minor node. You can also define model cross-domain resources in one or more cross-domain resource major nodes. Each model cross-domain resource represents a potential group of similarly named cross-domain resource minor nodes. See “Model definition of cross-domain resources” on page 538 for more information.

You define a cross-domain resource major node with one VBUILD definition statement for the major node, one CDRSC definition statement for each cross-domain resource in the major node, and one CDRSC definition statement for each model cross-domain resource in the major node. The same set of CDRSC definition statements can be used throughout the network.

The name on the CDRSC definition statement represents the name for the resource that is controlled by another VTAM. The CDRM operand, if used, specifies the name of the VTAM that controls that resource. If you do not specify the CDRM operand, the name of the other VTAM that controls the resource is not known. Therefore, you need to use an adjacent SSCP table, so that your VTAM can locate the CDRSC in another VTAM domain. An adjacent SSCP table is simply a list of other VTAMs with which your VTAM can have an SSCP-SSCP session. This list of VTAMs is used to search your network for the CDRSC. For more information about adjacent SSCP tables, see “Adjacent SSCPs” on page 541.

To statically define two or more CDRSCs that have the same name but are in different networks, use the NQNMODE=NQNAME start option to allow network-qualified names. For more information, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

In the following example of a CDRSC major node, some of the CDRSC definition statements do not specify a CDRM operand, so an adjacent SSCP table is required. However, some of the other cross-domain resource definition statements identify the CDRM owner.

```
*
* EXTERNAL VTAM APPLICATIONS — CDRSCS
*
         VBUILD  TYPE=CDRSC          CDRSC MAJOR NODE
*
* EXTERNAL VTAM APPLICATIONS — CDRSCS FOR VTAM SUBAREA NODE 03
*
NJE03    CDRSC                       JES/NJE
NETV03   CDRSC                       NETVIEW
CICS03   CDRSC                       CICS
NVAS03   CDRSC                       NETVIEW ACCESS SERVICES
*
* EXTERNAL VTAM APPLICATIONS — CDRSCS FOR VTAM SUBAREA NODE 70
*
S04VSCS  CDRSC  CDRM=A04M             VSCS
S04RSCS  CDRSC  CDRM=A04M             RSCS
```

Note: Information on coding the CDRSC definition statement for TSO/VTAM is in "Multiple-domain network" on page 716.

A dependent logical unit and a CDRSC with the same network-qualified name can coexist. In a backup and recovery situation where one host is assuming ownership of a logical unit from another host in the same network, the dependent logical unit can be activated by the new host even though an application program within it currently has a cross-domain session with the logical unit. If the physical and logical units being recovered support ACTPU(ERP) and ACTLU(ERP) requests, the sessions with the logical units are not affected when the physical units and logical units are activated. The CDRSC definition automatically becomes a shadow resource, and the logical unit is now defined as a same-domain (APPL or LU) resource. If the current host wants to relinquish ownership of the logical unit, it releases or deactivates the logical unit and its associated physical unit. VTAM then makes the logical unit a shadow resource, and the CDRSC again becomes the active definition.

Note: For MNPS applications that are active on a network node, you cannot activate a CDRSC definition with the same name. You can activate the CDRSC *before* the MNPS application issuing OPEN ACB, and the CDRSC will become a shadow resource at that time.

When you define your network resources for XRF, consider the following:

- If you have predefined all CDRSCs in the network, you need to predefine a CDRSC having the USERVAR name and predefine a CDRSC for both the active and the backup application program.
- If you allow dynamic CDRSCs and trial-and-error rerouting, you do not need to do anything.

Regardless of whether you are using USERVARs with XRF, the names of your USERVARs must be unique. To ensure proper session setup with the desired

partner, there can be no other resource within the network that has the same name as any of your USERVARs (except when the USERVAR is identical to its value).

Model definition of cross-domain resources

An alternative method of defining cross-domain resources is with a model CDRSC definition, which can be used as the definition for one or more CDRSCs. You code a model CDRSC definition by placing wildcard characters in the name field of a CDRSC definition statement that defines characteristics for one or more CDRSCs.

If VTAM cannot find an existing definition for a CDRSC, it searches a list of model CDRSC definitions to find the pattern that best matches the name of the cross-domain resource. If VTAM finds a match, it uses that model definition statement to build the definition for the cross-domain resource. If it does not find a match, it attempts to create a dynamic CDRSC, if dynamics are allowed. Model CDRSC definitions enable you to reduce the number of CDRSC definitions you must code in VTAMLST, when dynamic CDRSCs are not an option or are otherwise undesirable.

Model CDRSC definitions can have any keyword specified on them that can be specified on a static CDRSC, which gives you the ability to have different sets of CDRSCs with different characteristics (for example, you might have one model CDRSC coded with DLOGMOD=BATCH and another one coded with DLOGMOD=INTERACT). A CDRSC that is built from a model CDRSC definition is called a clone CDRSC. The clone CDRSC is added to the same CDRSC major node in which the model CDRSC was defined and the clone CDRSC has all the characteristics of the model CDRSC. It is treated as a predefined CDRSC in that any operation that can be done on a predefined CDRSC can be done on a clone CDRSC.

Note: Any discussion in this publication related to a predefined CDRSC can be assumed to include building a clone CDRSC from an appropriate predefined model CDRSC.

The building of a clone CDRSC is not dependent on the coding of CDRDYN (as a start option or on a CDRM definition statement), nor is it subject to the coding of the CDRSC parameter on the CDRM definition statement. The activation or updating of a model CDRSC does not have any effect on any existing clone CDRSC.

For additional information on modeling cross-domain resources, refer to Model definition of VTAM cross-domain resources in *z/OS Communications Server: SNA Resource Definition Samples*.

Coding guidelines

To code a model CDRSC definition, code a CDRSC definition statement to define CDRSC characteristics that you expect to be used by one or more CDRSCs. Use wildcard characters in the name field of the CDRSC definition statement. You can use the following characters:

Asterisk (*)
Represents 0 or more unspecified characters.

Question mark (?)
Represents a single unspecified character.

When placing wildcard characters in the name fields of model CDRSC definitions, you should have some idea of which clone CDRSCs might be built from those model definitions. Use a naming scheme that ensures that clone CDRSCs are not

accidentally built from model CDRSC definitions; that is, ensure that a cross-domain resource does not use a model definition that you did not intend for it to use.

A model CDRSC name, including wildcard characters, can be a maximum of eight characters in length. A question mark (?) can be used anywhere in the model CDRSC name. An asterisk (*) can be used in the second to eighth characters of the model CDRSC name. Model CDRSC names must be unique across all resource names known to this VTAM (including model APPL definitions). There is no defined limit on the number of clone CDRSCs that can be created from one model CDRSC definition, nor is there a defined limit on the number of model CDRSC definitions that you can define.

Model CDRSC definitions can be defined in any number of CDRSC major nodes. Those model CDRSC definitions can appear in a CDRSC major node along with conventionally defined CDRSC definition statements. The model definitions and conventionally defined CDRSC definition statements can appear in any order. If the model is coded prior to a NETWORK statement with NETID= coded with a valid value, it will be an alias CDRSC and any clones created from it will be considered as predefined alias CDRSCs. If the model is coded after a NETWORK statement with NETID= coded with a valid value, it will be a real CDRSC and any clones created from it will be considered as predefined real CDRSCs.

The rules for the model CDRSC name and how the best matching pattern is found are similar to the rules for model applications. See “How VTAM finds the best match” on page 377 for examples of model names and how matches are found. Table 40 on page 376 gives samples for model names and matching names that apply to CDRSCs as well as applications.

In addition to the rules for finding the model CDRSC with the best matching pattern for the name, other factors are taken into account when selecting the model CDRSC. These factors are network ID, NQNMODE, and whether a real or alias CDRSC is being requested.

Guidelines:

If the input netid for this clone CDRSC request is not the local host netid, but there is another RDTE known by the local host netid, the following apply:

- If an alias CDRSC is being requested, no model is eligible because a clone CDRSC would collide with this RDTE. Therefore, only a dynamic CDRSC is allowed.
- If a real CDRSC is being requested, a model is eligible if the following are true:
  - The model is a real CDRSC.
  - The model's netid matches the input netid.
  - The model has NQNMODE=NQNAME (whether by start option or coded on the model CDRSC).

If the input netid for this clone CDRSC request is the local host netid or if there is no other RDTE known by the local host netid and the input netid for this request is not the local host netid, the following apply:

- If an alias CDRSC is being requested, a model is eligible if one of the following is true:
  - The model is an alias CDRSC.
  - The model's netid is the local host netid.

  - The model has NQNMODE=NAME (whether by start option or coded on the model CDRSC).
- If a real CDRSC is being requested, a model is eligible if the model is an alias CDRSC or if the model's netid matches the input netid.

An example of a CDRSC major node with model CDRSCs follows:

```
CDRSCSEG   VBUILD     TYPE=CDRSC,...
APPL*      CDRSC      CDRM=HOST1,...
           NETWORK    NETID=NETB
TERM1      CDRSC      CDRM=HOST2,...
APPLB*     CDRSC      CDRM=HOST2,DLOGMOD=BATCH,...
APPLB1*    CDRSC      CDRM=HOST2,DLOGMOD=INTERACT,...
           NETWORK    NETID=NETC
APPLC?     CDRSC      CDRM=HOST3,DLOGMOD=INTERACT,...
```

In this example, four model CDRSC definitions and one conventional CDRSC definition are shown.

APPL*
: A model CDRSC from which alias CDRSCs can be created.

APPLB*
: A model CDRSC from which real CDRSCs with a netid of NETB can be created. The clone CDRSCs will have a default logmode of BATCH.

APPLB1*
: A model CDRSC from which real CDRSCs with a netid of NETB can be created. The clone CDRSCs will have a default logmode of INTERACT.

APPLC?
: A model CDRSC from which real CDRSCs with a netid of NETC can be created. The clone CDRSCs will have a default logmode of INTERACT.

Tips:

- In the above sample major node, if the definition for APPLB* is active, but the definition for APPLB1* is not active, a session request for NETB.APPLB11 creates a clone CDRSC based on the APPLB* definition. That clone CDRSC is used for all session requests for NETB.APPLB11 until the clone CDRSC is deleted, even if the APPLB1* definition, which is a better match, has been activated in the meantime.
- When a clone CDRSC is requested for the origin logical unit of a search, a real CDRSC will be requested.
- When a clone CDRSC is requested for the destination logical unit (DLU) of a search, an alias CDRSC will usually be requested. A real CDRSC will be requested for the DLU if the real name has been learned as a result of alias name translation or a prior host in the search path having a predefined real CDRSC for the DLU.

Restriction: The name of each model CDRSC must be unique even if different netids are used.

Resource state requirements

A model CDRSC must be active before it can be used to build clone CDRSCs. You can activate a model CDRSC as you activate other CDRSC definitions by doing one of the following:

- Issue a VARY ACT command.
- Include the major node in which the model CDRSC is defined in the configuration list that VTAM uses when it is initialized.

By activating the model CDRSC, you ensure that the state of the model CDRSC is active, making it available to build clone CDRSCs.

Clone CDRSCs are different from dynamic (clone) applications in that you can choose to retain the clone CDRSC even when they are deactivated or are no longer being used for sessions. You can do this by specifying DELETE=NO on the model CDRSC definition or on a VARY INACT operator command. The specification of the DELETE operand on the VARY command overrides the specification of the DELETE keyword on the model CDRSC definition. This also means that you can issue VARY ACT against an inactive clone CDRSC, which is not possible for dynamic (clone) applications.

Guidelines: The following issues should be considered when specifying the DELETE operand on the model CDRSC definition:

1. Deleting the clone CDRSCs immediately could result in storage thrashing (freeing and getting clone CDRSCs).
2. Deleting the clone CDRSCs immediately could result in search thrashing (losing search history and then rebuilding it).
3. Not deleting the clone CDRSCs could consume system resources unnecessarily.

For example:

- If a particular set of CDRSCs is located once a day, then you should code DELETE=YES.
- If a particular set of CDRSCs is located many times a day for a short lived session, then you should code DELETE=NO.

Adjacent SSCPs

To supplement the VTAM session request routing service, you can code adjacent SSCP tables or have them dynamically defined. The adjacent SSCP table contains lists of SSCPs that can be in session with a host VTAM and can be used to search for the VTAM that controls a destination CDRSC.

When VTAM receives a session initiation or directory search request for a resource that is not located in that VTAM domain, it attempts to locate the resource by sending the request to its adjacent SSCPs. VTAM routes the request to the SSCP coded in the CDRM operand of the CDRSC definition statement. If one is not coded, the CDRSC is dynamically defined, or the session request with the SSCP in the CDRM operand fails, VTAM uses its adjacent SSCP table.

You can supplement the VTAM search with an adjacent SSCP table even though you code a CDRM operand. VTAM simply concatenates the proper adjacent SSCP list to the CDRM owner specification and uses the entries for the search. A session request for the CDRSC is routed to each SSCP in the list until the resource is located or the list is exhausted.

An adjacent SSCP table is particularly useful in networks where the SSCP ownership of resources changes frequently. VTAM can locate the owners dynamically rather than by having the programmer maintain the CDRSC definitions. Also, in networks where a large number of cross-domain resources are owned by only a small number of SSCPs, coding an adjacent SSCP table in each host can be easier than coding individual CDRSC definition statements.

If you code adjacent SSCP tables or have them dynamically defined, you do not have to code CDRSC definition statements for logical units in other domains, but VTAM performance is slower because of the time it takes to send startup requests

to SSCPs that do not own the logical unit. For information on how to improve the performance of adjacent SSCP tables, see “Improving performance” on page 546.

Alternatively, a coded CDRSC can provide direct session setup, rather than waiting for VTAM to locate a logical unit using the adjacent SSCP tables. By placing CDRSC definitions at a central host (for instance, a CMC host) and placing single entry adjacent SSCP tables in other hosts that point to the central host, session setup requests are sent to the central host if nothing is known about a target resource. When the request reaches the central host, if there is a predefined real CDRSC coded with the resource name, the same NETID as the central host (the CDRSC definitions at the central host must have the same network identifier as the central host), and CDRM equal to the actual owning CDRM (note that the central host does not actually confirm that the coded CDRM is the owner), that information is returned with volatile USERVAR characteristics to the origin SSCP, which can rebuild its adjacent SSCP table with the owning SSCP as the first entry (similar to USERVAR processing). The session setup request can then be forwarded directly to the correct target.

You can also increase control over adjacent SSCP selection by creating adjacent SSCP lists for CDRSCs in an adjacent SSCP table. Then, if an adjacent SSCP list is identified for a CDRSC, session setup requests are sent to only the SSCPs in that list. If the target resource is not owned by (or found through) one of the SSCPs in the list, session establishment fails.

For information on using adjacent SSCP tables for routing requests to other networks, see “Defining adjacent SSCPs” on page 579.

Adjacent SSCP lists for CDRSCs

To define an adjacent SSCP list for a CDRSC, include the ADJLIST operand on the CDRSC definition statement. You can also code the ADJLIST operand on the GROUP definition statement, and the ADJLIST operand sifts down to any CDRSCs in the group for which ADJLIST is not coded. The ADJLIST operand specifies the name of a list of adjacent SSCPs, defined in an adjacent SSCP table using the ADJLIST definition statement followed by any number of ADJCDRM definition statements to identify the specific adjacent SSCPs in the list.

For example, following is a sample adjacent SSCP list that could be used by A02M to locate LU1 shown in Figure 138 on page 544.

```
A02ADJ    VBUILD TYPE=ADJSSCP
** ADJACENT SSCP LIST: THIS LIST IS USED WHEN THE ADJLIST OPERAND
**                     FOR A CDRSC SPECIFIES LIST1
**
LIST1    ADJLIST
A01M     ADJCDRM
```

Following is a sample CDRSC definition statement for LU1.

```
          VBUILD TYPE=CDRSC
LU1      CDRSC ADJLIST=LIST1
```

When an adjacent SSCP list is identified for a CDRSC, session setup requests are sent to only the SSCPs in the list. The values coded for SORDER and SSCPORD do not affect the search sequence. If the target resource is not owned by (or found through) one of the SSCPs in the list, session establishment fails. This enables strict control over adjacent SSCP selection.

When using adjacent SSCP lists for CDRSCs, the SSCP selection function of the session management exit routine can be used to delete or reorder entries in a list. VTAM cannot dynamically add entries to an adjacent SSCP list.

To add, delete, or change an adjacent SSCP list for a CDRSC, use the MODIFY RESOURCE command. Adjacent SSCP lists for a CDRSC are displayed using the DISPLAY ADJSSCPS command. Refer to *z/OS Communications Server: SNA Operation* for details on these commands.

Defining adjacent SSCP tables

You can define adjacent SSCP tables by using one, or both, of the following ways:

- Statically define the tables using VTAM definition statements. If this method is used, the SORDER or SSCPORD operand on the NETWORK or CDRM definition statements, or both, can be used to specify a different SORDER or SSCPORD value for each adjacent SSCP table. For more information about SORDER, see "Using SORDER to control network search order" on page 519 or refer to *z/OS Communications Server: SNA Resource Definition Reference*. For more information about SSCPORD, see "Improving performance" on page 546 or refer to *z/OS Communications Server: SNA Resource Definition Reference*.
- Specify the DYNASSCP start option as YES so that VTAM dynamically builds the tables. The search order (SORDER) for dynamically created adjacent SSCP tables is determined by the SORDER start option.

One or more adjacent SSCP tables can be activated in each VTAM host.

Dynamic definition of adjacent SSCPs: VTAM can dynamically create an adjacent SSCP list. This table consists of all of the adjacent SSCPs with which VTAM has an SSCP-SSCP session. VTAM can then route session-establishment requests to all active adjacent SSCPs until the correct SSCP is found. If you code adjacent SSCP tables and use dynamically defined adjacent SSCP tables, VTAM uses the appropriate adjacent SSCP table without being affected.

To enable dynamically defined adjacent SSCP tables, code the DYNASSCP start option. If you specify YES and VTAM does not locate an appropriate adjacent SSCP table, VTAM dynamically routes the session-establishment request to all active adjacent SSCPs until the correct SSCP is found. If you specify NO, VTAM does not perform this dynamic routing.

By using the dynamic adjacent SSCP table function in conjunction with the adjacent SSCP selection function of the session management exit routine, you can control the order of the dynamic adjacent SSCP table. See "Improving performance" on page 546.

Static definition of adjacent SSCPs: Following is an example of an adjacent SSCP table. This table is coded in host A01M, whose configuration is illustrated in Figure 138 on page 544.

```
A01ADJ    VBUILD TYPE=ADJSSCP
** DEFAULT SSCP LIST: THIS LIST IS USED WHEN THE CDRM IS
**                     UNKNOWN AND THE NETWORK IS UNKNOWN
**
A02M     ADJCDRM                      DEFAULT SSCP#1
A03M     ADJCDRM                      DEFAULT SSCP#2
**
**
         NETWORK NETID=NETA
**
** ADJACENT SSCPS FOR A CDRM — USED IF CDRSC OWNER IS A02M
```

```
**                              AND NETID IS NETA
**
A02M     CDRM
A02M     ADJCDRM                      ROUTE ONLY TO THIS HOST
**
** ADJACENT SSCPS FOR A CDRM – USED IF CDRSC OWNER IS A03M
**                              AND NETID IS NETA
**
A03M     CDRM
A03M     ADJCDRM                      ROUTE ONLY TO THIS HOST
```

The *cdrmname* used on the adjacent CDRM definition statement in the adjacent SSCP table must be the same as:

- The name on the CDRM definition statement that defines the SSCP to VTAM
- The SSCPNAME used in the VTAM or the name on the host CDRM definition statement in the external VTAM host

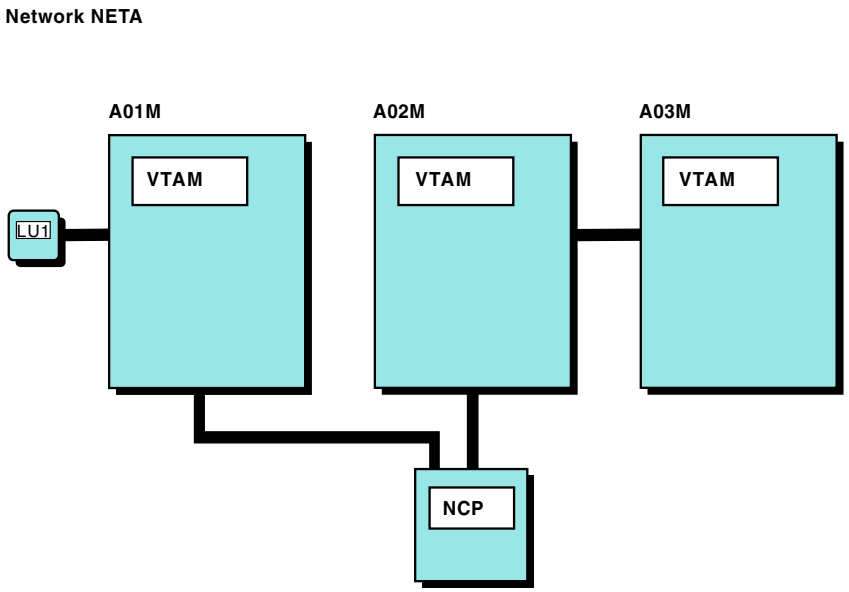


Figure 138. Example of adjacent SSCP table connection

In the SSCP of the originating logical unit, the CDRM name of the destination logical unit, if known, must be correct. If the CDRM name of the destination LU is not correct, the operator must issue a MODIFY CDRM command to change the CDRM name.

If SSCPORD=DEFINED and network specific tables have been coded, session requests for DLU resources are not guaranteed to succeed unless at least one entry in the network specific table provides a connection to the DLU.

During adjacent SSCP routing for the DLU resource, SSCP entries are taken from either the network specific table or the default table. If the NETID of the DLU resource is known and a network specific table is coded for that NETID, that table will be used. Otherwise, entries from the default table will be used when the NETID is not known.

It is possible to code an entry in the default table that has a connection to the DLU and not code an equivalent entry in the DLU network specific table. If SSCPORD has the value of DEFINED, a session request for an unknown DLU would be

successful because of the working entry in the default table. Subsequent session requests will now use the network specific table because the DLU NETID is known. If the network specific table (matching the NETID of the DLU) does not contain entries which provide a connection to the DLU, the routing for these subsequent sessions will fail.

Table search order

You can individually activate multiple adjacent SSCP table definitions. One or more adjacent SSCP tables can be activated in each VTAM host. Each new set of definitions is added to the internal table used by VTAM. New definitions for destinations already in the internal table replace the current adjacent SSCP list for those destinations, but you cannot deactivate existing adjacent SSCP lists. If a new table replaces an existing one, VTAM issues a message informing the operator.

If the CDRM operand is coded on the CDRSC definition statement, it is used in an adjacent SSCP list. If the CDRM name is an entry in the list, it is moved to the top of the list. If the CDRM name is not an entry in the list, VTAM places the CDRM name at the top of the list. The list is then used to send session setup requests to SSCPs in the list until either the owning SSCP is found or the end of the list is reached. When the owning SSCP is found, VTAM remembers its name. None of these actions affect the actual adjacent SSCP table; they affect only the internal copy of the table that is maintained for each CDRSC.

The adjacent SSCP tables are searched in the following manner:

1. If the NETID and CDRM name are known, a search is done for a table coinciding with the known information.
2. If that table is not found or if only NETID is known, a search is done for a table defined with only the NETID given.
3. If an attempt to find that table fails, or if only the CDRM name is known, the indicated CDRM adjacent SSCP table for all networks is used.
4. If an attempt to find that table fails, the default adjacent SSCP table for all networks is used.
5. If an attempt to find that table fails and if dynamic adjacent SSCP tables are permitted, the dynamic table is used.

Routing with adjacent SSCP tables

If you have defined an SSCP list, VTAM sends the session setup request for an undefined destination logical unit to each SSCP in the list until either the owning SSCP is found or the end of the list is reached. If VTAM does not have an SSCP-SSCP session with an adjacent SSCP in the list, the SSCP is ignored. If a session setup request is routed to an SSCP that does not own the destination logical unit, the request is rejected.

For reliability during session setup, VTAM does the following:

- Associates the adjacent SSCP with a cross-domain resource whenever it receives a session initiation request from an SSCP that is not already in the table
- When looking for an SSCP with which to initiate a session, scans the adjacent SSCP table in priority order, giving preference to the following:
  - SSCPs for which the most recent session initiation attempt succeeded
  - SSCPs for which no session initiation attempts have been made

VTAM bypasses any SSCP in the list with which it does not have an SSCP-SSCP session. If the owning SSCP is not found, the session request is rejected, and the dynamic CDRSC is deleted immediately. VTAM does not wait for the time value specified in the CDRSCTI start option.

Routing rules: Session initiation and INQUIRE APPSTAT requests are routed from the originating LU SSCP to the destination LU SSCP using trial-and-error routing. There are several different means of determining the list of adjacent SSCPs to try when performing trial-and-error routing. However, after the list is determined, the processing is very similar for each of these requests. The rules for routing session requests are as follows:

- The request is not routed if its visit count (SSCP rerouting count) is 0.
- The request is never routed back to the SSCP from which it was received.
- The request is never routed back to its originating SSCP.
- The request is never routed in this network if it was received from an SSCP in this network.
- The request is never routed back to the network from which it was received through the same gateway NCP.

Routing and dynamic tables: If you specify the dynamic adjacent SSCP table function and you have not defined adjacent SSCP tables, VTAM dynamically routes session initiation requests to every SSCP with which it is currently in session. This feature eliminates the requirement to define adjacent SSCP tables. You can, however, continue to define adjacent SSCP tables. VTAM uses the user-defined table if one is defined.

Activating adjacent SSCP tables

Activate adjacent SSCP tables using the VARY ACT command. The ID operand of the VARY ACT command specifies either an adjacent SSCP table or a default SSCP list. Adjacent SSCP tables are not major nodes and cannot be deactivated.

Improving performance

For performance reasons, you might want to change the way VTAM routes session requests. You can use either or both of the following to improve SSCP search performance:

- Use the adjacent SSCP selection function of the session management exit routine to shorten or reorder the list of adjacent SSCPs to which an LU-LU session request is directed. The list is composed of the adjacent SSCPs that are coded in the ADJSSCP table for a specific network or a default list and the list of adjacent SSCPs that was built during the previous LU-LU session setup requests. This exit routine can also be used in conjunction with the dynamic adjacent SSCP facility to avoid having to define any ADJSSCP table while maintaining control of session routing requests.
- Code the SSCPDYN and SSCPORD start options. SSCPORD can be specified as a start option, coded on individual ADJSSCP table definitions, or both. If SSCPORD is not coded for a given ADJSSCP table, the current value of the SSCPORD start option value is used each time that ADJSSCP table is selected for use.

If the CDRM name is coded for the destination CDRSC, that entry is still tried first.

Note: When a resource name that is not valid is entered in a logon request, the request is not rejected until the initiate request has been sent to every SSCP in the default SSCP list and every SSCP has rejected it. The search reduction function can be used to minimize the above.

Adjacent SSCP selection function: VTAM constructs a dynamic adjacent SSCP table in the order in which the CDRM-CDRM sessions are established. Therefore, the search order for a resource might not be the same as that which can be achieved using a statically defined adjacent SSCP table. Extra session establishment time might be required if the SSCP list contains many entries or if it is arranged so that several SSCPs must be tried before the owning SSCP can be located. However, to control the order of search in this case, you can use the session management exit routine (adjacent SSCP selection function) in conjunction with a dynamic adjacent SSCP table.

In the session management exit routine, the adjacent SSCP selection function for LU-LU sessions allows you to modify the list of SSCPs in the search. You can shorten or reorder the list of SSCPs from which the next SSCP used in session setup is chosen during session request routing, or you can decide not to route at all.

The list that the session management exit routine receives as input is either an adjacent SSCP table or a list of adjacent SSCPs that is built during the previous LU-LU session setup for the same resource. If no adjacent SSCPs exist, this function is not invoked. If you do not provide this function in your session management exit routine, VTAM uses the adjacent SSCP table as it exists.

z/OS Communications Server: SNA Customization describes the adjacent SSCP selection function of the session management exit routine.

SSCPORD and SSCPDYN routing methods: Use the following table to help you decide how to set SSCPDYN and SSCPORD in your system.

Settings	Description
SSCPDYN=YES SSCPORD=PRIORITY	VTAM adds entries dynamically to the adjacent SSCP table. When establishing sessions in releases prior to VTAM Version 3 Release 2, VTAM gives priority to SSCPs for which the most recent session initiation attempt succeeded or for which no attempt has been made, even if a more direct path to the CDRM that owns the destination LU becomes available. When establishing sessions, VTAM Version 3 Release 2 and later releases give priority to the CDRM that owns the destination LU (if known), and then to SSCPs for which the most recent session attempt succeeded. This combination of options gives you the greatest flexibility for setting up routes across networks, and, if your adjacent SSCP table is large, it gives you the best performance during session setup.
SSCPDYN=YES SSCPORD=DEFINED	Entries are added dynamically to the adjacent SSCP table, but VTAM always searches the table from top to bottom when it tries to initiate a session. Because the dynamic entries are added at the bottom of the list, they are tried last during session initiation. This combination of options results in increased session setup time if the primary paths (those near the top of the list) are not available. But if a secondary path (one nearer the bottom of the table) is used, it does not get priority for subsequent sessions.

Settings	Description
SSCPDYN=NO SSCPORD=PRIORITY	No entries are added to the adjacent SSCP table, except when the owner of a cross-domain resource is defined explicitly by the CDRM operand of the CDRSC definition statement. Therefore, VTAM can use only the paths you define in the table, and if one of these paths fails, VTAM does not attempt to use it again until all other paths also fail. This combination of options tends to limit the number of available cross-network routes. You might also need to define additional adjacent SSCP tables because VTAM does not build them automatically each time an application program issues a CLSDST PASS macroinstruction for a cross-network session.
SSCPDYN=NO SSCPORD=DEFINED	No entries are added to the adjacent SSCP table, except when the owner of a cross-domain resource is defined explicitly by the CDRM operand of the CDRSC definition statement. VTAM always searches the table from top to bottom when it tries to initiate a session. Again, VTAM can only use the paths you define in the table; but, because VTAM always searches the table from top to bottom, a path that fails can be used again as soon as it is restored. This combination of options results in increased session setup time if the primary paths (those near the top of the list) are not available. But if a secondary path (one nearer the bottom of the table) is used, it does not get priority for subsequent sessions. You might also need to define additional adjacent SSCP tables because VTAM does not build them automatically each time an application program issues a CLSDST PASS macroinstruction for a cross-network session. VTAM does not add new entries unless it is the owner of the resource. The owner is determined by the CDRM operand of the CDRSC definition statement, or by session initiation.

CDRM owner verification for cross-domain resources

You can enable SSCPs to take over cross-domain resources without requiring intermediate SSCPs to issue a MODIFY CDRM command. This enables VTAM to automatically change the CDRM ownership of a CDRSC. This can be useful in cross-network sessions where one of the session partners has been taken over by a different SSCP and an attempt is made to establish a new session.

Note: Owner verification of a cross-domain resource is optional.

This SSCP takeover of cross-domain resources is accomplished in the definition of the CDRSC with the VFYOWNER operand. VFYOWNER=YES can be specified only for a CDRSC that is coded with the NETID and CDRM operands. Specifying VFYOWNER=NO (which is the default) allows VTAM to automatically change the CDRM ownership of a CDRSC. VFYOWNER=YES in the CDRSC definition makes VTAM reject session setup requests that contain a conflicting owner. The default (VFYOWNER=NO) allows the takeover of a CDRSC without the need for intermediate SSCPs to issue a MODIFY CDRM command.

The MODIFY CDRM command does not need to be used for dynamically defined CDRSCs. If a session setup request is received for a dynamically defined CDRSC from a CDRM other than the current CDRM owner, VTAM automatically changes the CDRM ownership of the CDRSC. However, the operator might still want to use the MODIFY CDRM command if LU-LU session recovery is initiated by the logical unit that received the original session request.

For example, a terminal user logs on to an application program, and the CDRSC representing the terminal is defined dynamically in the domain of the application program. If the session between the application program and the terminal is disrupted, the application program might try to reinitiate the session instead of requiring the terminal user to log on again. In this case, the VTAM operator in the domain of the application program needs to use the MODIFY CDRM command to change the CDRM owner of the CDRSC representing the terminal, if the terminal is taken over by a backup SSCP other than the SSCP in the application program domain.

A VTAM domain that takes over a cross-domain resource does not need to use the MODIFY CDRM command because in this case, the VTAM shadow resource function applies. For more information on shadow resources, see "Shadow resources."

Changing ownership of cross-domain resources

Each cross-domain resource (CDRSC) definition statement indicates its controlling SSCP (CDRM=*cdrmname*). The controlling SSCP is also called the owning CDRM of the cross-domain resource. The ownership of a cross-domain resource can be changed if the owning CDRM loses its SSCP-LU session with the logical unit, and another SSCP takes over the logical unit. For information on resource takeover, see "Steps for resource takeover" on page 631.

You can use the MODIFY CDRM command to inform VTAM of the takeover, naming the new owner of the cross-domain resource. This command can be used whether the cross-domain resource is active or inactive. If the cross-domain resource is already in session with a logical unit, the session is undisturbed and the owner is saved until the session terminates. Session startup requests continue to be routed to the old owner. Specifying TYPE=IMMED on the MODIFY CDRM command can be used to force the ownership change immediately regardless of active sessions.

Shadow resources

The VTAM shadow resource function enables VTAM to dynamically change what it defined previously as CDRSCs into same-domain resources during takeover.

A shadow resource is an alternate definition of a network resource. For example, an LU is defined to a backup host both as a same-domain resource and as a cross-domain resource. Under normal operation, the LU is a cross-domain resource, and the cross-domain definition is used. The same-domain resource definition, which is not used, is the shadow resource. Whichever definition is not being used is considered to be the shadow resource.

If the host that owns the LU fails, the backup host assumes ownership of the LU, the same-domain definition becomes the active definition, and the cross-domain definition becomes the shadow resource. Resource takeover can be accomplished without disrupting existing active sessions. When the original host becomes

available, the backup host relinquishes ownership of the LU. The backup host then activates its cross-domain definition for the LU, and its same-domain definition once again is the shadow resource.

When the LU is owned by the backup host, other hosts that need to access the LU must be able to communicate with and route requests to the backup host.

Chapter 20. Connecting multiple subarea networks

This section describes a multiple-network environment that uses SNA network interconnection (SNI). A multiple-network environment consists of multiple independent SNA networks that are interconnected. The SNI facility enables communication between these separate networks.

SNA network interconnection enables you to:

- Interconnect multiple, independent SNA networks so that terminal users in one SNA network can access information (application programs) in other SNA networks
- Divide an existing network into smaller, independent networks, with communication between these networks maintained by the SNI function

Following are descriptions of environments in which SNI can be used. (An enterprise is a business organization.)

Independent enterprises
If two or more independent enterprises need to access and exchange information using their current SNA telecommunication networks, SNI can be used to interconnect these autonomous networks, which might have dissimilar characteristics.

Enterprise mergers and acquisitions
If two enterprises have merged or if one enterprise has acquired another, and the dissimilar SNA networks need to interconnect, SNI can be used so that any terminal or application program in one network can access any terminal or application program in the other network.

Multiple networks within an enterprise
If multiple SNA networks have evolved within a large enterprise, each with different network characteristics, SNI can be used to merge these dissimilar networks into a single logical network, yet maintain the autonomy of each.

In each of the preceding examples, SNI can be used to interconnect networks with dissimilar characteristics while preserving the independence of each participating network. These participating networks do not have to change to a common address structure or resource-naming convention, and each network can preserve its existing management procedures and controls.

Defining a multiple-network environment

Defining your multiple-network environment involves identifying your different networks and defining the resources that will help you communicate between the networks. You need the following for controlling cross-network sessions:

Gateway VTAMs
Enable at least one gateway VTAM for each network gateway. The gateway VTAM controls sessions across the network gateway. The gateway VTAM is in session with the gateway NCP, assisting in address translation and cross-network routing. The gateway VTAM must be either an MVS host running VTAM Version 2 Release 2 or higher releases or a VM host running VTAM Version 3 Release 1.1 or higher releases.

© Copyright IBM Corp. 1984, 2011

Gateway NCPs

You must specify at least one gateway NCP for each network gateway. The gateway NCP works with the gateway VTAM to control address translation for cross-network sessions. The gateway NCP provides network address translation from a pool of network alias addresses for each network. The gateway NCP also assists in activating explicit and virtual routes for cross-network sessions.

The gateway NCP is a resource in each network to which it connects. Each gateway NCP has one native network, where the gateway VTAM that can activate the NCP subordinate resources resides. Each gateway NCP also has one or more nonnative networks. For example, in Figure 139 on page 553, NCP21 is subarea 21 in NETA and subarea 51 in NETX. NCP21 native network is NETA; its nonnative network is NETX.

The configuration in Figure 136 on page 529 is the same as the configuration in Figure 139 on page 553, except that Figure 139 on page 553 shows a multiple-network environment because it contains two independent SNA networks. NCP21 in NETA is also subarea 51 in NETX and NCP11 in NETB is also subarea 52 in NETX. There are two NCP11s in this environment. The names can be the same because they reside in different networks (hosts, however, cannot have the same name due to CDRM conflicts).

Code everything that you would for Figure 136 on page 529. NETID must be coded in the start options to the definition of HOST1 in NETA:

Add at least the following to the definition of HOST2 in NETA:

- GWSSCP=YES must be defaulted or coded and NETID should be coded in the start options.
- One NETWORK definition statement for each CDRM or group of CDRMs, identifying which CDRMs reside in each network.
- CDRM definition statement in the cross-domain resource manager major node for VTAM3 in NETB.
- A GWPATH definition statement following the cross-domain resource manager minor node for VTAM3 in NETB. Code the ADJNET, ADJNETEL, ADJSA, and GWN or SA operands.
- NETID and GWCTL operands on the PCCU definition statement defining NCP21 in NETA.
- NETID, HSBPOOL, and CANETID operands on the BUILD definition statement in the NCP major node for NCP21 in NETA.
- GWNAU definition statement in the NCP major node to identify subarea 52 in NETX.
- NETWORK definition statement in the NCP major node to identify NETX. Code the NUMHSAS and SUBAREA operands.
- GWNAU definition statement to represent NCP21 in NETA as NCP51 in NETX.

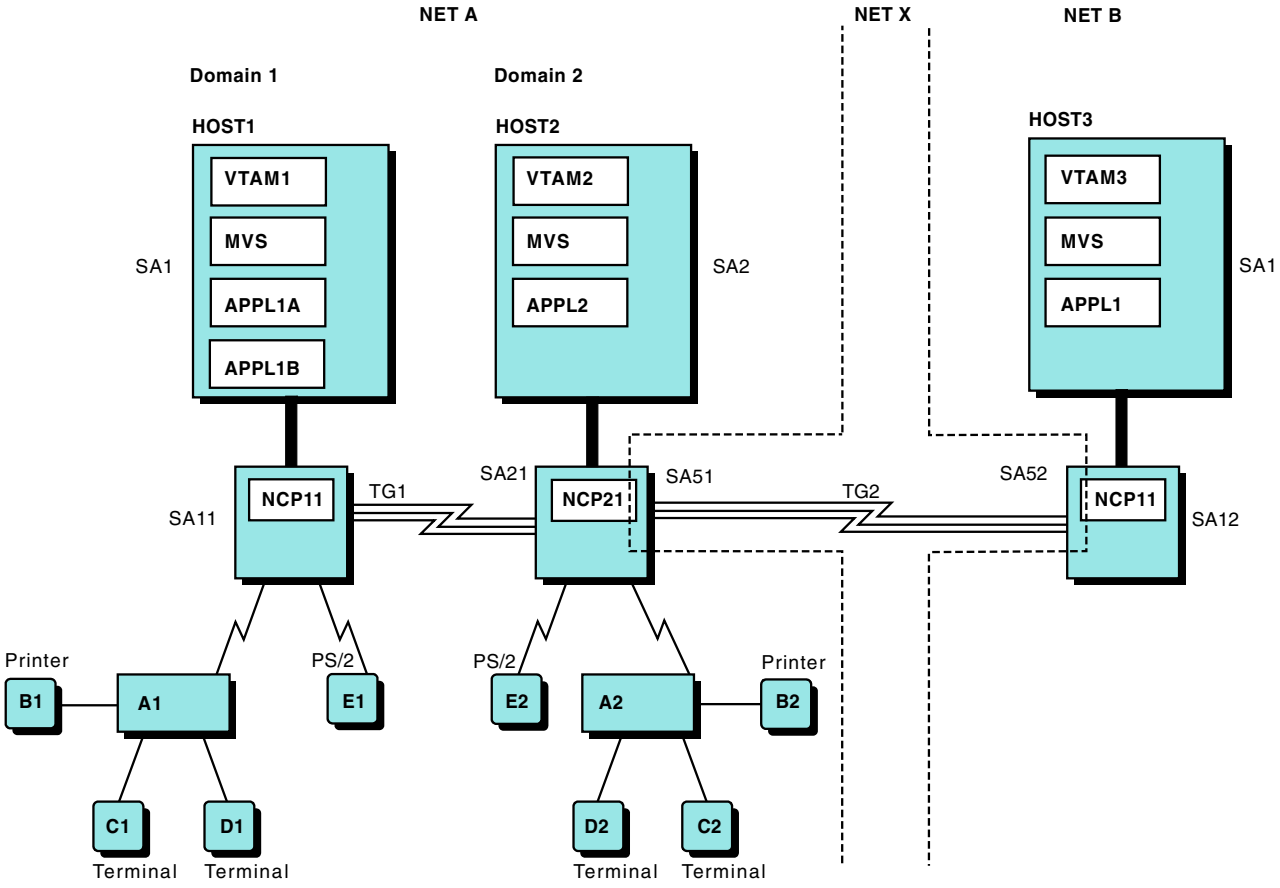


Figure 139. Multiple-network environment

Also code at least the following in HOST3 in NETB:

- Start options for HOST3 in NETB (NETID, SSCPID, and SSCPNAME); code or default to GWSSCP=YES.
- Application program minor node for APPL1 in NETB.
- One NCP major node for NCP11 in NETB.
- One NETWORK definition statement for each CDRM or group of CDRMs, identifying which CDRMs reside in each network.
- CDRM definition statements in the cross-domain resource manager major node for VTAM3 in NETB and VTAM2 in NETA.
- A GWPATH definition statement following the cross-domain resource manager minor node for VTAM2 in NETA. Code the ADJNET, ADJNETEL, ADJSA, and GWN or SA operands.
- NETID and GWCTL operands on the PCCU definition statement defining NCP11 in NETB.
- NETID, HSBPOOL, and CANETID operands on the BUILD statement in the NCP major node for NCP11 in NETB.
- GWNAU definition statement in the NCP major node to identify NCP51 in NETX.
- NETWORK definition statement in the NCP major node to identify NETX. Code the NUMHSAS and SUBAREA operands.
- GWNAU definition statement to represent NCP11 in NETB as NCP52 in NETX.

Note: Resources with identical names can exist in interconnected networks. To resolve naming conflicts, use the NetView alias name translation facility or the NQNMODE=NQNAME start option. For information on the NetView alias name translation facility, see "NetView alias name translation facility" on page 593. For information on the NQNMODE=NQNAME start option, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

SNA network interconnection provides the following major functions:

Cross-network session control
Enables VTAM to perform cross-network session initiation and termination. As required, VTAM can request address translation, name translation, and configuration data.

Address translation
Enables networks with different addressing schemes to communicate. The gateway NCP performs the function of address translation when crossing network boundaries.

Resource name translation
Enables networks with conflicting resource names to communicate. This function can be performed by the NetView program or the VTAM session management exit routine.

Note: The NQNMODE=NQNAME start option is an alternative to resource name translation. For more information, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Cross-network problem determination and network management
Collects and correlates problem determination and network configuration information for cross-network sessions. The NetView program provides the function of cross-network problem determination and network management.

SNI configurations

This section describes some basic SNI configurations and their attributes.

Single gateway configuration
This configuration consists of one gateway VTAM and one gateway NCP. This configuration is the easiest to configure and define, and cross-network session control is centralized. However, failure of the gateway NCP disrupts all cross-network sessions. This type of gateway can be used to connect up to 255 networks, with the gateway NCP residing in multiple nonnative networks.

Single gateway VTAM with multiple gateway NCPs
In this configuration, two or more NCPs act as the gateway NCPs. This configuration has the advantages of the single gateway configuration, but the failure of one gateway NCP does not disrupt all cross-network sessions.

Serial configuration
In this configuration, one gateway VTAM resides in a transport network and controls two or more gateway NCPs that connect to the nonnative networks. This configuration is best if the networks on either side of the transport network do not exchange much information.

Back-to-back configuration
This configuration consists of two gateway VTAMs and two gateway NCPs

that connect through a null network. This configuration provides the most network isolation and security, but it requires more network definition. It also means that address and name translation must be performed twice, and the cross-network session path is longer. The SSCP-SSCP session between HOST2 and HOST3 in Figure 139 on page 553 is an example of a back-to-back configuration.

Start options defining other networks

The start options that you set for VTAM can determine whether the VTAM is a gateway VTAM or a nongateway VTAM.

Note: You can use the XNETALS start option to connect to nonnative type 2.1 PUs. For more information see "Nonnative network type 2.1 connections" on page 254.

Start options for gateway VTAMs

To create a gateway VTAM, specify the following start options:

NETID (required)
NETID specifies the name of the network that contains this gateway VTAM. The network identifier allows VTAM to determine which gateway NCP definition statements apply to the gateway host.

SSCPID (required)
Change SSCPID as necessary. SSCPID uniqueness must be maintained across connecting networks because two SSCPs with the same ID cannot establish a session.

SSCPNAME (required)
SSCPNAME specifies the unique name by which this gateway SSCP is known to the gateway NCP and other networks. This name must be the same as the NAME operand on a GWNAU definition statement that is used to predefine an SSCP element address in a gateway NCP. This name must also be the same as the owner name (CDRM operand) used on the CDRSC definition statements in other networks that represent resources owned by this VTAM.

HOSTPU (recommended if the NetView program is installed)
HOSTPU specifies a unique network name for the VTAM host physical unit. The NetView program uses this name to determine which VTAM host physical unit it is tracing. (If HOSTPU is not specified, all VTAM host physical units are named ISTPUS.)

GWSSCP=YES (required)
GWSSCP=YES indicates that this SSCP can serve as a gateway SSCP.

Note: Even if this SSCP is not performing a gateway function, GWSSCP can be used to enable this SSCP to act as an intermediate SSCP during a session setup.

GWSSCP start option for nongateway VTAMs

If you code GWSSCP=NO, trial-and-error routing of session initiation requests in the originating domain can be done, but no rerouting is possible for session initiation requests received by this host from other SSCPs.

Coding GWSSCP=YES for a VTAM that does not activate gateway NCPs permits VTAM to reroute received session establishment and directory requests. It also enables you to code GWPATH definition statements in conjunction with a CDRM definition to define multiple gateway paths to that CDRM.

Configuration lists for gateway VTAMs

You might want to update the initial configuration list to include any new major nodes and tables, such as the following:

- Gateway NCP major nodes
- CDRM major nodes
- CDRSC major nodes
- Adjacent SSCP tables
- Path tables

Connecting networks

When you are planning to connect networks, you should consider the following, and if necessary, discuss these things with the system programmers for the networks that are being connected:

- To guarantee uniqueness, know the SSCPIDs and the NetView domain IDs used in each network, or use a naming convention.
- If you own a gateway NCP:
  - Obtain the MAXSUBA specification used for any attaching network that contains nonextended subarea addressing nodes so that you can code the MAXSUBA operand on the NETWORK definition statement in the gateway NCP.
  - Specify an available subarea number in the attaching network on the SUBAREA operand on the NETWORK definition statement.
  - Know how many subarea nodes can communicate with the gateway NCP in the attaching network so that you can code the NUMHSAS operand on the NETWORK definition statement in the gateway NCP.
  - Exchange information about explicit route and virtual route structures so that you can code PATH definition statements for nonnative networks in the gateway NCP.
- If you plan to code GWPATH definition statements, you need to know the subarea and element addresses of all CDRMs in adjacent networks with which your gateway VTAM is to establish sessions.
- If you are not using network-qualified names, to avoid future name conflicts, decide upon a naming convention for alias names and for real names.
- If you want to guarantee that certain application programs can always obtain an alias address, you need to know the real logical unit names of those application programs. These names are required on the GWNAU definition statements that predefine specific resources in the gateway NCP.
- Tell system programmers for the other networks the names to be used for logical units. Discuss equivalent COS and logon mode entry names.
- Some application programs require a definition for each of their session partners. If these application programs are to have cross-network sessions, define one of the following to the application programs:
  - Alias names for session partners in other networks
  - LU alias names (on the CDRSC definition statements) for session partners in other networks

  - USERVARs that map to the network-qualified names of session partners in other networks
- If you code a CDRM operand on a CDRSC definition statement and you specify a NETWORK definition statement for it, use the real SSCP name for the host in the other network.

  The real SSCP name is the name specified in the SSCPNAME start option. For any host for which SSCPNAME is not specified, the real SSCP name is the name of the host CDRM for that host. You should use this name on your definition for this CDRM if you code such a definition.
- If you code a CDRM definition statement in an adjacent SSCP table, specify the real SSCP name on the label of the CDRM definition statement.
- If you code an SSCP name on an LU definition statement for the NetView alias name translation facility, you must specify the real SSCP name.
- You must code the real SSCP name on the ADJCDRM operand in an adjacent SSCP table.
- Ensure that the COS table named in the COSTAB operand of the BUILD definition statement or a NETWORK definition statement in the gateway NCP is available to the gateway SSCP. You must install it in VTAMLIB. The table specified in the COSTAB operand is used for mapping Class of Service table entries for the routes that originate in the gateway NCP and terminate in the network being defined. If a COS table for the connected network is not supplied, the COS table for your network must include any COS names that can be requested by logical units in other networks.
- Ensure that NCP data transfer related operands (such as UNITSZ, TRANSFR, and BFRS) that are coordinated between domains in a multiple-domain network are also coordinated between interconnected networks.
- If you use the NetView program, you need to know the NetView suppression characters used in every domain in every network to ensure that the characters are all the same.

Defining a gateway VTAM

The gateway VTAM, in conjunction with the gateway NCP, performs cross-network session initiation and termination by acting as an intermediary between VTAMs in the connected networks. (After the session is set up, there is no need for the LU-LU session path to go through the gateway VTAM.) The gateway VTAM also directs the gateway NCP to set up the address transformations and routes needed for cross-network sessions. The gateway VTAM determines the routing for cross-network sessions and resolves resource names in conjunction with the name translation facility.

Note: A gateway VTAM must be either an MVS host with VTAM Version 2 Release 2 or higher, or a VM host with VTAM Version 3 Release 1.1 or higher.

The gateway VTAM can be in any one of the connected networks.

To convert a non-SNI VTAM to a gateway VTAM, you need to make the following changes:

- Update your start options.
- Update your initial configuration list.
- Update your application program major node if you are using NetView alias name translation facility.

- Update your CDRM major node.
- Create CDRSC definition statements (optional).
- Create an adjacent SSCP table.

In a gateway configuration, VTAM has several different names. The following three names are assigned using start options:
- The ID specified on the SSCPID start option
- The name specified on the SSCPNAME start option
- The name specified on the HOSTPU start option

These are the names by which this SSCP CDRM is known by other domains in this and other networks. No requirement exists for these names to match any of the other names listed previously. However, you might want to have each CDRM known by a single name throughout all the networks. You can do this by coding a single set of CDRM definitions that is used in all the connected networks. Coding the same name for this SSCP on all definitions makes network management easier.

Another name can be assigned on the CDRM definition statement. However, if that name differs from the SSCPNAME that is used, the results are unpredictable because of VTAM unique use of each of the varying names. It is recommended that these names be the same to avoid inconsistencies. One exception to this is that HOSTPU must be different from the CDRM name or SSCPNAME.

Defining a gateway NCP

The gateway NCP assists in cross-network session initiation and termination by providing network address transformations from the pool of network alias addresses for each network. In addition, the gateway NCP assists in activating explicit and virtual routes for use by cross-network sessions. The gateway NCP is a part of all the networks it connects. The resources attached to the gateway NCP, however, are owned by VTAMs in only one of the attaching networks.

The gateway NCP is attached to other subareas in the various networks. It connects with SDLC links, token rings, or channel attachments.

Every gateway NCP has one native network. The native network is identified by the NETID operand on the BUILD definition statement in the NCP generation definition. The native network contains the VTAMs that are allowed to own (activate) the subordinate resources (the physical units, LUs, peripheral links, and cross-subarea links) of the gateway NCP. The gateway NCP resources can be owned by any or all of the VTAMs in the native network, whether they are gateway VTAMs or not.

Every gateway NCP has one or more nonnative networks. Each nonnative network is identified by a NETWORK definition statement in the NCP generation definition. A nonnative network is one that is attached to the gateway NCP for the purpose of interconnection. A VTAM in a nonnative network can activate the gateway NCP but cannot activate any of the NCP subordinate resources.

At least one gateway NCP is required for network interconnection. The gateway NCP performs the address translation necessary for cross-network session traffic.

A pool of alias addresses is used for cross-network sessions. The coding of the GWNAU definition statement determines whether VTAM can activate SSCP-SSCP

sessions in some configurations, because it defines representations of cross-network logical units and SSCPs within the gateway NCP.

An SSCP can neither send nor receive a request to establish a cross-network session with another SSCP unless it knows that SSCP subarea and element address. A nongateway SSCP cannot get the address from the gateway NCP; thus, the nongateway SSCP depends on the GWNAU definition statement corresponding to the other-network SSCP to supply the alias address. You can code this address on the SUBAREA and ELEMENT operands of the CDRM definition statement that defines the other-network SSCP to the nongateway host.

In a back-to-back gateway configuration, a gateway SSCP must be able to tell its gateway NCP what address represents the other-network SSCP in the second gateway NCP. The address in the GWNAU definition statement for the other-network SSCP in the second gateway NCP is the same as that in the ADJNETSA and ADJNETEL operands of the GWPATH definition statement following the CDRM definition statement that defines the other-network SSCP to the gateway host.

Sometimes a request arrives at a gateway NCP before VTAM has determined the address. In this case, the gateway NCP rejects the request. At the same time, the NCP signals the destination SSCP to set up the session. The NCP knows the address of the destination SSCP only if the SSCP name on the GWNAU definition statement for the alias address to which the ACTCDRM request is sent matches an SSCP name that the NCP received in an ACTPU request. This SSCP name sent in an ACTPU request from a gateway SSCP is determined by the SSCPNAME start option.

If you are dividing a network into multiple networks and some subarea nodes are not extended network addressing nodes, group those subarea nodes, if possible, into one or more unique networks. You might want to change MAXSUBA to meet a particular addressing requirement for a newly created network.

If a nonextended addressing node VTAM activates a gateway NCP, the NCP must define the MAXSUBA operand on the BUILD and NETWORK definition statements. A nonextended addressing NCP, functioning as a gateway, does not support sessions with resources in nodes having subareas greater than MAXSUBA. The nonextended addressing node gateway cannot activate an explicit route to a subarea greater than MAXSUBA in that network.

Converting a non-SNI NCP to a gateway NCP

To define a gateway NCP, code an NETID and a GWCTL operand on the PCCU definition statement in your NCP definition.

NETID specifies the network containing the host node that the PCCU definition statement defines. The NETID and SUBAREA operands are used together to determine which VTAM host should process this PCCU definition statement. If NETID is omitted, only the SUBAREA value is used. This could result in the PCCU definition statement being processed by the wrong VTAM host.

The GWCTL operand tells VTAM whether other gateway SSCPs can perform some of the cross-network session setup functions. GWCTL specifies this SSCP role only for the gateway NCP being defined. It applies only if this SSCP is on the session setup path.

In addition to defining the gateway NCP itself, define the network that is native to it and the nonnative networks.

Defining native networks

The native network contains the VTAM that is allowed to own and activate this gateway NCP. The native network is identified by the NETID operand on the BUILD definition statement.

Coding the BUILD definition statement

Identify the NCP native network with the NETID operand on the BUILD definition statement.

On the BUILD definition statement, add the following operands:

NETID (required)
The NETID operand specifies the name of the native network. VTAM compares the NETID coded on the BUILD definition statement to the NETID start option coded for VTAM. If these values match, VTAM is in the NCP native network. Resources attached to an NCP are controlled only by SSCPs in the native network of the NCP.

If the NETID specifications do not match, VTAM searches for a NETWORK definition statement with a matching NETID.

HSBPOOL (required)
Two half-session control blocks (HSCBs) are required to represent each cross-network session, one for each session partner. HSCBs can be statically allocated with the GWNAU definition statement.

The value specified for the HSBPOOL operand is the total number of HSCBs in the gateway NCP to represent cross-network sessions that require dynamic assignment of an HSCB. HSCBs for each network attached to the gateway NCP are dynamically allocated from this pool. You can limit the number of HSCBs that can be allocated from the pool for a given network by specifying a NETLIM value on the BUILD or NETWORK definition statement. You can also limit the HSCBs for a given address by specifying SESSLIM on the BUILD or NETWORK definition statement.

CANETID
The CANETID operand applies only to a channel-attached gateway NCP and the 3720 communication controller. CANETID specifies the network ID of the network containing the host attached through each channel adapter. The gateway NCP uses the CANETID specification to determine the network affinity for each specified channel adapter. CANETID thus specifies in which network each channel-attached host is located.

The gateway NCP uses CANETID to determine which channel-attached hosts can activate resources attached to the gateway NCP. It also determines whether a particular channel-attached host is in the native network.

Also do the following:

- Add one ERLIMIT operand (optional).
- Add one SALIMIT operand (optional).
- Add one GWSESSAC operand.
- Add one GWAEXIT operand (optional).
- Add one COSTAB operand (optional VTAM-only operand).

- Update the VRPOOL to include the additional virtual routes from other networks that end in the gateway NCP.

Coding the GWNAU definition statement

For the native network, place a GWNAU definition statement after the SYSCNTRL definition statement:

```
BUILD NETID=neta
SYSCNTRL
GWNAU NAME=xname,NETID=yname,ELEMENT=nn
```

NAME
Identifies a CDRM for the connected network.

When you code the CDRM definition for this resource in your gateway VTAM, this name must match the CDRM name.

NETID
Identifies the network in which this CDRM resides.

This value must match the value for the NETID operand of the BUILD definition statement found in the other network gateway NCP.

ELEMENT
Reserves an element address in the native network to represent the nonnative CDRM.

This value must match the ELEMENT value in the GWPATH statement for gateway VTAM.

GWNAU reserves one or more element addresses in the native network to represent other network resources. Some elements can be reserved in a dynamic pool while others can be predefined (that is, associated with a particular resource name).

Following are two circumstances when the GWNAU definition statement must be coded to predefine an element address in the gateway NCP:

- If a nongateway SSCP in the network is to establish an SSCP-SSCP session with a gateway SSCP in another network, the GWNAU definition statement must be coded to predefine an element address for a gateway SSCP. This element address must also be specified on the ELEMENT operand of the CDRM definition statement that defines the gateway SSCP to the nongateway SSCP.
- If you have cross-network sessions between CDRMs in nonadjacent networks, the GWNAU definition statement must be coded to reserve an element address in the gateway NCP subarea within the adjacent network. This element address must be specified on the ADJNETEL operand of the GWPATH definition statement that defines the nonadjacent CDRM.

Optionally, application programs that have multiple cross-network sessions can be predefined so that half-session control blocks (HSCBs) required to support the sessions with the application program are preassigned to the element used to represent the application program in the gateway NCP. This ensures that at least the preassigned number of control blocks are available to represent sessions with the application program. Statically allocated HSCBs and addresses cannot be used by any other resource.

Note: If you plan to have session monitors communicating from one network to another, it is strongly recommended that you code two GWNAU definition statements for each network management session (one for the

primary-to-secondary session and one for the secondary-to-primary session) to reserve session control blocks in the NCP for the network management session.

Coding the HOST definition statement

On each HOST definition statement defining VTAM, you should add one NETID operand. This is not required but is recommended. If NETID is omitted, only the SUBAREA value is used. This can result in the HOST definition statement being processed by the wrong VTAM host.

NETID specifies the network containing the host node to which this HOST definition statement applies. The NETID and SUBAREA operands are used together to determine which VTAM host should process this HOST definition statement.

Update NCP PATH definition statements by doing the following:

- Define the new routes that originate in the gateway NCP subarea.
- Map virtual routes to explicit routes for routes that originate in the gateway NCP subarea. For virtual routes that terminate in a VTAM host subarea, the forward and reverse virtual route numbers must match.

On each PU definition statement that represents a type 4 physical unit, add one NETID operand (required).

NETID specifies the NETID of the network in which the physical unit, represented by this link station, resides.

Defining nonnative networks

Use the NETWORK definition statement in the NCP major node to identify each network attached to a gateway NCP.

To define the nonnative networks, the following changes are required:

1. Add one or more NETWORK definition statements (one is required for each nonnative network).

   NETWORK identifies a nonnative network that attaches to the gateway NCP. On the NETWORK definition statement, the following operands can be coded:

   - Add one ACTPU operand (optional).

     ACTPU specifies whether SSCPs in the nonnative network can activate the gateway NCP.
   - Add one COSTAB operand (optional VTAM-only operand).

     COSTAB specifies a user-written Class of Service table to be used for routes that originate in the gateway NCP and that are within the nonnative network being defined. No table is needed if the only COS table entries used are ISTVTCOS and the default Class of Service.
   - Add one MAXSUBA operand (optional).

     This operand specifies the MAXSUBA value for the nonnative network. It is not required if all nodes in that network use extended network addressing.
   - Add one SALIMIT operand (optional).

     The SALIMIT operand specifies the maximum number of subareas that can be used in the nonnative network.
   - Add one NUMHSAS operand (required).

This operand specifies the number of subarea nodes in the nonnative network that can communicate concurrently with the gateway NCP.

- Add one SUBAREA operand (required).

  This operand specifies the subarea address of the gateway NCP as known in the nonnative network.

2. Add one or more GWNAU definition statements (required).

   Add one or more GWNAU definition statements after the NETWORK definition statement when defining nonnative networks.

   ```
   NETWORK NETID=netb
   GWNAU NETID=neta,NAME=cdrmname,ELEMENT=nn
   ```

   NETID
   Identifies the network in which the following CDRM name resides to the nonnative network.

   This value must match the value for the NETID operand in the BUILD definition statement.

   NAME
   Identifies a CDRM to the nonnative network.

   ELEMENT
   Reserves an element address in the nonnative network to represent that CDRM.

   Some elements can be reserved in a dynamic pool, while others can be predefined. A predefined element is one that is associated with a particular resource name. Each resource that has a cross-network session with a resource in this network must have at least one address. (Some application programs require more than one address if they have multiple sessions.)

   Note: If you plan to have session monitors communicating from one network to another, it is strongly recommended that you code two GWNAU definition statements for each network management session (one for the primary-to-secondary session and one for the secondary-to-primary session) to reserve session control blocks in the NCP for the network management session.

   Optionally, you can preassign session-related control blocks (HSCBs) to application programs (or SSCPs) that have many cross-network sessions. The HSCBs are preassigned to the element that represents the application program in the gateway NCP. This ensures that at least the preassigned number of control blocks are available to represent sessions with the application program.

3. Update NCP PATH definition statements by doing the following:
   - Add new NCP PATH definition statements to define routes that originate in the gateway NCP subarea for the network identified by the previous NETWORK definition statement. See “Cross-network routing” on page 584.
   - Map virtual routes to explicit routes for these routes. For virtual routes that terminate in a VTAM host subarea, the forward and reverse virtual route numbers must match. See “Cross-network routing” on page 584.

Dynamic network identification

A single gateway NCP can back up any number of similarly configured gateway NCPs. A gateway VTAM accepts the names of the networks connected to a gateway NCP, without requiring any more definition than NCP already requires. Any VTAM connected directly to a gateway NCP through a leased SDLC line can transmit its own NETID.

Dynamic network identifier enables you to dynamically add and delete NETIDs (networks) within a gateway NCP. Its primary purpose is to provide more backup flexibility; a single gateway NCP can be used to back up numerous NCPs with different NETIDs (residing in different networks). It might also be useful in situations where a gateway NCP occasionally uses links in other networks, especially on a random basis.

The gateway NCP implementing the dynamic NETID must be an NCP Version 5 Release 4. The gateway VTAM that owns the gateway NCP implementing the dynamic NETID must be in a host running MVS or VM/ESA®. VTAMs that are connecting the gateway NCP can run VSE/ESA, MVS, or VM/ESA.

Enabling dynamic NETID

To enable dynamic NETID, code a model network without the NETID and SUBAREA coded. Do the following:

1. Define NETWORK definition statements for all the networks for which you want continuous contact.
2. Define a model network statement with the configuration of any other network the NCP can contact.
3. Control the number of networks represented by a model with which the NCP can have contact simultaneously by specifying the number in the COPIES operand of the NETWORK definition statement for the model network. The COPIES operand is required. If you specify COPIES=3, three model networks are created in the gateway NCP.

Creating dynamic networks from model networks

When the gateway NCP receives contact from a network that has an unknown NETID, NCP accepts the NETID in the XID exchange. NCP uses an available pool network (which is a copy of the model network), assigns the unknown NETID to it, and marks it as dynamic, thus dynamically creating the network. As long as there are users of the dynamic network (the INN links are active), the network remains in existence. After there are no more active users of the network, the DYNAMIC network is cleared and returned to the pool, making it available for use by another network.

When a network is dynamically created within a gateway NCP, all gateway VTAMs that have activated the physical unit of that NCP are notified and made aware of the new network. Also, whenever a gateway VTAM activates the physical unit of a gateway NCP, that VTAM is notified of all dynamic and static (predefined) networks. VTAM is also notified when dynamic networks are cleared and no longer exist.

Note: VTAMs prior to Version 3 Release 4 and nongateway VTAMs are not notified about the addition or deletion of dynamic networks.

Dynamic network capability is supported on leased, subarea dial, and NTRI INN links. The protocols used for dynamic NETID depend on whether the link is a leased (including NTRI) or switched line. For detailed information on the NETID negotiation protocols used for link activation between two NCPs, refer to the *NCP, SSP, and EP Resource Definition Reference*.

Considerations for implementing dynamic network identifier

You need to consider the following when implementing dynamic network identifier:

- All dynamically created networks have the same subarea, but have different NETIDs.
- The model network must have paths defined to at least the first adjacent subarea that is contacted when the NETID is initially established. After this, you can use dynamic path update to dynamically define paths. See "Dynamic path update" on page 369. Paths require careful planning, especially when the paths go to the same subarea number in several different networks.

  Note: When a dynamic network is cleared, any paths created by dynamic path update for that network are also cleared and lost.
- Because the model network represents other networks, the model network must contain all necessary GWNAU definition statements for any NETID of any network that the model represents. GWNAU definition statements cannot be dynamically added, modified, or deleted.
- For a link station and its line to support dynamic NETID, the GROUP definition statement for the INN link must have NETID=(name,ANY) coded as a parameter. For leased and switched links, this is coded on the respective GROUP definition statements. For NTRI links, this is coded on the LOGICAL GROUP(s). To establish dynamic NETID capability, code NETID=ANY on the GROUP definition statement. All INN links defined under the dynamic GROUP have the ability to use it.

  Note: When coding NETID=(name,ANY) on the GROUP, *name* must be either the default or predefined network within the gateway NCP. If *name* is omitted and NETID=(,ANY) is coded, the default network is assumed.
- Ensure that the necessary GWPATH definition statements are coded for the cross-domain resource managers that you are contacting.
- To display static and dynamic networks, use the command

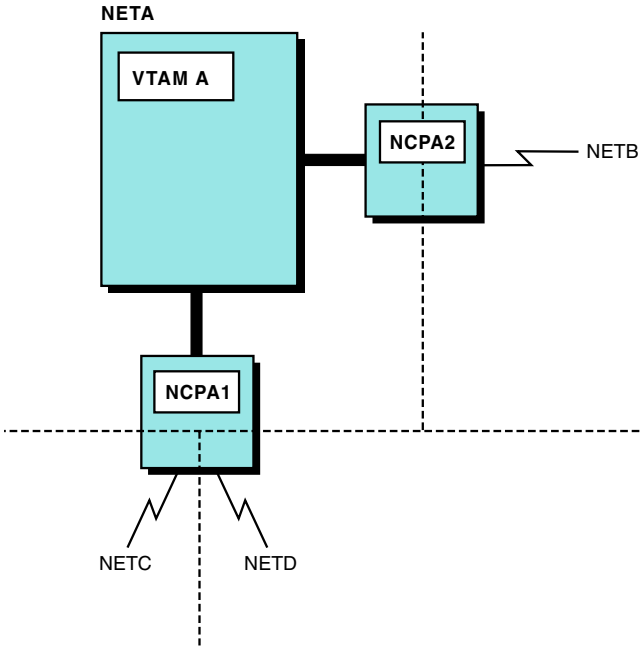
  ```
  D NET,COS,ID=ncp_name,NETID=*
  ```

  This displays the COS table associated with a NETID. The model network is displayed as `*NETWORK`. Any static networks appear above the model network, and any dynamic networks are displayed below it. Remember, the results are with respect to the VTAM where you issue the command. When displaying a remote NCP, a dynamic network might appear as static because it is a predefined network in the local NCP or because it is in the network that issued the VTAM network identifier.
- To display the network identifier of active link stations, use the command

  ```
  D NET,STATIONS,A
  ```
- A leased line can change its network identifier to a predefined network identifier. To do this, code `NETID=(name,ANY)` on the GROUP definition statement for the leased line. A model network is not needed.
- You need to ensure that the value specified on the HSBPOOL operand on the BUILD definition statement is large enough to accommodate the number of addresses needed to represent all SSCPs and logical units that participate in cross-network sessions (specified on the NUMADDR operand on the GWNAU definition statement) for all predefined and dynamic networks.

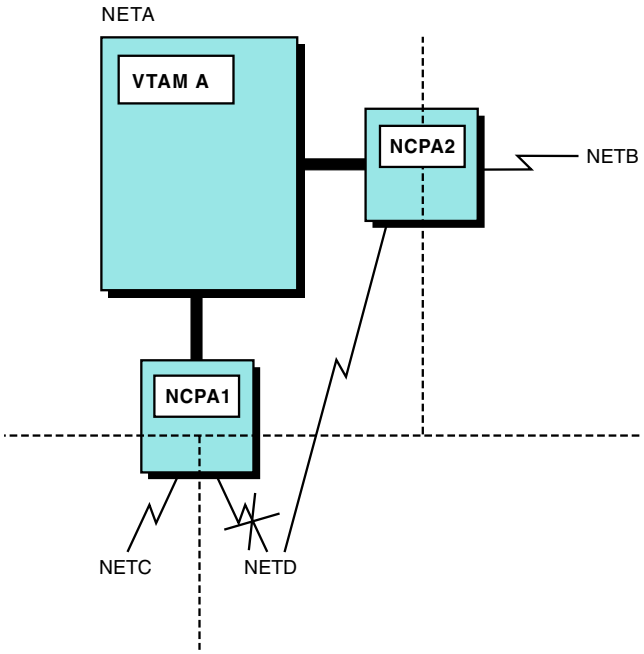
Sample dynamic NETID configuration

Figure 140 on page 566 is a sample dynamic NETID configuration.



NCPA1 and NCPA2 are gateway NCPs. NCPA1 supports NETC and NETD; NCPA2 supports NETB. NCPA2 has an extra switched line defined.

Figure 140. Backup of a gateway NCP (Part 1 of 2)



NCPA1 link to NETD goes down. NETD dials NCPA2 to connect to NETA.

Figure 140. Backup of a gateway NCP (Part 2 of 2)

Following is the coding for the backup gateway NCP (NCPA2):

```
DNETID   BUILD VERSION=V5R4,
  .
  .
  .
GLS      GROUP LNCTL=SDLC,ACTIVTO=420.0,TYPE=NCP,DIAL=NO,REPLYTO=3,    *
               SDLCST=(STPRIM,STSECD),PUTYPE=4,                        *
               NETID=(NETA,ANY)
LLS      LINE  ADDRESS=(10,FULL),                                      *
               DUPLEX=FULL,                                            *
  .
  .
  .
PLS      PU    PUTYPE=4,TGN=18,                                        *
               RETRIES=(,2,0)
GSW      GROUP LNCTL=SDLC,ACTIVTO=0,DIAL=YES,REPLYTO=3,PUTYPE=4,       *
               SDLCST=(STPRIM,STSECD),V25BIS=(YES,DLSDLC),             *
               NETID=(NETB,ANY)
LSW      LINE  ADDRESS=(035,FULL),                                     *
  .
  .
  .
NETB     NETWORK  SUBAREA=9,NUMHSAS=12,NETID=NETB,                     *
               PATHEXT=2,TGBXTRA=3
         GWNAU ELEMENT=6,NETID=NETA,NAME=SSCPA,NUMSESS=4
         GWNAU NUMADDR=7
         PATH  DESTSA=19,                                              *
               ER0=(19,1),ER1=(19,1),ER2=(19,1),ER3=(19,1),            *
               ER4=(19,1),ER5=(19,1),ER6=(19,1),ER7=(19,1),            *
               VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
         PATH  DESTSA=(8,18),                                          *
               ER0=(8,36),ER1=(8,36),ER2=(8,36),ER3=(8,36),            *
               ER4=(8,36),ER5=(8,36),ER6=(8,36),ER7=(8,36),            *
               VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
COPIES   NETWORK  SUBAREA=5,NUMHSAS=12,COPIES=6,                       *
               PATHEXT=6,TGBXTRA=5
         GWNAU ELEMENT=8,NETID=NETA,NAME=SSCPA,NUMSESS=3
         GWNAU ELEMENT=15,NETID=NETB,NAME=SSCPB
         GWNAU NUMADDR=8
         PATH  DESTSA=15,                                              *
               ER0=(15,3),ER1=(15,3),ER2=(15,3),ER3=(15,3),            *
               ER4=(15,3),ER5=(15,3),ER6=(15,3),ER7=(15,3),            *
               VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
         PATH  DESTSA=24,                                              *
               ER0=(24,18),ER1=(24,18),ER2=(24,18),ER3=(24,18),        *
               ER4=(24,18),ER5=(24,18),ER6=(24,18),ER7=(24,18),        *
               VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
GENEND   GENEND
         END
```

Notes:

1. Omit the NETID operand from the NETWORK definition statement.
2. Ensure that you have PATH definition statements defined for all the adjacent subareas. Thus, for the above coding, the adjacent subarea in NETD must be either 15 or 24.

Defining cross-domain resource managers

You need to code CDRM definition statements for VTAMs in other networks with which this VTAM requires direct communication. In a CDRM major node:

1. Add one NETWORK definition statement for each network to be connected through a gateway NCP. This is necessary to identify the networks that contain the defined CDRMs.

   The NETID operand of the NETWORK definition statement identifies the network that contains the defined CDRMs. All the CDRMs that follow a given NETWORK definition statement must belong to that network.

   The value coded for NETID on the NETWORK definition statement must match the NETID start option specified for the CDRM in its own domain. CDRM definition statements for CDRMs in the same network as this host

CDRM can follow a NETWORK definition statement whose NETID identifies the network that contains this host CDRM.

By coding a NETWORK definition statement for every network (including that of the host CDRM), you can use the same set of CDRM definitions at hosts in different networks. (However, these hosts must all have VTAM Version 2 Release 2 or later releases.)

2. Add one CDRM definition statement for each SSCP in another network with which the gateway VTAM is in session (this is required).

   The CDRM names defined in the gateway SSCP must be unique and must immediately follow their own NETWORK definition statement.

3. Add one or more GWPATH definition statements after each CDRM definition statement that defines an SSCP in another network. If you specify the GWSSCP=YES start option, you can use a GWPATH definition statement in conjunction with a CDRM definition statement. In this case, the value of the ELEMENT operand on the GWPATH definition statement must match that defined with a GWNAU definition statement in the gateway NCP through which the gateway SSCP is reached. If the gateway SSCP can be reached through multiple gateway NCPs, multiple GWPATH definition statements can be associated with the single CDRM definition statement defining the gateway SSCP. (This is required for CDRMs in other networks that are not predefined in the gateway NCP.)

   GWPATH definition statements define one or more cross-network session paths to be used for cross-network LU-LU and SSCP-SSCP sessions.

   If no GWPATH definition statement is coded, VTAM uses the SUBAREA value specified on the CDRM definition statement to select a gateway NCP. The ELEMENT value on the CDRM definition statement must match that specified by the GWNAU definition statement for a predefined CDRM in the gateway NCP.

   The ADJNETSA and ADJNETEL operands on the GWPATH definition statement must specify the subarea and element numbers of the CDRM as they are known in the adjacent network. For a back-to-back gateway NCP configuration, these numbers are obtained from the SUBAREA and ELEMENT operands on the GWNAU definition statement that defines this external CDRM in the second gateway NCP in the back-to-back configuration. For information on how the gateway path is selected, see “Gateway path selection” on page 587.

   The ADJNETCS operand on the GWPATH definition statement allows you to specify the Class of Service name to be used for establishing the route for the SSCP-SSCP session. Usually, this Class of Service name is ISTVTCOS. If you specify ADJNETCS, VTAM does not invoke the NetView alias name translation facility or the session management exit routine to perform Class of Service translation.

4. The value you code on the SUBAREA operand of the CDRM definition statement should be the subarea in the gateway NCP through which the gateway SSCP is reached. The ELEMENT operand for the CDRM definition statement must match the element predefined for the gateway VTAM with a GWNAU definition statement in that gateway NCP. If these operands are not defined, it is impossible to initiate an SSCP-SSCP session between the nongateway VTAM and a gateway SSCP in another network.

   If the gateway SSCP can be reached through multiple gateway NCPs, it appears as multiple CDRMs (that is, as SSCPs in different subareas) from the point of view of the nongateway SSCP. If a gateway NCP failure disrupts the SSCP-SSCP session between the nongateway SSCP and the gateway SSCP,

reactivation of the session through a different gateway NCP is possible if you provide multiple CDRM definition statements for the gateway SSCP. Each of these definition statements must have a unique label. The SUBAREA operand specifies the subarea of one of the gateway NCPs, and the element value is reserved for the gateway SSCP with a GWNAU definition statement coded in that gateway NCP subarea. Given these definitions, the operator at either the gateway SSCP or the nongateway SSCP can reestablish the SSCP-SSCP session through an alternate gateway NCP.

Figure 141 is an example of an SNI back-to-back configuration.

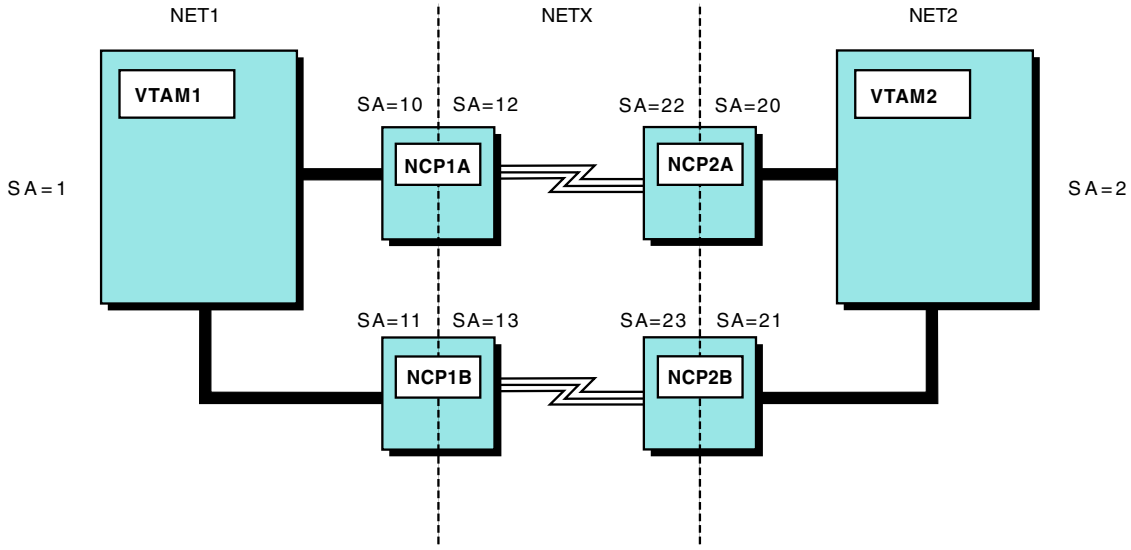


Figure 141. Multiple-network configuration: CDRM major nodes

Code the following in the CDRM major node in VTAM1:

```
NET1     NETWORK NETID=NET1
VTAM1    CDRM  SUBAREA=1,
:
NET2     NETWORK NETID=NET2
VTAM2    CDRM
NCP1A    GWPATH SUBAREA=10,
               ELEMENT=1,
               ADJNET=NETX,
               ADJNETSA=22,
               ADJNETEL=1,
NCP1B    GWPATH SUBAREA=11,
               ELEMENT=1,
               ADJNET=NETX,
               ADJNETSA=23,
               ADJNETEL=1,
```

It is recommended that you code the ELEMENT operand rather than accepting the default. In this way, it is easier to ensure that you do not have the same subarea and element number for multiple resources.

The CDRM major node in VTAM1 shows that you need to code a GWPATH definition statement for each gateway NCP. Note that you need to be careful to match the NCP pairs (NCP1A with NCP2A and NCP1B with NCP2B) to ensure that the CDRM-CDRM session uses the same pair of gateway NCPs.

Thus, code the following in the CDRM major node in VTAM2:

```
NET2     NETWORK NETID=NET2
VTAM2    CDRM  SUBAREA=2,
:
NET1     NETWORK NETID=NET1
VTAM1    CDRM
NCP2A    GWPATH SUBAREA=20,
               ELEMENT=1,
               ADJNET=NETX,
               ADJNETSA=12,
               ADJNETEL=1,
NCP2B    GWPATH SUBAREA=21,
               ELEMENT=1,
               ADJNET=NETX,
               ADJNETSA=13,
               ADJNETEL=1,
```

Defining cross-domain resources

Following are methods of defining resources in other networks:

- Use a CDRSC definition statement (or a model CDRSC definition statement) without an owning network specified.

  This method of defining CDRSCs can be used by hosts within either of the networks in which the session partners reside or by intermediate networks. The definition can be used to satisfy either a real or an alias name, but if the NetView alias name translation facility or the alias function in the session management exit is used, you cannot define a CDRSC for both. For example, in HOST1 (in Figure 143 on page 574), you can code the following:

  ```
  CDRSCSEG  VBUILD     TYPE=CDRSC,...
  TERM1     CDRSC      CDRM=HOST2,...
  ```

  To code the NQNMODE or LUALIAS operand for a CDRSC, the owning network must be specified.

- Use a CDRSC definition statement (or a model CDRSC definition statement) with an owning network specified.

  This method of defining CDRSCs can be used by any of the hosts involved in establishing the requested session. This method provides for the best session setup performance. For example, in HOST1 (in Figure 143 on page 574), you can code the following:

  ```
  CDRSCSEG   VBUILD     TYPE=CDRSC,...
             NETWORK    NETID=NETB
  TERM1      CDRSC      CDRM=HOST2,...
  ```

  This is the method to use if you want to define cross-domain resources by their network-qualified names. To define cross-network resources by only their network-qualified names, specify NQNMODE=NQNAME (in the definition or using the start option). To define cross-network resources by both their network-qualified names and their unqualified names, specify NQNMODE=NAME (in the definition or using the start option).

  The LUALIAS operand is valid only for cross-network resources with an owning network specified.

- Use a dynamically defined CDRSC.

  Instead of predefining CDRSCs, you can decide whether some or all CDRSCs are dynamically allocated and defined. The host that owns the logical unit initiating a session can dynamically define the requested session partner. In the intermediate SSCP, both the originating LU and the destination LU can be dynamically defined. In the destination LU SSCP, the originating LU can be

dynamically defined. Independent LUs acting as either the destination or the originating LU can be dynamically defined at either the destination or originating LU SSCP.

Dynamically defined CDRSCs take the NQNMODE value that is defined on the NQNMODE start option. The NQNMODE start option affects dynamically defined CDRSCs in the following way:

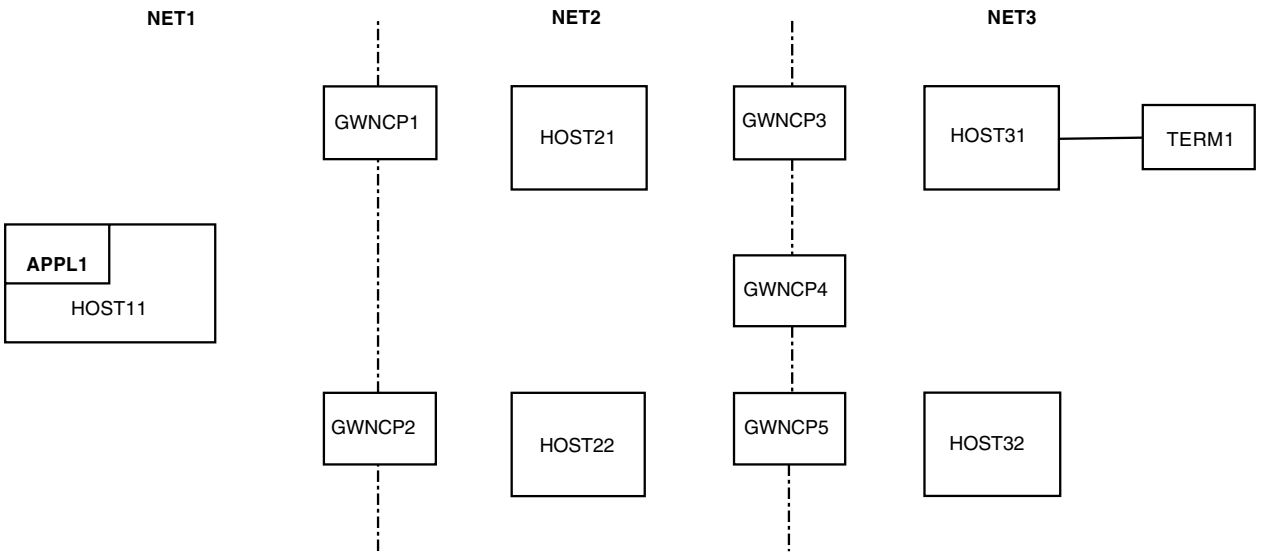
- If NQNMODE=NAME, a cross-network dynamic CDRSC is defined by its network-qualified name and by its alias name and alias network, even if the alias name and alias network are assumed (that is, the alias name is equal to the real name and the alias network is equal to the network of the session partner).
- If NQNMODE=NQNAME, a cross-network dynamic CDRSC is defined by its network-qualified name. It can also have a dynamic LUALIAS created for it if the session partner application specified LUAPFX on its APPL definition statement. A cross-network dynamic CDRSC is not defined by its assumed alias name and alias network unless the name portion of the alias name and the real name are different.

Session initiation request

In Figure 142 on page 572, assume that HOST11 and HOST21 are gateway VTAMs, and APPL1 wants to initiate a session with TERM1. The value of NQNMODE, either as a start option or defined on the CDRSC definition statement, affects the name by which the cross-network resource is known. If NQNMODE=NAME is specified as a start option or on the CDRSC definition statement for TERM1, because there is no TERM1 in NET1, TERM1 real name is TERM1 in NET3 and its alias name is TERM1 in NET1. (If a resource TERM1 actually existed in NET1, TERM1 in NET3 must be translated to a name that is not being used in NET1, such as TERM3 in NET1.)

If NQNMODE=NQNAME is specified as a start option or on the CDRSC definition statement for TERM1, TERM1 real name is NET3.TERM1. It is *not* known by the alias name NET1.TERM1. (If a resource TERM1 actually existed in NET1, and the session initiation request used a nonnetwork-qualified name, the session would be established with TERM1 in NET1. If the session initiation request used NET3.TERM1, the session would be established with TERM1 in NET3. Another option to using a network-qualified name in a session request would be to translate TERM1 to a name that is not being used in NET1, such as TERM3 in NET1.)

Note: A network-qualified session initiation request (for example, NET3.TERM1) cannot use the application ACBNAME unless it is the same as its APPL name.



Note: The connections between hosts and NCPs have been omitted for clarity. You can assume that each NCP connects to each host in the networks to the left and right of it.

Figure 142. Example of three interconnected networks

The information in the session initiation request that is flowing from HOST11 to HOST21 is one of the following:

- If APPL1 initiates a session for TERM1 with a nonnetwork-qualified name, and the CDRSC for TERM1 is dynamically defined or predefined using an alias name in HOST11, the session initiation request carries only the alias name (TERM1 in NET1).
- If APPL1 initiates a session for TERM1 using a network-qualified name (NET3.TERM1), or if the session initiation request is network qualified, the session initiation request carries both the alias name (TERM1 in NET1) and the real name (TERM1 in NET3).

The request flow described here assumes that GWCTL=SHR is in the host PCCU definition statement. If, according to the rules for gateway control, HOST11 is designated to request an alias address for GWN1, the real name for TERM1 must be properly determined in HOST11 by defining a CDRSC with NETWORK specified or by using the NetView alias name translation facility.

Name assumption

In Figure 142, assume that HOST21 is a gateway VTAM. Also assume that HOST11 sends only the alias name to HOST21 in the session initiation request. The following steps occur:

1. HOST21 dynamically defines a CDRSC for TERM1 or uses a predefined CDRSC.
2. HOST21 attempts to determine the real name of TERM1 using the NetView alias name translation facility or the session management exit routine.
3. HOST21 routes the session setup request to HOST31 using the adjacent SSCP tables, and the request leaves the first gateway.
4. When the request leaves the first gateway, VTAM does the following:

a. If the real name and network identifier are not determined using the NetView alias name translation facility or the VTAM session management exit routine, VTAM sends out a search using the alias name to find the real name and network identifier.
b. The alias CDRSC representation (TERM1 in NET1) is updated to reflect the real CDRSC representation (TERM1 in NET3).

5. In SSCP21, a CDRSC is defined or dynamically created for APPL1 also. Because APPL1 is the originating LU, its real name and network identifier are known. The predefined CDRSC can be defined with or without a network identifier.
6. HOST31 receives the session initiation request from HOST21, which contains both real and alias names for TERM1. HOST31 then determines that it owns TERM1.
7. HOST31 locates a CDRSC representation for APPL1 or dynamically defines one using the real name (APPL1 in NET1). If you have defined a CDRSC definition statement for APPL1 in NET3, HOST31 will use that CDRSC definition statement and will not use the dynamically defined CDRSC.
8. When the CDRSC is found, HOST31 merges any pertinent information from the alias CDRSC representation with the real representation.
9. If session initiation request processing is successful, HOST31 sends a positive response.
10. HOST11 receives the response, and further processing depends on how TERM1 is defined in HOST11.
    - If TERM1 is dynamically allocated, VTAM merges the acquired information (real network ID and owning SSCP name) into the existing CDRSC.
    - If TERM1 is predefined using its alias name, the predefined CDRSC representation is updated with the real name, network identifier, and owning SSCP.
    - If TERM1 is predefined using its real name and network identifier, no further processing is required. Session setup continues, and resources are recognized by their real names, alias names, and network identifiers.

If both the real and alias names are passed in the session initiation request, either a predefined real CDRSC representation for TERM1 in NET3 is located, or a dynamic CDRSC is allocated. A predefined CDRSC without a network identifier specified can be located. In any case, the resource name is not passed to the alias name translation facility to perform name translation. However, VTAM can invoke the facility to determine if the owning SSCP is defined to the alias name translation facility.

With interconnected networks, defining CDRSCs becomes more complex as cross-network sessions, alias name translation, and adjacent SSCP rerouting are introduced. The configuration in Figure 143 on page 574 shows two interconnected networks.

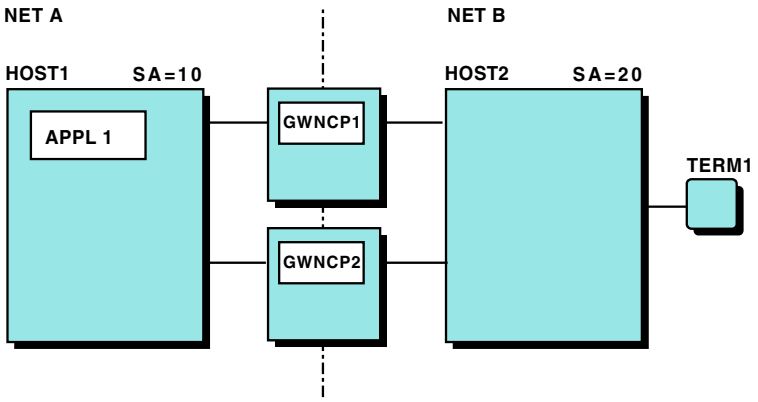


Figure 143. Example of two interconnected networks

To establish cross-network sessions between APPL1 in NETA and TERM1 in NETB, the following CDRM definitions are required if sessions are initiated from either HOST1 or HOST2. This CDRM major node is filed in HOST2 for the external CDRM HOST1.

```
         NETWORK   NETID=NETA
HOST1    CDRM      CDRDYN=YES,CDRSC=OPT
         GWPATH    GWN=GWNCP1,ADJNETSA=10,ADJNETEL=1,...
         GWPATH    GWN=GWNCP2,ADJNETSA=10,ADJNETEL=1,...
```

The following CDRM major node is filed in HOST1 for the external CDRM HOST2:

```
         NETWORK   NETID=NETB
HOST2    CDRM      CDRDYN=YES,CDRSC=OPT
         GWPATH    GWN=GWNCP1,ADJNETSA=20,ADJNETEL=1,...
         GWPATH    GWN=GWNCP2,ADJNETSA=20,ADJNETEL=1,...
```

Predefined cross-domain resources without network specification

If you do not code the NETWORK definition statement and you want to predefine CDRSCs (or model CDRSCs), you should specify the adjacent SSCP rather than the owning SSCP in the CDRM operand of the CDRSC definition statement.

Note: You cannot specify the NQNMODE or LUALIAS operand for cross-network resources without specifying a NETWORK definition statement.

The name can represent the alias or real name for the resource. When the real name and NETID are determined, VTAM updates the predefined CDRSC to represent the real name.

Note: If the real NETID is cross-network, no other same-network resource can be identified by the originally coded name.

When the last session for that CDRSC ends, VTAM restores the originally coded name so that the CDRSC is the same as that previously predefined. However, if NETID is coded, whether in this network or another, that indicates a real CDRSC.

You can define TERM1 (the destination LU in Figure 142 on page 572) in the following way:

```
         VBUILD TYPE=CDRSC,...
TERM1    CDRSC  CDRM=HOST21,...
```

The CDRM, HOST21, is not the owning SSCP. Rather, it is the SSCP adjacent to the origin logical unit owning SSCP (HOST11). Because the CDRSC is defined in HOST11 network (or is allowed to default), the resource name (TERM1) is assumed to be the name known in HOST11 network (that is, the alias name).

Following are some advantages of predefining CDRSCs using this method:

- Coding the CDRM operand on the CDRSC definition statement allows you to select the next SSCP in the session setup path. For simple networks that do not plan to provide alternate adjacent SSCPs, this method can eliminate the need for defining adjacent SSCP tables. For more complex configurations, this method of defining CDRSCs identifies the first SSCP that should be tried when alternate adjacent SSCPs are available. In this case, VTAM reorders the adjacent SSCP table you defined; the first SSCP in the table is now the SSCP named on the CDRM operand of the CDRSC definition statement, causing this CDRM to be used first for routing.
- For some configurations, new CDRSC definitions are not required. If a multiple-network configuration is the result of splitting a single network and if the owning CDRMs have not changed, the CDRSC definitions previously defined still cause the request to be routed to the correct SSCP.

The disadvantage of predefining CDRSCs with this method is that the length of the session setup path is increased, though not as much as with dynamically defined CDRSCs using the resource alias name (because no definitions have to be built dynamically). When the real SSCP name and NETID are determined, VTAM updates the predefined CDRSC to represent the real name. When the last session for that CDRSC ends, VTAM restores the alias name so that the CDRSC is the same as originally predefined and restores the CDRM if it is coded. The NQNMODE and LUALIAS operands are only valid for cross-network CDRSCs with with an owning network specified.

This method of predefining CDRSCs is most suitable for simple configurations (for example, where one network is split into two) or for hosts that rely on another host for gateway support. In the example, HOST11 passes all session requests to HOST21 or HOST22. Those hosts are then responsible for alias name translation, alias address requests, and adjacent SSCP rerouting.

Predefined cross-domain resources with network specification

Predefining CDRSCs (or model CDRSCs) with network specification involves including NETWORK definition statements in the CDRSC definition to identify the network in which the resource is located. When a CDRSC definition follows a NETWORK definition statement, VTAM recognizes the specified CDRM as the actual owning SSCP and not as the next SSCP on the session setup path. A session with that CDRM is required only if it is also the next SSCP on the session setup path. The CDRSC definition statement can optionally include the name of the CDRM that owns the logical unit represented by the CDRSC.

In the following example, which is used to define TERM1 in HOST11, TERM1 is defined to be in NET3, and its owning CDRM is HOST31.

```
         VBUILD TYPE=CDRSC,...
         NETWORK NETID=NET3
TERM1    CDRSC CDRM=HOST31,...
```

VTAM recognizes that HOST31 is the actual owning SSCP and not the next SSCP on the session setup path.

Following are some advantages of using this method:

- In a configuration with no name conflicts, this method defines the shortest session setup path. However, unless you are using NQNMODE=NQNAME, a resource, even when defined with a NETID, cannot be predefined if its name is not uniquely known in the host network. If you are using NQNMODE=NQNAME, a resource, when defined with a NETID, can be predefined even if the name is duplicated in another network.

  This method of predefining CDRSCs is designed for those systems that want to quickly locate a resource and use its real name. Therefore, the name must be uniquely known within the network in which it is defined. If the name duplicates a name in an interconnected network, it must be using its network-qualified name.
- Predefining CDRSCs in this way allows you to define very specific adjacent SSCP tables. Because the NETID and, optionally, the destination SSCP are known, you can define adjacent SSCP tables for a specific NETID or NETID and SSCP name. This can eliminate the VTAM need for a default SSCP list, which might not provide the shortest session setup path.
- It adds security by restricting only authorized SSCPs to access resources in this host, assuming no dynamic CDRSCs are permitted.
- You can use NQNMODE=NQNAME to enable CDRSCs in different networks to have the same name.

Following are some disadvantages:

- If you specify the VFYOWNER=YES operand on the CDRSC definition statement, changing CDRMs in a backup situation can be complex. That is, if you have coded the CDRM operand on the CDRSC definition statement, VTAM assumes that it is the owning SSCP name. The operator must issue a MODIFY CDRM operator command in each network that contains a CDRSC definition, with the previous CDRM listed as the owner. Then (as in Figure 142 on page 572), if HOST31 fails and TERM1 is acquired by HOST32 but the CDRM is not updated in HOST11, the session setup attempt fails because HOST31 is still defined as the destination of the CDRM owning the logical unit . By specifying VFYOWNER=NO on the CDRSC definition statements along the route, VTAM does not verify if that CDRM is the owner but assumes that the session request contains the backup owner and thus routes the session request to that CDRM. Then, only the operators at the session origin need to modify the CDRSC owner to the backup CDRM owner.
- If you are not using NQNMODE=NQNAME, this method does not work for interconnected networks with duplicate names. The method requires that all the connected networks have unique resource names. In many cases, renaming and predefining CDRSCs using NETID or using network-qualified names is preferable to using the alias name translation facility.

This method is recommended for networks that contain unique names and require fast session setup.

If the CDRSC moves randomly among networks, specifying VFYOWNER=NO allows session setup to succeed, but added security is lost in that the intermediate VTAMs on the session setup path do not verify the owning host.

Dynamic cross-domain resources

Instead of predefining CDRSCs, you can let VTAM dynamically allocate and define some or all of them.

Dynamic CDRSC definition is used when VTAM receives a request to establish a session from or for an undefined cross-domain resource or for directory search requests. VTAM handles the request by dynamically creating a temporary CDRSC definition for the cross-domain resource. Without dynamic definition of CDRSCs, the number of resources requiring definition can be extremely large. If you provide networking services within an enterprise or for external users, this method is recommended.

For example, in Figure 142 on page 572, if you choose not to predefine a CDRSC for TERM1 and the CDRM definition for HOST11 and HOST31 is set up to allow dynamic CDRSC definition, the following occurs:

1. A CDRSC for TERM1 is dynamically allocated in HOST11.
2. Unless you are using network-qualified names, if duplicate resource names exist, VTAM must select an alias name to refer to the session partner located in the other network. For example, if TERM1 exists in both NET1 and NET3, VTAM chooses an alias name whereby NET1 refers to the TERM1 located in NET3. VTAM uses the NetView alias name translation facility or the VTAM session management exit routine to identify the network in which the session partner resides. Then, VTAM resolves the alias name to the name used in the session partner network. Optionally, the NetView alias name translation facility can be used to obtain the name of the SSCP that owns the session partner.

   If you are using network-qualified names, the NETID uniquely identifies the resources, and VTAM does not have to select an alias name for the session partner located in the other network.
3. VTAM uses an adjacent SSCP table to locate the next SSCP to which the session-setup request should be sent to reach the destination network.

When the session-initiation request leaves the first gateway (that is, the gateway NCP connecting NET1 and NET2), VTAM sends out a search to find the real name and NETID to replace the alias name for the destination logical unit if the alias name has not yet been translated into a real name and NETID. For information on how this affects routing, see “Name assumption” on page 572.

If no translation takes place in the configuration in Figure 142 on page 572, the session initiation request means that the destination network is assumed to be NET3 when the request is sent to either HOST31 or HOST32.

Following are some advantages of using this method:

- The number of CDRSC definition statements required in each host is greatly reduced. VTAM storage requirements are also reduced because only needed definitions exist.
- After the real name and network are known, the lack of predefined CDRSCs does not affect the length of the session setup path because the dynamic CDRSCs are retained for a user-specified time interval (CDRSCTI start option) after the last session terminates.

Following are the disadvantages of using this method:

- The main disadvantage of eliminating predefined CDRSCs is that the length of the session setup path for unqualified session requests is increased before VTAM identifies the destination logical unit real name and network. In this case, a CDRSC is dynamically allocated using the alias name. VTAM must assume that the name supplied to it in a session request is an alias name. After the real name is determined (by using the alias name translation facility or the alias function of the session management exit routine or from the session initiation response), the

acquired information (the real name and network) is merged into the existing dynamically allocated CDRSC. Unless an alias name translation facility that supplies the name of the SSCP owning the session partner is used, each adjacent SSCP leading to the destination network is tried until the session is established or it is determined that none of the SSCPs owns that requested session partner. This process is repeated by each gateway-capable SSCP that receives the session initiation request.

- If an inactive LU is found which represents the destination LU, VTAM can allocate a dynamic CDRSC to enable the session initiation request to be sent to other VTAMs. The decision to allocate the dynamic CDRSC depends on several factors. If the session initiation request specified that the SSCP owning the destination LU was this host, the request fails. Also, if the LU is not eligible to be made into a shadow resource, the session is rejected.
- An adjacent SSCP table is required if you want to use dynamic CDRSC definition for destination LUs. You can predefine this table or specify the DYNASSCP start option to have VTAM create a dynamic table.

Using this method of dynamically allocating CDRSCs is most beneficial for intermediate and destination networks along the session setup path; the destination logical unit real name and network identifier are known by the time the session request reaches these hosts, so replacing CDRSCs is not required. Without dynamic definition of CDRSCs, the number of resources requiring definition can be extremely large. If you provide networking services within an enterprise or for external users, this method is recommended.

If the dynamic CDRSC definition is authorized, VTAM does the following:
- Builds a CDRSC minor node definition to represent the undefined LU
- Records the owning CDRM name (when determined)
- Initializes LU information from the session-initiation request or response

Dynamically defined CDRSC minor nodes are collected in a CDRSC major node named ISTCDRDY. ISTCDRDY is activated automatically during VTAM initialization, and deactivated automatically during VTAM termination. ISTCDRDY can be deactivated and activated by an operator command; this causes all currently active dynamic CDRSCs to be deleted. In addition, all sessions involving dynamically created CDRSCs (including CP-CP sessions with this host, if the partner CP was dynamically defined) are terminated.

While allowing dynamic CDRSC definition for session partners in another domain, you can still restrict access to application programs using the following:
- Session management exit routine (refer to *z/OS Communications Server: SNA Customization*)
- Session authorization exit routine (refer to *z/OS Communications Server: SNA Customization*)
- Application program logon exit routine (refer to *z/OS Communications Server: SNA Programming*)

Dynamically defined CDRSCs are deactivated and deleted by VTAM on a periodic basis if they are not in use. That is, if a CDRSC has no active sessions and has had none for a defined interval of time, the definition is discarded. This interval is set by the CDRSCTI start option of VTAM.

A dynamically defined CDRSC that becomes a shadow resource has all of its sessions moved to the corresponding LU definition. Because of this, it is considered to have no active sessions as soon as it becomes a shadow resource and is then deleted.

If dynamically defined CDRSCs are used, initial session initiation time can be slightly longer unless network-qualified names are used in session initiation requests.

Defining adjacent SSCPs

This section discusses adjacent SSCP tables that are used for cross-network session requests. For general information about adjacent SSCP tables, see "Adjacent SSCPs" on page 541.

You can use adjacent SSCP tables in a multiple-network environment to do the following:

- Reach destination SSCPs in other networks
- Determine the next SSCP in the session-initiation path to reach a destination network for an LU-LU session

Types of adjacent SSCP tables

There are three types of SSCP tables that are used for cross-network session requests:

Network-specific list
The network-specific list has a NETWORK definition statement coded in the adjacent SSCP table, and the NETID operand specifies the NETID of the destination network.

Network-specific and CDRM-specific list
The network-specific and CDRM-specific list specifies both a destination network and a destination CDRM.

Default list
The default SSCP list is used whenever one of the following conditions is true:

- The destination network is not known.
- The destination network is known, but there is not a network-specific table associated with the network.
- The destination network and owning SSCP name are known, but the specific table cannot be located.

When a CDRSC network is known, but its owner is not, VTAM uses the network-specific adjacent SSCP table for that network. If none is defined, the default is used.

The default SSCP list is part of the adjacent SSCP table. You can include network-specific SSCP lists for each network and a default list for use when the real network is not known.

Deciding whether to code adjacent SSCP tables

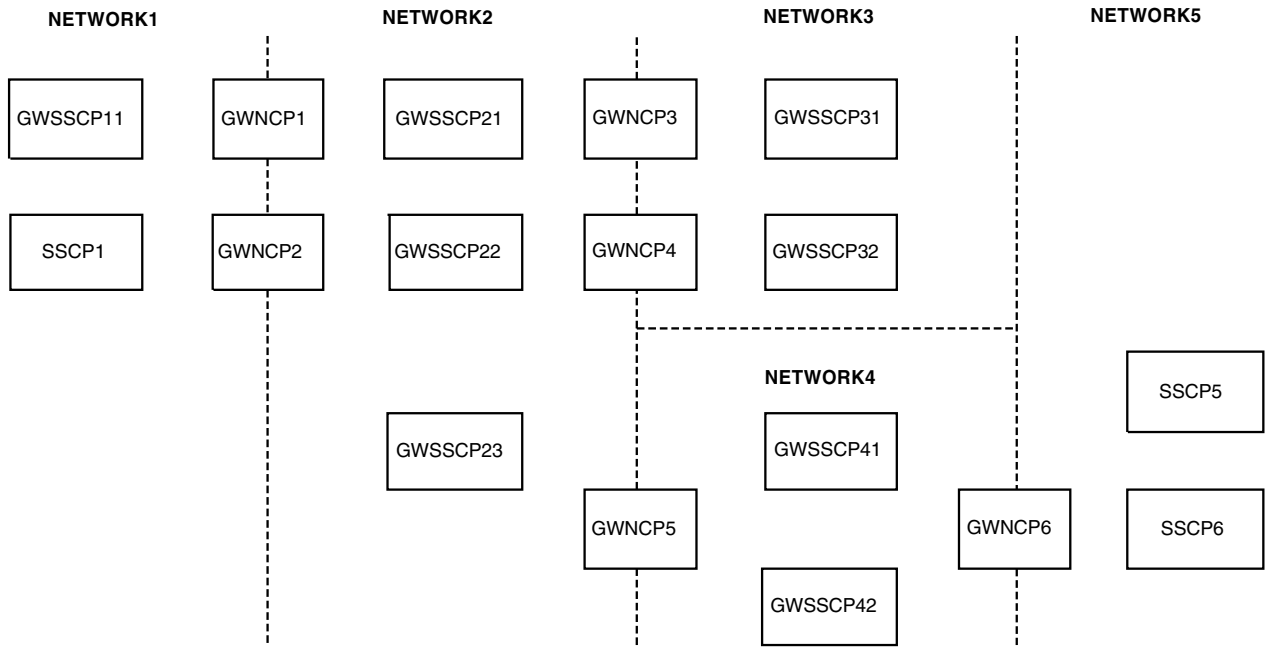
A host does not need an adjacent SSCP list for a network if the following are true:

- The host SSCP is directly in session with the destination SSCP (that is, there are no intermediate SSCPs on the session path), and the host SSCP knows the destination SSCP name.
- Predefined CDRSCs without NETWORK, but with CDRM, are specified.
- An alias name translation program or the alias function of the session management exit routine is used to find the destination SSCP name.

Note: You might need to define adjacent SSCP tables containing the type 2.1 network ID for session requests destined for a nonnative network logical unit.

Sample of adjacent SSCP tables for a multiple-network environment

The sample adjacent SSCP table definitions included below are written for the configuration in Figure 144.



Note: The connections between hosts and NCPs have been omitted for clarity. You can assume that each NCP connects to each host in the networks it joins.

Figure 144. Multiple-network configuration

The adjacent SSCP table in the following example applies to GWSSCP21 in NETWORK2 and contains default SSCP lists for all the networks shown in Figure 144.

```
TABLE2   VBUILD   TYPE=ADJSSCP
*************************************************************************
*  DEFAULT SSCP LIST: THIS LIST IS USED WHEN THE CDRM                   *
*                     IS UNKNOWN AND NETWORK IS UNKNOWN                 *
*************************************************************************
         NETWORK                    UNNAMED NETWORK DEFINITION
GWSSCP11 ADJCDRM                    WITH DEFAULT ADJACENT SSCP
GWSSCP31 ADJCDRM                       LIST FOR ALL NETWORKS
GWSSCP32 ADJCDRM
GWSSCP41 ADJCDRM
GWSSCP42 ADJCDRM
```

```
*
*********************************************************************
*  DEFAULT SSCP LIST FOR NETWORK1: USED WHEN THE CDRM IS UNKNOWN       *
*                                  AND NETWORK IS KNOWN TO BE NETWORK1 *
*********************************************************************
         NETWORK  NETID=NETWORK1   DESTINATION NETWORK
GWSSCP11 ADJCDRM                   ADJACENT SSCP FOR NETWORK1
*
*********************************************************************
*  DEFAULT SSCP LIST FOR NETWORK2: USED WHEN THE CDRM IS UNKNOWN       *
*                                  AND NETWORK IS KNOWN TO BE NETWORK2 *
*********************************************************************
         NETWORK  NETID=NETWORK2   DESTINATION NETWORK
GWSSCP22 ADJCDRM                   DEFAULT ADJACENT SSCP LIST
GWSSCP23 ADJCDRM                     FOR NETWORK2
*
*********************************************************************
*  DEFAULT SSCP LIST FOR NETWORK3: USED WHEN THE CDRM IS UNKNOWN       *
*                                  AND NETWORK IS KNOWN TO BE NETWORK3 *
*********************************************************************
         NETWORK  NETID=NETWORK3   DESTINATION NETWORK
GWSSCP32 ADJCDRM                   DEFAULT ADJACENT SSCP LIST
GWSSCP31 ADJCDRM                     FOR NETWORK3
*
*********************************************************************
*  DEFAULT SSCP LIST FOR NETWORK4 OR NETWORK5: CDRM IS UNKNOWN AND     *
*                                  NETWORK IS KNOWN TO BE NETWORK4 OR 5*
*********************************************************************
         NETWORK  NETID=NETWORK4   DESTINATION NETWORK
         NETWORK  NETID=NETWORK5   DESTINATION NETWORK
GWSSCP41 ADJCDRM                   DEFAULT ADJACENT SSCP LIST
GWSSCP42 ADJCDRM                    FOR NETWORK4 AND NETWORK5
*
*********************************************************************
*  CDRM-SPECIFIC LIST FOR SSCP5: CDRM IS KNOWN                        *
*                                  TO BE SSCP5 IN NETWORK5            *
*********************************************************************
SSCP5    CDRM                      DESTINATION SSCP
GWSSCP42 ADJCDRM                   ADJACENT SSCP LIST
GWSSCP41 ADJCDRM                     FOR SSCP5 IN NETWORK5
```

Example default lists

In the above adjacent SSCP table, the default SSCP list for all networks is:

GWSSCP11
GWSSCP31
GWSSCP32
GWSSCP41
GWSSCP42

This list is used whenever the following are true:

- The destination logical unit network is not known.
- The destination network ID is known, but no entries correspond to the destination network.

The default SSCP list for NETWORK4 and NETWORK5 is as follows:

GWSSCP41
GWSSCP42

This list is used whenever VTAM knows the destination network to be either NETWORK4 or NETWORK5, but does not know the destination SSCP name.

If the destination SSCP is known to be SSCP5 in NETWORK5, the following adjacent SSCP list is used instead:

GWSSCP42
GWSSCP41

Example network-specific lists

The rest of the adjacent SSCP table is used when VTAM sets up sessions with CDRSCs whose owning CDRM and network are known. For example, if GWSSCP21 has the following definition statement coded in its CDRSC major node:

```
        NETWORK  NETID=NETWORK3
TSO02   CDRSC    CDRM=GWSSCP32
```

Because the CDRM is coded on the CDRSC definition statement, VTAM assumes that this CDRM is the true owner of the resource. Because the NETWORK definition statement is coded, VTAM uses the coded CDRM and the network-specific SSCP table to try to locate the CDRSC.

VTAM first builds an adjacent SSCP table for the CDRSC. The CDRM is defined as SSCP1, so GWSSCP32 is the first entry in the list. The rest of the list consists of the SSCPs that are adjacent to GWSSCP32, as defined in the default portion of the sample table:

```
GWSSCP32 CDRM
GWSSCP31 ADJCDRM
```

If the session setup fails through GWSSCP32, VTAM routes the request through GWSSCP31.

If no network had been coded for TSO02, VTAM would have used the default SSCP list (GWSSCP11, GWSSCP31, GWSSCP32, GWSSCP41, GWSSCP42). This could result in some session setup requests being routed outside the network through the gateway SSCPs. The adjacent SSCP table can thus prevent time and resources from being used in a search that might be unnecessary. The session management SSCP selection function can also be used to alter the SSCP list.

For VTAM Version 3 Release 2 and later releases, an ADJSSCP table defined with a NETWORK definition statement for the host network is used only when the destination resource is known to reside in the host network. If a destination resource is known to reside in NETWORK2, the list of ADJSSCPs used for a session request is:

GWSSCP22
GWSSCP23

Prior to VTAM Version 3 Release 2, an ADJSSCP table defined with a NETWORK definition statement for the host network is treated as a default table.

Overriding the SSCP lists

If a list of adjacent SSCPs exists for a destination network, an SSCP, or both, and if VTAM already has an SSCP-SSCP session with the destination SSCP, session setup processing can override the list; that is, the destination SSCP becomes the first choice for the specific session request. This override is temporary and does not change the actual adjacent SSCP table that you define.

Thus, the default SSCP list (TABLE2) includes the following definition statements:

```
         NETWORK  NETID=NETWORK3  DESTINATION NETWORK
GWSSCP32 ADJCDRM                  DEFAULT ADJACENT SSCP LIST
GWSSCP31 ADJCDRM                   FOR NETWORK3
```

Assume that a cross-network SSCP-SSCP session exists between GWSSCP21 and GWSSCP31. If GWSSCP21 receives a cross-network Initiate request for a session with a logical unit owned by GWSSCP31 in NETWORK3, the list of adjacent SSCPs in the table (GWSSCP32 and GWSSCP31) is reordered for this session setup as follows:

GWSSCP31
GWSSCP32

The actual table you define is unaltered.

The destination SSCP name, GWSSCP31, is available in the Initiate request if the alias name translation facility or a predefined CDRSC with CDRM name is used.

Request routing

Session initiation and INQUIRE APPSTAT requests are routed from the originating LU SSCP to the destination LU SSCP using trial-and-error routing. There are several different means of determining the list of adjacent SSCPs to try when performing trial-and-error routing. However, after the list is determined, the processing is very similar for each of these RUs.

Rerouting of requests

If the adjacent SSCP to which a cross-network session setup request is routed does not own the destination logical unit, it reroutes the request, provided that the following are true:

- An appropriate adjacent SSCP is available.
- The maximum SSCP rerouting count has not been reached.
- This SSCP is gateway capable.

Otherwise, the request is rejected.

If the SSCP that sent the session setup request is in the list, the entry is not used to reroute the request. This prevents a request from looping. The actual table you define is unaltered.

Rules for cross-network session request routing

The rules for routing cross-network session requests are as follows:

- The request is not routed if its visit count (SSCP rerouting count) has reached 0.
- The request is never routed back to the SSCP from which it was received.
- The request is never routed back to its originating SSCP.
- The request is never routed in this network if it was received from an SSCP in this network.
- An SSCP receiving a session request from another network can route the request to a different SSCP in that network through a different gateway NCP. It cannot route the request back using the same gateway NCP.
- The request received by a nongateway SSCP cannot be rerouted by that SSCP.
- The cross-network request received by a gateway-capable SSCP can be rerouted by that SSCP.
- A nongateway SSCP can perform trial-and-error routing for a session that it initiated.
- If the real NETID of the DLU is known, routing is not attempted to a nongateway SSCP with a different NETID, unless the adjacent SSCP supports the nonnative network connection function.

- A request is not routed to an SSCP that does not support the required function. (For example, automatic logon sessions are not routed to SSCPs that do not support automatic logon sessions.)

Dynamically defined CDRSCs and adjacent SSCP tables

If dynamic definition of resources is allowed and if no predefined CDRSC is located, VTAM automatically creates a CDRSC for the logical unit as part of routing. After the owning SSCP is found by trial-and-error routing of the session initiation request, the dynamically defined CDRSC is updated with the real network ID and owning SSCP name.

For releases of VTAM prior to Version 4, if two hosts attempt to establish a cross-network session with a logical unit that is not predefined in either of the initiating hosts, a conflict can arise in any intermediate host that receives these requests. The receiving host will then issue a sense code indicating duplicate DLU names. The sessions can be successfully established if the requests can be rerouted. This situation can be avoided, however, by predefining the DLU with NETID or by using the NetView alias name translation facility or the alias function of the session management exit routine to define the name of the resource instead of allowing it to be dynamically defined.

Alias name translation and adjacent SSCP tables

The NetView alias name translation facility can be used to determine the owning SSCP for a requested resource in the same network. To use this facility for an LU, define the LU and its owning SSCP to the alias name translation facility, because the alias name translation facility cannot translate a name as known in a network to a different name in the same network.

Define each alias name to be the same as the real name. During a session setup request, if VTAM does not have a definition of a destination logical unit or does not know the owning SSCP for the resource, VTAM can invoke the alias name translation facility with the name of the logical unit. The alias name translation facility returns the name of the SSCP defined as the owner of the logical unit, as well as the destination network identifier and the logical unit alias name in that network. Because this system has only one network, the network identifier is the VTAM network identifier, and the alias name is the same as the real name. Thus, the alias name translation facility is used as a directory to locate the owning SSCP and can direct the session setup request to the proper VTAM.

Cross-network routing

In a multiple-network environment, application programs and terminals managed in one network can access SNA resources controlled and managed in another network, but each network can preserve its autonomy and maintain unique network characteristics, such as the network address structure and network naming conventions.

Network address structures

When SNA networks are interconnected, each can maintain its current network addressing scheme. However, there are some addressing considerations.

Maintaining address structures

The network addresses assigned to nodes (host processors or communication controllers) in one network can be duplicated in other connected networks, as can the following subarea addressing capabilities:

- Extended subarea addressing
- Extended network addressing
- Nonextended network addressing (MAXSUBA)

The maximum number of addressable network nodes or subareas does not have to be consistent or compatible across the interconnected networks.

SNA network interconnection (SNI) eliminates the architectural requirement that interconnected networks must use a common network address structure with all unique network addresses. Instead, it permits each network to establish its own network addressing capabilities.

Addressing considerations for interconnected networks

Any network that has greater addressing capability than its adjacent networks must have extended subarea addressing-capable gateway NCPs and gateway SSCPs that are involved in setting up cross-network sessions for high subareas. Specifically, consider the following:

- The gateway NCP addressing capability in each network must be at least equal to the largest subarea address in the network, and its subarea address in the network must be within the range supported by the subarea with the lowest addressing capabilities. This restriction applies only for subareas that are using the gateway NCP for cross-network sessions.
- The gateway VTAM addressing capability must be greater than or equal to the largest subarea address in the network, and its subarea address must be within the range supported by the SSCP with the lowest addressing capabilities. That is, the gateway SSCP must be capable of supporting a CDRM session with any SSCP in the network, and it must be capable of owning the gateway NCPs it is to control.
- To properly support communication between networks where either network contains subareas that support cross-network sessions and that have subarea addresses above the capabilities of any of the other subareas involved in supporting or setting up the cross-network session, the gateway SSCPs in each network must be capable of supporting extended subarea addressing.
- For an extended subarea addressing gateway NCP to successfully send a CONTACT request unit to an adjacent cross-network gateway NCP with a subarea address greater than 255, the link station must be owned by an extended subarea addressing-capable VTAM.

Network naming conventions

SNA requires each terminal and application program (logical unit) to be identified by a unique, network-qualified name. Network-qualified names are used by specifying the NQNMODE=NQNAME start option or defining cross-network resources with the NQNMODE=NQNAME operand on the CDRSC definition statement.

If you choose not to use network-qualified names, you can use the optional facility SNI provides for translating terminal and application program names to cross-network (alias) names to avoid name duplication conflicts. By using this facility, the name structure used in each network can be maintained. A common network naming convention among the interconnected networks does not have to

be established. If multiple SNA resources are known by the same name in different networks, the name conflict can be resolved by using one or a combination of the following:

- The NetView alias name translation facility
- The VTAM session management exit routine
- Network-qualified names

If you choose to fully implement network-qualified names, you do not need to use either the NetView name translation facility or the VTAM session management exit routine to resolve duplicate names.

Controlling paths for interconnected networks

For SNA network interconnection (SNI), paths are defined the same as for single- and multiple-domain environments within each network (see "Network routing for subarea nodes" on page 348 and "How to plan routes in your network" on page 357). Rules for explicit and virtual routes do not cross network boundaries. For example, in Figure 145 assume that you are setting up a session between LU1 and LU2 and that they are in the same network. The same explicit route that is used from HOST1 to GWNCPB (ER0) would have to be used from GWNCPB to HOST2. However, because HOST1 and HOST2 are in different networks, the two explicit route definitions are independent of each other.

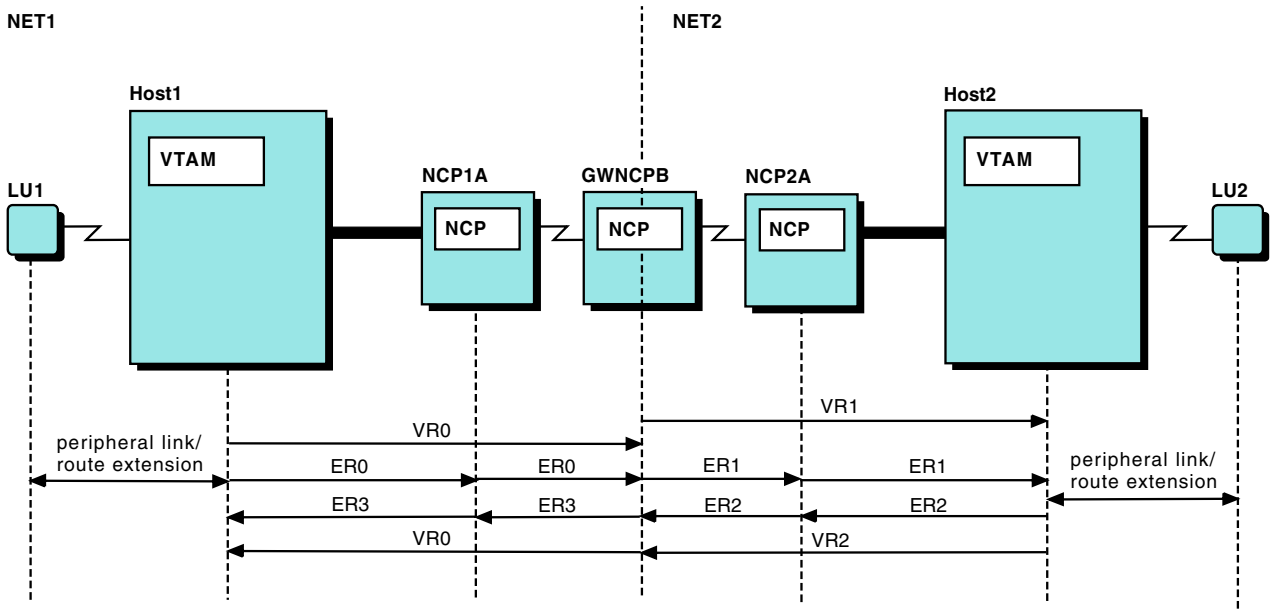


Figure 145. Multiple-network paths

Paths do not cross network boundaries. Cross-network routing requires separate paths in each network, each having the gateway NCP as the endpoint. If you are combining networks, you might have to add paths in nongateway NCPs that go to the gateway NCP.

If you are splitting an existing network into two networks, redefine the paths so that each path passing through a gateway NCP is defined separately in each of the networks.

Gateway path selection

For cross-network sessions, the gateway path selection function of the session management exit routine can be used to modify a list of alternate gateway paths to distribute LU-LU sessions among two or more gateway NCPs. You can shorten the list by deleting entries or reorder it by putting the preferred paths at the top of the list.

If you do not provide this function in your session management exit routine, VTAM uses the list of gateway paths determined by the GWPATH definition statements. However, the gateway NCP used for the SSCP-SSCP session is always the first one used for LU-LU sessions. For information about using the session management exit routine, refer to *z/OS Communications Server: SNA Customization*.

A gateway path is selected by one of the following methods:

SSCP-SSCP sessions

Each successive gateway path between two VTAMs is tried until either the SSCP-SSCP session is established or all of the possible session paths have been tried unsuccessfully.

The class of service name that is normally used is the COS name for SSCP sessions, ISTVTCOS. This is the default name that is used for establishing a cross-network SSCP-SSCP session and, if used, is the COS name that is passed to the NetView alias name translation facility or the VTAM session management exit routine. You can specify a different Class of Service name by coding the value on the ADJNETCS operand on the GWPATH definition statement. VTAM uses that COS name to locate the virtual route list in the proper COS table as specified on the COSTAB operand of the BUILD or NETWORK definition statements in NCP. If the ADJNETCS operand is specified, VTAM does not invoke the NetView alias name translation facility or the VTAM session management exit routine. For more information about network routing in a multiple-network environment, see "Cross-network routing" on page 584.

LU-LU sessions

Each gateway NCP (defined by a GWPATH definition statement) between the two VTAMs is tried until one of the following occurs:

- The requesting logical unit receives a positive response to the Initiate request for the session, meaning that the requested session partner has been located.
- All the possible gateway NCPs (defined by the GWPATH definition statements for the CDRM in another network) have been tried unsuccessfully.
- A failure not related to the GWNCP is detected.

If multiple gateway paths are used, VTAM selects a particular gateway NCP for each LU-LU session. The gateway NCP used for the cross-network SSCP-SSCP session is always the one VTAM attempts first for LU-LU session setup. If this is unsuccessful, the gateway VTAM selects another gateway NCP for the session. If you coded a session management exit routine to shorten or reorder the list of gateway NCPs, the gateway VTAM uses that altered list to establish the LU-LU session path.

During session establishment of a cross-network LU-LU session, the gateway VTAM that is controlling gateway path selection gets the list of alternate gateway NCPs defined by the GWPATH definition statements. If the BIND RU that flows between the requesting logical unit and the session partner fails because a virtual

route required by the session cannot be activated, no further gateway path selection is attempted. VTAM messages are issued to inform the operator that session establishment failed, and the requesting logical unit is informed.

Coding the gateway path

The GWPATH definition statement is required for all gateway VTAMs that initiate cross-network SSCP-SSCP sessions. If you do not code GWPATH definition statements following a CDRM definition statement for a VTAM in another network, code the SUBAREA operand on that CDRM definition statement so that VTAM can determine the gateway NCP to use.

Dynamic cross-domain resource definition must be allowed for the host CDRM if the NetView alias name translation facility or the alias function in the session management exit is used to translate logical unit names.

Handling class of service tables

Following are several important issues to consider when using COS tables for interconnected networks:

- Using COS tables for gateway VTAMs and gateway NCPs
- Using multiple identical COS tables (that is, using one COS table for more than one network or for more than one subarea node within a network)
- Using different COS tables for interconnected networks
- Using conflicting COS table names

You can define different COS tables for VTAM to use within its own network or within gateway NCPs that it owns. When the PLU is in the VTAM subarea, VTAM uses ISTSDCOS to resolve the Class of Service name to a virtual route list. The virtual routes originating in the host subarea extend to the SLU subarea node or to a gateway NCP providing a path to the SLU.

When VTAM is acting as a gateway VTAM that has been designated to resolve COS names for a gateway NCP, VTAM uses the COS table named on the NETWORK or BUILD definition statement in the gateway NCP generation definition (whichever applies to the adjacent network to which you are routing). The gateway VTAM passes the virtual route list to the gateway NCP, and the gateway NCP performs the route activation for one of the routes in the resulting virtual route list within that network.

To establish a cross-network SSCP-SSCP session, VTAM uses the ISTVTCOS Class of Service name to establish a route to the adjacent network SSCP. On the GWPATH definition statement that identifies the gateway NCP to be used for the session, you can specify the ADJNETCS operand to identify a COS name other than ISTVTCOS. VTAM uses this COS name to determine the specific route in the adjacent network for the session.

All of the COS tables to be used for gateway NCPs controlled by a gateway VTAM must be available to that SSCP. If VTAM attempts to load a COS table that it cannot find, session requests associated with the table fail.

If more than one gateway VTAM shares control of a gateway NCP, those gateway VTAMs can be defined in such a way that some of them allow the others to resolve the COS name. This is specified by the GWCTL operand on the PCCU definition statement. In this case, only those gateway VTAMs responsible for resolving COS names for that gateway NCP must have the COS tables stored in their hosts.

For example, in Figure 146, if Host1 has GWCTL=ONLY coded on the PCCU definition statement for GWNCPB, Host1 COS table is used for an SSCP-SSCP session between Host1 and Host2. If Host1 and Host2 have GWCTL=SHR coded on the PCCU definition statements for GWNCPB, the originating VTAM COS table is used.

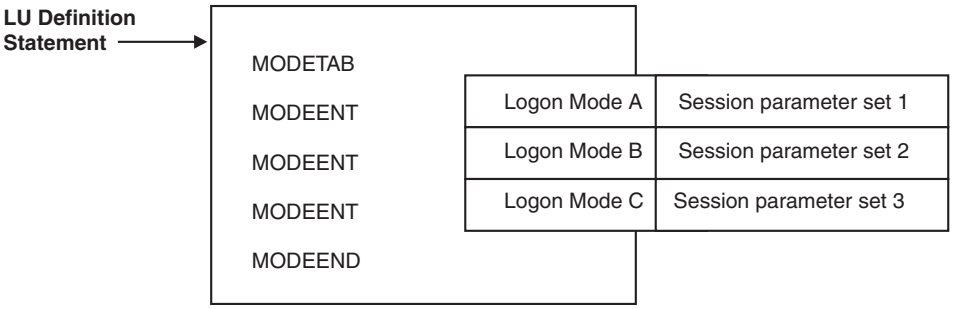


Figure 146. COS resolution in a multiple-network environment

The same COS table can be used for more than one network. You can also create a universal COS table to be used for all networks. This is possible if the COS names are the same and if virtual routes originating in different subarea nodes have identical VR numbers for routes providing the same levels of service.

If the same COS table can be used in more than one network connected to a gateway NCP, code the same table name on the COSTAB operands for the BUILD or NETWORK definition statements representing those networks in the NCP major node. VTAM loads only a single copy of the table into the VTAM storage.

If your interconnected networks require different COS tables, you can associate these table names with their proper networks by coding the table names on the COSTAB operands of the appropriate BUILD or NETWORK definition statements in the NCP generation definition.

The COS table for routes originating in a VTAM host must be named ISTSDCOS. If the same COS table can be used for a gateway NCP that is controlled by a different VTAM host, the COS table must be stored in the other VTAM host under a name other than ISTSDCOS (unless that VTAM can also share the same COS table for routes originating in that host).

In Figure 147 on page 590, networks NETA and NETB are connected in a back-to-back configuration.

The session between the PLU in NETA and the SLU in NETB uses the following three routes:

- A route within NETA from the PLU subarea to the gateway NCP (GWN1) subarea in NETA
- A route from the NETX subarea within GWN1 to the NETX subarea within GWN2
- A route from the NETB subarea within GWN2 to the SLU subarea

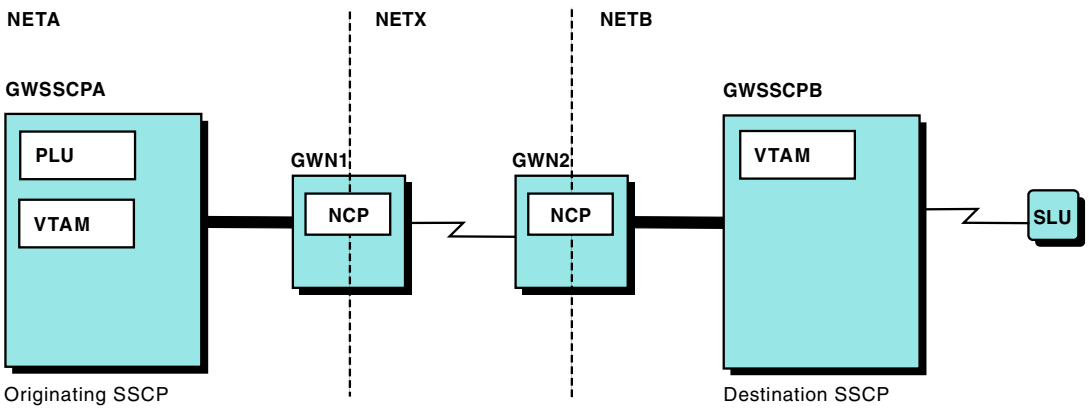


Figure 147. COS tables and routing in an SNI back-to-back configuration

Assuming that GWN1 resources reside in NETA (that is, NETA is GWN1 native network), GWN1 generation definition contains the following:

```
BUILD    NETID=NETA,...,COSTAB=COSNETA1
⋮
NETWORK  NETID=NETX,...,COSTAB=COSNETX1
```

Also, assuming that GWN2 native network is NETB, GWN2 generation definition contains the following:

```
BUILD    NETID=NETB,...,COSTAB=COSNETB2
⋮
NETWORK  NETID=NETX,...,COSTAB=COSNETX2
```

The following COS tables are used:

- In GWSSCPA, the SSCP of the PLU uses ISTSDCOS to select the list of virtual routes in NETA from which a route is selected for the session.
- GWSSCPA, as the gateway VTAM controlling GWN1, resolves the COS name for NETX to a list of virtual routes in NETX. It uses the table COSNETX1.
- GWSSCPB, as the gateway VTAM controlling GWN2, resolves the COS name for NETB to a list of virtual routes in NETB. It uses the table COSNETB2.

VTAM uses its own default virtual route list when a cross-network SSCP-SSCP session activation is attempted with a COS name other than ISTVTCOS or the unnamed Class of Service. Session activation proceeds, but a message is sent to warn the operator that the COS table should be corrected. Whenever convenient, you should correct the problem by restarting VTAM.

Note: For user session requests, you can specify a substitute set of COS parameters that are used if the COS name is unknown to this VTAM. See "How session traffic is assigned to a specific route" on page 352.

Address translation

A gateway NCP uses one subarea for each network that is to be connected. Thus, a gateway NCP appears to reside in more than one subarea. Associated with that subarea is a pool of element addresses that can be used for cross-network sessions. When a cross-network session is requested, the gateway NCP assigns a pair of element addresses, one in each network on either side of the gateway NCP, to the session partners. This enables the gateway NCP to create within each network a representation of the session partner in the other network.

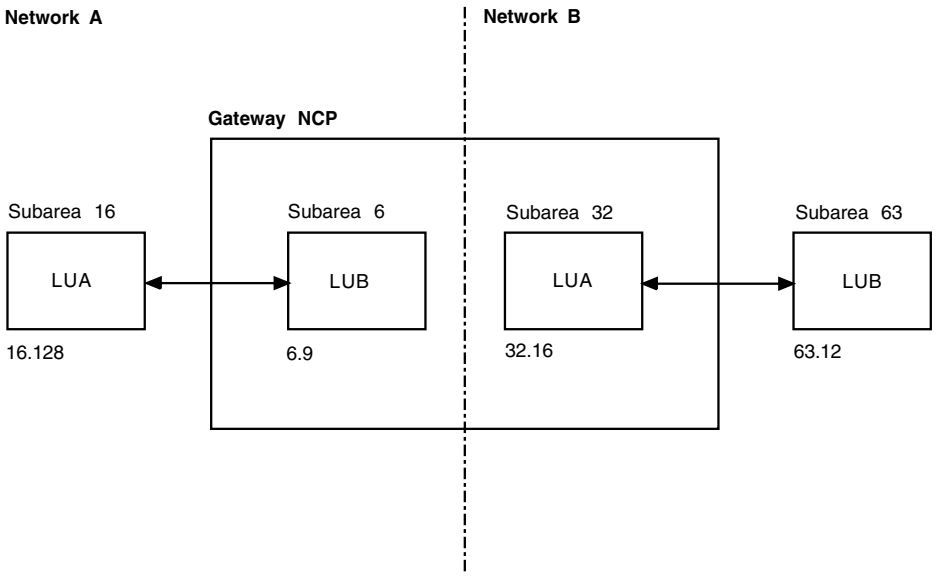


Figure 148. Address translation

For example, Figure 148 shows two networks (networks A and B) that are connected through one gateway NCP. An LU in Network A (LUA) is in session with another LU in Network B (LUB). LUB has a subarea address of 63 and an element address of 12 in its own network. Because these addresses might not be valid in Network A, the gateway NCP has given LUB a subarea number of 6 (the subarea of the gateway NCP in Network A) and an element address of 9. As a result, LUB now has a valid address in Network A that can be used for routing session data. When an LU in Network A sends data to LUB, it is routed to subarea 6, element 9, in the gateway NCP. The gateway NCP translates this address into the address to be used for LUB in Network B. It then passes the session data over to Network B. The session data is then routed through Network B to LUB in the usual manner.

Similarly, the gateway NCP has given LUA an address in Network B. Any session data destined for LUA from Network B is sent to this address.

Element addresses

Address translation is also done for cross-network SSCP-SSCP sessions. Each LU and SSCP with which you want to have a cross-network session requires at least one element address. The number of addresses that can be used to represent cross-network resources within a given network depends on the following:

- If extended network addressing is used, 32 768 element addresses can be used to assign alias addresses.
- If nonextended network addressing is used, the element pool is determined by the MAXSUBA specification. For example, if a network has MAXSUBA=63, each subarea can have at most 1024 element addresses.

Resources attached to the gateway NCP require element addresses in the native network. Therefore, while the number of possible cross-network sessions is not equal to the maximum number of elements in each subarea, the two must be considered together.

Parallel sessions and element addressing

Application programs that use nonparallel sessions require only one address to represent the primary LU to all of the secondary LUs. However, application

programs that use parallel sessions require a unique address to represent the primary LU to each one of the secondary LUs. If you plan to have cross-network parallel sessions, the network containing the secondary LU must have one element address for each parallel session with the primary LU. In the network of the primary LU, only one element address is required for all sessions with the secondary LU. Therefore, the number of addresses you should reserve in any particular network to represent application programs in other networks depends on whether the application program requires a unique primary LU address for each of its sessions.

For example, if terminals in your network are logged on to an application program such as CICS, you need only one element address in the gateway NCP to represent that CICS to your network. If parallel sessions are established between your CICS system and the CICS in the other network, the PLU must use an address for each session. If your CICS is acting as the SLU for these parallel sessions, an element address is required for each session in the gateway NCP to represent the PLU to your network. If the CICS in the other network is acting as the SLU for these parallel sessions, only one element address is required in the gateway NCP to represent that CICS as the SLU to your network.

The gateway NCP needs an additional address for the other network to represent each of your CICS parallel sessions to the other network. The gateway NCP also needs element addresses to represent your terminals to the network containing the other CICS.

Multiple ACBs and element addressing

If terminals in your network are logged on to an application program that opens a separate ACB for each user (such as the NetView or TSO programs), the number of element addresses that you need to represent the NetView program in your network is equal to the maximum number of terminals in your network that are concurrently logged on to that application program. In addition, the network containing the application program must have an element address for each of your terminals.

Resource name translation

SNA requires that each network resource be identified by a unique name. This requirement also applies to a multiple-network environment, unless you are using network-qualified names. Network-qualified names are implemented by specifying the NQNMODE=NQNAME start option or defining cross-network resources with the NQNMODE=NQNAME operand on the CDRSC definition statement. If you do not use network-qualified names, cross-network resources cannot be identified by the same name. For example, a problem occurs when a logical unit in one network requires access to a resource in another network that uses the same name to identify a different logical unit.

The class of service (COS) name and the logon mode entry name are also important when establishing sessions. The same COS and logon mode entry names must be used in each of the interconnected networks to establish cross-network sessions.

To meet these naming requirements when you are connecting networks, do one of the following:

- Redefine any duplicate resource names to preserve name uniqueness, and ensure that you use the same COS and logon mode entry names as the attached

networks. (If you are using network-qualified names, you do not need to redefine resources to resolve duplicate names.)

- Define any resources with duplicate names so that no more than one of them is specified (either by definition or by start option) with NQNMODE=NAME and the others are specified (either by definition or by start option) with NQNMODE=NQNAME.
- Use the VTAM alias selection function in the session management exit routine to translate any duplicate resource names to unique resource names, and to map unlike COS and logon mode entry names in one network to the proper equivalents in the other network. For details about the session management exit, refer to *z/OS Communications Server: SNA Customization*. (If you are using network-qualified names, you do not need to use the session management exit routine to resolve duplicate names.)
- Use the NetView alias name translation facility to translate any duplicate resource names to unique resource names and map unlike COS and logon mode entry names in one network to the proper equivalents in the other network. (If you are using network-qualified names, you do not need to use the NetView alias name translation facility to resolve duplicate names.)

The resource names that can be translated by the NetView alias name translation facility or in the session management exit routine are:

- Logical unit names
- Logon mode names
- Class of Service names

Alias selection function of the session management exit routine

You can use the alias function of the session management exit routine to perform name translation. By using this exit routine, you can provide your own alias and real names for logical units, logon mode names, Class of Service names, and SSCP names, instead of using the NetView alias name translation definitions. You can also indicate whether the NetView alias name translation facility is to be invoked for additional translation.

The session management exit routine can be invoked to translate a real or alias logical unit name associated with one network to a real or alias logical unit name for the destination network. You can map different LOGMODE and COS names in different networks to be functionally equivalent, and can also provide directory services by providing the owning SSCP names for the destination resource.

If you are using the NetView alias name translation facility, you can selectively invoke it to perform additional translations not performed in the exit routine. When the exit is invoked, there is an indicator that informs you if the NetView alias name translation facility is currently active. The exit routine can determine whether the session requests should be rejected, depending on the need for further translation by the NetView alias name translation facility and whether it is active. It can also accept or reject the use of a substitute COS.

For information on using the alias selection function of the session management exit routine, refer to *z/OS Communications Server: SNA Customization*.

NetView alias name translation facility

The NetView program provides the alias name translation facility that can do the following:

- Translates any LU name duplication conflicts
- Maps logon mode and COS names

Translating LU names

No two LUs in an SNA network can have the same name. When you connect independent SNA networks, however, there might be some duplicate names among these networks. To provide the unique LU names required for SNA sessions but still allow each network to have independent resource names, interconnection provides for name translation between networks. Logical unit names are translated from names known in one network to names known in the network of the session partner to avoid duplicate names.

Network-qualified names provide an option to name translation and enable you to have resources by the same name in different networks without requiring you to use name translation.

If you are not using network-qualified names, you can use name translation, which allows each LU to have two names: a real name by which it is known in its own network and an alias name by which it is known in another network. For example, an LU in NETA is known by its real name LUA1A3 in NETA, and an LU in NETB is also known by its real name LUA1A3 in NETB. If NETA is connected to NETB, a name duplication conflict develops. The alias name translation facility enables LUA1A3 in NETA to be identified by an alias name in NETB (for example, BBA1A3). By using this procedure, you can maintain name uniqueness within a network and adhere to SNA naming requirements.

The NetView alias name translation facility uses installation-defined translation tables. The tables are a one-to-one mapping of the alias LU name to the real LU name and their corresponding networks. This technique differs somewhat from that used for maintaining alias addresses across a network boundary. That is, an exact one-to-one mapping is defined by an installation using the translation tables. However, terminal users and network operators do not normally know an LU by its network subarea and element address. Instead, it is known by its unique name in the network. If the alias name (by which an LU is known) is able to change dynamically with each session, network operations might be very difficult, and network security might be affected. The NetView translation tables therefore predefine the relationship between an alias name and a real name.

Note: The alias name translation facility cannot translate a name as known in one network to a different name in the same network. Also, because NetView domain identifiers cannot be translated, they must be unique for all NetView programs that have cross-network sessions. Resource names for LU 6.2 sessions also cannot be translated.

Mapping logon mode names

Before a session is established, both session partners must agree on a common set of session parameters. They do this by specifying a logon mode name that identifies a particular set of session parameters. Within an SNA network, both session partners use the same logon mode name, thus ensuring that the correct session parameters are used. When you connect independent SNA networks, however, a given set of session parameters might be known by different names in different networks. Also, a given name might refer to different session parameters in each network. The alias name translation facility can be used to map a logon mode name in one network to another name in the destination network that

provides equivalent session parameters. This translation can occur in any SSCP on the setup path, after the network identifier of the destination network is determined.

As usual, the logon mode name that is used to establish a session must exist in the logon mode table of the secondary logical unit SSCP.

Mapping Class of Service names

Before a session is established, both session partners must agree on a common Class of Service that defines a particular ordered list of virtual routes. They do this by specifying a Class of Service name. Within an SNA network, both session partners can use the same Class of Service name, thus ensuring that the correct Class of Service is used. When you connect independent SNA networks, however, a given Class of Service might be known by different names in different networks, or the same name might be used for different Classes of Service. When the alias name translation facility is used, a Class of Service name in one network can be mapped to a different name in each adjacent network along a cross-network session path to provide equivalent service. This translation can occur in any SSCP on the setup path. For SLU initiated sessions, you must also map the COS name for the origin logical unit (OLU) to the destination logical unit (DLU).

The COS name entry used for the session must exist in the COS table of the primary logical unit SSCP and in the gateway VTAM that performs the SETCV. For information on determining which gateway VTAM performs the SETCV, see "Cross-network routing" on page 584.

The NetView alias name translation facility can be used to establish equivalence for the required session parameters. The NetView alias name translation facility maps the logon mode name in one network to the logon mode name used in the other network. The same process is used for the COS name. If the COS name needed to establish a virtual route and transmission priority in one network does not exist in the COS table of the PLU in the other network, the NetView alias name translation facility can be used.

This alias name translation facility maps the COS name in one network to the COS name in the other network that provides a network route with equivalent characteristics (for example, high speed, high availability, and security).

A substitute COS entry can also be used. The substitute COS entry, chosen by specifying the SUBSTUT operand on the COS macroinstruction for one COS entry, indicates the COS entry to be used when the COS specified in a session request is unknown. For more information on how the substitute COS works, see "Substituting Class of Service Parameters" in "How session traffic is assigned to a specific route" on page 352.

Defining alias names

When defining resources in interconnected networks, you must reconcile any name conflicts among the connected networks. Following are the ways you can do this:

- Adopt a naming convention in all networks that ensures that all names are unique.
- Define alias name translation tables.
- Use the alias function in the session management exit routine.
- Implement network-qualified names.

The method that is best for you depends on how many names must be changed and how much control you have over the various networks.

To use an alias name translation facility, define all the logical units in the interconnected networks that are not already defined and their owning SSCPs to the NetView alias name translation facility and assign alias names. Then, if VTAM does not have a complete definition of a destination logical unit for a session setup request, it invokes the NetView alias name translation facility with the name of the logical unit. The NetView alias name translation facility returns the name of the SSCP defined as the owner of the logical unit, as well as the destination network ID and the logical unit real name in that destination network.

If the following conditions are true, the request for the session is sent to the owning SSCP:
- No predefined CDRSCs are used.
- VTAM has a session with the owning SSCP.
- Dynamic definition of cross-domain resources is allowed.

VTAM automatically creates a CDRSC definition for the logical unit.

If you have not installed a name translation facility, or if the name translation facility does not return the name of the owning SSCP, SSCP selection using the adjacent SSCP table is tried next.

Note: VTAM uses real names and alias names when referencing an LU. An alias name is the name of the LU as known in the network of its session partner. Because one LU can be in session with partners in more than one network, an LU can have multiple alias names. VTAM can obtain real and alias names through an alias translation application program (for example, NetView). If such an application program or the alias function of the session management exit routine is not in use, VTAM uses network qualifiers to define the real and alias names for logical units, in which case, all resource names must be unique. Therefore, no two LUs can have the same name, whether or not they are in the same network. However, if you are using network-qualified names, you can have duplicate resource names in different networks.

In general, when VTAM receives a session setup request and the alias name translation facility is active, VTAM invokes the alias name translation facility to translate LU names, Class of Service names, and the logon mode name. You can use the TRANSLAT start option to limit the scope of the translation functions for which VTAM invokes the NetView program.

The default for the CDRDYN start option is YES, enabling dynamic CDRSC definition. The CDRDYN start option overrides the CDRDYN operand on the host CDRM definition statement. CDRDYN must be YES for the VTAM in which the NetView alias name translation facility or the alias function of the session management exit routine is used.

If an alias name is defined to VTAM with a CDRSC definition (either predefined or dynamically defined), the DISPLAY operator command can be used to display the real name associated with that alias name (if a session exists with the resource and if VTAM has determined the real name).

You should have a convention for creating alias names. This alias naming convention should identify a name as an alias name. An operator often needs to

know whether a resource name is an alias name or a real name. For example, the NetView program requires that the operator enter the real name of a resource when a display is requested.

Note: The generic name used by the alias function should not also be used by USERVAR.

As an example of name translation, Figure 149 shows a gateway NCP connecting two networks, NETA and NETB. Both networks contain an application program named IMS. A terminal user, TERMU, in NETA wants to log on to the IMS that is in NETB.

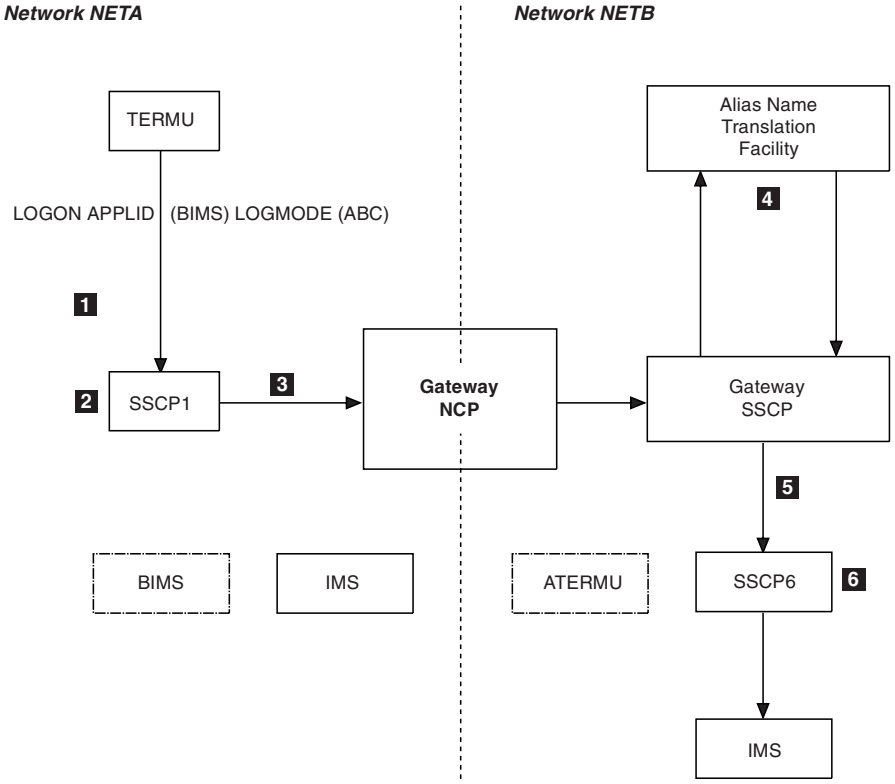


Figure 149. Example of name translation

The following steps show how name translation is used in setting up a session between TERMU in NETA and IMS in NETB. The numbers shown in Figure 149 correspond to the following steps.

1. TERMU enters a LOGON command specifying BIMS, which is an alias name for IMS in NETB.
2. The LOGON command is processed by SSCP1, the SSCP that owns TERMU. The COS name FAST is obtained from the logon mode entry named ABC in the logon mode table associated with TERMU.
3. SSCP1 has a dynamic or predefined CDRSC definition without NETID for the resource named BIMS, which indicates that BIMS is owned by the gateway SSCP. SSCP1 therefore passes the request to the gateway VTAM.
4. Because the NetView alias name translation facility is active or because an alias function in the session management exit routine is used, the gateway VTAM supplies the following names for translation:
   - Origin network ID (NETA)

- Origin LU (TERMU)
- Alias name of destination LU (BIMS)
- COS name (FAST)
- Logon mode name (ABC)

What NetView or the session management alias function returns to the gateway VTAM depends on what user defined name translations are coded in the NetView program or are available from the session management alias function. In this example, the following names are returned:

- Destination network ID (NETB)
- Real name of destination LU (IMS)
- Name of owning SSCP for IMS (SSCP6)
- Alias name for TERMU (ATERMU)
- Equivalent COS name (QUICK)
- Equivalent logon mode name (XYZ)

5. A dynamic CDRSC definition is created in SSCP6 for TERMU using the alias name and is associated with the real name.

After the session is set up, it appears to SSCP1 that TERMU is in session with a CDRSC in NETA named BIMS. To SSCP6, it appears that IMS is in session with a CDRSC in NETB called ATERMU. However, if SSCP1 or SSCP6 were a VTAM with the NETID start option specified, SSCP1 or SSCP6 would also be aware that this is a cross-network session.

The rules for name translation are as follows:

- Logical unit names and logon mode names are translated only once. If class of service is provided on the session-setup request, it is translated once from the name known in the origin network to the name known in the destination network. As the response to the session setup request flows back to the origin host, it is translated in each network to determine the COS name as it is known in the adjacent network.
- If the destination LU name must be translated, at least one of the gateway VTAMs that controls the first gateway NCP in the session setup path must have an alias name translation facility.

  Translation of the destination LU name is attempted in every gateway or gateway-capable SSCP that has an active name translation facility until the names are translated or until a gateway VTAM requests alias addresses from a gateway NCP. A gateway VTAM requests alias addresses if it has designated control of the gateway NCP or if it is the last gateway VTAM on the first gateway NCP. At that point the destination LU name is assumed to be real if it has not been translated.
- After the destination logical unit real name and network identifier are determined, VTAM still tries to translate the origin LU name at every gateway or gateway-capable SSCP along the session setup path that has an active name translation facility until the origin LU alias name is determined. Also, if VTAM knows the destination LU name is real, it still invokes the alias name translation facility to try to translate the COS and logon mode names and to determine the owning SSCP.
- If you specify the owning SSCP in the alias tables, use the name specified on the SSCPNAME start option used by the VTAM that owns the resource.
- The logon mode name, Class of Service name, and the origin LU name can be translated at the gateway VTAM that translates the destination LU name or any subsequent gateway or gateway-capable SSCP along the session setup path.

  Prior to VTAM Version 3 Release 2, the Class of Servicename for the destination network had to be translated at the same gateway VTAM that translates the

destination LU name. If the destination LU were a real name, determined by CDRSC definition, the COS name had to be translated at the same host.

- NetView domain IDs (NCCFID) must be unique and cannot be assigned alias names.
- Alias name translation facility support for LU 6.2 is restricted. This facility can be used to determine the correct network identifier of the destination logical unit and the name of the owning SSCP. However, it cannot be used to translate logical unit names. Most peripheral nodes that use LU 6.2 communication protocol do not support alias name translation.
- The VTAM constants module limits the information to be translated by the alias name translation facility.

Resolving alias name requirements

An alias name is defined in a host to represent a logical unit, logon mode table, or Class of Service (COS) name in another network.

The following method can be used to determine which alias names are required and to distribute the required alias name definitions to the correct gateway SSCPs. The alias names you choose cannot match the names of any resources in your network.

1. Create a table listing all of your LUs, even if they do not have cross-network sessions. You need not list every resource in your network if it is obvious from your naming conventions that certain names cannot be duplicated in any network to which you are connecting.
2. List all of the resources in other networks with which you communicate. If you use an adjacent SSCP list and do not specify the destination network, include all resources in all networks in the second column, even those with which you have no sessions. (This includes resources such as lines and physical units.) Otherwise, a resource might receive a session request that is intended for an LU that has the same name but is in a different network.

   Table 48 illustrates this process:

Table 48. Network resource list example

All resources in your network (NETA)	All other network resources that communicate with owning network	Owning NETID for resources in other networks
ABLE	BAKER	NETY
CHARLIE	BAKER	NETZ
DOG	CHARLIE	NETX
FOX	EASY	NETX
EASY	JOAN	NETY
MARY	PETER	NETZ
ROGER	.	.

3. Find any duplicates in the table you have created.

   In the example shown in Table 48, the duplicate names are CHARLIE and EASY. NETX.CHARLIE and NETX.EASY need alias names so you can distinguish them from resources in your network. Either NETY.BAKER or NETZ.BAKER needs an alias name so that you can distinguish them from each other.

   For this example, the alias name definitions that you need to code for the NetView alias name translation facility are the following:

```
ORIGNET     NETA
     LU     CHARLIE,NETX,ALCHARLI
     LU     EASY,NETX,ALEASY
     LU     BAKER,NETY,ALBAKER
```

ORIGNET always represents the network that knows the resource by its alias name. The network specified on the LU definition statement is always the network that knows the resource by its real name.

With these alias names replacing the real names, you now know every resource you communicate with by a unique name.

After every network goes through the preceding process, consolidate the results so that every network that is to request name translations has a complete set of alias name definitions for LUs. For example, if NETX has a NetView alias name translation facility, you should supply NETX with the following:

```
ORIGNET   NETA
     LU   CHARLIE,NETX,ALCHARLI
     LU   EASY,NETX,ALEASY
```

Similarly, if NETY has a NetView alias name translation facility, you should supply NETY with the following:

```
ORIGNET  NETA
     LU  BAKER,NETY,ALBAKER
```

If your network has a NetView alias name translation facility, you should keep a copy of all the definitions you created. In addition, other networks might give you alias name definitions that they created to refer to your resources. For example, NETX might give you the following:

```
ORIGNET NETX
     LU MARY,NETA,MARYX
     LU EASY,NETA,EASYX
     LU CHARLIE,NETA,CHARLIEX
```

By distributing the alias name definitions as indicated, you might find that several networks have the same definition. However, you are assured that LU names get translated at the first NetView alias name translation facility called, regardless of the direction of session setup.

Establishing and controlling SNA sessions

This section contains information to be considered for nonnative network type 2.1 connections. Also, to control cross-network sessions you can use the automatic logon function of VTAM and take advantage of certain NetView Performance Monitor functions.

Nonnative network type 2.1 connections

For nonnative network type 2.1 connections, consider the following when defining and specifying destination logical units.

Considerations for network nodes

In Figure 150 on page 601, two AS/400 network nodes are attached to a VTAM network node by nonnative network type 2.1 connections. If two resources (one at each of the AS/400 network nodes) are to communicate with each other through the intermediate VTAM network node, at least one of the connections must be a LEN connection or APPN multiple network connectivity support must be implemented (see “APPN multiple network connectivity” on page 89).

Considerations for end nodes

In Figure 150, some resources are attached to the end node by nonnative network type 2.1 connections. If one of the resources initiates a session, the network node server caches the network-qualified name of the originating LU, and the resource can be found subsequently by the APPN network. However, APPN might not be able to locate these resources when attempting to locate them as destination LUs, unless the resources are predefined at the end node (with CDRSC definitions for all possible network IDs) and registered with the network node server. If all possible network-qualified names of the resource are known at the network node server, APPN searches can locate the destination LU. The use of a predefined alias name for these resources at the end node is not recommended.

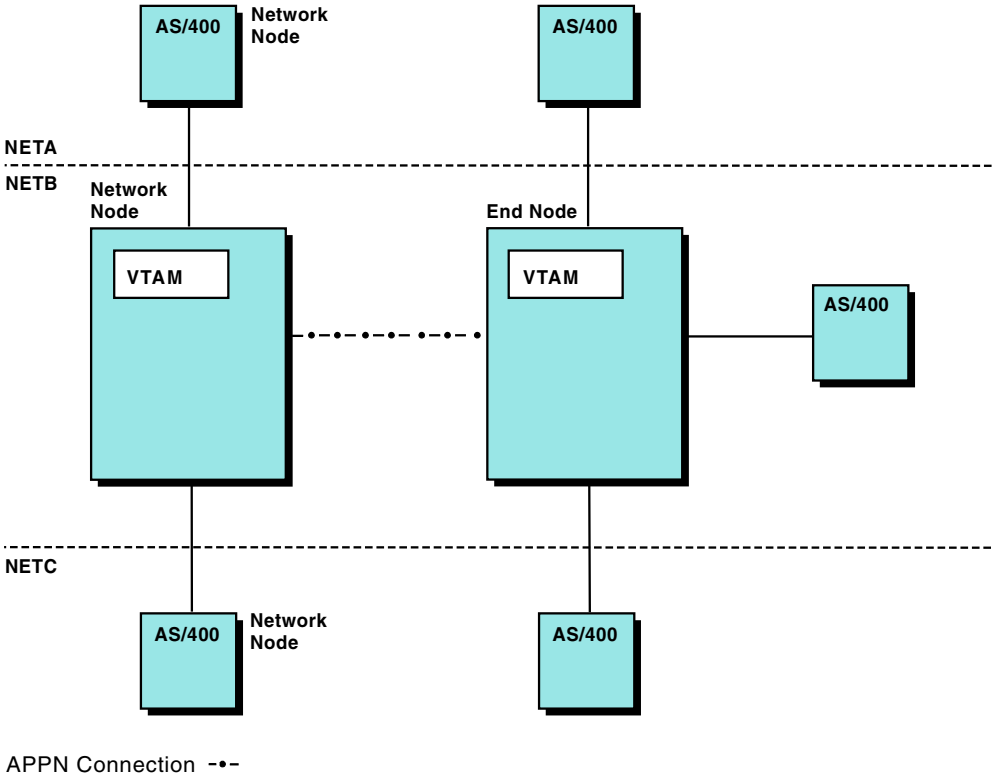


Figure 150. Nonnative network type 2.1 connections

Specifying and defining the destination LU

When requesting a session with resources that are attached by nonnative network type 2.1 connections, the originating LU should specify the NETID of the adjacent node or no NETID. If the originating LU specifies the NETID of the adjacent node, the destination LU must be located in the same network as the adjacent node. If a session request does not include a NETID for the destination LU and a predefined alias CDRSC is defined for the destination LU, the resource can be located in any network.

Automatic logon

Using the LOGAPPL operand, you can define a cross-network controlling primary logical unit on any peripheral node (local or remote, switched or nonswitched). The effect is essentially the same as if a CDRSC within the same network were the controlling primary logical unit, except that the resulting session is cross-network

rather than within the same network. The network-qualified name of the controlling primary logical unit can be specified with the LOGAPPL operand.

In a single-network environment, when the SSCP-SSCP session necessary for the establishment of a controlling primary logical unit session is not immediately available, VTAM stores information necessary for session setup. When the necessary SSCP-SSCP session becomes active, VTAM establishes the LU-LU session.

However, this process does not occur for cross-network sessions. You can establish cross-network controlling primary logical unit sessions through any number of intermediate SSCPs only if all of the required SSCP-SSCP sessions are active when the LU-LU session setup is attempted. In the case where the session setup path consists of more than one SSCP-SSCP session, the session setup proceeds as far as establishing the session control blocks at each active SSCP along the setup path. At some point along the path, an SSCP can determine that the session request cannot be routed any further because it has no active sessions with any SSCPs in its adjacent SSCP list for the specific destination logical unit. If this occurs, the request is rejected, the session control blocks are freed, and the operator is notified of the failure. After the necessary SSCP-SSCP sessions are active, the operator must try to reestablish the session. If all required SSCP-SSCP sessions exist, but the controlling application program is unavailable, VTAM fails the initiation request until the application program is available. When the application program becomes available, VTAM automatically attempts to reestablish the session.

Operating VTAM

This section contains information about operation considerations in a multiple-network environment. If you are migrating to a multiple-network, take this information into consideration when making decisions about your operating procedures.

Gateway NCP and VTAM interaction

A gateway VTAM does not have to reside in the gateway NCP native network, but having the gateway VTAM in the NCP nonnative network puts some restrictions on VTAM. Only a small number of commands from a VTAM in a nonnative network are accepted by the gateway NCP: activating, deactivating the gateway NCP, route testing, and requests for address translations. Only VTAMs in the native network can request dynamic dumps or traces of resources.

You can prevent link-attached hosts in the nonnative network from loading or dumping the gateway NCP by specifying IPL=NO on the LINE definition statement for that link. You can prevent VTAMs in the nonnative network from activating the gateway NCP by specifying ACTPU=NO on the NETWORK definition statement for that network.

Using the NetView program for network management

The NetView program provides facilities for problem determination and network management in a gateway configuration. These facilities include the following:

- Information about the alias addresses, alias and real LU names, and owning SSCPs for LUs involved in cross-network sessions
- Session awareness data, such as session protocols
- Inquiry, on a session basis, as to whether a complete physical path exists between the controlling subarea nodes of the two session partners

- Display of the subarea nodes and transmission groups

The availability of these functions for a particular cross-network session depends on the NetView program being installed in the appropriate hosts. It also depends on the appropriate level of VTAM or NCP being installed in each subarea.

It is recommended that the NetView program be installed at every gateway VTAM host and at every host that controls resources that might have cross-network sessions. This ensures that the NetView program is able to provide the maximum amount of problem determination data for cross-network sessions.

Note: When Netview is active, some messages will be sent to the Netview instead of the MVS system console. Refer to *z/OS Communications Server: SNA Messages* for further information about solicited and unsolicited messages and their destinations.

Application programs

For a multiple-network environment, you need to define the NetView program or another name translation facility and control application program security across networks.

Note: An application program can become a shadow resource if a CDRSC with the same name already exists when the major node containing the application program definition is activated. For more information on shadow resources, see "Shadow resources" on page 549.

Defining the NetView program

If you are running the NetView program, ensure that at least one application program major node contains definitions for the NetView program and the NetView alias name translation facility if you are using it. If you are using network-qualified names, it is not necessary to use the NetView alias name translation facility. Network-qualified names provide an alternative to using the facility for resolving duplicate names. The NetView alias name translation facility can be used in conjunction with network-qualified names.

If you want to use the alias name translation facility provided by the NetView program, by the NCCF licensed program, or by a similar user-written application program, include an APPL definition statement for that application program in the application program major node. The name on the APPL definition statement must be ALIASAPL, as shown in the following example:

```
ALIASAPL APPL  AUTH=(CNM),PRTCT=password
```

Note: You can use a different name than ALIASAPL, but if you do, change and reassemble the CNM routing table. If you want to do this, refer to *z/OS Communications Server: SNA Customization* for more information.

The AUTH=CNM operand on the APPL definition statement means that this application program can use the communication network management (CNM) interface. This operand must be coded for all CNM application programs, including the following:

- The session monitor component of the NetView program
- The NetView program alias name translation facility
- The NLDM licensed program
- A user-written application program used for name translation

The following NetView program definition statements must also be coded:

- One ORIGNET definition statement for each network that requires alias name translation. The definition statements that follow this ORIGNET definition statement apply to the named network.
- One LU definition statement for each LU that is given an alias name.
- One COS definition statement for each Class of Service name that must be mapped to the name of an equivalent Class of Service used in another network.
- One MODE definition statement for each logon mode name that must be mapped to the name of an equivalent logon mode used in another network.

Chapter 21. Operating VTAM

Operating VTAM involves starting and stopping VTAM, activating and deactivating resources, and monitoring the domain. Although the operator usually performs these tasks, the system programmer must define the procedures that the operator uses when doing these tasks. These procedures are usually written by the system programmer in a locally-produced document called a *run book*.

A run book is a list of procedures for operators to follow. For example, you might want to list the start options that operators should use when starting different hosts. Following is an example of what might be in a run book:

```
After you activate S013H:

VARY NET,ACT,ID=S013H1
VARY NET,ACT,ID=S013H2
VARY NET,ACT,ID=S013H3
VARY NET,ACT,ID=S013H4
VARY NET,ACT,ID=S013H5
VARY NET,ACT,ID=S013H6
```

Notice that this gives very specific information about your own network, such as the names of the nodes.

For operators who are learning about operating VTAM, refer to *z/OS Communications Server: SNA Operation*.

The following list shows where to find information about operating VTAM.

- "Starting the domain"
- "Activating resources" on page 610
- "Monitoring the domain" on page 615
- "Controlling the domain" on page 618
- "Deactivating resources" on page 619
- "Halting VTAM" on page 621
- "Canceling VTAM" on page 623
- "Automatic operations" on page 623
- "Operating VTAM in a multiple-domain subarea network" on page 626

Starting the domain

To start VTAM, use the START command. You can enter start options when you start VTAM, or you can specify a start list. See "Start options" on page 28 for more information on start lists and start options.

Following is the general start procedure for VTAM:

1. Respond to message IEA101A during operating system IPL by entering RERP=*xx*, where *xx* represents the last two characters of the alternate ERP list name, if you want to use the resident error recovery procedures (ERP) option or the ERP option with an alternate ERP list.
2. Make available the necessary channel-attached devices using the MVS command VARY.
3. Use the VTAM start procedure.

© Copyright IBM Corp. 1984, 2011

4. Enter VTAM start options, if necessary.
5. Enter the logon manager START command, if necessary.

Configuration restart

You can use configuration restart to restart your network with the same status information prior to VTAM deactivation or failure. VTAM can maintain certain information about VTAM resources, such as status (active or inactive) and operator-specified values of activation and operating parameters. When restarting VTAM after a halt or failure or when reactivating an individual major node after a deactivation or failure, VTAM can use this information to restore the resources to their status prior to the deactivation or failure.

Recording changes to the network configuration

When the host operating system is generated, VTAM definition statements are filed in the VTAM definition library. The first time a major node is activated, activation information is recorded in the resource definition table and in the NODELST file, if one exists. Any further activations or deactivations of major nodes are also recorded in the resource definition table and in the NODELST file (if any). Every time the VTAM operator changes the status of a minor node, the change is recorded in the resource definition table and in a configuration restart file, if one exists.

The network configuration can be restored to either its initial status or its status before failure or deactivation. The status being restored depends on whether you have defined configuration restart files and on how you use the CONFIG operand and the optional COLD or WARM operand when restarting the network. Figure 151 on page 607 illustrates this process.

Using configuration restart warm start

As shown in Figure 151 on page 607, if you specify the WARM operand when activating a major node, VTAM updates its resource definition table from information in the VSAM configuration restart file for the major node. VTAM then restores the minor nodes within the major node to their status prior to the failure or deactivation.

Also shown in Figure 151 on page 607, if you specify the WARM start option when starting VTAM, VTAM can update its resource definition table from information in the NODELST file.

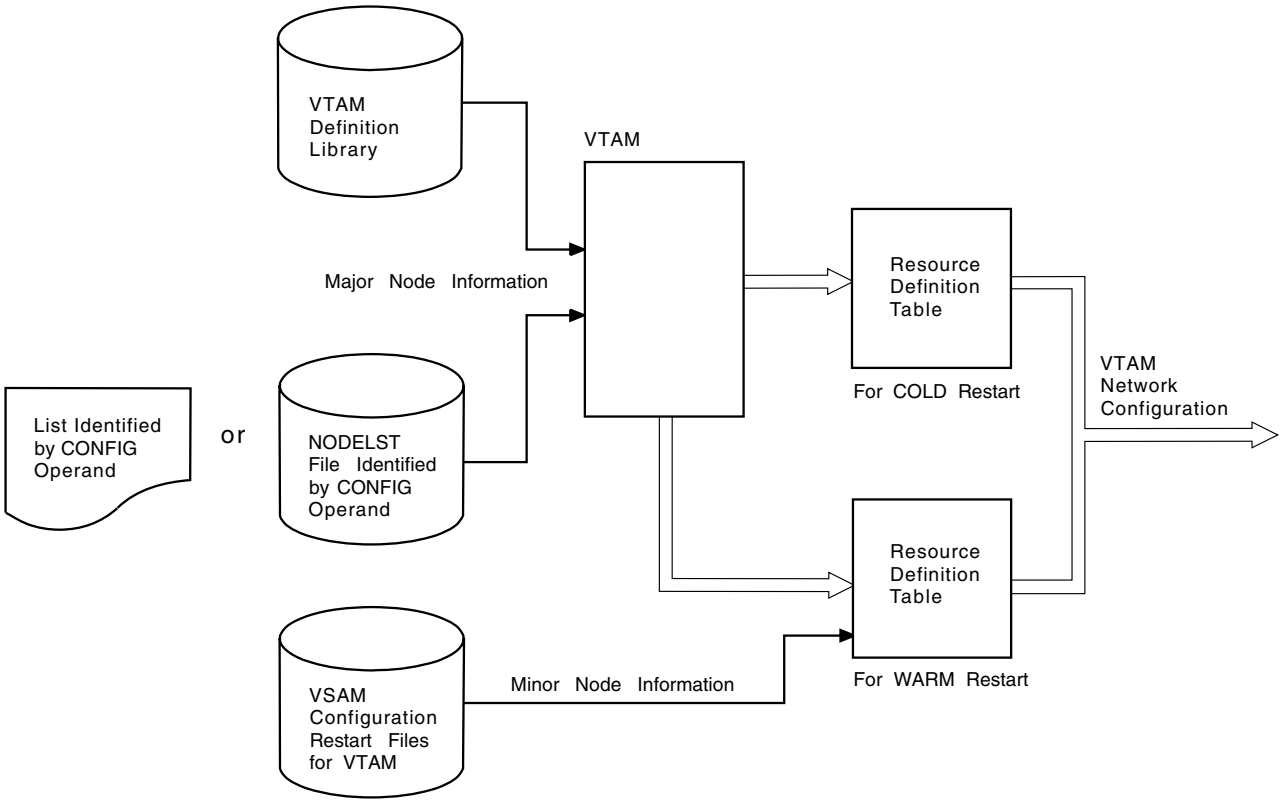


Figure 151. Restoring resource definitions with configuration restart and NODELST files

If you specified a NODELST file when the network was last started, the major nodes that were active and the DR files that were applied when the failure occurred can be reactivated by specifying the NODELST file name on the CONFIG operand during a warm start. If you used the VARY ACT,UPDATE technique of dynamic reconfiguration, major node definitions reflect dynamic reconfiguration changes, and the changes are reestablished when the major node is activated during NODELST processing. See "Dynamic reconfiguration and change of operands" on page 235 for more information on the techniques that are available for dynamic reconfiguration.

Note: Dynamically defined independent logical units cannot be warm started.

For an example of how NODELST and configuration restart files are used, see "Configuration restart example" on page 608.

VTAM uses the VSAM configuration restart files to record changes to the named minor nodes within the major nodes (unnamed minor nodes are not checkpointed). If the VSAM files are defined, the VTAM operator can specify how VTAM is to use the files for configuration restart.

To associate VSAM configuration restart files with individual major nodes, code the CONFGDS and CONFGPW operands on the definition statements for the major nodes.

A NODELST file can be defined to record which major nodes and DR files are active in the network. A NODELST file is specified with the NODELST start option.

You can use either or both of these file types when using configuration restart.

If the user defines configuration restart files, VTAM provides two additional forms of recovery: manual switching to a backup host processor and manual switching to a backup communication controller.

If you do not associate a configuration restart file with a major node, or if the configuration restart file is empty, then the major node is not activated to its prior status.

If NODELST is in effect and you are doing a warm start of a configuration that has been dynamically reconfigured using the VARY DRDS technique, the DR statements that are executed again as part of the start process might fail because the nodes they specify became active earlier in the process.

Configuration restart example: Configuration restart files have been specified for the major nodes A, B, and C (CONFGA for major node A, CONFGB for major node B, CONFGC for major node C). Each major node is comprised of four minor nodes (for example, major node A is comprised of minor nodes A1, A2, A3, and A4). The following shows the minor nodes associated with each major node.

Major node A	Major node B	Major node C
A1	B1	C1
A2	B2	C2
A3	B3	C3
A4	B4	C4

Minor nodes B3 and B4 are to be initially active and the other minor nodes inactive.

Note: The initial status (active or inactive) can be specified for each minor node by coding the ISTATUS operand on the minor node definition statement.

The following chart shows the current contents of the configuration restart files for each major node.

CONFGA	CONFGB	CONFGC
empty	empty	empty

The VTAM operator enters the following start options on the START command:

```
CONFIG=XX,NODELST=ABC
```

The CONFIG start option indicates which major nodes are to be activated when VTAM is started. In this example, configuration list ATCCONXX contains major nodes A and C. The NODELST start option specifies that the VSAM file ABC is to be used to record which major nodes are active in the network.

Because of the above start options:

- VTAM activates major nodes A and C if it is able to open the NODELST file ABC.
- VTAM records the activation of major nodes A and C in file ABC.

Then to activate major node B, the operator enters:

```
VARY NET,ACT,ID=B
```

VTAM activates major node B and records its activation in ABC. The VSAM NODELST file ABC now contains the following active nodes:

- Major node A
- Major node B
- Major node C

VTAM also activates minor nodes B3 and B4 because ISTATUS=ACTIVE has been specified for them. Because B3 and B4 are defined as initially active, VTAM does not record the activation of B3 and B4 in CONFGB.

After the major nodes are active, the operator activates, deactivates, or modifies the status of minor nodes with VARY commands. Every time VTAM activates, deactivates, or modifies the status of a minor node, it records that action in the configuration restart file (if one exists) for the major node of which the minor node is a part.

```
VARY NET,ACT,ID=A2
VARY NET,ACT,ID=B3,LOGON=logapp1
VARY NET,INACT,ID=B3
VARY NET,ACT,ID=C4
```

As a result of the preceding commands, the status information stored in the configuration restart files can be represented as:

CONFGA	CONFGB	CONFGC
A2 active	B3 Status: LOGON B4 inactive	C4 active

If the network is now halted or if a failure occurs in the network, VTAM deactivates all the active major and minor nodes and closes the configuration restart files. The information needed to restore the network is in the VTAM definition statements and these files. When the VTAM operator restarts the network, start options can again be entered.

Start options specified	Major nodes restarted	Minor nodes restarted
CONFIG=XX,COLD	A and C	None
CONFIG=XX,WARM	A and C	A2 and C4
CONFIG=ABC,COLD	A,B, and C	B3 and B4
CONFIG=ABC,WARM	A,B, and C	A2, B3, and C4

Specifying CONFIG=XX,COLD or CONFIG=XX restores the network to its initial status while CONFIG=ABC,WARM restores the network to the status it had when VTAM stopped.

Note: COLD is assumed if neither COLD nor WARM is specified.

The operator can also specify another NODELST file:

```
CONFIG=ABC,WARM,NODELST=QRS
```

Or, the operator can continue using the one used before:

```
CONFIG=ABC,WARM,NODELST=ABC
```

Information recorded by configuration restart

The following major nodes have the indicated information recorded:

NCP major node

- The channel device address (if channel-attached) and the RNAME values (if link-attached)
- For each line, its status (active or inactive), non-SDLC line scheduling parameters, and for dial SDLC lines, the answer mode (whether the line is dial-in)
- For each link station, its status (active or inactive)
- For each logical unit, its status (active or inactive) and the names specified in the LOGAPPL and LOGMODE operands
- For each physical unit, its status (active or inactive)
- For each logical unit, the cryptographic status (REQD, SEL, OPT, or NONE)

Channel-attachment major node

- For each link, its status (active or inactive) and the channel unit address
- For each link station, its status (active or inactive)

Local SNA major node

- For each physical unit, its status (active or inactive) and its channel device address
- For each logical unit, its status (active or inactive) and the names specified in the LOGAPPL and LOGMODE operands and whether it is active or inactive
- For each logical unit, the cryptographic status (REQD, SEL, OPT, or NONE)

Switched SNA major node

- For each logical unit, its status (active or inactive) and the names specified in the LOGAPPL and LOGMODE operands
- For each physical unit, its status (active or inactive)
- For a physical unit with dial-out capability, the PATH=USE|NOUSE setting
- For each logical unit, the cryptographic status (REQD, SEL, OPT, or NONE)

Local non-SNA major node

Each channel-attached terminal, its status (active or inactive), its channel device address, and the names specified in the LOGAPPL and LOGMODE operands

CDRSC or CDRM major node

The status of each minor node (active or inactive)

Activating resources

You can initially activate resources by doing one of the following:

Specifying the name of the resource in a configuration file
Do this to activate the resource when VTAM is initialized. For information on coding a configuration file, see "Configuration lists" on page 37.

Using the VARY ACT command after VTAM is initialized
Enter a VARY ACT command with the ID operand specifying the name of the node to be activated.

To control activation of resources that are lower in the hierarchy, you can do one or both of the following:

Use the SCOPE operand on the VARY ACT command
If you code or use the default SCOPE=COMP on the VARY ACT command, all resources subordinate to this resource are activated if the

ISTATUS operands for each are defaulted or coded as active. SCOPE=ALL causes all resources to be activated regardless of the coding of the ISTATUS operand.

Code the ISTATUS operand on the resource definition statement
The ISTATUS operand controls the initial status of the resource. If you code or default to ISTATUS=ACTIVE, this resource is activated when its major node is activated.

If you make changes to your definition statements, deactivate and then reactivate the major nodes that are affected. Otherwise, the changes are not recorded.

Note: If you specify a warm start of VTAM, the status recorded in a configuration restart file overrides initial status. VTAM either activates a node or leaves a node inactive according to the status of the node when VTAM was last running. See "Configuration restart example" on page 608 for additional information.

Order of activation

A resource cannot be activated unless all other resources above it in the hierarchy have been activated. For example, a logical unit cannot be activated unless the physical unit to which it is subordinate is already active. When a resource is deactivated, all of the resources subordinate to it in the hierarchy are also deactivated.

Although there is no logical hierarchy within the set of major nodes themselves or between resources in different major nodes, there is an obvious topological relationship among the major nodes representing subarea nodes in the network and the lines and link stations connecting them. Therefore, the activation of an NCP major node depends on the lines, link stations, and other subarea nodes that constitute the routes between VTAM and the NCP being active. Likewise, deactivation of an NCP major node or a line or link station within it can affect those other subarea nodes for which the resource being deactivated constitutes a part of the path from VTAM.

All of the elements of a communication path between two session endpoints must be active before a session can be established. In domains with more than one subarea, a path definition set must be activated before communication between subareas can be attempted. Major nodes are activated, and then their minor nodes are activated to enable sessions to be established for carrying user message traffic.

Although an activation command for a major node can be entered at any time after VTAM is initialized and before it is halted, the activation of NCP major nodes and external CDRM minor nodes requires fully operative explicit routes. If these routes are not available, the activations remain pending until the routes become available or until you deactivate the resources.

If automatic logons to a controlling application program have been defined for any logical units, all of the necessary application program major nodes must be activated before any of these logical units are activated (directly or indirectly). The logical units must be active before a session can begin.

The specific order in which you activate major nodes depends on how you have coded your network. For example, if you defined a terminal to automatically log

on to an application program, activate the application program first. Following is a general guideline for activating your resources (the activation order may vary due to coding differences in your network):

1. Local resources, which includes the host CDRM, application programs, and other local resources. However, avoid activating any resources that will automatically log on to resources in other domains.
2. Routes, which include explicit and virtual route path definitions. Activate these paths before any connection to another subarea node (NCP or VTAM).
3. Peripheral resources, which include channels, NCPs, switched devices, cross-domain resource managers and cross-domain resources.

In multiple-domain networks, ADJSSCP tables should be activated before cross-domain resources, unless the dynamic adjacent SSCP function is specified.

When activating cross-domain resource managers, first activate the host cross-domain resource manager in each domain. This enables each host cross-domain resource manager to request sessions with and accept session requests from other known cross-domain resource managers.

After you have activated the host cross-domain resource manager, you can then activate any CDRM major nodes that represent other domains. VTAM then requests a session with the other SSCP. The request is rejected unless the other domain has activated both its host cross-domain resource manager and the external cross-domain resource manager representing the SSCP making the request.

Requests for sessions with cross-domain resources that are not defined are defined dynamically by the cross-domain resource manager, if you have authorized dynamic definition of cross-domain resources.

For autologon sessions, if a session request cannot be routed to its destination because an SSCP-SSCP session has not been established, the session fails and the operator receives a notice of the failure. After the needed SSCP-SSCP session is active, the session request is retried.

To cancel the autologon specification, a VARY NOLOGON command can be issued in the originating logical unit domain, or the logical unit can log on to a controlling LU. For nonautologon sessions, the operator receives notification of the failure, but the session attempt is not automatically retried.

Before logical units in one network can establish sessions with logical units in another network, successive sessions between gateway SSCPs connecting the networks must be established.

For an SSCP-SSCP session, if an NCP on the originating side of a gateway is not available, a session activation request remains pending. If an NCP is not available on the destination side, a session activation request fails.

In interconnected networks, the operator can test routes starting in a gateway NCP by using the NETID operand of the command with the ORIGIN operand.

Resources automatically activated by VTAM

Certain resources are automatically activated by VTAM. Some internally maintained resources are automatically activated when the message “VTAM

INITIALIZATION COMPLETE" is issued. These resources can be displayed, but cannot be activated or deactivated by an operator. The following resources are automatically activated:

- VTAMSEG application program major node:
  - VTAM (or name from the CDRM definition statement for this VTAM)
  - ISTATA00
  - ISTNOP
  - ISTPDCLU
  - ISTAPNCP
- VTAMSEG2 application program major node:
  - *?-?* (model application program definition for Telnet server shared ACBs)

  Note: The definition of the model application program for Telnet server shared ACB names cannot be displayed.
- ISTPUS PU (or name from HOSTPU start option) type 5 node
  - ISTGROUP
- ISTPDILU predefined independent LU major node
- ISTADJCP adjacent CP major node
- ISTCDRDY dynamic cross-domain resource major node

  Note: The ISTCDRDY major node can be deactivated and activated by an operator. For further information, see "Dynamic definition of independent LUs" on page 276.
- ISTRTPMN rapid transport protocol major node
- ISTTRL transport resource list major node
- ISTLSXCF local SNA major node

  Note: ISTLSXCF can also be deactivated and activated by the operator.

VTAM dynamically builds and activates transport resource list elements (TRLEs) within the ISTTRL major node for some TCP/IP communication interfaces. All of these TRLEs are created when needed, but cannot be deleted. These dynamic TRLEs are created with the following naming convention:

IUTSAMEH
: This TRLE is created for communication between multiple TCP/IP stacks on the same MVS image, and for communication between TCP/IP and VTAM for Enterprise Extender.

IUTIQDIO
: This TRLE is created for TCP/IP dynamic XCF communications over HiperSockets devices. Up to 10 subchannel addresses are allocated: one READ and one WRITE device, and eight DATAPATH devices.

IUTIQD*xx*
: This TRLE is created when TCP/IP activates an MPCIPA HiperSockets device with a CHPID parameter of *xx*. Up to 10 subchannel addresses are allocated: one READ and one WRITE device, and eight DATAPATH devices.

IUTIQX*xx*
: This TRLE is created for TCP/IP dynamic Internal Queued Direct I/O extensions (IQDX) function IPv4 communications over HiperSockets devices that are connected to the intraensemble data network (IEDN). The value *xx* is the OSX CHPID number that is associated with this IQDX TRLE. Up to 10 subchannel addresses are allocated: one READ and one WRITE device, and eight DATAPATH devices.

IUTIQ6*xx*
This TRLE is created for TCP/IP dynamic IQDX IPv6 communications over HiperSockets devices that are connected to the intraensemble data network (IEDN). The value *xx* is the OSX CHPID number that is associated with this IQDX TRLE. Up to 10 subchannel addresses are are allocated: one READ and one WRITE device, and eight DATAPATH devices.

IUTXT0*xx*
This TRLE is created when TCP/IP activates an MPCIPA INTERFACE with CHPIDTYPE OSX and a CHPID parameter of *xx*. Up to 19 subchannel addresses are allocated: one READ and one WRITE device, and 17 DATAPATH devices.

IUTMT0*xx*
This TRLE is created when TCP/IP activates a dynamically defined OSM interface, where VTAM assigned CHPID *xx* for this communication. Up to 11 subchannel addresses are allocated: one READ and one WRITE device, and nine DATAPATH devices.

ISTT*lsrs*
TRLEs of this name are created when VTAM is started with either XCFINIT=YES (the default) or XCFINIT=DEFINE and another VTAM joins the XCF group (ISTXCF).

- *ls* is the two character &SYSCLONE value of the VTAM on the local MVS image
- *rs* is the two character &SYSCLONE value of the VTAM on the partner MVS image.

TCP/IP uses these TRLEs in one of the following situations:

- DYNAMICXCF is specified on the IPCONFIG or IPCONFIG6 statement and device or interface definitions are dynamically created to other VTAMs with XCF connectivity.
- DEVICE/LINK statements of type MPCPTP contain a device name that is the CPNAME or SSCPNAME of another VTAM with XCF connectivity.
- INTERFACE definitions of type MPCPTP6 contain a TRLENAME that is the CPNAME or SSCPNAME of another VTAM with XCF connectivity.

IUT*tdddd*
This TRLE is created when TCP/IP activates a CDLC, CLAW, Hyperchannel, CTC, or LCS device.

- *t* identifies the type of device that is dynamically created:
  - C - TCP/IP CDLC
  - W - TCP/IP CLAW
  - H - TCP/IP Hyperchannel
  - X - TCP/IP CTC
  - L - TCP/IP LCS
- *dddd* identifies the read device address for this device.

Activating application programs

Before an application program can use VTAM services, the application program minor node that defines the application program must be activated. An application program minor node is typically activated at VTAM startup by placing the name of the application program major node that contains the application program minor node into the VTAM configuration list.

After an application program minor node is activated, it is up to the application program to open its ACB and begin using VTAM services. The order in which the minor node is activated and the application program is started is not critical, but the minor node must be active when the application program attempts to open its ACB for VTAM.

If the operator cancels an application program, the operating system forces the application program disconnection from VTAM. For the operator to be able to cancel an application program that is active to VTAM, the operator must first associate the application program defined VTAM name with the application program job name.

The logon manager, which runs as a VTAM application, can be terminated with the MODIFY STOP command. For information on using the MODIFY STOP command, refer to *z/OS Communications Server: SNA Operation*.

Monitoring the domain

VTAM automatically provides you with messages that contain information about the VTAM domain. If more information is required, various DISPLAY commands are available to request status information about domain resources, and to verify changes resulting from previous operator requests.

Using the DISPLAY command

In general, DISPLAY commands can be used to obtain status information about active major nodes and their minor nodes (active or inactive). Some DISPLAY commands can be used to obtain information about resources that are not nodes. For example, the DISPLAY BFRUSE command provides information about VTAM use of buffers, and the DISPLAY COS command provides information about the network Class of Service tables.

Monitoring I/O problems

You can change how long some VTAM I/O operations and internal procedures remain pending before VTAM notifies you. To do this, you use the IOINT start option, the MODIFY,VTAMOPTS command, or the MODIFY,IOPD command to change the I/O problem determination timeout interval. For example, if you use the command `MODIFY,IOPD,IOINT=120`, VTAM notifies you of I/O operations or internal procedures that remain pending for 120 seconds or more. After you are notified of a pending I/O operation, you can take any necessary action.

If you do not want to be notified of pending operations, you can specify IOINT=0. This deactivates the I/O problem determination function.

For information about the IOINT start option, refer to the *z/OS Communications Server: SNA Resource Definition Reference*. For information about the MODIFY VTAMOPTS and the MODIFY IOPD commands, refer to *z/OS Communications Server: SNA Operation*. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for more information about the IOPD trace facility.

Suppressing messages

To help you control the amount of information you receive when monitoring the network, VTAM allows some messages to be suppressed. Messages are divided into the following categories based on suppression level (in order, from the lowest suppression level to the highest):

1. Informational
2. Warning
3. Normal
4. Serious
5. Insuppressible

The categories to be suppressed can be specified, either with the SUPP start option or with the MODIFY SUPP command. By specifying one category, VTAM suppresses all messages in that category and all messages in lower categories. For example, if the operator issues MODIFY SUPP=NORM or if you code SUPP=NORM in the start options, all messages within the normal category and those below it, warning and informational, are suppressed.

Note: You cannot suppress messages classified as "insuppressible." Examples of insuppressible messages are those that solicit information from the operator, and messages that are responses to DISPLAY commands.

The MODIFY SUPP command changes only the categories of messages that are suppressed, not the suppression level of messages. You can change the suppression level of a message by using USS facilities.

For information on the suppression level of each VTAM message, refer to *z/OS Communications Server: SNA Messages*.

Message flooding prevention

VTAM messages that could potentially flood the operator console can be suppressed by a message flooding prevention facility. This facility recognizes and filters duplicates occurring within a given time span, thus preventing critical information from being buried in a mass of repetitious information.

You can choose which messages are candidates for message flooding prevention by copying, renaming, and modifying VTAM default table (ISTMSFLD) to create your own message-flooding prevention table. Suppression criteria (variable text similarity) and suppression time intervals may be specified for each message in your table, and messages selected for suppression can also be suppressed from the hardcopy log. Message-flooding prevention support also provides the ability to modify a message flooding table dynamically.

For complete information on how to define a message-flooding prevention table, refer to the *z/OS Communications Server: SNA Resource Definition Reference*. For information on modifying (MODIFY TABLE) and displaying (DISPLAY TABLE) the message-flooding prevention table, refer to *z/OS Communications Server: SNA Operation*.

Other methods of controlling messages

Some VTAM messages can also be controlled using the following VTAM start options:

- ASIRFMSG
- CNNRTMSG
- DSIRFMSG
- DSPLYDEF
- DSPLYMAX
- ESIRFMSG
- FSIRFMSG
- HPRITMSG

- HPRPSMSG
- IOMSGLIM
- LSIRFMSG
- PLUALMSG
- RSIRFMSG
- SIRFMSG
- SLUALMSG

For information on these start options, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Displaying and testing routes

There are two VTAM commands used for displaying and testing routes, DISPLAY ROUTE and DISPLAY APING.

DISPLAY ROUTE command

The DISPLAY ROUTE command provides information about what routes are available between the VTAM host node and a destination subarea node. In addition, the TEST operand of the DISPLAY ROUTE command allows the operator to test the explicit routes between the host node and a destination subarea node for the routes' ability to transfer data between the subareas. The routes to be displayed can be specified by explicit route number, virtual route number, or Class of Service name. If the TEST operand is specified and the routes to be displayed are specified by virtual route number, the explicit routes tested are those to which the specified virtual routes are currently mapped. If the TEST operand is specified and the routes to be displayed are specified by Class of Service, the explicit routes tested are those to which the virtual routes in the specified Class of Service are currently mapped.

You can also test explicit routes starting in an NCP. The ORIGIN operand on the DISPLAY ROUTE command specifies the NCP in which the route starts. Although the originating node must be NCP Version 3 or later, the subarea at the other end can be any of the products that can coexist with VTAM Version 3 or subsequent releases.

DISPLAY APING command

The DISPLAY APING command enables the network operator to do the following:

- Verify connectivity with any LU 6.2 resource in the network
- Verify that another VTAM 4.3, or later, node is operational
- Check the performance of the network using a particular logon mode
- Display routing information to the destination node, if a new session is established for the APING transaction

For further information, see “APING support” on page 421.

Defining operator messages and commands

You can modify certain messages and operator commands. These commands and messages are defined in unformatted system services (USS) tables and are called USS messages and USS commands. Whenever VTAM receives a USS command, it uses either an IBM-supplied default USS table or a user-written USS table to process it. Similarly, whenever VTAM is to send a USS message, it uses one of these tables to determine the message level (the message associated with this release of VTAM or a previous release), message text, and other characteristics of the message. The IBM-supplied operation-level USS table is named ISTINCNO.

By creating a supplementary USS table, you can change the text or other characteristics of a message or the syntax or default values for a command. You can create supplementary tables using USS macro instructions to redefine the VTAM commands or messages that you want to change. To use your USS table, specify the name of the table on the USSTAB start option. If you specify a user-defined table for operator commands, the ISTINCNO table is not used. If you specify a user-defined table for operator messages, the ISTINCNO table is used in conjunction with the user-defined USS table. For more information on USS tables, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Note: Because program operators depend on VTAM messages, changes to the operator messages could disrupt the functioning of a program operator. If you change an operator message, notify the responsible system programmer.

You can specify an operation-level USS table for a program operator using the SSCPFM and USSTAB operands of the program operator APPL definition statement.

Multiple console support (MCS) in VTAM

An operator can enter VTAM commands and receive messages at the system master console and at designated secondary consoles. To enter VTAM commands at a secondary console, the console must be authorized to accept commands from command group 1 (system control group) and command group 2 (I/O control group).

A VTAM command response goes to the console where the command was entered. Depending on their contents, VTAM messages use routing codes 1, 2, 4, 6, 8, or 10.

Therefore, a secondary console should be assigned routing codes based on the kinds of VTAM messages that are to be routed to that console.

To determine the routing codes used by particular messages, refer to *z/OS Communications Server: SNA Messages*.

The operator can change command authorization and message routing using the MVS command VARY. The following command, for example, designates a secondary console where any VTAM command can be entered and where any VTAM message that has not been suppressed can be received:

```
VARY 01F,CONSOLE,AUTH=(SYS,IO)
```

The channel device name of the console is 01F. It is authorized to enter commands from Command Group 1 (SYS) and Command Group 2 (IO). This console receives all messages that have not been suppressed.

Details about using the VARY command to designate a secondary console are contained in the MVS reference document *z/OS MVS System Commands*.

Controlling the domain

Part of operating VTAM is the task of starting and stopping sessions under special conditions, controlling incoming and outgoing calls, and replacing or reassociating tables that affect how VTAM operates.

Establishing and terminating sessions with operator commands

A session is usually established by an LU requesting session establishment. However, you can establish a particular session or set of sessions using the VARY LOGON command, or you can establish a single session between a user and an application program using the LOGON operand of the VARY ACT command. If the logical unit you are activating is defined with a LOGAPPL on its LU definition statement, you can establish a session without using the LOGON operand.

A session is usually terminated by one of the session partners requesting termination. However, you can terminate a particular session or set of sessions without deactivating either of the session partners using the VARY TERM command. You can also terminate a session using the VARY INACT command for one of the logical units participating in the session. This command with a TYPE=IMMED or TYPE=FORCE operand breaks any sessions with the logical unit. (Note that TYPE=IMMED has the same effect as TYPE=UNCOND.)

Note: For cross-network sessions, when any SSCP-SSCP session along a cross-network session path of an LU-LU session is disrupted, the only termination requests that can be handled are those that are forced, such as VARY INACT with the FORCE operand or VARY TERM with the FORCE operand.

Dynamic table replacement

The MODIFY TABLE operator command allows you to refresh, replace, or reassociate VTAM tables without halting and restarting VTAM. The following tables are affected by the MODIFY TABLE operator command:

- Associated LU tables
- Class of Service (COS) tables
- Interpret tables
- Logon mode tables
- Message-flooding prevention tables
- Model name tables
- Session awareness data filter tables
- USS tables

Before you can use the MODIFY TABLE command to replace a USS or interpret table, the new table must be assembled using the VTAM macro libraries.

Deactivating resources

This section provides an overview to deactivating VTAM resources with a VARY INACT command or a VARY REL command. For additional information on deactivating an NCP, see “Deactivating an NCP” on page 653.

Order of deactivation

VTAM deactivates resources in the opposite order from activation. When VTAM receives a command to deactivate a resource, it first deactivates any resources subordinate to the named resource, moving from the bottom of the hierarchy upward, and then deactivates the named resource. Deactivating resources can be disruptive to existing LU-LU sessions. However, depending on the type of connection, a resource can be deactivated without disrupting the LU session if you use TYPE=GIVEBACK on the VARY INACT or the VARY REL command.

Deactivation of certain VTAM resources requires more attention than does their activation. Be aware of the changing connectivity of the domain and the effects on the rest of the network of NCP major node and channel-attachment major node deactivation. Also consider the effects of cross-subarea link and link station deactivations on the operation of the rest of the network. In particular, you should be aware of the impact of deactivating any of the following:

- An NCP through which other NCPs are attached to the network
- A link or link station, whether channel or SDLC, that provides the only connection between a subarea and the rest of the network

VTAM deactivates a cross-subarea link to a host processor when the channel-attachment major node is deactivated. Deactivating this link (a channel) destroys any sessions using a route on that link, possibly including the SSCP session with another subarea node. Also, VTAM might be an intermediate routing node for cross-domain sessions that traverse the link. For this reason, care should be taken not to deactivate a cross-subarea link that could be supporting session traffic.

Note: When deactivating any channel-attachment major node, or any of its subordinate links or link stations, you might be deactivating a link that carries intermediate routing node traffic. Deactivating the link will disrupt these sessions.

When you enter a VARY INACT command for an NCP, VTAM does not deactivate any cross-domain link or link station unless CDLINK=INACT is specified on the VARY INACT command.

You can deactivate and reactivate CDRMs without disrupting active sessions. Use the SAVESESS operand on the VARY INACT command to specify whether active sessions should be terminated when you deactivate the CDRM. This operand enables you to keep active sessions when you use the FORCE or IMMED option of the VARY INACT command. However, SAVESESS specifies that all pending or queued sessions using the specified CDRM are terminated.

The loss of an SSCP-SSCP session (for example, as the result of a link failure) is not the same as external cross-domain resource manager deactivation and does not terminate existing LU-LU sessions. However, new sessions with logical units controlled by the SSCP that were lost cannot be started until the SSCP-SSCP session is reestablished.

Automatic deactivation

A VARY INACT command for an NCP major node can result in the automatic deactivation of cross-subarea links and link stations in adjacent subarea nodes (other NCPs or VTAM itself). For example, if the operator enters a command to deactivate an NCP, VTAM automatically deactivates any links and link stations that were automatically activated in adjacent nodes, provided these adjacent links and link stations were not also directly or indirectly activated.

VTAM does not automatically deactivate a multipoint subarea link, even if all physical units on the line have been deactivated.

Normal deactivation

Normal deactivation is the default for the VARY INACT command. When normal deactivation is requested, resources are not actually deactivated by VTAM until

LU-LU sessions associated with the resources have been terminated. Active and pending-active sessions are not affected by normal deactivation of a resource. All queued requests for sessions involving the resources to be deactivated are discarded and the initiators are notified. No new requests for sessions with these resources are accepted.

Immediate deactivation

Immediate deactivation is specified with the TYPE=IMMED operand of the VARY INACT command. When immediate deactivation is requested, LU-LU sessions associated with the resources are effectively broken. All queued requests for sessions involving the resources to be deactivated are discarded, and the initiators are notified. No new requests for sessions involving these resources are accepted. All input and output operations for active LU-LU sessions are immediately halted, with possible loss of data. If the deactivated logical unit is in session with an application program, the application program is notified of the deactivation. The application program must complete the session termination for the deactivation to be completed.

Immediate deactivation provides tight control over the domain; normal deactivation provides less stringent control, but allows for a more orderly deactivation. Normal and immediate deactivations are not completed until the proper SNA requests and responses are exchanged between the VTAM SSCP and the affected resources. If conditions in the domain prevent the sending of the proper requests or responses, normal and immediate deactivation might not be successful.

Forced deactivation

Forced deactivation is specified with the TYPE=FORCE operand of the VARY INACT command. When forced deactivation is requested, all of the actions described for immediate deactivation apply. An application program notified of the termination of a session with a deactivated logical unit might have to complete the session termination (depending on which user exits have been coded for the program) for the deactivation to be completed. Forced deactivation can be used when immediate and normal deactivation are not successful.

Forced reactivation

Forced reactivation is specified with the TYPE=REACT operand of the VARY INACT command. When forced reactivation is requested, VTAM performs a forced deactivation and then attempts to reactivate the resource. However, for lines, PUs, and LUs, VTAM does wait for responses from the resource. Forced reactivation of an NCP reactivates the SSCP-PU session, but has no effect on lines and link stations in adjacent subarea nodes. (In this case, the forced reactivation command should be entered for the adjacent line or link station if one of these resources is the source of the problem.) Forced reactivation does not apply to all types of resources.

Halting VTAM

You can halt VTAM using the HALT or HALT QUICK commands. When VTAM is halted, all of its resources are deactivated, and all sessions involving those resources are terminated. Additionally, if the VTAM being halted is an intermediate routing node, any sessions passing through that VTAM are disrupted. Sessions running under the IBM Network Routing Facility program might be disrupted if

the VTAM host is halted. This is true in any situation where the session ends are not in the host subarea. If the HALT or HALT QUICK commands do not execute properly, the operator can cancel VTAM.

The CDLINK operand for the HALT command is the same as for the VARY INACT command, except that the cross-subarea links to which it applies are determined when the HALT command is entered and are not redefined as the domain contracts. That is, the CDLINK operand applies to all cross-subarea SDLC links between NCPs that are cross-domain at the time the HALT command is entered.

Except for channel links, all other cross-subarea links (those that are entirely within the domain at the time the HALT command is entered) remain active during and after the halt, regardless of how CDLINK is specified. That is, their final status does not depend on whether the links and their link stations were automatically, directly, or indirectly activated. This allows sessions using that link to remain active. It also allows nondisruptive, simultaneous NCP deactivations, regardless of how the communication controllers are connected within the domain. Cross-subarea channel links are always deactivated during a VTAM halt.

The HALT command notifies application programs within the VTAM subarea of the domain shutdown and waits for them to close their ACBs. Intermediate routing node traffic passing through the VTAM being halted will be disrupted without notification.

New sessions are not permitted and new ACBs cannot be opened for any resources of the host that have received a HALT command. Except for this restriction, application programs of the host that have received a HALT command can continue their current operations. Any pending-active sessions are terminated, but active sessions remain active. After all application programs have stopped using VTAM services (that is, after they have closed their ACBs), processing is the same as for the HALT QUICK command.

During a normal halt, you can monitor the shutdown of the domain by displaying the status of application programs and other logical units. The VARY TERM command can be used to terminate sessions, and, if necessary, to speed the processing of the HALT command.

The HALT QUICK command can also be used at this time, if necessary, to speed halt processing. The HALT QUICK command notifies application programs of the domain shutdown and then proceeds with the equivalent of the VARY INACT,TYPE=IMMED command for each major node. All active LU-LU sessions are disrupted. After VTAM has received a HALT QUICK command, the only VTAM commands that can be used are DISPLAY commands, VARY TERM commands (to terminate sessions), and VARY INACT,TYPE=FORCE commands (to force the deactivation of resources).

If during HALT QUICK processing, after all the major nodes are inactive, any application programs have not yet closed their ACBs, VTAM displays a message listing the names of these application programs. Because VTAM cannot shut down the domain until all application programs have been disconnected from VTAM, you might want to speed up the halt process by canceling the jobs of application programs with open ACBs. The HALT QUICK command can prevent an application program from using VTAM services, but VTAM cannot force the program to disconnect itself from VTAM. When the operator cancels an application program, the host operating system disconnects it from VTAM.

Canceling VTAM

It might be necessary to cancel VTAM if VTAM cannot be halted in any other way. VTAM should be canceled only if the HALT and HALT QUICK commands do not work properly. Canceling VTAM causes the immediate abnormal termination of VTAM without an orderly shutdown of the domain.

The HALT CANCEL command cancels VTAM. This command depends only on the proper functioning of the abnormal termination facilities on the host operating system. The HALT CANCEL command causes an abnormal termination with a system completion code of X'0A9'. When VTAM is canceled, no further I/O operations occur. Application programs using VTAM are notified of a VTAM shutdown through their TPEND exit routines, if possible. Data being transmitted on sessions might be lost. If TPEND cannot be scheduled for an application program, the application program is also abnormally terminated, if possible. In some circumstances, you might have to cancel some application programs.

Note: Any multinode persistent capable application programs will go into recovery pending state if the application is currently enabled for persistence.

Automatic operations

You can use special application programs to assist the VTAM operator in controlling a VTAM domain. These application programs are either program operators or communication network management (CNM) application programs.

Program operators

A program operator is an application program that is authorized to issue certain VTAM commands and receive VTAM messages. A program operator can be used to:

- Enter VTAM commands (except for commands to start or halt VTAM) from a terminal in the domain
- Monitor and control elements in the domain at program execution speed
- Define specialized commands (for example, to display the status of the entire domain)
- Reformat responses to VTAM commands (such as to reformat the status display of the entire domain to fit on a 3270 display screen)

Secondary and primary program operators

A program operator can have varying degrees of network management capability, depending on the options specified in its APPL definition statement. If you code AUTH=SPO (secondary program operator) or PPO (primary program operator), the application program can issue VTAM operator commands (except START and HALT) using the SENDCMD and RCVCMD macro instructions. The application program can then monitor and control the VTAM domain.

An application program that you specify as a primary program operator (PPO) receives all unsolicited messages, that is, all informational and error messages that are not replies to operator commands. If no application program that is designated as PPO is active when these messages occur, they are directed to the system console.

An application program that you specify as a secondary program operator (SPO) can receive solicited messages, that is, messages that are in reply to VTAM

operator commands issued by that program operator only. You can, however, limit the number of messages that can be queued for a program operator by using the POAQLIM operand on the APPL definition statement.

CNM application programs

A communication network management (CNM) application program can request and receive information that might be useful in managing an SNA network. The application program can communicate with certain physical units in the network to obtain status information. *z/OS Communications Server: SNA Programming* describes in detail the macro instructions used to create a program operator.

CNM routing table

VTAM refers to a communication network management (CNM) routing table to determine which CNM application program is to receive an unsolicited network services request unit (NSRU) requiring further processing.

An application program can put its own procedure-related identifier (PRID) in each request sent to VTAM. When a reply to the request is returned, VTAM uses the PRID to route the reply to the application program. However, unsolicited request units contain network information, but they do not have a PRID. Therefore, VTAM needs the CNM routing table to determine which application program is to receive the unsolicited request unit.

VTAM provides default routing information for IBM CNM licensed programs, such as the NetView program. If you authorize a user-written application program to use the CNM interface, you should write a supplementary CNM routing table to route unsolicited requests to that application program.

For information on the request units for which VTAM provides default routing, refer to *z/OS Communications Server: SNA Customization*.

NetView program and other CNM application programs

The NetView program can be used as a CNM application program or as a program operator. The hardware monitor and session monitor, which are components of the NetView program, can also be used as CNM application programs. Because the NetView program performs the functions of a program operator, code the program operator and CNM interface as authorized (AUTH operand on the APPL definition statement).

AUTH=CNM must be coded for the following application programs:

- NetView session monitor
- NetView alias name translation facility
- Network Logical Data Manager (NLDM) licensed program

Defining a CNM application program

To define a CNM application program, code AUTH=CNM on the APPL definition statement. The application program is then authorized to use the CNM interface to communicate with physical units in the domain, using embedded network services request units. Such an application program can use the CNM interface to gather error or link test information.

Collecting session awareness (SAW) data

Session awareness (SAW) data includes all information about sessions that is known by VTAM. A SAW data filter specifies which application programs receive SAW data and which do not. It can also control from which logical units the CNM application programs receive data.

Using the default SAW data filter

The default SAW data filter table allows data for all sessions to be passed across the CNM interface. You use the default filter only to reduce the amount of data that is passed to the session monitor, and therefore increase the amount of data that is sent to the CNM application programs.

ISTMGC10 in VTAMLIB contains the default SAW data filter. You can use the default table, modify the default table or dynamically change it using the MODIFY TABLE operator command, or you can replace it with one of your own.

NetView customers can use the default filter with the session monitor filter. The default filter then supplements the NetView filter. The session monitor filter controls the actual processing of NetView session awareness data and is still needed.

Coding your own SAW data filter

If you want more control over the amount of data that is filtered, you can use your own SAW data filter table, or you can modify the default table or dynamically change it using the MODIFY TABLE operator command.

You define a SAW data filter to VTAM using the KEEPMEM, KCLASS, and MAPSESS macros. Use the KCLASS and MAPSESS macros to define the session monitor filter. NetView customers can copy their existing filter definitions, modify them slightly, and create equivalent VTAM filters. The only change necessary is to place a KEEPMEM START macro at the beginning of the filter definitions and add a KEEPMEM STOP macro and an END statement at the end of the existing definitions. VTAM ignores any NetView operands that do not apply to the VTAM filter.

Following is a simple example of a SAW data filter:

```
ISTMGC10  KEEPMEM  START
SAW       KCLASS   SAW=YES
NOSAW     KCLASS   SAW=NO
M1        MAPSESS  KCLASS=SAW,PRI=SSCP1,SEC=*
M2        MAPSESS  KCLASS=SAW,PRI=IMS,SEC=*,PRINET=NETA
M3        MAPSESS  KCLASS=NOSAW,PRI=*,SEC=T3?7*,PRINET=*,SECNET=NETB
          KEEPMEM  STOP
          END
```

The KCLASS macro instruction specifies whether to pass session awareness data over the CNM interface (SAW=YES|NO). The labels given to the KCLASS macro instructions are specified on the MAPSESS macro instruction. MAPSESS specifies a PLU-SLU session pair and specifies the KCLASS definition that VTAM uses to determine whether to pass session awareness data over the CNM interface.

The PLU and SLU are derived from the PRI and PRINET operands, and the SEC and SECNET operands respectively. Each has a maximum length of eight characters. The special characters ? and * used in the pattern have the following meanings:

? Matches any single letter or digit

* Matches any letters or digits and can be used only as the last character in the pattern

The order of macro instructions is significant. VTAM examines the MAPSESS definitions from top to bottom when searching for a match for the PLU-SLU pair. The macro instructions in the simple example of a session awareness data filter table cause VTAM to:

- Pass session awareness data to the NetView session monitor for a session with a PLU named SSCP1 or IMS
- Ignore session awareness data for a session with an SLU whose name matches the T3?7* pattern if the PLU is not named SSCP1 or IMS
- Pass session awareness data for a session with IMS as the PLU, even if the SLU name matches the T3?7* pattern, because the M2 definition is before the M3 definition

Eliminating session initialization failure data: You can request that certain session initiation failure data not be sent to application programs collecting SAW data. By coding a SAW sense filter, you enable the SAW data to be filtered by sense code. Use the SNSCODE definition statement to specify either two or four bytes of the sense data code to be filtered. The following is a sample SAW sense filter:

```
SNSTBL1   VBUILD   TYPE=SNSFILTR
SENSE0    SNSCODE  SENSE=0857
SENSE1    SNSCODE  SENSE=087D0001
SENSE2    SNSCODE  SENSE=087D0002
SENSE3    SNSCODE  SENSE=087D0003
SENSE4    SNSCODE  SENSE=087D0004
SENSE5    SNSCODE  SENSE=8013
```

Based on this example, VTAM will not send:

- All sense data beginning with X'0857'
- The following sense data:
  - X'087D0001'
  - X'087D0002'
  - X'087D0003'
  - X'087D0004'

  All other sense data beginning with X'087D' will be sent.
- All sense data beginning with X'8013'

Store the SAW sense filter in the VTAM definition library. Use the VARY ACT command to activate the SAW sense filter or to replace the current table. Only one SAW sense filter can be active in each VTAM host at the same time.

Operating VTAM in a multiple-domain subarea network

This section contains VTAM operation considerations for a multiple domain subarea network.

Links and link stations

In Figure 152 on page 627, if NCP2 is inactive and LINK1 is active, the operator at HOST1 can dump NCP2 by specifying ID=LINKSTA1 on the MODIFY DUMP command. If HOST2 activates NCP3, HOST2 can dump NCP2 by specifying ID=LINKSTA4.

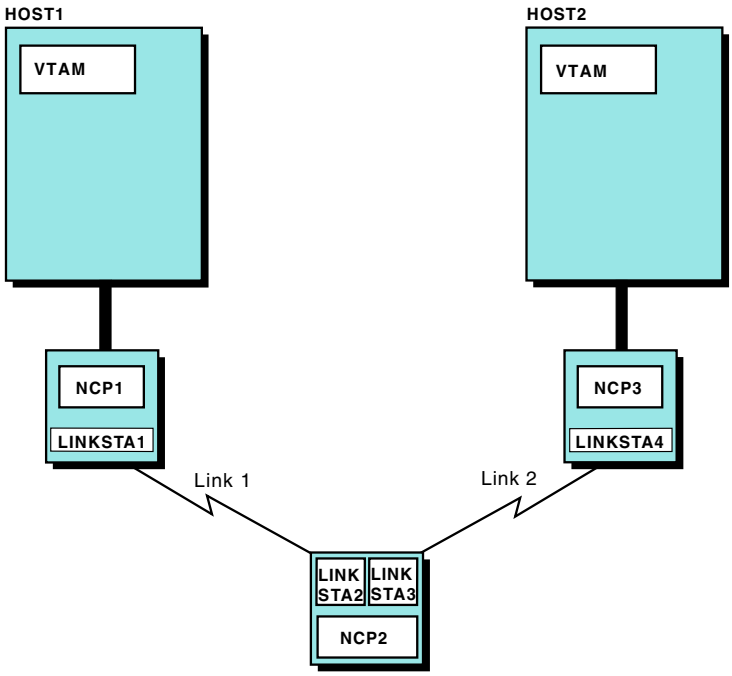


Figure 152. Effects of NCP deactivation on cross-subarea links and link stations

Activating links

The same-domain link between NCP1 and NCP2 in Figure 152 is made available through the activation of LINK1 and LINKSTA1 by HOST1. This link and link station can be:

- Directly or indirectly activated. (For example, you can enter specific VARY ACT commands for LINK1 and LINKSTA1, or LINK1 and LINKSTA1 can be defined to be initially active in NCP1 definition statements.)
- Automatically activated when NCP2 is activated (by specifying LINKSTA1 on the RNAME operand of the VARY ACT command or on the RNAME operand in the PCCU definition statement for NCP2).

Note that in Figure 152, the activation of LINK2 and LINKSTA2 by HOST1 has no effect on the availability of the link between NCP1 and NCP2. The activation of LINK2 and LINKSTA2, which is required to make the link itself available, must occur by SDLC monitor mode being specified for LINK2 in NCP2 definition, or by HOST2 activating NCP2, LINK2, and LINKSTA2. If LINK2 and LINKSTA2 are activated by HOST1, HOST1 VTAM does not deactivate them when NCP2 is deactivated. In this case, VTAM loses its ownership of LINK2 and LINKSTA2 within NCP2 by virtue of the deactivation command sent to NCP2 itself.

Deactivating links

If LINK1 and LINKSTA1 are active when NCP2 is deactivated, LINK1 and LINKSTA1 are automatically deactivated, unless you directly or indirectly activate them.

Therefore, if the link between NCP1 and NCP2 continues to carry session traffic after NCP2 is deactivated by HOST1, you should ensure that LINK1 and LINKSTA1 are directly or indirectly activated. This can be verified by entering a DISPLAY command for the link or link station (a DISPLAY of either resource

displays the status of both). As a result, the only path available for session traffic to flow using the link between NCP1 and NCP2, given that HOST1 is deactivated, is from NCP1 to NCP2 to NCP3 to HOST2 for PLUs in the host. If PU type 2.1 nodes are attached to NCP1, NCP2, or NCP3, that link can also be used for session traffic.

The order in which the NCPs are deactivated determines how a VTAM domain contracts. The deactivation of an NCP by a VTAM host removes that NCP from the domain of that host. After an NCP is deactivated in a domain, all links leading to it are considered cross-domain links. During an NCP deactivation, then, same-domain links might become cross-domain links. If two or more NCPs are deactivated simultaneously (except as a result of a HALT command; see "Halting VTAM" on page 621), the time when the change occurs is unpredictable, and it might be difficult to apply the previously described rules for not disrupting sessions. Unless these rules are followed so all links have the same final disposition (active or inactive), regardless of which rule applies, do not deactivate the NCPs simultaneously.

Cross-domain links

The cross-domain link between NCP2 and NCP3 is made available through the activation of LINK3 and LINKSTA3 by HOST1. Because this link is a cross-domain link, by definition, NCP3 is not activated by HOST1, and LINK3 and LINKSTA3 cannot be automatically activated. Therefore, a cross-domain link and link station must be directly or indirectly activated. When NCP2 is deactivated and CDLINK=INACT is specified in the VARY INACT command, VTAM indirectly deactivates all cross-domain links and link stations (because they are below the NCP in the resource hierarchy). LINK2 and LINKSTA4 must have become active in NCP3 either through the SDLC monitor mode function or through activation by HOST2.

Delayed activation of logical lines

This function enables you to perform an orderly activation of a physical resource and its associated logical resources. Because there is a hierarchical relationship between the physical resources (line and physical unit) and the logical lines, you can use this function to implicitly activate a logical line when the physical resources are explicitly activated.

The function creates a hierarchy between a logical line in a group (within an NCP major node) that has a PHYSRSC operand coded and the physical resource indicated by the PHYSRSC operand. This includes the following:

- NCP/Token Ring interconnection (NTRI) physical resources (line and physical unit) and logical lines for INN link stations
- NCP Packet Switched Interface (NPSI) physical resources (line and physical unit) and logical lines
- Frame-relay physical resources (line and physical unit) and logical lines

If the OWNER operand is specified for both the physical unit and the logical line, the logical line is subordinate to the physical unit only when the owner of the physical unit is an owner of the logical line and the PHYSRSC operand is coded for the logical line group. Remember that a physical unit can have only one owner, but a nonswitched logical line can have multiple owners. If the OWNER operand is not coded for both the higher-level physical unit and the logical line, the logical line is always hierarchically subordinate to the physical unit, if this VTAM can be the owner of both resources. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for information about the OWNER operand.

Because only one VTAM can own the physical unit, yet the physical unit can be built into the hierarchy in multiple VTAMs (unless the OWNER operand is specified), only one VTAM can have the physical unit active and have logical lines subordinate to the physical unit. When explicit owners are not specified, if another VTAM owns a higher-level physical unit and the physical unit is activated by a second VTAM, the activation request issued by the second VTAM is rejected by the NCP. In this case, the second VTAM removes the higher-level physical unit from the hierarchy of all its subordinate logical lines. In this way, the logical lines within the second VTAM can still be activated by a mechanism other than being subordinate to the physical unit. After the hierarchy is broken in this way, it is not reconstructed even if this VTAM successfully reactivates the physical unit at a later time.

When the higher-level physical unit is a switched resource, not only is the physical unit limited to one owning VTAM, but the switched line can only have one owning VTAM as well. If a second VTAM attempts to activate the physical line, the activation request is rejected by the NCP. In this case, the second VTAM removes the higher-level physical PU from the hierarchy of all its subordinate logical lines.

For a switched physical line with a logical hierarchy, DWACT can be used in the switched major node definition or on the VARY ACT command issued for a switched major node. DWACT specifies whether to dial the switched PU when it is activated. This lets you avoid issuing a VARY DIAL command. DWACT can apply to both the switched physical PU and to one or more logical PUs in the logical hierarchy . Because the activation of the switched logicals cannot take place until the activation of the switched physical PU is complete, using DWACT can result in an activation failure for the switched logicals. For example, if the switched physical PU and the associated switched logical PUs are in the same switched major node, a VARY ACT of the switched major node with DWACT=YES results in a DIAL for the physical PU, but a DIAL failure for the associated logicals. However, a second VARY ACT of the switched major node with DWACT=YES results in activation of the associated switched logicals.

If there is a switched physical with a logical hierarchy, one of the following actions is recommended:

- Define the switched physical PU and the associated switched logical PUs in the separate switched major nodes, then activate the switched major node with the physical PU before activating the switched major node with the associated logical PUs. DWACT can be specified in the switched major nodes or on the VARY ACT command of the switched major nodes.
- Define both the switched physical PU and the associated switched logical PUs in the same switched major node, then activate the switched major node twice. To avoid error messages, DWACT can be specified on the physical PU in the switched major node (but not on the logical PUs in the switched major node). Then, issue a VARY ACT, DWACT=NO to activate the physical PU. Then issue a VARY ACT,DWACT=YES to activate the associated logical PUs.

For most multiple-domain environments, the OWNER operand should be specified for both the physical unit and the logical lines so that the owner of the physical unit and the set of subordinate resources is predetermined and does not vary. The OWNER operand provides a way to control activation while also providing a backup owner for the physical resource. This relationship remains even if there is a takeover of the NCP. For example, if the VTAM that owns the physical unit (and

possibly some logical lines) releases the NCP and another VTAM acquires the physical unit, the physical unit is placed hierarchically above any logical line acquired from the other VTAM.

Activating logical lines hierarchically
Activation processing in a multiple-domain environment is the same as in a single-domain environment except for the difference in hierarchical relationships that could exist because of the checking of the OWNER operand. When a logical line is subordinate to a physical unit, the physical unit must be activated before the logical line is activated. When the physical unit is owned by another VTAM, but this VTAM is to own a logical line subordinate to the physical unit, the activation of the physical unit must be attempted so that this VTAM can see that the physical unit is owned by another VTAM. Note that a logical line that specifies a different OWNER from its associated physical resource is not hierarchically subordinate to that physical unit. In this case, the activation of the physical unit need not be attempted before the logical line can be activated.

Deactivating logical lines hierarchically
Deactivation processing in a multiple-domain environment is the same as in a single-domain environment except for the difference in hierarchical relationships that could exist because of the checking of the OWNER operand. In addition, hierarchical relationships could be affected by the rejection of a activation request sent by a VTAM to a higher-level physical unit that is owned by another VTAM.

Discontiguous domains

Because the connectivity of a domain changes when an NCP is deactivated, you should be aware of the possibility of creating a discontiguous domain. In Figure 152 on page 627, for example, if HOST1 had activated NCP3 as well as NCP1 and NCP2, and if HOST1 then deactivated NCP2, NCP3 would be in an isolated part of HOST1 domain. That is, HOST1, NCP1, and NCP3 would all be part of the domain, but the only access to NCP3 from HOST1 would be through NCP2, which would not be in the domain. Continued session traffic might not be possible to resources in or beyond NCP3, depending on whether the deactivation of NCP2 caused deactivation of the cross-subarea links connected to NCP2.

Discontiguous domains should be avoided. As shown in Figure 152 on page 627, because HOST1 does not own the route to NCP3, the HOST1 operator does not get notification of failures of links and link stations in NCP2. However, the HOST1 operator always gets notification of explicit route failures. In addition, the ability for VTAM to recover link station failures in NCP2 is lost. Finally, if NCP3 fails, the HOST1 operator does not have the capability of reloading that communication controller.

To avoid a discontiguous domain, always first deactivate those NCPs that are farthest away from the host, working inward toward the host. In this example, HOST1 should deactivate NCP3 first, NCP2 second, and NCP1 last. NCPs need not be deactivated before halting VTAM to ensure that the proper order of deactivation is followed. VTAM follows special procedures during its halt processing to simultaneously deactivate all NCPs that are still active in the domain (see “Halting VTAM” on page 621).

If a discontiguous domain exists, all cross-subarea links connected to an isolated NCP are cross-domain links.

Note: If you are implementing an SDLC switched connection between NCP subareas in a multiple-domain network, the link station in each NCP can be controlled by VTAM, and you can implement call security verification. For information about implementing call security verification, see "Switched SDLC subarea connection" on page 108.

Backing up resource owners

Every resource in a network must be owned by an SSCP in some domain. When a host fails, another SSCP can, with the proper planning and definitions, assume ownership of the failing host resources. Any host can take over the resources of any NCP with which it has connectivity, and the takeover can be nondisruptive for sessions.

Steps for resource takeover

The following steps describe this process:

1. NCP and the physical and logical units that it is to back up, including resources defined in any dynamic reconfiguration files, should be already defined to the backup host.
2. The operator for the backup host issues the appropriate commands to acquire and activate the appropriate resources. This results in the takeover host activating the resources.
3. For dependent LUs, the backup host redefines to itself the CDRSCs it is backing up as LUs as each LU PU becomes active. Each statically-defined CDRSC definition representing a dependent LU in the failing domain becomes a shadow resource definition. Independent LUs continue to be defined as CDRSCs but are now displayable under the PU that is serving as their adjacent link station, if an independent LU has an active LU-LU session over that adjacent link station.

   Note: There are situations that cause a dependent LU to remain as a shadow resource following its PU becoming active. In those situations the LU is brought out of shadow once the LU-LU session is terminated.
4. CDRMs that have cross-domain resource definitions for the resources taken over by the backup host have these resources associated with the wrong owning CDRM name. Operators in these domains should use the MODIFY CDRM command to reassociate these CDRSCs with the new CDRM so that session-initiation requests can be routed to the proper domain or the CDRSC definitions in those domains can specify the VFYOWNER operand in conjunction with an ADJSSCP table to locate the CDRSC.

Restarting the host

To restart a host, both of the following must occur:

- VTAM representation of the network must be restored. Using configuration restart facilities, you can have it restored to the state it was in before the host failed. NetView command lists can be used to automate the restarting of the network resources.
- Sessions must be reestablished. Some sessions can continue despite the failure. For example, sessions between devices in this domain and application programs in other domains that have been defined to continue after automatic network shutdown of the NCP can continue. Of these, some sessions also continue when VTAM is started. If the device in session supports recovery, the session continues when VTAM reactivates physical units and LUs. If not, the session is disrupted.

Note: If the session is established with a nonextended BIND, the restarted VTAM loses knowledge of all sessions. Therefore, you should plan for the following two situations:

- Cross-domain sessions with peripheral LUs owned by the failing host can continue when VTAM fails, but without VTAM support. Those sessions that continue when VTAM restarts do so with limited VTAM support. That is, the DISPLAY operator command, when issued from the restarted host, provides only limited session information. Unformatted LOGOFF requests from these LUs must be specified as TYPE(FORCE) or TYPE(UNCOND) and must not refer to the session partner; that is, the APPLID operand must be omitted. Formatted Terminate Self requests from these peripheral LUs must be specified with the type code set to cleanup. TYPE=UNCOND or TYPE=FORCE must be specified in VARY TERM operator commands.
- Sessions that do not continue after VTAM failure must be reestablished by the operator or one of the session partners, as necessary.

Returning ownership

When the original resource owner is restarted, the backup host can release ownership of the resources using the GIVEBACK operand on the VARY INACT or VARY REL operator command. The original owning host can then resume ownership of the resources using the VARY ACT or VARY ACQ operator command. The operations of assuming ownership in the backup host, releasing ownership of the resources in the backup host, and resuming ownership of the resources in the restarted original host are all nondisruptive to the following session types:

- Nonswitched SDLC connections
- Switched SDLC connections
- Token-ring connections
- X.25 connections using permanent or switched virtual circuits
- T2.1 channel connections to APPN nodes
- Frame Relay connections

However, each of the devices attached using these connections must be capable of supporting the activation recovery.

Note: If you specify the DELAY option on the ANS operand in NCP, the disruption of binary synchronous (BSC) sessions can be delayed when the original host fails. However, the activation of the BSC resources by the backup host results in the disruption of these sessions.

Sharing an NCP

An NCP can be shared by more than one SSCP. A VARY ACT command for an NCP causes the VTAM SSCP to activate the NCP, establishing an SSCP-PU session with the NCP.

Owning resources

If multiple VTAMs activate an NCP, each of those VTAMs establishes an SSCP-PU session with the NCP, and each is considered an owner of that NCP. Each VTAM can also activate and share ownership of the NCP links and link stations. However, the NCP physical units and logical units can belong to only one VTAM at a time. The VTAM that activates a physical unit first becomes the exclusive owner of that physical unit; the other VTAM cannot activate that physical unit until the SSCP that currently owns the physical unit deactivates it.

Defining owners

The resources of an NCP that cannot be shared can be explicitly divided among the VTAMs that own the NCP. You can use the OWNER operand in NCP definition statements to explicitly identify the VTAM that is to own a particular resource. If a resource is allocated to a particular VTAM by the OWNER operand, that resource can be controlled only by operator commands entered from that VTAM. Therefore, VARY ACT and VARY INACT commands can be entered for these resources from the VTAM that owns them, but not from any of the other VTAMs that share the NCP. The OWNER operand can be specified on the PCCU, GROUP, LINE, and PU definition statements.

Backup owner

You can also specify that the NCP resources can be taken over by a backup VTAM by using the BACKUP operand in the NCP definition. The operator at a backup VTAM can enter a VARY ACQ command to acquire resources that belong to another VTAM. The VARY ACQ command can be used to acquire all of the NCP resources, regardless of how the OWNER operand is specified, or can be used selectively to acquire only resources assigned to a particular VTAM through the OWNER operand. You can return the resources by using the VARY REL command.

Dump or IPL of a shared NCP

You can automatically dump or IPL the same NCP from more than one channel-attached VTAM. (You can code YES for AUTODMP and AUTOIPL on multiple VTAMs for the same NCP.) If the NCP is already being IPLed or dumped by one VTAM and it receives a request for an IPL or dump from another VTAM, the second request is rejected and the operator is informed that the second VTAM is waiting for the communications controller to become available. When the contending operation completes, VTAM restarts the requested IPL or dump operation if the state of the communications controller indicates that it is required.

Note: More than one host might take a dump successfully.

This function requires SSP Version 3 Release 7 or a subsequent release.

Chapter 22. Operating with an NCP

This section helps you plan to run and control a domain that includes NCP subareas.

The following list shows you where to find more information about NCP.

- "Loading an NCP"
- "Activating an NCP" on page 643
- "Monitoring an NCP" on page 651
- "Deactivating an NCP" on page 653
- "Deactivating links" on page 653
- "Dumping an NCP" on page 654
- "IBM Communication Controller for Linux on System z considerations" on page 658

Loading an NCP

Note: You cannot load or activate an NCP over a type 2.1 link.

To load NCP into a communication controller from VTAM:

1. Include information on loading NCP in your start procedure:

```
.
.
.
//NCPLOAD      DD      DSN=SYS1.NCPLOAD,DISP=SHR
.
.
.
```

2. Use the VARY ACT command with LOAD=YES specified to load the communication controller.

NCP load process

NCP LOADLIB is the load library containing the NCP load modules and resource resolution tables (RRTs).

When the NCP is loaded, the NCP definition statements, which are stored in VTAMLST, are read into storage on the host to define the resources connected through the communication controller. The NCP resource resolution table (RRT) is loaded into the host, and the NCP load module is loaded into storage on the communication controller. Figure 153 on page 636 illustrates the NCP load process.

© Copyright IBM Corp. 1984, 2011

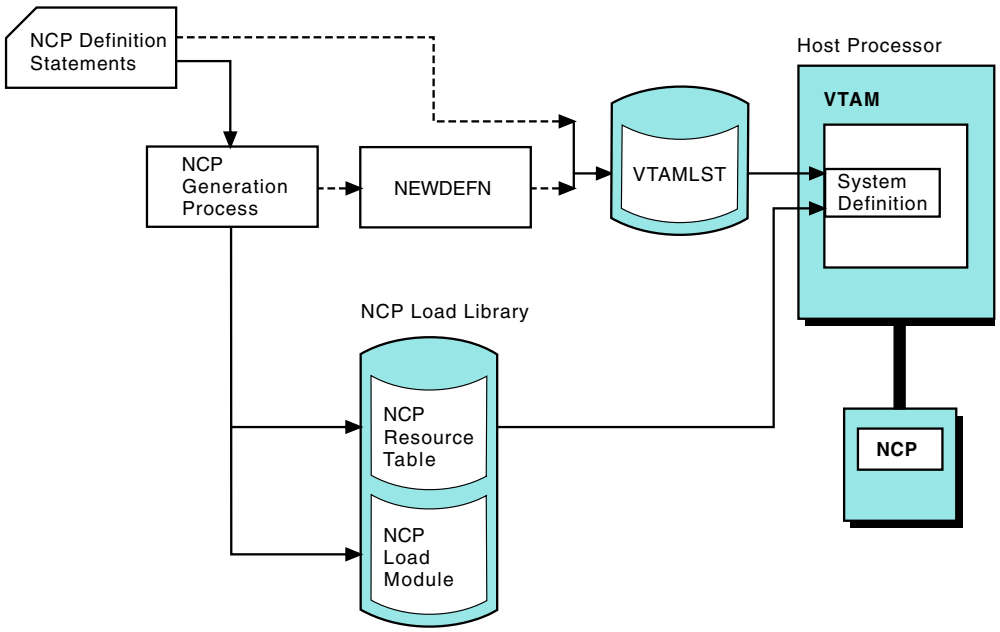


Figure 153. NCP generation

The NCP definition statements are loaded directly into VTAMLST. However, if you generate a NEWDEFN data set, take the contents of the NEWDEFN data set and manually transfer them into VTAMLST. For more information about NCP loading and the NEWDEFN data set, refer to the *NCP, SSP, and EP Generation and Loading Guide*.

Identifying the communication controller

The communication controller into which the NCP is loaded is determined by the adjacent link stations specified for automatic activation with the NCP. These link stations in adjacent nodes locate the communication controller in the network.

If no link station is available, the load fails. In this case, specify RENAME=*adjacent link station name* on the command line.

Within its domain, VTAM ensures that no more than one load or dump operation is attempted at one time for a given NCP through any adjacent link stations that might be available.

If you are activating an NCP in a communication controller that already contains a different NCP, you should first deactivate the current NCP and any active link stations in adjacent subarea nodes that are connected to that NCP. You can then enter the command to activate the new NCP, which causes the communication controller to be reloaded as described in 638 (if LOAD=U is specified or assumed). If adjacent link stations are not deactivated, the activation of the new NCP might fail because one of the following occurred:

- There are no link stations available. (VTAM does not automatically break the existing connections with the old NCP.)
- The names of the old and new NCPs are the same, but their subarea addresses are different.
- The subarea addresses of the old and new NCPs are the same, but their names are different.

Selecting a load station

When loading an NCP, VTAM needs to know the name of the adjacent link station to be used. The load station can be either a channel link station or an SDLC link station within an NCP. VTAM can select a load station, or you can specify a load station. You specify the load station on the LOADSTA operand of the PCCU definition statement for the NCP or on the VARY ACT command, or during NCP error recovery if VTAM asks you whether to reload the NCP. VTAM selects the load station whenever there is no load station name currently in effect.

VTAM maintains a list of link stations over which a remote NCP can be contacted, loaded, or dumped. This list is maintained whether the NCP is active or inactive. This list of link stations is organized in a last-in-first-out (LIFO) order.

When activating the NCP, VTAM updates this list with any new link stations that are found in the NCP definitions or the VARY ACT command. VTAM searches this list for a suitable link to contact or load and then selects a link station by activating any link station that is not already active.

VTAM uses the first active link station it finds in the list. If it does not find an active link station by the time it reaches the end of the list, VTAM selects the last link station in the list and waits for it to complete activation before proceeding with the NCP activation.

If you want to override a load station name specified in an NCP definition, use the LOADSTA operand of the VARY ACT command to specify the name of another link station. You can also use this override capability to specify that VTAM should select the load station, despite the NCP definition. Do this by specifying a null value for the load station (that is, specifying the LOADSTA operand without a value). If you choose not to override the NCP definition value, VTAM uses that definition value to determine the link station to use.

When selecting a load station, the following restrictions apply:

- The load station must be specified for automatic activation with the NCP, or must be already active when it is needed (with VTAM thus knowing of its connection to the NCP). The former method is recommended.
- The link station must not be connected to a communication controller other than the one you want to load. Otherwise, you might load the wrong communication controller.
- The reconnection of cross-subarea links from one communication controller to another might require the changing of load station specifications for those NCPs intended to run in the communication controllers.
- Avoid the WARM start option and the WARM operand of the VARY ACT command when activating the affected NCPs if you change the load station specifications.
- Improperly specified LOADSTA values for two or more NCPs can result in a deadlock during their activation, with the activation of each NCP load station waiting for the adjacent NCP activation, which in turn can be waiting for the availability of a load station.
- You must define any link to be used as a load station link to both the NCP and the communication controller as being capable of initial program load (IPL):
  - For the NCP, specify IPL=YES on the LINE definition statement in the NCP definition.

– For the communication controller, indicate the ability of a link to accept an IPL in the configuration file (CDF) residing in the communication controller diskette storage.

Loading from the host or from external storage

For IBM 3720 and 3745 communication controllers, the VARY ACT command enables you to control whether the NCP is loaded from the host or from the 3720 or 3745 Communication Controller external disk storage.

The first NCP load can be from the host or a diskette. For more information on loading an NCP from a diskette, refer to the *NCP, SSP, and EP Generation and Loading Guide*. For communication controllers other than 3720s or 3745s, the NCP can be loaded only from the host.

Reloading an NCP

If VTAM is allowed to decide whether to reload the communication controller, one of the following occurs:

- If the communication controller contains no program, VTAM loads the NCP into the communication controller.
- If VTAM decides that a load is not necessary, VTAM issues a message to inform you, and it performs no disk-related functions.
- If the communication controller already contains the NCP being activated, you might be asked whether the existing NCP should be refreshed with a new copy. If you specified AUTOSYN=NO on the PCCU definition statement in the NCP definition, VTAM does not automatically resynchronize itself with the existing NCP, but rather asks you whether resynchronization or reloading should occur. Otherwise, VTAM resynchronizes itself with the NCP without operator intervention and without reloading the communication controller.
- If the communication controller contains an NCP other than the one being activated, you might be asked whether the existing NCP should be replaced. If you specified VFYLM=YES on the PCCU statement in the NCP definition, you are asked to confirm that the communication controller should be reloaded. Otherwise, VTAM proceeds with the reload without operator intervention.
- If the communication controller contains a program other than an NCP (for example, an EP program in a channel-attached communication controller), the NCP activation fails. During activation VTAM attempts channel contact, but when the identification exchange fails, VTAM assumes that the communication controller might contain an EP program. VTAM then attempts to resynchronize with the assumed NCP. Activation of the channel link station fails when the resynchronization fails. This results in failure of the NCP activation itself. Another VARY ACT command specifying LOAD=YES can be entered to force loading of the NCP.

Scheduled automatic reloading of a communication controller

To automatically IPL your IBM 3745 Communication Controllers at a scheduled date and time, perform the following steps:

1. Set the date and time associated with the specific load module.
2. Maintenance operator subsystem (MOSS) sends an alert before starting the IPL (default is 60 minutes). The alert is not displayed by VTAM. The alert is forwarded to the NetView program and can be viewed using the NetView hardware monitor. You can have the NetView program send a message to the NetView operator when an alert is received by modifying the NetView program automation table. For information on modifying this table, refer to the

NetView for z/OS Automation Guide. For information on the format of the ALERT, refer to the *3745 Guide to Timed IPL and Rename Load Module*.

3. MOSS IPLs the load module from the communication controller disk at the specified time.

For example, if you want to have your communication controller IPLed at 1600 on 24 May 1993, then you would use the command:

```
MODIFY NET,LOAD,ID=ncpa,ACTION=SETTIME,NOTIFY=30,
IPLTIME=(05/24/93,16:00),LOADMOD=ncp1a
```

This command causes MOSS to load the load module *ncp1a* into NCP *ncpa* at 16:00 on 24 May 1993. This command also indicates that MOSS will send out an alert 30 minutes before the scheduled IPL (NOTIFY operand).

Functions of scheduled automatic reload

You can also do the following when implementing this function:

Modify the format of the date
: You can use the DATEFORM start option to modify the format you use to set the date. The default is MM/DD/YY. For the syntax of this start option, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Specify the alert interval
: You can change the length of the interval between the time MOSS sends an alert to the time that MOSS IPLs the communication controller. Use the MODIFY LOAD command with the NOTIFY operand. The default is 60 minutes.

Display a scheduled IPL
: You can display the information about a scheduled IPL using the DISPLAY DISK command. For the syntax of this command, refer to *z/OS Communications Server: SNA Operation*.

Cancel a scheduled IPL
: You can cancel a scheduled IPL using the MODIFY LOAD command with ACTION=SETTIME,IPLTIME=CANCEL specified.

Example of a scheduled automatic reload

Following is an example of a scheduled automatic reload of an NCP:

1. Load and activate the NCP using the options PUNAME=X, NEWNAME=A, and LOADMOD=A.
2. Change the original NCP generation or write a new NCP definition using the options NEWNAME=B and PUNAME=Y.
3. Load the changed or new NCP generation, store it on the hard disk, and set a time for it to IPL.
4. The NCP is reloaded at the specified time. A soft INOP is generated, and VTAM tries to bring up a session with the NCP. Because the PU name has changed, the session state will be pending ACTPU (PAPU2).
5. Even if you reloaded NCP with the same load module name, any new resources defined to NCP in the new NCP generation (which is now loaded in the NCP) will not be known to VTAM. Therefore, any attempts to work with these newly defined resources fail. For VTAM to be aware of the changes, use the VARY INACT command with the FORCE option to bring down the session between VTAM and NCP and then activate the session using the LOAD=NO option.

Note: If your NCP is in pending ACTPU state and you have an active virtual route between VTAM and the NCP through another VTAM, then the ACTPU is sent over this virtual route (rather than directly to the NCP over the original link). NCP responds with an ACTPU response, but the load module name returned is not the one VTAM is expecting. You will receive message IST605I, and the session between NCP and VTAM is deactivated. In this case, reactivate the session with the new load module name and the LOAD=NO option.

Considerations for scheduled automatic reloading

Consider the following issues for automatic reloading:

- You cannot schedule 2 load modules for the same communication controller unit to be loaded within 5 minutes of each other.
- A 3745 Communication Controller can have two central controller units. You cannot load one central controller unit using a scheduled automatic reload while the other central controller unit is waiting to be loaded.
- If MOSS goes down after you have scheduled an automatic reload, one of the following occurs:
  - If MOSS returns online after the time for the scheduled reload, the scheduled reload is canceled. No alert is sent.
  - If MOSS returns online before the time for the scheduled reload and the time to send the alert has passed, MOSS performs the reload as scheduled and an alert is sent if possible.
  - If MOSS returns online before the time to send the alert, MOSS sends the alert and performs the reload as scheduled.
- If you schedule an automatic reload with a load module that contains errors, MOSS attempts to load the load module again before failing the load.
- If you have more than one VTAM that owns a remote 3745 communication controller, consider the following:
  - It is possible for a person at each VTAM to set different scheduled automatic reloads for the same load module. If this happens, the scheduled automatic reload set by one VTAM can be written over by the other VTAM. MOSS performs the reload at the last time set for the load module.
  - If MOSS receives a command to purge a load module that has a scheduled automatic reload, MOSS cancels the scheduled reload and purges the load module. No message is sent to the VTAM that scheduled the reload.
  - If MOSS receives a command to replace a load module that has a scheduled automatic reload, MOSS replaces the load module and cancels the scheduled reload. No message is sent to the VTAM that scheduled the reload.
- AUTODMP and AUTOIPL operands on the PCCU definition statement are always set to YES after a scheduled automatic reload. For more information about these operands, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.
- If the time you specify for the IPL time is the same as the current system time, MOSS does not IPL.
- If you set the IPL time for more than 15 days in advance, you might encounter minor MOSS clock deviations. However, if this is not a problem, then you can set the IPL time up to 90 days in advance.

Renaming NCP load modules

You can change the name MOSS associates with the load module on the communication controller disk to another name. This is useful if you want to load from a disk or keep load module changes transparent to your operators.

Use the MODIFY LOAD command with the NEWNAME operand and ACTION=RENAME specified.

To associate the NCP name with different load module names without using the renaming step:

1. Generate the NCP source programs using different load module names.
2. Activate the NCP and specify the load module by using the LOADMOD operand.
3. Use the name specified in the ID operand, not the load module name, to identify the NCP.

Sample load module name change

Following is a sample scenario of renaming an NCP load module. In this scenario, one load module is being replaced by another. However, the final load module name will be the same as the original load module name.

1. Activate an NCP with load module NCPA and start the NCPA using the VARY ACT command:

   ```
   V NET,ACT,ID=NCPA,LOADMOD=NCPA,LOAD=YES,SAVEMOD=YES
   ```

2. The NCP is activated and a load module NCPA is saved on the MOSS disk because you specified SAVEMOD. Now an NCPA member is in VTAMLST, an NCPA member is in NCPLOAD, and you have a session between VTAM and NCPA.
3. Copy NCPA to a file named NCPA1, add your changes, and generate the file on another system. On the other system, the VTAMLST member is called NCPA1. Because the NEWNAME parameter in the generation was not changed, but other keyword values in the generation were changed, the generation name of the NCPLOAD library member will be NCPA. Use an external MVS job to rename this member to NCPA1.
4. Move NCPA1 (the VTAMLST member) and NCPA1 (the NCPLOAD library member) to the original MVS system. Now you have two load modules:
   - NCPA on the VTAMLST library and NCPA on the NCPLOAD library.
   - NCPA1 on the VTAMLST library and NCPA1 on the NCPLOAD library

   The NCP name has not changed and the NEWNAME parameter in the generation has not changed.
5. Use the MODIFY LOAD command to add NCPA1:

   ```
   F NET,LOAD,ID=NCPA,ACTION=ADD,LOADMOD=NCPA1
   ```

   A nondisruptive ADD is performed.
6. Use the MODIFY LOAD command to rename the current load module on the MOSS disk:

   ```
   F NET,LOAD,ID=NCPA,ACTION=RENAME,LOADMOD=NCPA,NEWNAME=NCPA2
   ```

7. Use the MODIFY LOAD command to rename the load module NCPA1 on the MOSS disk to NCPA:

   ```
   F NET,LOAD,ID=NCPA,LOADMOD=NCPA1,NEWNAME=NCPA,ACTION=RENAME
   ```

8. Use the VARY ACT command to deactivate the session with the NCP:

   ```
   V NET,INACT,ID=NCPA
   ```

9. Rename the member name NCPA on the VTAMLST and the NCPLOAD library to NCPA2 using an external MVS job. Then rename NCPA1 (the VTAMLST member) and NCPA1 (the NCPLOAD library member) to NCPA. You now have the following:
   - NCPA2 member in VTAMLST (MVS job only, not generation)
   - NCPA2 member in NCPLOAD (MVS job only, not generation)
   - NCPA member in VTAMLST
   - NCPA member in the NCPLOAD
10. Use the VARY ACT command to activate the NCP and to load the NCP from the MOSS disk.

    ```
    V NET,ACT,ID=NCPA,LOADMOD=NCPA,LOADFROM=EXT,LOAD=YES
    ```

    Note: Remember, the NEWNAME in this new generation was never changed, so to VTAM the NEWNAME parameter is NCPA and is not different from the name of the load module NCPA (which is also the name of the NCP). Now you have:
    - NCPA2 member in VTAMLST
    - NCPA2 member in NCPLOAD
    - NCPA member in VTAMLST
    - NCPA member in the NCPLOAD

Considerations for changing load module names

If you only rename a load module on the MOSS disk and do not change the VTAMLST and NCPLOAD library members, the following problems might occur:

- The load module does not activate if the name is changed only on the MOSS disk.

  If you code the following, the activation of the load module will not occur:

  ```
  F NET,LOAD,ID=NCPA,LOADMOD=NCPA,NEWNAME=NCPA2,ACTION=RENAME
  V NET,ACT,ID=NCPA,LOADMOD=NCPA2,LOADFROM=EXT,LOAD=YES
  ```

  VTAMLST or NCPLOAD library members have not been changed.

  Note: VTAM searches the VTAMLST file for load module NCPA2. If VTAM cannot find it, VTAM rejects the command. VTAM does not activate NCPA, even if the VTAMLST and NCPLOAD library member names have changed to NCPA2, because the NEWNAME parameter in the VTAMLST will not have a matching NCP generation on the NCPLOAD library. Remember, you changed only the external member names; you did not change the NEWNAME parameter and generate the load module.
- The renamed load module cannot be replaced if it is renamed only on the MOSS disk. The following procedure will not work:

  ```
  F NET,LOAD,ID=NCPA,LOADMOD=NCPA,NEWNAME=NCPA2,ACTION=RENAME
  F NET,LOAD,ID=NCPA,ACTION=REPLACE,LOADMOD=NCPA2.
  ```

  VTAMLST and NCPLOAD library members were never changed.

  Note: VTAM processes the MODIFY LOAD command and tries to load NCPA2. VTAM looks for the NCPA2. Because VTAM does not find the NCPA2, VTAM fails the load. If the NCPA (the VTAMLST member) and NCPA (the NCPLOAD library member) are changed externally to NCPA2, the replace will occur.

Functions for changing load module names

You can do the following even if a load module is renamed on the MOSS disk. The assumption is that a load module has already been renamed. In the following cases, a load module NCPA has been renamed to NCPA2:

- Purge a renamed load module on the MOSS disk.

  ```
  F NET,LOAD,ID=NCPA,LOADMOD=NCPA2,ACTION=PURGE
  ```

  VTAM does not perform any checking for the VTAMLST library member NCPA2.
- Set an IPL time so that the load module can IPL at a certain desired time.

  ```
  F NET,LOAD,ID=NCPA,LOADMOD=NCPA2,ACTION=S,IPLTIME=(4/28,11:30)
  ```

  VTAM does not check its VTAMLST members and the NCP load library members. MOSS IPLs NCP at the required time.
- Rename a previously renamed load module.

  ```
  F NET,LOAD,ID=NCPA,ACTION=RENAME,LOADMOD=NCPA#,NEWNAME=NCPA
  ```

  VTAM does not check for the load module, and the load module is renamed.

Loading or purging a load module on a communication controller disk

You can use the MODIFY LOAD command to explicitly load a load module from the host into the 3720 or 3745 Communication Controller external disk storage while an NCP is already active in the communication controller.

At activation time, you can use the LOADMOD operand of the VARY ACT command to specify the load module to be loaded into the communication controller. After activation, the MODIFY LOAD command allows you to add, replace, or purge load modules on the 3720 or 3745 Communication Controller disk.

You can add a load module to the communication controller external storage by using the ADD operand of the MODIFY LOAD command. If the load module to be added exists on the communication controller disk or if the maximum number of load modules allowed is already on the disk, the command fails.

You can replace a load module already on the communication controller disk by using the REPLACE operand of the MODIFY LOAD command. VTAM then replaces the load module specified on the communication controller disk. If the load module is not present on the disk, an ADD operation is attempted.

The REPLACE operand affects the automatic activation of a load module if an automatic re-IPL is required. If the load module being replaced is the load module tagged for automatic re-IPL, then after the REPLACE operation starts, the load module is no longer tagged to be automatically activated when an automatic re-IPL is initiated. If an automatic re-IPL occurs while a REPLACE operation is running, the communication controller is not activated with any load module.

The CANCEL option of the MODIFY LOAD command can be used to stop an ADD or REPLACE operation that is in progress. If the CANCEL operand is used, the partially stored load module is purged from the disk.

Activating an NCP

Although there is no resource hierarchy among major nodes, there are some considerations for NCP activation.

There must be at least one defined route to the NCP; this requires that an appropriate path definition set be activated before the NCP is activated. NCP activation remains pending until a route to the NCP becomes physically available.

If the member name of the NCP source program differs from the name that you want to be associated with the NCP after activation, use the LOADMOD operand on the VARY ACT command to identify the member name in both VTAMLST and in the specified LOADLIB. The member name must be the same as the load module name.

Automatic activation of adjacent link stations

When activating an NCP, VTAM can be provided with a list of link stations in subarea nodes adjacent to the NCP that are to be automatically activated along with the NCP. Automatically activated link stations and already active link stations adjacent to the NCP connect the NCP to the rest of the network. In addition to providing access to the NCP being activated, activating the link stations provides link station ownership to the VTAM SSCP. The NCP containing the link stations supplies VTAM with notification of link station failures. Activation of the link stations enables you to load or dump the NCP through the link stations.

A VARY ACT command for an NCP major node can result in the automatic activation of cross-subarea links and link stations in adjacent subarea nodes (other NCPs or VTAM itself). For example, if you enter a command to activate an NCP, VTAM automatically activates the link stations in adjacent nodes that are known to be needed for the connection to the NCP being activated. VTAM knows this from the channel device name or RNAME specification (or both) of either the NCP definition statement or the VARY ACT command. However, the full route must be available for automatic activation to be completely successful.

Specifying adjacent link stations

You can specify the link stations that are to be automatically activated using the CUADDR and RNAME operands on the PCCU definition statement for the NCP. (This method is recommended to simplify operating procedures and reduce the chance for operator error.) If you do not specify these link stations, or if you need to override the defined link station names, you can use the U or RNAME operands (or both) on the VARY ACT command when activating the NCP.

When the NCP is activated, the specified link stations and their associated links are automatically activated if they are not already active. If any of these link stations are not yet known to VTAM (because the NCP to which they are subordinate has not been activated), VTAM remembers them until they become known (through the activation of their major node). Then VTAM completes the activation. VTAM informs you of the link stations that are awaiting the activation of their NCP.

When activating a link in a channel-attachment major node, the operator needs to supply the channel device name of the link unless it is supplied in the LINE definition statement. The operator can also specify the channel device name to override the value in the LINE definition. To supply the channel device name, use the U operand of the VARY ACT command when activating the link.

If VTAM needs the operator's assistance to create a connection for a link in a channel-attachment major node, VTAM displays a message indicating that a connection request is denied because the link is not physically attached or is unknown to the operating system. In either case, the link must be available before the channel-attachment major node can be activated.

Example of automatic activation of adjacent link stations

Figure 154 shows an example of the automatic activation process. Assume that LSA is automatically activated when you activate NCP1. If you want to activate NCP2 through NCP1, first either manually activate LSB and LINKA or have them automatically activated (coded using RNAME=LSB).

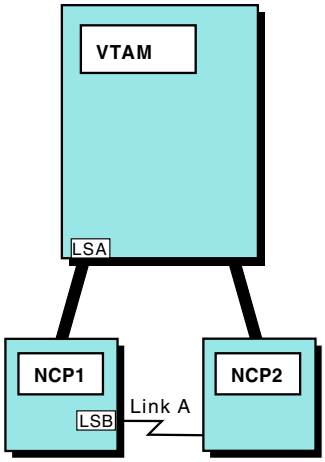


Figure 154. Activating NCPs and link stations: Through another NCP

Going in the other direction, if you want to activate NCP2, LSC must be active or automatically activated. If you want to activate NCP1 through NCP2, you must manually or automatically activate LSD and LINKA. Figure 155 illustrates this concept.

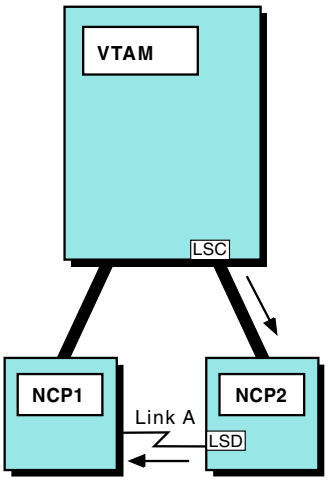


Figure 155. Activating NCPs and link stations: Through host link station

If NCP1 was inactive when you tried to activate NCP2, the activation of LINKA and LSB would be delayed until NCP1 was activated. VTAM would communicate with NCP2 through its channel connection (LSC).

Figure 156 on page 646 shows an example of automatic activation of link stations.

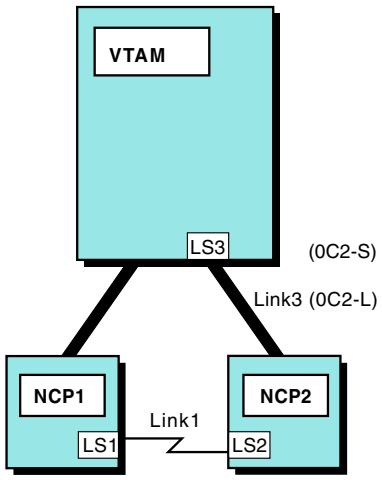


Figure 156. Automatic activation of link stations

For Figure 156, the following is coded for NCP2:

```
CUADDR=0C2,
RNAME=LS1,
```

- NCP1 is active. LINK1, LS1, and NCP2 are not.
- Activate NCP2.
- VTAM automatically activates LS3 (0C2-S) and LINK3 (0C2-L), enabling communication over LINK3. VTAM also automatically activates LS1 and LINK1, enabling communication through NCP1 over LINK1.

Following is an example of delayed automatic activation:

- In Figure 156, NCP1, LINK1, LINK3, and NCP2 are inactive.
- Activate NCP2.
- VTAM automatically activates LS3 (0C2-S) and LINK3 (0C2-L), enabling communication over LINK3. VTAM delays automatic activation of LS1 and LINK1 until you activate NCP1.

Delay of automatic activation

If VTAM must delay the activation of all the link stations specified for automatic activation, and if there are no other active connections to the NCP, the NCP activation itself is delayed to await the availability of an active connection. You should activate the adjacent NCPs containing the link stations to allow the original NCP activation to complete. Similarly, if none of the routes to the NCP are physically available, VTAM displays a message to you and delays the activation of the NCP until at least one route becomes available. However, if a request is entered to activate an NCP to which no route has been defined, the activation fails.

Any suspended activation of an NCP or link station can be terminated by deactivating that NCP or link station.

Updating adjacent link stations

If the physical links to a communication controller running NCP are reconnected to a different communication controller running another NCP, the link stations specified for automatic activation with the NCPs might need to be updated to reflect the new connections. That is, the CUADDR and RNAME values on the PCCU statement in the NCP definitions might need to be updated. Also, the WARM start option and the WARM operand of the VARY ACT command should

be avoided when activating NCPs if the link station specifications are changed. Failure to update the adjacent link station specifications might result in the activation or loading of the wrong communication controller when an NCP is activated.

When VTAM automatically activates an adjacent link station during activation of an NCP, VTAM expects to find that the link station is connected to a communication controller that either requires loading of an NCP or contains the NCP being activated. If, through an operator or system programmer error, the list of link stations to be automatically activated includes one or more link stations that are connected to the wrong communication controller, VTAM might not detect the error and could assume that this wrong communication controller is to be reloaded with the NCP being activated. Therefore, you should ensure that link stations are correctly chosen for automatic activation with an NCP. Ensure that the link stations are in fact connected to the communication controller that does contain that NCP.

Reactivating a failed NCP

If an NCP fails or is manually reloaded immediately before or during the activation of an NCP by VTAM, the activation might wait in a pending ACTPU state (PAPU1 or PAPU2). The adjacent link station must wait through its failure timeout period (specified on the ACTIVTO operand of the NCP GROUP definition statement) before it recognizes that its adjacent NCP has failed and that the link station is now inoperative. This timer value defaults to its maximum value of 7 minutes in the NCP generation process, so it can be up to 7 minutes before VTAM becomes aware that an SDLC-link-attached NCP has failed. VTAM bases its activation processing for the NCP during this period on the assumption that the NCP and its apparently active adjacent link stations are functioning normally.

You can recover by deactivating and reactivating the failed NCP. If it is certain that the NCP has failed and should be reloaded, you should specify LOAD=YES on the VARY ACT command for the NCP. If you do not specify LOAD=YES, any adjacent link stations in surrounding NCPs should also be deactivated prior to the reactivation of the NCP. Then you can avoid the possibility of the timeout described previously. These link stations can then be reactivated automatically as part of the reactivation of the NCP, or they can be reactivated directly after the NCP has been successfully reloaded.

Activating links

An SDLC, LAN, or X.25 link between two communication controllers can be activated directly, indirectly, or automatically.

How a link station is activated

Each cross-subarea link station is uniquely activated in the following ways:

- For channel link stations adjacent to an NCP, there are two methods of activation:
  - If a link station has been defined and named by VTAM, it can be activated only automatically. Any VARY ACT command specifying a channel link station is rejected. However, these channel link stations can be deactivated directly, indirectly (by halting VTAM), or automatically.
  - If the link station has been defined and named by the user (with a PU definition statement), it can be activated by the VARY ACT command.

- Channel link stations adjacent to another host processor can be activated only directly or indirectly, and can be deactivated with VARY commands. These channel link stations are named in the definition for the channel-attachment major node.
- SDLC, LAN, and X.25 link stations are activated directly or indirectly with the VARY ACT command.

When a cross-subarea link station becomes active, VTAM issues a message to the operator indicating the name and subarea number of the adjacent subarea node with which contact has been established.

Owning link stations

Activation of the link station makes the link station active to VTAM and makes VTAM SSCP an owner of that link station. For cross-subarea SDLC, LAN, and X.25 link stations, more than one SSCP can share ownership of an NCP link station if the link station is activated by each SSCP. However, for cross-subarea SDLC, LAN, and X.25 link stations within a host processor node, only one SSCP can own and activate the link station.

An NCP link station can also be active without SSCP ownership if its link has been defined to use the SDLC monitor mode function in the NCP. As a result, a cross-subarea SDLC link station within an NCP can be capable of supporting session traffic without having been activated by VTAM.

Delayed activation of logical lines

This function enables you to perform an orderly activation of a physical resource and its associated logical resources. Because there is a hierarchical relationship between the physical resources (line and physical unit) and the logical lines, you can use this function to implicitly activate a logical line when the physical resources are explicitly activated.

The function creates a hierarchy between a logical line in a group (within an NCP major node) that has a PHYSRSC operand coded and the physical resource indicated by the PHYSRSC operand. This includes:

- NCP token-ring interconnect (NTRI) physical resources (line and physical unit) and logical lines for INN link stations
- NCP Packet Switched Interface (NPSI) physical resources (line and physical unit) and logical lines
- Frame relay nonswitched and switched physical resources (line and physical unit) and logical lines

Lines within any line group for which the PHYSRSC operand is coded are logical lines. Lines within a line group for which PHYSRSC points to a valid physical unit within the same NCP major node are hierarchically subordinate to the physical unit. Not all logical lines are hierarchically subordinate to a physical resource. Because an NTRI switched peripheral line can be used serially for connections through different token-ring interface couplers (TICs), lines within a line group where the PHYSRSC, ECLTYPE, and DIAL=YES operands are coded are not subordinate to the physical unit. In this case, VTAM does not create the hierarchy between the physical resources and those logical lines. PHYSRSC must point to a physical unit to create the subordinate logical line relationship.

Activating logical lines hierarchically

A subordinate logical line can be activated only after its higher-level physical unit has been activated. The SCOPE operand on the VARY ACT command works with this hierarchy like it does with other hierarchical

relationships in VTAM. For example, SCOPE=ALL on the higher-level physical unit causes the subordinate logical lines and their subordinate resources to be activated. SCOPE=U causes only the logical lines with ISTATUS=ACTIVE (and any subordinate resources with ISTATUS=ACTIVE) to be activated. SCOPE=COMP and SCOPE=ONLY also work with this hierarchy.

Deactivating logical lines hierarchically
When a logical line is subordinate to the physical resource, deactivation of the physical resource deactivates the logical lines.

Loading and activating an NCP over a switched link

When a remote NCP is activated using the VARY ACT operator command, you can specify the switched link station name on the RNAME operand of the PCCU definition statement or on the RNAME operand of the VARY ACT operator command to have VTAM automatically perform the dial operation and establish the switched connection. The following examples show the two ways to specify the RNAME operand to load and activate the remote NCP2 in Figure 157.

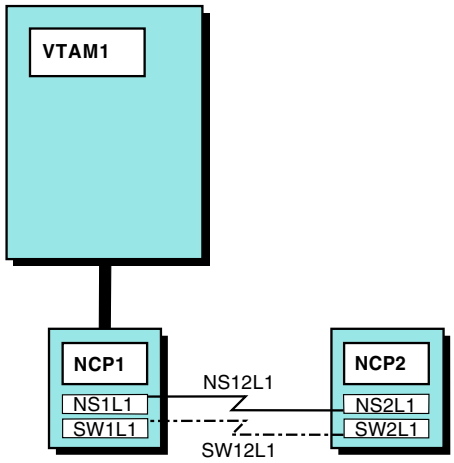


Figure 157. Loading and activating NCP using switched link

On the PCCU definition statement for NCP2

```
PCCU  SUBAREA=1,
⋮
      LOADSTA=NS1L1,
      RNAME=(NS1L1,SW1L1),
⋮
```

On the VARY ACT operator command

```
VARY ACT,ID=NCP2,LOAD=YES,RNAME=(NS1L1,SW1L1)
```

The load can be performed over either of the two possible link stations (NS1L1 or SW1L1) in Figure 157. Because both link station names are specified on the RNAME operand, VTAM does the following:

- Activates the nonswitched link
- Performs the dial operation and establishes the switched connection

The VTAM default load station preference is:

1. Active channel link
2. Active nonswitched link
3. Active dial-out switched link

4. Active dial-in switched link
5. Pending active nonswitched link
6. Pending active dial-out switched link
7. Pending active dial-in switched link

Therefore, if the nonswitched link is operative, the load is performed using the nonswitched link.

If you omit the switched load station (SW1L1) from the RNAME operand, the switched link connection is not established, and only the nonswitched connection is used. If a failure occurs on the nonswitched link, the operator can issue a subsequent VARY ACT operator command with the RNAME operand specifying the switched load station name (SW1L1).

If the switched link is not used for the initial activation and load process, you can activate it at any time using a VARY ACT operator command.

Note: When you are loading an NCP over switched lines and the NCP has channel-attached or leased line connections to another host, the NCP load is not complete until the host that initiated the command receives an indication that the command has completed.

Enabling communication controller channels for activation

The channel-enable switches on a communication controller panel (or the MOSS console for the 3720 and 3745) determine whether VTAM can activate a channel link and link station during an NCP activation. For a communication controller that is both channel-attached and SDLC-link-attached, these switches also determine whether an NCP can be loaded over a channel or an SDLC link. If any channel adapters are enabled on the communication controller, the remote program load (RPL) feature of the communication controller is disabled, and the communication controller can be loaded only over one of these channels. If none of the channel adapters are enabled, the communication controller can be loaded only over SDLC links defined as being capable of IPL (previously described under “Selecting a load station” on page 637).

All channel adapters should be kept enabled during normal operation with channel-attached NCPs. This is because a channel-link station activation fails (with the channel appearing to be “not operational”) if the channel adapter for the channel VTAM is attempting to use is disabled. VTAM does not recover the use of the channel (even if the channel adapter is later enabled) unless the NCP is deactivated and reactivated. If all channel adapters are enabled, the SDLC links are still available for normal use after the NCP is loaded or if the NCP has been previously loaded. The channel-enable switches should be turned off only when a communication controller must be loaded over an SDLC link (such as when the channel-attached host has failed along with the NCP).

Figure 158 on page 651 shows an example of an NCP that is both channel-attached and SDLC-link-attached to a VTAM host. For this example, assume that NCP1 is active and that NCP2 is inactive and requires loading of its NCP load module.

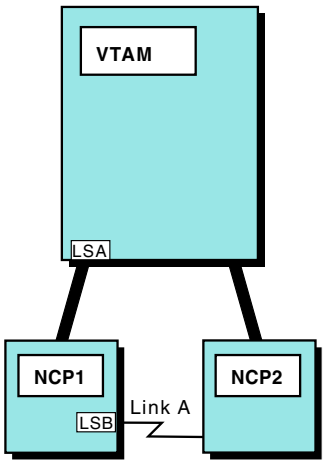


Figure 158. Loading over a channel

You activate NCP2 with the following command (with NCP2 channel adapter enabled on the communication controller panel):

```
VARY NET,ACT,ID=NCP2,U=0C2,RNAME=LINKSTA1
```

VTAM then attempts to activate the adjacent link stations 0C2-S and LINKSTA1. Activation of LINKSTA1 in NCP1 is delayed because NCP1 receives no response to link-level commands sent on the SDLC link to NCP2 communication controller before and during the loading of NCP2 over the channel. Because the channel adapter is enabled, activation of 0C2-S proceeds, and VTAM begins the loading of NCP2 over the channel. After the load completes and LINKSTA2 is activated, contact is established with NCP1, and the activation of LINKSTA1 completes.

Monitoring an NCP

Monitoring your NCPs includes checking for hardware errors in devices attached to your NCPs, checking the contents of external disk storage, and testing links.

Sending line errors to VTAM

The NCP maintains statistical information regarding temporary line errors and other hardware error conditions for links and PUs. Normally the NCP sends this statistical information to VTAM only when a statistical counter overflows or a permanent error occurs. With intensive mode recording (IMR), error information is sent to VTAM as the errors occur. You can use the MODIFY IMR command to start IMR. With the MODIFY IMR command, you can request intensive mode recording for a communication link between an NCP and one of its peripheral physical units, or between two NCP major nodes.

The error information is sent to VTAM in a miscellaneous data recorder (MDR) record. MDR records are written to the system error log (LOGREC). If the NetView hardware monitor is in your system, VTAM can send copies of the MDR records to them.

To begin intensive mode recording on a link between an NCP and a peripheral physical unit, specify the name of the physical unit on the ID operand of the MODIFY IMR command. The SDLC link must be active.

To begin intensive mode recording on a link between two NCPs, specify the name of the link station in the NCP initiating IMR on the ID operand of the MODIFY IMR command. The SDLC link must be active.

You can also use the MODIFY IMR command to specify the number of intensive mode recording events to be recorded before intensive mode recording is automatically terminated. You can cancel intensive mode recording before this limit is reached.

Displaying NCP storage

The DISPLAY DISK command displays the following information about the contents of IBM 3720 and 3745 Communication Controller external disk storage:

- Information about each load module, including:
  - The load module name
  - The date and time the load module was stored on the disk
  - Stability of the load module (stored, storing, or suspended)
  - An indication of whether the load module on the disk is the same as the load module currently active in the communication controller
- Information regarding dumps, including:
  - The dump name
  - The date and time the dump was written to the disk
- An indication of whether the load module is tagged for automatic IPL.

The DISPLAY NCPSTOR command displays:

- The storage contents of an NCP communication controller
- The storage contents of an NCP communication controller external disk storage (valid only for IBM 3720 Communication Controllers with the communication controller external disk storage feature or for the IBM 3745 Communication Controller)
- The state vector from the NCP communication controller external disk storage (valid only for IBM 3720 Communication Controllers with the communication controller external disk storage feature or for the IBM 3745 Communication Controller).

The DISPLAY NCPSTOR command displays up to 256 bytes, beginning at any address within the communication controller or communication controller external disk storage. The bytes are displayed on the VTAM operator's console.

Testing a nonswitched SDLC line

A link level 2 test is used to test a nonswitched SDLC communication line between one of the following pairs of items:

- NCP and one of its peripheral PUs
- VTAM and one of its peripheral PUs
- Two NCPs
- Two VTAMs (The command must be initiated from the VTAM with the highest subarea number.)
- NCP and VTAM (The command must be issued from the NCP domain.)

You can specify:

- A continuous link test or the number of test transmissions to be sent
- The cancellation of an ongoing link test
- Any user data to be included in the test transmissions

The test data is returned from the remote station (NCP, host node, or peripheral PU) to the controlling NCP or to VTAM. The data received by the NCP or VTAM is compared with the data sent, and the following information is displayed by VTAM:

- Number of test messages sent
- Total number of test messages returned (including those in error)
- Number of test messages returned without error

The SDLC link must be active, and the physical unit must be inactive. You must specify the name of the physical unit on the ID operand of the MODIFY LL2 command. You can run this test on a point-to-point line or on a multipoint line without affecting any other physical units on the line.

Before you request a link level 2 test between two NCPs, ensure that the link station at the originating end is inactive and the link station at the receiving end is active. In a multiple-domain network with shared host ownership of link stations, both link stations must be inactive to all SSCPs. The name of the link station in the NCP initiating the test must be specified on the ID operand of the MODIFY LL2 command.

Deactivating an NCP

Because there is no resource hierarchy among major nodes, any NCP major node can be deactivated at any time. The deactivation of an NCP does not cause VTAM to deactivate any other NCP. However, there are physical dependencies you must consider. When an NCP is deactivated, all paths using the NCP are also deactivated. Therefore, any NCPs that are link-attached to the NCP being deactivated are rendered inoperative. In addition, deactivation of an NCP or a cross-subarea link or link station can have significant and disruptive effects on other subarea nodes.

To avoid disrupting sessions, you need to be concerned about the status of cross-subarea links and link stations after an NCP deactivation.

Same-domain links are governed by the following rule: when you enter a VARY INACT command for an NCP, VTAM automatically deactivates any link or link station in an adjacent subarea node that was automatically activated when the NCP was activated. VTAM does not perform these automatic deactivations if you activate links and link stations directly or indirectly.

Deactivating links

VTAM deactivates a cross-subarea link station adjacent to a host processor when the channel-attachment major node or the associated channel, SDLC, LAN, or X.25 link is deactivated (even if the associated link is currently carrying session traffic). Deactivating such a link station drops any sessions using a route going through that link station, possibly including the SSCP session with another subarea node. Also, VTAM might be an intermediate routing node for cross-domain sessions that pass through the link station. Care should be taken not to deactivate a cross-subarea link station that could be supporting session traffic.

The NCP deactivates an SDLC, LAN, or X.25 cross-subarea link and its subordinate link station when the last of the link SSCP owners deactivates the link. When the last active link station in a transmission group is deactivated, any session traffic being carried by that transmission group is disrupted. Therefore, if the VTAM operator has not activated a cross-subarea link and link station, and VTAM is

using that link transmission group to carry session traffic, those sessions are disrupted when the link owners all deactivate the link and link station and there are no other active link stations in the transmission group. On the other hand, if the VTAM operator activates the link and link station, VTAM shares the ownership of them, and the link and link station remain active, regardless of the deactivation requests of other domain operators.

If the VARY INACT command is used to deactivate a channel link owned only by VTAM, and the link station of that link is the only active link station in a transmission group, deactivating that link destroys any sessions using a route containing that transmission group, possibly including the SSCP session with another SSCP or with the NCP containing the link being deactivated. For this reason, care should be taken not to deactivate such a cross-subarea link.

Note: To prevent the automatic deactivation of cross-subarea NCP SDLC, LAN, or X.25 links in adjacent subarea nodes when an NCP is deactivated, you can directly or indirectly activate the links before the NCP is deactivated. The operator uses individual VARY ACT commands or VARY ACT with the SCOPE operand; the system programmer uses the initial status specification in the link definitions.

Deactivation of a cross-domain link also causes the deactivation of its subordinate link station.

Dumping an NCP

You can request either a dynamic dump or a static (destructive) dump of an NCP.

Dynamic dump
When you request a dynamic dump (MODIFY DUMP OPTION=DYNA), NCP processing continues while the NCP contents are being dumped. A dynamic dump reflects the storage contents of the communication controller over a period of time.

Static dump
You request a static dump by entering the MODIFY DUMP command with OPTION=STATIC (the default value), or by replying "yes" when VTAM offers to dump a failed NCP. When you request a static dump, NCP processing stops and the current storage contents of the NCP are dumped. This dump effectively destroys a running NCP. The NCP must be reactivated (and reloaded) before communication with that NCP can resume.

The method for obtaining a static dump of an NCP depends upon whether the NCP is active or inactive.

Active NCP
If the NCP is active, specify the name of the NCP to be dumped. Select the dump station and dump file to use as described under "Selecting a dump station for a static dump of an NCP" on page 655 and "Selecting a dump file for an NCP dump" on page 656. As a result of the dump, the NCP is deactivated.

Inactive NCP
If the NCP is inactive, specify the name of an SDLC link station within an NCP adjacent to the communication controller containing the NCP to be dumped.

The link station specified on the command is the dump station. This link station can be either active or inactive, but the associated SDLC link must be active so that it can be used to carry the dump traffic. As a result of the dump, the link station is deactivated, if it is not already inactive. Because there is no dump file associated with a link station, specify the logical unit name (SYSxxx) on the DUMPDS operand on the MODIFY DUMP command.

Although this method can also dump an active link-attached NCP, it is primarily used for dumping an inactive link-attached NCP. If an active NCP is dumped by specifying ID=*adjacent link station* on the MODIFY DUMP command, the adjacent link station, not the NCP, is deactivated when the dump is completed. To deactivate the NCP, issue the VARY INACT command.

You can also obtain a static dump by using the NCP independent dump utility if the NCP is in a channel-attached communication controller. For more information on the NCP independent dump utility, refer to the *NCP, SSP, and EP Diagnosis Guide*.

When you request a static dump by specifying ACTION=STORE, OPTION=STATIC on the MODIFY DUMP command, the NCP terminates. The contents of the NCP are dumped to the controller hard disk only if the automatic dump/load was previously set (`DUMPLOAD=YES,SAVEMODE=YES,LOADFROM=HOST` or `DUMPLOAD=YES,LOADFROM=EXT`) and the dump slot on the MOSS controller hard disk is empty. The load module stored on the controller hard disk is reloaded, but you need to reactivate the NCP before you can resume communication with that NCP. To transfer an NCP dump stored on a controller hard disk to the host, use the ACTION=TRANSFER operand on the MODIFY DUMP command.

Use the DISPLAY NCPSTOR command to display the contents of the NCP external disk dump file. To print the contents of the NCP dump file, use the NCP independent dump utility described in the *NCP, SSP, and EP Diagnosis Guide*.

If an expiration date exists for the dump file and it has not yet been reached, do not respond to message IEC507D by entering REPLY *xx*,M. Enter REPLY *xx*,U to override the expiration date and allow write access to the file. Entering REPLY *xx*,M causes the dump to be abnormally terminated by the operating system.

Selecting a dump station for a static dump of an NCP

When you take a static dump of an NCP, VTAM requires the name of the adjacent link station through which the dump operation is to be carried out. If you specify an SDLC link station name on the ID operand of the MODIFY DUMP command, VTAM uses that link station as the dump station. If you specify an NCP name, a choice of dump stations is possible. The following paragraphs describe the possible ways in which a dump station can be selected when the MODIFY DUMP command specifies an active NCP.

The dump station can be either a channel link station or an SDLC link station within an NCP. VTAM can select a dump station, or you can specify a dump station. Specify the dump station (DUMPSTA) in the PCCU definition statement, on the VARY ACT command, on the MODIFY DUMP command, or during NCP error recovery if VTAM asks you whether the NCP should be dumped. VTAM selects the dump station whenever there is no dump station name currently in

effect from one of the above sources. VTAM selects a channel link station if one is available. If no channel link station is available, VTAM uses any available SDLC link station.

If you want to override a dump station name specified in the PCCU definition statement, you can use the DUMPSTA operand of the VARY ACT command to specify the name of another link station. This override capability can also be used to specify that VTAM should select the dump station. This is done by specifying a null value for the dump station (that is, specifying DUMPSTA without a value). If you choose not to override the PCCU definition statement value, VTAM uses that definition value to determine the link station to be used. The DUMPSTA operand of the MODIFY DUMP command can likewise be used to override a dump station name specified in the PCCU definition statement or on the VARY ACT command.

A link station chosen as a dump station must also be specified for automatic activation with the NCP, or must be already active when it is needed (with VTAM thus knowing of its connection to the NCP). The former method is recommended. If a specified dump station is not available for a dump operation resulting from a MODIFY DUMP command, the command fails. If a specified dump station is not available for a dump operation resulting from error recovery procedures for an NCP, VTAM either prompts you for a link station name or selects a link station from among those that are available.

As with the link stations specified for automatic activation with an NCP, the reconnection of cross-subarea links from one communication controller to another might require the changing of dump station specifications for those NCPs intended to run in the communication controllers.

Switched, X.25, and token-ring link stations are valid dump stations when you specify OPTION=STATIC,ACTION=STORE on the MODIFY DUMP command.

If you change the dump station specifications, avoid the WARM start option and the WARM operand of the VARY ACT command when you activate the affected NCPs.

Selecting a dump file for an NCP dump

When you take a dynamic or static dump of an NCP, VTAM requires the name of the file in the host that is to receive the dump. You can specify the dump file in the NCP definition with the DUMPDS operand (on the PCCU statement), or you can specify it on the MODIFY DUMP command with the DUMPDS operand. The value you specify on the DUMPDS operand should be a logical unit name (SYSxxx). Use the symbolic device name that is coded on the EXTENT or ASSIGN statement for the DLBL statement that contains the file name for the dump data set. The command overrides the specification in the PCCU definition statement, or specifies a file if one was not specified.

Selecting the disposition of an NCP dump

Dynamic or static NCP dumps can be stored on the host or, for IBM 3720 or 3745 Communication Controllers, on the communication controller external disk storage. You can specify where the dump is to be stored (in the event of NCP module error) by using the ACTION operand on the MODIFY DUMP command. The command overrides the DUMPDS operand of the PCCU definition statement.

Other communication controller dumps

In addition to an NCP dump, you can obtain two other dumps for an IBM 372x or 3745 Communication Controller:

- A maintenance operator subsystem (MOSS) dump
- A communication scanner processor (CSP) dump

Unlike the NCP dump, these dumps are not requested by you. If a controller MOSS fails, the MOSS is automatically dumped to the controller MOSS disk (for the 3720 and 3745 Communication Controllers) or diskette (for the 3725 Communication Controller). Likewise, if the controller CSP fails, the CSP is automatically dumped to the controller hard disk. When either a MOSS or CSP dump is taken, a notification is sent to VTAM, alerting you of the failure.

You issue the MODIFY DUMP command to move the contents of the dump to a file in the host processor, where it can be formatted and printed. In addition, the MODIFY DUMP command lets you purge the dump from the controller hard disk without transferring it to the host.

Use the MODIFY DUMP command to do the following:

- Transfer a MOSS dump file. Specify TYPE=MOSS and ACTION=TRANSFER or COMP.
- Transfer a CSP dump file. Specify TYPE=CSP and ACTION=TRANSFER or COMP.
- Purge a MOSS or CSP dump file without transferring it to the host. Specify TYPE=MOSS or CSP and ACTION=PURGE.

If you have the NetView program in your system, VTAM is notified that the MOSS or CSP dump transfer to the host has failed. The command facility then notifies you of the failure and gives you the option of requesting the transfer of the dump file to a file in the host processor. Either you or the command facility can issue the MODIFY DUMP command.

To print the contents of the MOSS or CSP dump file, use the NCP independent dump utility described in the *NCP, SSP, and EP Diagnosis Guide*.

Selecting a dump file for a MOSS or CSP dump

When transferring a MOSS or CSP dump file, VTAM requires the name of the file in the host that receives the dump.

For a MOSS dump, the dump file can be specified in the PCCU definition statement with the MDUMPDS operand of the PCCU statement or with the DUMPDS operand on the MODIFY DUMP command. You can use the DUMPDS operand to override the specification in the PCCU definition statement or to specify a file if one was not specified. If you do not use the DUMPDS operand on the MODIFY DUMP command, and the PCCU definition statement does not include a MDUMPDS specification, VTAM places the MOSS dump in the file specified for an NCP dump. That is, the MOSS dump is transferred to the file specified by the DUMPDS operand of the PCCU definition statement.

For a CSP dump, the dump file can be specified in the PCCU definition statement with the CDUMPDS operand of the PCCU statement or with the DUMPDS operand on the MODIFY DUMP command. You can use the DUMPDS operand to override the specification in the PCCU definition statement or to specify a file if one was not specified. If you do not use the DUMPDS operand on the MODIFY

DUMP command, and the NCP definition does not include a CDUMPDS specification, VTAM places the CSP dump in the file specified for an NCP dump. That is, the CSP dump is transferred to the file specified by the DUMPDS operand in the PCCU definition statement.

To obtain a MOSS or CSP dump, follow the instructions in "Dumping an NCP" on page 654.

If available, the bit error rate (BER) log, component descriptor file (CDF), token-ring interface coupler (TIC) dump, and communication adapter dump are transferred by the MODIFY DUMP command from the hard disk along with the MOSS or CSP dump. If the CSP or MOSS dump is not present, the BER log, CDF, TIC dump, and communication adapter dump are still transferred from the hard disk, but the command fails because the dump you specified was not transferred.

IBM Communication Controller for Linux on System z considerations

With IBM Communication Controller for Linux on System z (CCL), you can run your existing NCPs (possibly with minor configuration changes) on a IBM System z machine running Linux. CCL provides an environment that mimics many of the functions provided by the 3745 Communication Controller hardware. The existing NCP product can run unchanged on a IBM System z Linux machine. Refer to the CCL product documentation for more information about IBM Communication Controller for Linux on IBM System z.

When the CCL product first became available, only LAN (token-ring or Ethernet) connectivity was supported by CCL. As a result, the only way to attach one of these early CCL NCPs to VTAM was through an External Commmunications Adapter (XCA) major node (the VTAM representation for LAN links). Because only XCA attachment was supported by CCL, VTAM provided support that enabled you to activate CCL NCPs over subarea XCA links. Prior to this support, you could configure VTAM to attach to NCPs over subarea XCA links and to exploit the virtual routes (VRs) and explicit routes (ERs) defined over these links for SSCP and LU sessions but you could not activate and own NCP resources over subarea XCA links. The new support enables you to activate (own) CCL NCPs that are directly attached to VTAM through an OSA port configured in LCS mode. The LAN attachment to VTAM is provided through an XCA-attached device (typically provided by another OSA port configured in SNA mode). In most cases, CCL NCPs attached to VTAM in this manner are treated as remote NCPs (rather than local NCPs, which implies that they are channel-attached).

Because CCL does not mimic all of the functions provided by the 3745 hardware, some functional limitations exist when running NCPs on the CCL engine when compared to running NCPs on real 3745 hardware. This information describes the operational differences and functional limitations of running NCP on the IBM Communication Controller for Linux on System z product (CCL NCP), compared to running NCP on real 3745 hardware (3745 NCP).

The IBM Communication Controller for Linux on System z V1R2 (and later) introduced additional connectivity options, including support for the new OSA OSN device, which appears to VTAM as an ESCON CDLC channel. The OSN support is fundamentally transparent to VTAM. Once the OSN CHPID and devices have been configured in your input/output configuration data set (IOCDS), simply modify your NCP definitions to specify one of the 3745 channel addresses configured for the OSN (3745) device on the CUADDR operand of the corresponding PCCU definition statement. CCL NCPs attached to VTAM using an

OSN device provide the same functionality and capabilities as channel-attached 3745 hardware; that is, they are not subject to the same operational differences as XCA-attached CCL NCPs.

Unless explicitly stated otherwise, the information presented in the following sections pertains only to CCL NCPs that are attached to VTAM using an XCA major node.

Defining and activating XCA-attached CCL NCPs

VTAM provides support to enable activation of CCL NCPs over subarea XCA links defined to VTAM using an XCA major node (VBUILD TYPE=XCA). Activation of CCL NCPs through subarea XCA PUs is supported only for leased links (DIAL=NO specified on the GROUP statement) and requires that ALLOWACT=YES be specified on the XCA PUs that represent links to CCL NCPs, as shown in the following example. (The default value of the ALLOWACT operand is NO.)

```
*********************************************************************
*  NAME:     XCA major node for host 1A
*********************************************************************
XCA1A    VBUILD  TYPE=XCA
PORT1A   PORT    MEDIUM=RING,ADAPNO=0,SAPADDR=4,CUADDR=500,TIMER=254
GP1AS    GROUP   DIAL=NO,ISTATUS=INACTIVE
LN1A12   LINE    USER=SNA
PU1A12   PU      MACADDR=404A11111111,PUTYPE=5,SUBAREA=12,TGN=1,       *
               SAPADDR=4,ALLOWACT=YES
```

The VTAM DISPLAY ID command output for subarea XCA PUs includes the letter *W* as a resource status modifier when ALLOWACT=YES is coded. Refer to the section on VTAM status codes in *z/OS Communications Server: IP and SNA Codes* for a description of this resource status modifier and the section on the DISPLAY ID command in *z/OS Communications Server: SNA Operation* for a sample of this output.

Use the existing VTAM VARY ACT, VARY INACT, VARY ACQ, and VARY REL commands to activate or deactivate CCL NCPs, or to acquire or release resources that are defined within CCL NCPs.

Loading XCA-attached CCL NCPs

To avoid making changes to the NCP and SSP products to support the CCL environment, CCL NCPs attached directly to VTAM through an XCA major node are treated as though they are remote (token-ring attached) NCPs for loading purposes. To minimize the changes required in VTAM in support of CCL NCPs attached in this manner, some restrictions are imposed on how XCA-attached CCL NCPs can be loaded.

You can use the following three methods of loading an NCP:

1. Using the VTAM VARY ACT command with LOADFROM=HOST to load and activate a CCL NCP from the host.

   This method of loading an NCP is *not* supported for CCL NCPs attached to VTAM through an XCA major node. This is because CCL relies on the native NCP token-ring (NTRI) support, and the existing NCP product does not support loading from host over token-ring.

2. Using the VTAM VARY ACT command with LOADFROM=EXTERNAL to load and activate a CCL NCP from disk. (In the CCL environment, loading a CCL NCP from disk means using an NCP load module that has previously been saved in the Linux file system.)

   This method of loading an NCP is *not* supported for CCL NCPs attached directly to VTAM through an XCA major node. When this command is used to load a real 3745 over a token-ring link, another 3745 NCP is required on the other side of the token-ring link in the host direction, because this adjacent NCP plays an important role in the load and IPL operation when loading from disk.

3. Using the VTAM MODIFY LOAD command to save NCP load modules to disk and schedule the automatic load and IPL (timed IPL) of a CCL NCP.

   This method of loading an NCP is the only method supported for CCL NCPs attached directly to VTAM through an XCA major node. VTAM provides the functions necessary to do the following:

   - Send NCP load modules from VTAM to a CCL NCP (for saving in the Linux file system).
   - Send requests to a CCL NCP to add, replace, rename, or erase NCP load modules that have been previously saved in the Linux file system.
   - Send timed IPL requests to a CCL NCP so that new or changed NCP load modules can be automatically loaded at a specified time.

   All of these functions are accomplished using existing variations of the VTAM MODIFY LOAD command. Because this command uses the SSCP-PU session to deliver these requests to the target CCL NCP, the CCL NCP must already be loaded and active (to VTAM) prior to issuing any of the MODIFY LOAD command variations. Refer to the CCL product documentation for information on how to load and activate a CCL NCP for the first time.

Table 49 and Table 50 on page 661 summarize the methods of loading an NCP that are supported for 3745 NCPs and CCL NCPs respectively for various types of NCP attachment. Table 49 shows what is supported for 3745 NCPs (for comparison purposes). Table 50 on page 661 shows what is supported for CCL NCPs (and for 3745 NCPs when attached to VTAM through CCL NCPs). Further clarification is provided in the corresponding notes following each table.

Table 49. Loading a 3745 from VTAM

Method of 3745 attachment	**VARY ACT, LOADFROM=HOST**	**VARY ACT, LOADFROM=EXTERNAL**	**MODIFY LOAD**
Direct attachment of 3745 to VTAM using channel	Supported	Supported	Supported
Direct attachment of 3745 to VTAM using LAN	Unsupported (a)	Unsupported (a)	Unsupported (a)
Remote attachment of 3745 to VTAM through 3745	Supported over SDLC Unsupported over token-ring	Supported	Supported
(a) VTAM does not support loading (or activating) 3745 NCPs over token-ring.			

Table 50. Loading a CCL or 3745 from VTAM

Method of CCL or 3745 attachment	VARY ACT, LOADFROM=HOST	VARY ACT, LOADFROM=EXTERNAL	MODIFY LOAD
Direct attachment of CCL to VTAM using channel	Supported (a)	Supported (a)	Supported (a)
Direct attachment of CCL to VTAM using LAN	Unsupported (b)	Unsupported (d)	Supported
Remote attachment of CCL to VTAM through 3745 or CCL	Unsupported (c)	Supported	Supported
Remote attachment of 3745 to VTAM through CCL	Unsupported (c)	Supported	Supported
(a) CCL V1R2 and later support channel attachment to VTAM through the new OSA OSN device.			
(b) NCP does not support LOADFROM=HOST over token-ring.			
(c) NCP does not support LOADFROM=HOST over token-ring and CCL does not support SDLC.			
(d) VTAM does not support LOADFROM=EXTERNAL over token-ring.			

Monitoring and managing XCA-attached CCL NCPs

For functions that are supported by CCL, monitoring and managing the operation of CCL NCPs is performed in much the same way as for NCPs running on 3745 hardware. For example, the VTAM DISPLAY DISK and DISPLAY NCPSTOR commands can be used to query CCL NCP disk storage and memory contents respectively.

Some VTAM commands used to monitor NCPs are intended for functions that are not supported by CCL NCPs. (For example, the MODIFY IMR command is used to start or stop intensive mode recording for a station on an SDLC line, but CCL NCPs do not support SDLC lines.) VTAM does not prevent these commands from being issued for CCL NCPs. The CCL NCP will reject requests for unsupported types of links and devices.

The VTAM MODIFY TRACE,TYPE=LINE and MODIFY TRACE,TYPE=SIT commands can also be used to start and stop NCP line and scanner interface traces respectively. However, for scanner interface traces, the trace data collected is written to one or more files in the Linux File System (rather than being sent directly to VTAM). These trace files can then be examined from the CCL MOSS console, the Linux system, or transferred to the host for problem diagnosis. Refer to the CCL documentation for more information.

Dumping XCA-attached CCL NCPs

Because CCL NCPs attached directly to VTAM through an XCA major node follow the remote NCP model and use native NCP token-ring (NTRI) support, only those functions supported by remote token-ring attached NCPs are supported for CCL NCPs.

To summarize:

- XCA-attached CCL NCPs support dynamic dumps sent directly to the host.
- XCA-attached CCL NCPs support static dumps to disk, and the subsequent transfer of these static dumps to the host. (In the CCL environment, dumping a CCL NCP to disk means writing the NCP dump to a file in the Linux file system.)

- XCA-attached CCL NCPs do not support static dumps sent directly to the host. This is true for both active and inactive (abended) NCPs.

To capture a static dump of an active XCA-attached CCL NCP, use the MODIFY DUMP command with ACTION=STORE. To capture a static dump of an inactive (abended) XCA-attached CCL NCP, you must preconfigure the CCL NCP to automatically dump to disk when it abends by ensuring that the Auto Dump/Load switch is set. You can then use the MODIFY DUMP command with ACTION=TRANSFER to transfer a static dump to the host for problem diagnosis.

Restriction: You cannot use the VTAM VARY ACT command with LOADFROM=HOST or LOADFROM=EXTERNAL to set the Auto Dump/Load switch for a CCL NCP that is directly attached to VTAM through an XCA major node. However, if you use the MODIFY LOAD command to set a Timed IPL request for a CCL NCP, CCL always ensures that the Auto Dump/Load switch is set immediately prior to executing the Timed IPL request. The only other way to set or reset the Auto Dump/Load switch for a XCA-attached CCL NCP is to use the CCL MOSS console.

Chapter 23. Tuning VTAM for your environment

This section describes the task of tuning VTAM. It contains the following topics:

- "Introduction to tuning"
- "Tuning tools" on page 665
- "Estimating the number of active sessions" on page 682
- "Common storage areas" on page 683
- "DISPLAY STORUSE pools" on page 684
- "Buffer pools" on page 691
- "Controlling NCP slowdown" on page 702
- "Maximizing coattailing" on page 702
- "Session-level pacing tuning considerations" on page 713

Introduction to tuning

Tuning is the process of balancing the network load among resources to eliminate congestion in any one resource. You need to tune VTAM to give you optimal service while using the least amount of resources. The objectives of tuning are to use storage in the host processor and in the communication controller more efficiently and to lessen the load on the host processor. For example, to use storage more efficiently, you should avoid allocating too many VTAM buffers in the host and choose a more appropriate buffer size in the communication controller.

The elements of a network can be regarded as a series of related storage spaces. A host processor provides storage for VTAM and application programs. Communication controllers contain storage for NCPs and related programs. Cluster controllers and programmable peripheral nodes also contain storage space.

When VTAM supports telecommunication services for your system, VTAM must use storage to build control blocks to keep track of sessions, blocks of data, and other information. If your system needs exceed your storage capacity, you might experience degraded response times. Similarly, an NCP can exceed its storage capacity if it receives more data from other parts of the network than it can send out.

VTAM uses storage for the following:

- VTAM modules
- VTAM buffer pools
- Tables representing major and minor nodes
- Tables representing routing capabilities of the SSCP
- Tables representing sessions controlled by the SSCP
- Temporary work space used for VTAM operator commands and session establishment or termination requests

The amount of storage required for tables and buffer pools depends on the number of major nodes being defined and the number of sessions required by application programs. By increasing VTAM efficiency, you can free storage for other uses.

Tuning tools

To tune your environment, you need to be able to collect information about your network. You need to know how to gather and analyze tuning statistics, determine

© Copyright IBM Corp. 1984, 2011

the amount of coattailing (when more than one piece of data is being transferred in or out of the host with a single I/O operation) taking place in your network, analyze slowdown conditions, and monitor your common storage areas.

VTAM provides operator commands and other facilities to help you gather this information. See "Tuning tools" on page 665 for a description of the tools that you can use to gather data to tune your network.

Estimating active sessions

When VTAM provides session services for applications and logical units, VTAM allocates storage to handle information about where data is to be sent. You can help system performance by estimating the number of sessions that VTAM will have with a particular application program or logical unit. This way, VTAM can set aside an appropriate amount of space for handling sessions. For further information, see "Estimating the number of active sessions" on page 682.

Common storage areas

VTAM uses common service area (CSA) to maintain some buffer pools and control blocks. Common storage areas are also used for other system needs, so you need to monitor VTAM use of common storage areas (for information see "Monitoring common storage areas" on page 665) and you might want to control VTAM use. For more information on limiting VTAM use of common storage areas, see "Common storage areas" on page 683.

Buffer pools

VTAM uses buffer pools to control the handling of data: VTAM control blocks, I/O buffers, and channel programs that control the transmission of data. Each buffer pool handles storage for a different VTAM service; therefore, the needs of your network will determine which buffer pools are used the most. After you determine which buffer pools are the most important to your system, you may want to change the sizes of some of the buffer pools.

When you set a buffer pool size, you are reserving storage for use only by that buffer pool. If you specify buffer pools that are larger than necessary, you are setting aside storage that is not being used. The storage is being wasted when it could be used in other areas.

When you set a buffer pool size that is too small, VTAM will reach the buffer pool limit and will then dynamically allocate more space in the buffer pool. When the current need is satisfied, VTAM will then dynamically deallocate space in the buffer pool. Specifying small buffer pools conserves storage, but can cause frequent CPU use for expansion and contraction. For information on types of buffer pools and buffer pool allocation, see "Buffer pools" on page 691.

NCP slowdown

NCP slowdown is caused when NCP runs out of buffers for transferring data to VTAM. For information on what you can do to alleviate the slowdown condition, see "Controlling NCP slowdown" on page 702.

Coattailing

Coattailing occurs when more than one piece of data is being transferred in or out of the host with a single channel input/output (I/O) operation. For example, without coattailing, each message read into VTAM generates a READ channel

program. A READ channel program uses CPU cycles. If you can read more than one message into VTAM and generate only one read channel program, then coattailing is taking place and you are saving CPU cycles. Coattailing can provide greater throughput between a host and an SNA controller because fewer interrupts and instructions are processed for each message. For further information, see "Maximizing coattailing" on page 702.

Tuning tools

This section describes the tools that you can use to gather data to tune your network.

Monitoring common storage areas

If availability of common service area (CSA) is a problem, you should monitor the VTAM buffer pools for unused storage. You can do this with the DISPLAY BFRUSE operator command or with the VTAM TYPE=SMS,ID=VTAMBUF trace. The output from this trace goes to the generalized trace facility (GTF).

The following example DISPLAY BFRUSE operator command report can be used to determine how much CSA is not being used.

```
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = BUFFER POOL DATA
IST920I IO00     BUFF SIZE  334         EXP INCREMENT   55
IST921I          TIMES EXP  0           EXP/CONT THRESH 36    / *NA*
IST922I          CURR TOTAL 110         CURR AVAILABLE  110
IST923I          MAX TOTAL  110         MAX USED        1
IST989I          EXP LIMIT  2147483647  BUFFS REQUESTED 0
IST924I -------------------------------------------------------------
IST920I BS00     BUFF SIZE  216         EXP INCREMENT   17
IST921I          TIMES EXP  0           EXP/CONT THRESH 17    / *NA*
IST922I          CURR TOTAL 34          CURR AVAILABLE  34
IST923I          MAX TOTAL  34          MAX USED        0
IST924I -------------------------------------------------------------
IST920I LP00     BUFF SIZE  2032        EXP INCREMENT   4
IST921I          TIMES EXP  0           EXP/CONT THRESH 22    / *NA*
IST922I          CURR TOTAL 64          CURR AVAILABLE  60
IST923I          MAX TOTAL  64          MAX USED        4
IST924I -------------------------------------------------------------
IST920I XD00     BUFF SIZE  681         EXP INCREMENT   5
IST921I          TIMES EXP  0           EXP/CONT THRESH 5     / *NA*
IST922I          CURR TOTAL 10          CURR AVAILABLE  10
IST923I          MAX TOTAL  10          MAX USED        0
IST924I -------------------------------------------------------------
IST920I LF00     BUFF SIZE  120         EXP INCREMENT   30
IST921I          TIMES EXP  0           EXP/CONT THRESH 33    / *NA*
IST922I          CURR TOTAL 120         CURR AVAILABLE  118
IST923I          MAX TOTAL  120         MAX USED        2
IST924I -------------------------------------------------------------
IST920I CRPL     BUFF SIZE  160         EXP INCREMENT   23
IST921I          TIMES EXP  0           EXP/CONT THRESH 10    / *NA*
IST922I          CURR TOTAL 207         CURR AVAILABLE  207
IST923I          MAX TOTAL  207         MAX USED        1
IST924I -------------------------------------------------------------
IST920I SF00     BUFF SIZE  112         EXP INCREMENT   32
IST921I          TIMES EXP  0           EXP/CONT THRESH 10    / *NA*
IST922I          CURR TOTAL 64          CURR AVAILABLE  62
IST923I          MAX TOTAL  64          MAX USED        2
IST924I -------------------------------------------------------------
IST920I SP00     BUFF SIZE  144         EXP INCREMENT   25
IST921I          TIMES EXP  0           EXP/CONT THRESH 1     / *NA*
IST922I          CURR TOTAL 50          CURR AVAILABLE  50
IST923I          MAX TOTAL  50          MAX USED        0
IST924I -------------------------------------------------------------
```

```
IST920I AP00     BUFF SIZE  56          EXP INCREMENT   56
IST921I          TIMES EXP  0           EXP/CONT THRESH 3     / *NA*
IST922I          CURR TOTAL 56          CURR AVAILABLE  56
IST923I          MAX TOTAL  56          MAX USED        0
IST924I ------------------------------------------------------------
IST920I TI00     BUFF SIZE  632         EXP INCREMENT   60
IST921I          TIMES EXP  0           EXP/CONT THRESH 29    / *NA*
IST922I          CURR TOTAL 60          CURR AVAILABLE  60
IST923I          MAX TOTAL  60          MAX USED        0
IST924I ------------------------------------------------------------
IST920I T100     BUFF SIZE  1004        EXP INCREMENT   32
IST921I          TIMES EXP  0           EXP/CONT THRESH 15    / *NA*
IST922I          CURR TOTAL 16          CURR AVAILABLE  16
IST923I          MAX TOTAL  16          MAX USED        0
IST924I ------------------------------------------------------------
IST920I T200     BUFF SIZE  2028        EXP INCREMENT   32
IST921I          TIMES EXP  0           EXP/CONT THRESH 7     / *NA*
IST922I          CURR TOTAL 8           CURR AVAILABLE  8
IST923I          MAX TOTAL  8           MAX USED        0
IST924I ------------------------------------------------------------
IST920I CRA4     BUFF SIZE  4080        EXP INCREMENT   6
IST921I          TIMES EXP  0           EXP/CONT THRESH 2     / *NA*
IST922I          CURR TOTAL 12          CURR AVAILABLE  12
IST923I          MAX TOTAL  12          MAX USED        2
IST924I ------------------------------------------------------------
IST920I CRA8     BUFF SIZE  8176        EXP INCREMENT   6
IST921I          TIMES EXP  0           EXP/CONT THRESH 2     / *NA*
IST922I   CURR TOTAL 12   CURR AVAILABLE  12
IST923I          MAX TOTAL  12          MAX USED        0
IST924I ------------------------------------------------------------
IST449I CSALIMIT = NOLIMIT, CURRENT = 1388K, MAXIMUM = 1388K
IST790I MAXIMUM CSA USED = 1388K
IST1667I SYSTEM CSA LIMIT = 1887436K
IST1831I 56% OF SYSTEM CSA STORAGE REMAINING = 1056964K
IST449I CSA24 LIMIT = NOLIMIT, CURRENT = 76K, MAXIMUM = 76K
IST790I MAXIMUM CSA24 USED = 76K
IST595I IRNLIMIT = NOLIMIT, CURRENT = 0K, MAXIMUM = 0K
IST981I VTAM PRIVATE: CURRENT = 4554K, MAXIMUM USED = 4603K
IST924I ------------------------------------------------------------
IST1565I CSA MODULES = 1384K
IST1565I CSA24 MODULES = 40K
IST1565I PRIVATE MODULES = 5205K
IST314I END
```

The general formula for determining the amount of CSA not being used for a given buffer pool follows:

```
NOTUSED = (MAXTOTAL - MAXUSED) x BUFFSIZE
```

Applying this formula to the I/O buffer pool shown in the sample DISPLAY BFRUSE command report, the amount of CSA not being used follows:

```
IOBUF unused = (110 - 1) x 334 = 36,406 bytes of fixed CSA
```

The unused CSA for the other pools shown follows:

```
BSBUF unused = 7344 bytes of fixed CSA
LPBUF unused = 121920 bytes of pageable CSA
XDBUF unused = 6810 bytes of fixed CSA
LFBUF unused = 14160 bytes of fixed CSA
CRPL  unused = 32960 bytes of pageable CSA
SFBUF unused = 6944 bytes of fixed CSA
SPBUF unused = 7200 bytes of pageable CSA
APBUF unused = 3136 bytes of fixed CSA
TIBUF unused = 37920 bytes of fixed CSA
CRA4  unused = 40800 bytes of pageable CSA
CRA8  unused = 98112 bytes of pageable CSA
```

Therefore, the conclusion is as follows:

```
TOTAL CSA unused = 406,512 bytes
TOTAL fixed     = 112,720 bytes
```

As this example illustrates, this procedure can help you to identify where CSA is not being used so that you can adjust your usage.

Analyzing slowdown

If you encounter a slowdown condition in your NCP, there are not enough buffers in the NCP to keep up with the amount of data being transferred to VTAM. An NCP slowdown condition causes a message to be issued to the operator. However, you can also examine the SLODN tuning statistic to analyze the frequency that an NCP reaches a CWALL threshold condition, which can be encountered after the NCP has entered slowdown.

Slowdown in a channel-attached SNA cluster controller means that the controller (for example, the 3174) is out of buffers. The SLODN tuning statistic identifies when an SNA controller enters this condition.

Slowdown for a multipath channel connection indicates the number of times that read subchannel has gone into slowdown. Slowdown occurs when there is a shortage of IOBUFs when deblocking a multipath channel (MPC) read buffer.

Gathering tuning information with the performance monitor interface

Some statistics pertinent to VTAM internal performance activity and resource utilization are collected by VTAM application programs known as monitors. After a monitor obtains the data, it can report information to its user. Monitors are useful in providing:

- Tuning and resource information
- Early warning of problems in the VTAM system
- Information useful in debugging
- Information for monitoring current usage and trends
- Historical data for long-term analysis

VTAM offers the performance monitor interface to facilitate the requesting and receiving of this data by monitors. This method is independent of other data collection techniques provided by VTAM, and does not affect their use.

In this collection process, VTAM provides raw data to a monitor, which is then responsible for providing:

- The operator interface to the user
- The end-user report mechanism
- Data analysis functions
- The management of any statistical database

Performance data can be selected by category to keep the overhead of statistical gathering at a minimum. Refer to *z/OS Communications Server: SNA Customization* for the types of information that can be monitored.

For information on how to implement the performance monitor interface, refer to *z/OS Communications Server: SNA Programming*.

Gathering tuning statistics

By using the TNSTAT start option or the MODIFY TNSTAT command, you can collect data that will help you set the proper values on resource definition operands that control VTAM I/O operations in your system.

You can use VTAM tuning statistics to gather information on the following connections:

- SNA controller
- Channel-to-channel
- Multipath channel
- TCP

System management facility (SMF) is required to record tuning statistics. Tuning statistics can optionally be displayed at the system console using the CNSL operand, and statistics are always recorded in the appropriate tuning statistics file. This file is an SMF data set. The tuning statistics record is SMF record type 50. The format depends upon the resource for which the tuning I/O operation is collected. The tables in this section show the formats that can be present in a tuning statistics record.

TNSTAT need not be specified in the VTAM start list to later activate tuning statistics.

No tuning statistics are provided for LANs connected through XCA lines.

Tuning statistics can be activated or deactivated for all devices simultaneously (global tuning statistics), and tuning statistics can also be activated or deactivated based on a TRLE name (TRLE tuning statistics). When a TRLE is first activated, the tuning statistics state for that TRLE is set to the global tuning statistics state. For instance, if global tuning statistics are active, TRLE tuning statistics are active for that TRLE.

Tuning input/output (I/O) operations for SNA controllers

VTAM performs I/O operations to channel-attached communication controllers and cluster controllers.

You can use VTAM tuning statistics to adjust operands in NCP and VTAM definition statements to make your system run more efficiently.

VTAM uses channel programs to send data to SNA controllers. The amount of data that VTAM can read in one operation depends on the number of buffers used by a read channel program and on the size of each buffer.

Each tuning statistics record contains information about the state of data-transfer operations between VTAM and one channel-attached SNA controller (communication or cluster controller). Each record contains statistics that cover the time period since the last tuning statistics record was written for that controller or channel-to-channel connection.

Online tuning statistics report for SNA controllers: Following is an example of a tuning statistics report for an SNA controller that can appear at the VTAM operator console:

```
IST440I    TIME = 14531106     DATE = 07093          ID = 0321-L
IST441I  DLRMAX = 3            CHWR = 109672       CHRD = 116797
IST442I    ATTN = 82624       RDATN = 73771        IPDU = 469283
IST443I    OPDU = 395495      RDBUF = 645489      SLODN = 0
IST1568I   INLP = 4202         ONLP = 4170        BFNLP = 853639
IST314I END
```

For a description of these fields, refer to message IST440I in *z/OS Communications Server: SNA Messages*.

Tuning statistics file report for SNA controllers: If statistics are produced, they are always written to the appropriate tuning statistics file in the format shown in Table 51.

Table 51. Record format for SNA controller

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	Packed	Date: 01YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	Controller name
26	4	Binary	Dump load restart requests (DLRMAX)
30	4	Binary	Write channel program count (CHWR)
34	4	Binary	Read channel program count (CHRD)
38	4	Binary	Total attention interrupts received (ATTN)
42	4	Binary	Attentions on ends of READs (RDATN)
46	4	Binary	Number of PDUs inbound (IPIU)
50	4	Binary	Number of PDUs outbound (OPIU)
54	4	Binary	Total read buffers used (RDBUF)
58	4	Binary	Number of slowdowns (SLODN)
62	1	Binary	Extension length (including this field)
63	1	Binary	Reserved
64	2	EBCDIC	Version = X'F0F4'
66	4	Binary	Reserved
70	4	Binary	Reserved
74	4	Binary	Reserved
78	4	Binary	Reserved
82	4	Binary	Number of inbound NLPs (INLP)
86	4	Binary	Number of outbound NLPs (ONLP)
90	4	Binary	Number of bytes read from inbound NLPs (BFNLP)

Table 51. Record format for SNA controller (continued)

Offset	Length	Format	Description
94	4	Binary	Reserved
98	4	Binary	Reserved
102	4	Binary	Reserved
106	4	Binary	Reserved
110	4	Binary	Reserved

Tuning I/O operations for channel-to-channel connections

VTAM performs I/O operations across a channel-to-channel link. By analyzing statistics provided by VTAM, you can select I/O buffer sizes, data transfer delay, and other options that can improve performance of these I/O operations.

VTAM uses channel programs to send data to other hosts. The amount of data that VTAM can read in one operation depends on the number of buffers used by a read channel program and on the size of each buffer.

Each tuning statistics record contains information about the state of data-transfer operations between two VTAMs (using channel-to-channel adapters). Each record contains statistics that cover the time period since the last tuning statistics record was written for that controller or channel-to-channel connection.

Online tuning statistics for channel-to-channel adapters: Following is an example of a tuning statistics report for a channel-to-channel attachment:

```
IST577I    TIME  = 10214650     DATE = 94010       ID = LINEBC0
IST578I   CHNRM  = 20          CHMAX = 0        RDBUF = 2091
IST579I    ATTN  = 17         TIMERS = 0        QDPTH = 14
IST580I  BUFCAP  = 0             PRI = 6        SLODN = 0
IST581I    IPIU  = 19            OPIU = 20     DLRMAX = 3
IST1022I  WRBUF  = 1192
IST314I  END
```

For a description of these fields, refer to message IST577I in *z/OS Communications Server: SNA Messages*.

Tuning statistics file report for channel-to-channel adapters: Table 52 shows the format of the record written to the appropriate tuning statistics file.

Table 52. Record format for channel-to-channel adapters

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	Packed	Date: 01YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	CTCA name (ID)
26	4	Binary	Dump load restart requests (DLRMAX)
30	4	Binary	Normal-sized channel program count (CHNRM)

Table 52. Record format for channel-to-channel adapters (continued)

Offset	Length	Format	Description
34	4	Binary	Large-sized channel program count (CHMAX)
38	4	Binary	Total attention interrupts received (ATTN)
42	4	Binary	Number of write buffers used (WRBUF). For packed channel programs the counts represent the number of bytes read or written. For old style (non-packed format) it is the number of buffers.
46	4	Binary	Number of PIUs inbound (IPIU)
50	4	Binary	Number of PIUs outbound (OPIU)
54	4	Binary	Total input bytes used (RDBUF). For packed channel programs the counts represent the number of bytes read or written. For old style (non-packed format) it is the number of buffers.
58	4	Binary	Number of slowdowns (SLODN)
62	1	Binary	CTCA extension length (including this field)
63	1	Binary	CTCA attachment type
64	2	EBCDIC	CTCA version = X'F0F1'
66	4	Binary	Channel program starts due to timer trigger (TIMERS)
70	4	Binary	Channel program starts due to queue depth limit trigger (QDPTH)
74	4	Binary	Channel program starts due to destination capacity limit trigger (BUFCAP)
78	4	Binary	Channel program starts due to high priority request trigger (PRI)
82	32	Binary	Reserved

Tuning I/O operations for multipath channel connections

Multipath channel (MPC) connections use XCF or channel connectivity.

Tuning I/O operations for multipath channel (MPC) connections using XCF: With XCF multipath channel attachment, VTAM uses the MVS XCF signaling facility to send data to other hosts. By analyzing statistics provided by VTAM, you can select I/O buffer sizes and other options that can improve performance of these I/O operations. The amount of data that VTAM can read in one operation depends on the size of the read buffer.

Each tuning statistics record contains information about the state of data-transfer operations between two VTAMs (using multipath channel connections). Each record contains statistics that cover the time period since the last tuning statistics record was written for that multipath channel connection.

Online tuning statistics for multipath channel adapters using XCF: Following is an example of a tuning statistics report for an XCF multipath channel attachment:

```
IST1230I TIME     = time     DATE     = date    ID = id
IST1231I IPDU     = ipdu     OPDU     = opdu
IST1569I INLP     = inlp     ONLP     = onlp
IST1232I TSWEEP   = tsweep   QSWEEP   = qsweep
IST924I  ------------------------------------------------------
IST1505I TYPE     = type     TOKEN    = token
IST1234I BSIZE    = bsize    MAXBYTES = maxbytes
IST1236I BYTECNTO = bytecnto BYTECNT  = bytecnt  DIR  = direction
IST1236I BYTECNTO = bytecnto BYTECNT  = bytecnt  DIR  = direction
IST1570I NBYTECTO = nbytecto NBYTECT  = nbytect
IST314I  END
```

For a description of these fields, refer to message IST1505I in *z/OS Communications Server: SNA Messages*.

Note: For XCF connections, *tsweep* and *qsweep* are always 0.

Tuning statistics file report for multipath channel connections using XCF: Table 53 shows the format of the record written to the appropriate tuning statistics file. For each MPC group, two records are written. The first record contains statistics for the entire MPC group. The second record contains statistics for the READ/WRITE subchannel.

Table 53. Record format for multipath channel connections (XCF)

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	packed	Date: 01YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	TRLE name
26	4	Binary	Reserved
30	8	Binary	MVS token
38	4	Binary	Number of timer sweeps (TSWEEP) [1, 3]
38	4	Binary	Number of received bytes [2]
42	4	Binary	Number of queue sweeps (QSWEEP) [1, 3]
42	4	Binary	Receive byte overflow count [2]
46	4	Binary	Number of write records [1]
46	4	EBCDIC	XCF identifier (XCF) [2]
50	4	Binary	Number of read records [1]
50	4	Binary	Transmit block size (BSIZE) [2]
54	4	Binary	Number of READ/WRITE records [1]
54	4	EBCDIC	READ/WRITE indicator (RDWR) [2]
58	4	Binary	Reserved [1]
58	4	Binary	Sent byte count [2]
62	1	Binary	CTCA extension length (including this field)

Table 53. Record format for multipath channel connections (XCF) (continued)

Offset	Length	Format	Description
63	1	Binary	CTCA attachment type X'01' [1] X'03' [2]
64	2	EBCDIC	CTCA version = X'F0F2'
66	4	Binary	Reserved [1]
66	4	Binary	Send byte overflow count [2]
70	4	Binary	Reserved
74	4	Binary	Reserved
78	4	Binary	Max transmit size (MAXBYTES) [2]
82	4	Binary	Reserved
86	4	Binary	Reserved
90	4	Binary	Reserved
94	4	Binary	Number of inbound NLPs (INLP)
98	4	Binary	Number of outbound NLPs (ONLP)
102	4	Binary	NLP byte count
106	4	Binary	NLP byte count overflow

Notes:

1. Indicates fields used only in the first record for statistics on the group.
2. Indicates fields used only in the subsequent record for statistics on the read/write subchannel.
3. TSWEEP and QSWEEP are always 0 for HPDT MPC connections.

Tuning I/O operations for multipath channel connections using channels: With MPC channel connectivity attachment, VTAM performs I/O operations across multiple single-direction channel links. By analyzing statistics provided by VTAM, you can select I/O buffer sizes, data transfer delay, and other options that can improve performance of these I/O operations.

VTAM uses channel programs or Direct Memory Access (DMA) to transmit and receive data. The amount of data that VTAM can read in one operation depends on the size of the read buffer.

Each tuning statistics record contains information about the state of data-transfer operations across the multipath channel connection. Each record contains statistics that cover the time period since the last tuning statistics record was written for that multipath channel connection.

Note: Non-HPDT multipath channel uses a special data block called a sweep that is exchanged with the adjacent host to verify that data has not been lost. A host initiating a sweep request holds all outbound multipath channel transmissions until it receives a sweep reply from the adjacent host.

A sweep is initiated when either of the following occurs:

- A timer expires in the host with the higher subarea number.
- Receive queue depth in either host is excessive.

The host initiating the sweep sends the sequence number of the last output transmit block. The adjacent host compares this number with its last input transmit block sequence number. The adjacent host then sends a response to the initiating host that includes the adjacent host last output transmit block sequence number. The initiating host makes the same comparison. If the numbers do not match, or the sweep does not complete within a time limit, the multipath channel group will be deactivated. Otherwise, normal flow continues. The tuning statistics contain a count of how many sweeps are initiated by an expired timer and how many are initiated by excessive receive queue depth.

Online tuning statistics for multipath channel connections using channels: The following is an example of an MPC channel-to-channel, QDIO and Hipersockets tuning statistics report. The first nine lines are common for all three device types. The remaining lines are specific to QDIO and Hipersockets.

```
IST1230I TIME     = time       DATE     = date       ID  = id
IST1231I IPDU     = ipdu       OPDU     = opdu
IST1569I INLP     = inlp       ONLP     = onlp
IST1232I TSWEEP   = tsweep     QSWEEP   = qsweep
IST924I -------------------------------------------------------------
IST1233I DEV      = dev        DIR      = dir
IST1234I BSIZE    = bsize      MAXBYTES = maxbytes
IST1235I SIO      = sio        SLOWDOWN = slowdown
IST1236I BYTECNTO = bytecnto   BYTECNT  = bytecnt
IST1570I NBYTECTO = nbytecto   NBYTECT  = nbytect
.
.
.
IST924I -------------------------------------------------------------
IST1233I DEV      = dev        DIR      = dir
IST1719I PCIREALO = pcirealo   PCIREAL  = pcireal
IST1720I PCIVIRTO = pcivirto   PCIVIRT  = pcivirt
IST1750I PCITHRSO = pcithrso   PCITHRSH = pcithrsh
IST1751I PCIUNPRO = pciunpro   PCIUNPRD = pciunprd
IST1752I RPROCDEO = rprocdeo   RPROCDEF = rprocdef
IST1753I RREPLDEO = rrepldeo   RREPLDEF = rrepldef
IST1754I NOREADSO = noreadso   NOREADS  = noreads
IST1721I SBALCNTO = sbalcnto   SBALCNT  = sbalcnt
IST1722I PACKCNTO = packcnto   PACKCNT  = packcnt
IST2185I FRINVCTO = frinvcto   FRINVCT  = frinvct
IST1236I BYTECNTO = bytecnto   BYTECNT  = bytecnt
IST1810I PKTIQDO  = pktiqdo    PKTIQD   = pktiqd
IST1811I BYTIQDO  = bytiqdo    BYTIQD   = bytiqd
IST924I -------------------------------------------------------------
IST1233I DEV      = dev        DIR      = dir
IST1755I SBALMAX  = sbalmax    SBALAVG  = sbalavg
IST1756I QDPTHMAX = qdpthmax   QDPTHAVG = qdpthavg
IST1723I SIGACNTO = sigacnto   SIGACNT  = sigacnt
IST1721I SBALCNTO = sbalcnto   SBALCNT  = sbalcnt
IST1722I PACKCNTO = packcnto   PACKCNT  = packcnt
IST2242I SIGMCNTO = sigmcnto   SIGMCNT  = sigmcnt
IST1236I BYTECNTO = bytecnto   BYTECNT  = bytecnt
IST1810I PKTIQDO  = pktiqdo    PKTIQD   = pktiqd
IST1811I BYTIQDO  = bytiqdo    BYTIQD   = bytiqd
.
.
.
IST314I END
```

Figure 159. Example of an MPC channel-to-channel, QDIO, and Hipersockets tuning statistics report

For a description of these fields, refer to message IST1230I in *z/OS Communications Server: SNA Messages*.

Note: For HPDT MPC connections, *tsweep* and *qsweep* are always 0.

Tuning statistics file report for multipath channel connections: Table 54 shows the format of the record written to the appropriate tuning statistics file. For each MPC group, a minimum of three records are written. For each OSA-Express MPC group, a minimum of eight records are written. The first record contains statistics for the entire MPC group. The subsequent records contain statistics for each write subchannel, read subchannel, or OSA-Express datapath queue.

Table 54. Record format for multipath channel connections (channel)

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	packed	Date: 01YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	MPC line name
26	4	Binary	Dump load restart requests (DLRMAX)
30	4	Binary	Number of inbound PIUs (IPIU) [1]
34	4	Binary	Number of outbound PIUs (OPIU) [1]
38	4	Binary	Number of timer sweeps (TSWEEP) [1, 3]
42	4	Binary	Number of queue sweeps (QSWEEP) [1, 3]
46	4	Binary	Number of write records
46	4	EBCDIC	Device address (DEV) [2]
50	4	Binary	Number of read records [1]
50	4	Binary	Transmit block size (BSIZE) [2]
54	4	EBCDIC	Subchannel polarity (DIR) [2]
58	4	Binary	Transmit or receive byte count (BYTECNT) [2, 4]
62	1	Binary	CTCA extension length (including this field)
63	1	Binary	CTCA attachment type X'01' [1] X'02' [2] X'04' [4]
64	2	EBCDIC	CTCA version = X'F0F2'
66	4	Binary	Overflow byte count (BYTECNTO) [2, 4]
70	4	Binary	Slowdown frequency (SLOWDOWN) [2]
74	4	Binary	Number of SIO issued (SIO) [2]
78	4	Binary	Max transmit size (MAXBYTES) [2]
82	4	Binary	Reserved

Table 54. Record format for multipath channel connections (channel) (continued)

Offset	Length	Format	Description
86	4	Binary	Reserved
90	4	Binary	Reserved
94	4	Binary	Number of inbound NLPs (INLP)
98	4	Binary	Number of outbound NLPs (ONLP)
102	4	Binary	NLP byte count
106	4	Binary	NLP byte count overflow
110	4	Binary	Number of OSA-Express datapath queues [1]
114	4	Binary	OSA-Express PCIR overflow for READ queue [4] OSA-Express maximum SBALs for WR/x queue [4]
118	4	Binary	OSA-Express PCIR count for READ queue [4] OSA-Express average SBALs for WR/x queue [4]
122	4	Binary	OSA-Express PCIV overflow for READ queue [4] OSA-Express maximum queue depth for WR/x queue [4]
126	4	Binary	OSA-Express PCIV count for READ queue [4] OSA-Express average queue depth for WR/x queue [4]
130	4	Binary	OSA-Express PCIT overflow for READ queue [4] OSA-Express SIGA overflow for WR/x queue [4]
134	4	Binary	OSA-Express PCIT count for READ queue [4] OSA-Express SIGA count for WR/x queue [4]
138	4	Binary	OSA-Express PCIU overflow for READ queue [4] OSA-Express SBAL count overflow for WR/x queue [4]
142	4	Binary	OSA-Express PCIU count for READ queue [4] OSA-Express SBAL count for WR/x queue [4]
146	4	Binary	OSA-Express processing deferrals overflow for READ queue [4] OSA-Express packet count overflow for WR/x queue [4]

Table 54. Record format for multipath channel connections (channel) (continued)

Offset	Length	Format	Description
150	4	Binary	OSA-Express processing deferrals for READ queue [4] OSA-Express packet count for WR/x queue [4]
154	4	Binary	OSA-Express replenishment deferrals overflow for READ queue [4] OSA-Express or Hipersockets accelerated packet count overflow for WR/x queue[4]
158	4	Binary	OSA-Express replenishment deferrals for READ queue [4] OSA-Express or Hipersockets accelerated packet count for WR/x queue[4]
162	4	Binary	OSA-Express reads exhausted overflow for READ queue [4] OSA-Express or Hipersockets accelerated byte count overflow for WR/x queue[4]
166	4	Binary	OSA-Express reads exhausted for READ queue [4] OSA-Express or Hipersockets accelerated byte count for WR/x queue[4]
170	4	Binary	OSA-Express SBAL count overflow for READ queue [4] Undefined for WR/x queue[4]
174	4	Binary	OSA-Express SBAL count for READ queue [4] Undefined for WR/x queue[4]
178	4	Binary	OSA-Express early interrupt count overflow for READ queue [4] Undefined for WR/x queue[4]
182	4	Binary	OSA-Express early interrupt count for READ queue [4] Undefined for WR/x queue[4]
186	4	Binary	OSA-Express early interrupt phase II spinout count overflow for READ queue [4] Undefined for WR/x queue[4]
190	4	Binary	OSA-Express early interrupt phase II spinout count overflow for READ queue [4] Undefined for WR/x queue[4]
194	4	Binary	OSA-Express packet count overflow for READ queue [4] Undefined for WR/x queue [4]

Table 54. Record format for multipath channel connections (channel) (continued)

Offset	Length	Format	Description
198	4	Binary	OSA-Express packet count for READ queue [4] Undefined for WR/x queue [4]
202	4	Binary	OSA-Express or Hipersockets accelerated packet count overflow for READ queue [4] Undefined for WR/x queue [4]
206	4	Binary	OSA-Express or Hipersockets accelerated packet count for READ queue [4] Undefined for WR/x queue [4]
210	4	Binary	OSA-Express or Hipersockets accelerated byte count overflow for READ queue [4] Undefined for WR/x queue [4]
214	4	Binary	OSA-Express or Hipersockets accelerated byte count for READ queue [4] Undefined for WR/x queue [4]
218	4	Binary	OSA-Express3 frame invalidation packet count overflow for READ queue [4] Undefined for WR/x queue [4]
222	4	Binary	OSA-Express3 frame invalidation packet count for READ queue [4] Undefined for WR/x queue [4]
226	32	Binary	Reserved
258	8	EBCDIC	READ queue name [4] Undefined for WR/x queue [4]

Notes:

1. Indicates fields used only in the first record for statistics on the group.
2. Indicates fields used only in subsequent records for statistics on each write and each read subchannel.
3. TSWEEP and QSWEEP are always 0 for HPDT MPC connections.
4. Indicates fields used only in subsequent records for statistics on each OSA-Express or Hipersockets datapath queue.

Tuning input/output (I/O) operations for TCP connections

VTAM performs I/O operations across TCP connections. By analyzing statistics provided by VTAM, you can improve performance of I/O operations.

Online tuning statistics report for TCP connections: Following are examples of tuning statistics reports for TCP connections. For a description of these fields, refer to the appropriate messages in *z/OS Communications Server: SNA Messages*.

CDLC:

```
IST1230I TIME     = 16500936       DATE     = 01017           ID = IUTC00B2
IST1613I TYPE     = CDLC           ATTN     =            38
IST1653I RWSIO    =            83  WCH      =            46  RCH =             43
IST1654I INPACKET =            50  INBYTE   =          5516  MAX =            243
IST1655I OTPACKET =            51  OTBYTE   =          5841  MAX =            255
IST314I END
```

CLAW:

```
IST1230I TIME     = 16591273       DATE     = 01017           ID  = IUTW0528
IST1613I TYPE     = CLAW           ATTN     =             0
IST1614I READSIO  =            51  PACKET   =            51  BYT =           5523
IST1615I ARPACKET =             1  ARBYTE   =           108  MAX =            243
IST1616I WRITESIO =            49  PACKET   =            52  BYT =           5888
IST1617I AWPACKET =             0  AWBYTE   =           111  MAX =            235
IST1618I READCCW  =            51  PCICNT   =             0
IST1619I WRITECCW =            52  APPEND   =             4
IST314I END
```

CTC:

```
IST1230I TIME     = 17050069       DATE     =  01017          ID  = IUTX0514
IST1613I TYPE     = CTC            ATTN     =             0
IST1614I READSIO  =            43  PACKET   =            48  BYT =           5594
IST1615I ARPACKET =             1  ARBYTE   =           130  MAX =            249
IST1616I WRITESIO =            44  PACKET   =            49  BYT =           5965
IST1617I AWPACKET =             1  AWBYTE   =           135  MAX =            241
IST314I END
```

HyperChannel:

```
IST1230I TIME     = 17102276       DATE     =  01017          ID  = IUTH0546
IST1613I TYPE     = HYP            ATTN     =             3
IST1614I READSIO  =            53  PACKET   =            53  BYT =           6529
IST1615I ARPACKET =             1  ARBYTE   =           123  MAX =            259
IST1616I WRITESIO =            53  PACKET   =            53  OUT =           6791
IST1617I AWPACKET =             1  AWBYTE   =           128  MAX =            271
IST314I END
```

LCS:

```
IST1230I TIME     =      17162412  DATE     =  01017          ID = IUTL0B3C
IST1613I TYPE     = LCS            ATTN     =             0
IST1614I READSIO  =            40  PACKET   =            40  BYT =             6280
IST1615I ARPACKET =             1  ARBYTE   =           157  MAX =              269
IST1616I WRITESIO =            41  PACKET   =            47  BYT =             6671
IST1617I AWPACKET =             1  AWBYTE   =           162  MAX =              281
IST314I END
```

Tuning statistics file report for TCP connections: If statistics are produced, they are always written to the appropriate tuning statistics file in the format shown in Table 55.

Table 55. Record format for TCP connections

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	Packed	Date: 01YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	TCP line name

Table 55. Record format for TCP connections (continued)

Offset	Length	Format	Description
26	4	Binary	Reserved
30	4	Binary	Count of write channel programs (not CDLC)
34	4	Binary	Count of read channel programs (not CDLC)
38	4	Binary	Number of attentions
42	4	Binary	Largest outbound packet sent
46	4	Binary	Largest packet received
50	4	Binary	Reserved
54	4	Binary	Reserved
58	4	Binary	Reserved
62	1	Binary	Extension length (including this field)
63	1	Binary	Reserved
64	2	EBCDIC	TCP version = X'F0F3'
66	4	Binary	Reserved
70	4	Binary	Reserved
74	4	Binary	Reserved
78	4	Binary	Reserved
82	4	Binary	Inbound packet count
86	4	Binary	Outbound packet count
90	4	Binary	Inbound byte count
94	4	Binary	Inbound byte count, overflow
98	4	Binary	Outbound byte count
102	4	Binary	Outbound byte count, overflow
106	1	Binary	TCP legacy type • X'10' = CTC • X'20' = LCS • X'30' = CLAW • X'40' = CDLC • X'50' = HYPERchannel • X'60' = SameHost
107	3	Binary	Reserved
110	4	Binary	Number of PCI interrupts (CLAW only)
114	4	Binary	Number of READ CCWs completed (CLAW only)
118	4	Binary	Number of WRITE CCWs completed (CLAW only)
122	4	Binary	Number of WRITE appends (CLAW only)
126	4	Binary	Number of READ/WRITE SIOs (CLAW only)

Analyzing tuning statistics

A single set of VTAM tuning statistics can be enough to indicate how a network is operating. However, these statistics become more valuable as you compare sets of values over time to see trends or the effects caused by changing buffer pool specifications and operands. You should analyze tuning statistics before and after making any change that might affect system performance.

Use tuning statistics to analyze the effects of VTAM host processor usage.

1. Compare the number of attention interrupts with channel READs. If the number of attention interrupts is the same as the number of channel READs, each READ is the result of the NCP sending the host a stand-alone attention. This condition causes increased CPU utilization because the host processor and VTAM must handle each attention interrupt. Conversely, if the number of attention interrupts is small in relation to the number of channel READs, the impact on the host is much less. If the number of attentions is large in relation to the number of channel READs (CHRD or RCH), then this could be a normal condition. Some controllers (for example, those running NCP Version 5 Release 2 and later releases) send attention polls to the primary logical unit during periods of no traffic. VTAM does not count the number of channel reads when no data is transferred.
2. Analyze your IOBUF size. If RDBUF is much larger than IPIU, the value specified for the I/O buffer size might be too small. However, you need to know the message characteristics of your network before you make decisions. Many systems receive the bulk of their inbound data in very short messages and respond with longer outbound messages. For details on choosing an appropriate value for the I/O buffer size, see "Guidelines for setting UNITSZ" on page 711.
3. Analyze data transfer operations between VTAM and an SNA controller. To analyze inbound data transfer, multiply MAXBFRU by the number of channel READs (CHRD or RCHD). MAXBFRU is specified on the GROUP, HOST, LINE, or PU definition statement. The product should be close to the value of RDBUF (the number of read buffers used). Otherwise, data is not being transferred for all the channel command words (CCWs) in the read channel program. Reducing the value of MAXBFRU should solve this problem. However, the value of MAXBFRU times UNITSZ must be at least as large as the largest PIU received. For more details on setting the value of MAXBFRU and UNITSZ, see "Guidelines for setting UNITSZ" on page 711 and "Setting the MAXBFRU operand" on page 708.
4. Examine the read attention (RDATN) information that is available in VTAM tuning statistics to understand the efficiency of the inbound data transfer operations. RDATN is the number of times that VTAM, after reading data, is requested by an attention to read more data. This happens when one of the following occurs:
   - The READ channel program CCW string is not long enough to contain all of the data sent.
   - Enough additional data comes into the communication controller during the read operation to cause it to request VTAM to do another read.
   - The controller is short on buffer space and is requesting a read to confirm VTAM receipt of data sent by the controller, so that the controller buffers can be reused.

   Although this coattailing of PIUs is preferable to stand-alone attention interrupts, a large RDATN value is not desirable. To increase the size of the

CCW string, increase the MAXBFRU value. For more details on setting the value of MAXBFRU, see "Setting the MAXBFRU operand" on page 708.

5. Analyze the outbound data transfer operation between VTAM and an SNA controller by analyzing VTAM blocking of outbound PIUs. To determine the average number of PIUs for each write operation, divide the outbound PIU (OPIU) count by the number of channel WRITEs (CHWR or WCH). This average number of PIUs indicates the effectiveness of the VTAM blocking algorithm. Increasing DELAY can also increase inbound coattailing. For more information on setting the value of DELAY, see "Guidelines for setting DELAY" on page 705.

Determining the amount of coattailing in your system

You can analyze the amount of coattailing in your system by examining VTAM tuning statistics. The tuning statistics indicate the total number of READ (CHRD or RCH) and WRITE (CHWR or WCH) channel programs that are used for data transfer operations. This information can be used to tune your system to improve coattailing if necessary. Increasing coattailing in your system can impact response times because coattailing means there are data transfer operations delays. When examining tuning statistics, also consider that short periods of high-attention counts can simply indicate a light load on the network.

Migrating from user-replaceable constants

User-replaceable constants were formerly used to control various VTAM functions, such as:

- Buffer use storage management services (SMS) trace "snapshot" value
- Maximum RU size for sessions
- BSC 3270 timeout (inoperative) value
- Virtual-route-selection subtask interval analysis block
- Host intermediate routing node (IRN) slowdown storage
- NetView trace data buffers
- Directory size of the symbol resolution table for the network

User-replaceable constants are now set only as start options. For additional information, refer to *z/OS Communications Server: SNA Customization*. For more information on start options, refer to the *z/OS Communications Server: SNA Resource Definition Reference*. Start options can be displayed with the DISPLAY VTAMOPTS command, and many of them can be changed with the MODIFY VTAMOPTS command. For more information on the DISPLAY VTAMOPTS and MODIFY VTAMOPTS commands, refer to *z/OS Communications Server: SNA Operation*.

Estimating the number of active sessions

The EAS operand on the LU and APPL definition statements specifies the estimated number of active sessions between VTAM and the application program or VTAM and the logical unit.

EAS operand for application programs

The EAS operand on the APPL definition statement indicates the amount of storage to allocate for a table that contains information on any sessions with the application program. The EAS value that you choose can affect performance in your system.

Specifying an EAS value that is too small can cause an increase in path length for VTAM processing of inbound data flows. The increase is due to VTAM scanning

control blocks for some of the sessions. As more sessions become active, the directory entries are used more frequently, and the chains become longer. Longer chains require more scan time. If you specify a larger EAS value, the chains for particular directory entries may be smaller and may use less CPU time for scanning the chains.

For a non-TSO application program, you should specify an EAS value that equals the average number of sessions, but does not exceed 32 767. If you have virtual storage constraints, you can specify a smaller EAS value.

For a TSO application program, you should specify EAS=1.

EAS operand for independent logical units

For performance reasons, if you anticipate that a particular logical unit can have more than 256 sessions, increase the value of the EAS operand to the nearest multiple of 256. For example, if you anticipate 700 sessions for a particular logical unit, you should define EAS to be 768 (3 x 256). Coding a poor estimate for EAS does not affect your sessions, but it does affect performance.

You can define the EAS operand for an independent logical unit only.

Common storage areas

Common storage areas (CSA) are used by VTAM and other resources in your network to maintain buffers and control blocks in the CSA for MVS systems. For performance reasons, VTAM offers you some control over its use of common storage areas with the start options. These options are discussed in the following sections.

Common service area limit

The CSALIMIT start option specifies the maximum amount of common service area (CSA) that is used by VTAM. It can be used to prevent VTAM from using CSA that is needed by the operating system. Most environments do not need to limit VTAM use of CSA because their peak CSA usage is a very small increment over their requirements when all of their logical units are in session.

If the limit specified by CSALIMIT is reached, the results are unpredictable. If LPBUF cannot be expanded, VTAM might enter a deadlock condition. Therefore, if you specify the CSALIMIT start option, you should define LPBUF large enough so that it does not have to expand. Other possible occurrences are that messages can be lost or a session initiation or termination can fail. Coding a value for CSALIMIT without the ,F modifier can limit these exposures. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for details.

Common service area 24-bit

The CSA24 start option specifies the maximum amount of 24-bit addressable common service area (CSA) that can be used by VTAM. The largest possible value for CSA24 is 16 megabytes.

The CSA24 start option limits only the amount of 24-bit addressable storage explicitly requested by VTAM. The CSA24 limitations do not apply when MVS returns 24-bit addressable storage in response to a VTAM storage request above the 16-megabyte line. MVS returns 24-bit addressable storage below the line because

not enough CSA above the line is defined. CSA storage can be allocated from above or below the line depending on where the allocating procedure resides.

DISPLAY STORUSE pools

The DISPLAY STORUSE command provides a way to remedy a possible shortage of storage space. Table 56 provides a list of storage pools that are displayed using the DISPLAY STORUSE command. Included for each pool is a short description of the pool function and characteristics. These pools are not customer-defined, unlike the buffer pools defined using the VTAM start options (for example, IOBUF). VTAM allocates and deallocates storage from these pools as needed.

If VTAM is in a storage shortage situation, Table 56 and the output from the DISPLAY STORUSE command can be used to determine where excess storage is being used, enabling you to take appropriate action to remedy the shortage.

Table 56. DISPLAY STORUSE pools

Pool name	Storage location	Description
ACDEB	SYSTEM	A pool element is allocated for every active application.
ACPCB	USER	An element is allocated for every adjacent control point with which this node has an active CP-CP session.
ADJCP	USER	Each pool element defines a single adjacent control point (ADJCP).
ADJNODE	USER	Elements are allocated for each CP-CP session partner to track topology flow status with an adjacent node.
ALPHCD	SYSTEM	Element used by HPDT MPC to manage CSM buffer descriptors. These descriptors represent the storage used for the physical transmission of data over HPDT MPC.
AMU	SYSTEM	Elements are short-lived signals used for intraproduct communication.
ANDCB	USER	One element is allocated the first time a node activates a link supporting CP-CP sessions with a specific adjacent node.
ATGB	USER	An element represents a single T2.1 connection or VR-TG to an adjacent CP.
AUTOLOGN	USER	Elements are used to keep track of autologon relationships.
BFRTRACE	SYSTEM	Elements are used to hold small buffer trace records.
BFRTRFUL	SYSTEM	Elements are used to hold large buffer trace records.
BSBEXT	USER	Elements are allocated for each session using SNA/IP support.
CAB	SYSTEM	One element is allocated for each VCNS connection resulting from a LOGON to a VCNS line (LANs) or from an X.25 OPEN command.
CACHE	USER	Elements are used for caching PCIDs during direct search list processing.
CANT	SYSTEM	One element is allocated for every 64 VCNS X.25 connections for the same LOGON to a VCNS line.
CDAJSCP	USER	Elements are used to define adjacent SSCP entries.

Table 56. DISPLAY STORUSE pools (continued)

Pool name	Storage location	Description
CDRSC	USER	Elements are used to define dynamic CDRSCs and clone CDRSCs.
CFSACCCD	SYSTEM	Coupling facility short-lived common storage pool.
CFSACCCS	SYSTEM	Coupling facility long-lived common storage pool.
CFSACCPD	USER	Coupling facility short-lived private storage pool.
CFSACCPS	USER	Coupling facility long-lived private storage pool.
CFSBUFCD	SYSTEM	Elements are short-lived buffer objects used to manage coupling facility structure data buffers.
CFSBUFCS	SYSTEM	Elements are long-lived buffer objects used to manage coupling facility structure data buffers.
CFSBUFPD	USER	Elements are short-lived buffer objects used to manage coupling facility structure.
CFSBUFPS	USER	Elements are long-lived buffer objects used to manage coupling facility structure.
CFSCSA	SYSTEM	Coupling facility common storage pool.
CFSPRIV	USER	Coupling facility private storage pool.
CMIPPVT	USER	CMIP services allocates most of its buffers from this pool.
CMOBJ	SYSTEM	Elements are used by VTAM connection manager to represent logical connections using HPDT DLCs.
CNSFACUD	SYSTEM	An element is allocated every time a VCNS application asks VTAM to initiate or receive an X.25 call request that will contain facilities or call user data.
CORCB	USER	Elements are used for APPN Locate request correlation.
COS	USER	Elements are used for APPN COS definitions and mode table mappings.
COWE	USER	One element is allocated by the VTAM topology agent for each CMIP operation request the agent processes. The element is freed when the operation ends.
CPRUPE	USER	Elements are request/response unit processing elements used when processing APPN-related requests.
CPWACSA	SYSTEM	An element is allocated when a USS command is entered from the network operator console or from a user terminal. One element from the pool is allocated when an application resource definition specifying SSCPFM=USSNOP is being processed.
CPWAPVT	USER	One element is allocated when a USS command is entered from the network operator console or from a user terminal. One element from the pool is allocated when an application resource definition specifying SSCPFM=USSNOP is being processed.
DCX	SYSTEM	Pool elements are used to maintain data compression information.
DDEL	USER	Elements are used to delay the disconnection of a PU that is defined with the DISCNT=DELAY parameter.
DECB	USER	An element is allocated for each resource in the APPN directory database.

Table 56. DISPLAY STORUSE pools (continued)

Pool name	Storage location	Description
DISKIO	USER	One element is allocated per component performing database hardening. The storage for this pool is allocated below the 16–M line.
DMTSQ	USER	One element is allocated every time a message contained in the message flooding table is issued. The allocated element is freed when the message suppression time expires.
DSERVER	USER	Elements are control blocks and short-term signals related to directory services and interchange nodes.
DSSIB	USER	An element is allocated when a DSRLST is received and freed when the DSRLIST response is sent.
DSUTIL	USER	Elements are used to perform locate search processing.
DYPATH	USER	Elements are path table entries for dynamic PUs.
EEHNMIPD	SYSTEM	An element from this pool is allocated when the SNAMGMT start option is set to YES and a client application connects to the SNA Network Management socket. More elements are allocated when a response is built due to a request from a client application. These additional elements are freed when the response is returned. The original element is freed when either the client or VTAM terminates the connection.
EPTDVT	SYSTEM	Elements are used to contain DLC-specific information.
ERICPOOL	USER	Elements are used while parsing request/response units. When the RU processing completes, the elements are freed.
ERTE	USER	Elements are used to define explicit routes resulting from PATH statements.
FMCB	SYSTEM	Elements are allocated once for each session associated with an application LU (for nonpersistent LU sessions) at session BIND time.
FMCBEXT	SYSTEM	Elements are allocated once for each session associated with an application LU at session BIND time.
FMH5	SYSTEM	One element is allocated for each incoming LU 6.2 conversation request.
GRINS	USER	Elements are used to maintain associations between network resources and generic names.
GWNAJSCP	USER	Elements are: • Adjacent SSCP routing tables used to route CDINIT or DSRLST RUs. Elements are freed when the routing completes or fails. • Information to determine the gateway NCP to use during session setup.
HIPOOLPS	USER	HPR table used by MNPS
HPRINFO	USER	Elements exist for each RTP connection for which data is being collected by a single performance monitor application.
HSICB	SYSTEM	Elements are allocated once for each APPC session associated with an application LU at session BIND time.

Table 56. DISPLAY STORUSE pools (continued)

Pool name	Storage location	Description
HSQH	SYSTEM	Elements are allocated once for the first of every 107 sessions set up across a VR, at session BIND time.
IOBLOCKL	SYSTEM	Elements in the pool are large DLC-related control blocks (for example a channel-attached NCP). The storage for this pool is fixed and is not paged out of memory by the operating system.
IOBLOCKP	USER	Elements in the pool are large DLC-related control blocks. The storage for this pool is not fixed and can be paged out of memory by the operating system.
IOBLOCKS	SYSTEM	Elements in the pool are small DLC-related control blocks (such as for a channel-attached NCP). The storage for this pool is fixed and is not paged out of memory by the operating system.
IOSIB	USER	Elements are used to process Init_Other (Cross-domain) requests.
IPADDR	USER	Element used to store IP addresses and host names.
ISTENDEL	USER	One element is allocated by a network node for each adjacent served end node or nonnative network node.
ISTSITCB	USER	One element is needed at endpoint and network node server roles for each APPN Search procedure.
ISTTRCEL	USER	Elements are used for the problem determination (PD) trace function of the CNM interface.
KEYTOKEN	SYSTEM	Elements map active cryptographic session key tokens.
LCB	USER	An element is allocated to process locate search requests.
LMTABLE	SYSTEM	Three types of elements come from this pool: • One element is allocated for every pair of LUs that have negotiated APPC session limits. The element is allocated when a CNOS with a partner LU is initiated, and freed when the application closes its ACB. • An element is used for every application that opens its ACB with APPC=YES. • Elements represents current session limits between two LU 6.2s on a particular mode. An element is allocated for every logmode that has been negotiated between two partner LUs. The elements are freed for deletion, or freed when the application closes its ACB.
MARB	USER	Elements are allocated by the VTAM topology agent when the agent sends response data to CMIP services. Elements are freed when the agent is notified that data was received.
MRPOOLPS	USER	Elements contain MNPS RTP information.
NDREC	USER	Elements are used for APPN topology node information.
NIDCB	USER	An element is allocated for each network identifier known to the APPN directory database.
NLPDELPD	USER	Elements contain MNPS NLP entry IDs.
NQDAT	USER	One element is allocated for each network-qualified SRTE.
NSRUL	SYSTEM	Elements are used to process LU 6.2 session activation. NSRUL is used for larger-sized requests.

Table 56. DISPLAY STORUSE pools (continued)

Pool name	Storage location	Description
NSRUS	SYSTEM	Elements are used to process LU 6.2 session activation. NSRUS is used for smaller-sized requests.
NSS	SYSTEM	Elements are used to process LU 6.2 session activation.
OSCB	USER	Elements are used to track outstanding locate search requests.
PAGBLBSB	USER	Elements are allocated for each session using HPR or SNA/IP support.
PAQ	USER	Elements are used to track PLU network addresses for a given LU.
PCDCA	USER	Elements are used for border node PCID caching.
PGIOBLK	SYSTEM	Elements are used when communicating with the 3172 or the OSA.
PLOCB	USER	Elements are used to process locate search requests and replies.
PLUSC	SYSTEM	Elements are used during persistent LU session CLOSE processing.
PLUSDATA	SYSTEM	Elements are allocated once for each persistent LU session associated with an application LU, at session BIND time. An additional element is allocated for each MNPS session at session BIND time.
PLUSFMCB	SYSTEM	Elements are allocated once for each persistent LU session associated with an application LU, at session BIND time.
POAPRIV	USER	One element is allocated for each message destined for a program operator application (POA). If the message requires a reply, a second POAPRIV element is allocated.
POWECOMM	SYSTEM	One element is allocated for every message issued when VTAM is running under a user task.
POWEPRIV	USER	One element is allocated for every message issued when VTAM is running under the VTAM task.
POWMCOMM	SYSTEM	One element is allocated for every single-line message and for every message group when VTAM is running under a user task.
POWMPRIV	USER	One element is allocated for every single-line message and for every message group when VTAM is running under the VTAM task.
PRDLE	SYSTEM	Elements represent random data being used for establishing LU 6.2 sessions with session-level security.
PRIDBLK	USER	Elements map procedure-relation identifier blocks (PRIDs)
PRIDQAB	USER	Elements are used to maintain procedure-relation identifier blocks (PRIDs).
PULURDTE	USER	Elements are used to define dynamic and predefined PUs and LUs.
PVTSTATC	USER	Generic utility pool for large blocks of storage that must be in VTAM private storage for an extended period of time.

Table 56. DISPLAY STORUSE pools (continued)

Pool name	Storage location	Description
PXBFIXED	SYSTEM	Elements are used to expand fixed buffer pools. The storage for this pool is fixed and is not paged out of memory by the operating system.
PXBPAGED	SYSTEM	Elements are used to expand pageable buffer pools.
RAB	SYSTEM	One element is allocated for each APPC conversation.
RAQ	USER	Elements are used to queue requests when a usable network address is not available and an RNAA must be sent.
RIBRANT	SYSTEM	One element is allocated for each LOGON of a VCNS application to a VCNS line, plus one extra element for every 16 LOGONs.
RPMNPSPS	USER	RTP elements used by MNPS
RTPINFO	USER	Contains a number of different elements, all of which are used for HPR. Both the RTP and RCM components use this pool.
RUCON	USER	One element is allocated each time a DISPLAY ROUTE,TEST=YES command is issued. After 168 elements have been allocated, VTAM will delete any elements that have not been used for more than 30 minutes.
RUPECOMM	SYSTEM	Elements are allocated from this pool to process request/response units (RUs) when execution is taking place in the VTAM address space.
RUPEPRIV	USER	Elements are allocated from this pool to process request/response units (RUs) when execution is taking place in a non-VTAM address space.
SAB	SYSTEM	One element is allocated for each APPC session.
SCCB	USER	Elements are used to perform search concentration.
SIB	USER	One element is allocated for each LU-LU session.
SIBEXT	USER	One or two elements are allocated for each cross-network session. The elements are freed when the session ends.
SIBIX	USER	One element is allocated during session initiation. The element is freed when the session becomes active.
SLD	SYSTEM	Elements are allocated to process APPCCMD CONTROL=OPRCNTL,QUALIFY=DISPLAY macro instructions, including those issued from the operator console.
SLENT	USER	A pool element is allocated when a CPSVCMG session is activated between an end node and serving network node, or for any CP SNASVCMG sessions that VTAM management services transport activates.
SPTPOOL	SYSTEM	Holds all of the SPTAEs for the associated pools.
SRTE	USER	Elements are entries in the symbol resolution table.
SSCPFMCB	USER	One element is allocated for each SSCP-PU or SSCP-LU session.
STB	USER	Elements store information concerning a T2.1 adjacent link station that an independent LU is using for session connectivity.

Table 56. DISPLAY STORUSE pools (continued)

Pool name	Storage location	Description
TCPIOCD	SYSTEM	TCP/IP IO buffer pool used for QDIO.
TGP	USER	One element is created for each TG profile (TGP) entry.
TGREC	USER	Elements are used for APPN topology TG information.
TIPACX	SYSTEM	Elements contain control information to support HPDT services. The storage for this pool is fixed and is not paged out of memory by the operating system.
TREEBLD	USER	Elements are used for APPN routing tree construction and maintenance.
TRSINFO	USER	Elements are used for topology broadcast lists (for use during topology database update broadcasting) and endpoint TG vector information (during route calculation).
UECB	SYSTEM	One element is allocated each time a user exit is to be scheduled.
UNSOL	USER	An element is allocated for every adjacent control point with which this node has an active CP-CP session.
UTILCSAL	SYSTEM	Generic utility pool for large blocks of storage that must be in CSA.
UTILCSAS	SYSTEM	Generic utility pool for small blocks of storage that must be in CSA.
UTILPVTL	USER	Generic utility pool for large blocks of storage that must be in private storage.
UTILPVTS	USER	Generic utility pool for small blocks of storage that must be in private storage.
UVRPL	USER	One element is allocated each time a user exit is to be scheduled.
VRDCB	SYSTEM	An element exists for each virtual route for which data is being collected by at least one performance monitor application.
VRPL	SYSTEM	Elements are VTAM copy of an application request parameter list (RPL). Elements are also used for BINDs and other RUs received from the network.
VRRSB	USER	An element exists for each virtual route for which data is being collected by a single performance monitor application.
WAR	USER	Elements represent autologon session originators (OLU-SLU) waiting for the availability of a required PU resource.
WREEID	USER	Elements are used to suspend and resume VTAM processes.
XNINFO	USER	Elements are used for search processing.

Note: Unless otherwise specified:

- All pool storage is pageable and can be paged out of memory by the operating system.
- All pool storage can be located above or below the 16M line.
- Some of the pools above are defined to be associated with certain VTAM tasks. Pools are associated with VTAM tasks to improve performance during storage allocation. The DISPLAY STORUSE command will not display usage for the associated pools. The total storage used does account for storage allocated by these pools

Buffer pools

VTAM uses buffer pools to control the handling of data. It dynamically allocates and deallocates space in buffer pools for the VTAM control blocks, I/O buffers, and channel programs that control the transmission of data. Specifying large buffer pools can waste storage, but does not require frequent CPU use for expansion. Specifying small buffer pools conserves storage, but requires frequent CPU use for expansion and contraction. With proper tuning, you can achieve a balance between storage use and performance that is suitable for your environment.

Types of buffer pools

Table 57 lists the types of buffer pools.

Fixed buffer pools are allocated in extended CSA subpool 227 with KEY=6. Pageable buffer pools are allocated in extended CSA subpool 231 with KEY=6.

Table 57. VTAM buffer pools

Buffer pool	Use	Default storage type	Recommendations and requirements
APBUF	Used to provide fixed common storage if VTAM fails to obtain other storage from the operating system. For buffers not related to I/O.	Fixed	IBM-supplied values are appropriate for most systems.
BSBUF	Used to maintain session information for peripheral nodes for which VTAM performs boundary function. One buffer is required for each boundary LU session (SSCP-PU, SSCP-LU, LU-LU).	Fixed	Set base number to the average number of concurrent boundary LU sessions.
CRA4	Used for scheduling and error recovery.	Pageable	IBM-supplied values are appropriate for most systems.
CRA8	Used for scheduling and error recovery.	Pageable	IBM-supplied values are appropriate for most systems.
CRPLBUF	RPL-copy pool. In general, one buffer is required for each VTAM application program request until the operation is complete.	Pageable	Monitor output from the DISPLAY BFRUSE command to ensure that the number of buffers specified is adequate for your system.
IOBUF	Used for input/output data. Every PIU that enters or leaves VTAM resides in an I/O buffer. This pool is 31-bit addressable.	Fixed	Monitor output from the DISPLAY BFRUSE command to ensure that the number of buffers specified is adequate for your system. Ensure that the largest MAXBFRU value is less than *xpanpt* minus *slowpt*. Also ensure that *bufsize* is greater than or equal to the UNITSZ operand on the HOST definition statement in the NCP. **Note:** Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for additional requirements and restrictions concerning the IOBUF pool.

Table 57. VTAM buffer pools (continued)

Buffer pool	Use	Default storage type	Recommendations and requirements
LFBUF	One buffer is required for each active application program with an EAS value (on APPL definition statement) less than 30. If the EAS value is greater than 30, this information is contained in SFBUF. One buffer is required for each TSO user who is logged on.	Fixed	IBM-supplied values are appropriate for most systems.
LPBUF	Used for scheduling and audit trail (error recovery). One buffer is required for each active VTAM process.	Pageable	Set *baseno* to 9, and set *xpanno* in the range 4–6 to force 2-page expansion.
SFBUF	Used to contain application program information and LU blocks. One buffer is required for each active application program with an EAS value (on APPL definition statement) greater than or equal to 30.	Fixed	IBM-supplied values are appropriate for most systems.
SPBUF	Used for large message (LMPEO) requests. One buffer is required per concurrent LMPEO send request.	Pageable	IBM-supplied values are appropriate for most systems.
TIBUF	Used to perform input/output operations for CSM-capable protocols. This pool is in 31-bit storage.	Fixed	IBM-supplied values are appropriate for most systems. The default value for this pool is set at a conservative value in case the functions that use this pool are not used. If using the functions that use this pool, use the DISPLAY net,BFRUSE command to monitor usage and then set the BASENO for the pool at the normal high period usage. For HPR and IP - Defining more initial buffers and larger buffer extents may help increase throughput. Defining too few may cause an IP retransmit.
T1BUF	Similar to the TIBUF but larger. Used as a packing buffer by HiperSockets accelerator and QDIO. Also used to contain the HPR headers and the media, IP, and UDP headers for an Enterprise Extender connection.	Fixed	IBM-supplied values are appropriate for most systems. The default value for this pool is set at a conservative value in case the functions that use this pool are not used. If using the functions that use this pool, use the DISPLAY net,BFRUSE command to monitor usage and then set the BASENO for the pool at the normal high period usage. For HPR/IP - Defining more initial buffers and larger buffer extents might help increase throughput. Defining too few might cause an IP retransmit.

Table 57. VTAM buffer pools (continued)

Buffer pool	Use	Default storage type	Recommendations and requirements
T2BUF	Similar to the T1BUF but larger. T2BUFs are used exclusively for HiperSockets accelerator and QDIO when the T1BUF is perceived to be of insufficient size to pack all the headers and data. Also used by HPR for all retransmissions over HiperSockets and QDIO.	Fixed	IBM-supplied values are appropriate for most systems. The default value for this pool is set at a conservative value in case the functions that use this pool are not used. If using the functions that use this pool, use the DISPLAY net,BFRUSE command to monitor usage and then set the BASENO for the pool at the normal high period usage. For Enterprise Extender connections using QDIO/iQDIO device drivers, defining more initial buffers and larger buffer extents might help increase throughput. Defining too few might cause an IP retransmit.
XDBUF	Used for physical I/O to VTAM peripheral nodes for connection activation (for example, XID exchange processing). This pool is 31-bit addressable.	Fixed	IBM-supplied values are appropriate for most systems. Because this buffer pool handles the concurrent activation of physical units, you should specify a small size and enable dynamic buffering. The buffer pool can then expand to accommodate the concurrent activation and can shrink after the resources are active.

Buffer pool allocation

VTAM provides two types of buffer pool storage allocations, basic and dynamic.

Basic allocation
The amount of space that you reserve for the buffer pool when VTAM is started.

Dynamic allocation
The process by which VTAM temporarily increases the size of a buffer pool when there are heavy demands for space in that pool.

While basic allocation sets the base size of your buffer pools, dynamic expansion enables a buffer pool to be expanded temporarily during periods of heavy demand. Its use can greatly increase the efficiency with which VTAM uses storage, particularly for I/O buffers.

Without dynamic expansion of a pool, basic allocation parameters must be specified large enough to meet the greatest possible demands on the pool. With dynamic expansion, smaller basic allocation values can be specified, and peak demands on the pool can be met with dynamic expansion. Dynamic expansion is strongly recommended for most buffer pools because the peak demand can vary considerably from the normal demand.

Setting buffer pool allocations

To set buffer pool space, use the buffer pool start options:

```
poolname=(baseno,bufsize,slowpt,F,xpanno,xpanpt,xpanlim)
```

For a description of these operands, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

The first four operands of the start option (*baseno*, *bufsize*, *slowpt*, and *F*) establish the base size and the characteristics of the buffer pool. The last three operands (*xpanno*, *xpanpt*, and *xpanlim*) specify how dynamic expansion of the buffer pools is to be performed.

Set basic allocation for each buffer pool when VTAM is started. If you do not specify a start option parameter for a buffer pool, VTAM uses an IBM-supplied value for the missing parameter. The IBM-supplied values are not necessarily the most efficient values for your system, and they are not necessarily compatible with any dynamic allocation specifications you might make.

How dynamic expansion operates

1. You enable dynamic expansion by coding the *xpanno* and *xpanpt* operands on the buffer pool start option.
2. The buffer pool size reaches the point specified by *xpanpt*.
3. VTAM acquires more buffers in blocks to accommodate the increased demand. The size of the blocks is specified by *xpanno*. VTAM acquires buffers as needed until the pool size reaches *xpanlim* (if specified).
4. Contraction is determined by *xpanno* and *xpanpt*. The *xpanno* value used is the *xpanno* value that was specified when VTAM was started, but rounded upward to the number of buffers that fills the nearest whole page of storage. If the buffer pool is the IO00 or TI00, when the number of available buffers is greater than or equal to (3 x *xpanno*) + *xpanpt*, VTAM verifies that the buffers that it acquired in previous expansions of the pool are not in use. For all other pools, the value is (2 x *xpanno*) + *xpanpt*. (If they are not, VTAM releases these buffers in blocks (as when they are acquired). If any of the buffers in a block are in use, VTAM does not release that block of buffers.

Figure 160 on page 695 shows the structure of a pool after basic allocation **A** and after one dynamic expansion of the pool **B**.

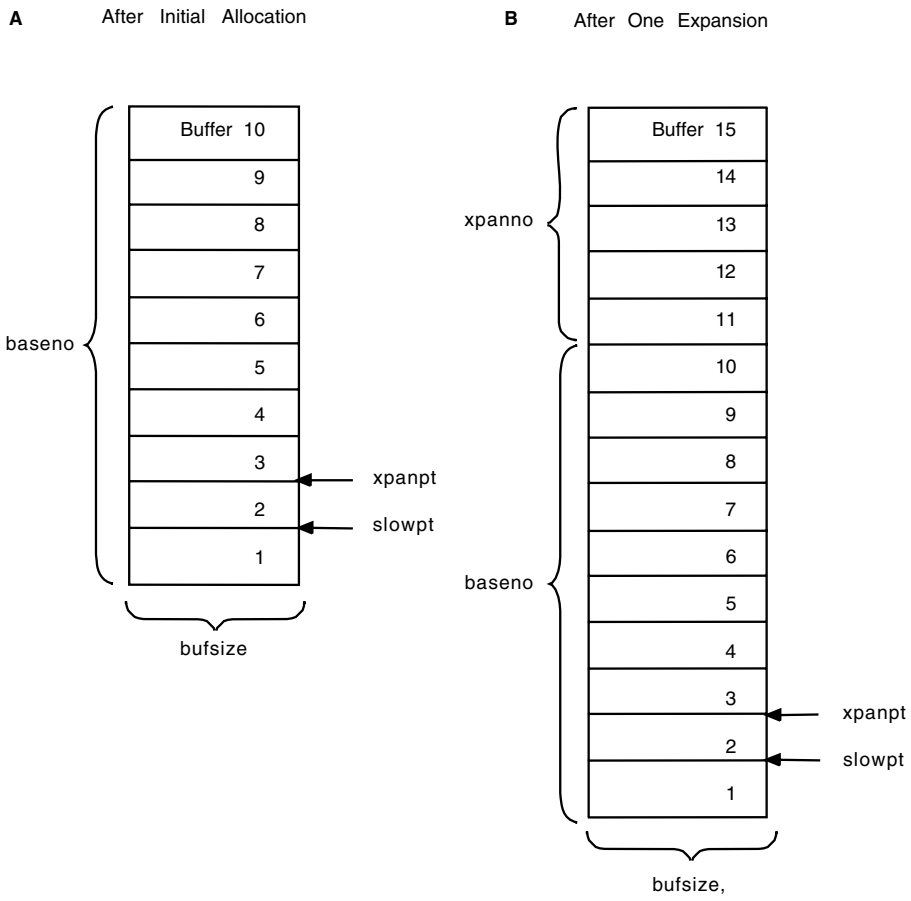


Figure 160. Buffer pool after initial allocation and after one expansion

A This example shows a buffer pool for which the start options were specified as `poolname=(10,bufsize,1,,5,2,xpanlim)`. After initial allocation, the pool contains ten buffers (baseno=10), the length in bytes of each buffer is *bufsize*, the slowdown point (*slowpt*) is one, the expansion size (*xpanno*) is 5 (assume that five buffers fill one page of storage), and the expansion point (*xpanpt*) is 2. The maximum allowed size of this buffer pool is determined by the value *xpanlim*.

B After one expansion, there are 15 buffers in the pool. Each of the five additional buffers has a length of *bufsize* and the same expansion point and slowdown point as before.

Guidelines for dynamic expansion

The following guidelines help you determine appropriate buffer pool values for your system.

If *xpanno* or *xpanpt* has a value of 0, no expansion occurs.

- **Set *baseno* for the I/O buffer to the steady state value plus *xpanpt*.**

  You can determine steady state by issuing the DISPLAY BFRUSE operator command. CURR TOTAL minus CURR AVAIL gives the number of buffers in use. Repeat this procedure several times, and take the average to determine the steady state value. Be sure to include in your calculation displays taken during times of peak network usage.

  Any additional buffers needed can be obtained using dynamic expansion.

- **Define dynamic expansion so that pools are expanded one page at a time.**

  To do this, define *xpanno* to be the number of buffers that fit on one page. To calculate *xpanno* for I/O buffers, use Table 59 on page 698, and do the following:

  1. Find the *bufsize* range in the table that includes the *bufsize* value that was specified for your system.
  2. Round up the current *bufsize* to the largest even number that appears in that range. (Some channel-attached SNA devices require that the I/O *bufsize* be an even number of bytes. Therefore, VTAM rounds up *bufsize* to an even number if an odd number is specified.)
  3. For one-page expansion, specify an *xpanno* value that is equal to the number in the left column of this row (the number of I/O buffers for each page).

- **Tune the expansion increments (*xpanno*) and expansion points (*xpanpt*) to keep CPU overhead from dynamic buffering low.**

  – Expansion and contraction of buffer pools increase the amount of real storage that VTAM requires. If the buffer pools are continually expanding and contracting, you should change the values specified for *xpanno* and *xpanpt*. You can determine what values to specify by monitoring the output of the DISPLAY BFRUSE operator command or the SMS buffer trace and by modifying the values for *xpanno* and *xpanpt* accordingly.

  – If the expansion point is set too small, VTAM might not be able to expand the buffer pool fast enough if there is a rapid demand for buffers. Setting the expansion point too small also causes the buffer pool to contract frequently.

    Message IST561I might indicate that storage is temporarily unavailable due to this rapid demand. Adjust the *xpanpt* and *xpanno* values to eliminate this problem.

    The expansion point for I/O buffers should be larger than the largest single request for buffers. Following are examples of the largest single request:
    - MAXBFRU allocation
    - NetView session monitor trace buffer size
    - JES/NJE TPBFSIZ

  – If excessive expansion and contraction are a problem (especially for LPBUF and CRPLBUF), define these buffers to expand two pages at a time.

    The buffer pool can expand and contract too frequently if the expansion increment and expansion point are not large enough to handle normal fluctuation in the pool. The I/O buffer pool is prone to this type of thrashing from application programs that send very large messages. VTAM always rounds up an expansion increment to fill a multiple of a full page.

- **Initially allocate one page to SFBUF.**

  Any elements (where an element consists of a buffer, whose size is rounded to a multiple of 8 bytes, plus a 16-byte header) that are required over a page should be controlled by dynamic expansion.

- **Set CRPLBUF.**

  In environments where many sessions come up simultaneously because of either the VARY LOGON operator command or the LOGAPPL operand on a definition statement, you might need to preallocate enough CRPLBUF buffers to handle the influx of requests.

- **Set LFBUF.**

  LFBUF is used only for EAS values (on APPL definition statements) less than 30. Because this pool is not used often, a *baseno* value of 2 is recommended. If you are using TSO, a buffer from the LFBUF buffer pool is used for each TSO user who is logged on. The pool should be expanded one page at a time.

- **Specify *xpanlim* for the IOBUF pool.**

  xpanlim is useful in constraining the I/O buffer pool from excessive growth, particularly when a logical unit attempts to flood VTAM with requests (HOT I/O). Typically, the logical unit is malfunctioning; however, the logical unit may be functioning correctly but may not have had proper session parameters specified. Specifying *xpanlim* allows VTAM to detect such sessions prior to complete CSA exhaustion and, if the HOTIOTRM Start Option is also specified, terminate such sessions without the system being impacted.

  To set *xpanlim*, issue DISPLAY BFRUSE,BUFFER=IO find the peak usage, and set the limit somewhat above that peak. Because *xpanlim* is specified in units of 1024 (K) bytes, you will need to convert buffers to KB. Use Table 59 on page 698 to determine the number of I/O buffers of a given size that fit on a 4K page. Use the equation **K=(number of buffers/number of buffers per page)*4**.

  Tip: You can modify the *xpanlim* value after VTAM startup using the MODIFY BFRUSE command. Refer to *z/OS Communications Server: SNA Operation* for information on this command.

  Also, bear in mind that in order for the expansion limit to be meaningful, CSA must be available were the expansion limit to be reached and VTAM must not reach any CSA limit value that has been specified. In other words, ensure that you have adequate CSA available in the system and your CSA limits allow VTAM to actually allocate buffers up to your expansion limit.

  Note: In Table 59 on page 698, where it lists a range of buffer sizes for a given buffers per page value, whenever possible use the highest value in the range. Some devices have specific I/O buffer size requirements that may prevent using the highest value.

Table 58 lists the number of buffers per page for the different buffer pools.

Table 58. Number of buffers per page

Buffer pool	Default storage type	Buffers per page
APBUF	Fixed	56
BSBUF	Fixed	14
CRA4	Pageable	1
CRA8	Pageable	NA
CRPLBUF	Pageable	25
LFBUF	Fixed	30
LPBUF	Pageable	2
SFBUF	Fixed	32
SPBUF	Pageable	21
TIBUF	Fixed	6
T1BUF	Fixed	4
T2BUF	Fixed	2
XDBUF	Fixed	5

Table 59. I/O buffer size and number of buffers per page

Number of buffers per page	Range of buffer sizes for MVS (with 4K paging)	Range of buffer sizes for MVS using data encryption facility (with 4K paging)
1	1958–3992	1951–3985
2	1270–1957	1263–1950
3	934–1269	927–1262
4	726–933	719–926
5	590–725	583–718
6	494–589	487–582
7	422–493	415–486
8	358–421	351–414
9	318–357	311–350
10	278–317	271–310
11	256–277	256–270
Note: The buffer size in the table is the size coded on the IOBUF start parameter. For information on the calculations used in arriving at these values, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.		

Dynamic expansion and storage

As buffer pools expand, expansion uses virtual storage that can potentially impact an operating system that has virtual storage constraints. VTAM use of virtual storage can be limited by the CSA start options and the virtual storage allocated to VTAM address space.

All buffer pools are defined in virtual storage above the 16 MB address line in extended virtual storage. Therefore, the size of the buffer pools does not usually impact virtual storage constraints.

You can use the SONLIM start option to limit the amount of fixed I/O buffer storage available for session outage notification messages. This controls VTAM use of fixed (real) storage. The default (60 percent) should be sufficient for most environments. This percentage is calculated from the base number of buffers specified when VTAM is started but does not include buffers from dynamic buffering in its calculations. Therefore, the default value of 60 percent is normally sufficient if you do not preallocate most of your I/O buffers at start time and if you use dynamic buffering. If you preallocate most of your I/O buffers at start time, you should consider overriding the VTAM default with a smaller percentage.

I/O buffers and application program data transfer

Data that is transferred between a VTAM application program and another logical unit in the network passes through VTAM I/O buffers. VTAM temporarily holds the data until an application program requests it or until it can be sent to a logical unit.

Figure 161 on page 699 shows how VTAM uses buffers for input operations.

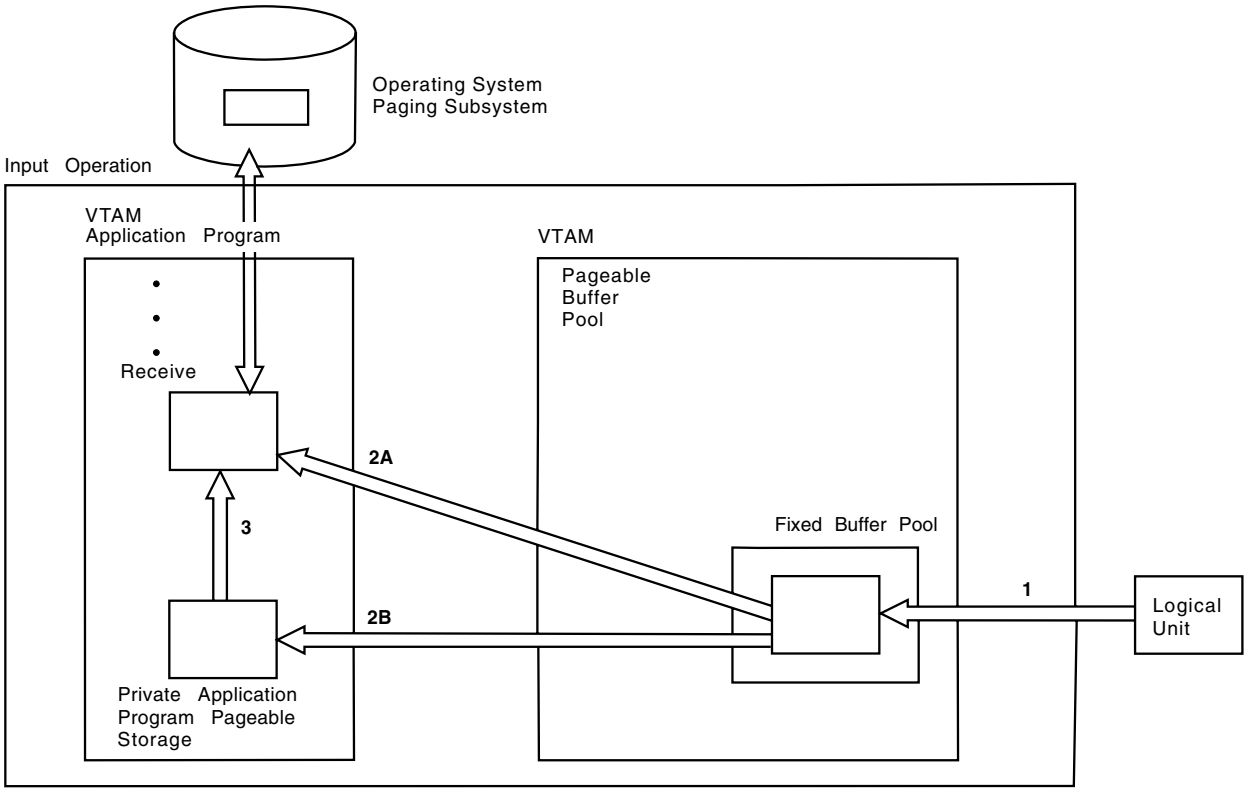


Figure 161. How VTAM uses input buffers

1. A logical unit in the network can send data to an application program independent of the application program requesting the data. The data is placed in a buffer in the fixed buffer pool of data buffers.
2. One of the following occurs:
   a. If the application program has an outstanding request for data from the logical unit, VTAM moves the data from the buffer in the fixed buffer pool to the specified application program data area. Later, data can be paged out of main storage by the operating system.
   b. If the application program does not request data from the logical unit, VTAM places the data in a data space.
3. When the application program issues an input request, the operating system pages the data into main storage (if necessary) and VTAM moves the data to the specified application program data area.

In general, an application program should be written so that at least one request for input is always outstanding. This reduces the chance of VTAM using excessive amounts of an application program's dataspace storage.

Figure 162 on page 700 shows how VTAM uses buffers for output operations.

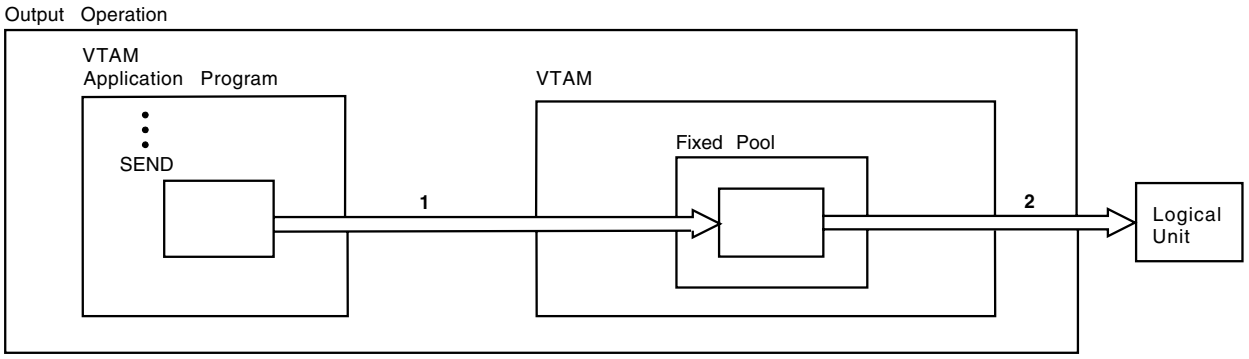


Figure 162. How VTAM uses output buffers.

1. The application program requests data transmission. When VTAM has sufficient buffers in the fixed pool, it moves the data to the fixed pool.
2. VTAM sends the data to the NCP or the logical unit, freeing the buffer.
3. If HPR is used, VTAM holds onto the I/O buffers in pageable storage until the other RTP endpoint acknowledges receipt of the data.

If you notice that your I/O buffers have expanded and you are running out of CSA, it is possible that an application program is not pacing data. Investigate the application programs that have a lot of traffic moving through the I/O buffers to determine whether pacing is being performed. For general information on pacing, see "Session-level pacing" on page 307. For specific information on application program pacing, see "Application program pacing" on page 314.

HOT I/O detection/termination

VTAM has functions to both detect and terminate sessions deemed to be using excessive amounts of the I/O buffer pool. The I/O buffer pool is the storage pool that contains session data. Typically, these sessions are undergoing some sort of malfunction causing an unrestrained transmission of data into VTAM. A failure of VTAM to detect and terminate these out-of-control sessions can result in a VTAM or system outage due to CSA exhaustion. These sessions are commonly referred to as HOT or HOT I/O.

HOT I/O session detection

VTAM scans the entire I/O buffer pool for sessions using excessive I/O buffers under two conditions:

- When a successful expansion of the I/O buffer pool brings the total size of the pool to 80 percent (or more) of the expansion limit specified (in the IOBUF statement). Note that this happens for each expansion in which the size of the pool is greater than the above mentioned threshold.
- When an expansion of the I/O buffer pool is unsuccessful (for any reason). In this case message IST154I is issued containing the expansion failure reason code. Note that if an expansion limit is not specified, then this is the only condition in which a HOT I/O scan is performed.

When one of the above conditions occurs, VTAM scans the entire I/O buffer pool looking for sessions using greater than 10 percent of the current size of the I/O buffer pool. Any session found to be using greater than 10 percent of the pool will have an IST930I message issued for it. This message contains the session partner names as well as the percent of the pool in use by the session.

Specifying an expansion limit for the I/O pool is advantageous because VTAM can detect and correct HOT I/O conditions prior to a complete exhaustion of common service area (CSA). There are disadvantages involved when an incorrect value has been specified. Consequently, it is recommended that the following guidelines be carefully evaluated so that a proper value may be specified.

If the expansion limit has been specified too low, one or more of the following may occur:

- VTAM uses extra CPU cycles doing HOT I/O scans. This primarily occurs when the expansion limit is specified so low that under normal workloads the size of the I/O buffer pool is already greater than 80 percent of the expansion limit. The amount of CPU cycles that VTAM expends doing a HOT I/O scan is dependent on the size of the I/O buffer pool.
- VTAM cannot expand the I/O buffer pool. This occurs when the expansion limit has been specified so low that VTAM cannot expand the I/O pool to accommodate a normal workload increase due to it reaching the expansion limit. VTAM may be adversely affected by a failure to obtain more I/O buffers.

If the expansion limit has been specified too high, one or more of the following may occur:

- VTAM does not detect a HOT I/O condition until after CSA has been exhausted or limited by some other factor (such as the CSALIMIT start option). The expansion limit was specified so high that the size of the pool never reached the 80 percent threshold and consequently VTAM never scanned for HOT I/O until an expansion failed. This works as if there is no expansion limit specified. Consequently, VTAM and the system may be adversely affected by this condition.
- A VTAM or system outage occurs due to an exhaustion of CSA. This primarily occurs when VTAM CSA use is not also limited with the CSALIMIT parameter. If VTAM is not limited in the amount of CSA it can acquire and a HOT I/O situation occurs, VTAM may use so much of CSA that the system (GCS or MVS) is adversely affected.

Tip: If the expansion limit is found to be too high or too low, you can modify the *xpanlim* value using the MODIFY BFRUSE command. Refer to *z/OS Communications Server: SNA Operation*.

HOT I/O session termination

The HOTIOTRM start option allows the installation to set the threshold at which VTAM will terminate "hot" (out-of-control) sessions. HOTIOTRM specifies the percentage (10–99) of the current size of the I/O buffer pool that a single session must have allocated to it to cause VTAM to take corrective action. If HOTIOTRM is allowed to default to 0, VTAM will not terminate sessions based on I/O buffer pool usage.

If HOTIOTRM is specified with a value in the range 10–99, and 1 or more sessions have been identified by VTAM as using a percentage of the I/O buffer pool greater than or equal to that value, VTAM will then attempt to stop the buffer pool expansion with one of the following actions:

- Deactivate the LU when the session type is SSCP-LU.
- Deactivate the PU when the session type is SSCP-PU.
- Terminate all sessions between the two logical units when the session type is LU-LU.

The above action will be taken if all of the following criteria are met:

- The percentage of the pool in use by the session is greater than or equal to the HOT I/O session termination threshold as specified by the HOTIOTRM start option.
- The subarea of one or both session partners must be the subarea in which the HOT I/O is being detected.
- When the session type is SSCP-PU, the PU is not an NCP.

If all of the above conditions are true and it is an LU-LU session, VTAM terminates all sessions between the two session partners. It then issues message IST1099I, indicating that the sessions have been terminated.

If all of the above conditions are true and it is a SSCP-LU or SSCP-PU session, VTAM deactivates the LU or PU. It then issues message IST1098I, indicating that the LU or PU has been deactivated.

Controlling NCP slowdown

To alleviate the slowdown condition, you can take one or more of the following actions:

- Select a buffer size that makes more efficient use of NCP buffers.
- Install more memory in the communication controller.
- Transfer inbound data to the CPU more quickly by decreasing DELAY.
- Transfer outbound data to the network more quickly by increasing line speed.

  Note: Many systems have much more outbound data than inbound. Therefore, increasing line speed can cause more data to be transferred out of the NCP through the line. This reduces buffer use.
- If the NCP slowdown can be attributed to any given session or sessions, session pacing can be used to slow down the sessions causing the congestion.

NCP clears slowdown conditions by accepting outbound data from the CPU more slowly and by accepting data from the network more slowly. It stops polling and requests the host to stop writing data. Slowdowns are reduced when a device is not polled, if a PIU READ puts the NCP in a slowdown condition. The value specified for MAXDATA can control the speed with which data is placed on intermediate routing node (IRN) links.

The following actions can be taken to alleviate slowdown for local SNA controllers:

- Increase the value of MAXBFRU. This allows the controller to send more data on every read channel program.
- Use session pacing to control the amount of data VTAM sends to the controller. Specify lower pacing values for primary-to-secondary pacing. This limits the number of PIUs that are sent to the controller by VTAM.

Maximizing coattailing

To maximize coattailing, you need to transfer more data either inbound to the host or outbound without generating an attention interrupt. If more messages are transferred in or out of the host than the number of READ or WRITE channel programs issued, coattailing is taking place. The effect is to transfer more data in a single channel I/O operation. Coattailing can provide greater throughput between a host and an SNA controller; however, response time can increase.

The amount of coattailing that you can achieve is directly related to the message traffic in your network. However, you can influence the amount of data that is available for a data transfer operation.

Coattailing allows more PIUs to be queued for transmission before a channel program is issued. This entails a trade-off between response time and host/controller cycles. Specifying a large value for coattailing delay usually results in a larger number of PIUs being accumulated before a channel program is issued. However, PIUs will experience a longer wait time, and consequently, the overall response time might increase. The benefits of specifying a coattailing delay are dependent on the volume, distribution, and predominant direction (if any) of traffic flow.

The coattailing delay is implemented based on PIU arrivals. The arrival of a PIU causes VTAM to compute the elapsed time since the beginning of the delay interval and determine if the coattailing delay has been exceeded. This has implications for the benefits of coattailing:

- At low traffic volumes, PIU arrivals might be infrequent. This could cause a PIU to stay in the queue until the next PIU arrives or the backup three-second timer expires. Setting a nonzero coattailing delay could delay transmission of a PIU up to three seconds. The coattailing delay should be set to 0 for low traffic volumes.
- At high traffic volumes, sufficient numbers of PIUs are queued while a previous channel program is running. This results in "automatic" coattailing. Specifying a coattailing value greater than 0 might not result in greater throughput or reduced host processing cycles.
- Coattailing can be beneficial at high processor utilization with moderate traffic volumes. In this case, tuning the coattailing delay parameters might result in reducing the impact on host processor cycles and provide improved throughput.
- For predominantly unidirectional data traffic, specifying a coattailing delay for traffic in the nonpredominant direction might result in larger overall delays and reduced throughput. For example, file transfers with application level acknowledgments for segments of files transferred might see lower throughputs with a nonzero coattailing value since the acknowledgment PIUs can be delayed for up to three seconds if no other PIU triggers the coattailing timer.

In general, transaction processing applications using 3270 data streams are characterized by bidirectional traffic with relatively small messages. In this case, a nonzero value of coattailing delay might be beneficial at some traffic volumes. Newer client-server applications are usually characterized by fewer and larger messages. These applications will see minimal benefit with nonzero coattailing delays.

Controlling outbound coattailing

By controlling the channel delay timer for VTAM and the attention interrupt delay timer, you can affect outbound coattailing in your system.

VTAM saves PIUs for transfer over a channel until one of the following events occurs:

- Data flowing at transmission priority two (TP2) is available.
- A virtual route pacing response must be transmitted.
- An attention interrupt from an NCP or an SNA cluster controller (for example, the 3174) occurs.
- The PIU queue for VTAM reaches its limit.

For each VTAM channel delay interval, VTAM calculates the queue depth (QDPTH tuning statistic) by multiplying the number of PIUs transferred during the channel delay interval by a percentage (currently 75 percent).

- The channel delay timer or the backup three-second timer expires.

Controlling inbound coattailing

To affect inbound coattailing, delay the VTAM WRITE operation. This is effective because VTAM normally reads data after transferring data to the SNA controller or to another VTAM. By delaying VTAM longer before it performs a WRITE operation, more data can potentially accumulate in the SNA controller or other VTAM host, which can be read in the same I/O operation as the WRITE channel program. You can control the time interval during which VTAM buffers low priority data before transferring it over the channel to another VTAM, NCP, or SNA cluster controller by coding the DELAY operand on the appropriate VTAM definition statement.

DELAY operand

When defining a channel, consider the effects of the DELAY operand on the LINE definition statement. This operand specifies the maximum amount of time that VTAM queues an outbound PIU with transmission priority 0 or 1 before transmitting it over the channel. When the PIU is eventually sent, it and the queued PIUs following it are sent as a block.

When you specify DELAY=0, VTAM sends each PIU over the channel as it arrives on the queue. When a nonzero value for DELAY is specified, the blocking of low-priority PIUs can save host processor instructions. However, a high channel delay time can cause increased response time because some of the PIUs are delayed before being sent.

Occasional bursts affect throughput only for the next second or two. However, a continuing series of bursts can severely affect capacity if there is little nonchannel-to-channel activity. A few sessions of interactive traffic are more likely to accentuate this condition than a large number, when traffic on the channel is likely to be more randomly distributed.

You might want to increase the DELAY operand value over that of the default to do the following:

- Decrease host processor overhead when response time is less important.
- Take advantage of a smoothing effect on the queuing that can occur with larger values, an effect that can be useful for application programs with sessions conducted in bursts.

You can specify different DELAY values for different channel-attached devices. The following are general guidelines for picking an appropriate DELAY value for a particular channel-attached device.

If you want to experiment with DELAY values greater than the default, you can activate in succession a series of major nodes with increasing values beyond the default.

MAXBFRU operand

For channel-to-channel connections, the MAXBFRU operand defines a constant sized buffer in increments of 4K pages. The value does not fluctuate throughout the life of the connection. This buffer must be defined as large as the largest PIU to cross the channel-to-channel connection. Also, if a nonzero DELAY value is chosen

and you want to transfer multiple RUs in a single I/O operation, the MAXBFRU value must also be large enough to handle multiple RUs.

After the data transfer operation is complete, the PIUs are deblocked into VTAM buffers as defined in the IOBUF start option. The VTAM buffers are allocated in 4K increments, so you might have to allocate more I/O buffers. If you are using dynamic buffer expansion, you should monitor the I/O buffers pool to ensure that there is not excessive buffer pool expansion and contraction because of these channel operations.

The maximum PIU size across a connection is defined by MAXBFRU times IOBUF size. For channel-attached cluster controllers, the maximum PIU size is defined by MAXBFRU times (IOBUF size minus 16).

If all LU-LU sessions end in a physical unit that provides segment assembly, path information unit (PIU) segmentation occurs across the link. If any LU-LU sessions end in a physical unit that does not provide segment assembly, the maximum PIU size must be large enough to handle RU sizes for these sessions. If the LU-LU session cannot have segments, either increase the value of the MAXBFRU operand or the I/O buffer sizes, or adjust the RU size in the MODEENT macroinstruction for the logical unit.

Coattailing for SNA controllers

The amount of coattailing of messages that occurs between a host and a channel-attached NCP or SNA controller is affected by the following:

- I/O buffer size
- The number of buffers allocated to receive incoming messages at the host
- The amount of time that VTAM and the channel-attached device allow to elapse before sending buffered messages

For NCP Version 4 Release 2 and later releases, the UNITSZ value specified in the NCP HOST definition statement is no longer used. If coded, the value is ignored. The value of VTAM IOBUF size is determined by NCP during XID exchange. For NCP releases prior to 4.2, the value of UNITSZ can affect VTAM, NCP, and network performance. This section provides guidelines for specifying this value.

The initial amount of storage you allocate for the I/O buffer, as well as the *xpanno* value you specify, depends on the *bufsize* specification. This, in turn, depends on the UNITSZ value specified in both the NCP HOST definition statement and the buffer pool start option for the I/O buffer. In the case of channel-attached cluster controllers, *bufsize* depends only on the buffer pool start option for the I/O buffer. The value in the VTAM start option is used by both VTAM and the NCP when they exchange information. The *bufsize* value entered as part of the buffer pool start option for the I/O buffer is interpreted by VTAM as the UNITSZ value used, except for channel-attached cluster controllers, where UNITSZ is IOBUF minus 16. The I/O buffer size, as it appears on the console after a DISPLAY BFRUSE operator command, is greater than the *bufsize* that was specified in the buffer pool start option because VTAM adds control information to each I/O buffer.

Guidelines for setting DELAY

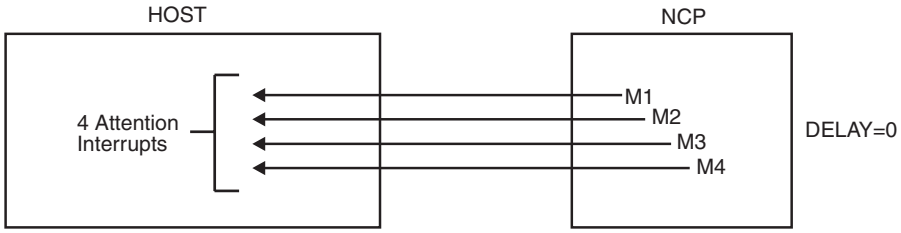
To specify the amount of time that elapses before VTAM or NCP sends buffered data, use the DELAY operand on NCP and VTAM definition statements. The DELAY operand on the NCP BUILD definition statement controls how long the NCP buffers data; it is the elapsed time between the receipt of the first inbound

message and the presentation of an attention interrupt to the host. This capability can affect your strategy for increasing coattailing use of the channel.

The DELAY operand that controls how long VTAM buffers data can be specified on the following:

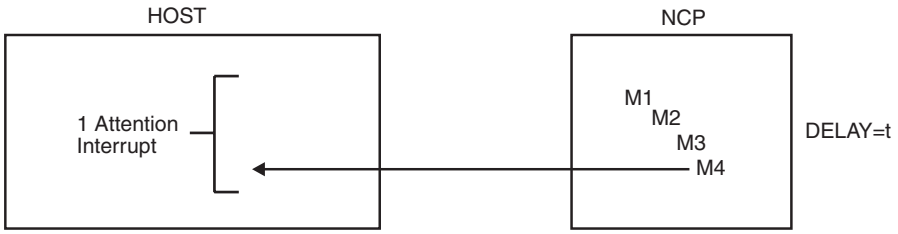
- PCCU definition statement for NCPs not defined in a channel-attachment major node
- GROUP, LINE, or PU definition statements for NCPs defined in a channel-attachment major node
- PU definition statement for local (channel-attached) SNA controllers

The three examples in Figure 163 illustrate the use of the DELAY operand.



In this example, four messages have come in over a period of time. Because DELAY is 0, each message has resulted in an attention interrupt being presented to the host. In all, four attention interrupts have been presented, and no coattailing has taken place.

Figure 163. Effect of DELAY time on coattailing (Part 1 of 3)



In this example, DELAY is equal to *t*. This time period is large enough to allow four messages to arrive at the NCP. The time *t* has expired, and the NCP presents an attention interrupt. VTAM reads all four messages, having received only one attention interrupt.

Figure 163. Effect of DELAY time on coattailing (Part 2 of 3)

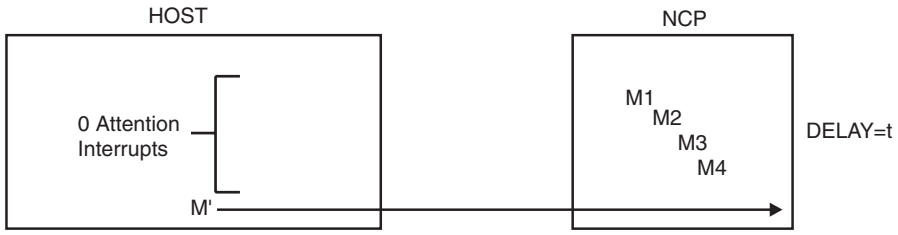


Figure 163. Effect of DELAY time on coattailing (Part 3 of 3)

For NCP, the DELAY operand allows time for more than one message to arrive in the NCP before an attention is presented to the host.

The following events cause PIUs on the queue to be sent before the DELAY time elapses:

- VTAM receives an attention interrupt from NCP.
- VTAM or NCP reaches the MAXBFRU limit.
- VTAM sends virtual route pacing responses.
- VTAM sends transmission priority two traffic.
- VTAM sends APPN network priority or highest priority traffic (to a local SNA PU).

The default value for the DELAY operand is 0 (for example, no coattailing). Take the default under the following conditions:

- You have a low rate of transactions that use the channel connection, and response time for interactive transactions is important.
- You have a high rate of transactions that use the channel connection, and response time for interactive transactions is important.
- You have predominantly unidirectional traffic (file transfer) with little other activity in the opposite direction.
- You are benchmarking single or few sessions of bulk data traffic (file transfers, backups, and so on).
- You have channel traffic with frequent bursts of SENDs (such as from an application program that makes inquiries of the other host in rapid succession).

To analyze inbound coattailing for channel-attached SNA communication or cluster controllers, compare the number of times that NCP signals VTAM that it has data to send (ATTN tuning statistic) to the number of times that VTAM issues a channel READ program to read data (CHRD or RCH tuning statistic). A less accurate method is to divide the number of inbound PIUs transferred by NCP (IPIU tuning statistic) by the number of read channel programs issued by VTAM (CHRD or RCH). This calculation gives you the average number of PIUs read from the channel-attached device with each channel program, which is an indicator of coattailing.

Note: For a 3274 to clear its buffers after a data transfer operation, the 3274 signals to VTAM that it has more data to send even though there is no data. This causes the RDATN tuning statistic to equal the CHRD or RCH tuning statistic, and because RDATN is included in the ATTN tuning statistic, the ATTN value is normally double the CHRD or RCH value.

To analyze outbound coattailing for channel-attached SNA communication or cluster controllers, divide the number of outbound PIUs (OPIU tuning statistic) by the number of WRITE channel programs VTAM issues (CHWR or WCH tuning statistic). This calculation gives you the average number of PIUs transferred by VTAM with each channel write program, which is a rough estimate of the effectiveness of coattailing.

Note: The DELAY specification for a 3174 is a customization option. However, even if you set the DELAY customization option for a 3174 that is performing a token-ring gateway function, the option is ignored. The DELAY operand in VTAM should also be set to 0 for a 3174 that is a token-ring gateway channel-attached cluster controller.

Specifying a nonzero attention DELAY value to activate attention DELAY does not automatically ensure coattailing. To ensure coattailing, all of the following must be true:

- The installation must have a transaction rate high enough to allow more than one message to accumulate in the DELAY time period in the NCP or the host.
- The MAXBFRU value must be large enough to allow more than one message to be sent to the host.
- The host processing speed has an effect on coattailing; the faster host processors reduce coattailing slightly. An attention interrupt occurs when MAXBFRU is reached or when the DELAY time period expires.

Begin by setting DELAY equal to or greater than 0.2. A 0.2-second delay has little effect on response time, but if the traffic speed is approximately 15 transactions per second or greater, coattailing occurs.

Setting the MAXBFRU operand

The MAXBUF operand allows you to specify the number of VTAM I/O buffers allocated for inbound data transfer. You need to set MAXBFRU on both the NCP HOST definition statement and the VTAM GROUP or LINE definition statements for channel-attached NCPs or SNA controllers. When VTAM creates a read channel program, it must always have MAXBFRU number of I/O buffers available for reading from the channel-attached device. VTAM always attempts to hold in reserve MAXBFRU number of I/O buffers for the next read from the channel-attached device.

VTAM and NCP do not support segmenting PIUs across channel connections between them. Therefore, ensure MAXBFRU times IOBUF size accommodates the largest PIU that NCP can send to VTAM. NCP informs VTAM the maximum size PIU that VTAM can send in its XID2 (this is based on TRANSFR*BFRS-18). VTAM also honors the MAXDATA keyword coded on the PCCU definition and will send PIUs to NCP no larger than the smaller of the MAXDATA keyword or what VTAM received from NCP in the XID2. Therefore, you should ensure that MAXDATA and TRANSFER*BFRS-18 is larger than the largest PIU VTAM can send to NCP.

In addition, choose MAXBFRU so that more messages can be coattailed. The MAXBFRU value determines when an attention interrupt is signaled to the host.

MAXBFRU=5 is a good starting point. There should be enough buffers available to transfer all the messages likely to arrive in the DELAY time period. (The use of MAXBFRU=5 assumes one inbound message fits in one I/O buffer. More details on I/O buffer size are given in the following paragraphs.) You can use tuning statistics to verify that MAXBFRU is sufficient for coattailing. Using the example in Figure 166 on page 710, for instance, MAXBFRU could have been set to 2. Instead, it was set to 6 to maximize coattailing.

MAXBUFU formulas: The number of buffers required for a message (RU) is determined by the following formula:

```
(29 + RU size) ÷ bufsize
```

Note: This formula applies to FID4 (subarea connections).

The number 29 is obtained by adding the size of the transmission header (26) to the size of the request/response header (3).

For channel-attached cluster controllers, the number of buffers required for a message is determined by the following formula:

```
(13 + RU size) ÷ (bufsize - 16)
```

The number 13 is the sum of the channel link header (4) plus the transmission header (6) plus the request/response header (3).

The *bufsize* is the buffer pool value you use to start VTAM. If you are unsure about this value, look at your start options for the IOBUF buffer size specification.

The MAXBFRU value you use must be greater (for NCPs and PU2.0 channel attached devices) than or equal to the number of buffers required for the largest message (RU) your system receives. For T2.1 channel-attached devices, the value should be greater, but does not have to be. This minimizes the number of attention interrupts needed to process large messages. Use the preceding formula to calculate the lower limit for MAXBFRU, given the size of the largest message.

Guidelines for setting I/O buffer size

Choosing the ideal I/O buffer size helps minimize host storage requirements and host processor usage.

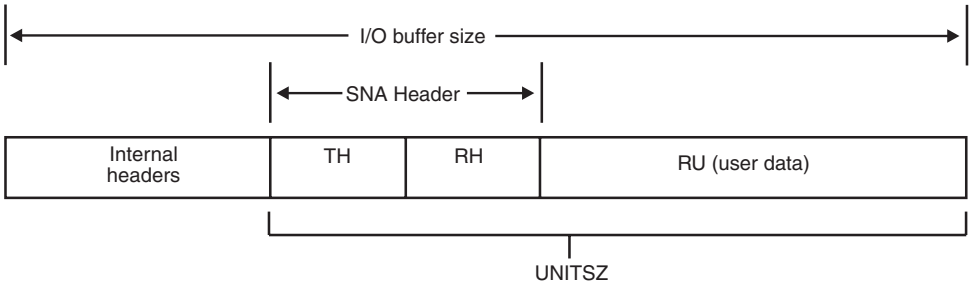
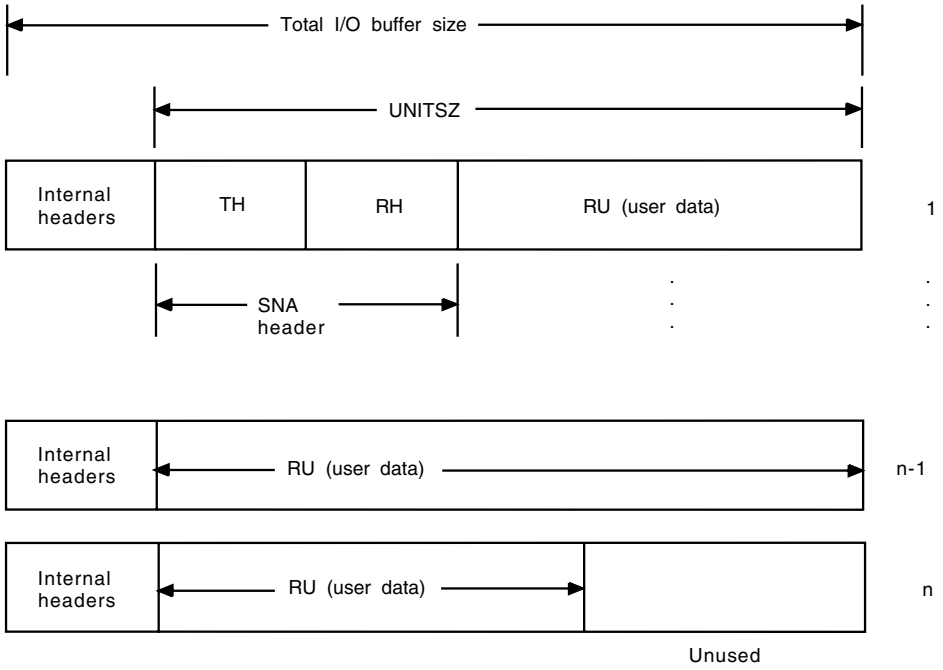


Figure 164. General I/O buffer format

Figure 164 illustrates UNITSZ. UNITSZ is not the size of an I/O buffer; it is a portion of the buffer size. It includes an SNA header field. In the remaining discussion, the I/O buffer size is referred to as the UNITSZ. (Note that one buffer size is used for both inbound and outbound data transfers.)

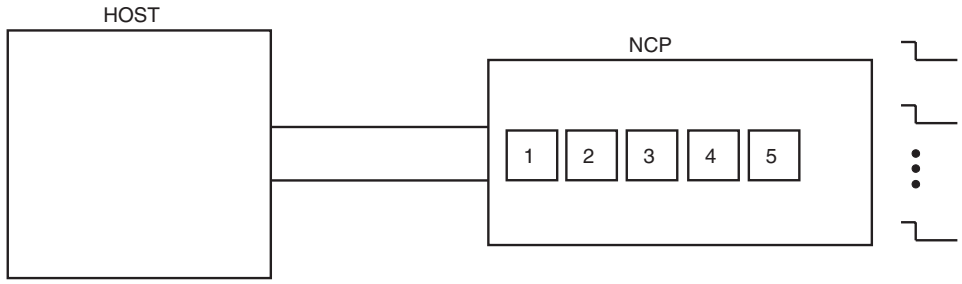


Note: Multiple I/O buffers are used where the length of outbound data is greater than UNITSZ minus the SNA header.

Note: A similar concept applies to inbound data.

Figure 165. Using multiple I/O buffers to transfer single message

Figure 165 illustrates how a message that is larger than UNITSZ spans more than one buffer. The message can be greater than UNITSZ. If it is, VTAM uses more than one buffer to contain the message. The unused space in the last buffer is not allocated to the next message.



MAXBFRU=6,DELAY=t, five messages arriving; each message needs two buffers.

1 ◄—— I/O attention interrupt

2 ◄—— Three messages transferred

3 ◄—— Attention bit set

4 ◄—— Two messages transferred

Figure 166. Multiple-buffer considerations

Figure 166 assumes that MAXBFRU=6 and DELAY=*t* and that there are five messages arriving at the NCP before the time *t* expires. However, it also assumes

that each message requires two I/O buffers. After three messages are received, the MAXBFRU limit is reached, and the NCP signals the host with an attention interrupt. The three messages are transferred, but two more remain. Therefore, the NCP signals VTAM that there is more data to read (RDATN tuning statistic). If the RDATN tuning statistic is high, NCP has more data to transfer to VTAM than VTAM has allocated buffers to process. Therefore, you can increase the MAXBFRU specification, thus allocating more VTAM buffers to reduce this condition.

However, coattailing has taken place; two messages have been transferred without having generated an attention interrupt. VTAM creates another channel program to read the remaining messages. Again, coattailing has occurred, because two messages have been transferred and only one attention interrupt has occurred. However, two I/O operations have been used where the data transfer might have been accomplished with only one. If UNITSZ were large enough to contain an entire message or if MAXBFRU were equal to 10, the extra I/O operation could have been avoided.

Guidelines for setting UNITSZ

To choose the best UNITSZ value, first determine the average size of inbound and outbound messages. Then, compare them to each other, and set the UNITSZ equal to the larger of the inbound or outbound message sizes. For channel-attached cluster controllers add 16.

Note: The message size is not determined by VTAM. You can determine the average message size by examining your application programs (for example, CICS and IMS). You can also use the NetView Performance Monitor to analyze the average message size for your network.

Suppose the inbound size is much smaller than the outbound size. Choose a UNITSZ at least large enough to contain one complete inbound message. However, the UNITSZ should be larger than the inbound message so that relatively few buffers are needed for the outbound message. Because the extra buffers increase path length, it is better not to specify too many buffers.

As another example, consider the case where the inbound message size is much greater than the outbound size. The process is similar to the previous one, except that the outbound message is the guide for choosing UNITSZ. Choose a UNITSZ that is greater than or equal to the outbound message size so that relatively few buffers are needed to contain one inbound message.

Where practical, choose UNITSZ so that no more than five to seven buffers are needed for transferring one message, whether it is inbound or outbound.

Coattailing for channel-to-channel operations

VTAM channel-to-channel I/O operations, which are similar to SNA controller operations, are impacted by factors such as VTAM I/O buffer size and the channel delay and maximum number of buffer pages on the CTC definitions. The following topics discuss variables that can affect VTAM channel-to-channel performance.

Guidelines for setting DELAY

To select a DELAY operand value, use the following procedure:

1. During periods of the day when it is most likely that you are experiencing performance problems, turn on the tuning statistics. For channel-to-channel attachments, turn on tuning statistics at each processor using the following operator command:

```
F NET,TNSTAT,CNSL,TIME=1
```

2. For each recording of the tuning statistics, check the amount of coattailing that is occurring. For channel-to-channel connections, compare the TIMERS and CHNRM values of the channel-attachment major node associated with each host processor. The desirable TIMERS value is 0, but an occasional nonzero value is acceptable.
3. For channel-to-channel attachments, if the TIMERS value in any tuning statistics record is too large, deactivate the channel-attachment major node at each host processor, and activate a previously defined major node in which DELAY=0 has been specified on the LINE definition statement.

   For other channel attachments, if the average number of PIUs seems too high (thereby indicating possible response time problems), you can activate alternate definitions with a lower DELAY value.

The following events cause PIUs on the queue to be sent before the DELAY time elapses:

- VTAM reaches the MAXBFRU limit.
- VTAM sends virtual route pacing responses.
- VTAM sends transmission priority two traffic.
- VTAM reaches the QDPTH value.

The default value for the DELAY operand is 0 (for example, no coattailing). Take the default under the following conditions:

- A low rate of transactions that use the channel-to-channel connection. A rate of under five transactions per second is considered low.
- Channel-to-channel traffic with frequent bursts of SENDs, such as from an application program that makes inquiries of the other host in rapid succession.

You can also analyze the average number of bytes transferred per I/O operation by dividing the total number of bytes transferred (RDBUF) by the number of READ channel programs (CHNRM).

Guidelines for setting I/O buffer size

Because data is transferred in the channel-to-channel buffers, increasing or decreasing the I/O buffer size has no effect on the channel program size.

Virtual route window sizes

The default minimum window size might be too small for a VTAM channel-to-channel virtual route. VTAM increases the window size only when the virtual route pacing response is not returned fast enough to prevent one window worth of PIUs from being queued up on the virtual route (a HELD condition). A channel-to-channel route is so fast that the virtual route pacing responses are turned around fast enough to prevent HELD conditions from occurring very often. Because of this, the current virtual route window size tends not to increase, but to stay very close to the minimum VR window size.

The smaller the VR window size, the more virtual route pacing responses flow over the channel. This can cause higher CPU utilization than necessary.

Setting a higher minimum window size reduces the number of virtual route pacing responses and helps CPU utilization and throughput. For channel-to-channel one-hop routes, the default minimum VR window size is 1. The recommended value is 15. The recommended maximum virtual route pacing window size value is 50 for this environment.

If VTAM is expanding its IOBUF buffer pool to service a virtual route, the lack of VTAM buffers is considered major congestion for the route. Therefore, VTAM decreases the virtual route window size to the minimum for all VRs that it services. You should monitor this VTAM buffer pool to ensure that there is not excessive expansion and contraction if you are using dynamic buffering.

You can also use VTAM tuning statistics to analyze a virtual route impact on the channel-to-channel connection. A virtual route that uses transmission priority two and traverses the channel-to-channel connection causes VTAM to immediately schedule the data transfer operation. A virtual route pacing response is also high-priority traffic, which has the same effect. The PRI tuning statistic in VTAM indicates the number of times that a VTAM channel program is started to transfer this high priority data (TP2 or VR pacing response) to the other VTAM host. If this number is high and the channel-to-channel connection is not used extensively for TP2 traffic, the minimum virtual route pacing window size is probably too small. Also, the higher this number is in relation to the sum of TIMERS, QDPTH, and BUFCAP, the less outbound coattailing occurs.

For more information about pacing, see "Session-level pacing" on page 307.

Session-level pacing tuning considerations

Usage of the following storage areas affects the efficiency of pacing in the network:
- Input/Output buffer
- Application program data space
- Common storage manager

When storage usage hits a constraint level (70 percent usage) or critical level (90 percent usage), VTAM takes an appropriate action to reduce the storage usage. Constraint and critical levels are in part determined by user specifications and vary for each of the listed storage areas.

Input/output buffers

The IOBUF start option defines the amount of storage available for I/O buffers. The XPANLIM parameter is used to increase storage available for I/O buffers when reaching the constraint level.

For a detail description of the XPANLIM parameter, see "Guidelines for dynamic expansion" on page 695.

Application program data space

VTAM uses the value of the ASRCVLM operand coded on the APPL definition statement in calculating the constraint and critical levels for application program data space. Selecting a value for the ASRCVLM operand should reflect the high-usage periods of the data space storage.

To determine current usage value for the application program data space, issue a DISPLAY ID operator command, specifying the application program name. Message IST1633I contains the value specified for the ASRCVLM operand and message IST1634I contains the address space usage.

CSM storage

Threshold values for constraint and critical levels of CSM storage is defined by the IVTPRM00 parmlib member. Refer to *z/OS Communications Server: New Function*

Summary for additional information on specifying tuning values for CSM storage.

Appendix A. TSO/VTAM

TSO/VTAM provides the capability of using TSO through VTAM. TSO is a standard feature in MVS that provides conversational time sharing.

TSO/VTAM supports the following terminals:

- IBM 3270 Information Display System
- IBM 3290 Information Display System
- IBM 3767 Communication Terminal
- IBM 3770 Data Communication System
- IBM 5550 (as 3270) single-byte character set (SBCS) and double-byte character set (DBCS) modes
- IBM 8775 Display Terminal
- IBM Displaywriter

TSO/VTAM can also support the following non-SNA terminals in conjunction with the IBM Network Terminal Option licensed program:

- CPT-TWX Terminal
- IBM 2741 Communication Terminal
- World Trade Telegraph (WTTY)

The basic elements of TSO/VTAM are the terminal control address space (TCAS) and the VTAM terminal I/O coordinator (VTIOC). TCAS accepts logons from TSO/VTAM users and creates an address space for each user. VTIOC is the interface between TSO and VTAM; it coordinates data flow. TCAS and VTIOC, together with existing TSO components such as the LOGON scheduler, the application programs, the terminal monitor program, and the TSO service routines, make up a TSO/VTAM time-sharing system.

The user logon to the TCAS address space is passed to the TSO user address space without initially establishing a session between the TCAS address space and the TSO terminal. Instead, the session is established between the TSO terminal and the TSO user address space, avoiding session establishment and termination with the TCAS address space and avoiding your receiving message USSMSG7.

TSO/VTAM and TSO through VTAM can reside in the same host processor.

Note: z/OS limits the number of concurrent TSO/E sessions to eight.

Defining TSO to VTAM

To use TSO/VTAM with VTAM, define the following to VTAM:

- TCAS and each TSO user
- TSO/VTAM session parameters

For compatible logons, you also need to create and define to VTAM an interpret table.

© Copyright IBM Corp. 1984, 2011

Defining the TCAS application to VTAM

An APPL definition statement defines an application program to VTAM. Because TCAS and each TSO user are VTAM application programs, you must code APPL definition statements for them and put the definition statements in SYS1.VTAMLST.

You can reduce the number of operands you need to code on the APPL definition statements for each TSO user by using the GROUP definition statement under the APPL major node. All of the operands on the APPL definition statement except ACBNAME can be coded on the GROUP definition statement and allowed to sift down. You only need to code the unique name on the APPL definition statements for the TSO users if you use the GROUP definition statement to specify all the other operands.

Note: To avoid obtaining unnecessary CSA storage for TSO users, you should define TSO application programs to VTAM with EAS=1; do not use the default. TSO/VTAM obtains CSA storage independently of VTAM CSALIMIT.

Single-domain network

In a single-domain network in which there are no overriding network naming conventions, you can use the following technique to code the APPL definition statements for TCAS and for the TSO users:

Code the following APPL definition statement for TCAS:

```
tsoa APPL ACBNAME=TSO,PRTCT=password,
          AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

The label *tsoa* is a unique network name. This name can be up to eight characters in length, but should be kept to less than eight so that it can be a prefix for the names on the subordinate TSO APPL definition statements. Code as many APPL definition statements, in the following format, as there will be users logged on to TSO/VTAM at one time:

```
TSOannnn APPL  ACBNAME=TSOnnnn,PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

The label *tsoannnn* is a unique network name, which can be up to eight characters in length. It can be helpful to use the network name for TCAS as a prefix in the network name when coding APPL definition statements for each terminal logged on to the TSO/VTAM in this VTAM host concurrently. For the *nnnn* suffix, use sequential decimal integers starting with 0001. For the ACBNAME, the *nnnn* suffix is a decimal integer; the numbering must start with 0001 and be sequential.

Note: It is recommended that session pacing should not be used; NVPACE should be coded. However, in situations where pacing is necessary, code a high value, such as VPACING=20.

You need to use the same *password* for TCAS and each TSO/VTAM user. You must code a different application program name, in the form TSO*nnnn*, for each user. The *nnnn* suffix is a decimal integer; the numbering must start with 0001 and must be sequential. NOACQ and NOPO need not be coded; they are default values.

Multiple-domain network

For TSO sessions to be established in a multiple-domain network, you must code CDRSC definition statements for each TSO/VTAM application program unless you specify SSCPDYN=YES. Code the CDRSC definition statements in every VTAM host owning logical units that will log on to the application program.

In the VTAM host containing the TSO subsystem, it is not necessary to define the LUs in the other domains that issue logons to TSO/VTAM if dynamic CDRSC definition is authorized. Also, the manager in the domain of the SLU can be authorized to create CDRSCs for the TSO application program associated with the TSO user.

Code the following CDRM definition statements in each domain to avoid defining both the SLUs that log on to TSO/VTAM as CDRSCs in the TSO/VTAM domain and the TSO user application programs as CDRSCs in the SLU domain.

```
name1    CDRM  CDRDYN=YES,CDRSC=OPT,...

name2    CDRM  CDRDYN=YES,CDRSC=OPT,...
```

The CDRDYN operand determines whether a CDRM is authorized to dynamically create a CDRSC representing a cross-domain LU when a logon request is received from the LU (through an Initiate from the CDRM managing the LU).

The CDRSC operand determines whether the dynamic creation of CDRSC definitions is permissible when a logon request is received from the CDRSC manager identified by this CDRM definition statement.

In a multiple-domain network where TSO/VTAM is run in more than one domain, you must have special definition statements with unique names for TCAS and the terminals in your domain and interacting domains. Code the following APPL definition statement for TCAS in your domain:

```
tsoa     APPL  ACBNAME=TSO,PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,
               TSO,NOPO),EAS=1
```

The label *tsoa* is a unique network name. This name can be up to eight characters in length, but should be kept to less than eight so that it can be a prefix for the names on the subordinate TSO APPL definition statements.

Code as many APPL definition statements in the following format as the maximum number of sessions that will be established with TSO/VTAM in your domain at one time:

```
tsoannnn APPL  ACBNAME=TSOnnnn,PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,
               TSO,NOPO),EAS=1
```

The label *tsoannnn* is a unique network name, which can be up to eight characters in length. It can be helpful to use the network name for TCAS as a prefix in the network name when coding APPL definition statements for each terminal logged on to the TSO/VTAM in this VTAM host concurrently. For the *nnnn* suffix, use sequential decimal integers starting with 0001. For the ACBNAME, the *nnnn* suffix is a decimal integer; the numbering must start with 0001 and be sequential.

Code the following definition statement for each TCAS in another domain with which an LU in your domain communicates:

```
tsob     CDRSC  CDRM=name of VTAM manager for tsob
```

If a dynamic cross-domain resource definition is not authorized, code the following CDRSC definition statement in each domain that contains an SLU that can communicate with your domain, and code a CDRSC definition statement in the domain of TSO/VTAM for each cross-domain SLU that can log on to TSO/VTAM:

```
tsobnnnn CDRSC  CDRM=name of VTAM manager for tsobnnnn
```

tsob identifies the owning CDRM for each domain and must be included on each TCAS and CDRSC definition statement. The *nnnn* suffix is a decimal integer; the numbering must start with 0001 and be sequential.

Using model application program definitions for TSO/VTAM application programs

You can reduce the number of APPL definition statements for terminals that can be logged on to TSO/VTAM by using wildcard characters to define model application program definitions. See "Model application program definitions" on page 374 for a full explanation of model application program definitions.

The following example shows how model application program definitions are used for TSO/VTAM application programs.

The following APPL definition statement defines TCAS.

```
tsoa     APPL  ACBNAME=TSO,PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

The following APPL definition statement defines a model that is used to define the TSO address space as terminals log on to TSO/VTAM.

```
tsoa*    APPL  ACBNAME=TSO*,PRTCT=password,MODSRCH=FIRST,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

Requirement: The label in these definitions must be 4 characters or less in length (without the *). In the previous examples, the labels *tsoa* and *tsoa** are 4 characters in length (without the *).

As each terminal logs on to TSO/VTAM, the * in TSO* on the ACBNAME operand and in the label tsoa* will be replaced with the appropriate sequential 4-digit decimal integer; for example, 0001, 0002, and so forth. You do not need to code separate APPL definition statements for each terminal that can be logged on to TSO/VTAM concurrently. You need only the one model application program definition.

Notes:

1. If you plan to use model application program definitions for TSO/VTAM, then you must code MODSRCH=FIRST on the TSO application definition statement that represents the subordinate TSO APPLs.
2. With wildcard characters, you cannot limit the number of TSO address spaces created by limiting the number of application definitions you code. However, you can use the USERMAX operand in the TSOKEY*xx* parmlib member for this purpose.

Using BASENAME to define TSO/VTAM applications programs

The BASENAME operand, coded in the TSOKEY*xx* parmlib member, can be used to specify up to four characters for the "base" part of the TSO/VTAM application program name. The default value for BASENAME is TSO.

Using BASENAME, you are not limited to the names TSO0001, TSO0002, and so forth. For example, if you code

```
BASENAME=TTTT
```

the names will be TTTT0001, TTTT0002, and so forth.

The ACBNAME operand must be coded for TCAS and must specify TSO. The value you specify for BASENAME does not apply to the TCAS definition.

When using BASENAME, coding the ACBNAME operand on the APPL definition statements for TSO users is not necessary. The default value for the ACBNAME on an APPL definition statement is the name specified on the APPL definition statement. When BASENAME is used, TSO will use the same name for the APPL name and the ACBNAME. Therefore ACBNAME is not required on the TSO user APPL definition statement when BASENAME is used.

By specifying a unique BASENAME on every VTAM, you provide the name TCAS needs to start the TSO application address spaces, as well as provide the network-unique application name. Using the example above, you can code the following application program definitions:

```
tso1     APPL  ACBNAME=TSO,PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1

tttt0001 APPL  PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1

tttt0002 APPL  PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1

tttt0003 APPL  PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),MODSRCH=FIRST,EAS=1

  ⋮
```

You can combine BASENAME with wildcard characters, eliminating the multiple application program definitions. The following example shows how to do this:

```
tso1     APPL  ACBNAME=TSO,PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,
               TSO,NOPO),EAS=1

tttt*    APPL  PRTCT=password,
               AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),
               MODSRCH=FIRST, EAS=1
```

As each terminal logs on to TSO/VTAM, the * in the label tttt* will be replaced with the appropriate sequential 4-digit decimal integer; for example, 0001, 0002, and so forth.

Resource registration in an APPN network

To prevent incorrect session setups to TSO or TSO logon failures, the default resource registration is different for TSO applications than for other applications. Applications are registered to a directory server by default because they are likely to be the object of a search. Because multiple TSO applications can be defined using the same name, registering all TSO applications would cause misleading information to be stored in the directory server. Instead, TSO applications are registered according to the following conditions:

- If the APPL resource name is TSO and the ACBNAME is TSO, the default resource registration for this resource is NO registration.
- If the APPL resource name is tso*xxxxx* and the ACBNAME is TSO, the default registration for this resource is CDSERVR (register to the central directory server).
- For the APPL resource names for the TSO address spaces (for example, the names ending in 0001, 0002, and so forth), the default is NO registration.

It is assumed that AUTH=TSO is coded on each APPL definition statement.

Note: It is recommended that you not modify the registration of TSO applications from the default to avoid session setup or logon problems.

Defining TSO/VTAM session parameters

VTAM needs to know the session protocols required by the terminals from which individual TSO users log on. This information is provided through logon mode tables and through VTAM and NCP definition statements.

Logon mode tables associate session protocols with device types. IBM provides a default logon mode table that describes session protocols for common devices and situations. However, you might need to replace or supplement the default table.

In addition, a specific Class of Service name can be chosen. The Class of Service name used for the session is selected from the logon mode entry. If you omit the COS operand on the MODEENT macroinstruction, the default COS is chosen. To choose a specific COS, you need to include the COS name on the MODEENT macroinstruction as follows:

```
MODEENT  COS=cos name,logon mode parameters
```

The connection between a logon mode entry and a device is made using VTAM operands on the LU, LOCAL, or TERMINAL definition statements during VTAM and NCP definition. Code the DLOGMOD operand to identify the name of a logon mode table entry containing the session parameters used for the 3270 device being defined. Code the optional MODETAB operand to identify the logon mode table containing the entry. If you do not code MODETAB, the IBM-supplied logon mode table (ISTINCLM) is used.

You can also code the MODETAB and DLOGMOD operands on the GROUP, LINE, PU, LU, or CLUSTER definition statements to take advantage of the sift-down effect.

TSO/VTAM support of 3270 devices

TSO/VTAM supports the following two types of 3270 devices:

- Non-SNA devices attached locally or by a bisynchronous line protocol
- SNA devices attached by SDLC links

If no logon mode table entry exists for a 3270 device, TSO/VTAM assumes that the device is non-SNA (LU0) and uses the buffer size in the SCRSIZE parameter of the TSOKEY00 parmlib member. The following are valid values for the SCRSIZE parameter:

- 480 [12x40]
- 1920 [24x80]
- 2560 [32x80]
- 3440 [43x80]
- 3564 [27x132]
- 9920 [62x160]

For non-SNA devices with different screen sizes or special features and for SNA devices (LU2), you need to provide a logon mode table entry.

Use the MODETAB and MODEENT macroinstructions to define 3270 characteristics in logon mode tables and their entries. The PSERVIC operand of the MODEENT macroinstruction defines device LU type, buffer sizes, single-byte language, and QUERY capability (that is, programmed symbol sets, extended color, extended data stream, or extended highlight support). The LANG operand of the

MODEENT macroinstruction defines the single-byte language of the device and whether it should be queried for double-byte capability.

Following are examples of definitions of MODEENT table entries.

MODEENT macroinstruction for non-SNA 3270 devices: The FMPROF, TSPROF, PRIPROT, SECPROT, and COMPROT values shown in the following example are the same as those used in the IBM-supplied logon mode table, ISTINCLM.

```
name     MODEENT FMPROF=X'02',
               TSPROF=X'02',
               PRIPROT=X'71',
               SECPROT=X'40',
               COMPROT=X'2000',
               PSERVIC=X'...[1]'
```

Note: [1] — See Table 60 on page 722.

MODEENT macroinstruction for SNA 3270 devices: The RUSIZES operand in the following example defines a 256-byte maximum secondary logical unit RU send size and a 1024-byte maximum primary logical unit RU send size. If the maximum primary logical unit RU size is 0, TSO/VTAM defaults to 1024 bytes.

```
name     MODEENT FMPROF=X'03',
               TSPROF=X'03',
               PRIPROT=X'B1',
               SECPROT=X'90',
               COMPROT=X'3080',
               RUSIZES=X'8587',
               PSERVIC=X'...[1]'
```

PSERVIC operand of the MODEENT macroinstruction: Associated with each terminal is a default logon mode table entry. Within this entry, the primary and alternate screen sizes are specified, as well as a code that describes how these screen sizes are used.

The 11th byte of the MODEENT macroinstruction can have the value of X'03' (PSERVIC=X'.....03..'). When the field is specified as X'03', the primary screen size is 24x80, and the alternate screen size is determined by VTAM.

The 11th byte of the MODEENT macroinstruction can have the value of X'00' (PSERVIC=X'.....00..'). for the device with extended data stream capability. When the field is specified as X'00', the primary screen size and the alternate screen size is determined by VTAM. VTAM queries the device to determine the primary and the alternate screen sizes.

You do not have to code different logon mode table entries for TSO users when the only difference is the screen size. Users logging on to TSO with a screen size different from the default do not have to specify a logon mode table entry. The screen size is determined dynamically.

Note: The 3274 Terminal Controller must be Release 65, or higher, or the logon fails.

Code the 12 bytes of device-specific hexadecimal data of the PSERVIC operand as described in Table 60 on page 722.

Table 60. Coding the device-specific hexadecimal data of PSERVIC

Hexadecimal code	Related device
X'00..00000000.....00'	For non-SNA (LU0)
X'02..00000000.....00'	For SNA (LU2)
X'..00.....'	Device without extended data stream capability
X'..80.....'	Device with extended data stream capability
X'.....0000000001..'	Buffer size 480 only (12x40)
X'.....0000000002..'	Buffer size 1920 only (24x80)
X'.....0C280C507F..'	Buffer sizes 480 or 960 (12x40 or 12x80)
X'.....185020507F..'	Buffer sizes 1920 or 2560 (24x80 or 32x80)
X'.....18502B507F..'	For buffer sizes 1920 or 3440 (24x80 or 43x80)
X'.....18501B847F..'	For buffer sizes 1920 or 3564 (24x80 or 27x132)
X'.....18503EA07F..'	Buffer size 9920 only (62x160)
X'.....0000000003..'	Primary buffer size 1920 only (24x80). If a device is extended data stream capable (see above), TSO/VTAM queries the device to determine the alternate size.

To prevent switching of screen sizes on a device that has more than one size possible, code the screen size you want in the primary area of PSERVIC and a X'7E', which means no switching. Following is an example:

PSERVIC=X'.....1B8400007E..' For 3564 buffer only.

For details on coding the PSERVIC operand for your particular device, see your component description.

For details of the bit settings in the PSERVIC operand that represent bytes 13–24 of the session parameters, refer to *z/OS Communications Server: SNA Programming*.

LANG operand of the MODEENT macroinstruction: For TSO/VTAM, this operand affects (but does not necessarily decide) the following functions:

- The language of certain TSO/VTAM terminal user messages
- The language returned on the GTTERM macro
- The single-byte character set (SBCS) filter
- The device double-byte character set (DBCS) capability

The high-order bit of LANG is used, in conjunction with other LOGMODE information, to determine if a device is queried. A device is queried if it is extended data stream capable and either the high-order bit of LANG (from the MODEENT macroinstruction) is on, or the LOGMODE indicates to query for alternate screen size. The query of the device can affect each of the functions listed above.

Note: The high-order bit of the LANG operand is always obtained from the logon mode table entry (specified by the LOGMODE operand). If the terminal user enters a LOGON command, the LANG or LANGTAB operands of the LOGON command will override the low-order 7 bits specified by the LANG operand of the MODEENT macroinstruction.

The language of TSO/VTAM terminal operator messages IKT00201I through IKT00204I (logon failure messages)

TSO/VTAM attempts to retrieve these messages from the MVS message service (MMS), so that they can be issued in the language desired by the terminal user. The language for all other TSO/VTAM terminal user messages is set (determined) by the PLANG operand of the PROFILE command.

If the device is queried and the high-order bit of LANG is on, the language obtained from the query (if known) is used for the language of the messages.

The language returned on the GTTERM macro

The TSO terminal monitor program (TMP) or command processor may issue the GTTERM macro to determine the language of a TSO/VTAM user. The PROFILE command does not affect the results of the GTTERM macro.

If the device is queried and the high-order bit of LANG is on, the language obtained from the query (if known) is used for the language returned by the GTTERM macro.

The single-byte character set (SBCS) filter

If the device is queried, the appropriate SBCS filter is used (U.S. English filter, Katakana filter, or no filter). If the device is not queried, the SBCS filter is determined from the language value.

TSO/VTAM uses this filter in TPUT data validation and translation. TSO/VTAM performs character filtering for the FULLSCR, EDIT, ASIS, and NOEDIT options of the TPUT and TGET macros.

The DBCS capability of a device

If the query reply indicates that a device is DBCS capable, TSO/VTAM enables mixed DBCS strings to be sent and received from the device. If not queried, the device is assumed incapable of DBCS processing.

If you want to disable the character filtering of TSO/VTAM, code `LANG=X'7F'` for a device that does not support double-byte capability, and code `LANG=X'FF'` for a potential double-byte character set (DBCS) device.

Note: If an incorrect LANG operand value is specified or if LANG is not specified at all, the single-byte language defaults to U.S. English.

The low-order 7 bits indicate the single-byte language that is to be used. If a query is issued to determine double-byte capability and the high-order bit of LANG is on, any supported single-byte language value returned in the query reply overrides the value specified on the LANG operand.

The following indicates the bit settings for single-byte languages:

Bit	Description
0...	Indicates that the device should not be queried for language information.
1...	Indicates that the device should be queried for language information. This value should be used for devices that will potentially process DBCS data.

.nnn nnnn Indicates the language. For a list of valid values, refer to *z/OS Communications Server: SNA Programming*.

.111 1111 Indicates that SBCS filtering is disabled (no SBCS filter). Although not a valid language, this value can be used to disable character filtering. Note that any language other than U.S. English (X'00' or X'01') or Katakana (X'11') will also disable SBCS filtering.

Defining the 3790/3270 configuration to TSO/VTAM

The 3270 attached to a 3790 (Version 7) uses the LU type 2 protocol. You need to give session parameters to TSO/VTAM through a logon mode table entry to properly identify the model number or screen sizes of the device being used. The 3270 operator selects FPID 932 when logging on to the 3790 to activate the 3270 data stream compatibility function. During the logon procedure, the operator is prompted for the application program ID and the logon mode table entry containing the session parameters to be used for that session. The information is transmitted to VTAM by the Initiate Self request generated by the 3790. If the operator does not supply a logon mode table entry, the 3790 uses the standard logon mode entry name EMU3790.

So that the terminal operator does not have to know what logon mode entry name to supply, create a separate logon mode table for each 3270 model attached to the 3790, and include the default entry name EMU3790 in each table with the correct session parameters for that device.

Use the MODETAB and MODEENT macroinstructions to define the logon mode table and its entry. The PSERVIC operand on the MODEENT macroinstruction carries the model number or screen sizes in row and column form. Following is an example of defining a logon mode entry for a 3270 attached to a 3790 with a screen size of 1920 bytes (Model 2).

MODETAB macroinstruction:

```
LU2TABLE MODETAB
```

MODEENT macroinstruction:

```
EMU3790  MODEENT LOGMODE=EMU3790,
                 FMPROF=X'03',
                 TSPROF=X'03',
                 PRIPROT=X'B1',
                 SECPROT=X'90',
                 COMPROT=X'3080',
                 PSERVIC=X'020000000000000000000200'
```

For a description of the PSERVIC values, see “PSERVIC operand of the MODEENT macroinstruction” on page 721.

The association between the logon mode entry and the device is made on the LU definition statement during network definition to VTAM. Use the optional MODETAB operand to identify the logon mode table containing the entry. If MODETAB is not coded, the IBM-supplied logon mode table (ISTINCLM) is used.

Defining 2741, TWX, or WTTY terminals to TSO/VTAM

A 2741, TWX, or WTTY device attached to a communication controller can be used with TSO/VTAM through its LU type 1 protocol managers. These devices are identified to VTAM during the NCP generation process. (For details, refer to *Network Terminal Option Migration and Resource Definition* and the *NCP, SSP, and EP*

Resource Definition Guide.) Because the NCP translates ASCII line code to EBCDIC for these devices, make sure that any logon mode entry named by the DLOGMOD and MODETAB operands on the PU or LU definition statements identifies the device to VTAM as EBCDIC. Following is an example of an ASCII TWX logon mode table entry definition that has the alternate code indicator in the COMPROT field set off (to indicate EBCDIC to TSO/VTAM).

MODETAB macroinstruction:

```
TWXTABLE MODETAB
```

MODEENT macroinstruction:

```
TWXDEVIC MODEENT LOGMODE=TWXDEVIC,
                 FMPROF=X'03',
                 TSPROF=X'03',
                 PRIPROT=X'B1',
                 SECPROT=X'90',
                 COMPROT=X'3040'
                 DCODE=X'00'
```

The DCODE operand on the MODEENT macroinstruction indicates to TSO/VTAM whether a TWX device is a keyboard display or a keyboard printer. If you specify DCODE to indicate that the device is a keyboard display, TSO/VTAM inhibits the display of the password when the password is entered at the device.

Defining an interpret table for compatible logons

Because TSO/VTAM uses VTAM logon facilities, the TSO LOGON command is not supported. You can make this transparent to terminal users by defining an interpret table to allow logon requests to have the same format as that used by the TSO LOGON command. The following interpret table definition can be used:

```
INTBL    INTAB
         LOGCHAR APPLID=(APPLICID,TSO),SEQNCE='LOGON'
         LOGCHAR APPLID=(APPLICID,TSO),SEQNCE='logon'
         ENDINTAB
         END
```

This interpret table allows a user to enter a logon command in uppercase or lowercase letters.

Defining TSO to MVS

This section describes how to define TSO to MVS.

Writing a procedure to start TSO/VTAM time sharing

You need to write a cataloged procedure to start TSO/VTAM time sharing and include it in either SYS1.PROCLIB or your own procedure library. For details, refer to *z/OS TSO/E Customization*.

Creating a TSOKEY00 PARMLIB member

TSO/VTAM time sharing parameters determine how time sharing buffers should be controlled, the maximum number of users, and other options. Unless you want the IBM default parameters to apply, you must provide a parmlib member containing the parameters that you want. Information on initialization and tuning for your operating system describes the parameters you can code, how to code them, and the IBM defaults.

You can use MVS system symbols in the TSOKEY00 parmlib member. For information about using MVS system symbols, see “Using MVS system symbols” on page 39.

Defining TCAS program properties

TCAS must have an entry in the MVS program properties table (PPT). This entry designates the execution attributes of TCAS, which should be as follows:

- The program cannot be canceled.
- Unique protection key 6 must be assigned to the program.
- The program is privileged and is not swapped unless the address space is in a long wait.
- The program is a system task and is not timed.
- There is no host affinity.

For coding conventions, refer to initialization and tuning information for your operating system.

Implementing TSO/VTAM

This section has the following subsections:

- Translation tables
- Coding TSO/VTAM exit routines
- Security
- Performance
- 3270 large screen considerations
- TSO considerations
- Multicultural support for TSO end-user messages
- Operating VTAM under TSO

Translation tables

Translation tables allow TSO/VTAM users to replace internally characters that are not available on a keyboard. If you call for character translation, translation tables (either your own or those supplied by IBM) are used. For more information, refer to *z/OS TSO/E Customization*.

Coding TSO/VTAM exit routines

Exit routines provide the following functions needed in TSO/VTAM:

- Input and output editing that replaces or supplements IBM-supplied editing
- Attention handling that replaces IBM-supplied attention handling
- Support for terminals not supported by TSO/VTAM
- Verifying whether a terminal is capable of receiving a TSO/VTAM end-user message in the language selected by the TSO user

For information on coding TSO/VTAM exit routines, refer to *z/OS Communications Server: SNA Customization*.

Security

The TSO subsystem is considered a secure application program. That is, confidential data is handled on behalf of the user in ways that prevent unauthorized disclosure of the data. The CONFTXT parameter in the TSOKEY00 parmlib member determines whether output data is considered confidential text.

The default is that the data is considered confidential. TSO/VTAM protects tracing of user data by setting the CONFTXT indicator in the NIB at the time the user logs on.

If CONFTXT=NO, VTAM can perform buffer or I/O traces on the data. If CONFTXT=YES (the default), the data is considered confidential and no data is recorded. The CONFTXT parameter, however, does not apply to the TSO type VTAM trace for TPUT/TPG/TGET buffers; these are always traceable. For details, see information on initialization and tuning for your operating system.

VTAM supports TSO message security by invoking RACF services to provide resource access control for the following:

- Cross-address space TPUTs (such as the TSO SEND command), which do the following:
  - Control who can send messages to whom
  - Ensure that a message will be received by the intended user
  - Ensure that a cross-address-space message can be received only by a user with a security classification that is equal to or greater than the sender
- Requests to open an ACB from a non-APF authorized application program or processor.

Note: The installation must ensure that both the sender and receiver of TSO/VTAM messages are authorized with the proper security level in the security management product. The TSO/VTAM user IDs should be registered with a class of SMESSAGE in the security management product.

Performance

Following are suggestions for improving TSO/VTAM performance:

- Terminal users should stack input data whenever possible. This means using the multiple-line input technique on 3270 terminals and buffered SDLC transmission on 3767 and 3770 terminals. Stacking input results in a decreased number of data transmissions to TSO. Users of 3270 terminals can stack up to a full screen of data; 3767 and 3770 terminal users can stack from 256 bytes to 2000 bytes of data, depending on the terminal buffer size.
- Users of 3767 and 3770 terminals can reduce idle time by "typing ahead." This means that whenever the terminal is not receiving data or is transmitting data, the user can enter input.
- Users of 3276 and 3278 terminals can expedite unlocking the keyboard if you use BREAK mode. BREAK mode reduces idle time by allowing typing ahead.
- Users of 3270 terminals can shorten the TSO/VTAM processing path when handling input if you use automatic line numbering. This reduces system overhead and expedites unlocking the keyboard.

3270 large screen considerations

The following information describes screen management techniques for IBM 3270 terminals that have default and alternate screen sizes determined by the logon mode table entries used for them at logon. The default size is accessed by sending an Erase/Write (EW) command to the terminal, and the alternate size is accessed by sending an Erase/Write Alternate (EWA) command.

TSO/VTAM screen management

TSO/VTAM can manage the screen for TSO application programs that use line-oriented I/O. In this case, TSO/VTAM uses the maximum screen size for the

device, whether it is the default or the alternate in the logon mode table entry. When screen erasure is necessary, TSO/VTAM uses either the EW or the EWA command as required to access the larger screen size.

Full-screen application program screen management

An IBM-supplied or user-supplied application program that uses full-screen I/O in TSO full-screen mode can be used to manage the screen. TSO/VTAM uses the screen size used by the application program. If the CLEAR key is pressed, TSO/VTAM erases the screen with either an Erase/Write (EW) or Erase/Write Alternate (EWA) command depending on which was last used by the application program.

TSO considerations

Information about the TSO-related steps that must be performed before a TSO/VTAM terminal user logs on is in *z/OS TSO/E Customization* and in information on initialization and tuning for your operating system. (Make sure you have updated these books with TSO/VTAM supplements before referring to them.) The following list outlines the steps and refers you to the appropriate publication:

1. Write LOGON cataloged procedures, and include them in SYS1.PROCLIB. Refer to *z/OS TSO/E Customization*.
2. Construct the TSOKEY00 member or an alternate parmlib member (or an alternate data set) to set VTIOC parameters. Refer to information on initialization and tuning for your operating system.
3. Include SYS1.CMDLIB in a LNKLST*xx* parmlib member or in a LOGON cataloged procedure. Refer to information on initialization and tuning for your operating system.
4. Create or convert the user attribute data set (UADS) and the broadcast data set. Refer to *z/OS TSO/E Customization*.
5. Build translation tables if you want character translation. Refer to *z/OS TSO/E Customization*.
6. Write the procedure that starts TSO/VTAM time sharing. Refer to *z/OS TSO/E Customization*.
7. Write any command exit routines you plan to use. Refer to *z/OS TSO/E Customization*.

Note: You are strongly encouraged not to VARY LOGON to TSO or code TSO as a LOGAPPL because unpredictable results can occur. For instance, if the original BIND fails, TSO messages and VTAM messages may be repeatedly issued to the console indicating the session setup failure. In some cases this may not stop even if the VARY NOLOGON command is used to break the automatic logon relationship.

Multicultural support for TSO/VTAM end-user messages

You can tailor TSO/VTAM end-user messages using the MVS Message Service (MMS). You define TSO/VTAM end-user messages to MMS the same way you define USS end-user messages. For details, see “Defining USS messages to the MVS message service” on page 289.

The TSO user selects the language used for TSO/VTAM messages using the PROFILE command with the PLANG operand. TSO/VTAM then retrieves its messages from MMS using the language specified by the user. This language continues to be used until the TSO/user changes the language (by reissuing the PROFILE command).

Before the TSO user can issue the PROFILE command, TCAS may issue messages IKT00201I through IKT00204I. The language used for these messages is specified on either the LANG operand of the MODEENT macroinstruction or on the USS logon command using the LANG or LANGTAB operands. The language is passed from these sources to the TSO/VTAM logon exit.

For information on how MMS translates messages, see Appendix G, "Message translation using the MVS Message Service," on page 769.

Operating VTAM under TSO

A TSO operator is defined as any TSO user with the authority to use the CONSOLE command. MVS or subsystem operators running under TSO are treated as extended multiple consoles. Authorized users can establish a console session with MVS console services. After the session is active, users can enter MVS and subsystem commands and obtain solicited and unsolicited messages.

By representing all consoles and operators as a 4-byte console ID, multiple console support (MCS) can extend its current command and message services to TSO operators. This allows TSO operators to send VTAM commands and receive VTAM messages.

To enable this function, you need to grant individual users CONSOLE command authority, so that they can use the MVS CONSOLE command. There are three ways that you can give CONSOLE authority to users:

- Using RACF facilities (RACF RDEFINE and RACF PERMIT commands)
- Using the TSO logon preprompt exit routine
- Using the TSO CONSOLE exit routine

For information on giving CONSOLE authority, the levels of authority, and the CONSOLE command, refer to *z/OS TSO/E Customization*.

A TSO operator can use the MVS Message Service (MMS) to retrieve translated messages, including VTAM, logon manager, and TSO/VTAM messages for network operators. The TSO operator can select a language using the PROFILE command with the PLANG operand. For information on how MMS translates messages, see Appendix G, "Message translation using the MVS Message Service," on page 769. For details on defining translated network operator messages, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Appendix B. Storage estimate worksheets

This appendix describes how to estimate the virtual storage required to run z/OS Communications Server on the z/OS operating system. From the following list, select and review the applicable worksheets:

- APPN
  - Interchange node (ICN) or network node (NN)
  - Migration data host (MDH) and end node (EN)
- Subarea
  - Communication management configuration (CMC)
  - Data host (DH)

This appendix also contains an APPL EAS storage estimate table; see "APPL EAS storage estimates" on page 738.

Users of APPC will notice an increase in storage utilization because VTAM will now allocate an additional 160 bytes (for a mini-VIT trace) per control block representing a single APPC conversation. Refer to *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT* for information on VTAM Internal Trace (VIT) tracing.

Users of APPN will notice an increase in storage utilization because VTAM will now allocate additional storage for TRS (Topology and Routing Services) topology traces. To calculate the increase, add the following:

- For the TRS topology trace where topology record deletions are recorded, one to ten 4K pages of storage will be allocated. One 4K page will be allocated at VTAM initialization. When that page is full of trace entries, another page will be allocated, up to a maximum of ten 4K pages.
- For the NDREC (node record) traces, 110 bytes of storage will be allocated for each node in the network or sub-network.
  - In a network node, this would include:
    - All network nodes
    - All served end nodes (the network node has acted as the NNS or DLUS for the end node)
    - All virtual nodes
  - In an end node, this would include:
    - Network nodes with which the end node has established connections
    - End nodes with which the end node has established connections
    - Virtual nodes through which the end node has established connections
- For the TGREC (TG record) traces, 180 bytes of storage will be allocated for each TG in the network or subnetwork.
  - In a network node, this would include:
    - Two TGs for every link between network nodes (one TGREC represents the connection in one direction and one TGREC represents the connection in the reverse direction)
    - One TG for every connection that a VTAM end node served by this network node server has with any other network node, end node, or virtual node

© Copyright IBM Corp. 1984, 2011

- One TG for every connection that a DLUR end node served by this DLUS has with any other network node, end node, or virtual node
- Two TGs for every connection between network nodes and virtual nodes (one TGREC represents the connection in one direction and one TGREC represents the connection in the reverse direction)
- One TG for every connection to an adjacent end node or an adjacent DLUR end node

– In an end node, this would include:
- One TG for every link from this end node to any other network node, end node, or virtual node.

General information

These worksheets address z/OS Communications Server storage above 16MB; storage below 16MB is allocated as 150KB common storage, and 64KB private storage.

The heading "DISPLAY STATS ID" refers to the particular statistic identifier issued by the **D NET,STATS** command; this statistic identifier is used in the corresponding step of the worksheet. There are some statistics that **D NET,STATS** does not capture; they are indicated by N/A.

"Dynamic storage" refers to storage created in response to a need, and required only so long as the process using it stays active. Dynamic storage can be used to establish normal sessions, and for error recovery. Dynamic storage usage varies by configuration; it is related to the number of sessions being established. The worksheet yields an approximation of dynamic storage needed for a given configuration, assuming worst case system recovery.

Estimation of z/OS Communications Server storage is based on the following assumptions. (These assumptions reflect no implied or expressed recommendation.)

- All PUs and LUs defined in the configuration are active.
- Tables and user exit routines are not used.
- Default buffer pool values, dynamic buffer expansion, and default start options are used.
- Dynamic storage requirements are based on full system recovery.

In order to establish storage estimates, try the following:

1. Set REGION=0 for Private Storage, and CSALIMIT=0 for CSA/ECSA.
2. Monitor storage for approximately 2 or 3 weeks to determine what the peak values are.
3. Once peak value is identified, then establish values for REGION and CSALIMIT. Use 1.5 times the peak value noted during the 2 or 3 week observation period.

APPN interchange node or network node

For APPN interchange node (ICN) or network node (NN) configurations, use the following worksheet:

Table 61. Worksheet for APPN interchange node or network node storage

Step	Description	Formulas (decimal)	Common	Private	DISPLAY STATS ID
1	Number of 4KB pages allocated for the SNA internal trace table	(trace tab + 4) * 4096	= _____		2
2	Determine the size of the **IOBUF** parameter. This value is defined in the SNA start list. These buffers hold data transmitted to and from SNA.	Used in questions 4 & 12			3
3	Number of channel-attached communication controllers (NCPs) activated and owned by this SNA.	COMMON STORAGE: (NCPs * 1200)	= _____		5
4	Sum value of **MAXBFRU** parameters for all channel-attached communications controllers activated by this SNA. **MAXBFRU** is defined in the **HOST** definition statement of the NCP channel-attached major node definition.	COMMON STORAGE: ((IOBUF size + 98) * 2 * maxbfru NCPs)	= _____		3 & 6
5	Number of PUs defined in this SNA. Include all PUs defined to SNA in **PU** definition statements, and controllers defined in **CLUSTER** definition statements. Include locally attached, remotely attached, dynamically added, switched, ICA and NCP (including NTRI) PUs.	PRIVATE STORAGE: (defined PUs * 1000)		= _____	48 & 67
6	Number of device type LUs defined in this SNA. Include the locally attached LUs, and LUs attached through an NCP. In addition, include LUs that are defined to SNA in **LU** definition statements, and those devices defined in **TERMINAL** statements. Do not include applications.	PRIVATE STORAGE: (defined LUs * 820)		= _____	50
7	Number of independent LUs defined locally, remotely or by way of CDRSC. Include all independent LUs for which SNA provides boundary function services, and all NTRI independent LUs.	COMMON STORAGE: (indep LUs * 270) PRIVATE STORAGE: (indep LUs * 400)	= _____	= _____	80

Table 61. Worksheet for APPN interchange node or network node storage (continued)

Step	Description	Formulas (decimal)	Common	Private	DISPLAY STATS ID
8	Number of LU 6.2 sessions with application LUs owned by this SNA. LU 6.2 sessions are valid only for applications where **APPC=YES** is specified in the APPL major node definition. Include all same domain, cross domain, and cross network LU 6.2 sessions.	COMMON STORAGE: (LU6.2 ses * 840)	= _____		58
9	Number of device type LUs owned by this SNA, and in session with an application program owned by this SNA.	COMMON STORAGE: (LUs w/appls * 500) PRIVATE STORAGE: (LUs w/appls * 420)	= _____	= _____	71
10	Number of device type LUs owned by this SNA, and in session with an application program owned by another SNA.	PRIVATE STORAGE: (cross node LU ses * 420)		= _____	73
11	Number of LU 6.2 sessions with both LUs owned by this SNA.	PRIVATE STORAGE: (LU6.2 ses * 450)		= _____	77
12	Number of device type (nonapplication) LUs in session with a TSO application.	COMMON STORAGE: (TSO LUs) * (2300 + IOBUF size)	= _____		55 & 3
13	Number of ENs that establish CP-CP sessions with this SNA.	PRIVATE STORAGE: (adj end node * 3170) COMMON STORAGE: (adj end node * 920)	= _____	= _____	104
14	Number of transmission groups used between this node and attached, or served, end nodes.	PRIVATE STORAGE: (end node TGs * 690)		= _____	142
15	Number of transmission groups used between this node and other network nodes.	PRIVATE STORAGE: (network node TGs * 690)		= _____	143
16	If the SNA topology agent is being used, enter the number of resources being monitored.	PRIVATE STORAGE: (num res * 3500)		= _____	N/A
		Total Common = $\frac{\text{_____}}{1024}$	= _____KB (*totcom*)		
		Total Private = $\frac{\text{_____}}{1024}$		= _____KB (*totpri*)	

Table 62. Summary of worksheet, APPN interchange node or network node storage

Calculated COMMON storage for ICN/NN configuration above	= _____KB (*totcom*)
DYNAMIC COMMON STORAGE (4 x *totcom*)	= _____KB

Table 62. Summary of worksheet, APPN interchange node or network node storage (continued)

SNA TOPOLOGY AGENT STORAGE (if used)	+ 2000KB	
SNA SYSPLEX STORAGE (if used)	+100KB	
SNA BASE STORAGE (COMMON)	+ 3002KB	
TOTAL SNA COMMON STORAGE	= _____KB	
Calculated PRIVATE storage for ICN/NN configuration above		= _____KB (*totpri*)
DYNAMIC PRIVATE STORAGE (2 x *totpri*)		= _____KB
SNA SYSPLEX STORAGE (if used)	+750KB	
SNA BASE STORAGE (PRIVATE)		+ 7057KB
TOTAL SNA PRIVATE STORAGE		= _____KB

APPN migration data host and end node

For an APPN migration data host (MDH) or end node (EN) configuration, use the following worksheet:

Table 63. Worksheet for APPN migration data host and end node

Step	Description	Formulas (decimal)	Common	Private	DISPLAY STATS ID
1	Number of 4KB pages allocated for the SNA internal trace table.	(trace tab + 4) * 4096	= _____		2
2	Number of device-type LUs owned by this SNA in session with an application program owned by this SNA.	PRIVATE: (LUs w/appls * 420) COMMON STORAGE: (LUs w/appls * 500)	= _____	= _____	71
3	Number of cross node sessions between an application program in this SNA and a device type LU owned by another node or SNA.	PRIVATE: (cross node appl * 980) COMMON STORAGE: (cross node appl * 540)	= _____	= _____	112
		Total Common = _____ / 1024	= _____KB (*totcom*)		
		Total Private = _____ / 1024		= _____KB (*totpri*)	

Table 64. Summary of APPN migration data host and end node

Calculated COMMON storage for MDH/EN configuration from above	= _____KB (*totcom*)	
DYNAMIC COMMON STORAGE (2 x *totcom*)	= _____KB	
SNA SYSPLEX STORAGE (if used)	+100KB	
SNA BASE STORAGE (COMMON)	+ 3002KB	
TOTAL SNA COMMON STORAGE	= _____KB	
Calculated PRIVATE storage for MDH/EN configuration from above		= _____KB (*totpri*)
DYNAMIC PRIVATE STORAGE (2 x *totpri*)		= _____KB
SNA SYSPLEX STORAGE (if used)	+750KB	
SNA BASE STORAGE (PRIVATE)		+ 7057KB
TOTAL SNA PRIVATE STORAGE		= _____KB

Subarea data host

For a subarea data host (DH) configuration, use the following worksheet:

Table 65. Worksheet for subarea data host

Step	Description	Formulas (decimal)	Common	Private	DISPLAY STATS ID
1	Number of 4KB pages allocated for the SNA internal trace table.	(trace tab + 4) * 4096	= _____		2
2	Number of device type LUs owned by this SNA in session with an application program owned by this SNA.	PRIVATE: (LUs w/appls * 420) COMMON STORAGE: (LUs w/appls * 500)	= _____	= _____	71
3	Number of cross domain sessions between an application program in this SNA and a device type LU owned by another SNA.	PRIVATE: (cross node appl * 910) COMMON STORAGE: (cross node appl * 540)	= _____	= _____	112
		Total Common = _____ / 1024	= _____KB (*totcom*)		
		Total Private = _____ / 1024		= _____KB (*totpri*)	

Table 66. Summary of subarea data host

Calculated COMMON storage for DH configuration from above	= _____KB (*totcom*)	
DYNAMIC COMMON STORAGE (2 x *totcom*)	= _____KB	
SNA BASE STORAGE (COMMON)	+2953KB	
TOTAL SNA COMMON STORAGE	= _____KB	
Calculated PRIVATE storage for DH configuration from above		= _____KB (*totpri*)
DYNAMIC PRIVATE STORAGE (2 x *totpri*)		= _____KB
SNA SYSPLEX STORAGE (if used)	+150KB	
SNA BASE STORAGE (PRIVATE)		+ 5507KB
TOTAL SNA PRIVATE STORAGE		= _____KB

Subarea communication management configuration

For a subarea communication management configuration (CMC), use the following worksheet:

Table 67. Worksheet for subarea communication management configuration

Step	Description	Formulas (decimal)	Common	Private	DISPLAY STATS ID
1	Number of 4KB pages allocated for the SNA internal trace table.	(trace tab + 4) * 4096	= _____		2
2	Determine the size of the **IOBUF** parameter. This value is defined in the SNA start list. These buffers hold data transmitted to and from SNA.	Used in questions 4 & 10			3

Table 67. Worksheet for subarea communication management configuration (continued)

Step	Description	Formulas (decimal)	Common	Private	DISPLAY STATS ID
3	Number of channel-attached communication controllers (NCP) activated and owned by this SNA.	COMMON STORAGE: (NCPs * 1200)	= _____		5
4	Sum value of **MAXBFRU** parameters for all channel-attached communications controllers activated by this SNA. **MAXBFRU** is defined in the **HOST** definition statement of the NCP channel-attached major node definition.	COMMON STORAGE: ((IOBUF size + 98) * 2 * maxbfru NCPs)	= _____		3 & 6
5	Number of PUs defined in this SNA. Include all PUs defined to SNA in PU definition statements, and those controllers defined in **CLUSTER** definition statements. Include locally attached, remotely attached, dynamically added, switched, ICA, and NCP (including NTRI) PUs.	PRIVATE STORAGE: (defined PUs * 850)		= _____	48 + 67
6	Number of device type LUs defined in this SNA. Include the locally attached LUs, and the LUs attached through an NCP. In addition, include the LUs defined to SNA in LU definition statements, and those devices defined in **TERMINAL** statements. Do not include applications.	PRIVATE STORAGE: (defined LUs * 820)		= _____	50
7	Number of independent LUs either locally, remotely, or CDRSC defined. Include all independent LUs for which SNA provides boundary function services and all NTRI independent LUs.	COMMON STORAGE: (indep Lus * 270) PRIVATE STORAGE: (indep Lus * 400)	= _____	= _____	80
8	Number of device type LUs owned by this SNA in session with an application program owned by this SNA.	COMMON STORAGE: (LUs w/appls * 500) PRIVATE STORAGE: (LUs w/appls * 420)	= _____	= _____	71
9	Number of device type LUs owned by this SNA in session with an application program in another SNA.	PRIVATE STORAGE: (cross node LU ses * 400)		= _____	73
10	Number of device type (nonapplication) LUs in session with a TSO application.	COMMON STORAGE: (TSO LUs) * (2300 + IOBUF size)	= _____		55 & 3
11	If the SNA topology agent is being used, enter the number of resources being monitored.	PRIVATE STORAGE: (num res * 3500)		= _____	N/A

Table 67. Worksheet for subarea communication management configuration (continued)

Step	Description	Formulas (decimal)	Common	Private	DISPLAY STATS ID
		Total Common = _____ / 1024	= _____KB (*totcom*)		
		Total Private = _____ / 1024		= _____KB (*totpri*)	

Table 68. Summary of subarea communication management configuration

Calculated COMMON storage for CMC configuration from above	= _____KB (*totcom*)		
DYNAMIC COMMON STORAGE (4 x *totcom*)	= _____KB		
SNA TOPOLOGY AGENT STORAGE (if used)	+ 2000KB		
SNA BASE STORAGE (COMMON)	+ 2953KB		
TOTAL SNA COMMON STORAGE	= _____KB		
Calculated PRIVATE storage for CMC configuration from above		= _____KB (*totpri*)	
DYNAMIC PRIVATE STORAGE (2 x *totpri*)		= _____KB	
SNA SYSPLEX STORAGE (if used)	+150KB		
SNA BASE STORAGE (PRIVATE)		+ 5507KB	
TOTAL SNA PRIVATE STORAGE		= _____KB	

APPL EAS storage estimates

EAS is the estimated number of concurrent sessions this application program will have with other logical units (LU-LU sessions). Accurate coding of the EAS value for your applications can save storage in your system. If your EAS value is specified as lower than the number of sessions that you actually have, sessions would still be established as usual. However, the efficiency of searching for the session representation could be impaired if a smaller table was allocated due to the lower EAS value.

For example, if you estimate that there will be less than 30 sessions with this application, but you let the EAS value default to 509, then an extra 4K table will be allocated from common storage. The size of the table is based on the EAS value that you code and is determined as follows:

Table size	EAS value
4K	30–4000
8K	4001–8000
16K	8001–16000
32K	16001–32000
64K	32001–48000
128K	48001–56000
256K	56001–64000
512K	greater than 64000

Note: For each OPEN ACB with EAS less than 30, one LF buffer will be allocated (for an LUCB) and there will be no additional table storage allocated. If EAS

is greater than 30, then one SF buffer will be allocated (for an LUCB) in addition to storage for the table size indicated in the table above.

Refer to *z/OS Communications Server: SNA Resource Definition Reference* for more information on EAS.

Appendix C. Communications storage manager

Communications storage manager (CSM) is a component of VTAM that allows authorized host applications to share data with VTAM and other CSM users without having to physically pass the data. A CSM user can be any system-authorized application program or product.

CSM is provided as part of the high-performance data transfer (HPDT) family of services. HPDT optimizes system performance for the transfer of bulk data. By providing a means for authorized applications to share buffers, CSM improves system performance during the transfer of bulk data by reducing the processing required for data movement. As a result, CPU resources (CPU cycles, memory bus, and cache) are conserved.

VTAM uses CSM to perform channel I/O over an HPDT MPC connection and to provide the HPDT services for host LU 6.2 applications. (HPDT MPC connections are explained in "Multipath channel connections" on page 51.) Host LU 6.2 applications use the IVTCSM and APPCCMD macroinstructions to reduce the use of system resources for large data transfers. For more information about HPDT services, see "High-performance data transfer (HPDT)" on page 424.

CSM includes an application programming interface (IVTCSM macroinstruction) that allows users to obtain and return storage in the form of CSM buffer pools by using the IVTCSM macroinstruction. Applications must be authorized to use the IVTCSM macroinstruction.

CSM buffer pools are identified by the fifteen distinct combinations of storage type and buffer size as described in Table 69 and specified on the IVTCSM REQUEST=CREATE_POOL macroinstruction.

Table 69. Buffer pools in CSM

Storage types	Buffer sizes				
31-bit backed data space	**4 KB**	**16 KB**	**32 KB**	**60 KB**	**180 KB**
64-bit backed data space	**4 KB**	**16 KB**	**32 KB**	**60 KB**	**180 KB**
ECSA	**4 KB**	**16 KB**	**32 KB**	**60 KB**	**180 KB**

Data space storage is a common area data space and is associated with the master scheduler address space. This association results in a data space that persists for the life of the system.

Data space storage is either 31-bit backed or 64-bit backed. 31-bit backed data space, when fixed, resides below the 2-gigabyte real storage bar. 64-bit backed data space, when fixed, can reside below or above the 2-gigabyte bar. Where the storage is backed is a concern for only those products performing I/O into or out of the storage.

The application programming interface for CSM is described in *z/OS Communications Server: CSM Guide*.

© Copyright IBM Corp. 1984, 2011

The rest of this appendix summarizes the interfaces that allow you to install, configure, monitor, and diagnose CSM storage.

CSM installation and definition

CSM is shipped and installed with the VTAM product tape. However, many CSM functions are independent of VTAM. CSM storage limits and tuning parameters are defined in the CSM parmlib member, IVTPRM*00*.

The first two lines in the CSM parmlib member define the maximum amount of storage to be dedicated to fixed and ECSA buffers in CSM. You can also specify one POOL definition for each CSM buffer pool of a particular *bufsize* and *bufsource* combination. On each POOL definition, you can specify the following:

- The minimum number of buffers to be free in the pool at any time (MINFREE). The storage pool is expanded when the number of free buffers falls below this limit.
- The number of buffers by which the pool is expanded (EXPBUF) when the number of free buffers falls below MINFREE.
- The initial number of buffers to be created in the pool (INITBUF). The pool does not contract below the contraction threshold value. CSM determines the contraction threshold value using the following formula:

  MAX(INITBUF,(MINFREE+(2*EXPBUF)))

If parameters are not provided for a given CSM buffer pool, the IBM-supplied default values are used unless an application has provided these values on a IVTCSM REQUEST=CREATE_POOL macro.

Refer to *z/OS Communications Server: New Function Summary* for a complete description of the CSM parmlib member.

IBM Health Checker for z/OS can be used to check whether appropriate values are defined for the maximum amount of storage to be dedicated to fixed and ECSA buffers in CSM. For more details about IBM Health Checker for z/OS, refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.

Initializing CSM

CSM is initialized by the first request to create a pool of buffers and remains active for the life of the system, independent of VTAM status. This could be issued by VTAM or a host application. Upon initialization, CSM reads the CSM parmlib member to determine storage limits and buffer pool related values.

Monitoring CSM

System operators can monitor the use of CSM storage by issuing the DISPLAY CSM command. CSM messages always start with the message prefix IVT. For a complete list of messages issued by CSM, see *z/OS Communications Server: SNA Messages*. An application can monitor the use of CSM storage by using the SNA network monitoring NMI. See "SNA network monitoring NMI" in *z/OS Communications Server: IP Programmer's Guide and Reference* for more detailed information.

The following information is provided by the DISPLAY CSM command.

- Amount of storage allocated to each pool

- Amount of storage allocated to each user of the pool
- If OWNERID=ALL, the cumulative storage allocated to each user across all pools
- If the OWNERID value is not specified, the maximum amount of fixed and ECSA storage that can be allocated by CSM and current values for fixed and ECSA storage
- If the OWNERID value is not specified, the highest level of fixed storage obtained since the last DISPLAY CSM command was issued without the OWNERID parameter
- If the OWNERID value is not specified, the highest level of fixed storage obtained since the IPL
- If the OWNERID value is not specified, the highest level of ECSA obtained since the last DISPLAY CSM command was issued without the OWNERID parameter.
- If the OWNERID value is not specified, the highest level of ECSA obtained since the IPL
- If the OWNERID value is not specified, the names of the CSM data spaces

The DISPLAY CSM command can be used to identify a user of the pool that is consuming inordinate amounts of storage. This could occur in situations where an application fails to free buffers that it obtained from CSM. The report of storage allocated to a user is based on the user *owner_ID* (OWNERID operand on the DISPLAY CSM command). CSM uses the application address space identifier (ASID) as the user OWNERID. The information by OWNERID indicates the amount of storage that must be freed by the user to enable the storage to be returned to the buffer pool.

Some messages are issued by CSM when CSM storage limits are either at a critical level (90 percent of defined limits) or exceeded. In this case, the system operator can issue the MODIFY CSM command to increase the amount of fixed or ECSA storage available for CSM. CSM storage limits can be increased or decreased without requiring a re-IPL by issuing the MODIFY CSM command.

A Display CSMUSE command is available to provide the lower level of detail regarding CSM storage usage. The Display CSMUSE command provides displays of CSM storage usage by z/OS Communications Server components. Display of this information is intended to improve the serviceability of z/OS Communications Server and to aid in CSM storage diagnosis.

For more information about these CSM commands, refer to *z/OS Communications Server: SNA Operation*. CSM storage information is also provided to the performance monitor interface (PMI). This information is equivalent to the information provided for the summary format of the DISPLAY CSM command. Refer to *z/OS Communications Server: SNA Customization* for more information.

CSM problem diagnosis

You can obtain trace output of application requests to CSM by using the CSM option on the VTAM internal trace (VIT) or, when VTAM is not active, the GTF trace facility.

- When VTAM is operational, the CSM trace facility is controlled using the VIT. CSM writes records to the VIT using VTAM trace interfaces. The CSM trace option is used to control the generation of CSM trace records for both internal and external tracing.

- When VTAM is not operational, the VIT is not available and only external tracing is provided. The external trace is generated using the VTAM GTF event ID to write trace records directly to GTF in the same format as those recorded using VIT.

CSM tracing records the parameter list information that flows across the CSM interface and key internal events (such as pool expansion and contraction) for functions that manipulate buffer states. This allows you to trace and analyze the usage history of a buffer.

The number of trace records required to represent one IVTCSM request is variable based on the number of buffer operations requested. Because the information required to trace one IVTCSM request may span several CSM trace records, you can use the trace record flag field to determine whether additional trace records exist for a particular IVTCSM request. If the first bit in the trace record flag field is set on, the trace record is continued. If the VIT is not active, multiple trace records for an IVTCSM request could be interspersed with trace records of IVTCSM requests from other users. A unique trace record number is provided to correlate the continuation trace records for each IVTCSM request.

For more information about tracing events over the CSM API, refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.

Appendix D. Logon manager

The transaction processing facility (TPF) is an application program dedicated to high-speed transaction processing. It is designed to handle a large volume of transactions submitted from many remote logical units. The logon manager is a VTAM application program that provides logon services for TPF.

The logon manager provides logon access to a TPF application program from independent and dependent logical units. If you define your TPF application program as a host processor (type 5), only dependent LUs can access the TPF application program. However, if you define your TPF application program as a type 2.1 physical unit, both dependent and independent LUs can access the TPF application program.

Note: To implement the logon manager, you need to have NCP Version 4 Release 3 or a subsequent release.

This appendix contains the following sections:
- How the Logon Manager Operates
- Installing the Logon Manager
- Starting the Logon Manager
- Defining the Logon Manager
- Monitoring Logon Manager Resources
- Halting the Logon Manager

How the logon manager operates

How the logon manager receives control depends on whether a dependent or independent LU is accessing the TPF application program. If the logon request is from a dependent SLU, the logon manager receives control when VTAM drives its logon exit. If the logon request is from an independent PLU, the logon manager receives control when VTAM drives the USERVAR exit, ISTEXCUV. ISTEXCUV is driven because the logon manager creates a USERVAR (with UVEXIT=YES) to represent the generic name of each TPF application program.

After the logon manager receives control, the control point LUs that support the generic application are identified. Each pair of supporting control point LUs is then passed to the CLU search exit, ELMCLUEX, to determine the LU best able to provide the service. The selection criteria of the ELMCLUEX exit can be determined by the user or the default can be used. The default criteria are:
- LU is below the session limit
- LU has the lowest hop count to the device
- LU has the fewest number of active sessions

For dependent LU access, the TPF application program can be defined as either a type 5 or type 2.1 PU.

For independent LU access, the TPF application program must be defined as a type 2.1 PU.

© Copyright IBM Corp. 1984, 2011

Installing the logon manager

The VTAM installation process places the logon manager load modules in SYS1.LINKLIB. You can put the logon manager in another load library by modifying the installation process. If you do this, you must either add this other load library to your system link list or point the logon manager JCL to this library by using a STEPLIB DD statement. Also, the logon manager must be APF authorized.

To run the logon manager, you must define the logon manager program ELMNGR as nonswappable with a protection key value of 6. To do this, update the MVS program properties table. Refer to SCHEDxx macroinstruction in *z/OS MVS Initialization and Tuning Guide* and *z/OS MVS Initialization and Tuning Reference*.

Starting the logon manager

To start the logon manager, issue the START command `START procname`, where *procname* is the name of the logon manager JCL procedure. The logon manager can also run as a job. If you want to run it as a job, modify the logon manager JCL and the start procedure to conform to your installation standards for such jobs.

You must place JCL for the logon manager in SYS1.PROCLIB or an equivalent library with a member name *procname* of your choice. Following is a sample of the required JCL:

```
//name     PROC MEMBER=partitioned data set member name,
               MAXSUBA=value
//GO       EXEC PGM=ELMNGR,PARM='MEMBER=&MEMBER,MAXSUBA=&MAXSUBA'
//ELMDEFDS DD   DSN=partitioned dataset name,DISP=SHR
```

Note: The quotes are required around the PARM field of the EXEC statement.

name
Optional. If specified, it is usually the same as the member name *procname* of this JCL procedure.

ELMNGR
The name of the logon manager load module.

ELMDEFDS
The required DD name of the DD statement for the partitioned data set that contains the logon manager configuration definition.

MEMBER
The 1 to 8 character name of the member of the ELMDEFDS partitioned data set that contains the logon manager configuration definition.

MAXSUBA
Specifies the maximum number of network subareas that can be defined to the logon manager. Valid values are in the range 1 – 511. The default is 1. The number specified must be at least as great as the number of network subareas defined in the logon manager configuration definition.

Defining the logon manager

This section describes how to define the logon manager and includes a description of a typical logon manager configuration.

Note: You can also define TPF as a type 5 data host, rather than a type 2.1 physical unit. As a data host, TPF supports only fixed pacing window sizes,

virtual route 0 (VR=0), and transmission priority 2 (TP=2). The TPF data host must reside in its own SNI network with VTAM in another host acting as the gateway VTAM. In this type of configuration, you need to code the TPF application programs as dependent logical units. These are the only resources the TPF data host can own. When TPF is a type 5 data host, logon manager should not be used.

Sample logon manager configuration

Figure 167 shows a logon manager configuration consisting of the following:

- TPF with control point LUs
- TPF defined as a type 2.1 PU
- VTAM with the logon manager
- Dependent SLUs

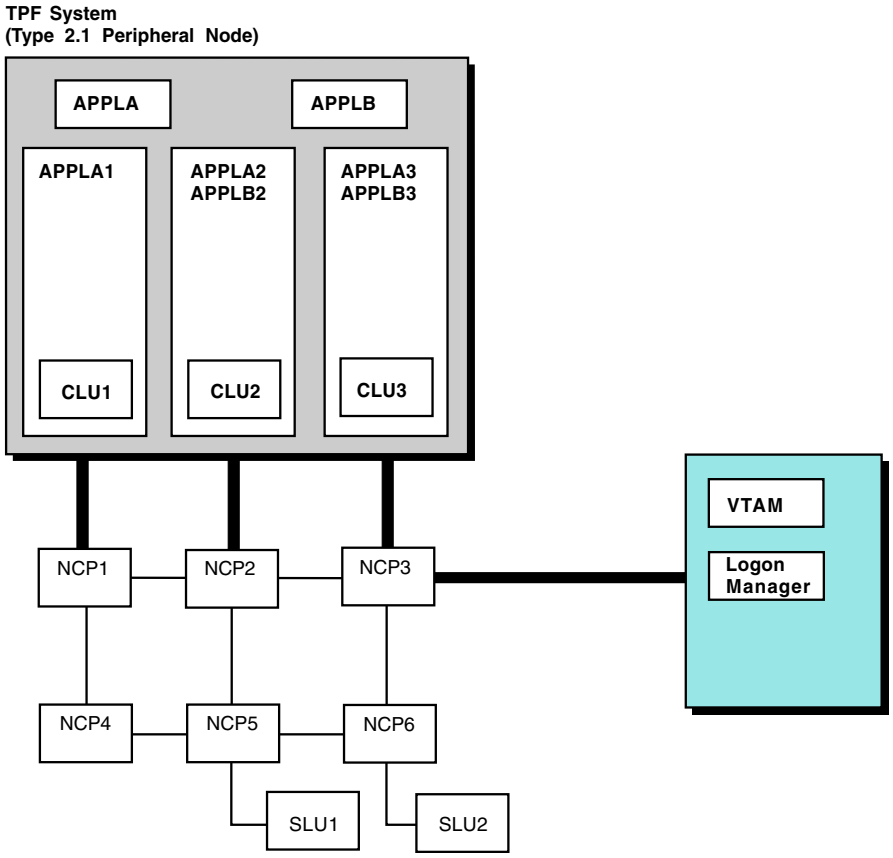


Figure 167. Sample Logon Manager Configuration

NCP1 through NCP3 are NCP Version 4 Release 3 subarea nodes, which are channel-attached to TPF. NCP3 is also channel-attached to VTAM. NCP1 through NCP6 are link-attached to each other. NCP4 through NCP6 do not have to be NCP Version 4 Release 3 or subsequent releases.

CLU1 through CLU3 are control logical units (CLUs). They manage the flow of transactions from their associated subarea nodes.

TPF is defined to VTAM as a type 2.1 peripheral node. Because the NCPs are channel-attached, TPF requires LNCTL=CA on the NCP GROUP definition statement. Interactions between logon manager and TPF occur in the form of control point LU to control point LU sessions.

APPLA and APPLB are TPF application programs. They are defined to VTAM as local applications.

Resources defined as independent LUs can initiate sessions with the TPF applications.

To take advantage of multiple-channel attachments into the subarea network, each application program (for example, APPLA) needs at least one LU per channel attachment. In this configuration, each application program requires three or more LUs (for example, APPLA1, APPLA2, APPLA3) to use all channels into the network. Because APPLB has only two LU definitions, it can use only two channels into the network.

SLU1 and SLU2 are defined to VTAM as dependent SLUs. They are link-connected to their associated subarea nodes and represent the remote terminal users of the TPF application programs.

You can define class of service (COS) tables for a subarea configuration. For information about COS tables, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

Defining the logon manager and TPF applications to VTAM

You need to define the logon manager and each TPF application program to VTAM using APPL definition statements. The following operands and operand values are required:

```
name     APPL  ACBNAME=acbname,
               AUTH=(ACQ,NOCNM,NOPASS,NOTSO,SPO),
               PARSESS=YES,
               SRBEXIT=NO,
               SONSCIP=YES,
               ENCR=NONE,
               VTAMFRR=NO,
               HAVAIL=YES
```

If *acbname* differs from the name *procname* of the logon manager JCL, you need to code the LMPROC definition statement in the logon manager configuration definition.

The following APPL definition statement operands are optional:

- PRTCT
- USSTAB
- VPACING

To use password protection for the logon manager, include the PRTCT operand on the APPL definition statement for the logon manager, and specify the same password as that on the LMPROC definition statement in the logon manager configuration definition. To use password protection for a TPF application program, include the PRTCT operand on the APPL definition statement for the TPF application program, and specify the same password as that on the LMAPPL definition statement in the logon manager configuration definition.

Defining the logon manager configuration

The logon manager configuration is defined by a sequence of source statements stored in a member of a partitioned data set with the member name that is specified in the logon manager JCL. You can write different types of source statements in any order in the configuration definition. Additional logon manager configuration definitions can be stored in members with other member names.

A logon manager configuration definition always contains one subarea (SA) definition statement identifying each network subarea. Other logon manager configuration definition statements are optional.

Subarea configuration

The system programmer responsible for identifying the network subareas and their adjacent subareas must provide a configuration of the network subareas. Each network subarea must be defined in the logon manager configuration definition using one of the following formats:

x The address for a subarea (SA).

a The address for an adjacent subarea (ADJSA). The subarea must be adjacent to subarea *x*.

You can specify multiple adjacent subarea addresses by enclosing them within parentheses and separating them with a comma.

All network subareas that you are defining to the logon manager must have their addresses appear in SA definition statements as either *x* or *a*.

Note: The value of MAXSUBA specified on the EXEC statement in the logon manager JCL must be at least as great as the number of different network subareas addressed as *x* or *a* in the SA definition statements in the logon manager configuration definition.

The SAs supplied by the system programmer are used by the logon manager to build a HOP table. The HOP table contains the minimum distance between subareas and is used for multitask support.

Class of service configuration

With logon manager, you can use class of service (COS) to distinguish between the needs of various sessions. Logon manager differentiates between session types based on the COS name. By designating different COS names for batch and interactive traffic (or any other preferred distinction), you can keep the various types of traffic from interfering with each other. Using this function, you define multiple views of the network that list preferred paths for different types of session traffic.

To specify the COS name, use the COS definition statement in the following format:

For example, in Figure 167 on page 747, NCP6 is only 1 hop away from NCP3, but is 2 hops away from NCP2. You can then specify that NCP3 is the preferred boundary function for any interactive sessions to the host and that NCP2 is the preferred route for batch traffic. To define the different routes to logon manager, you can define a number of Classes of Service, each with its view of the network. In the previous example you would code one Class of Service for interactive traffic, and one for batch.

To define these different views, use the COS definition statement. For example, to code the interactive view described previously, you would code the following:

```
COS=INTERACT
SA=6, ADJSA=(3)
```

Using this technique, NCP3 would be used for interactive traffic to the host.

The result is that each session is assigned to a TPF server based on predefined routing requirements, and enables you to assign suitable routes.

The SA definition statements that follow a COS definition statement apply only to the table representing that COS. If you code an SA definition statement without a COS definition statement preceding it, that SA definition statement is used as part of the default table.

If a COS name is not found for a COS that is requested, logon manager uses the default tables for the session:

- If no COS name is specified for the session, the default table is used.
- If a COS name is for the session, but cannot be found, the default table is used.
- If the default table is to be used, but is not defined to logon manager, an error is reported.

In the following example, coded for the configuration in Figure 167 on page 747, the first entry is used as part of the default table, and the other entries are used for interactive and batch-directed traffic.

```
SA=1, ADJSA=(2)
COS=INTERACT
SA=6, ADJSA=(3)
SA=5, ADJSA=(2)
COS=BATCH
SA=6, ADJSA=(2)
SA=5, ADJSA=(3)
```

Application passwords and minimum channel links

To define password protection for a TPF application program, include the following definition statement in the logon manager configuration definition. You can also use this definition statement to define the number of active channel links required before the application program is supported by the logon manager.

application program name
The name of the TPF application program. It must be the name of a real application program.

password
The 1 to 8 character password for the TPF application program. Passwords are optional.

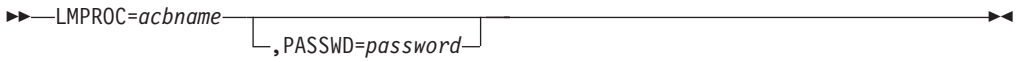
If password protection is required for a TPF application program, the password must be specified on the PASSWD operand on the LMAPPL definition statement and on the PRTCT operand on the VTAM APPL definition statement. There is no default password.

value
Defines the minimum number of channel links that must be active before the application program can engage in a session. Valid values are int the range 1 – 255. The default is 1. This definition is optional.

If all application programs on the TPF have the same password and the same required number of active channel links, you can use the following definition statement:

Logon manager procedure

If the logon manager requires password protection or if the logon manager ACBNAME on the APPL definition statement differs from the MVS *procname* or *jobname*, include the following definition statement in the logon manager configuration definition:



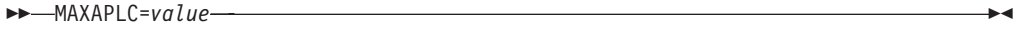
acbname
The 1 to 8 character name of the logon manager's ACB. If you do not supply the LMPROC definition statement, the MVS *procname* or *jobname* is used.

password
The 1 to 8 character password for the logon manager. The logon manager uses this password when opening the ACB for the *procname* or *jobname*. The logon manager password is optional.

If password protection is required for the logon manager, the password must be specified on the PASSWD operand on the LMPROC definition statement and on the PRTCT operand on the VTAM APPL definition statement. There is no default password.

Number of TPF applications

To specify the number of TPF application programs that can be defined, include the following definition statement in the logon manager configuration definition:



value
The number of TPF application programs that can be defined. Valid values are in the range 1 – 4095. The default is 4. This definition is optional.

Maximum number of buffers

To specify the maximum number of buffers to allocate for each channel-attached control point LU, include the following definition statement in the logon manager configuration definition:

value

The maximum number of buffers to allocate for each control point LU. Valid values are in the range 2 – 65 535. The default is 30. This definition is optional.

Logon manager takes the value specified in the MAXBUF definition statement and multiplies it by the value specified in the CHANATT definition statement to determine the total number of buffers available to logon manager. There is no restriction to the number of buffers that can be used for a given control point LU as long as there are free buffers available.

Maximum number of subareas

To specify the maximum number of channel-attached subareas, include the following definition statement in the logon manager configuration definition:

value

The maximum number of channel-attached subareas. This number is used to determine the maximum number of channel-attached control point LUs. Valid values are in the range 1 – 255. The default is 4. This definition is optional.

Reserved keywords

The following keywords are reserved for use with the TPF logon manager:

- ADJSA
- ALL
- APPL
- CHANATT
- CLU
- HELP
- ID
- INFO
- LMAPPL
- LMPROC
- MAXAPLC
- MAXSUBA
- MEMBER
- MINLINK
- PASSWD
- PEND
- SA
- STOP

Monitoring logon manager resources

The VTAM logon manager application offers commands to monitor the status of the logon manager and the resources under control of the logon manager.

By using the logon manager MODIFY command, you can display information about logon manager resources (TPF applications, CLUs, and LUs), including session status. For more information on the logon manager operator commands, refer to *z/OS Communications Server: SNA Operation*.

Halting the logon manager

The logon manager functions as a stand-alone VTAM application and, thus, can be stopped independently of VTAM. The logon manager MODIFY STOP command stops logon manager processing. For more information on the logon manager operator commands, refer to *z/OS Communications Server: SNA Operation*. When the command is issued, the logon manager shuts down its subtasks, closes its ACBs, and returns control to MVS. Current sessions are allowed to end, but new sessions are not permitted.

The VTAM HALT, HALT QUICK, and CANCEL commands terminate the logon manager, as they do any VTAM application.

Appendix E. Cryptographic keys

If you use the VTAM data encryption facility, you need to file cryptographic keys on the cryptographic key data set at the appropriate host processors. For information about which hosts require cryptographic keys, see "Cryptography facility" on page 386.

This appendix describes how to file these keys for different types of cryptographic facilities for both single-domain and multiple-domain sessions. The available cryptographic services are:

- z/OS Integrated Cryptographic Service Facility (ICSF) and S/390 or zSeries Cryptographic Co-Processor

  ICSF is a licensed program that runs under MVS and provides access to the hardware cryptographic feature for programming applications. The combination of the hardware cryptographic feature and ICSF provides secure high-speed cryptographic services.
- Other PCF/CUSP or Common Cryptographic Architecture (CAA) compatible cryptographic products

Note: Triple-DES 24-byte encryption requires the use of the ENCRYPTN=CCA start option and that the Common Cryptographic Architecture (CCA) product is present. Otherwise, sessions that require triple-DES 24-byte encryption will fail. CCA defines a set of cryptographic functions, external interfaces, and a set of key management rules that provide a consistent, end-to-end cryptographic architecture across different IBM platforms.

The following references are used with compatible cryptographic products:

PCF/CUSP
Refers to any cryptographic product that is compatible with PCF/CUSP.

CCA Refers to any cryptographic product that is compatible with Common Cryptographic Architecture (CCA).

Notes:

1. If ICSF/MVS runs in CUSP mode, use the information for PCF/CUSP.
2. When using ICSF in PCF compatibility mode and migrating from an existing PCF cryptographic key data set (CKDS), an importer key with a key value of the PCF master key value must be included. Use the PCF master key 8 bytes twice to create the ICSF 16–byte key. Refer to the ICSF publications for additional information.

   Specific commands and control statements for key input may differ by product.

For more information on establishing cryptographic sessions, refer to *z/OS Communications Server: SNA Programming*.

Filing SLU keys for single-domain cryptographic sessions

The process for filing SLU keys for single-domain cryptographic sessions varies, depending on which cryptographic service you use.

© Copyright IBM Corp. 1984, 2011

Single-domain cryptographic sessions that use PCF/CUSP

To allow cryptographic sessions that use PCF/CUSP to be established within a single domain, install the PCF/CUSP compatible cryptographic product.

Use the PCF/CUSP compatible cryptographic product to file secondary logical unit (SLU) keys on the cryptographic key data set (CKDS) as follows:

- For each device-type LU that is to be used as the secondary end of a cryptographic session, code the following:

  ```
  LOCAL luname
  ```

  where *luname* is the name of the LU.

  This LOCAL statement generates an SLU key for the LU and adds it to the CKDS enciphered under the first variant of the host master key. It also returns a clear SLU key. Enter the clear SLU key into the device.

  The CKEYNAME operand on the LU definition statement can be used to reduce definition statements coded in the CDKS. Multiple devices can use the same LOCAL statement in the CDKS by specifying the same value for the CKEYNAME operand.

  ```
  LOCAL ckeyname
  ```

  where the value specified in the CKEYNAME operand is the same as the *ckeyname*.

  Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for information on coding the CKEYNAME definition statement.
- For each VTAM application program that is to be the secondary end of a cryptographic session, code the following:

  ```
  REMOTE name
  ```

  where *name* is the name of the application program.

  This REMOTE statement generates an SLU key for the application program and adds it to the CKDS enciphered under the second variant of the host master key.

Single-domain cryptographic sessions that use ICSF/MVS

To allow cryptographic sessions that use ICSF/MVS to be established within a single domain, install ICSF/MVS.

Use ICSF/MVS to file SLU keys on the CKDS as follows:

- For each device-type LU that is to be used as the secondary end of a cryptographic session, code the following, where *name* is the name of the LU:

  ```
  ADD LABEL(name) TYPE(EXPORTER) CLEAR SINGLE
  ```

  Note: SINGLE is needed only if the cryptographic support of the device-type LU does not support double length keys.

  This key generation utility program (KGUP) statement generates an exportable SLU key for the LU and adds it to the CKDS that is enciphered under the host master key. It also returns a clear SLU key. Enter the clear SLU key into the device.
- For each VTAM application program that is to be used as the secondary end of a cryptographic session, code the following, where *name* is the name of the application program:

  ```
  ADD LABEL(name) TYPE(IMPORTER) CLEAR SINGLE
  ```

Note: SINGLE is needed only if the cryptographic support of the device-type LU does not support double length keys.

This KGUP statement generates an importable SLU key for the application program and adds it to the CKDS that is enciphered under the host master key.

Filing CDRM keys for cross-domain cryptographic sessions

The process for filing SLU keys for cross-domain cryptographic sessions varies, depending on which cryptographic service you use and whether both hosts use the same cryptographic service.

Note: CCA processing must be present to use triple-DES encryption. The ENCRYPTN=CCA start option is also required for triple-DES encryption. PCF/CUSP compatible cryptographic products typically support DES but do not support triple-DES.

Cross-domain cryptographic sessions in which both hosts use PCF/CUSP

To allow cross-domain cryptographic sessions to be established, file SLU keys for each domain as described in "Single-domain cryptographic sessions that use PCF/CUSP" on page 756. Start the PCF/CUSP compatible cryptographic product before you activate the external CDRM for which cross statements have been filed in the cryptographic keys data set. Use the PCF/CUSP compatible cryptographic product to file cross-domain keys on the cryptographic key data set (CKDS) at each host processor as follows:

- For each pair of host processors (HOST1 and HOST2) that are to have cross-domain cryptographic sessions between their domains, code the following at HOST1, where *name* is the name of HOST2 CDRM:

  ```
  CROSS name
  ```

  This CROSS statement generates two cross-domain keys, one defined as local and the other defined as remote.[5] It stores the local cross-domain key in the CKDS enciphered under the first variant of HOST1 host master key and stores the remote cross-domain key in the CKDS enciphered under the second variant of HOST1 host master key. Both of these keys are associated with the name of HOST2 CDRM. This CROSS statement also returns clear copies of the two keys.
- The cross-domain keys generated at HOST1 must be used at HOST2 and supplied as input to PCF/CUSP in a CROSS statement.

  When HOST2 has PCF:

  ```
  CROSS name,KEYLOC=x,KEYREM=y,ADD
  ```

  When HOST2 has CUSP:

  ```
  CROSS name,KEYLOC=x,IKEYLOC=xx,KEYREM=y,IKEYREM=yy,ADD
  ```

  where:

  name is the name of HOST1 CDRM
  x is the clear remote cross-domain key from HOST1
  xx is the second half of double key value from HOST1 remote key
  y is the clear local cross-domain key from HOST1
  yy is the second half of double key value from HOST1 local key

  This CROSS statement does the following:

5. The terms *local* and *remote* in reference to the keys used by the CROSS statement do not have the same meaning here as they do in other contexts in this document. For more information on PCF, refer to the *z/OS Cryptographic Services ICSF Application Programmer's Guide*. For more information on CUSP, refer to the *z/OS Cryptographic Services ICSF Administrator's Guide*.

- Adds the two cross-domain keys to HOST2 CKDS
- Associates the two keys with the name of HOST1 CDRM
- Reverses the local/remote relationship of the two keys
- Enciphers key *x* and its intermediate key *xx* under the first variant of HOST2 host master key, and key *y* and its intermediate key *yy* under the second variant of HOST2 host master key

Figure 168 illustrates the possible coding for cross-domain cryptographic sessions where both hosts use PCF/CUSP. APPL2A can initiate a cryptographic session with LU1A. APPL2A can be the primary LU (PLU) in the cryptographic session with APPL1A. However, APPL2A cannot be the secondary LU in a cryptographic session with APPL1A because there is no REMOTE statement for APPL2A in VTAM2.

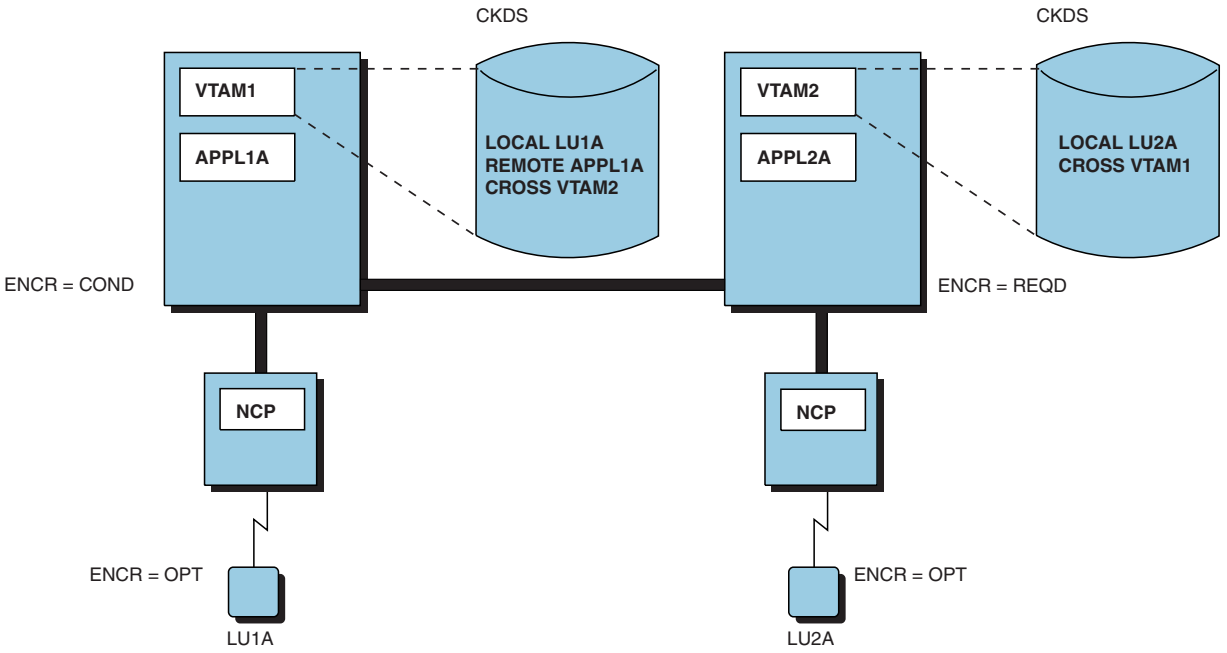


Figure 168. Cryptography in multiple-domain environment (Both hosts use PCF/CUSP)

For the configuration in Figure 168 to have any encrypted sessions, start cryptography in both hosts before activating a session.

Cross-domain cryptographic sessions in which both hosts use ICSF/MVS

To allow cross-domain cryptographic sessions to be established, file SLU keys for each domain as described in "Single-domain cryptographic sessions that use ICSF/MVS" on page 756. Then use ICSF/MVS to file cross-domain keys on the cryptographic key data set (CKDS) at each host processor as described in the following paragraphs.

A complementary pair of exporter and importer keys must be generated for the two host processors (HOST1 and HOST2). To allow for cross-domain cryptographic sessions between hosts using the ICSF/MVS cryptographic service, perform the following steps:

- At HOST1, code the following key generation utility program (KGUP) statements:

```
ADD LABEL(name) TYPE(EXPORTER) CLEAR
ADD LABEL(name) TYPE(IMPORTER) CLEAR
```

where *name* is the name of HOST2 CDRM

These KGUP statements generate an exporter key for HOST1 to use to encrypt session keys sent from HOST1 to HOST2 and an importer key for HOST1 to decrypt session keys sent from HOST2 to HOST1. The exporter and importer keys are placed in HOST1 CKDS. The statements also cause the clear key values to be placed in the KGUP key output data set. Assume clear key value X,XX for the exporter key and clear key value Y,YY for the importer key. These values are to be used on HOST2 to build complement key control statements for these keys.

- At HOST2, code the following KGUP statements using the clear key values from HOST1:

  ```
  ADD LABEL(name1) TYPE(IMPORTER) CLEAR KEY(x,xx)
  ADD LABEL(name1) TYPE(EXPORTER) CLEAR KEY(y,yy)
  ```

  where:

  name1 is the name of HOST1 CDRM.

  x is the first half of the double-length clear key value of HOST2 exporter key generated in HOST1.

  xx is the second half of the double-length clear key value from HOST2 exporter key generated in HOST1.

  y is the first half of the double-length clear key value from HOST2 importer key generated in HOST1.

  yy is the second half of the double-length clear key value from HOST2 importer key generated in HOST1.

  These control statements place these keys in HOST2 CKDS.

Figure 169 on page 760 illustrates the possible coding for cross-domain cryptographic sessions where both hosts use ICSF/MVS. APPL2A can initiate a cryptographic session with LU1A. APPL1A can initiate a cryptographic session with LU2A. APPL2A can be the PLU in a cryptographic session with APPL1A. However, APPL2A cannot be the SLU in a session with APPL1A because there is no TYPE(IMPORTER) statement coded in VTAM2, and APPL2A requires cryptography when it is the SLU.

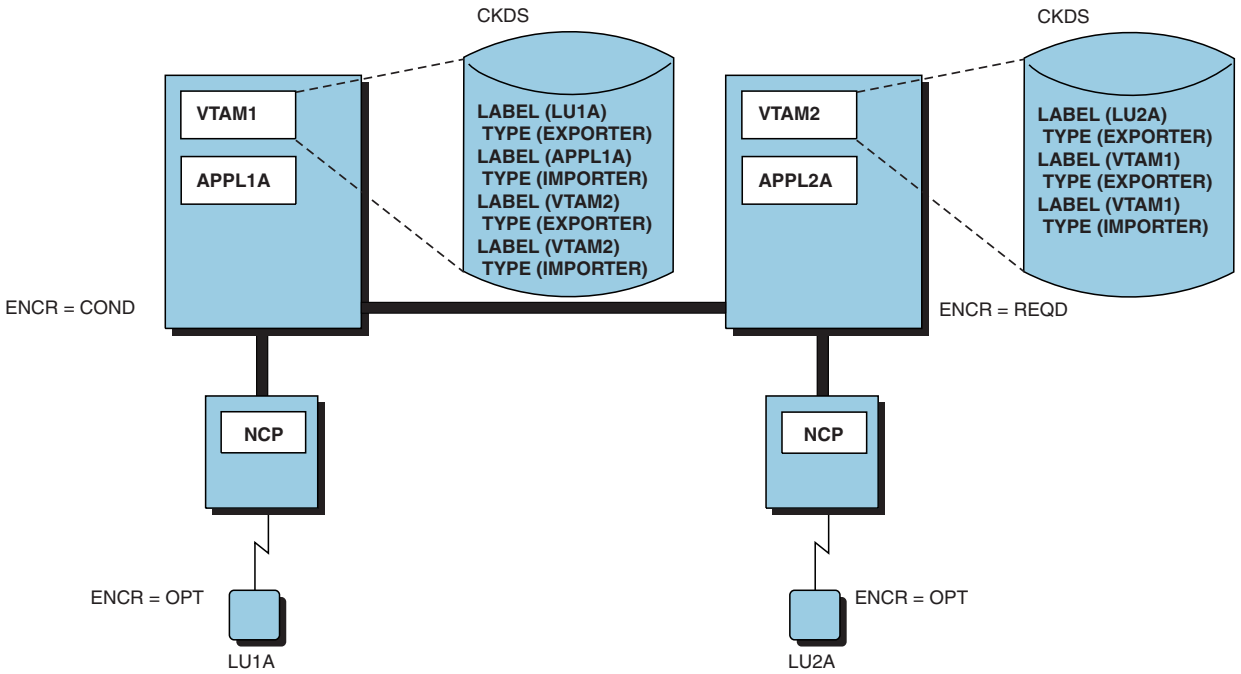


Figure 169. Cryptography in multiple-domain environment (Both hosts use ICSF/MVS)

For the configuration in Figure 169 to have any encrypted sessions, start cryptography in both hosts before activating a session.

For more information on ICSF/MVS, refer to the *z/OS Cryptographic Services ICSF Administrator's Guide*.

Cross-domain cryptographic sessions in which the hosts use different cryptographic services

In this description of possible coding, assume that ICSF/MVS is installed in HOST1 and a PCF/CUSP compatible cryptographic product is installed in HOST2. A complementary pair of key-encrypting keys must be generated for HOST1 and HOST2. To allow for cross-domain cryptographic sessions between the hosts, perform one of the following, depending upon where key processing is to be performed first:

- When key processing is to be performed from HOST2 first, perform these steps:
  1. At HOST2 where the PCF/CUSP compatible cryptographic product is installed, code the following key generation utility program (KGUP) statement:

     `CROSS name`

     Where *name* is the name of HOST1 CDRM.

     This PCF/CUSP KGUP statement generates a LOCAL and REMOTE key pair and places them in HOST2 cryptographic key data set (CKDS). A copy of the key pair generated will also appear in the KGUP SYSPRINT output listing. This gives HOST2 the capability to send encrypted session keys to HOST1.
  2. At HOST1, use the PCF/CUSP compatible cryptographic product SYSPRINT output from the cross key pair to create the following ICSF control statement for the appropriate environment.

     When HOST2 has PCF:

     `ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV KEY(key-value)`

When HOST2 has CUSP:

```
ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV KEY(key-value,ikey)
```

where:

name is the name of HOST2 CDRM.
key-value is the key value from HOST2 of the local key.
ikey is the intermediate key value from HOST2 of the intermediate local key.

This ICSF KGUP statement places the importer key, which is the same clear value as HOST2 LOCAL key in HOST1 CKDS. This allows HOST1 to decrypt session keys sent from HOST2.

3. At HOST1, use the PCF/CUSP compatible cryptographic product SYSPRINT output from the cross-key pair to create the following ICSF control statement for the appropriate environment.

   When HOST2 has PCF:

   ```
   ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV KEY(key-value)
   ```

   When HOST2 has CUSP:

   ```
   ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV KEY(key-value,ikey)
   ```

   where:

   name is the name of HOST2 CDRM.
   key-value is the key value from HOST2 of the remote key.
   ikey is the intermediate key value from HOST2 of the intermediate remote key.

   This ICSF KGUP statement places the exporter key, which has the same clear value as HOST2 REMOTE key, in HOST1 CKDS. This allows HOST1 to send encrypted session keys to HOST2.

- When key processing is to be performed from HOST1 first, perform these steps:

1. At HOST1 where ICSF/MVS is installed, code the appropriate key generation utility program (KGUP) statements listed below for the environment in HOST2:

   When HOST2 has PCF:

   ```
   ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV SINGLE
   ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV SINGLE
   ```

   When HOST2 has CUSP:

   ```
   ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV
   ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV
   ```

   where *name* is the name of HOST2 CDRM.

   These statements cause two keys to be generated in HOST1 CKDS. One key, the exporter, is to be used when sending session keys to HOST2. The other key, the importer, is to be used to decrypt session keys sent from HOST2.

   The complement clear key values of both the exporter and importer are placed in the ICSF key output data set. Use these values as the clear key values for the CROSS statements on HOST2.

2. At HOST2 where the PCF/CUSP compatible cryptographic product is installed, code the appropriate key generation utility program (KGUP) statement listed below for the environment in HOST2.

   When HOST2 has PCF:

   ```
   CROSS name,KEYLOC=x,KEYREM=y,ADD
   ```

   When HOST2 has CUSP:

   ```
   CROSS name,KEYLOC=x,IKEYLOC=xx,KEYREM=y,IKEYREM=yy,ADD
   ```

   where:

name is the name of HOST1 CDRM.
x is the key value from HOST1 of the IMPORTER key.
y is the key value from HOST1 of the EXPORTER key.
xx is the second half of double key value from HOST1 IMPORTER key.
yy is the second half of double key value from HOST1 EXPORTER key.

Figure 170 illustrates the possible coding for cross-domain cryptographic sessions where HOST1 uses ICSF/MVS and HOST2 uses PCF/CUSP. APPL2A can initiate a cryptographic session with LU1A. APPL1A can initiate a cryptographic session with LU2A. APPL2A can initiate a cryptographic session with APPL1A. However, APPL1A cannot be a PLU in a session with APPL2A because there is no TYPE(IMPORTER) statement coded in VTAM2, and APPL2A requires cryptography.

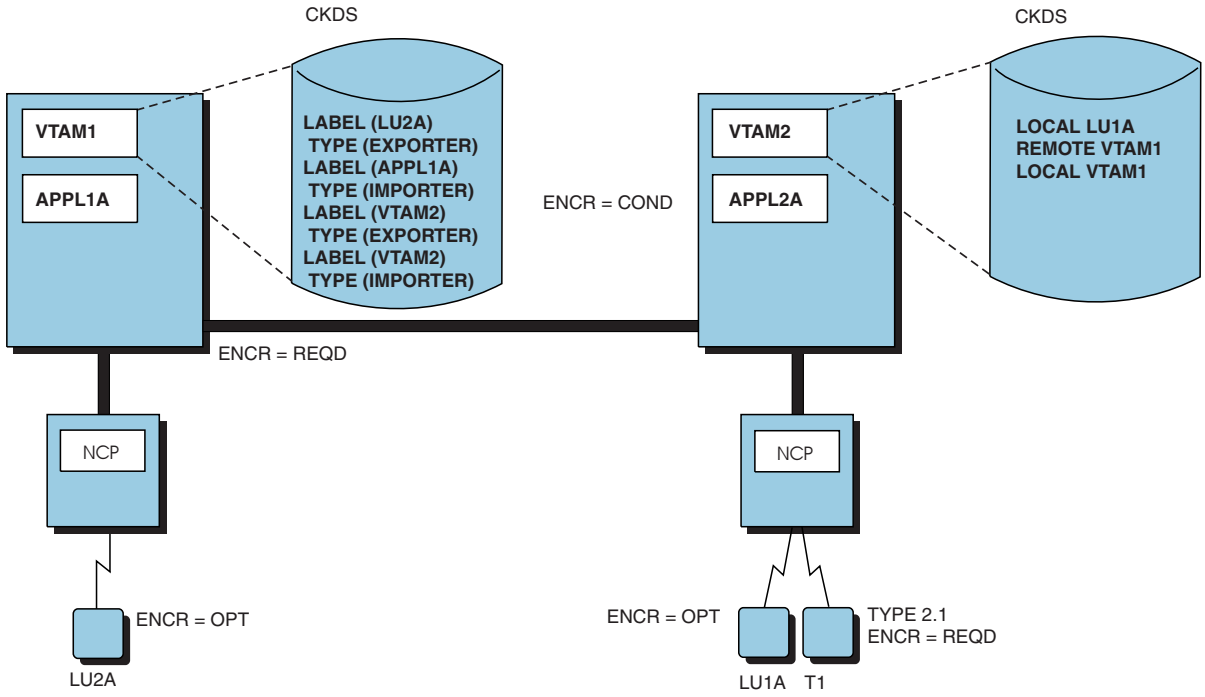


Figure 170. Cryptography in multiple-domain environment (Hosts use different cryptographic services)

For the configuration in Figure 170 to have any encrypted sessions, start cryptography in both hosts before activating a session.

Changing the cryptographic capability of a logical unit

If the VTAM encryption facility is installed, the cryptographic capability of the logical unit can be changed for future sessions by using the MODIFY SECURITY or MODIFY ENCR command. The ENCRTYPE keyword can only be changed using the MODIFY SECURITY command. For the syntax of the command, refer to *z/OS Communications Server: SNA Operation*.

In addition, an alternate LU key-encrypting key can be used while the master key is being changed in the CKDS. See "Changing cryptographic keys dynamically" on page 394 for additional information.

Appendix F. Command lists: Dynamic configuration of channel-attached devices

The IBM-supplied command lists are used to illustrate dynamic configuration of channel-attached devices (dynamic I/O). If you want to modify the command lists, this section describes how to call the command lists and what output the command lists return to assist you during modification. For general information, see "Dynamic configuration of channel-attached devices" on page 230.

ISTDINFO: VTAM device information services

The ISTDINFO command list accepts requests from a NetView REXX command list and uses the information to retrieve information about VTAM devices. ISTDINFO then returns the device information to the caller. You can call the ISTDINFO command list using following format:

```
ISTDINFO  DEV_INFO, dev_string
```

The comma separating the parameters is mandatory. The parameters provide the following information:

DEV_INFO
The character string "DEV_INFO", in either uppercase or lowercase.

dev_string
A character-string constant representing a hexadecimal device address. Device addresses consist of 1–4 hexadecimal characters (for example, X'F12').

Dependencies and restrictions

If you are using the ISTDINFO command list, you should be aware of the following dependencies and restrictions:

- You can only IPL the configuration using an HCD I/O Definition (IODF). You cannot use MVS/CP to define your configuration. The LOADxx parmlib member, which contains the IPL parameters, must have an IODF statement coded, not an MVS/CP statement.
- You can define only the following devices:
  - Local SNA clusters
  - Communication controllers
- The devices must be defined to MVS using the IBM-supplied unit information modules.
- Complete device information is returned only when the hardware is self-describing.

Output variable

The ISTDINFO command list returns information in the REXX special variable RESULT. RESULT is a string containing the following tokens:

return_status
device_address
device_class
self_describing
device_type
device_model

© Copyright IBM Corp. 1984, 2011

defined_type
generic_family
PU_type
TG_number
IPL_online
current_online
allocate
config_token
maxbufru
maxdata
logical_terminals
owner
ncp_pu_type
ncp_timeout
ncp_delay
ncp_maxbufru
ncp_inbfrs
ncp_maxdata
ncp_transfr
ncp_bfrpad
ncp_TPS_feature

Output tokens

The following describes the information contained in the tokens:

return_status Returned status from the device information services routine:

- **0** Device specific information returned
- **2** Not valid option string specified for parameter one
- **3** Not valid device address specified for parameter two
- **4** No VTAM supported devices
- **5** Unsupported operating system environment
- **101** Internal error 101
- **102** Internal error 102
- **201** Internal error 201
- **301** Internal error 301
- **302** Internal error 302
- **303** Internal error 303
- **304** Internal error 304
- **305** Internal error 305
- **306** Internal error 306
- **307** Internal error 307
- **308** Internal error 308
- **309** Internal error 309
- **310** Internal error 310
- **311** Internal error 311
- **401** Internal error 401
- **501** Internal error 501
- **502** Internal error 502
- **503** Internal error 503
- **504** Internal error 504
- **505** Internal error 505
- **601** Internal error 601
- **602** Internal error 602
- **603** Internal error 603
- **604** Internal error 604
- **605** Internal error 605

606 internal error 606
607 Internal error 607
608 Internal error 608
609 Internal error 609
610 Internal error 610
611 Internal error 611

device_address
Address of the device associated with the returned information. This is not necessarily the same as the input *dev_string*.

device_class
Device class stored in the unit control block (for example, COMM, DISP or CTC).

self_describing
An indication of whether the device is self-describing (YES or NO).

device_type
Actual device type returned to MVS during VARY ONLINE processing, or *N/A* if not applicable.

device_model
Model number for this device, or *N/A* if not applicable.

defined_type
Defined device type specified by the user. For example, 3705 is specified for a 3705, 3725, 3745, or *N/A* if not applicable. This should always be defined.

generic_family
Generic family of this device specified by the device: 3745 for 3745, or *N/A* if not applicable.

PU_type
Physical unit type of the device (for example, 2 or 21 or 4, or *N/A* if not applicable).

TG_number
Transmission group number of the device: 0–255, or *N/A* if not applicable.

IPL_online
Status of the device at IPL: ONLINE, OFFLINE, or *N/A* if not applicable.

current_online
Status of the device currently: ONLINE or OFFLINE.

allocate
Status of allocation: ALLOCATED or UNALLOCATED.

config_token
Token representing the current I/O configuration. The actual I/O configuration is not printable in EBCDIC.

maxbufru
Maximum number of buffers to be set aside by the host, or *N/A* if not applicable.

maxdata
Maximum number of bytes the PU can receive in one PIU, or *N/A* if not applicable.

logical_terminals
Character string of 256 ones and zeros. A 1 indicates a logical terminal exists at that offset, or *N/A* if not applicable. This string defines the number of logical units to be generated and the local address for each.

owner Value entered into HCD for the subsystem or access

method using this device, or *N/A* if not applicable. Any embedded blanks in the owner string are replaced with the underscore (_) character.

ncp_pu_type
Value entered into HCD for PU type, or *N/A* if not applicable.

ncp_timeout
Value entered into HCD for attention response timeout for the device, or *N/A* if not applicable.

ncp_delay
Value entered into HCD for maximum delay before attention to channel, or *N/A* if not applicable.

ncp_maxbufru
Value entered into HCD for maximum number of host buffers allocated for the device, or *N/A* if not applicable.

ncp_inbfrs
Value entered into HCD for initial allocation of control buffers for the device, or *N/A* if not applicable.

ncp_maxdata
Value entered into HCD for the maximum number of bytes the PU can receive in one data transfer for the device, or *N/A* if not applicable.

ncp_transfr
Value entered into HCD for the number of NCP buffers for MAXDATA transfer for the device, or *N/A* if not applicable.

ncp_bfrpad
Value entered into HCD for number of pad characters to the access method for the device, or *N/A* if not applicable.

ncp_TPS_feature
Value entered into HCD for mode of the two-processor switch feature for the device: TPS, TCS, NONE, or *N/A* if not applicable.

ISTDEFIN: VTAM device information services

The ISTDEFIN command list accepts one or more device addresses, gets information about each device, defines it to VTAM, ensures that each device is online, and activates the VTAM resource definitions. You can call the ISTDEFIN command list using the following format:

```
Call ISTDEFIN func_code dev_str
```

The parameters provide the following information:

func_code
One of the following function codes that indicate how the input device addresses correlate to the device processed by the command list:

1 Process only device addresses passed to this command list.

2 Use input device address as a base address to locate communication, display or CTC device to process.

dev_str
A string of one or more hexadecimal device addresses separated by blanks.

For example, if you wanted to call the ISTDEFIN command list with a function code of 1 and 3 device addresses, X'00A', X'123', and X'FE3', you would send the following:

```
ISTDEFIN 1 A 123 FE3
```

Dependencies and restrictions

If you are using the ISTDEFIN command list, you should be aware of the following dependencies and restrictions:

- You need NetView 1.3 or later releases or a product providing a similar interface and services.
- ISTDEFIN must execute in the NetView address space.
- You can define only the following devices:
  - Local SNA clusters
  - Communications controllers
- You need to concatenate USER1.AUTO.VTAMLST with SYS1.VTAMLST.
- You can define NCP major nodes only if they are channel-attached.
- The IBM-supplied naming convention supports only 36 subarea addresses.
- Error messages for the VARY NET,ACT command are diagnosed only to the extent that syntax errors in the resource definitions generated by this sample are identified.

Output variable

The ISTDEFIN command list returns information in the REXX special variable RESULT. RESULT is a string containing the following tokens:

```
return_status
device_address1
device_rc1
device_address2
device_rc2
⋮
device_addressn
device_rcn
```

where *n* is the number of devices processed by the command list.

Output tokens

The following describes the information contained in the tokens:

return_status
Highest return code encountered while processing the device string.

device_address$_1$
Actual device address processed.

device_rc$_2$
Return code that resulted from processing device_address$_i$. The possible values are:

0 device successfully defined and activated
1 device address not found
2 not valid option string (parameter 1) specified on ISTDINFO call
3 not valid device address (parameter 2) specified on ISTDINFO call
4 device is currently allocated
5 operating environment not supported
6 operator canceled processing for device
7 timer expired while varying device online

8 error occurred while updating USER1.AUTO.VTAMLST
9 VARY ONLINE command failed for device
10 PU type not supported by VTAM
11 timer expired during VARY ACTIVATE
12 error activating device
14 device not defined to come online at IPL
15 device does not support sysdef automation
101 ISTDINFO internal error 101
102 ISTDINFO internal error 102
201 ISTDINFO internal error 201
301 ISTDINFO internal error 301
302 ISTDINFO internal error 302
303 ISTDINFO internal error 303
304 ISTDINFO internal error 304
305 ISTDINFO internal error 305
306 ISTDINFO internal error 306
307 ISTDINFO internal error 307
308 ISTDINFO internal error 308
309 ISTDINFO internal error 309
310 ISTDINFO internal error 310
311 ISTDINFO internal error 311
401 ISTDINFO internal error 401
501 ISTDINFO internal error 501
502 ISTDINFO internal error 502
503 ISTDINFO internal error 503
504 ISTDINFO internal error 504
505 ISTDINFO internal error 505
601 ISTDINFO internal error 601
602 ISTDINFO internal error 602
603 ISTDINFO internal error 603
604 ISTDINFO internal error 604
605 ISTDINFO internal error 605
606 ISTDINFO internal error 606
607 ISTDINFO internal error 607
608 ISTDINFO internal error 608
609 ISTDINFO internal error 609
610 ISTDINFO internal error 610
611 ISTDINFO internal error 611

Appendix G. Message translation using the MVS Message Service

Multicultural support uses the MVS Message Service (MMS) for the translation of VTAM messages. This section describes how MMS is used to translate VTAM messages. For an overview of multicultural support, see "Multicultural support" on page 288. The following topics are discussed in this section:

- Overview of MMS support
- Internal translation
- External translation
- Skeleton file use

Overview of MMS support

The MVS Message Service translates VTAM messages using message skeleton files for particular languages. MMS finds a skeleton file matching the language requested and then searches for the message and returns the text from the skeleton file.

VTAM provides a base U.S. English message skeleton file named ISTVTMEU. This file contains all VTAM messages including USS, TSO/VTAM, and logon manager messages. The way in which ISTVTMEU is used depends on whether the translation of the message occurs internally or externally to VTAM.

Internal translation

VTAM uses internal translation to translate USS and TSO/VTAM end-user messages. These messages must be translated internal to VTAM so that they can be sent on either an SSCP-LU (for USS) or LU-LU session (for TSO/VTAM). For end-user messages, ISTVTMEU defines the U.S. English text that should be used for these messages. This text can be changed, providing a function similar to USS tables.

Selecting internal translation

To use internal translation:

1. To support a language other than U.S. English, first define a language skeleton file to MMS. The language skeleton file must have the language code or language name that will be specified by the logical unit. If the required language skeleton file does not exist when the USS command is entered and the MMS is active, the USS command will be rejected. For TSO/VTAM, default U.S. English messages are issued if the required language skeleton file does not exist.
2. An LU must explicitly request that translation be performed before messages will be translated internally.
   - For USS, the LANG operand on USS commands determines the language.
   - For TSO/VTAM, the language can be determined from the LANG or LANGTAB operands (the language information is provided on the CINIT) or by the TSO PROFILE command with the PLANG operand.

© Copyright IBM Corp. 1984, 2011

Defining messages for internal translation

The IBM-supplied file, ISTVTMEU, contains messages for internal and external translation. The end-user messages defined in ISTVTMEU for internal translation are USSMSG01 through USSMSG13 for USS and IKT00201I through IKT00405I for TSO/VTAM. For example, USS message 1 is defined as follows:

```
USSMSG01          INVALID &1. COMMAND SYNTAX    &TIME.
```

You can change the wording of this message to different wording if desired. The &1. is an example of a substitution token and is used here to identify where the incorrect command name is to be placed in the message. Additional substitution tokens are &2., &3. and so on. Also, the &TIME., &DATE., &LUNAME., and &NETID. substitution tokens are used for the time, date, LU name, and network ID. If you want to provide USS messages in other languages using MMS, another skeleton file can be created for the desired language with the desired message text. For example, you could code the following in a new file named ISTVTFRA with the text in French:

```
USSMSG01          THE SYNTAX FOR COMMAND &1. IS INVALID
```

Then, if the user specified LANG(FRA) on a USS command, the above message text would be provided in the case of a USS command syntax error. The above message could also be defined as a multiline message, and, when translation occurs, all of the lines in the translated message are written to the LU. There is a 23-line limit on the number of lines in a translated message for USS and an 8-line limit for TSO/VTAM.

USS messages defined with MMS skeleton files must follow the same restrictions as USS messages defined with USS tables. Only a limited character set is available to SNA logical units on the SSCP-LU session.

For TSO/VTAM messages, any character set supported by the end-user terminal can be used, including double byte-character sets. Using incompatible character sets and code pages can result in erroneous output or I/O errors.

External translation

External translation occurs outside VTAM and is only available to authorized TSO/E 2.2 users using multiple console support (MCS) extended consoles.

Selecting external translation

In external translation, the language for the extended console is determined by the user profile, which is set with the PLANG operand of the TSO PROFILE command.

When a language other than U.S. English is selected, the following occurs:

1. ISTVTMEU is used to parse the U.S. English message to determine the variable text.
2. MMS looks up the corresponding translated message and substitutes the variable text.

ISTVTMEU and ISTINCNO must match for external translation to work correctly. It is recommended that you use the IBM-supplied copies of these files and that you do not use the USSTAB start option. Customization of messages can then be accomplished by changing the language skeleton files used by MMS.

Defining messages for external translation

Messages for external translation are also defined in ISTVTMEU. All of the messages for external translation are messages that are written to the system console. You should not change the U.S. English form of these messages in ISTVTMEU, because ISTVTMEU must correspond exactly with ISTINCNO for external translation to occur. For operator messages, you should use the IBM-supplied default table, ISTINCNO, rather than using a customized operator USS table.

For translation to other languages, provide a language skeleton file similar to ISTVTMEU with the translated text. For example, VTAM message IST001I is defined as follows in ISTVTMEU:

```
IST001I     001    VTAM START REJECTED - &1.
IST001I     002    &modnm. VTAM START REJECTED - &1.
```

The text of this message could be changed to French in a new skeleton file named ISTVTFRA. It is necessary to have two formats of this message: one for the message without the module name and one for the message with the module name. MMS will select the format that most closely matches the message received and perform the translation by selecting the corresponding message with the corresponding format number from the new French skeleton file.

For example, assume ISTVTFRA has the following definition for IST001I:

```
IST001I     001    &1. - REJECT START
IST001I     002    &modnm. &1. - REJECT START
```

With the above skeletons coded, a TSO/E 2.2 user using an MCS extended console with the primary language of FRA will have the following VTAM message translated as shown below:

```
Message issued by VTAM:
IST001I VTAM START REJECTED — TERMINATION IN PROGRESS

Translated message on MCS Extended console
IST001I TERMINATION IN PROGRESS — REJECT START
```

The translated messages can be in any language supported by MMS, including double-byte character set languages. The only restriction is that the terminal being used for a given language must support that language with the correct character set and code page, or erroneous output or I/O errors can occur. Translated messages can also be defined as multiline messages if more than one line is needed to express the error in the translation language.

Skeleton file use

ISTVTMEU, and any other translation language skeletons required by the installation, must be compiled by the MMS compiler before use. Also, MMS must be active before VTAM can use any of the skeleton files.

ISTVTMEU is shipped in library SYS1.SISTDAT2. This member can be copied to another partitioned data set where other MMS skeleton files are kept. Skeleton files for languages other than U.S. English must be provided by the installation. For information on compiling message skeleton files, refer to *z/OS MVS Planning: Operations*. For information on defining message skeleton files, refer to the *z/OS MVS Programming: Assembler Services Guide*.

Appendix H. Forcing an APPN route in a VTAM network

This appendix describes how to force the path of a route through your network. You might want to force a route if you want certain sessions to use a specific path rather than the default path.

If you use only the defaults, the nodes and links are considered equal and the route with the fewest nodes and links will have the least weight. The route with the least weight is selected as the route for the session.

For example, if all the defaults are used in the network shown in Figure 171 and an LU in NN1 requests a session with an LU in NN3, the route chosen for the session will go from NN1 through NN2 to NN3.

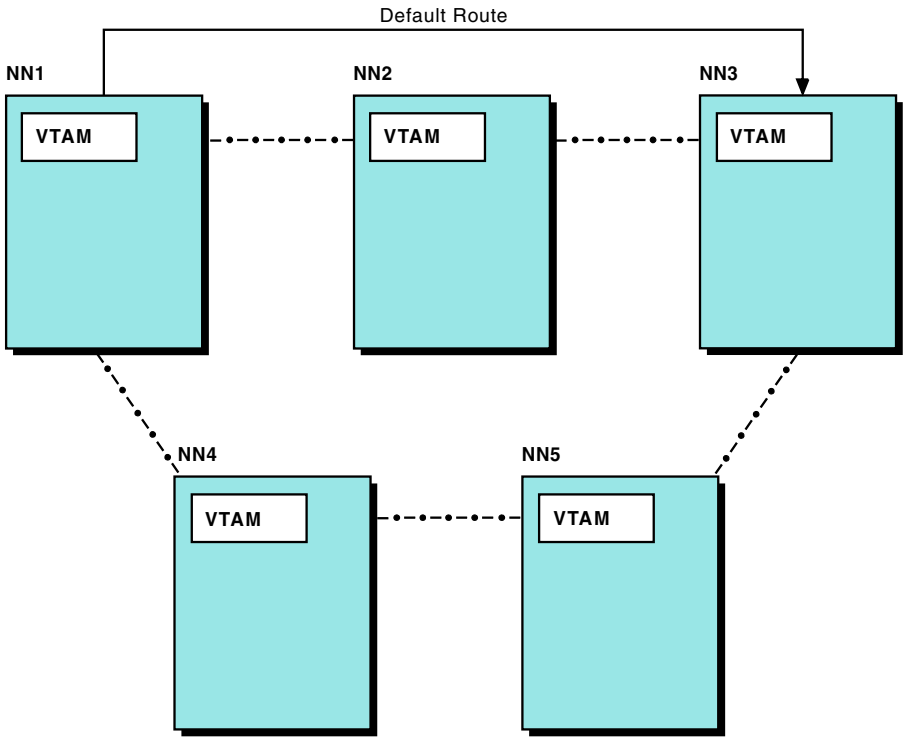


Figure 171. Sample network showing default route

If you want to force a different route for a certain type of session, you need to cause that route to have a lower weight than the original default route. For example, you might want the session between the LU in NN1 and the LU in NN3 in Figure 171 to be routed through NN4 and NN5. By defining a new APPN Class of Service and a transmission group (TG) profile, you can create a lower weight for the route you choose than the weight of the default route.

This example demonstrates using the UPARM1 operand to force a route to traverse NN1, NN4, NN5, and NN3. With the defaults, the values for UPARM1 for the links in the network would look like the example in Figure 172 on page 774.

© Copyright IBM Corp. 1984, 2011

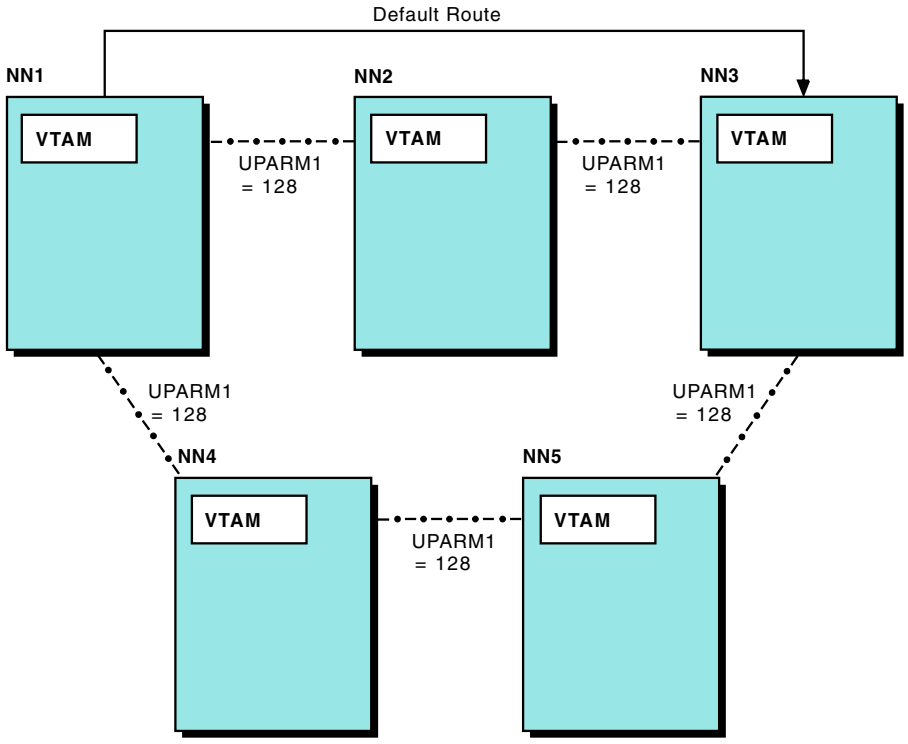


Figure 172. Sample network using default for UPARM1

To create a lower weight for the chosen route, you can define a TG profile with a UPARM1 value of 0 and specify that TG profile on the PUs associated with the links between NN1 and NN4, NN4 and NN5, and NN5 and NN3. The UPARM1 values in the network would then appear as they do in Figure 173 on page 775.

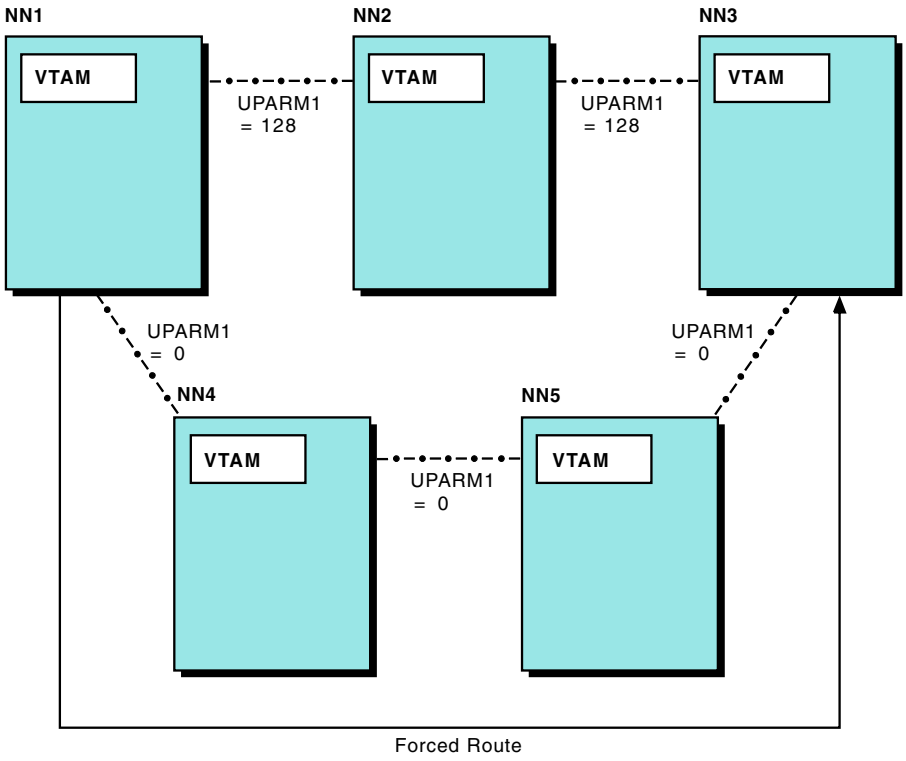


Figure 173. Sample network using TG profile on some links

By defining UPARM1 on the TG profile as 0, the links that specify that TG profile have a unique characteristic. You can define a new Class of Service that will give preference to links with a lower UPARM1 value. The following steps describe how to define the TG profile and define a Class of Service to take advantage of the UPARM1 value:

1. Code a TG profile with UPARM1=0 as shown in the following example.

```
************************************************************************
* TG Profile to change UPARM1 value                                    *
************************************************************************
*
UPARMLOW TGP   UPARM1=0
```

2. Specify this TG profile for the PUs associated with each line in the preferred route.
3. Code a Class of Service (COS) that gives preference to links using the TG profile.

   Note: Use the APPNCOS statement to name the new COS UPARMCOS.

   a. Copy the eighth LINEROW and NODEROW statements from the IBM-supplied COS definition #CONNECT (found in the COSAPPN member of SYS1.VTAMLST).
   b. Make a second copy of the LINEROW statement.
   c. On the LINEROW statements, modify only the NUMBER, UPARM1, and WEIGHT operands.

      Modify the first LINEROW statement to specify NUMBER=1, UPARM1=(0,99), and WEIGHT=30.

      Modify the second LINEROW statement to specify NUMBER=2, UPARM1=(100,255), and WEIGHT=240.

4. File this new APPN COS definition in SYS1.VTAMLST under the new member, COSSAMP. Use the VARY ACT command to activate this definition by specifying ID=COSSAMP. (As an alternative, if VTAM is not already active, you can specify COSSAMP in your CONFIG list so that the definition will be activated as part of VTAM initialization.)
5. Define a mode table entry that specifies the new APPN COS definition and add it to ISTINCLM or the mode tables associated with the LUs (if the default mode table is not being used). The following code is an example of the mode table entry.

```
         TITLE 'UPARMODE'
*********************************************************************
*                                                                   *
*        LOGMODE ENTRY FOR FORCED APPN ROUTE EXAMPLE -              *
*                                                                   *
*********************************************************************
UPARMODE MODEENT LOGMODE=UPARMODE,FMPROF=X'03',                      *
               TSPROF=X'03',PRIPROT=X'B1',SECPROT=X'A0',             *
               COMPROT=X'3040',                                      *
               APPNCOS=UPARMCOS
```

6. Specify the new mode name, UPARMODE, when requesting the session.
7. Because UPARMCOS is the APPN COS definition used by the route selection process, the NN1 to NN2 to NN3 route has a weight of 240 + 160 + 240 = 640 based on the values of the WEIGHT operand from the UPARMCOS definition. Because the NN1 to NN4 to NN5 to NN3 route has a weight of 30 + 160 + 30 + 160 + 30 = 410 based on the values of the WEIGHT operand, it will be the chosen least-weight route.

Note: For this example, it is required that the new mode and Class of Service tables be stored in the nodes that are the endpoints of the session path; however, we strongly recommend that the Class of Service and mode tables be consistent among all the nodes in the network.

The following example shows the new user-defined Class of Service.

```
*********************************************************************
* user-defined class-of-service definition                        *
*********************************************************************
UPARMCOS APPNCOS  PRIORITY=MEDIUM          transmission priority
         LINEROW  WEIGHT=30,               TG weight                 *
               NUMBER=1,                   line row number           *
               UPARM1=(0,99),              user defined parameter 1  *
               UPARM2=(0,255),             user defined parameter 2  *
               UPARM3=(0,255),             user defined parameter 3  *
               CAPACITY=(MINIMUM,MAXIMUM), line speed                *
               COSTTIME=(0,255),           cost per connect time     *
               COSTBYTE=(0,255),           cost per byte transmitted *
               PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
               SECURITY=(UNSECURE,MAXIMUM) security level for TG
         LINEROW  WEIGHT=240,              TG weight                 *
               NUMBER=2,                   line row number           *
               UPARM1=(100,255),           user defined parameter 1  *
               UPARM2=(0,255),             user defined parameter 2  *
               UPARM3=(0,255),             user defined parameter 3  *
               CAPACITY=(MINIMUM,MAXIMUM), line speed                *
               COSTTIME=(0,255),           cost per connect time     *
               COSTBYTE=(0,255),           cost per byte transmitted *
               PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
               SECURITY=(UNSECURE,MAXIMUM) security level for TG
         NODEROW  NUMBER=2,                node row number           *
               WEIGHT=160,                 node weight               *
               CONGEST=(LOW,HIGH),         congestion                *
               ROUTERES=(0,255)            route addition resistance
```

Appendix I. Border node connection types

This appendix lists connection types for various VTAM and partner node combinations. For further discussion of border nodes and APPN multiple network connectivity support, see "APPN multiple network connectivity" on page 89.

Table 70. Connection type for selected VTAM and partner node combinations

Local VTAM characteristics								
Partner node characteristics	**BN**		**NATIVE**			**NETID**		**Connection type**
	YES	**NO or Not Coded**	**YES**	**NO**	**Not Coded**	**Same**	**Different**	
Network node	X		X				X	Native connection, CP-CP sessions supported
Network node	X		X				X	Native connection, CP-CP sessions not supported
Network node	X			X		X		Peripheral connection, CP-CP sessions supported
Network node	X			X			X	Peripheral connection, CP-CP sessions supported
Network node	X				X	X		Native connection, CP-CP sessions supported
Network node	X				X		X	Peripheral connection, CP-CP sessions supported
Network node		X			X	X		Native connection, CP-CP sessions supported
Network node		X			X		X	Native connection, CP-CP sessions not supported
Peripheral border node	X		X			X		Native connection, CP-CP sessions supported
Peripheral border node	X		X				X	Connection failed by local VTAM
Peripheral border node	X			X		X		Peripheral connection, CP-CP sessions supported
Peripheral border node	X			X			X	Peripheral connection, CP-CP sessions supported
Peripheral border node	X				X	X		Native connection, CP-CP sessions supported

© Copyright IBM Corp. 1984, 2011

Table 70. Connection type for selected VTAM and partner node combinations (continued)

Local VTAM characteristics								
Partner node characteristics	**BN**		**NATIVE**			**NETID**		**Connection type**
	YES	**NO or Not Coded**	**YES**	**NO**	**Not Coded**	**Same**	**Different**	
Peripheral border node	X				X		X	Peripheral connection, CP-CP sessions supported
Peripheral border node		X			X	X		Native connection, CP-CP sessions supported
Peripheral border node		X			X		X	Peripheral connection (managed by partner node), CP-CP sessions supported
Extended border node, NATIVE=YES	X		X			X		Native connection, CP-CP sessions supported
Extended border node, NATIVE=YES	X		X				X	Native connection, CP-CP sessions not supported
Extended border node, NATIVE=YES	X			X		X		Connection failed by partner node
Extended border node, NATIVE=YES	X			X			X	Connection failed by partner node
Extended border node, NATIVE=YES	X				X	X		Native connection, CP-CP sessions supported
Extended border node, NATIVE=YES	X				X		X	Connection failed by partner node
Extended border node, NATIVE=YES		X			X	X		Native connection, CP-CP sessions supported
Extended border node, NATIVE=YES		X			X		X	Native connection, CP-CP sessions not supported
Extended border node, NATIVE=NO	X		X			X		Connection failed by local VTAM
Extended border node, NATIVE=NO	X		X				X	Connection failed by local VTAM
Extended border node, NATIVE=NO	X			X		X		Extended connection, CP-CP sessions supported

Table 70. Connection type for selected VTAM and partner node combinations (continued)

Local VTAM characteristics								
Partner node characteristics	**BN**		**NATIVE**			**NETID**		**Connection type**
	YES	**NO or Not Coded**	**YES**	**NO**	**Not Coded**	**Same**	**Different**	
Extended border node, NATIVE=NO	X			X			X	Extended connection, CP-CP sessions supported
Extended border node, NATIVE=NO	X				X	X		Extended connection, CP-CP sessions supported
Extended border node, NATIVE=NO	X				X		X	Extended connection, CP-CP sessions supported
Extended border node, NATIVE=NO		X			X	X		Peripheral connection (managed by partner node), CP-CP sessions supported
Extended border node, NATIVE=NO		X			X		X	Peripheral connection (managed by partner node), CP-CP sessions supported
Extended border node, NATIVE not coded	X		X			X		Native connection, CP-CP sessions supported
Extended border node, NATIVE not coded	X		X				X	Connection failed by partner node
Extended border node, NATIVE not coded	X			X		X		Extended connection, CP-CP sessions supported
Extended border node, NATIVE not coded	X			X			X	Extended connection, CP-CP sessions supported
Extended border node, NATIVE not coded	X				X	X		Native connection, CP-CP sessions supported
Extended border node, NATIVE not coded	X				X		X	Extended connection, CP-CP sessions supported
Extended border node, NATIVE not coded		X			X	X		Native connection, CP-CP sessions supported
Extended border node, NATIVE not coded		X			X		X	Peripheral connection (managed by partner node), CP-CP sessions supported

Table 70. Connection type for selected VTAM and partner node combinations (continued)

Local VTAM characteristics								
Partner node characteristics	**BN**		**NATIVE**			**NETID**		**Connection type**
	YES	**NO or Not Coded**	**YES**	**NO**	**Not Coded**	**Same**	**Different**	

Notes:

1. The local VTAM node is always configured with the NODETYPE=NN specified.
2. XNETALS=YES is assumed whenever the NETIDs of the two nodes are different. If XNETALS=NO is specified, connections between two nodes with different NETIDs fail.
3. When BN=YES is coded, XNETALS=YES is the default start option. When BN=YES is not coded, XNETALS=NO is the default start option.

Appendix J. VTAM restricted materials

The VTAM modules and macroinstructions listed in this appendix are considered to be Restricted Materials of IBM and are available through the View Program Listings (VPL) application. This information can be accessed by using ServiceLink or Dial IBM. All other VTAM modules are designated as object code only (OCO).

Table 71. User replaceable or modifiable modules

Module	Description
ISTMGC01	CNM routing table (IBM-provided default table)
ISTINCLM	Logon mode table
ISTINCDT	Session-level USS table

Table 72. VTAM message modules

Module	Description
ISTCFCMM	English text of VTAM messages
ISTINCNO	Formatted English text of VTAM messages
ISTCFCML	Defines variables and variable lengths for VTAM messages

© Copyright IBM Corp. 1984, 2011

Appendix K. Architectural specifications

This appendix lists documents that provide architectural specifications for the SNA Protocol.

The APPN Implementers' Workshop (AIW) architecture documentation includes the following architectural specifications for SNA APPN and HPR:

- APPN Architecture Reference (SG30-3422-04)
- APPN Branch Extender Architecture Reference Version 1.1
- APPN Dependent LU Requester Architecture Reference Version 1.5
- APPN Extended Border Node Architecture Reference Version 1.0
- APPN High Performance Routing Architecture Reference Version 4.0
- SNA Formats (GA27-3136-20)
- SNA Technical Overview (GC30-3073-04)

For more information, refer to the AIW documentation page at http://www.ibm.com/support/docview.wss?rs=852&uid=swg27017843.

The following RFC also contains SNA architectural specifications:

- RFC 2353 *APPN/HPR in IP Networks APPN Implementers' Workshop Closed Pages Document*

RFCs can be obtained from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

Many RFCs are available online. Hardcopies of all RFCs are available from the NIC, either individually or by subscription. Online copies are available using FTP from the NIC at http://www.rfc-editor.org/rfc.html.

Use FTP to download the files, using the following format:

```
RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS
```

where:

- *nnnn* is the RFC number.
- TXT is the text format.
- PS is the postscript format.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to `service@nic.ddn.mil` with a subject line of `RFC nnnn` for text versions or a subject line of `RFC nnnn.PS` for PostScript versions. To request a copy of the RFC index, send a message with a subject line of `RFC INDEX`.

For more information, contact `nic@nic.ddn.mil`.

© Copyright IBM Corp. 1984, 2011

Appendix L. Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, you may view the information through the z/OS Internet Library website or the z/OS Information Center. If you continue to experience problems, send an email to mhvrcfs@us.ibm.com or write to:

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer or Library Server versions of z/OS books in the Internet library at www.ibm.com/systems/z/os/zos/bkserv/.

One exception is command syntax that is published in railroad track format, which is accessible using screen readers with the Information Center, as described in "Dotted decimal syntax diagrams."

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the Information Center using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always

© Copyright IBM Corp. 1984, 2011

present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should refer to separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- A question mark (?) means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- An exclamation mark (!) means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted

decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

- An asterisk (*) means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

  Notes:

  1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
  2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
  3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.

- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Notices

This information was developed for products and services offered in the USA.

IBM may not offer all of the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14 Shimotsuruma,, Yamato-Shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

© Copyright IBM Corp. 1984, 2011

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or

imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by The University of California. Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Portions of this publication relating to RPC are Copyright © Sun Microsystems, Inc., 1988, 1989.

Some portions of this publication relating to X Window System** are Copyright © 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts, and the Massachusetts Institute Of Technology, Cambridge, Massachusetts. All Rights Reserved.

Some portions of this publication relating to X Window System are Copyright © 1986, 1987, 1988 by Hewlett-Packard Corporation.

Permission to use, copy, modify, and distribute the M.I.T., Digital Equipment Corporation, and Hewlett-Packard Corporation portions of this software and its documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T., Digital, and Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T., Digital, and Hewlett-Packard make no representation about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright © 1983, 1995-1997 Eric P. Allman

Copyright © 1988, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

```
This product includes software developed by the University of
California, Berkeley and its contributors.
```

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software program contains code, and/or derivatives or modifications of code originating from the software program "Popper." Popper is Copyright ©1989-1991 The Regents of the University of California, All Rights Reserved. Popper was created by Austin Shelton, Information Systems and Technology, University of California, Berkeley.

Permission from the Regents of the University of California to use, copy, modify, and distribute the "Popper" software contained herein for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies. HOWEVER, ADDITIONAL PERMISSIONS MAY BE NECESSARY FROM OTHER PERSONS OR ENTITIES, TO USE DERIVATIVES OR MODIFICATIONS OF POPPER.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THE POPPER SOFTWARE, OR ITS DERIVATIVES OR MODIFICATIONS, AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE POPPER SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Copyright © 1983 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to

endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1991, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   ```
   This product includes software developed by the University of
   California, Berkeley and its contributors.
   ```
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 1990 by the Massachusetts Institute of Technology

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original M.I.T. software. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright © 1998 by the FundsXpress, INC. All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)". The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related.
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include acknowledgement:

   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

This product includes cryptographic software written by Eric Young.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 2004 IBM Corporation and its licensors, including Sendmail, Inc., and the Regents of the University of California. All rights reserved.

Copyright © 1999,2000,2001 Compaq Computer Corporation

Copyright © 1999,2000,2001 Hewlett-Packard Company

Copyright © 1999,2000,2001 IBM Corporation

Copyright © 1999,2000,2001 Hummingbird Communications Ltd.

Copyright © 1999,2000,2001 Silicon Graphics, Inc.

Copyright © 1999,2000,2001 Sun Microsystems, Inc.

Copyright © 1999,2000,2001 The Open Group

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

X Window System is a trademark of The Open Group.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

PostScript is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Bibliography

This bibliography contains descriptions of the documents in the z/OS Communications Server library.

z/OS Communications Server documentation is available in the following forms:

- Online at the z/OS Internet Library web page at www.ibm.com/systems/z/os/zos/bkserv/
- In softcopy on CD-ROM collections. See "Softcopy information" on page xxvi.

z/OS Communications Server library updates

An index to z/OS Communications Server book updates is at http://www.ibm.com/support/docview.wss?uid=swg21178966. Updates to documents are also available on RETAIN® and in information APARs (info APARs). Go to http://www.ibm.com/software/network/commserver/zos/support to view information APARs. In addition, Info APARs for z/OS documents are in *z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation*, which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/ BOOKS/ZIDOCMST/CCONTENTS.

z/OS Communications Server information

z/OS Communications Server product information is grouped by task in the following tables.

Planning

Title	Number	Description
z/OS Communications Server: New Function Summary	GC31-8771	This document is intended to help you plan for new IP for SNA function, whether you are migrating from a previous version or installing z/OS for the first time. It summarizes what is new in the release and identifies the suggested and required modifications needed to use the enhanced functions.
z/OS Communications Server: IPv6 Network and Application Design Guide	SC31-8885	This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server's support of IPv6, coexistence with IPv4, and migration issues.

Resource definition, configuration, and tuning

Title	Number	Description
z/OS Communications Server: IP Configuration Guide	SC31-8775	This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Reference*.

© Copyright IBM Corp. 1984, 2011

Title	Number	Description
z/OS Communications Server: IP Configuration Reference	SC31-8776	This document presents information for people who want to administer and maintain IP. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Guide*. The information in this document includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC31-8777	This document presents the major concepts involved in implementing an SNA network. Use this document in conjunction with the *z/OS Communications Server: SNA Resource Definition Reference*.
z/OS Communications Server: SNA Resource Definition Reference	SC31-8778	This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document in conjunction with the *z/OS Communications Server: SNA Network Implementation Guide*.
z/OS Communications Server: SNA Resource Definition Samples	SC31-8836	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: IP Network Print Facility	SC31-8833	This document is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Operation

Title	Number	Description
z/OS Communications Server: IP User's Guide and Commands	SC31-8780	This document describes how to use TCP/IP applications. It contains requests that allow a user to log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC31-8781	This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.
z/OS Communications Server: SNA Operation	SC31-8779	This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.
z/OS Communications Server: Quick Reference	SX75-0124	This document contains essential information about SNA and IP commands.

Customization

Title	Number	Description
z/OS Communications Server: SNA Customization	SC31-6854	This document enables you to customize SNA, and includes the following: • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines

Writing application programs

Title	Number	Description
z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference	SC31-8788	This document describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this document to adapt your existing applications to communicate with each other using sockets over TCP/IP.
z/OS Communications Server: IP CICS Sockets Guide	SC31-8807	This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using z/OS TCP/IP.
z/OS Communications Server: IP IMS Sockets Guide	SC31-8830	This document is for programmers who want application programs that use the IMS TCP/IP application development services provided by the TCP/IP Services of IBM.
z/OS Communications Server: IP Programmer's Guide and Reference	SC31-8787	This document describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
z/OS Communications Server: SNA Programming	SC31-8829	This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
z/OS Communications Server: SNA Programmer's LU 6.2 Guide	SC31-8811	This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.)
z/OS Communications Server: SNA Programmer's LU 6.2 Reference	SC31-8810	This document provides reference material for the SNA LU 6.2 programming interface for host application programs.
z/OS Communications Server: CSM Guide	SC31-8808	This document describes how applications use the communications storage manager.

Title	Number	Description
z/OS Communications Server: CMIP Services and Topology Agent Guide	SC31-8828	This document describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The document provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis

Title	Number	Description
z/OS Communications Server: IP Diagnosis Guide	GC31-8782	This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
z/OS Communications Server: ACF/TAP Trace Analysis Handbook	GC23-8588-00	This document explains how to gather the trace data that is collected and stored in the host processor. It also explains how to use the Advanced Communications Function/Trace Analysis Program (ACF/TAP) service aid to produce reports for analyzing the trace data information.
z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures and *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT*	GC31-6850 GC31-6851	These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
z/OS Communications Server: SNA Data Areas Volume 1 and *z/OS Communications Server: SNA Data Areas Volume 2*	GC31-6852 GC31-6853	These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and codes

Title	Number	Description
z/OS Communications Server: SNA Messages	SC31-8790	This document describes the ELM, IKT, IST, IUT, IVT, and USS messages. Other information in this document includes: • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
z/OS Communications Server: IP Messages Volume 1 (EZA)	SC31-8783	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)	SC31-8784	This volume contains TCP/IP messages beginning with EZB or EZD.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC31-8785	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)	SC31-8786	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC31-8791	This document describes codes and other information that appear in z/OS Communications Server messages.

Index

Numerics

1, type physical unit 235
2 254, 600
2, type physical unit 235
2.1, nonnative network connection 254, 600
31-bit backed data space 741
3172 Nways Interconnect Controller
  APPN connections 60
  connection network, defining a connection to 67
  link-attached NCP support 127
  multiple-domain connections
    Ethernet or Ethernet-type LAN 112
    Fiber Distributed Data Interface network (FDDI) 112
    token-ring network 112
  NetView and 258
  single-domain connections
    Ethernet or Ethernet-type LAN 258
    Fiber Distributed Data Interface network (FDDI) 258
    token-ring network 258
  tuning statistics 668
  VCNS and 411
3272 cluster controllers 255
3277 terminals 255
6.2 sessions
  description 419
  non-LU 6.2 sessions and 419
64-bit backed data space 741

A

ACB (access method control block), opening by application program 373
ACB macroinstruction
  APPLID operand 373
  PERSIST operand 409
ACBNAME operand on APPL definition statement 373
accessibility 785
accounting for cross-domain sessions 307
activating an NCP
  activating adjacent link stations 644
  overview 643
  reactivating a failed NCP 647
  switched link, using 649
activating resources
  ADJSSCP tables 612
  application programs 614
  automatic activation of cross-subarea links 644
  cross-domain resource managers 612
  multiple-domain environment 612
  multiple-network environment 612
  NCP subarea node 644
  order of activation 611
  overview 610
adding PUs and LUs dynamically 235
address space identifier (ASID) 743
address space, spanning multiple 412
address subarea node
  address incompatibility 362
  addressing methods 361
  extended addressing compatibility 364
address subarea node *(continued)*
  extended network addressing 361
  path definition, subarea 359
  subarea addressing restrictions 364
address translation 590
address, element 360
addressing
  compatibility
    extended network and extended subarea 364
    nonextended and extended network 361
  LU, dependent 283
  multiple-network (SNI) 584
  overview 360
  restrictions 362
  structure types 361
  subarea restrictions 364
  translation of address 590
addressing, enhanced for session managers 34
ADDSESS operand on the BUILD definition statement 282
adjacent link station (ALS) 273
adjacent SSCP
  lists for CDRSCs 542
  multiple-domain environment 541
  overview 541
  performance 546
  request routing 583
  selecting with the session management exit 306
  tables
    alias name translation and 584
    combined APPN and subarea network 519
    default lists 579
    defining 543, 579
    dynamic CDRSCs and 584
    dynamically defining 543
    multiple-network environment 579
    multiple-network sample 580
    network-specific lists 579
    requirements for 579
    types of 579
adjacent SSCP lists for CDRSCs 542
adjacent SSCP table, dynamic
  CDRSC definition statement 537
  controlled by another VTAM domain 536
  cross-domain resource major node 536
  dynamic adjacent SSCP table 543
  locating cross-domain resources 541
  network resources for XRF 537
  USERVAR with XRF 537
ADJCP definition statement 490
agent, VTAM topology enabling 3
agent, VTAM topology, overview 427
alerts, VTAM to NPDA, NMVTLOG start option 36
alias name translation
  adjacent SSCP tables and 584
  alias selection function (VTAM) 593
  defining 595
  LU 6.2 599
  NetView 603
  overview 593
allocating private storage 412
ALS (adjacent link station) 273

© Copyright IBM Corp. 1984, 2011

ALSLIST operand
APPN connections 488
combined networks 519
independent LUs 278
ALSREQ start option 278, 281
ANR (automatic network routing) 493
ANS operand on PU definition statement 263
APBUF buffer pool
number of buffers per page 697
overview 691
APF (authorized program facility) 381
API, cross-memory 412
APING support 47, 421, 617
APPC operand on APPL definition statement 412
APPCCMD macroinstruction
CONFTXT operand 398
requirement for LU 6.2 412
APPL definition statement
ACBNAME operand 373
APPC operand 412
ATNLOSS operand 419
AUTH operand 380
AUTODMP operand 123
AUTOIPL operand 123
AUTOSES operand 416
DDRAINL operand 416
description 373
DLOGMODE operand 399
DMINWNL operand 416
DMINWNR operand 416
DRESPL operand 416
DSESLIM operand 416
EAS operand 682
ENCR operand 388
HAVAIL operand 407
LIMQSINT operand 420
LIMRES operand 420
MODETAB operand 399
PARSESS operand 381, 591
sample 373
SYNCLVL operand 419
VCNS operand 411
APPL EAS storage estimates 738
application password 750
application program
ACBNAME operand 373
acquiring sessions 381
activating 614
APPLID operand 373
ATTN exit 419
authorized path 381
communication network management (CNM) 624
confidential data 398
controlling 300
cross-memory API 412
data transfer, privileged 381
defining 373
dial number digits, overriding 380
draining conversations 416
encryption 386
extended recovery facility (XRF) 407
failed sessions, reconnecting to 409
generic resources 406
impact of outages, reducing 407
logon mode 399
logon mode parameters 399
logon mode table entries, create 399
application program *(continued)*
logon request validation 380
LU 6.2 412
LU 6.2 security 417
LU 6.2 session 412
LU 6.2 sync point service 419
model definition 374
naming 373
network management capability 623
overview 25
parallel sessions 381, 591
passing logon request 380
private storage, allocating 412
program operator (PPO and SPO) 623
resource verification reduction 414
sample definition 373
security facilities 386
security protocol, conversation 417
spanning multiple address space 412
start-stop devices, communicating with 412
starting
name of application and 374
overview 397
TSO 715
USERVAR 399
validate logon 380
XRF (extended recovery facility) 407
application-supplied operands for dial or token ring 380
APPLID operand on ACB macroinstruction 373
APPN
addressing overview 21
ALSLIST operand on CDRSC definition statement 488
bisynchronous (BSC) 3270 sessions 510
border node
connection types 777
description 9, 89
BSC 3270 sessions 510
central directory server
benefits of 329
CDSREFER start option 329
overview 8
resource registration 330
search requests 330
using to improve efficiency 329
Class of Service (COS), overview 19, 334
combined with subarea network 505
composite network node, overview 11
connection network 62
connections
3172 Nways interconnect controller 60
assigning NCP addresses 489
converting LEN connections to APPN 505
definitions of 487
leased lines 59
overview 49
requirements 488
sample type 2.1 channel 59
session control block usage 488
setting maximum number of sessions 488
type 2.1 channel 51, 506
control point (CP)
CP name 29, 490
overview 6
converting LEN connections 505, 507
CP name for VTAM 29
CPNAME operand on PU definition statement 488
dependent LUs 510

APPN *(continued)*
directed search 326
directory services, controlling the size of 331
end node, overview 7
extended subnetwork boundaries 91
generic resources 406
host-to-host channel connection 52
IDBLK operand on PU definition statement 488
IDNUM operand on PU definition statement 488
implementing a network 487
interchange node
coding 506
combined APPN and subarea network 506
overview 10
LEN connections, converting 505, 507
locating nodes 325, 509
migrating from subarea 505
migration data host
in APPN 507
overview 10
multiple network connectivity 89
network accessible units 12
network node server
defining list 100
overview 8
replacing list 101
resource registration 330
sample list 101
network node, overview 6
nodes
adding and moving 527
adjacent 490
connecting to VTAM 49
types of 5
peripheral subnetwork boundary 91
resource registration 330
resource verification reduction 414
route calculation, overview 20
route selection, overview 20
routing
directory services 325
improving efficiency 329
route selection 20
topology database 18
types of searches 326
searches
broadcast search 327
directed search 326
improving efficiency 329
resource verification reduction 414
sessions in a combined network 519, 600
start options
APPNCOS 93
BN 92
BNDYN 92
BNORD 93
CDSERVR 330
CDSREFER 329
CONNTYPE 505
CPCP 489
DIRSIZE 331
DIRTIME 331
DYNADJCP 490
HPR 495
HPRPST 494
INITDB 333
NODETYPE 506, 507
APPN *(continued)*
start options *(continued)*
SNVC 93
VERIFYCP 418
VFYRED 415
VFYREDTI 415
VRTG 97
VRTGCPCP 97
XID operand on PU definition statement 488
APPNCOS start option 93
APPNTOSA COS mapping table 527
ASID (address space identifier) 743
ASIRFMSG start option 35
associated LU table 291
assuming resource ownership 631
Asynchronous transfer mode connections
ATM LAN emulation connections 68
ATM native connections 69
ATCCON (configuration list) 37
ATCSTR00 (start option list) 27, 30, 253
ATM connections
ATM LAN emulation connections 68
ATM native connections 69
ATM native connections
connection to the Open Systems Adapter 69
port on the Open Systems Adapter 71
TGs over SVCs 75
ATNLOSS operand on APPL definition statement 419
ATTN exit 419
AUTH operand on APPL definition statement 380
AUTHLEN start option and operand 282
authorized path 381
authorized program facility (APF) 381
authorized transmission priority for LEN connections 282
authorizing cross-domain sessions 306
AUTODMP operand on APPL definition statement 123
AUTOIPL operand on APPL definition statement 123
automatic logon
changing an LU's automatic logon specification 301
coding 300
controlling 306
defined 300
fails, warning 404
LOGAPPL operand 300
multiple-network 601
operator-controlled 301
automatic network routing (ANR) 493
AUTORTRY start option 302
AUTOSES operand on APPL definition statement 416
AUTOSYN operand on PCCU definition statement 124
AUTOTI start option 302
AUXADDR operand on the BUILD definition statement 281

B

backup
NCP owner 631
sessions, extended recovery facility and 407
backup start option list (LISTBKUP) 30, 31
binary synchronous (BSC) connection
and APPN 510
NCP-attached 270
BIND, extended 273
BN start option 92
BNDYN start option 92
BNORD start option 93

border node
connection types 777
description 9, 89
boundary function-based transmission groups 50
broadcast search (APPN)
central directory server and 327
limiting 329, 521
originating 327
overview 327
when used 327
BSBUF buffer pool
number of buffers per page 697
overview 691
BSC (binary synchronous) connection
and APPN 510
NCP-attached 270
BSCMDRS start option 35
buffer pools
allocating storage 693
dynamic expansion
basic allocation and 693
guidelines for 695
operation of 694
virtual storage and 698
monitoring 665
number of buffers per page 698
overview 664
types of 691
buffer pools, CSM
size of 741
source of 741
BUILD definition statement
description 127
LOADLIB operand 125
NEWNAME operand 125
sample, channel-attached 117, 124

C

CACHETI start option 93
call security verification 110
canceling VTAM 623
CCL (Communication Controller for Linux on zSeries) 658
CDRM (cross-domain resource manager)
activating 612
coding of 532
deactivating 620
description of 532
DISJOINT operand 521
dynamic definition of cross-domain resources 534
external CDRM 532
gateway VTAM, for 567
host CDRM 532
multiple domains 532
sample of 533
CDRSC (cross-domain resource)
adjacent SSCP (ADJSSCP) table, defining 579
adjacent SSCP lists for CDRSCs 542
CDRM ownership
changing by operator 549
changing dynamically 548
verifying 548
combined APPN and subarea network 515
defining 515, 534
definition statement, VFYOWNER operand 548
description 579
dynamic 576
CDRSC (cross-domain resource) *(continued)*
locating
adjacent SSCP table 541
ADJCDRM definition statement 543
CDRSC routed to each SSCP 541
cross-network 579
multiple-network configuration 580
multiple-network environment 570
performance improving 546
predefined 574
SSCPDYN start option 546
SSCPORD start option 546
CDRSCTI start option 509, 521
CDSERVR start option (APPN) 330
CDSREFER start option (APPN) 329
central directory server (APPN)
benefits of 329
broadcast search and 327
defining 330
overview 8
resource registration 330
search requests 330
selection of 329
using to improve efficiency 329
change-number-of-sessions
contention loser and 416
modifying parameters 415
operating commands for 417
overview 415
channel
connections, VTAM-to-peripheral node 255
connections, VTAM-to-VTAM 103
device name
non-SNA devices, for 256
SNA devices, for 256
enterprise systems connection (ESCON) 258
multipath channel (MPC) for hosts 51
sample APPN 59
type 2.1 APPN 51
channel unit address, NCP, channel-attached 116
channel-attached
APPN type 2.1 51
between host 103
I/O performance with nodes 668
loop-adapter attachment 258
MVS/CP devices 230
NCP 116
non-SNA devices 255
SNA cluster controllers, defining statically 256
type 2.1 254
channel-link
NCP 117
parallel transmission group and 118
channel-to-channel adapter connections
defining 103
tuning I/O for 670
checkpointing TRS and directory databases 333
CICS, extended recovery facility (XRF) 407
Class of Service (COS)
APPN, overview 19
combined APPN and subarea network 523
database (APPN) 19
defaults, IBM-specified 354
logon mode table 399
mapping table 527
multiple-network 588
name 595

Class of Service (COS) *(continued)*
substituting parameters 354
table 353
unnamed entry 353
cluster controllers
non-SNA, defining 255
SNA
defining statically 256
I/O performance to 668
sample tuning statistics report 668
CMC (communication management configuration)
overview 10, 221
problem determination 225
CMIP services associations, limited resources 420
CMIP services, overview 427
CMPMIPS start option 382
CNM (communication network management)
application program 624
application program, defined 624
filtering session awareness data 625
NetView program and 603
procedure-related identifier (PRID) 624
routing table 624
routing unsolicited requests 624
CNOS
contention loser and 416
modifying parameters 415
operating commands for 417
overview 415
coattailing
channel-to-channel adapter
DELAY operand 711
I/O buffer size 710
virtual route (VR) window size 712
DELAY operand 704
determining amount of 682
inbound 704
MAXBFRU operand 704
maximizing 702
outbound 703
overview 664
UNITSZ value 705
coded data
CDRM key 757
cross-domain cryptography 757
filing secondary logic unit (SLU) keys 755
single-domain encryption 755
coding concepts 38
coding resource definition statements
overview 38
sift-down 39
combined APPN and subarea network 505
common service area (CSA)
buffer pool waste 666
common service area 24-bit (CSA24) 683
CSA limit 683
monitoring 665
communication controller
adapter type for NTRI 269
channel-enable switch 650
dumping of 657
enabling channels for activation 650
I/O performance with 668
links between
activating 653
deactivating 647
loading or purging the disk 643
communication controller *(continued)*
remote program load (RPL) facility 650
sample tuning statistics report 668
tuning I/O for 668
Communication Controller for Linux on zSeries 658
activating OSA-attached CCL NCPs 659
defining OSA-attached CCL NCPs 659
dumping OSA-attached CCL NCPs 661
loading OSA-attached CCL NCPs 659
managing OSA-attached CCL NCPs 661
monitoring OSA-attached CCL NCPs 661
communication management configuration (CMC)
overview 10, 221
problem determination 225
communication network management (CNM)
application program 624
application program, defined 624
filtering session awareness data 625
NetView program and 603
procedure-related identifier (PRID) 624
routing table 624
routing unsolicited requests 624
Communications Server for z/OS, online information xxviii
communications storage manager
definition of 741
starting 742
system definition 742
composite network node
overview 11
routing failure message 344
compression of data 381
confidential data 398
CONFIG operand on START command 37
configuration list
advantages of 27
CONFIG operand 37
for gateway VTAM 556
overview 25
sample 37
configuration of I/O devices, dynamic
command list programs 763
customization 234, 407
default naming convention 233
installation and preparation 232
overview 230
configuration restart
defined 606
NODELST data set 606
WARM start 606
configuration restart file, VSAM
associating with major nodes 607
CONFGDS operand 607
CONFGPW operand 607
MVS 607
CONFTXT operand on APPCCMD macroinstruction 398
congestion, route 367
connecting networks 556
connection networks (APPN)
overview 62
CONNTYPE
operand on PU definition statement, overview 487
parallel transmission groups and 50
start option 487, 505
consoles, multiple operator 618
contention loser, CNOS and 416

control point (CP)
adjacent
ADJCP definition statement 490
DYNADJCP start option 490
predefinition of 490
dynamic creation of adjacent CPs 490
overview 6
control vector 29 465
conversations, LU 6.2
draining 416
overview 413
converting LEN connections to APPN 505
COS (Class of Service)
APPN, overview 19
combined APPN and subarea network 523
database (APPN) 19
defaults, IBM-specified 354
logon mode table 399
mapping table 527
multiple-network 588
name 595
substituting parameters 354
table 353
unnamed entry 353
coupling facility failures
Sysplex Wide Security Associations 484
TCP/IP stack 484
VTAM node 484
Sysplexports 485
Coupling facility failures
Sysplexports
TCP/IP stack 485
VTAM node 485
coupling facility structure
allocating storage for 443
attributes 438
connecting to 443
duplexing 444
dynamic altering of 445
failures for generic resource configuration 457
failures for multinode persistent session configuration 474
for multinode persistent sessions 470
rebuild 443
setting up the sysplex environment 436
size, determining 441
storage shortages 444
CP (control point)
adjacent
ADJCP definition statement 490
DYNADJCP start option 490
predefinition of 490
dynamic creation of adjacent CPs 490
overview 6
CP name for VTAM 29
CP-CP sessions (APPN)
establishing between VTAM nodes 490
overview 14
CPCP, start option 489
CPNAME and PU name uniqueness 505
CPNAME operand on PU definition statement
type 2.1 PU and 253
VTAM switched SDLC connection 264
CPSVRMGR pipe 511
CPSVRMGR session 510
cross-domain resource (CDRSC)
adjacent SSCP (ADJSSCP) table, defining 579
adjacent SSCP lists for CDRSCs 542
cross-domain resource (CDRSC) *(continued)*
CDRM ownership
changing by operator 549
changing dynamically 548
verifying 548
combined APPN and subarea network 515
defining 515, 534
definition statement, VFYOWNER operand 548
description 579
dynamic 576
locating
adjacent SSCP table 541
ADJCDRM definition statement 543
CDRSC routed to each SSCP 541
cross-network 579
multiple-network configuration 580
multiple-network environment 570
performance improving 546
predefined 574
SSCPDYN start option 546
SSCPORD start option 546
cross-domain resource control
CDRSC definition statement 537
controlled by another VTAM domain 536
cross-domain resource major node 536
dynamic adjacent SSCP table 543
locating cross-domain resources 541
network resources for XRF 537
USERVAR with XRF 537
cross-domain resource manager (CDRM)
activating 612
coding of 532
deactivating 620
description of 532
DISJOINT operand 521
dynamic definition of cross-domain resources 534
external CDRM 532
gateway VTAM, for 567
host CDRM 532
multiple domains 532
sample of 533
cross-memory API 412
cross-network routing
communication between network 585
COS tables 590
extended subarea address gateway 585
extended subarea addressing 585
gateway NCP 585
gateway VTAM 585
virtual route, default 590
CRPLBUF buffer pool
guidelines for 696
overview 691
cryptographic capability, changing 762
cryptographic key
CDRM key 757
cross-domain cryptography 757
filing secondary logic unit (SLU) keys 755
single-domain encryption 755
cryptography
application program 386
APPN and cryptography 392
end-to-end cryptography 392
extended recovery facility (XRF) 407
host-by-host cryptography 393
sample
multiple-domain 389

cryptography *(continued)*
sample *(continued)*
single-domain 388
CSA (common service area)
buffer pool waste 666
common service area 24-bit (CSA24) 683
CSA limit 683
monitoring 665
CSA24 start option 683
CSALIMIT start option 683
CSM
buffer pools 741
definition of 741
starting 742
storage types 741
system definition 742
CSM messages 742
CSM parmlib member 742
CSM tracing 743
CTCA
defining 103
tuning I/O for 670
CUADDR operand, NCP, channel-attached 116
CWALL threshold condition (NCP) 667

D

data compression 381
data control, subarea to network 365
data hosts, overview 10
data transmission shutdown 367
DDRAINL operand on APPL definition statement 416
deactivating resources
automatic deactivation of cross-subarea links 620
cross-domain resource managers 620
forced deactivation 621
forced reactivation 621
immediate deactivation 621
multiple-domain environment, in a 620
NCP subarea node 653
order of deactivation 620
overview 619
DELAY operand on LINE definition statement
CTC coattailing and 711
inbound coattailing and 704
NCP coattailing and 705
delayed activation of logical links
multiple-domain, in a 628
single-domain, in a 648
delayed disconnection 266
deleting PUs and LUs dynamically 235
dependent LU requester 510
dependent LU server 510
dependent LUs in an APPN network 510
diagnosis
alerts to NPDA 36
module name, viewing 36
NMVTLOG start option 36
SDLC statistical MDRs 36
start options 35
time interval notification 35
trace facility 37
dial connections
dial-in, VTAM switched SDLC 264
dial-out, VTAM switched SDLC 265
dial number digits, overriding 380
DIALNO operand on PATH definition statement for NTRI 270
directed search (APPN)
originating 326
overview 326
resource verification reduction 414
when used 326
directory services (APPN)
checkpointing 333
controlling the size of the database 331
database 325
improving performance 333
learning about resources 325
overview 325
DIRSIZE start option (APPN) 331
DIRTIME start option (APPN) 331
disability 785
DISCNT (disconnect facility) 266
DISCNTIM start option 267
disconnect facility (DISCNT) 266
disconnection, delayed (DISCNT=DELAY) 267
discontiguous domains 630
DISJOINT operand on CDRM definition statement 521
DISPLAY APING command 47, 421, 617
DISPLAY BFRUSE operator command 665
DISPLAY CSM command 742
DISPLAY STORUSE pools 684
DLCADDR 270
DLCADDR operand on PATH definition statement
dependent LU requester, defining 512
NTRI, for 270
DLOGMODE operand on APPL definition statement 399
DLUR 510
DLURNAME operand on PATH definition statement 512
DLUS 510
DMINWNL operand on APPL definition statement 416
DMINWNR operand on APPL definition statement 416
DNS, online information xxix
DRESPL operand on APPL definition statement 416
DSESLIM operand on APPL definition statement 416
DSIRFMSG start option 35
dumps, NCP
dynamic 654
performance, improving 33
selecting a dump file 656
selecting the disposition of a dump 656
static (destructive) 654
DYNADJCP start option 490
dynamic definition of CDRSCs 534
dynamic definition of independent LUs 276
dynamic definition of PUs 227
dynamic definition of VTAM-to-VTAM connections 445
dynamic I/O configuration
command list programs 763
customization 234, 407
default naming convention 233
installation and preparation 232
overview 230
dynamic network identification 563
dynamic path update 369
dynamic reconfiguration
effect on WARM start 608
frame relay lines, PUs on 244
implicit dynamic reconfiguration 246
overview 235
samples 246
VARY ACT, UPDATE technique 238

dynamic reconfiguration *(continued)*
VARY DRDS technique 241
dynamic resource definition (model name) 290
dynamic selection of session connections 281
dynamic switched definitions 227
dynamic switched major node 228
dynamic table replacement 619
DYNDLGMD start option 277, 293
DYNLU start option 490
DYNMODTB start option 277, 293, 524
DYNPU operand 227

E

EAS (estimated number of active sessions)
APPL definitions 682
improving session establishment performance 282
LFBUF and 696
LU definitions 683
TSO, specifying for 716
EAS storage estimates 738
element address 360
ENA (extended network addressing)
addressing compatibility 361
addressing restrictions 361
compatibility, nonextended address 361
element address 360
ENCR operand
APPL definition statement 388
LU definition statement 388
encryption facility
cryptographic keys, defining 755
description 386
extended recovery facility (XRF) 407
sample
multiple-domain 389
single-domain 388
encryption of data
application program 386
APPN and cryptography 392
CDRM key 757
cross-domain cryptography 757
end-to-end cryptography 392
extended recovery facility (XRF) 407
filing secondary logic unit (SLU) keys 755
host-by-host cryptography 393
sample
multiple-domain 389
single-domain 388
single-domain encryption 755
encryption of data, changing 762
end nodes (APPN)
coding 507
combined APPN and subarea network 507
overview 7
resource registration 330
ENHADDR start option 34
enhanced addressing for session managers 34
enterprise system connection channel 258
environment, multiple-network
activating resources 612
combined APPN and subarea network 521
disjointed networks 521
network management with NetView 602
terminating sessions 619
error recovery activation 263
ESCON 258
ESIRFMSG start option 35
estimated number of active sessions (EAS)
APPL definitions 682
improving session establishment performance 282
LFBUF and 696
LU definitions 683
TSO, specifying for 716
Ethernet or Ethernet-type LAN 112, 258
exchange of ID processing
type 2.1 PUs 253
VTAM switched SDLC connection 266
exit routine
session accounting 306
session authorization 306
warning 306
explicit route 348
extended BIND 273
extended network addressing 591
extended network addressing (ENA)
addressing compatibility 361
addressing restrictions 361
compatibility, nonextended address 361
element address 360
extended recovery facility (XRF)
cryptography and 407
description 407
extended subnetwork boundaries 91
external communication adapter (XCA)
APPN connections 60
connection network, defining a connection to 67
link-attached NCP support 127
multiple-domain connections
Ethernet or Ethernet-type LAN 112
Fiber Distributed Data Interface network (FDDI) 112
token-ring network 112
NetView and 258
single-domain connections
Ethernet or Ethernet-type LAN 258
Fiber Distributed Data Interface network (FDDI) 258
token-ring network 258
tuning statistics 668
VCNS and 411
EZBDVIPA
disconnect 485
rebuild 484
EZBEPORT 485
disconnect 486
rebuild 486

F

failure, warning on initiation 404
FDDI 112, 258
Fiber Distributed Data Interface 112, 258
filters
NetView filter 625
session awareness data filter table 625
session monitor filter 625
VTAM filter 625
flow control, data 21
flow control, subarea to network 365
forced reactivation 621
frame relay lines, dynamic reconfiguration 244
FSIRFMSG start option 35

G

gateway NCP
  backup 563
  GWNAU definition statement 558
  native network
    BUILD definition statement 560
    GWNAU definition statement 561
    HOST definition statement 562
    NCP PATH definition statements 562
    PU definition statement 562
  nonnative network
    GWNAU definition statement 563
    NETWORK definition statement 562
gateway NCP backup 563
gateway VTAM
  CDRM definition statement 568
  defining 557
  GWPATH definition statement 568
  NETWORK definition statement 567
generic resources
  and application programs 406
  and TSO 483
  configuration, coupling facility failures for 457
  implementation considerations 456
  in a sysplex 448
  maintenance 464
  mapping 450
  members, initiating sessions with 454
  partner LU mapping 450
  requirements 448
GID operand on PATH definition statement 265
GROUP definition statement
  PAUSE operand 271
  POLIMIT operand 271
  PU operand 270
GTF trace facility 744
GWSSCP start option 555

H

HALT operator command 621
halting VTAM 621
Hardware Configuration Definition (HCD) 230
HAVAIL operand on APPL definition statement 407
HCD (Hardware Configuration Definition) 230
high performance data transfer MPC 51
high performance routing (HPR) 491
HOLD operand
  session termination, VTAM peripheral nodes 266
  VTAM peripheral nodes, overriding 267
HOST definition statement
  MAXDATA operand 126
  sample 126
  UNITSZ operand 126, 705
host physical unit name 29
host-to-host channel connection, APPN, defining 52
host, restart 631
HOSTPU start option
  changing 29
  ISTPUS major node 613
  multiple-network 555
HOT I/O detection and termination 700
HPDT MPC 51
HPR (high performance routing) 491
HPR start option (APPN) 495
HPRPST start option (APPN) 494

I

I/O (input/output) operations
  analyzing tuning statistics 668
  information gathering 668
IBM Open Systems Adapter
  ATM LAN emulation connections 68
  ATM native connections 69
  LAN connections 68
IBM Software Support Center, contacting xxiii
ICSF (Integrated Cryptographic Service Facility 755
IDBLK operand on PU definition statement
  type 2.1 PU 253
  VTAM switched SDLC connection 264
identifying host, single-domain 28
idle LU 6 420
IDNUM operand on PU definition statement
  type 2.1 PU 253
  VTAM switched SDLC connection 264
ILU (independent LU)
  adjacent link station 273
  characteristics 273
  defining 275
  description 273
  dynamic definition of 276
  dynamic selection of sessions 281
  extended BIND 273
  multiple connections 279
  NCP connections 281
  overview 13, 273
  restrictions 282
  sessions in a combined network 519
implicit dynamic reconfiguration 246
IMS, extended recovery facility (XRF) 407
independent LU (ILU)
  adjacent link station 273
  characteristics 273
  defining 275
  description 273
  dynamic definition of 276
  dynamic selection of sessions 281
  extended BIND 273
  multiple connections 279
  NCP connections 281
  overview 13, 273
  restrictions 282
  sessions in a combined network 519
Information APARs xxvi
Information Management System, extended recovery facility (XRF) 407
initial status
  overview 612
  warm start and 611
initiation failure 404
initiation, automatic 300
INOPCODE start option 35
INOPDUMP start option 35
input/output (I/O) operations
  analyzing tuning statistics 668
  information gathering 668
Integrated Cryptographic Service Facility (ICSF) 755
intensive mode recording (IMR) 651
interchange node (APPN)
  coding 506
  combined APPN and subarea network 506
  overview 10
interconnected networks, sample 574
intermediate routing node 359

Internet, finding z/OS information online xxviii
interpret table
defined 321
defining for TSO 725
LOGTAB operand 262, 322
nonswitched SDLC connection 262
sample 725
IOBUF buffer pool
application data transfer 698
coattailing guidelines 709
MAXBFRU operand and 705
overview 691
IOINT start option 35
IOPURGE start option 509, 532
IPL and dump contention 633
IPL, timed 638
ISO standards, NETID 29
ISTAPNPU generic representation 278, 488
ISTCDRDY major node 535, 613
ISTCOSDF 297, 299, 353
ISTDEFIN command list 231, 232, 766
ISTDINFO command list 231, 232, 763
ISTINCLM 353
ISTMSFLD default message flooding prevention table 616
ISTPDILU major node, initial status ande 613
ISTPUS major node
non-SNA devices and 256
overview 613
ISTVTCOS 354
ISTVTMEU translation file 770, 771
IVTPRM00 742

J

JCL for the logon manager 746

K

keyboard 785

L

LAN (local area network)
3172 Nways Interconnect Controller, single-domain connections 258
Ethernet or Ethernet-type LAN 258
FDDI 258
token-ring, XCA (3172) single-domain 258
type 2.1 PU and 254
XCA (3172), single-domain connections 258
LANG operand 288
LANGTAB operand 288
LBUILD definition statement
non-SNA devices, defining 255
sample 256
leased lines, APPN 59
LEN connections, converting to APPN 505
LEN node
adjacent link station list 519
combined APPN and subarea network 505
migrating to APPN nodes 505
overview 11
LFBUF buffer pool
application data transfer 698
EAS values and 696
guidelines for 696
LFBUF buffer pool *(continued)*
overview 692
TSO and 696
license, patent, and copyright information 789
limited resources, effect on CMIP services associations 420
limited resources, LU 6.2
description 419
non-LU 6.2 sessions and 419
limited resources, network management sessions 420
LIMQSINT operand on APPL definition statement 420
LIMRES operand on APPL definition statement 420
line control operand (LNCTL)
channel-attached NCP 120
logon manager 748
LINE definition statement
DELAY operand
CTC coattailing and 711
inbound coattailing and 704
NCP coattailing and 705
MODE operand 253
XMITDLY operand 261
lines
errors, recording and sending to VTAM 651
speed, reducing buffer use and 702
link stations
overview 626
PRI or SEC indicator 253
primary or secondary, determining 253
sample of activating 627
specifying during NCP activation 644
link-attached NCP
activating 644
LINE definition statement 128
overview 127
PU definition statement 128
recovery operations for 127
type 2.1 PUs, attached to 261
XCA (3172) and 127
link-level role negotiation, type 2.1 PU 253
links
activating 627
deactivating 627
overview 626
LIST option of START command 31
LISTBKUP (backup start option list) 30, 31
LNCTL operand on the GROUP definition statement
channel-attached NCP 120
logon manager 748
load module, NCP
identifying 125
multiple, maintaining 126
time stamps 124
verifying 124
LOAD operand 635
load station
IPL-capability of NCP and communication links 637
LOADSTA operand 637
selecting 637
loading a communication controller disk 643
loading an NCP
performance, improving 33
reloading, automatic (scheduled) 638
switched link, using 631
local area network (LAN)
3172 Nways Interconnect Controller, single-domain connections 258
Ethernet or Ethernet-type LAN 258

local area network (LAN) *(continued)*
FDDI 258
token-ring, XCA (3172) single-domain 258
type 2.1 PU and 254
XCA (3172), single-domain connections 258
LOCAL definition statement 256
locating resources
APPN network 17
subarea network 16
logical unit
addressing, dependent LU 283
dependent
addressing 283
combined subarea and APPN network 510
defining 283
overview 13
ownership of 283
secondary and 283
session establishment 283
description 253
name, unique 594
secondary, dependent and 283
types of 12
logon
automatic 300
passing request 380
validating request 380
LOGON exits, associated LU names 291
logon manager JCL 746
logon manager, defining
access
dependent logical unit 745
independent logical unit 745
APPL definition statement (operands) 748
configuration definition 749
defining the logon manager to VTAM 746
halting 753
installing 746
logon manager operation 745
logon manager, copying to system 746
monitoring 752
source statements
application password 750
buffers, maximum number 752
logon procedure 751
minimum channel link 750
reserved keywords 752
subarea configuration 749
subareas, maximum number 752
TPF applications, number 751
logon mode table
Class of Service and 352
combined APPN and subarea network 527
DYNMODTB start option 277, 293, 524
entry
Class of Service 399
DLOGMOD operand 399
DYNDLGMD start option 277, 293
name 594
parameters 399
session parameters 292
logon, automatic 300
LOGTAB operand
interpret table and nonswitched SDLC 262
interpret tables, defined 322
sample 262
LookAt xxvi
loop-adapter-attached devices 258
low-entry network node
adjacent link station list 519
combined APPN and subarea network 505
migrating to APPN nodes 505
overview 11
LPBUF buffer pool
CSALIMIT start option 683
expansion 683
overview 692
LSIRFMSG start option 35
LU
addressing, dependent LU 283
dependent
addressing 283
combined subarea and APPN network 510
defining 283
overview 13
ownership of 283
secondary and 283
session establishment 283
description 253
name, unique 594
secondary, dependent and 283
types of 12
LU 6.2
APPCCMD macroinstruction, requirement for LU 6.2 412
APPL definition statement
APPC operand 412
AUTOSES operand 416
DDRAINL operand 416
DMINWNL operand 416
DMINWNR operand 416
DRESPL operand 416
DSESLIM operand 416
SYNCLVL operand 419
VERIFY operand 418
ATTN exit 419
CNOS 415
conversation 413
description 412
encryption facility 386, 417
limited resources 419
LU-LU verification 417
mode name 413
resource verification reduction 414
session limit 415
session-level verification 419
sessions
automatic activation 416
characteristics 413
contention 413
establishment 413
negotiating number of 415
overview 413
sending data 413
terminating idle 419
termination 413
with LU 6.2 partners 413
sync point services 419
LU definition statement
EAS operand 683
ENCR operand 388
LU names, associated 291
LU-LU verification 417
LUDRPOOL definition statement 127, 238, 281
LUGROUP operand 284

LUSEED operand 284

M

mainframe
  education xxvi
major node names, list (ATCCONxx) 37
MAXBFRU operand
  coattailing and 704
  guidelines for setting 708
MAXDATA operand on PCCU definition statement 126
MAXSESS operand 282
MAXSUBA operand
  extended subarea addressing 361
  gateway NCP 556
  HOSTSA start option and 361
  logon manager JCL 746, 749
  nonextended addressing, communication with (SNI) 559
  nonnative networks 562
message flooding prevention 616
message skeleton file 771
messages
  module name, viewing 36
  suppression of 615
  translation of 769
  types 615
messages, USS
  defining 617
  Multicultural support 288
  translation of 769
migration data host (APPN)
  coding 507
  combined APPN and subarea network 507
  overview 10
migration, subarea to APPN 505
MMS 769
MODE operand on LINE definition statement 253
MODEENT macroinstruction
  coattailing and 705
  LANG operand 722
  Non-SNA 3270 devices 721
  PSERVIC operand 721
  SNA 3270 devices 721
model application program definitions 374
model CDRSC definition
  coding guidelines 538
  described 538
  resource state requirements 540
model definition 290
model major node 227
model name table
  overview 290
  sample 290
MODETAB operand on APPL definition statement 399
MODIFY CDRM operator command 548
MODIFY CHKPT operator command 333
MODIFY CNOS operator command 417
MODIFY CSM command 743
MODIFY DEFINE operator command, CNOS and 417
MODIFY DR operator command 243
MODIFY NEGPOLL operator command 271
MODIFY POLL operator command 271
MODIFY TABLE command 619
MODIFY TNSTAT operator command 668
module name, viewing 36
monitoring the domain
  DISPLAY command 615, 617
monitoring the domain *(continued)*
  messages
    suppression of 616
    types 616
moving PUs and LUs dynamically 235
MPC (multipath channel) connection for hosts
  connection for hosts 51
  tuning I/O 671
MPC, high performance data transfer 51
MPC, HPDT 51
MSGMOD start option 36
MTU 219
Multicultural support
  defining TSO to 728
  description 288
  LANG operand 288, 722
  LANGTAB operand 288
  MVS message service 769
  TSO/VTAM end-user messages 728
multinode persistent sessions
  configuration requirements 467
  configuration, coupling facility failures for 474
  establishing 470
  failure recovery processing 475
  overview 464
  planned takeover processing 478
  recovery does not occur or complete 481
  using multiple coupling facility structures for 470
multipath channel (MPC) connection for hosts
  connection for hosts 51
  tuning I/O 671
multipath channel connections, high performance data transfer (HPDT) 51
multipath channel connections, HPDT 51
multiple connections for a type 2.1 node 279
multiple operator consoles 618
multiple-network environment
  activating resources 612
  combined APPN and subarea network 521
  disjointed networks 521
  network management with NetView 602
  terminating sessions 619
multiple-network, verification 47
MVS Message Service 769
MVS operator functions for TSO 729
MVS system symbols 39

N

name
  class-of-service 595
  logon mode 594
  subarea-unique, single-domain 38
name assumption 572
name restrictions, single-domain 38
Native ATM connections
  connection to the Open Systems Adapter 69
  port on the Open Systems Adapter 71
  TGs over SVCs 75
native operand for subnetwork boundaries 94
NCP
  activating 643
  amount of data sent to VTAM, MAXBFRU operand 126
  APPN connections
    assigning addresses 489
    exhausting session control blocks 488
    maximum number of sessions 488

NCP *(continued)*
attaching to VTAM, overview 115
automatic logon (SNA sessions) 300
backup of owner 631
BUILD definition statement 115
channel unit address 116
channel-attached to VTAM 116
CMC configuration 221
coattailing for 705
congestion, reducing 702
connecting hosts through 106
connection networks 66
controlling in CMC environment
channel-attachment major node 223
dynamic switched major node 228
NCP major node 223
resource ownership 224
switched major node 224
CWALL threshold condition 667
deactivating 653
defining
BUILD definition statement 124
GROUP definition statement 117
HOST definition statement 126
PCCU definition statement 122
dial-in (to VTAM), overview 131
dump
automatic 123
dynamic 654
performance, improving 33
remote 129
selecting a dump disposition 656
selecting a dump file 656
static (destructive) 654
storing 123
dynamic control facilities 127
enhanced remote controller support 129
enterprise system connection channel 258
generation, overview 635
increasing data transfer speeds 702
independent LU 281
line speed, reducing buffer use 702
link-attached NCP
activating 644
LINE definition statement 128
link and link-station definition 128
overview 127
PU definition statement 128
XCA (3172) and 127
links
activating 627
deactivating 627
load module, verifying 124
loading
overview 635
performance, improving 33
reloading, automatic (scheduled) 638
remote 129
loading and activating using switched link
PCCU definition statement 649
VARY ACT operator command 649
monitoring 651
NCP definition statement 115
ownership of 633
owning discontiguous domains 630
pacing
overview 21
NCP *(continued)*
pacing *(continued)*
reducing congestion 21, 702
parallel transmission groups 118
peripheral nodes, attached to 261
PIU size, MAXDATA, setting for 126
polling, reduction due to slowdown 702
recording and sending line errors to VTAM 651
reload, automatic 123
remote 129
sharing in combined APPN and subarea network 515
sharing of 632
SLODN tuning statistic 667
slowdown, controlling 702
storage, displaying 652
switched SDLC 130
token ring connections (NTRI) 268
transmission groups, VTAM and NCP 118
tuning I/O for 668
tuning statistics 667
NCP, peripheral nodes attached to
binary synchronous (BSC) connection 270
nonswitched SDLC connection 262
switched SDLC connection 262
token-ring connection 268
NCPBUFSZ start option 33
NETID
ISO standards for 29
single-domain 29
start option 29, 555
NETSRVR definition statement, SLUINIT operand 100
NetView program
alias name translation
adjacent SSCP tables and 584
defining 605
overview 593
APPL definition statement 603
defining to VTAM 603
network management with 602
XCA (3172) and 258
network accessible units 12
network ID, dynamic 563
network identification 556
network node server (APPN)
list
defining 100
replacing 101
sample 101
overview 8
resource registration 330
supporting SLU-init sessions 100
network nodes (APPN), overview 6
network qualified names
CDRSCs and 570
concurrent sessions with independent LUs 274
defining the NetView program and 603
overview 585
Network Registry, SNA 29
network routing 348
NMVTLOG start option 36
NODELST data set 606
nodes
APPN 5
subarea 9
subarea and APPN 10
types of 5
NODETYPE start option 506, 507, 508

non-SNA devices
cluster controllers, defining 255
data-stream characteristics 412
deactivating 255
defining 255
loop-adapter-attached 258
nonnative network connection 254, 600
nonswitched SDLC connection
multiple domain, NCP to domain 106
single domain, NCP to peripheral node 262
NQNMODE start option 535, 570
NSRU (unsolicited network services request unit) 624
NTRI (NCP/Token-Ring Interconnection)
defining a connection to a connection network 66
description 268
NUMILU operand on the LUDRPOOL definition statement 281

O

Open Systems Adapter
ATM LAN emulation connections 68
ATM native connections 69
LAN connections 68
operands, dynamically changing 235, 238
operator commands
DISPLAY BFRUSE 665
MODIFY NEGPOLL 271
MODIFY POLL 271
MODIFY TNSTAT 668
operator-initiated search 328
OPNDST macroinstruction, OPTCD operand 381
OPTCD operand on OPNDST macroinstruction 381
OSA
ATM LAN emulation connections 68
ATM native connections 69
LAN connections 68
outages, reducing application program 407
OWNERID 743

P

pacing control for sessions
congestion, reducing 702
congestion, route 367
defining values, overriding defined pacing counts 312
overview 307
pacing of data flow 21
pacing, virtual route 365
parallel sessions 381, 591
parallel transmission groups (TG), NCP, channel-attached 118
parmlib member, CSM 742
PARSESS operand on APPL definition statement 381, 591
password verification, two-way 110
PATH definition statement
dependent LU requester, defining 512
GID operand 265
NTRI connection
DIALNO operand 270
DLCADDR operand 270
NTRI connection, for 269
PID operand 265
VTAM switched SDLC connection 265
path definition, dynamic update
description and sample 369
restrictions 370
path update, dynamic 369
paths
forcing a route (APPN) 773
logical 350
multiple-domain environment 359
multiple-network environment 586
network routing for subarea nodes 348
physical 348
routing in a subarea network 348
single-domain environment 348
PAUSE operand on GROUP definition statement 271
PCCU definition statement
AUTOSYN operand 124
description 122
multiple 123
sample, channel-attached 116
SUBAREA operand 123
VFYC operand 124
VFYLM operand 124
peak traffic, unusual 21
performance
APPN
checkpointing 333
improving search performance 329
buffer pools specifications 691
collecting data 668
common service area (CSA) 683
input/output (I/O) operations 668
introduction 663
NCP slowdown 664, 702
NCP tuning statistics 667
session pacing 307
performance monitor interface 667
peripheral link 349
peripheral nodes
attaching to NCP 261
attaching to VTAM 255
controlling 221
loop-adapter-attached devices 258
SNA major node, local 256
peripheral subnetwork boundaries 91
PERSIST operand on ACB macroinstruction 409
persistent LU-LU sessions
description 409
multinode 464
sync point and 419
physical unit
adding dynamically 235
attaching to VTAM 255
BSBUF buffer pool start option 274
deleting dynamically 235
loop-adapter-attached 258
moving dynamically 235
non-SNA devices 255
SNA devices
defining 256
operator-assisted connect 257
type 2.1
channel, defining 254
characteristics 253
combined APPN and subarea network 505
CPNAME operand and 253
identifying 253
independent LU 273
local area network 254
MODE operand and 253
NCP-attached 261

physical unit *(continued)*
type 2.1 *(continued)*
non-SNA major node 255
nonnative network connection 254
restrictions, alias name 274
restrictions, independent LU 274
session capabilities 273
session control block, maintaining 274
session establishment 253
sessions, maximum number 282
XID services 253
types of 12
PHYSRSC operand on PU definition statement
OWNER operand and 628
single-domain, in a 648
PID operand on PATH definition statement 265
POLIMIT operand on GROUP definition statement 271
polling terminals (NCP)
changing parameters for 271
reduction due to slowdown 702
pools, CSM buffer
size of 741
source of 741
PPOLOG start option 36
predefined cross-domain resource 575
prerequisite information xxvi
private storage, allocating 412
problem diagnosis
alerts to NPDA 36
module name, viewing 36
NMVTLOG start option 36
SDLC statistical MDRs 36
start options 35
time interval notification 35
trace facility 37
program operators
associating with USS tables 618
defined 623
making decisions on setting up 623
protection of subarea dial 110
PSERVIC operand 721
PSSTRACE start option 36
PU
adding dynamically 235
attaching to VTAM 255
BSBUF buffer pool start option 274
deleting dynamically 235
loop-adapter-attached 258
moving dynamically 235
non-SNA devices 255
SNA devices
defining 256
operator-assisted connect 257
type 2.1
channel, defining 254
characteristics 253
combined APPN and subarea network 505
CPNAME operand and 253
identifying 253
independent LU 273
local area network 254
MODE operand and 253
NCP-attached 261
non-SNA major node 255
nonnative network connection 254
restrictions, alias name 274
restrictions, independent LU 274
PU *(continued)*
type 2.1 *(continued)*
session capabilities 273
session control block, maintaining 274
session establishment 253
sessions, maximum number 282
XID services 253
types of 12
PU definition statement
ANS operand 263
CONNTYPE operand 487
CPCP operand 487
CPNAME operand, type 2.1 PU and 253
IDBLK operand 253
IDNUM operand 253
PHYSRSC operand
OWNER operand and 628
single-domain, in a 648
PU name and CPNAME uniqueness 505
PU operand on GROUP definition statement 270
PUDRPOOL definition statement, NCP type 2.1 and 127, 238, 261
purging a communication controller disk 643

R

rapid transport protocol (RTP) 492
reactivation, forced 621
reconfiguration, dynamic
effect on WARM start 608
frame relay lines, PUs on 244
implicit dynamic reconfiguration 246
overview 235
samples 246
VARY ACT, UPDATE technique 238
VARY DRDS technique 241
recovery
cryptography and 407
description 407
recovery operations, link-attached NCP, for 127
registering resources 330
reloading a controller, automatic 638
renaming NCP load modules 641
resource control, cross-domain
CDRSC definition statement 537
controlled by another VTAM domain 536
cross-domain resource major node 536
dynamic adjacent SSCP table 543
locating cross-domain resources 541
network resources for XRF 537
USERVAR with XRF 537
resource control, single-domain
activating
automatic activation of cross-subarea links 644
order of activation 611
overview 610
deactivating
automatic deactivation of cross-subarea links 620
forced deactivation 621
forced reactivation 621
immediate deactivation 621
normal deactivation 620
order of deactivation 620
resource definition statements
coding 38
sample 39
storing 39

resource definition statements *(continued)*
updating 39
resource definition, dynamic (model name) 290
resource ownership, SSCP 631
resource registration 330
resource resolution table (RRT)
description 635
time stamps 124
resource verification reduction 414
resource, restoring
defined 606
NODELST data set 606
WARM start 606
resources in other networks
adjacent SSCP (ADJSSCP) table, defining 579
adjacent SSCP lists for CDRSCs 542
CDRM ownership
changing by operator 549
changing dynamically 548
verifying 548
combined APPN and subarea network 515
defining 515, 534
definition statement, VFYOWNER operand 548
description 579
dynamic 576
locating
adjacent SSCP table 541
ADJCDRM definition statement 543
CDRSC routed to each SSCP 541
cross-network 579
multiple-network configuration 580
multiple-network environment 570
performance improving 546
predefined 574
SSCPDYN start option 546
SSCPORD start option 546
response time
channel-to-channel connection 712
coattailing and 682, 702
DELAY operand and 704, 707
overview 663
RESSCB operand on the LU definition statement 282
restart a host 631
restoring resource
defined 606
NODELST data set 606
WARM start 606
restricted materials 781
restrictions
addressing 362
independent LU 274
name, single-domain 38
subarea addressing 364
RFC (request for comments)
accessing online xxviii
route
assigning session traffic to a 352
explicit 348
extension 351
planning 357
virtual 350
route congestion 367
routing
APPN
broadcast search 327
directed search 326
directory services 325
routing *(continued)*
APPN *(continued)*
failure message, CNN routing 344
forcing a route 773
improving efficiency 329, 414
local topology database 18
network topology database 18
topology database 325
NETDA/2 (route planning tool) 357
network 348
overview 17
route calculation (APPN), overview 20
routing tables 349
selection of route, subarea 17
topology database (APPN), overview 18
routing subarea node
address incompatibility 362
addressing methods 361
extended addressing compatibility 364
extended network addressing 361
path definition, subarea 359
subarea addressing restrictions 364
routing, cross-network
communication between network 585
COS tables 590
extended subarea address gateway 585
extended subarea addressing 585
gateway NCP 585
gateway VTAM 585
virtual route, default 590
routing, network, subarea nodes
addressing compatibility 361
addressing restrictions 361
compatibility, nonextended address 361
element address 360
RRT (resource resolution table)
description 635
time stamps 124
RSIRFMSG start option 36
RTP (rapid transport protocol) 492

S

SALIMIT operand 562
samples
display buffer use command output 665
networks
multiple-domain 529
multiple-network 552
single-domain 27
tuning statistics report
channel-to-channel 670
SNA controller 668
SATOAPPN COS mapping table 527
SAW (session awareness) data, filtering 625
scheduled NCP reload 638
SDLC (synchronous data link control)
nonswitched connection, NCP to peripheral nodes 262
switched connection
dial-in (to VTAM) 264
dial-out (from VTAM) 265
NCP to peripheral node 262
testing links 652
SDLC statistical MDRs 36
SDLCMDRS start option 36
search reduction 32

searches
broadcast 327
controlling
combined APPN and subarea network 519
order of 519
subarea network searches 520
directed 326
eliminating, for unavailable resources 31
improving performance 333
limiting broadcast searches 521
limiting SNI searches 521
limiting subarea searches 520
operator-initiated 328
performance enhancements 329, 414
reducing, for unavailable resources 31
time interval, controlling 509, 532
types of 326
security facilities
confidential data 398
single-domain 386
security of data
application program 398
CDRM key 757
conversation-level 417
cross-domain cryptography 757
filing secondary logic unit (SLU) keys 755
session-level 417
single-domain encryption 755
security verification 110
selective termination of idle LU 6.2 sessions
description 419
non-LU 6.2 sessions and 419
selective termination of idle network management sessions 420
session awareness (SAW) data, filtering 625
session initiation request 571
session management exit routine (SME)
adjacent SSCP selection 306
alias selection function 593
for multiple-domain sessions 306
session accounting 307
session authorization 306
when called 306
session managers, enhanced addressing 34
session-partner resource 291
sessions
accounting for cross-domain 307
acquiring 381
adjacent SSCP selection 546
application program, controlling 300
APPN, setting maximum number 488
associated LU table 291
authorize, establish, or terminate 287
authorizing cross-domain 306
automatic logon 300
backup, extended recovery facility (XRF) 407
between logical units 287
changing an LU's automatic logon specification 300
combined APPN and subarea network 519
control blocks
controlling use of (APPN) 488
exhausting (APPN) 488
controlling 306
CP-CP (APPN), overview 14
cross-network session control, gateway SSCP 557
disruption, BSC 632
establishing 306
sessions *(continued)*
establishing and terminating with commands 619
establishment
dependent LU 283
type 2.1 PU 253
gateway path selection 587
gateway session accounting 307
LOGON exits 291
logon mode table 292
logon, automatic 300
LU-LU, overview 15
model name table 290
number of half-sessions 307
overview 14
pacing of data flow 21
parallel 381
persistent LU-LU 409
protocol 292
security, LU 6.2 session-level LU-LU verification 417
session accounting exit routine 306
session authorization exit routine 306
session management exit routine 306
SLU-init in network node server (APPN) 100
SNA network interconnection 307
SSCP-SSCP 14
tailoring
authorize or establish 287
between logical units 287
routines used 287
tables used 287
terminating in a multiple-network environment 619
termination 619
USERVAR, session failure 405
SFBUF buffer pool
guidelines for 696
overview 692
shadow resource
application programs 25
CDRSCs 536, 537
overview 549
takeover and 631
shortcut keys 785
shut down data transmission 367
sift-down coding 39
single-domain name restrictions 38
single-domain network
overview 25, 529
sample 27
SIRFMSG start option 36
skeleton file, message 771
SLODN tuning statistic 667
slowdown condition
analyzing 667
controlling NCP 702
SLUINIT operand on NETSRVR definition statement 100
SME (session management exit routine)
adjacent SSCP selection 306
alias selection function 593
for multiple-domain sessions 306
session accounting 307
session authorization 306
when called 306
SNA devices
cluster controllers, defining statically 256
loop-adapter-attached 258
SNA network interconnection (SNI) 554
SNA Network Registry 29

SNA protocol specifications 783
SNAPREQ start option 36
SNI (SNA network interconnection) 554
SNVC start option 93
softcopy information xxvi
SONLIM start option 698
SORDER start option 509, 519
SPBUF buffer pool, overview 692
SRCHRED start option 32
SRCLEAR operand on the MODIFY RESOURCE command 33
SRCOUNT start option 32
SRTIMER start option 32
SSCP
  identifier 28
  overview 9
  owning 595
SSCP resource ownership 631
SSCP takeover
  combined APPN and subarea network 516
  description 631
SSCP, gateway
  CDRM definition statement 568
  defining 557
  GWPATH definition statement 568
  NETWORK definition statement 567
SSCPDYN start option 546
SSCPID start option 28, 555
SSCPNAME start option 29, 555
SSCPORD start option 546
SSEARCH start option 509, 520
start option list (ATCSTR00) 27, 30, 253
start option list, backup (LISTBKUP) 30, 31
start options
  APPN
    APPNCOS 93
    BN 92
    BNDYN 92
    BNORD 93
    CDSERVR 330
    CDSREFER 329
    CONNTYPE 505
    CPCP 489
    DIRSIZE 331
    DIRTIME 331
    DYNADJCP 490
    HPR 495
    HPRPST 494
    INITDB 333
    NODETYPE 506, 507
    SNVC 93
    VERIFYCP 418
    VFYRED 415
    VFYREDTI 415
    VRTG 97
    VRTGCPCP 97
  AUTHLEN 282
  AUTORTRY 302
  AUTOTI 302
  CMPMIPS 382
  combined APPN and subarea network 508
  DISCNTIM 267
  DYNDLGMD 277, 293
  DYNLU 490
  DYNMODTB 277, 293, 524
  ENHADDR 34
  gateway VTAM 555
start options *(continued)*
  IOINT start option 35
  IOPURGE 509, 532
  MSGMOD start option
    alerts to NPDA 36
    definition 36
    VTAM trace facility 37
  multiple-domain
    performance improvement 530
    performance problems 530
  multiple-network
    GWSSCP 555
    HOSTPU 555
    NETID 555
    SSCPID 555
    SSCPNAME 555
  NODETYPE 506, 507, 508
  NQNMODE 535
  overriding 30
  overview 25, 28
  performance 530
  prompting for start options 30
  single-domain
    creating start options 31
    host physical unit name 29
    LISTBKUP 30, 31
    NCPBUFSZ 33
    NETID 29
    sample 31
    source of start options 30
  SORDER start option 509, 519
  SRCHRED start option 31
  SRCOUNT start option 31
  SRTIMER start option 31
  SSCP identifier 28
  SSEARCH start option 509, 520
  SWNORDER 253, 264
  XNETALS start option 254
start-stop devices, communicating with 412
start, warm
  configuration restart 606
  initial status and 611
starting
  application program
    name of application and 374
    security 398
  VTAM
    configuration list 37
    creating start options 31
    general start procedures, MVS 605
    problem diagnosis 35
    prompting for start options 30
    sample start options, single-domain 31
    source of start options 30
    start options 28
status information, resource
  defined 606
  NODELST data set 606
  WARM start 606
storage
  allocating private for application program 412
  balancing storage needs 663
  calculating waste 666
  displaying NCP storage 652
  monitoring buffer use 665
  shortage of 684
  VTAM uses 663

storage constraints, CSM 743
storage estimates, APPL EAS 738
storage limits 742
storing resource definition statements 39
subarea node address
  address incompatibility 362
  addressing methods 361
  extended addressing compatibility 364
  extended network addressing 361
  path definition, subarea 359
  subarea addressing restrictions 364
subarea nodes, network routing for
  dynamic path update 369
  extended network and subarea nodes 364
  nodes, nonextended and extended 361
  PATH definition statements 348
SUBAREA operand on PCCU definition statement 123
subarea to network data control 365
subarea-unique name, single-domain 38
substituting COS parameters 354
SUBSTUT operand on COS macroinstruction 354
SVCs, ATM, , defining TGs over 75
switched SDLC connection
  dial-in to VTAM 264
  dial-out from VTAM 265
  dynamic definition of 227
  multiple-domain, NCP to domain 108
  PATH definitions, VTAM connections 265
  single-domain
    NCP to peripheral node 262
    VTAM to NCP 130
switched virtual channels, ATM, defining TGs over 75
SWNORDER start option 253, 264
sync point services, LU 6.2
  description 419
  persistent LU-LU sessions and 419
synchronous data link control (SDLC)
  nonswitched connection, NCP to peripheral nodes 262
  switched connection
    dial-in (to VTAM) 264
    dial-out (from VTAM) 265
    NCP to peripheral node 262
  testing links 652
SYNCLVL operand on APPL definition statement 419
syntax diagram, how to read xxiv
SYS1.VTAMLST, start options 31
Sysplex Wide Security Associations 484
sysplex, dynamic definition of VTAM-to-VTAM connections in a 445
sysplex, functions provided by VTAM in a 435
Sysplexports 485
system symbols, MVS 39

T

table, replacing 619
takeover of resources
  combined APPN and subarea network 516
  description 631
tasks
  (generic resource, removing)
    steps 458
TCP/IP
  online information xxviii
Technotes xxvi
terminals
  3277 255
terminals *(continued)*
  control address space (TCAS) 716
  non-SNA 255
  polling, changing parameters for 271
  taking off-line 255
termination of idle sessions, effect on associations 420
TIC (token-ring interface connector)
  address 269
  dump 658
time sharing
  buffers 725
  cataloged procedures 725
time stamps, RRT and load module 124
timed IPL 638
TNSTAT start option 37, 668
token-ring connection
  APPN, sample 3172 Nways interconnect controller 60
  NCP/Token-Ring Interconnection (NTRI) 111, 268
  XCA connections
    VTAM, multiple domain 112
    VTAM, single domain 258
token-ring interface connector (TIC)
  address 269
  dump 658
topology agent 3
topology agent, overview 427
topology and routing services (TRS) 325
topology database (APPN)
  checkpointing 333
  creating 325
  improving performance 333
  local, overview 18
  network, overview 18
  overview 18
  topology exchange 325
TP (transmission priority)
  authorized, for LEN connections 282
  description 351
TPF (Transaction Processing Facility)
  access
    dependent logical unit 745
    independent logical unit 745
  APPL definition statement (operands) 748
  configuration definition 749
  defining the logon manager to VTAM 746
  halting 753
  installing 746
  logon manager operation 745
  logon manager, copying to system 746
  monitoring 752
  source statements
    application password 750
    buffers, maximum number 752
    logon procedure 751
    minimum channel link 750
    reserved keywords 752
    subarea configuration 749
    subareas, maximum number 752
    TPF applications, number 751
trace facility 37
tracing CSM 743
trademark information 797
Transaction Processing Facility (TPF)
  access
    dependent logical unit 745
    independent logical unit 745
  APPL definition statement (operands) 748

Transaction Processing Facility (TPF) *(continued)*
configuration definition 749
defining the logon manager to VTAM 746
halting 753
installing 746
logon manager operation 745
logon manager, copying to system 746
monitoring 752
source statements
application password 750
buffers, maximum number 752
logon procedure 751
minimum channel link 750
reserved keywords 752
subarea configuration 749
subareas, maximum number 752
TPF applications, number 751
translation, alias name
adjacent SSCP tables and 584
alias selection function (VTAM) 593
defining 595
LU 6.2 599
NetView 603
overview 593
transmission groups (TG)
boundary function-based (APPN) 50
parallel
NCP, channel-attached 118
overview 50
parallel sessions 368
PCCU definition statement 118
PU definition statement 118
VR-based 95
transmission priority (TP)
authorized, for LEN connections 282
description 351
TRL major node 52
TRLE operand on the PU definition statement 52
TRS (topology and routing services) 325
TSO Operator 729
TSO/E sessions 715
TSO/VTAM
2741, TWX, or WTTY 724
3270 characteristics 720
3270 large screen considerations 727
3790/3270 configuration 724
definition 715
estimating active sessions for (EAS) 683
exit routines 726
implementing 726
interpret table, defining 725
LFBUF buffer pool and 696
messages, end-user 728
multiple-domain network 716
performance 727
security messages 726
session parameters 720
single-domain network 716
TCAS application, defining 716
TCAS program properties, defining 726
translation of messages 769
translation table 726
tuning I/O operations
analyzing tuning statistics 668
information gathering 668
tuning statistics
blocking outbound PIUs, analyzing 682
tuning statistics *(continued)*
channel-to-channel adapter 670
data transfer, analyzing 681
gathering 668
host processor use, analyzing 681
I/O buffer size, analyzing 681
performance monitor interface 667
read attentions (RDATN) 681
record format for SNA controller 669
slowdown, analyzing 667
SNA controllers 668
tuning your network
APPN
checkpointing 333
improving search performance 329
buffer pools specifications 691
collecting data 668
common service area (CSA) 683
input/output (I/O) operations 668
introduction 663
NCP slowdown 664, 702
NCP tuning statistics 667
session pacing 307
type 1 physical units 235
type 2 physical units 235
type 4 subarea node 103
type 5 subarea node 103

U

UNITSZ operand on HOST definition statement 126, 705
unsolicited network services request unit (NSRU) 624
UPARM 773
updating resource definition statements 39
user variable (USERVAR), classes
class
automatic 401
user-managed 400
type
dynamic 401
static 401
volatile 402
USERVAR (user variable), classes
class
automatic 401
user-managed 400
type
dynamic 401
static 401
volatile 402
USS commands, defined 617
USS messages
defining 617
Multicultural support 288
translation of 769
USS table
associating with program operators 618
changing, modifying 617, 619
logon and logoff requests from logical units 321
Multicultural support 288
operation-level 617
session-level 321

V

VCNS
  overview 411
  XCA (3172) and 411
VCNS operand on APPL definition statement 411
VERALSID operand on PU definition statement 264
verification, LU-LU 417
VERIFYCP start option 418
verifying a multiple-domain network 46
verifying single-domain network 46
VFYC operand on PCCU definition statement 124
VFYLM operand on PCCU definition statement 124
VFYOWNER operand on CDRSC definition statement 548
VFYRED start option 415
VFYREDTI start option 415
virtual node 64
virtual route (VR)
  coattailing and 712
  description of 350
virtual route pacing 365
VN=YES operand on the ADJCP definition statement 67
VNGROUP operand on the LINE definition statement 66
VNGROUP operand on the PORT definition statement 67
VNNAME operand on the LINE definition statement 66
VNNAME operand on the PORT definition statement 67
VR (virtual route)
  coattailing and 712
  description of 350
VR-based TG 95
VRTG operand on the CDRM definition statement 97
VRTG start option 97
VRTGCPCP operand on the CDRM definition statement 97
VRTGCPCP start option 97
VSAM configuration restart file
  associating with major nodes 607
  CONFGDS operand 607
  CONFGPW operand 607
VTAM Common Network Services
  overview 411
  XCA (3172) and 411
VTAM internal trace 743
VTAM restricted materials 781
VTAM terminal I/O coordinator (VTIOC) 715
VTAM topology agent 3
VTAM topology agent, overview 427
VTAM-to-VTAM connections, dynamic definition of 445
VTAM, defined as
  CDRM major node 532
  CDRM minor node 533
  cross-domain resource manager (CDRM) 532
  external CDRM 532
  host CDRM 532
VTAM, gateway
  CDRM definition statement 568
  defining 557
  GWPATH definition statement 568
  NETWORK definition statement 567
VTAM, online information xxviii
VTAMLST file, sample 39
VTAMSEG major node 613
VTAMSEG2 major node 613
VTIOC (VTAM terminal I/O coordinator) 715

W

warm start
  configuration restart 606
  initial status and 611
warning
  changing messages 618
  exit routine 306
window size, adaptive pacing 309

X

XCA (external communication adapter)
  APPN connections 60
  connection network, defining a connection to 67
  link-attached NCP support 127
  multiple-domain connections
    Ethernet or Ethernet-type LAN 112
    Fiber Distributed Data Interface network (FDDI) 112
    token-ring network 112
  NetView and 258
  single-domain connections
    Ethernet or Ethernet-type LAN 258
    Fiber Distributed Data Interface network (FDDI) 258
    token-ring network 258
  tuning statistics 668
  VCNS and 411
XCF
  dynamic definition of VTAM-to-VTAM connections 445
  setting up the sysplex environment 436
XDBUF buffer pool, overview 693
XID operand for an APPN connection 488
XID processing
  type 2.1 PUs 253
  VTAM switched SDLC connection 266
XMITDLY operand on LINE definition statement 261
XNETALS start option 254
XRF (extended recovery facility)
  cryptography and 407
  description 407

Z

z/OS Basic Skills information center xxvi
z/OS Basic Skills Information Center xxvi
z/OS, documentation library listing 799

Communicating your comments to IBM

If you especially like or dislike anything about this document, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Please send your comments to us in either of the following ways:

- If you prefer to send comments by FAX, use this number: 1+919-254-1258
- If you prefer to send comments electronically, use this address:
  - comsvrcf@us.ibm.com
- If you prefer to send comments by post, use this address:

```
International Business Machines Corporation
Attn: z/OS Communications Server Information Development
P.O. Box 12195, 3039 Cornwallis Road
Department AKCA, Building 501
Research Triangle Park, North Carolina 27709-2195
```

Make sure to include the following in your note:

- Title and publication number of this document
- Page number or topic to which your comment applies.

© Copyright IBM Corp. 1984, 2011

Program Number: 5694–A01

Printed in USA

SC31-8777-11