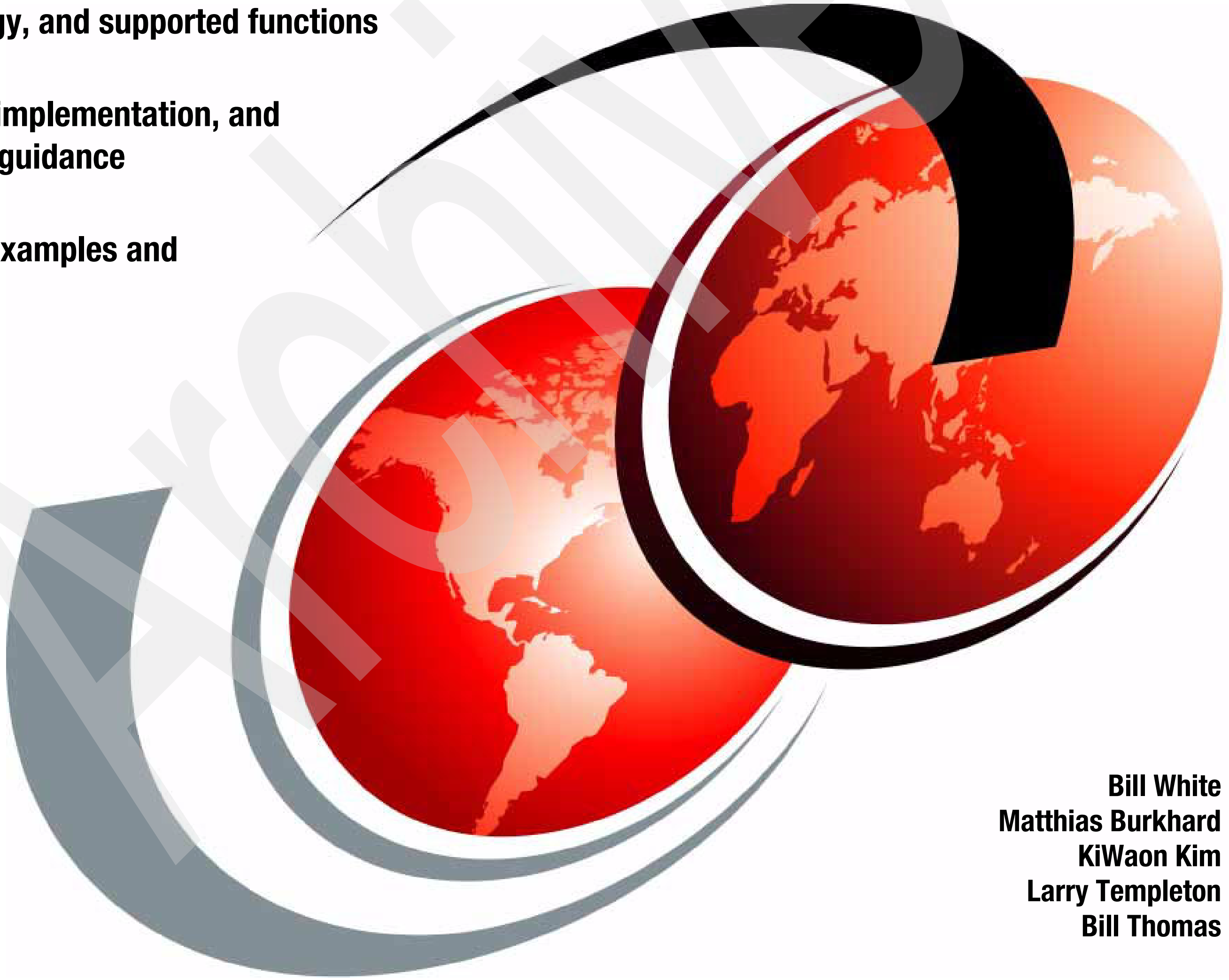
IBM

Enterprise Extender Implementation Guide

- Enterprise Extender concepts, terminology, and supported functions
- Planning, implementation, and migration guidance
- Realistic examples and scenarios



Bill White
Matthias Burkhard
KiWaon Kim
Larry Templeton
Bill Thomas

Redbooks

ibm.com/redbooks

International Technical Support Organization

Enterprise Extender Implementation Guide

April 2007

SG24-7359-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (April 2007)

This edition applies to Version 1, Release 8 of Communications Server for z/OS.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer 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.

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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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.

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 illustrate 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.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

i5/OS®
z/OS®
z/VM®
z/VSE™
z9™
Advanced Peer-to-Peer Networking®
AnyNet®
AIX®
AS/400®
DB2®
ESCON®
FICON®
HiperSockets™
IBM®
MVS™
NetView®
OS/2®
OS/400®
Redbooks®
Redbooks (logo) ®
RACF®
System i™
System p™
System z™
System z9™
VTAM®

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication will help you to tailor and configure Communications Server for z/OS® (CS z/OS) to make full use of Enterprise Extender (EE) capabilities. It focuses on the migration of your Advanced Peer-to-Peer Networking (APPN) environment to Enterprise Extender, while offering easy-to-understand, step-by-step guidance. Sample scenarios are provided that discuss Enterprise Extender connections between multiple z/OS systems as well as between z/OS and non-mainframe systems. The non-mainframe platforms in our examples include IBM Communications Server for AIX (CS/AIX), IBM Communications Server for Linux (CS Linux), IBM Communications Server for Windows (CS Windows), IBM Personal Communications for Windows (PCOMM), and i5/OS® Enterprise Extender support.

This publication provides information to assist you with the planning, implementation, and setup of Enterprise Extender. In addition, it describes helpful utilities and commands that you can use to monitor and operate the Enterprise Extender environment. It discusses the motivation for migrating to Enterprise Extender and explains the planning decisions that must be considered before attempting each phase of a migration. Further, it illustrates working scenarios, highlighting the important aspects of configuration and connectivity, and discusses techniques for avoiding undesirable situations. It explains the changes that are necessary in the VTAM® and TCP/IP definitions to support Enterprise Extender in each scenario.

You should have a solid background in SNA, APPN, and TCP/IP networking, as well as experience in the setup and operation of Communications Server for z/OS (VTAM and TCP/IP). Whether you are a systems engineer, network administrator, or system programmer that will plan for and configure Enterprise Extender, this book will be useful to you. Enterprise Extender requires an APPN/HPR environment. Because base APPN functionality has been available on z/OS for years, and many product manuals and Redbooks have been written on basic APPN implementation, this publication does not cover the details of migrating from an SNA subarea environment to a base APPN environment. That information can be obtained from existing publications. However, adding HPR support to the base APPN environment is explained and described in detail, with examples.

We assume that you have working knowledge of or have a z/OS networking environment, that is stable in both APPN and TCP/IP base functions. The definitions that are used to establish the basic APPN and TCP/IP network for our scenarios are included and discussed in the appropriate chapters. The definitions provide a starting point for the migration from base APPN, toward APPN/HPR, and finally to HPR/IP (EE).

Our implementation environment for this book

Given the complexity of our test environment, we needed to be creative in organizing the definitions so that each scenario could be managed and tested with minimal coordination with (and interference from) the other scenarios. To enable concurrent work on each of the scenarios, we set up the following test environment as illustrated in Figure 1 on page xiv.

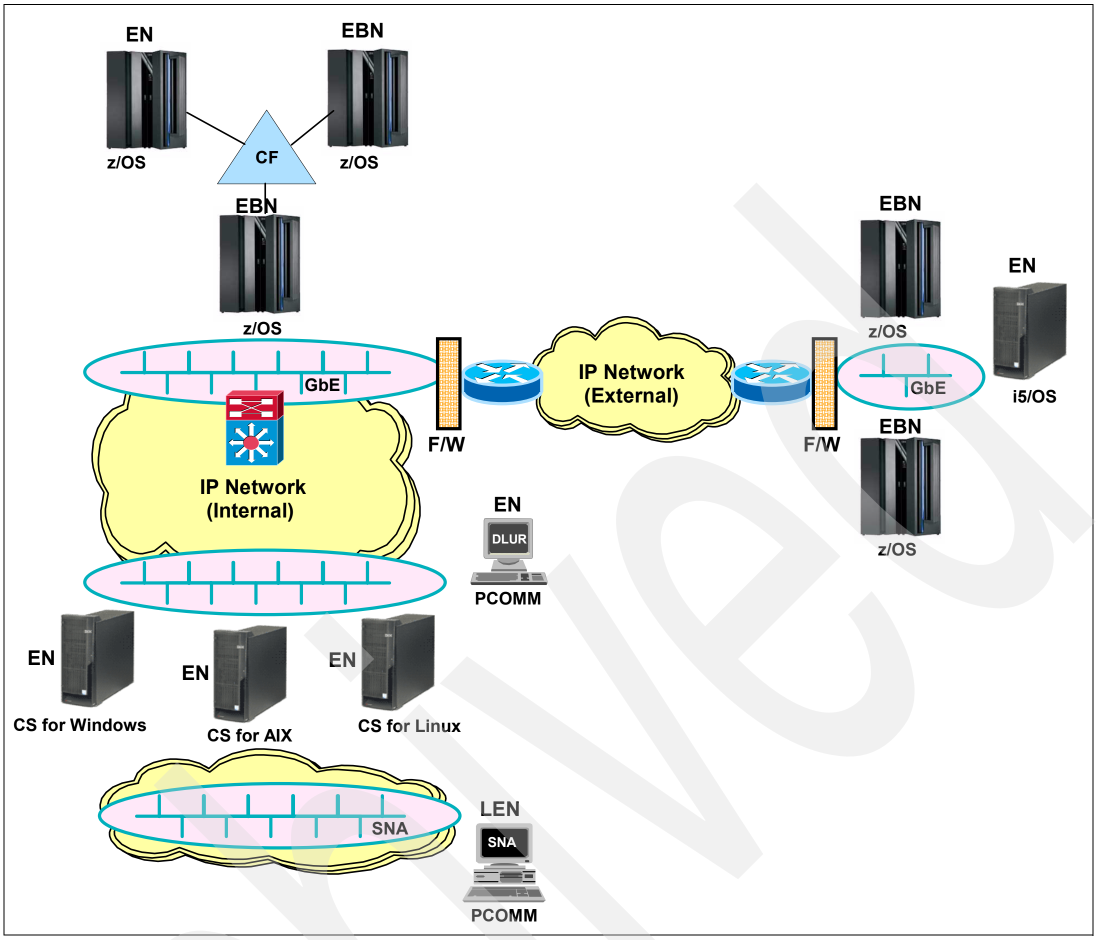


Figure 1 Enterprise Extender lab diagram

Our lab environment and examples were prepared using six z/OS logical partitions (LPARs) on an IBM System z9™ server, and a number of midrange platforms including Personal Communications (PCOMM), CS Windows®, Communications server for AIX® (CS/AIX), and CS Linux®, as described in Table 1.

Table 1 Enterprise Extender lab systems

System	NETID	CP name	Node type
CS z/OS	RDBOOKEE	SC30M	EBN
CS z/OS	RDBOOKEE	SC31M	EBN
CS z/OS	RDBOOKEE	SC32M	EN
PCOMM	RDBOOKEE	EECPCOM	DLUR
CS Windows	RDBOOKEE	EECPWIN	EN
CS/AIX	RDBOOKEE	EECPAIX	EN
CS Linux	RDBOOKEE	EECPLNX	EN
CS z/OS	USIBMSC	SC47M	EBN
CS z/OS	USIBMSC	SC69M	EBN
CS z/OS	USIBMSC	SC42M	EN

The team that wrote this IBM Redbooks publication

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Bill White is a Project Leader and Senior Networking Specialist at the International Technical Support Organization, Poughkeepsie Center.

Matthias Burkhard is the Technical Leader of IBM Networking Software Support that is located in Mainz, Germany. He holds a Diploma (FH) in Electrical Engineering from Fachhochschule Wiesbaden. With 17 years experience in SNA support and 10 years in TCP/IP support, he is familiar with both protocols on all platforms that communicate with IBM mainframes. He has consulted and supported many customers in their early migrations to APPN, HPR, and EE networks in Germany and Central Europe. He also teaches networking implementation and debugging classes, and he co-authored *Migrating Subarea Networks to an IP Infrastructure Using Enterprise Extender*, SG24-5957.

KiWaon Kim is an IT Specialist at IBM Korea. He has seven years of experience with IBM mainframe, System p™ servers, and network protocols, such as TCP/IP and SNA, and he has worked at IBM for seven years. His current responsibilities include network related problem solving in System z™ and System p. He also helps clients to design and implement network infrastructures in multiprotocol environments.

Larry Templeton is a Network Architect with IBM Integrated Technology Delivery, Network Services. He has 37 years of experience with IBM mainframe and networking systems, consulting with clients throughout the world. His current responsibilities include designing mainframe IP connectivity and load balancing solutions, designing inter-company Enterprise Extender configurations, and assisting clients with high-availability data center implementations.

Bill Thomas is a Certified IT Infrastructure Architect with IBM Integrated Technology Delivery, Service Management in Australia. He has 25 years of experience with mainframe networking, routing, and switching. His current responsibilities include designing IP networking and Enterprise Extender solutions.

Thanks to the following people for their contributions to this project:

Bob Haimowitz
International Technical Support Organization, Raleigh Center

Alfred Christensen
IBM Enterprise Networking and Transformation Solutions, Raleigh

Sam Reynolds
IBM Enterprise Networking and Transformation Solutions, Raleigh

Mike Riches
Global Technology Services, IBM UK

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review IBM Redbooks form found at:

  `ibm.com/redbooks`

- Send your comments in an email to:

  `redbooks@us.ibm.com`

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Introduction to APPN and Enterprise Extender

This chapter defines and positions Advanced Peer-to-Peer Networking® (APPN) and Enterprise Extender (EE). The benefits of APPN and EE are discussed and the relationship between the two is explained. The discussion includes a review of the base APPN terminology and functions that must be in place before beginning a migration to EE. Finally, we introduce the functions of EE.

This chapter contains the following introductory discussions:

- “What is Enterprise Extender?”
- “What is APPN and APPN/HPR?”
- “What is the importance of Enterprise Extender?”
- “Why is APPN/HPR required by Enterprise Extender?”

1.1 What is Enterprise Extender?

The Enterprise Extender (EE) technology was implemented on most of IBM networking platforms during 1998. Its main objective is to provide SNA-over-IP integration that is significantly superior to its predecessors, such as Data Link Switching (DLSw) and AnyNet®. AnyNet is no longer supported in Communications Server for z/OS V1R8 and higher.

EE is valid only for APPN configurations and is an extension of APPN and High Performance Routing (HPR) protocols. Your subarea network must first be migrated to APPN for the subareas where EE is to be deployed.

Implementing EE is a very simple process and requires little change to your APPN-enabled network. There are four categories of definitions needed for VTAM:

- Start options
- EE XCA major node
- Model PU major node definition
- Switched major node definitions

There are three definitions needed for the TCP/IP stack profile:

- A static VIPA for EE partner communication
- The IUTSAMEH Device for VTAM-to-TCP/IP communication
- A port reservation for the EE ports

EE uses APPN's HPR technology to provide encapsulation of SNA application traffic within UDP frames by HPR-capable devices at the *edges* of an IP network. These edge devices are called *endpoints*. This is illustrated in Figure 1-1.

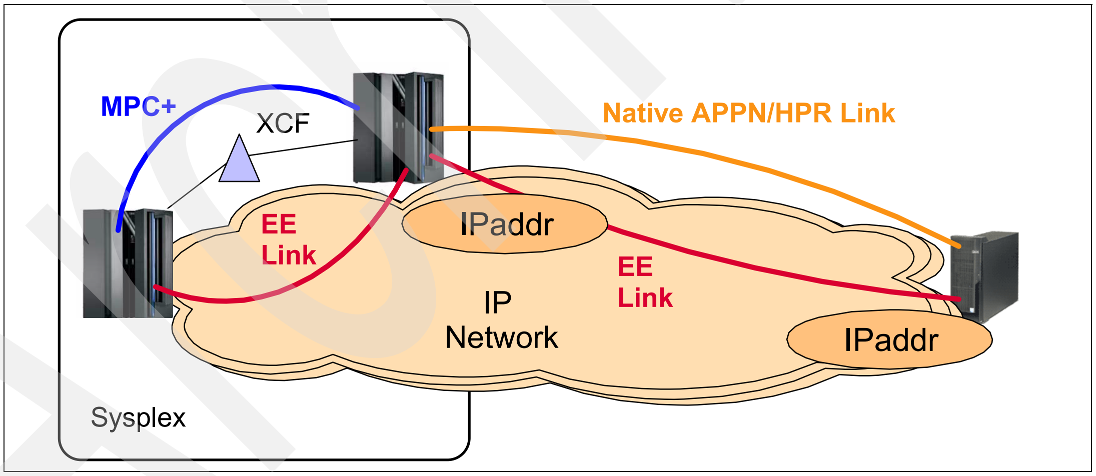


Figure 1-1 To VTAM, an EE Link is just another type of DLC.

Note the following explanation for Figure 1-1:

- To the IP network, the SNA traffic is UDP datagrams that get routed without hardware or software changes to the IP network. To the SNA application, 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 network without changing either the SNA applications or the IP hardware.

This section discusses the following EE topics:

- "Positioning Enterprise Extender"
- "Description of Enterprise Extender"

- “Features of Enterprise Extender”
- “Availability of EE”

1.1.1 Positioning Enterprise Extender

Enterprise Extender is aimed at those installations who have decided to implement an IP network throughout the organization, yet require SNA-like consistency and predictability for critical applications. The objectives of EE are:

- To be configurable with minimum definition
- To operate with minimal changes to the typical IP network
- To provide highly reliable SNA connectivity over an IP network
- To support current SNA-exploiting functions such as those available in a sysplex environment: generic resources (GR), and multinode persistent sessions (MNPS)
- To prioritize SNA data traffic in relationship to TCP and UDP traffic in the same network according to business policy (COS and TOS).
- To provide better levels of service (response time and throughput) than are available from previous SNA-over-IP technologies such as DLSw and AnyNet
- To be a viable replacement of SNA/SNI connections involving IBM 3745/NCP configurations.
- To be a replacement for AnyNet

1.1.2 Description of Enterprise Extender

Because Enterprise Extender’s main objective is to carry SNA traffic over an IP network without requiring changes to that network, it must treat the IP network as a particular type of SNA *logical* connection. The UDP protocol was chosen as the method of transporting network layer packets (NLPs). UDP packets provide a way to distinguish EE IP traffic from other types of IP traffic because they contain port numbers. The use of port numbers also permits a priority scheme to be implemented independent of the type of service bits, because many routers are able to prioritize traffic based on the received port number. UDP also has low overhead because it does not concern itself with error recovery or flow control. Recovery and flow control are the responsibility of the EE endpoints (partners). EE exploits the robust error recovery and congestion flow control algorithms of APPN’s HPR.

1.1.3 Features of Enterprise Extender

Enterprise Extender exploits the APPN/HPR architecture. Refer to 1.4.2, “Functions of APPN and HPR” on page 14, for details related to the APPN and HPR functions. The following features of EE are most critical to its success and performance:

- Supports APPN control flows using HPR
- Supports Connection Networks
- UDP port numbers mapped to transmission priority
- SNA COS mapped to IP TOS
- Maintains integrity of SNA data across the IP network
- Can use any IP interface supported by the TCP/IP stack
- Improves upon LLC2 error handling
- Is an effective replacement of SNA/SNI
- Is a replacement for AnyNet
- Supports Multinode Persistent Sessions (MNPS)
- Supports multiple VIPAs and multiple Connection Networks

- Supports IPv4 and IPv6 addressing models
- Compatible with IPSec and SNA session-level encryption (SLE)

1.1.4 Availability of EE

Enterprise Extender is available on the following platforms (this may not be an exhaustive list):

- Communications Server for z/OS
- Communications Server for AIX
- Communications Server for Linux
- Communications Server for Windows
- Communications Server for OS/2®
- Personal Communications
- i5/OS
- Microsoft® Host Integration Server
- Hewlett-Packard SNA Plus
- Data Connection SNAP/IX
- Cisco SNASw

Restriction: VTAM on z/VM® and z/VSE™ do not support Enterprise Extender.

1.2 What is APPN and APPN/HPR?

An Advanced Peer-to-Peer Networking (APPN) network comprises groups of connected type 2.1 nodes. The APPN architecture provides direct communications between any network-attached devices without the need for specific platforms or VTAM SSCP intervention.

This section discusses the following topics:

- "Positioning APPN and HPR"
- "Description of base APPN"
- "Features of base APPN"
- "Description of APPN/HPR"
- "Features of APPN/HPR"
- "Additional reading material for APPN and HPR"

1.2.1 Positioning APPN and HPR

HPR is referred to as APPN+ in the IBM networking blueprint. APPN+ is a powerful addition to Advanced Peer-to-Peer Networking (APPN) and a step toward the Networking Broadband Services (NBBS). NBBS is the IBM architecture for very high speed networking referred to as Gigabit APPN.

APPN is aimed at those installations who have decided to run applications in a sysplex environment. To assist EE in its objective to support current SNA-exploiting functions that exist in a sysplex environment, such as generic resources (GR) and multinode persistent sessions (MNPS), APPN enhances the management of these functions.

APPN/HPR improves the availability of these sysplex resources. Generic resources enables multiple application copies within a sysplex to present a single image to the user, resulting in better performance through load balancing and improved availability. The multinode persistent sessions function enables sessions to survive the failure of a VTAM, or a z/OS system, or even a processor, without disruption.

1.2.2 Description of base APPN

All APPN nodes implement a base set of functions according to their node type. Some functions are base for a network node or an end node and not applicable or optional for the other. Several base network node functions are optional for end nodes.

The content of the APPN base has changed over time and products have normally implemented those functions to comply with the architecture, especially where the change significantly improved function. However, products that complied with the level of architecture at the time of implementation may not implement functions that have later been added to the base set, especially when those products are functionally frozen. All new APPN implementations are expected to comply with the current level of the APPN architecture.

Inside APPN: The Essential Guide to the Next-Generation SNA, SG24-3669, has an appendix that identifies the APPN functions. The different functions are numbered for easy reference. That appendix lists base functions and option sets and is organized as follows:

- Base functions are described for Network Nodes (NNs) and End Nodes (ENs).
- Option sets are described for all node types.
- Base and option sets for various APPN link types are described.
- Dependencies among functions sets are discussed as related to NNs and ENs.

1.2.3 Features of base APPN

There are many features in the base APPN architecture; too many to list here. However, there are three important basic features of APPN that form a foundation for all other APPN functions. They are:

- "Base function Transmission Groups (TGs)"
- "Base function Network Addressable Unit (NAU)"
- "Base function addressing and session identifiers"

Base function Transmission Groups (TGs)

The connections between APPN nodes are called *transmission groups (TGs)*. The base APPN architecture supports *single-link* transmission groups only. APPN with HPR supports both *single-link* and *multilink* transmission groups (MLTGs).

Note: Do not confuse *multilink* transmission groups with *parallel* transmission groups.

A *multilink* TG consists of multiple DLC-level connections between two nodes made to appear to higher layers as a *single connection*. The essential purpose of this single-link-image is to have one link between the nodes that is better than the component links individually, typically in bandwidth and availability.

A parallel TG, on the other hand, comprises several links or several groups of links designed to appear to higher layers as *multiple connections* between the nodes. Their essential purpose is to augment the pool of possible routes to the endpoint.

Figure 1-2 shows how multi-link TGs differ from parallel TGs.

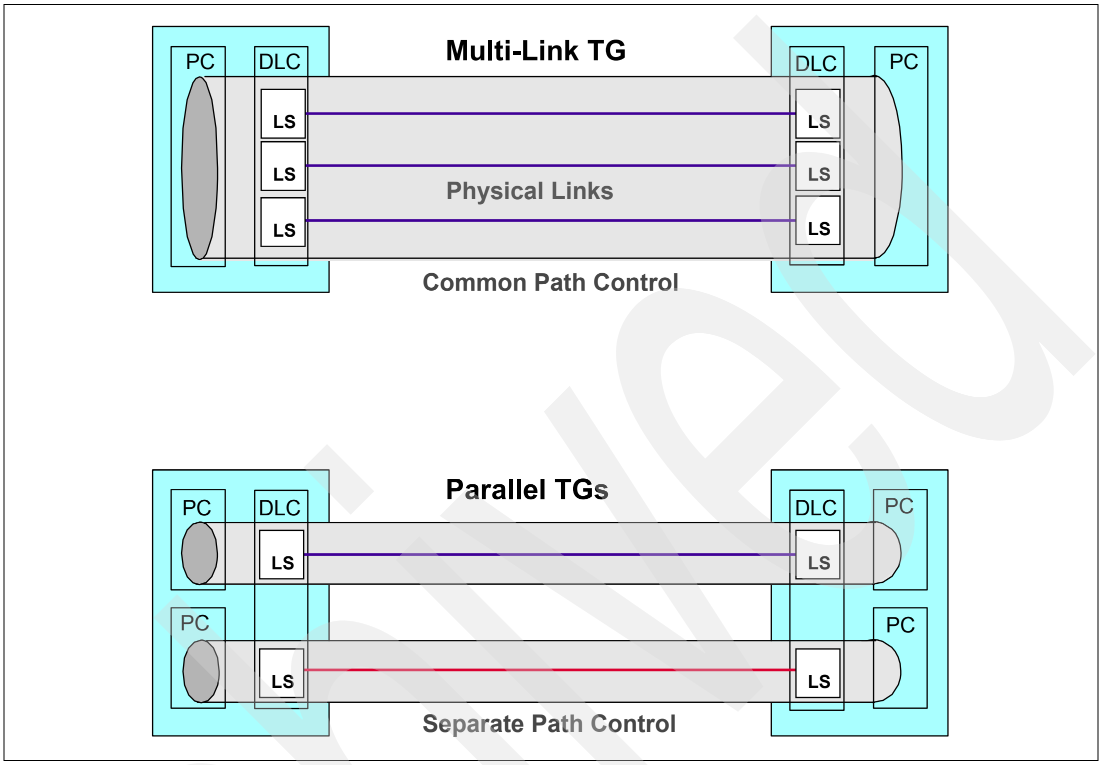


Figure 1-2 Multilink TG versus Parallel TGs

Base function Network Addressable Unit (NAU)

In an APPN environment, all components that can establish sessions with one another are called *network accessible units* (NAU). Examples are CPs and LUs. The term NAU was used in the subarea environment as an abbreviation for network addressable unit. The terminology changed with base APPN, in that NAUs are represented by *names* rather than by addresses.

Note: NAU names must be unique within an APPN environment. To ensure this uniqueness, a consistent naming convention is very important.

Base function addressing and session identifiers

Session identifiers play a very important role in session management and in routing within the APPN environment. Addressing in APPN is quite different from that in subarea, and addressing in HPR/IP is quite different from that in base APPN.

In traditional subarea SNA, each resource is assigned its own distinct network address. In an APPN environment, routing information is session oriented throughout. The address used in an APPN transmission header is an identifier unique on the given TG for a particular session, rather than the address of the NAU. The address used in an HPR/IP transmission header includes not only the HPR session identifier, but also the IP address of the participating endpoint.

1.2.4 Description of APPN/HPR

High Performance Routing (HPR) is an extension to the base APPN architecture. It can be implemented on an APPN network node or an APPN end node. HPR does not change the basic functions of the architecture, but provides enhancements to APPN. Further details on HPR can be found in the documentation listed in 1.2.6, "Additional reading material for APPN and HPR" on page 7.

HPR is a set of enhancements for base APPN whose main objectives are:

- To improve APPN data routing
- To improve APPN reliability and performance
- To provide compatibility with base APPN
- To enable easy migration to higher speed technologies (gigabit and beyond)

1.2.5 Features of APPN/HPR

The two important features of HPR are:

- Automatic Network Routing (ANR) on intermediate nodes

  ANR is designed to function on the *intermediate nodes* along the transport path, routing data quickly and efficiently from one link to another, without regard to session awareness.

- Rapid Transport Protocol (RTP) on endpoints

  RTP is implemented in the *endpoints* of a connection, providing error recovery from packet loss and support of control flows between endpoints. It does have session awareness.

Base APPN does not support the ANR and RTP functions. HPR includes the advanced capabilities to:

- Exchange HPR capabilities via XID exchange
- Exchange HPR-related topology information on CP-CP sessions
- Calculate HPR-only routes (Network Nodes only)
- Support nondisruptive path switch around undesirable or failed links
- Understand and process ANR labels
- Establish and maintain an RTP connection with a partner
- Recover from packet corruption or loss
- Resequence packets arriving out of order
- Perform adaptive rate-based (ARB) flow control
- Act as the boundary between base APPN and HPR
- Support control flows over RTP pipes

For details on these HPR capabilities, refer to 1.4.2, "Functions of APPN and HPR" on page 14.

1.2.6 Additional reading material for APPN and HPR

Regardless of the size of the network, with enough effort in the APPN/HPR planning stage, a smooth transition to APPN/HPR/IP can be achieved. Refer to subarea-to-APPN migration manuals to help in your understanding of base APPN and its migration to HPR.

The following documentation will help provide the necessary APPN background.

- *Subarea to APPN Migration: VTAM and APPN Implementation*, SG24-4656
- *Subarea to APPN Migration: HPR and DLUR Implementation*, SG24-5204
- *Inside APPN: The Essential Guide to the Next-Generation SNA*, SG24-3669
- *High Performance Routing (HPR) Early User Experiences*, SG24-4507

- *Dynamic Subarea and APPN Management using Netview* , SG24-4520
- *Managing Your APPN Environments Using NetView*, GG24-2559
- *z/OS Communications Server: New Function Summary*, GC31-8771

1.3 What is the importance of Enterprise Extender?

Enterprise Extender combines features of SNA and IP to offer the best of both environments when running SNA traffic over an IP network. The media method for EE is referred to as HPR/IP because EE requires APPN's HPR while using native IP.

This section discusses the following topics:

- "The role of EE within IP"
- "Benefits of using EE"

1.3.1 The role of EE within IP

Because of its design, the EE architecture is extremely flexible. It can be used in all networks from the smallest to the largest, and provides the network architect with a wide choice of locations within the network to place the SNA/IP boundary. EE exploits all the latest routing performance enhancements provided by APPN/HPR. See the detailed HPR discussion in "APPN with High Performance Routing (HPR)" on page 26.

1.3.2 Benefits of using EE

The advantages of using EE are in the areas of availability, performance, and usability.

The following benefits of using EE are discussed in this section:

- "Maintains integrity of SNA data across the IP network"
- "EE can use any IP interface supported by the TCP/IP stack"
- "Connection Network support"
- "Improvements upon LLC2 error handling"
- "EE is an effective replacement of SNA/SNI"
- "Failure protection better than native IP"
- "Failure recovery better than DLSw"
- "Supports multinode persistent sessions (MNPS)"
- "UDP port numbers mapped to transmission priority"
- "Class of Service priority support (COS)"
- "Flow and congestion control"
- "Flexibility and simplification"
- "Support for IPv4 and IPv6 addressing models"
- "Compatibility with IPSec and SNA session-level encryption (SLE)"

Maintains integrity of SNA data across the IP network

For HPR, the IP network appears to be a link. For that reason, the SNA session-level security functions (user authentication, LU authentication, session encryption, and so on) are still available for use. In addition, because HPR traffic flows as UDP datagrams through the IP network, IPsec can be used to provide network-layer security inside the IP network.

The following discussion is an excerpt from RFC 2353, *APPN/HPR in IP Networks*, and therefore, this copyright notice is printed here.

> The UDP Checksum is a 16-bit optional header field that provides coverage of the UDP header and the user data; it also provides coverage of a pseudo-header that contains the source and destination IP addresses. The UDP checksum is used to guarantee that the data has arrived intact at the intended receiver. When the UDP checksum is set to zero, it indicates that the checksum was not calculated and should not be checked by the receiver. Therefore, Enterprise Extender requires the use of UDP checksum with the native IP DLC to assist with data integrity, a function required by HPR.

When an HPR/IP connection includes the firewall Network Address Translation (NAT) function, and IPSec is also being used, consideration must be given to when and where the IPSec process takes place. Because the checksum field covers a pseudo header including the source and destination IP addresses, checksum must be updated whenever these address fields are modified. If IPSec is being used to encrypt the packets, the UDP header is included in that encryption. If encryption occurs *before* NATing, the NAT process does not recognize any valid UDP headers, and NAT fails to work properly. If encryption occurs *after* NATing, the NAT process works as expected. See RFC3022, *Traditional NAT*, for details.

Typically, access to UDP is provided by a sockets API. UDP provides an unreliable connectionless delivery service using IP to transport messages between nodes. UDP has the ability to distinguish among multiple destinations within a given node, and utilizes port-number based prioritization in the IP network. UDP provides detection of corrupted packets, a function required by HPR. Higher-layer protocols such as HPR are responsible for handling problems of message loss, duplication, delay, out-of-order delivery, and loss of connectivity. UDP is adequate because HPR uses RTP to provide end-to-end error recovery and in-order delivery; in addition, LDLC detects loss of connectivity.

The Transmission Control Protocol (TCP) was not chosen for the native IP DLC because the additional services provided by TCP such as error recovery are not needed with HPR's equivalent RTP function. Furthermore, the termination of TCP connections would require additional node resources (control blocks, buffers, timers, and retransmit queues) and would, thereby, reduce the scalability of the design.

EE can use any IP interface supported by the TCP/IP stack

All major APPN platforms have either implemented or made a statement of intent to implement HPR/IP routing. HPR/IP provides a base for high-speed networking technologies, hence migrating the network to HPR/IP will help position it for future increases in bandwidth.

Because EE can use any z/OS-supported IP network interface, SNA traffic can exploit the latest in high-bandwidth technologies, including the OSA-Express Gigabit Ethernet and 1000BASE-T features, OSA-Express2 1000BASE-T, Gigabit, and 10 Gigabit Ethernet features, as well as HiperSockets™.

Queued Direct Input/Output (QDIO) is currently one of the most efficient modes of operation for data transfer into the z/OS system. Certain OSA interfaces support the QDIO mode only. Even though EE is supported by *any* IP interface that the TCP/IP stack supports, these QDIO interfaces represent the *preferred* interface type for EE's HPR/IP transport mechanism because of QDIO's efficiency.

Connection Network support

A Connection Network can be defined on the IP network, which uses logical EE links. Making use of this logical Connection Network avoids defining all such logical links between each pair of a large number of IP addresses. The dynamics of APPN are an advantage in supporting a Connection Network for EE. For more details, see topic "APPN connection networks" on page 38.

Improvements upon LLC2 error handling

The underlying EE transport network appears as an APPN TG but uses logical data link control (LDLC) to exchange XIDs and network layer packets (NLPs). LDLC is a subset of LLC2 that eliminates much of the error handling and acknowledging that HPR's Rapid Transport Protocol (RTP) makes unnecessary at link level. There are some major differences that LDLC has to accommodate, such as:

- LLC2 requires the use of several fixed timers, which are by their nature incompatible with a variable-route variable-delay IP network.
- LLC2 performs error recovery, which is not necessary in HPR.
- LLC2 requires in-order delivery, which cannot be guaranteed on an IP network.
- LDLC includes only the XID, TEST, DISC, DM and UI frame types. These are sufficient to establish the connection (XID), send data (UI), terminate the connection (DISC) and respond in the negative to a previous frame (DM). The TEST frames are used to check whether a connection is still active, a function required by HPR over IP.

EE is an effective replacement of SNA/SNI

When used in conjunction with an extended border node (EBN), EE provides a mechanism to replace SNA Network Interconnection (SNI) functionality in a way that does not require SNA former or existing hardware, such as the 3745. APPN's Extended Border Node capability is exploited by EE to accommodate connectivity to non-native NETIDs. 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 the other partner company.

Failure protection better than native IP

TCP/IP has always had the ability to reroute packets around failing components, without disrupting the connection, by means of its *connectionless* IP transport. More recently SNA has implemented a similar function in APPN. The HPR extension to APPN is *connection-oriented* as SNA has always been, but when it detects a failure it will move an existing connection around a failing component.

Failure recovery better than DLSw

HPR/IP used by EE provides nondisruptive rerouting using either IP or HPR methods as necessary. By contrast, DLSw provides nondisruptive rerouting across the IP network alone, and is susceptible to failures of the DLSw routers at the edge of the network. Because DLSw does not support HPR, it cannot provide full end-to-end nondisruptive rerouting in a combined SNA/IP network.

Supports multinode persistent sessions (MNPS)

Enterprise Extender is particularly suited for installations wishing to exploit the multinode persistent sessions (MNPS) function in a sysplex environment. MNPS enables an application to move between MVS™ images in a sysplex after a failure, without disrupting existing sessions. Because MNPS utilizes HPR to achieve this, EE is a perfect complement to MNPS even in the situation where there is no SNA network. The only requirement is a remote partner node capable of EE and, therefore, HPR.

UDP port numbers mapped to transmission priority

The UDP port number identifies the destination of the datagram as being the partner IP host's ANR routing function. Five UDP ports (12000-12004) have been registered with the Internet Assigned Number Authority (IANA) for this purpose. Each of these default ports is mapped to one of the APPN transmission priority values, with 12000 being used for XID exchange. An EE implementation may choose to alter these port numbers, but by using the registered defaults you can be reasonably sure that no other application will conflict with EE. ANR labels are mapped to the partner's IP address.

Important: Many firewalls are configured by default to block UDP datagrams. When using EE between distinct enterprises, make sure that intermediate firewalls along the path do not block the five UDP ports 12000 - 12004.

Class of Service priority support (COS)

One of the biggest issues facing those who want to transport SNA over an IP network is the requirement of maintaining SNA's class of service (COS). In native SNA the COS specified for a particular session is used to determine both the route taken by the session and the transmission priority allotted to it.

The performance of packet delivery in an IP network is essentially unpredictable because of IP's connectionless transport. However, IP provides for a transmission priority using the precedence bits in the IP header. Even though many routers now support the use of these bits, they have traditionally used the TCP or UDP port number as a means of assigning priorities to packets.

EE supports the use of *both* the precedence bits *and* the port numbers to inform the IP network of the transmission priority. The SNA transmission priority is mapped to the UDP port number, which is why five UDP ports have been registered for EE use. Many IP routers can be configured to prioritize traffic based on the port number. However, the EE architecture provides for the use of the precedence bits in the IP header for the same purpose. LLC commands (XID, TEST, DISC, DM) use the same precedence bit setting (the highest) as network priority NLPs.

Note: Use of the precedence bits provides more consistency in accommodating the desired priority than the use of port numbers. The port numbers are carried inside the UDP datagram, whereas the precedence bits are in the IP header. Thus, *encrypted* packets have unreadable port numbers and *fragmented* packets have no port numbers after the first fragment; therefore, intermediate routers cannot determine the port-number-based priority in these cases.

Flow and congestion control

Because there is no link-level error recovery and no guarantee that packets will arrive in order on an IP connection, EE transports only HPR network layer packets (NLPs) once the XID flows are completed. Because both partner nodes are *endpoints*, they must support control flows over HPR's Rapid Transport Protocol (RTP). By placing the responsibility of error recovery and data integrity at the endpoints, the intermediate nodes can be off-loaded with that responsibility and be allowed to focus on highly efficient routing techniques. This function, *control flows over RTP*, is a critical feature of EE.

TCP/IP and HPR both provide their own unique, network-specific mechanisms for flow and congestion control. TCP uses a windowed technique, whereas HPR uses a formula based on data rate. EE introduced a new variant of the HPR flow control method known as responsive mode adaptive rate-based (ARB) flow control. Responsive mode ARB, like basic mode ARB, is designed to prevent network congestion; however, it also ensures a fair division of network capacity between the four SNA priorities and the native IP traffic.

Nodes that support Responsive Mode ARB can negotiate their level of ARB support during route setup exchange, and fall back to the original ARB if their partner does not support Responsive Mode. Responsive Mode ARB provides the following features:

- It competes fairly with TCP congestion control.
- Reservation of bandwidth for SNA traffic is no longer necessary (as it often is with DLSw).
- It can be tuned to tolerate a certain level of lost data.
- It gives priority to short transmissions.
- It allocates fair bandwidth to sustained transmissions, independent of propagation delays.
- It can ramp up its transmission rate faster at startup.

Flexibility and simplification

Enterprise Extender technology can reduce the demands on the data center routing platforms such as routers and front-end processors and thus provide a more cost-effective solution than other integration technologies. The boundary nodes between the SNA and IP networks have less work to do than, for example, DLSw routers that must maintain TCP connections with their attendant flow control and error recovery requirements. EE does all that at the HPR endpoints only, wherever they may be located.

There are two very important benefits of using EE related to a composite network containing a mix of SNA and IP:

- EE has been designed to run over existing IP networks without requiring any change to applications or to IP routers. SNA applications see the same network interface as before, whereas IP routers see the same IP (UDP) packets as before. Only the HPR nodes at the edges of the IP network need to be aware of EE.
- In conjunction with the branch extender technology (BrX), EE enables the implementation of extremely large networks that provide SNA application access using any combination of SNA and IP networks and clients.

Support for IPv4 and IPv6 addressing models

Optionally, the EE XCA’s 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.

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, provided that an IPv4 and an IPv6 local VIPA address have been defined to the stack.

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.

Compatibility with IPSec and SNA session-level encryption (SLE)

The most significant difference between IPSec and SLE is that IPSec encrypts part of the UDP header, including the UDP port numbers, but SNA session-level encryption does not.

Refer to *z/OS Communications Server SNA Resource Definition Reference*, SC31-8778, for specifics about session-level 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.

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.

1.4 Why is APPN/HPR required by Enterprise Extender?

High Performance Routing uses one of the most robust routing algorithms available today that incorporates both dynamic path switching around failed links and end-to-end data prioritization.

One of the main reasons for networks to migrate to APPN is the implementation of a sysplex. A sysplex environment provides some unique advantages to the VTAM installation, namely Generic Resources (GR) and Multinode Persistent Sessions (MNPS). With the latest emphasis on IP application load balancing within the sysplex through the use of Sysplex Distributor, and SNA application load balancing within the sysplex through the use of generic resources, migrating to an HPR/IP (EE) infrastructure has become a natural step toward supporting SNA applications through native IP connectivity to the mainframe. HPR/IP accommodates the presence of SNA data within the IP packet, and exploits the dynamic routing capabilities of both HPR and IP.

We discuss the relationships between APPN/HPR and HRP/IP in the following sections:

- “Benefits of APPN and HPR”
- “Functions of APPN and HPR”
- “Functions of HPR/IP (EE)”

1.4.1 Benefits of APPN and HPR

As its name implies, Advanced *Peer-to-Peer* Networking (APPN) architecture reverses the *hierarchical* nature of subarea SNA. Consider the following benefits of APPN and HPR:

- “APPN offers advantages over subarea SNA”
- “HPR offers advantages over base APPN”
- “HPR/IP offers advantages over native IP”
- “Dynamic definitions”

APPN offers advantages over subarea SNA

By using a *peer-to-peer* approach, APPN offers these advantages over subarea SNA:

- Better performance during session initiation: APPN uses (in most cases) fewer line flows per LU-LU session during initiation.
- Improved performance during network activation: APPN can eliminate control sessions, such as SSCP-PU and SSCP-LU, thereby eliminating many control flows during network activation.
- Reduced system definitions: APPN does not use PATH decks as it learns about network topology dynamically.
- Increased availability: As the topology is learned dynamically, there is no need to shut down parts of the network in order to add a single node.

HPR offers advantages over base APPN

By using *improved routing* techniques, HPR offers these advantages over base APPN:

- Non-disruptive rerouting: Rapid Transport Protocol (RTP) with its path switching capability, provides a higher availability in the network, bringing the benefits of connectionless network rerouting to the benefits of connection-oriented SNA networks.
- Reduced storage and loading: Intermediate nodes using Automatic Network Routing (ANR) need fewer CPU cycles to route HPR frames, and APPN session control blocks are not needed, reducing significantly the requirements in large networks.
- Staged migration: With defaults in most products to use HPR and keywords to allow controlled use, HPR can be implemented simply across the network as desired.
- Future positioning: HPR capabilities position you for the emerging high-speed technologies which rely on the new routing techniques in HPR.

HPR/IP offers advantages over native IP

The use of HPR transport over an IP network provides *four* benefits, only the *first one* being available to traffic using native IP:

- Non-disruptive rerouting upon a failure can be accomplished using either IP or HPR methods, depending on the location of the failure.
- HPR retains the connection-oriented nature of SNA, making it more predictable and easier to manage. This is of particular importance when planning and managing bandwidth in the network.
- The use of HPR makes it possible for the above benefits to be realized in a hybrid network that uses both SNA and IP routing in different portions. EE provides seamless integration, because the IP network is seen as an additional hop in an SNA connection path.
- Adaptive rate-based (ARB) flow control provides a fair allocation of bandwidth among SNA sessions and IP traffic while maintaining the SNA class of service.

For a detailed discussion of High Performance Routing, Automatic Network Routing, and Rapid Transport Protocol as it relates to EE see the topic “APPN with High Performance Routing over IP” on page 30.

Dynamic definitions

Another major advantage of APPN is its dynamic nature. Systems programmers can eliminate a large part of predefined resources and allow them to be built dynamically. An effort is required to reduce the large number of definitions previously required by subarea, but this can be reduced gradually.

1.4.2 Functions of APPN and HPR

This section discusses the following APPN functions, setting the stage for Enterprise Extender:

- “APPN nodes and their roles”
- “APPN with Intermediate Session Routing”
- “APPN with High Performance Routing (HPR)”
- “APPN with High Performance Routing over IP”
- “Dependent LU Requester and Server (DLUR/DLUS)”
- “APPN connection networks”

APPN nodes and their roles

APPN includes the functions and node types defined by the following topics. Not all of the node types listed here are used in the scenarios in this publication. However, they are defined in this section to provide a thorough discussion on APPN's functionality:

- "Communications Management Configuration (CMC)"
- "Low Entry Network Node (LEN)"
- "End Node (EN)"
- "Network Node (NN)"
- "Network Node Server (NNS)"
- "Composite Network Node (CNN)"
- "Interchange Node (ICN)"
- "Migration Data Host (MDH)"
- "Central Directory Server (CDS)"
- "Border Node (BN)"
- "Peripheral Border Node (PBN)"
- "Extended Border Node (EBN)"
- "Branch Extender Node (BrX, BeX, BrNN)"
- "Virtual Routing Node (VRN)"

Communications Management Configuration (CMC)

A CMC is a subarea VTAM that owns NCPs and has session establishment and error recovery responsibility for one or more data hosts. A data host is a subarea VTAM that owns application LUs, but has no responsibility for LUs in the network. A data host does not own or activate any NCPs. It may own local LUs.

When migrating to APPN, consider the following:

- A CMC should be converted into an interchange node (ICN).
- A data host can be converted into a migration data host (MDH) or a pure APPN End Node.

Low Entry Network Node (LEN)

A LEN node has the lowest level of function within the APPN environment. It provides the minimum functions required to connect and establish a session over which data can be transported. The relation between LEN nodes is purely peer-to-peer, either side may activate the connection or start a session to its partner. A significant feature of the LEN architecture is that there are only *two* adjacent nodes involved in a LEN connection. No matter how many nodes there may be in the network, a LEN node is only aware of the adjacent node(s) to which it is connected. This type of basic LEN connection is shown in Figure 1-3.

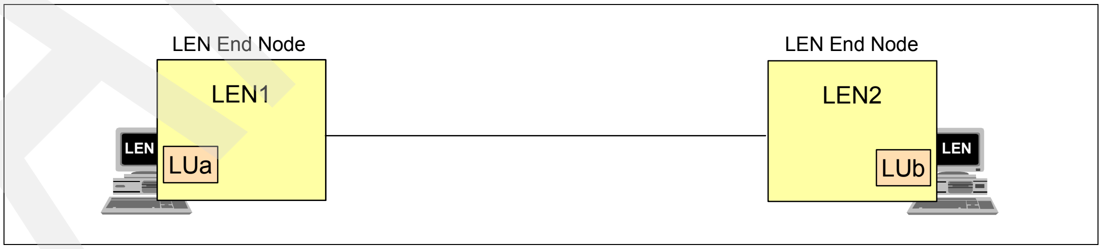


Figure 1-3 Direct LEN connection

Because LEN nodes cannot have CP-CP sessions, partner LUs and locations need to be defined in both the LEN node and the APPN node to which it is attached. The functions of LEN nodes are very limited. They are not able to exchange topology or configuration data. A LEN can depend on an intermediate node to assist in making a connection to a partner LEN. Each LEN views the intermediate node as the owner of the LU resources in the *other* LEN. By the LEN treating the intermediate node as the owner, the intermediate node must know where any requested resources are and how to get to the partner LEN. An intermediate node between two LEN nodes is depicted in Figure 1-4.

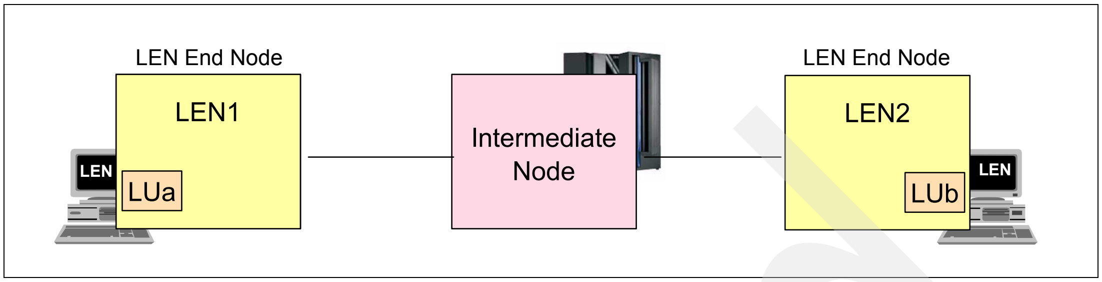


Figure 1-4 LEN connection through an intermediate node

Additional function is needed to reduce the number of definitions and maintenance effort when building larger networks. For this purpose, the APPN architecture was developed as an extension to SNA. APPN architecture defines two basic node types, the APPN End Node (EN) and the APPN Network Node (NN).

End Node (EN)

The APPN EN is similar to a LEN node, except that the *control point* (CP) of the end node exchanges information with the CP in the *adjacent* Network Node. The APPN nodes communicate using CP-CP sessions between adjacent nodes. The communication over this CP-CP session reduces the requirement for network definitions, and makes installation and maintenance of the network easier and less complex.

Figure 1-5 shows a couple of End Nodes connected through a common intermediate Network Node.

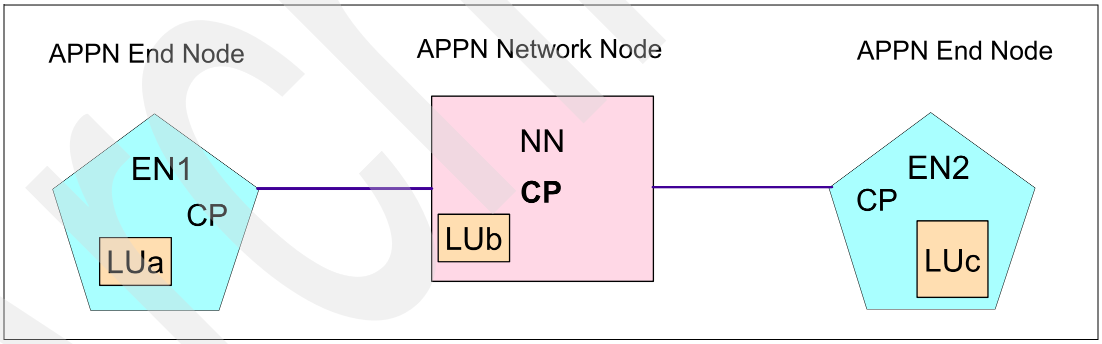


Figure 1-5 End Nodes connected through a common intermediate Network Node

An EN provides limited directory and routing services for its local LUs. It can select an adjacent NN and request this NN to be its network node server (NNS). If approved by the NN, the EN may register its local resources with the NNS. This enables the NNS to intercept Locate search requests for the EN's resources and pass these requests to the EN for verification

Network Node (NN)

The APPN NN has intermediate routing functions and provides network services to either APPN or LEN nodes that are attached to it. It establishes a CP-CP session with its adjacent APPN network nodes to exchange network topology and resource information. An APPN network node provides distributed directory and routing services for all LUs that it controls. These LUs may be located on the APPN network node itself or one of the adjacent LEN or APPN end nodes for which it is the network node server. Jointly, with the other active APPN network nodes, an APPN network node is able to locate all destination LUs known in the network. Figure 1-6 shows a number of Network Nodes managing LU sessions for a Low Entry Network node and a partner End Node.

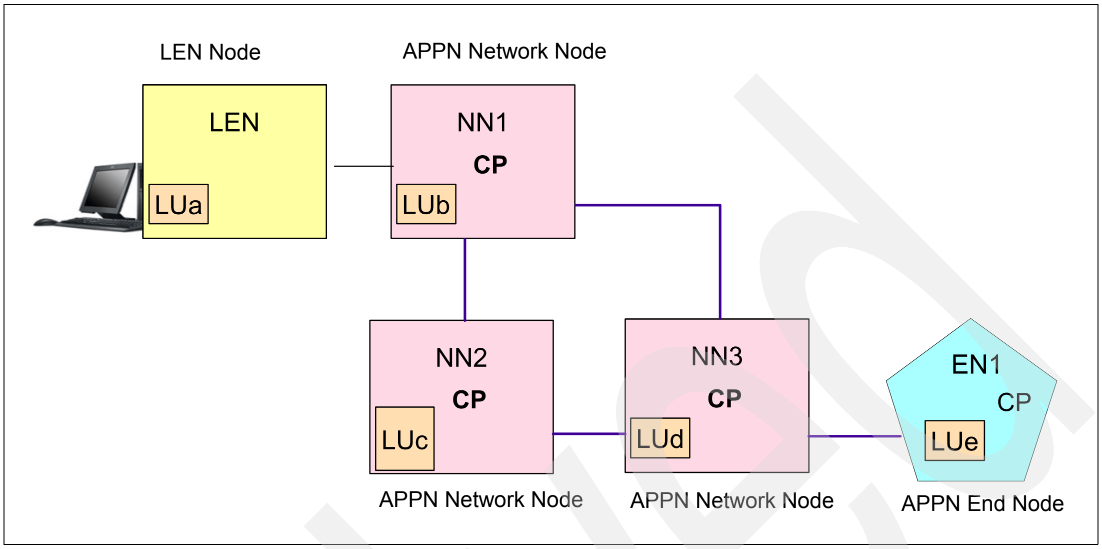


Figure 1-6 The path of an LU-LU session can traverse multiple intermediate nodes

After the LU is located, the APPN network node is able to calculate the route between origin and destination LU according to the required class of service. All network nodes exchange information about the topology of the network. When two adjacent network nodes establish a connection, they exchange information about the network topology as they know it. In turn, each network node broadcasts this network topology information to other network nodes with which it has CP-CP sessions. If the connection between two network nodes is deactivated, each network node broadcasts this change to all other active adjacent network nodes. Network nodes that are removed from service eventually are removed from the topology information along with their routing capabilities to any other nodes.

A CP-CP session between an APPN network node and an adjacent APPN end node is required only if the APPN end node is to receive network services, such as partner location, from this particular APPN network node. In this mode, the network node is said to be the network node server of the end node.

Network Node Server (NNS)

When an EN needs to establish a session to an LU whose location is unknown, it sends a Locate search request to its NNS. The NNS uses its distributed directory and routing facilities to locate the LU and calculates the optimal route to the destination LU from the EN. These facilities use various searches such as a directed search, a central directory search, or a broadcast search.

An APPN end node may have active connections to multiple adjacent network nodes. However, at any given time, only one of these network nodes can be acting as its network node server. The APPN end node establishes CP-CP sessions with a network node to select that network node as its network node server. Figure 1-7 shows multiple NNs, but only one can be an NNS for a given EN at any given time.

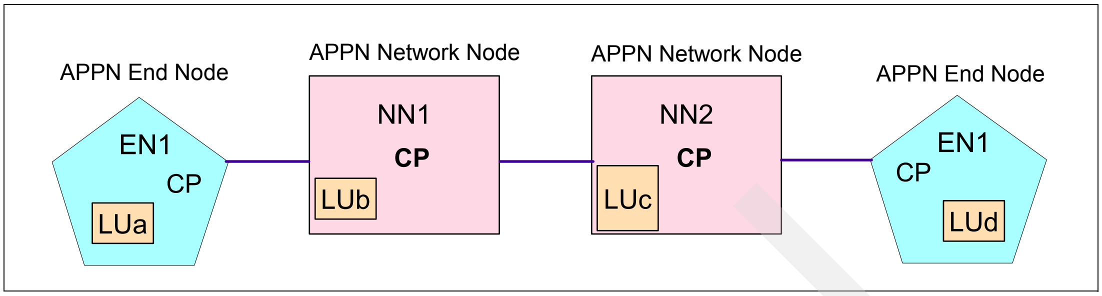


Figure 1-7 Network Nodes can be a Network Node Server for an End Node

On APPN network nodes, APPN end nodes are categorized as either authorized or unauthorized. An authorized APPN end node may send registration requests to register local network-accessible resources with the network node server and may additionally request that these resources be registered with the central directory server. If during session establishment a network node server does not know where an LU is located, it will query authorized APPN end nodes within its domain that have indicated they are willing to be queried for unknown resources. Network accessible resources on unauthorized nodes require explicit definition at the network node server, either statically as part of its system definition, or dynamically by the network node server's operator.

Composite Network Node (CNN)

The term *composite node* is used in some publications to represent a group of nodes that appear as one APPN or LEN node to other nodes in an APPN environment. A VTAM host with some NCPs together form a CNN. A CNN is itself a multi-node network, but when connected to an APPN node, it appears as one logical APPN or LEN node.

A subarea composite node may appear as either a LEN node or as an APPN network node. In the LEN case, the term composite LEN node is used; in the APPN case, the term *Low Entry Network* (LEN) is used. Figure 1-8 illustrates the CNN.

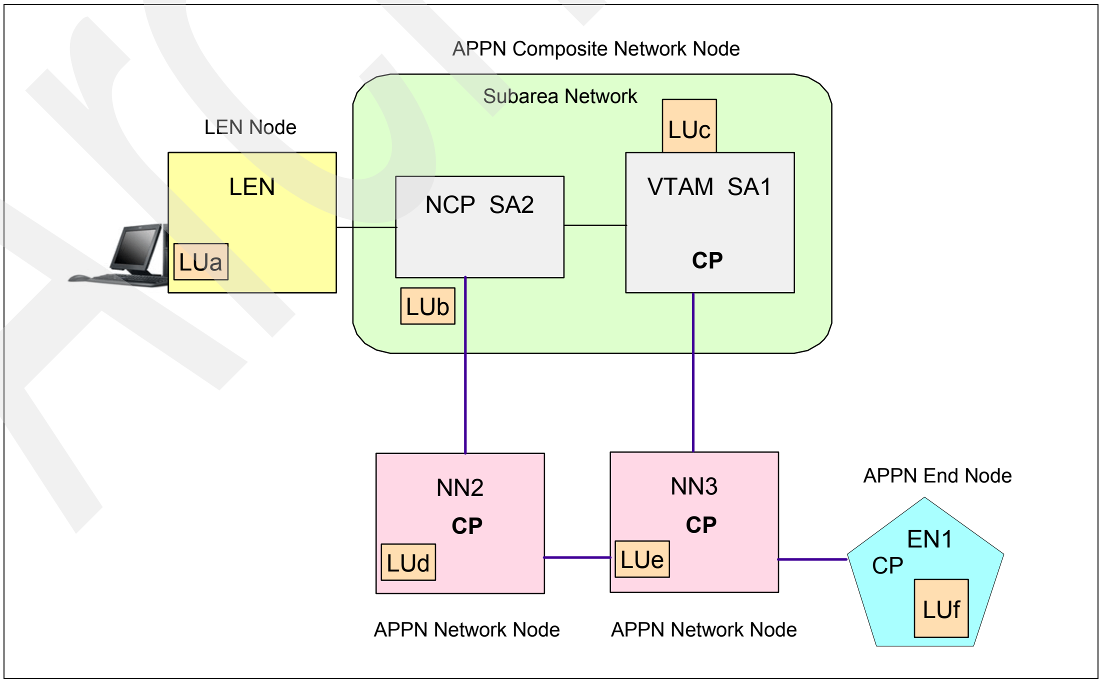


Figure 1-8 Composite Network Node appears as one logical node

Interchange Node (ICN)

A Network Node that supports both a subarea environment and an APPN environment is referred to as an ICN. An ICN is intended to replace the subarea CMC host. It may own NCPs and is the repository of all the functions provided by the CMC host. It provides ownership of dependent LUs, allowing these LUs to operate unchanged. A VTAM host acting as an ICN can be a stand-alone APPN VTAM network node or a composite network node, as described above. The ICN is always a network node and routes sessions from APPN nodes into and through the subarea network using subarea routing, without exposing the subarea implementation to the APPN part of the network. This is accomplished by making the APPN VTAM node, plus all of its owned resources, appear to other nodes as a single APPN network node with multiple connections. At the same time, the ICN, and the NCPs it owns, maintain their subarea appearance to other subarea nodes. An ICN is illustrated in Figure 1-9.

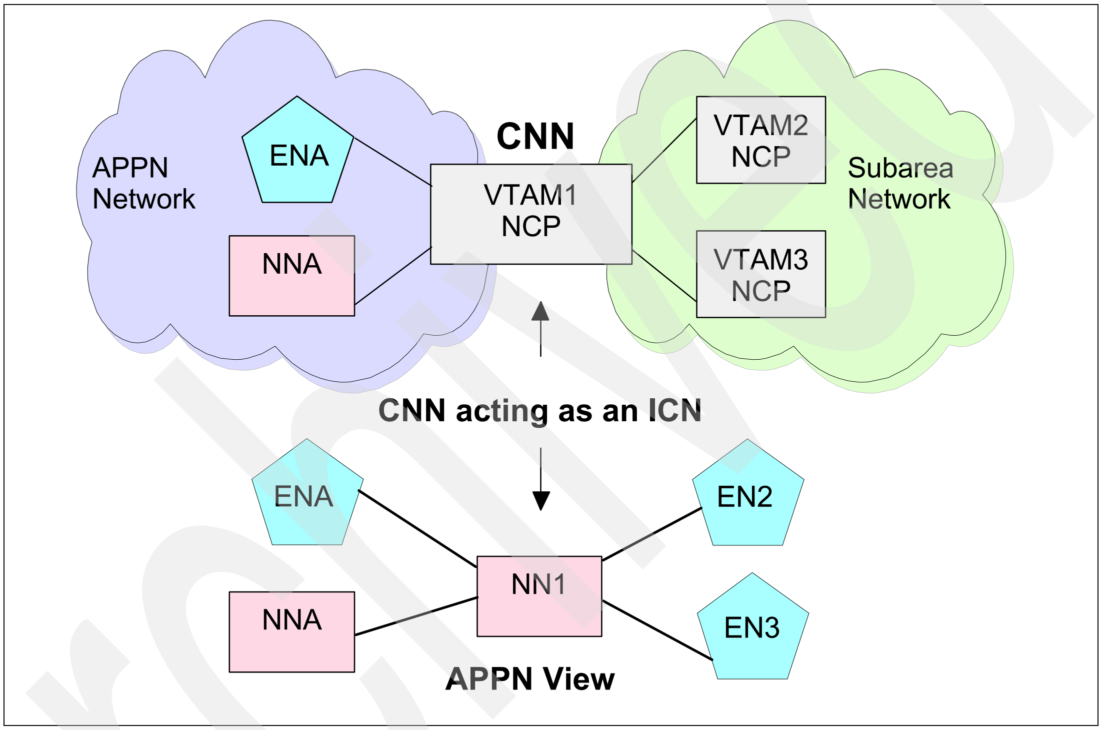


Figure 1-9 An Interchange Node is a CNN supporting both SNA subarea and APPN

The ICN has SSCP-SSCP sessions with other VTAM nodes as well as CP-CP sessions with adjacent APPN network nodes and end nodes. This support makes it possible for the ICN to use both APPN and subarea data flows to locate LUs and to provide the best route between nodes. APPN session setup protocols, which flow on CP-CP sessions, are converted to the corresponding subarea protocols that flow on SSCP-SSCP sessions.

Migration Data Host (MDH)

An End Node that supports both a subarea environment and an APPN environment is referred to as a MDH. The MDH is always an end node. Just as the ICN is an NN with subarea responsibilities, the MDH is an EN with subarea responsibilities. The NN and EN are considered to be *pure* APPN nodes, but the ICN and MDH are considered to be *hybrid* nodes because of their concurrent support of both subarea and APPN.

Central Directory Server (CDS)

The central resource registration facility provides a method for an APPN network node to register its resources at a CDS. Once a resource is registered, APPN network nodes can locate the resource by querying the CDS instead of using a broadcast search, thus improving network search performance during session establishment.

Depending on the size of the network, one or more CDSs are usually defined. The default registration value for VTAM applications is to register with the NN server and a CDS. Using a CDS reduces network broadcast searches, which is of utmost importance in an APPN environment. It is common for the NN to also be given the role of CDS.

Border Node (BN)

Base APPN architecture does not allow two adjacent APPN network nodes to connect and establish CP-CP sessions when they do not have the same NETID. The border node is an optional feature of an APPN network node that overcomes this restriction.

A border node can connect to an APPN network node with a different NETID, establish CP-CP sessions with it, and support session establishment between LUs in different NETID subnetworks. Topology information is not passed between the subnetworks. Hence, this provides the benefit of reducing the size of the APPN topology in each subnetwork. Similarly, a border node can also connect to another border node. There are two types of border nodes defined in APPN architecture: peripheral border node and extended border node.

Peripheral Border Node (PBN)

The peripheral border node accepts the connection of network nodes having different NETIDs and permits session establishment between LUs in different, *adjacent*, subnetworks. A PBN is always an NN. The peripheral border node is illustrated in Figure 1-10.

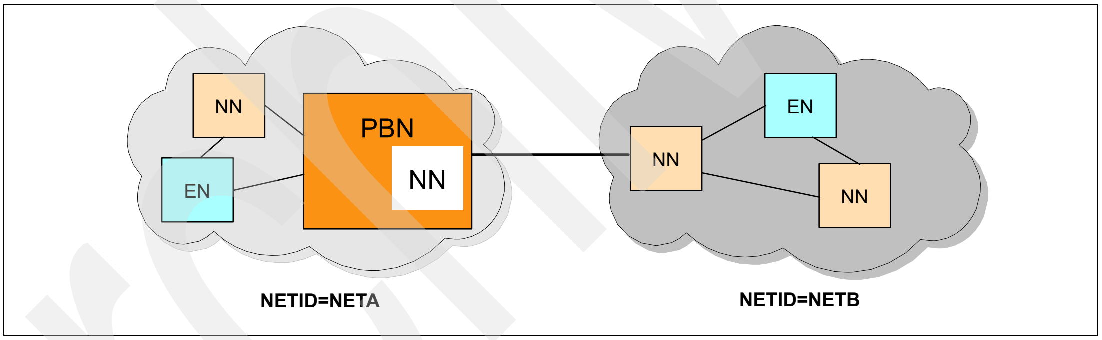


Figure 1-10 A Peripheral Border Node supports adjacent non-native NETIDs

A peripheral border node provides directory, session setup, and route selection services across the boundary between paired subnetworks with different NETIDs while isolating each subnetwork from the other network's topology information. This reduces the flow of topology updates and the storage requirements for the network topology database on network nodes in each of the network partitions.

Extended Border Node (EBN)

The EBN supports the connection of network nodes that have different NETIDs, and enables session establishment between LUs in different NETID subnetworks that *need not be adjacent*. An EBN is always an NN. The Extended Border Node is illustrated in Figure 1-11, where LUs in NETB can connect to LUs in NETX, and LUs in NETA can connect to LUs in NETY.

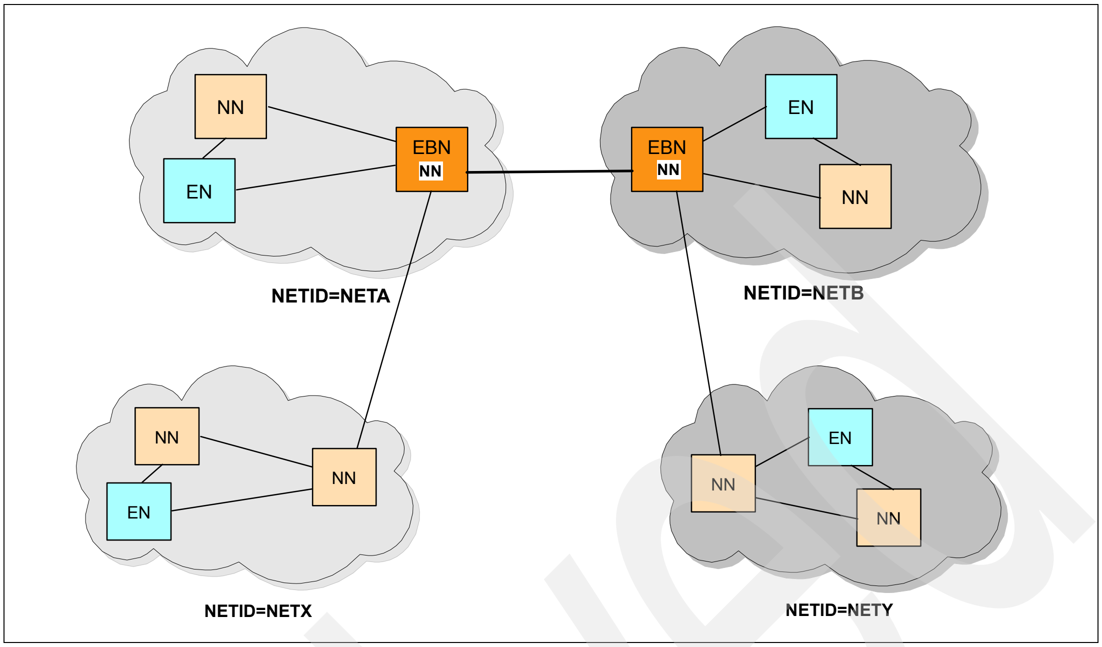


Figure 1-11 An Extended Border Node supports cascaded non-native NETIDs

An extended border node provides directory, session setup, and route selection services across the boundary between paired or cascaded nonnative NETID subnetworks. An extended border node can also partition a single NETID subnetwork into two or more clusters or topology subnetworks with the same NETID, thus isolating one from the topology of the other. Figure 1-12 shows a number of different node types performing various functions within multiple networks.

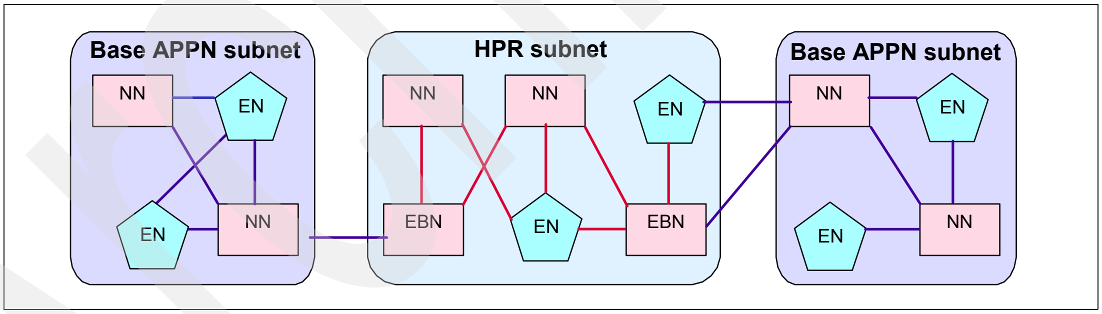


Figure 1-12 Multiple node types participating in APPN and APPN/HPR networks

Branch Extender Node (BrX, BeX, BrNN)

A branch extender network node (BrX) is a network node that acts as a network node to the APPN nodes downstream of it, but presents an end node image to the APPN nodes upstream of it. By appearing as an end node to the network, the number of network nodes in the network can be reduced. This, in turn, reduces the overall size of the topology and the frequency of topology updates. Figure 1-13 illustrates a Branch Extender Node.

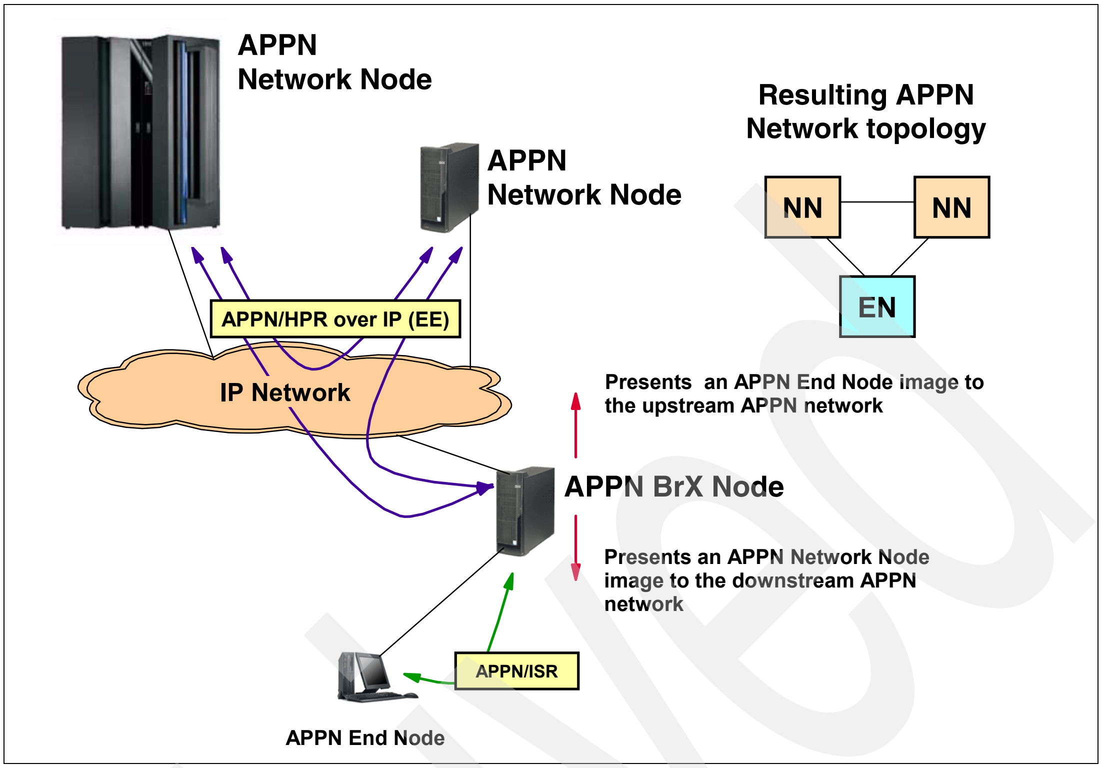


Figure 1-13 The Branch Extender node is two nodes in one, EN and NN

Virtual Routing Node (VRN)

The design for an APPN environment can reduce the addressing information stored at each node that is connected to a *shared-access transport facility* (SATF), such as a LAN environment, by having each participating node define a common *virtual routing node* (VRN), also referred to as a *virtual node* (VN). The virtual routing node represents this node's connection to the shared facility and all other nodes similarly configured. The SATF and the set of nodes having defined a connection to the common virtual routing node are said to comprise a *Connection Network* (CN). Figure 1-14 shows the logical concept of a connection network with its virtual routing node.

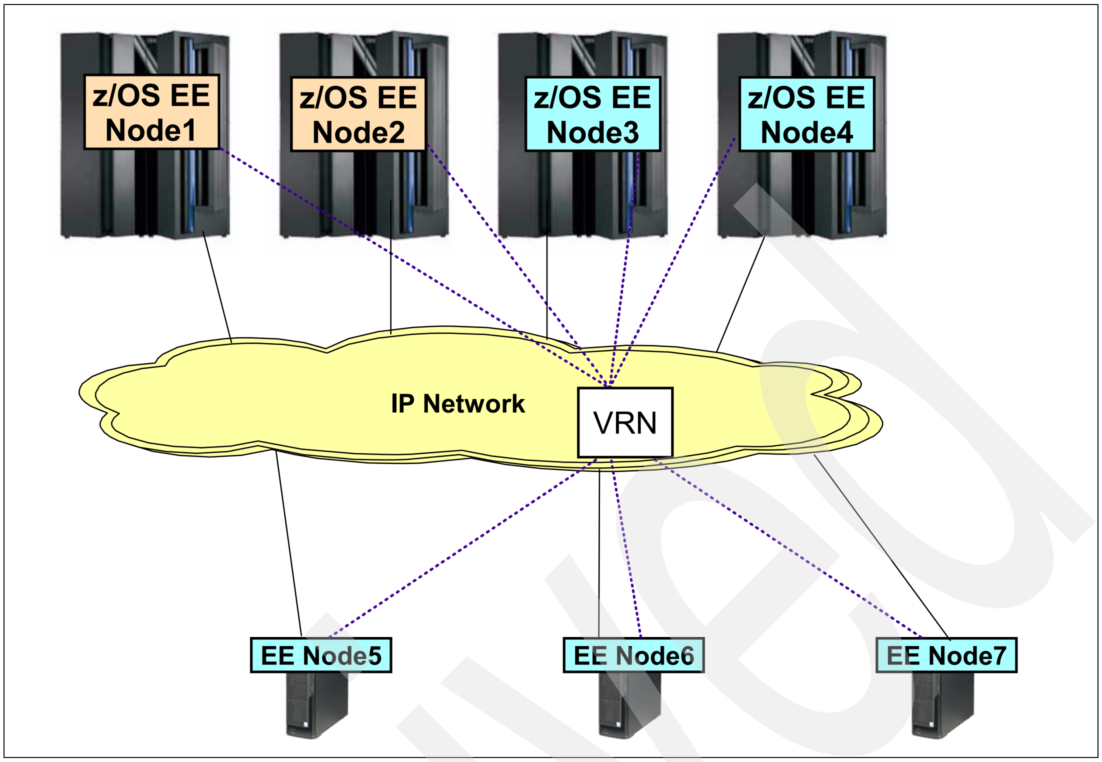


Figure 1-14 Logical concept of a Virtual Routing Node (VRN)

Technically, a VRN is not really a node, but rather a way of defining an APPN node's attachment to an SATF. It reduces end node definition and requirements by relying on the network node server to *discover* the common connection and supply necessary link-level signaling information as part of the regular Locate search process. This discovery process occurs when the End Nodes register themselves to their Network Node Server, supplying the identity of the connection network, the VRN name. LU-LU session data can then be routed directly, without intermediate node routing, between APPN nodes attached to the SATF. Figure 1-15 shows the CP-CP sessions between ENs and their NNS, and the direct LU sessions across the Connection Network using the VRN.

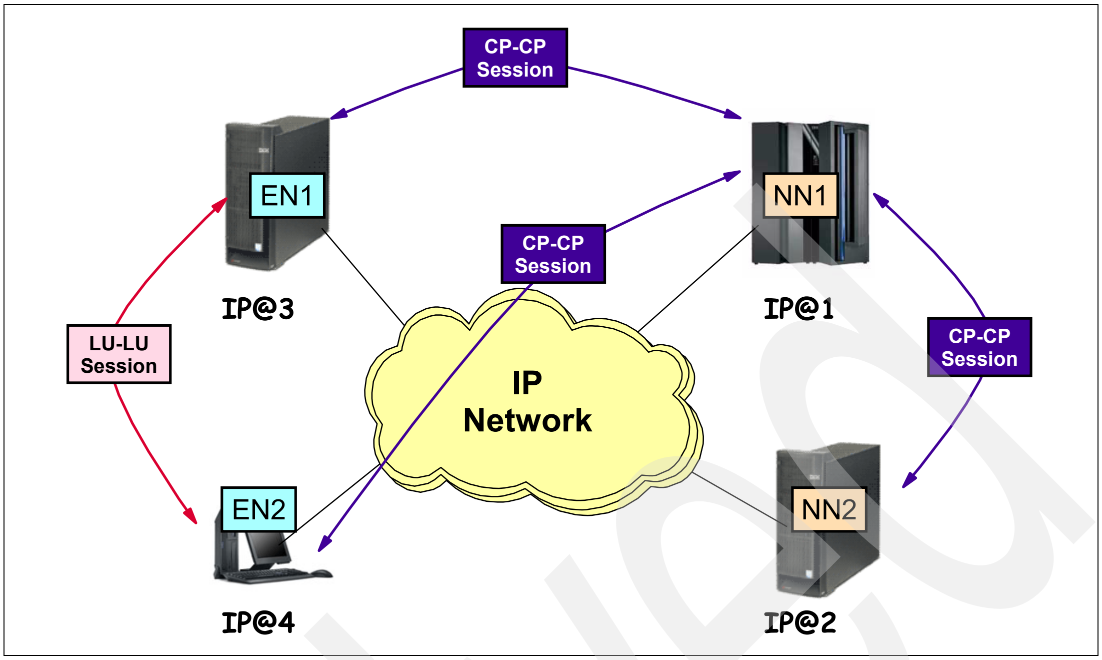


Figure 1-15 Direct EN-to-EN connectivity through the mechanism of the VRN

Figure 1-16 shows a network without a connection network (no VRN), where all EN LU-LU sessions flow through the NNS. The ENs define only the NNS, not each other. This approach avoids manual effort in managing all the definitions, but places a traffic management burden on the NN server, possibly introducing some performance issues.

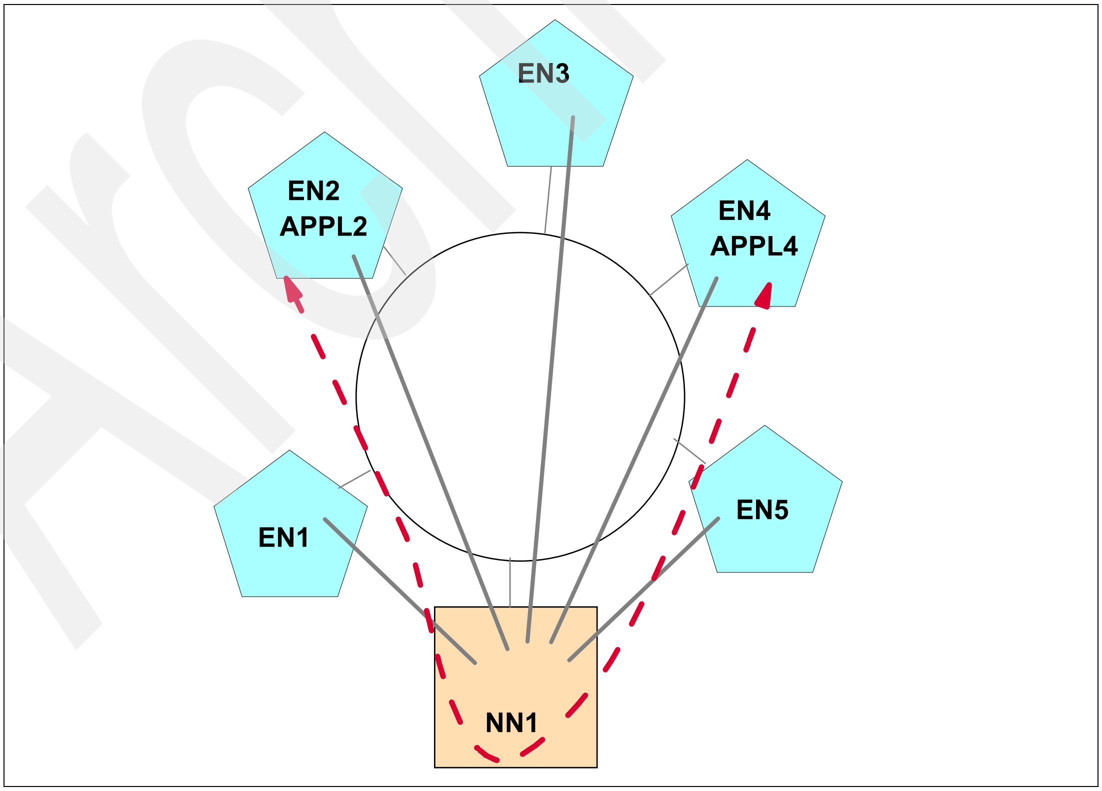


Figure 1-16 A SATF with no VRN (no connection network): traffic flows through the NNS

Figure 1-17 shows the same network (no VRN), but the ENs define direct connections between each other. This approach is clerically intensive, but yields the best performance and offloads the traffic burden from the NNS. When a new EN member is added to the SATF, the clerical effort intensifies even more: not only does the new member have to define all existing members, but all existing members have to add the definition of the new member. Scheduling and synchronizing these modifications may become difficult.

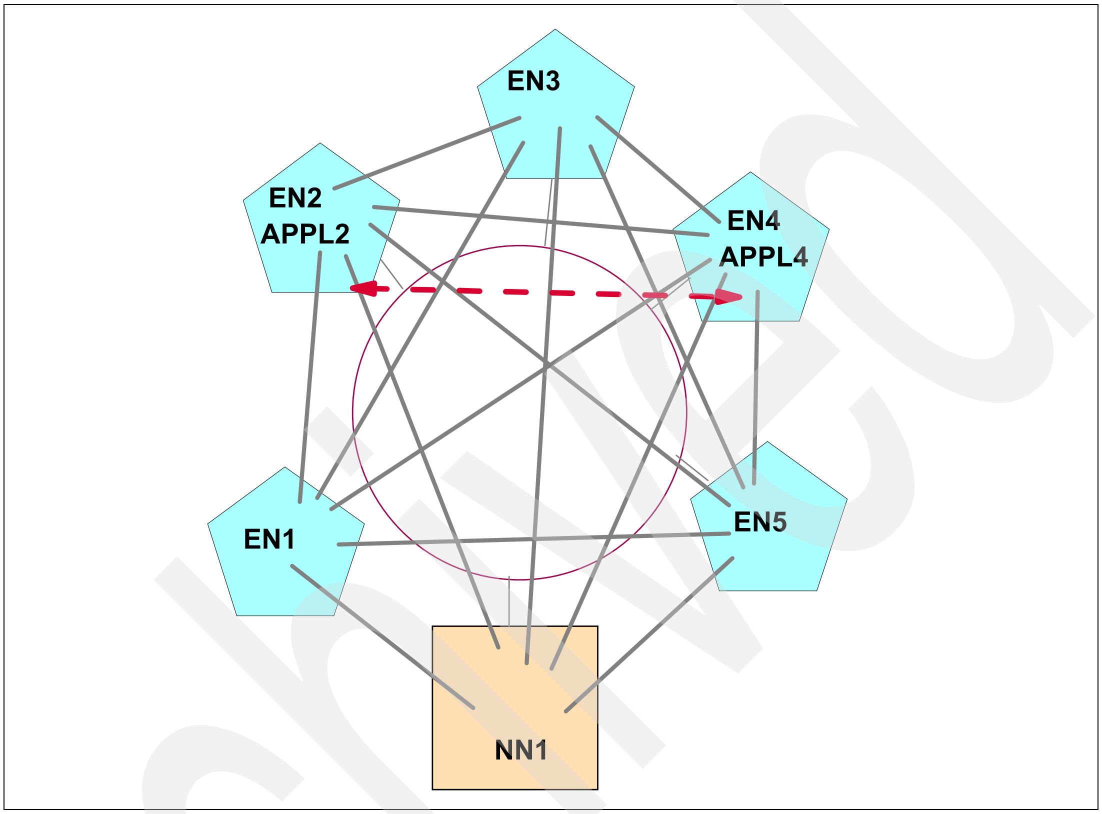


Figure 1-17 A SATF with no VRN (no connection network): all possible connections predefined

Figure 1-18 shows the network with a VRN network where each EN defines only its NNS and the VRN. Two possible routes between the partner ENs exist: the direct route through the VRN, and the indirect route through the NN Server. This approach lessens the clerical effort considerably and offloads traffic through the NN Server.

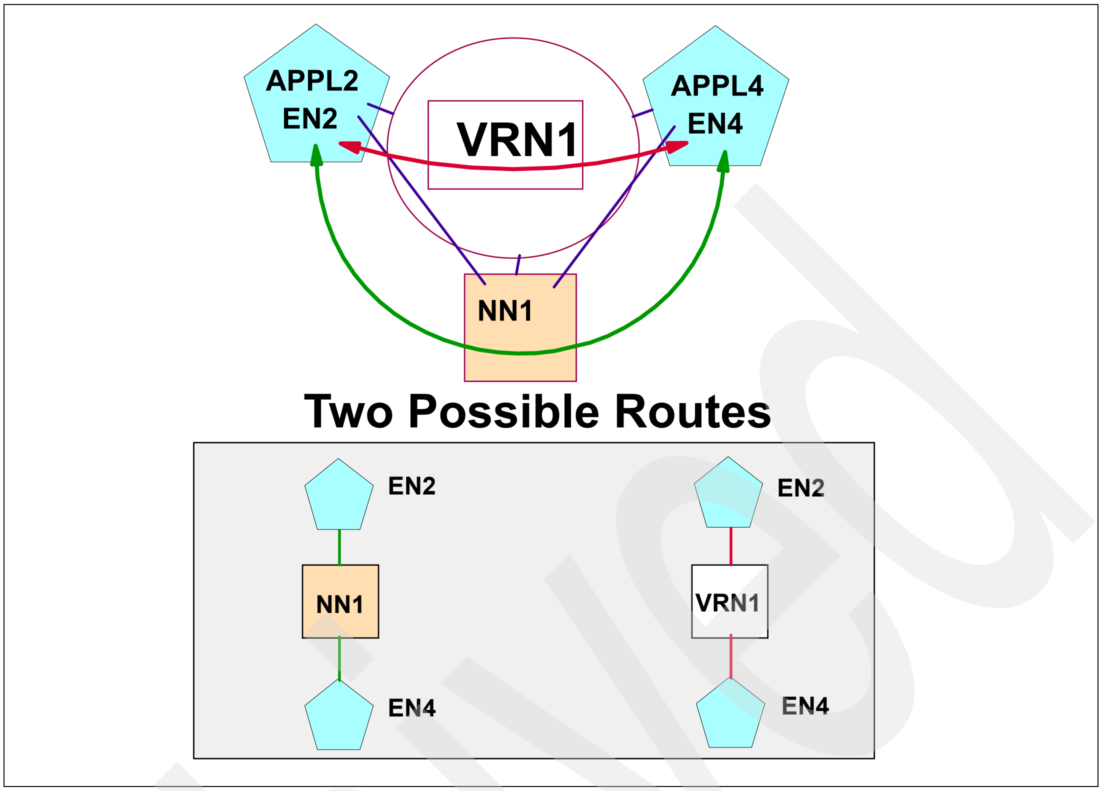


Figure 1-18 A SATF with a VRN (connection network): Always two possible routes between ENs

> **Note:** With EE, a connection network (CN) does not represent a physical network segment. Rather, the entire logical IP network itself is viewed as the SATF, because nodes can now communicate directly over this shared medium by using simple IP addressing.

APPN with Intermediate Session Routing

The APPN network node is capable of routing LU-LU sessions through itself from one adjacent node to another adjacent node. For example:

- In base APPN, this is done by *Intermediate Session Routing* (ISR).
- In HPR, this function is accomplished by *Automatic Network Routing* (ANR) which is:
  - Defined in topic “APPN with High Performance Routing (HPR)”
  - Illustrated in topic “APPN with High Performance Routing over IP”

APPN with High Performance Routing (HPR)

High-performance routing (HPR) is a set of enhancements for APPN whose main objectives are:

- Improved APPN data routing by performing error recovery at the endpoints only
- Improved APPN reliability by dynamic path switching around failed paths
- Compatibility with base APPN functions and use of the common topology
- Migration path to gigabit networks

HPR’s error recovery and control flows are implemented in the endpoints of a connection. The major part of this *endpoint* function is called *rapid transport protocol* (RTP). RTP provides a reliable end-to-end connection for sessions using HPR, and includes the following characteristics:

- *End-to-end error recovery* removes the need for intermediate routing nodes to detect, check, acknowledge and retransmit packets in error.
- *Adaptive rate-based* (ARB) flow control provides a congestion avoidance and control mechanism that removes from intermediate nodes the need to do adaptive pacing.
- The *nondisruptive path switch* function changes a session route without affecting the flow of data on the session.
- The APPN/HPR *boundary function* makes it possible for a network to include both base APPN and HPR portions, providing translation of the appropriate protocols at the boundaries.

The intermediate nodes in an HPR network have only one function, to route data quickly and efficiently from one link to another. This technique is known as *automatic network routing* (ANR). ANR is a low-level routing mechanism that minimizes processing cycles and storage requirements for routing packets through intermediate nodes. HPR uses RTP connections to transport LU-LU and (optionally) CP-CP session traffic. RTP provides reliability, in-order delivery, segmentation and reassembly, and a congestion and flow control algorithm. These functions seem similar to the roles of TCP and IP in the IP stack layers. Even though they are very different implementations, TCP can be compared to RTP, and IP can be compared to ANR, in generic routing concepts. They are the implementation of a common transport service, as compared in Figure 1-19.

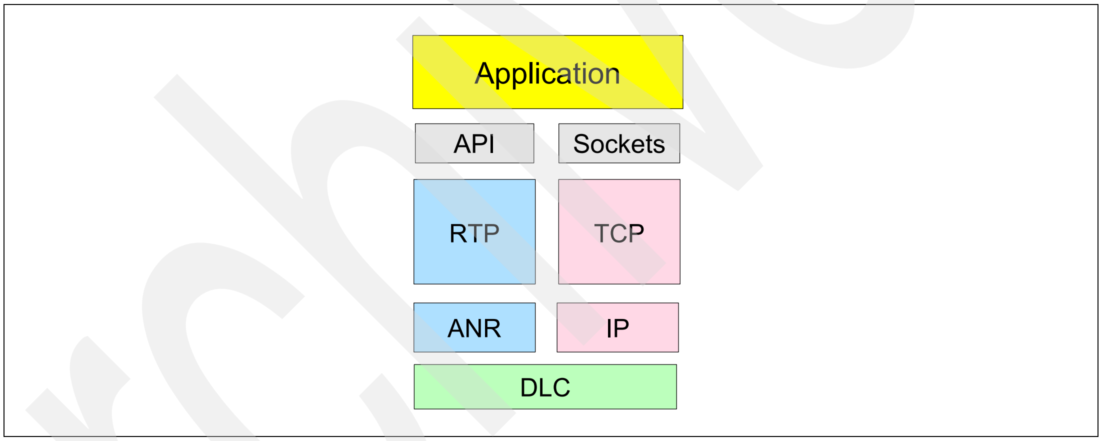


Figure 1-19 Similarities between TCP/IP and RTP/ANR

The general objective of RTP is to manage the connection between endpoints and provide the flow control mechanism as shown in Figure 1-20.

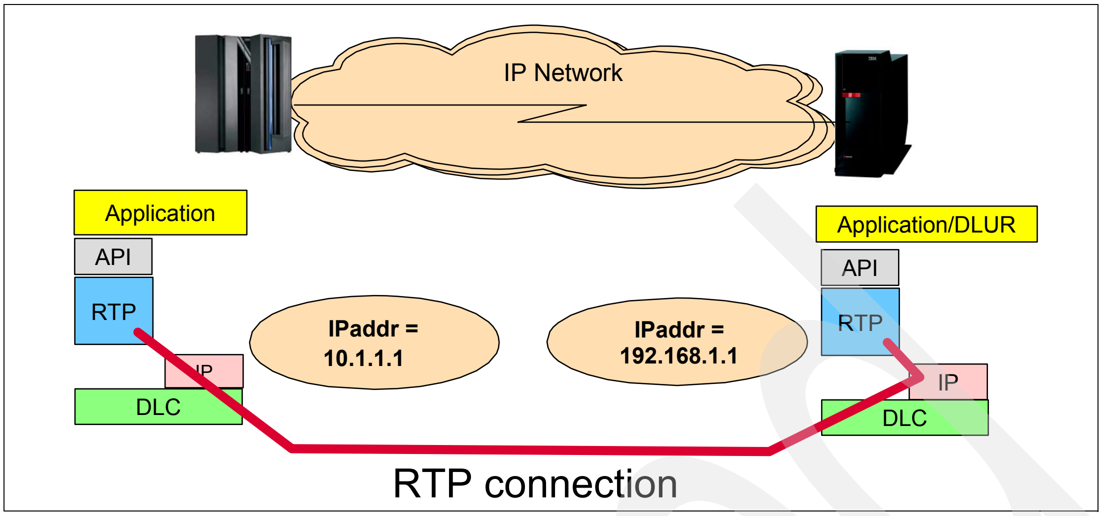


Figure 1-20 RTP manages connections between endpoints

Figure 1-21 shows an overview of basic RTP and ANR functionality. RTP is at the endpoints, ANR routes along the path. Upon recognizing a failure, RTP initiates and manages the path switching function.

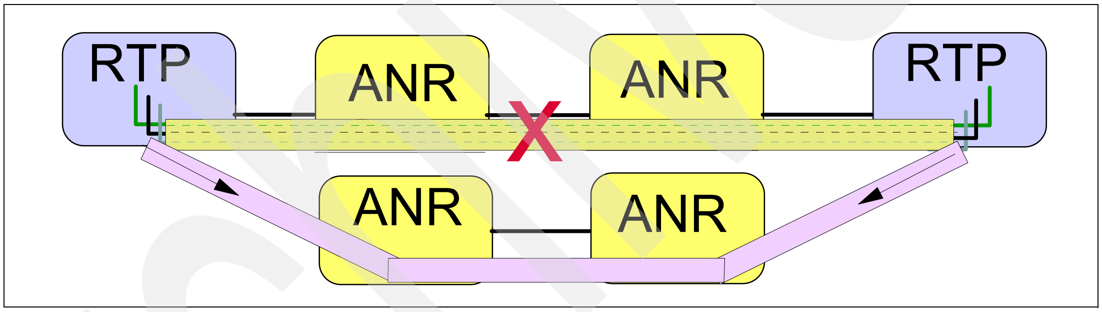


Figure 1-21 Basic RTP path switching

With faster, more reliable communication lines it is neither necessary nor desirable to perform error recovery, flow control, and complicated routing functions at intermediate nodes in a network. HPR takes full advantage of modern technology to eliminate these functions on the intermediate nodes, by ensuring that these checks are performed only at the endpoints of a session path. With HPR, each intermediate node has only a minimal switching function to perform.

Figure 1-22 shows where an RTP pipe can exist in the network and where the RTP function can be placed. In the first illustration, NN1 and NN3 do not have RTP support, but only the ANR routing support. NN2 has the RTP support and is used to provide the endpoint flow control and error checking on behalf of the End Node resource. In the second illustration, the RTP capability has been moved to the End Node, thus freeing NN2 from that responsibility. In the second illustration, the intermediate NNs can focus on routing and not have to concentrate on error recovery.

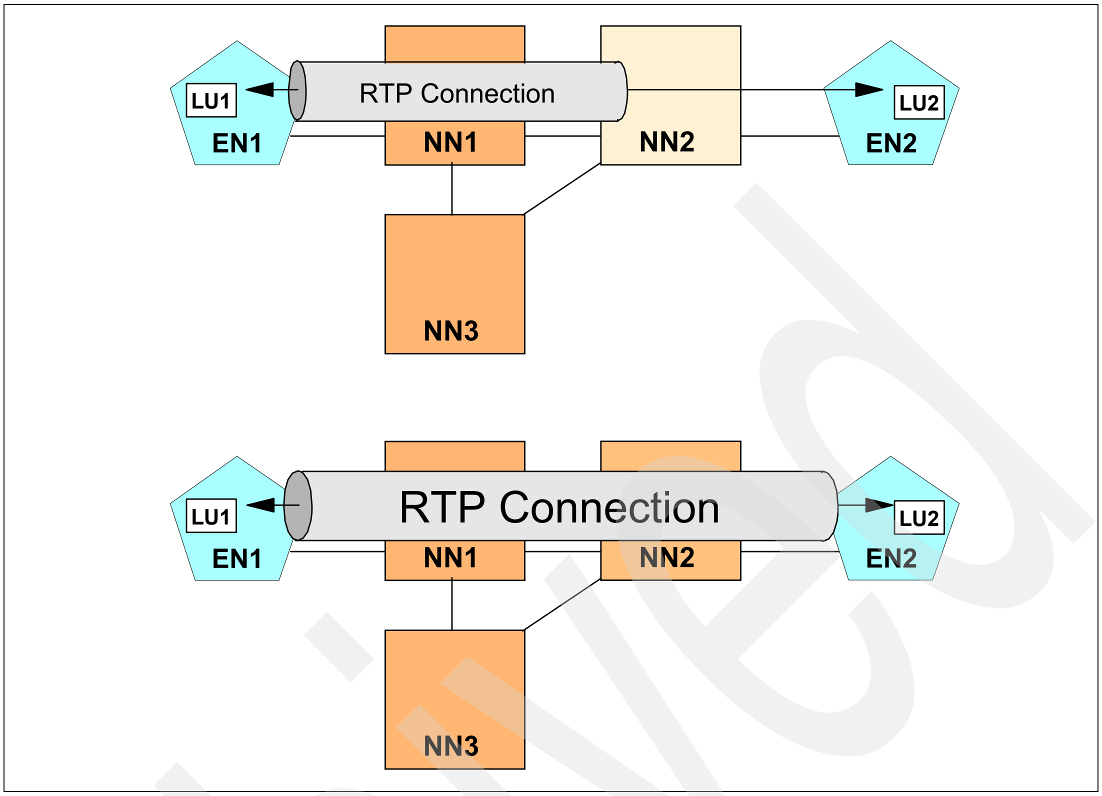


Figure 1-22 HPR Base and optional RTP support

Network administrators have always wanted the network to recover from errors, and to find an alternate path without requiring the user or operators to take action. HPR switches session routes to bypass link and node failures when an acceptable alternate path is available. This occurs transparently to the sessions, which are not disrupted.

HPR implements many functions in exactly the same way as base APPN. The same topology, the same directory, the same search methods, and the same route calculations are used. Priority queuing, Connection Networks, cross-network sessions, and other functions are supported by HPR. The same management data is carried on the same sessions in the same ways. HPR functions are invoked only at session initiation time, when the first HPR-capable node on the session path encounters a new session request. An RTP pipe represents a Class of Service (COS), and it can carry multiple sessions that use the same Class of Service, as illustrated in Figure 1-23.

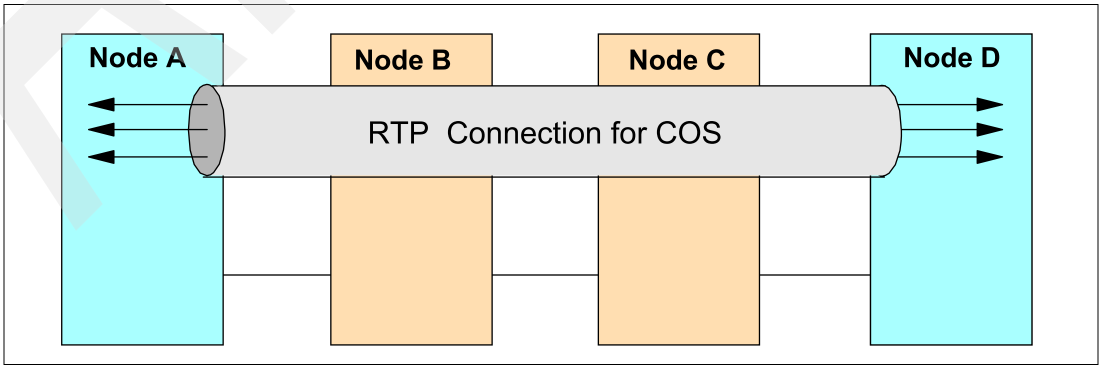


Figure 1-23 Multiple sessions over an RTP pipe for the same Class of Service

APPN with High Performance Routing over IP

In a mixed APPN and HPR topology network, a group of interconnected HPR nodes is sometimes referred to as an HPR subnetwork or an HPR subnet. When an HPR link is activated between a pair of adjacent HPR nodes, an HPR subnet is formed. The terms *base APPN subnetwork* and *base APPN subnet* may be used when referring to a part of the network that is not an HPR subnet.

Enterprise Extender requires *HPR-only connections*. An HPR-only connection is one in which the ANR and RTP functions are required. Therefore, an EE connection does not support nor permit just base APPN flows, because base APPN does not support the ANR and RTP functions. Figure 1-24 illustrates how Enterprise Extender utilizes HPR links.

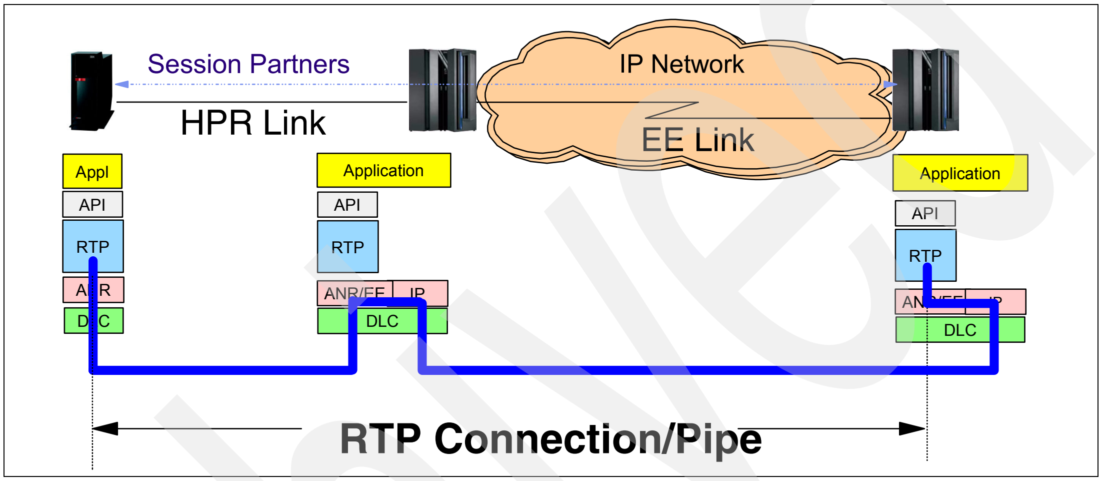


Figure 1-24 An Enterprise Extender link can be part of a longer HPR link: just one hop along the way

Characteristics of Automatic Network Routing (ANR)

The intermediate nodes in an HPR network have only one function, to route data quickly and efficiently from one link to another. This technique is known as *automatic network routing* (ANR). ANR is a low-level routing mechanism that minimizes processing cycles and storage requirements for routing packets through intermediate nodes.

A node performing ANR is aware of neither the sessions nor the RTP connections passing through it. If there is no awareness there is no need for cleanup or restoration after a failure, so the ANR technique provides for easy switching of session paths. Aside from the routing information, ANR nodes are aware of the transmission priority so that this can be maintained throughout the session path.

ANR is a source-routing protocol, which means the sender of a packet provides the necessary information about the physical path that the packet will use through the network in the network header. As HPR provides the ability to do nondisruptive path switching, the HPR architecture handles the case where the route changes in mid-session. Furthermore, the ANR label represents the onward link for each node, not the session as with base APPN routing. There is no session awareness in a node performing ANR routing. All it has to do is inspect the first ANR label in the packet header, strip it off, and forward the packet to the correct outbound link.

This label stripping technique is much more efficient than the label swapping technique used by intermediate session routing (ISR) in base APPN nodes. ISR requires that the node inspects the session identifier in the incoming packet, uses it to look up an entry in the session table, swaps it to the outbound session identifier, and forwards the packet.

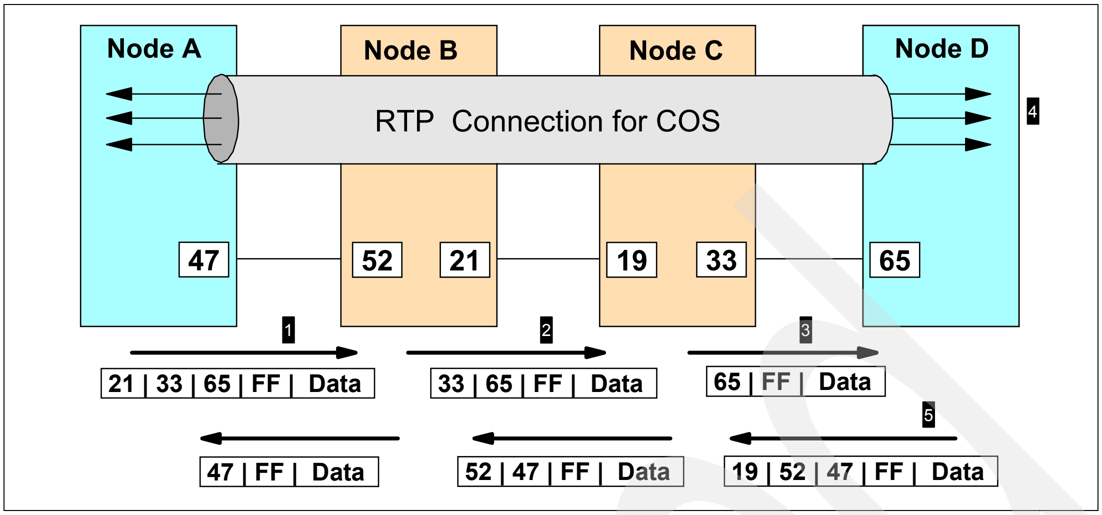


Figure 1-25 Use of ANR routing tables within an RTP connection

Figure 1-25 illustrates how ANR labels are processed:

1. Node A sends an NLP to node B with ANR labels 21 / 33 / 65 / FF in the NLP header as shown.
2. Node B looks in the header for the first ANR label. This is 21, so node B removes it from the header and transfers the truncated NLP to the link it knows as 21.
3. Node C receives the NLP, removes the next ANR label (33), and sends the NLP on the link it knows as 33.
4. Node D receives the NLP and recognizes that the next ANR label (65) represents not a link but the endpoint of the RTP connection. Therefore, it passes the data in the NLP to the higher protocol layers for processing.
5. The response to this message takes a similar course through the network in the opposite direction.

In the example, the ANR labels are one byte in length just for illustration purposes. Different products implement different lengths of the label. Because each node on an HPR path assigns the ANR labels that it is to interpret, there is no need for any other node to be aware of their length or meaning. Each node will find its own label at the start of any NLP it receives.

Characteristics of Rapid Transport Protocol (RTP)

In HPR, sessions are grouped together on logical connections called rapid transport protocol (RTP) connections. RTP is a connection-oriented, full-duplex protocol designed to transport data in high-speed networks. Sessions traversing the same route and using the same class of service can, and usually do, share an RTP connection. Data flowing on RTP connections flows as network layer packets (NLPs).

Of particular importance is the congestion control function called the adaptive rate-based (ARB) algorithm. This becomes significant as we tend to view EE as the implementation of HPR APPN as a transport layer over an IP network. It is this congestion and flow control algorithm that enables EE, and HPR, to compete with traditional transport layers in the TCP/IP protocol suite.

Note: An HPR network consists of a minimum of two APPN nodes that have implemented the HPR base functions and the RTP functions. These RTP nodes must be directly connected to each other, or they must be connected by a path of consecutive network nodes that support the ANR functions.

Figure 1-26 illustrates multiple RTP pipes between two nodes, based on COS assignments.

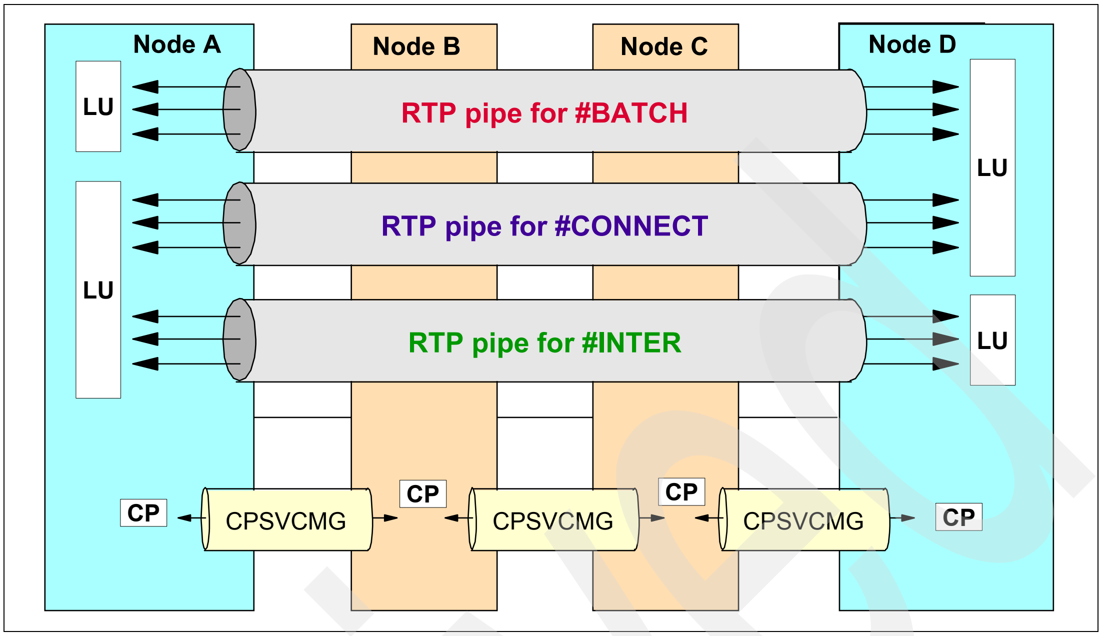


Figure 1-26 Multiple RTP pipes supporting respective Classes of Service

During activation of a link between two HPR nodes, these nodes exchange their HPR capabilities. When the APPN topology database is updated to reflect the new connection, the HPR capability of each node and link is included in the update. Thereafter, the APPN searching and route calculation algorithms are the same for HPR as for base APPN; the major difference comes when a session is started.

At session initiation time, the node that contains the primary LU receives a route selection control vector (RSCV) from its network node server, which represents the route that the session is to take. Base APPN processing requires that the RSCV is appended to the BIND, which then flows through the network establishing the session path. With HPR, however, the first RTP-capable node on the path must establish an RTP connection, if possible, over which the BIND may flow as an NLP. This RTP-capable node, therefore, examines the RSCV containing the HPR information from the topology database to see if such a connection exists or can be set up.

The RTP-capable node determines the furthest RTP-capable partner on the session path that is linked to it by a contiguous chain of ANR-capable nodes. If an RTP connection already exists over the same route and for the same class of service, the BIND (and therefore the new session) flows over that connection. If such a connection does not exist, one is established by means of the route setup message (RSETUP session) and its response. The route setup carries that portion of the RSCV that represents the HPR part of the session route, and finds its own way through the network exactly as a BIND would. The route setup exchange makes it possible for each intermediate node to assign ANR labels to the links on the path, and to make those labels known to the RTP endpoints that will use them to route NLPs. The route setup exchange also determines the performance characteristics of the RTP connection, so that the ARB flow control algorithm can be initialized with reasonable values.

Figure 1-27 illustrates RTP path switching around a failed route, keeping the sessions intact.

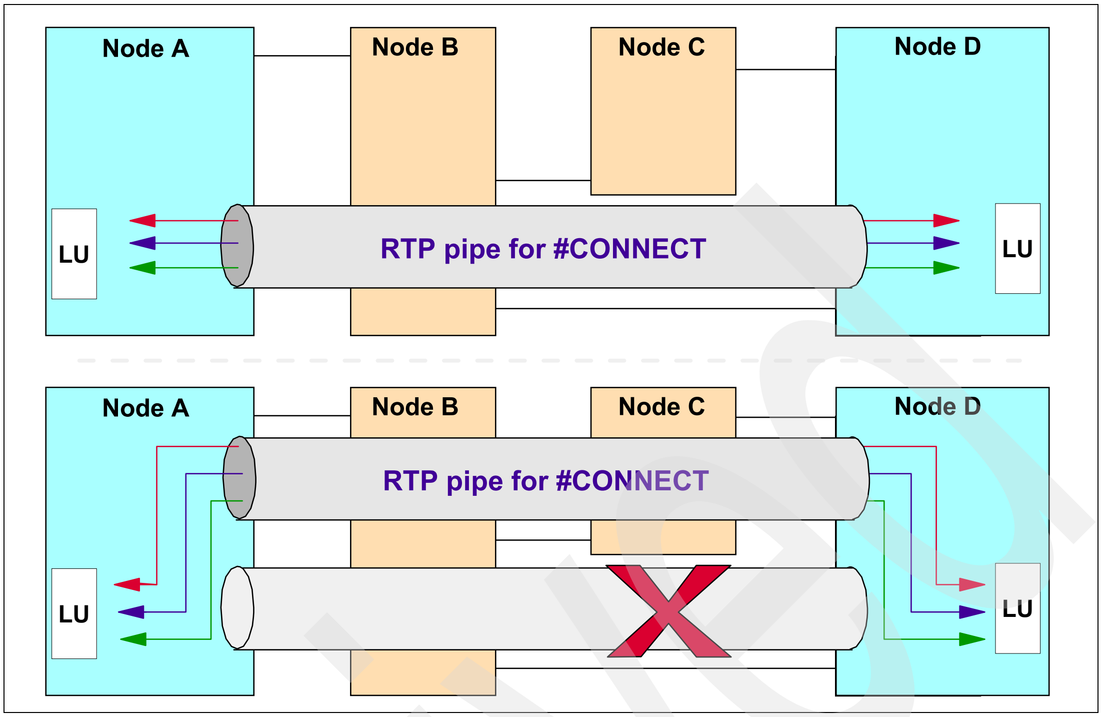


Figure 1-27 RTP path switching around a failure, using an alternate route

Dependent LU Requester and Server (DLUR/DLUS)

The APPN base architecture supports only type 6.2 LUs that do not require the services of a system services control point (SSCP). These independent LUs are able to start a session by sending a BIND and by doing so become the primary logical unit (PLU) of that session. APPN base architecture does not support other LU types, nor does it provide functions such as SLU-initiated sessions, session queuing, or third-party initiation, which are widely used in subarea SNA.

Session services extensions are optional APPN functions that may be implemented to support dependent logical units and provide additional services for independent logical units.

Implementing the session services extensions:

- Enhances the support provided for LU 6.2 sessions
- Is required to fully support the attachment of dependent LUs to APPN environments
- Is necessary on the interchange node connecting APPN and subarea networks in order to provide transparency to the LUs in subarea networks

While the session services extensions function makes it possible for dependent LUs to use APPN environments for LU-LU sessions, the following restrictions still apply:

- SSCP-PU and SSCP-LU sessions cannot use APPN connectivity.
- The PU2.0 node containing the LU must be adjacent to a subarea boundary node.
- The node containing the LU must have subarea connectivity to its owning SSCP.

The SLU-PLU session can use APPN between a composite network node and the node containing the PLU, but must use subarea connectivity for SSCP-PU and SSCP-LU sessions, as depicted in Figure 1-28.

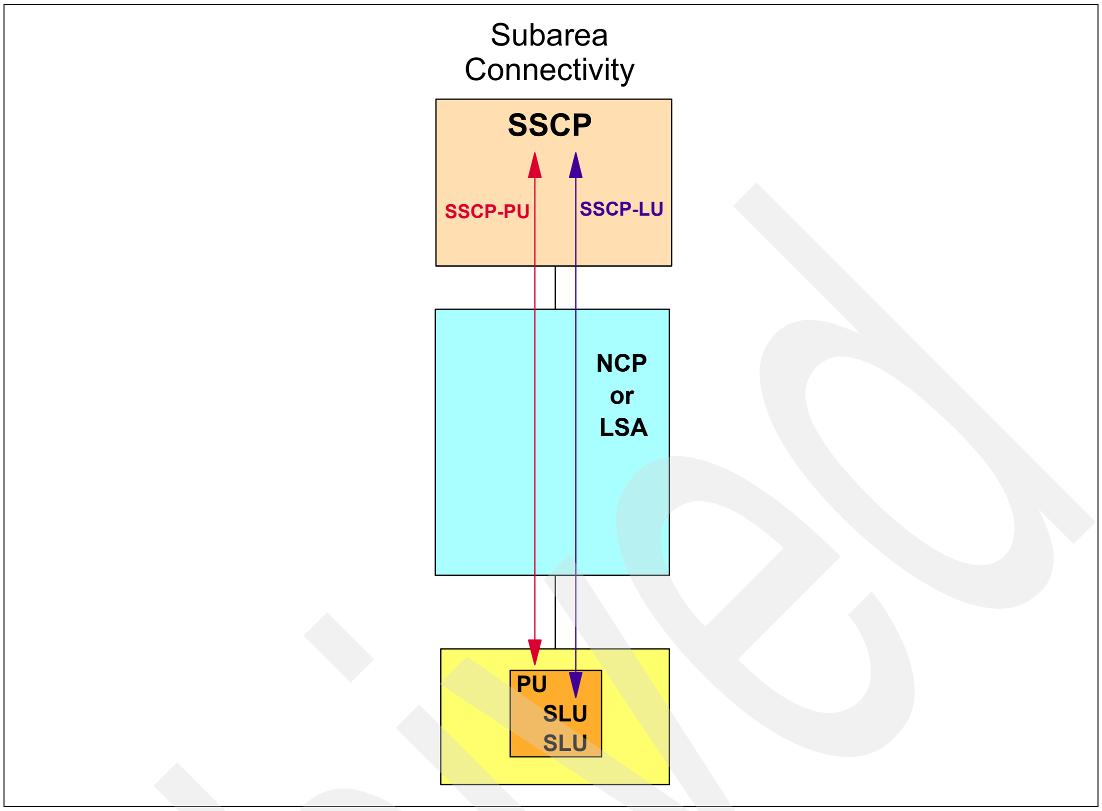


Figure 1-28 Subarea Dependent LU SSCP connectivity without Dependent LU Requester/Server

The DLUR/DLUS functions remove the requirement that a PU2.0 node containing dependent LUs must be adjacent to a subarea boundary node. The *Dependent LU Server* (DLUS) and *Dependent LU Requester* (DLUR) are optional APPN functions that provide more flexibility in connecting dependent logical units to their owning system service control points (SSCPs).

Dependent LU Server

The dependent LU server function is a feature of an interchange node (ICN) with session services extensions. This function provides server support for dependent LU requester clients in which SSCP-PU and SSCP-LU flows that are going to a PU2.0 node are encapsulated within LU 6.2 sessions. The PU2.0 can be externally attached to the requester or be a PU2.0 image within the requester.

Dependent LU Requester

The dependent LU requester function is an enhancement for an APPN end node or network node. This function is the client side of the dependent LU server function in which SSCP-PU and SSCP-LU flows going to a PU2.0 node attached to the requester are encapsulated within LU 6.2 sessions. Figure 1-29 shows an implementation of DLUR working on behalf of a dependent LU.

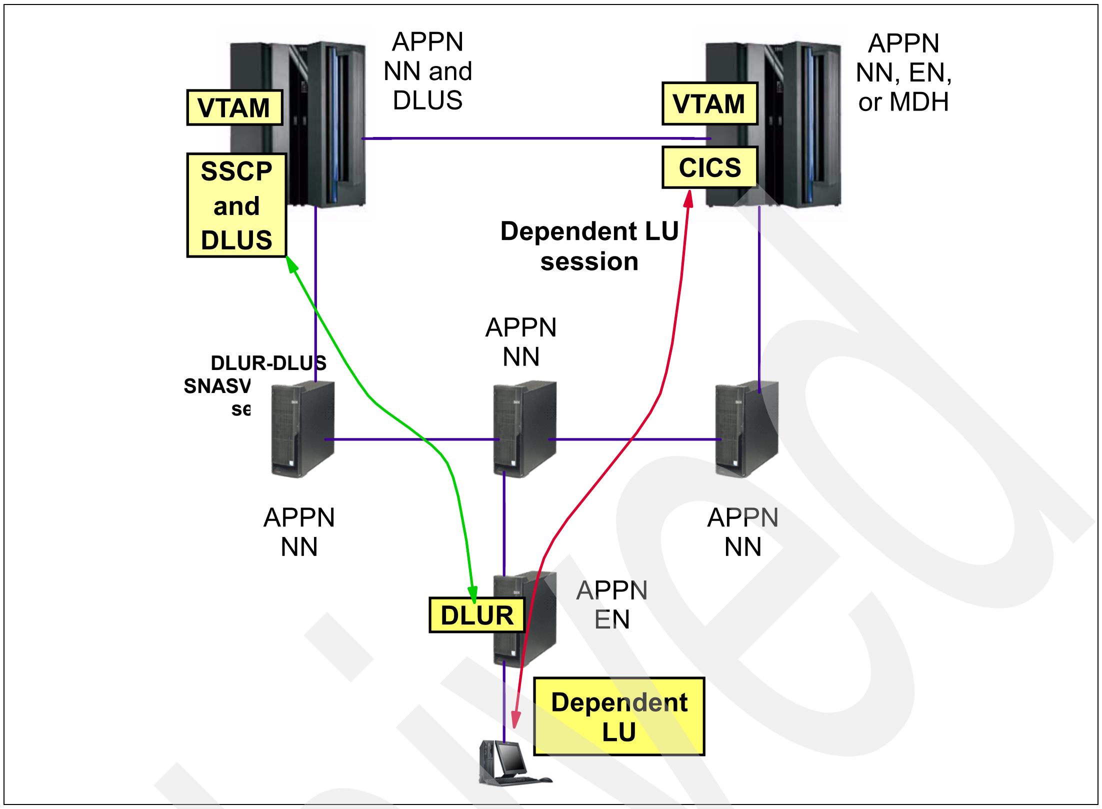


Figure 1-29 DLUR/DLUS session path can be different from the LU-LU session path

The requester function provides a remote boundary function for dependent LUs. The DLUR function may reside in the same node as the secondary LU or be provided by a node adjacent to and upstream from the secondary LU. These two configurations are illustrated in Figure 1-30, where:

- In configuration A, the DLUR is in the same node as the dependent LU.
- In configuration B, the DLUR is in a node connected to the PU2.0 node containing the dependent LU.

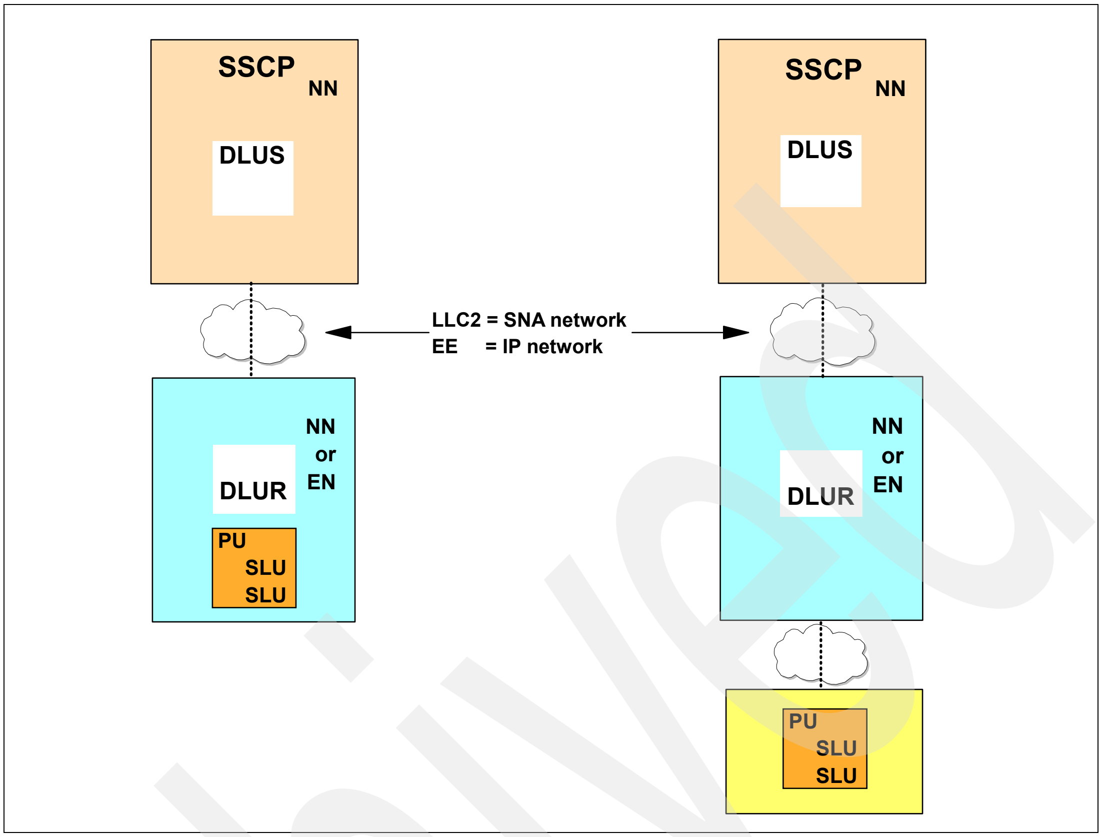


Figure 1-30 Dependent LU Server/Requester: Internal and External DLUR

CP-SVR Pipe

The CP-SVR pipe is the term used to describe the LU 6.2 sessions encapsulating the SSCP-PU and SSCP-LU session flows between the DLUS and the DLUR. These sessions are similar to CP-CP sessions in that each node has a contention-winner and contention-loser session to the other. The CP-SVR pipe is established using a mode called CPSVRMGR that uses the SNASVCMG COS. The DLUR must know its DLUS, and the DLUS must know which dependent LU requester to contact for a particular PU activation.

SSCP-PU and SSCP-LU flows that are required to set up and manage a dependent LU-LU session are carried encapsulated inside the CP-SVR pipe between the requester and the server. The CP-SVR pipe can carry encapsulated SSCP-PU and SSCP-LU sessions for multiple PUs as shown in Figure 1-31.

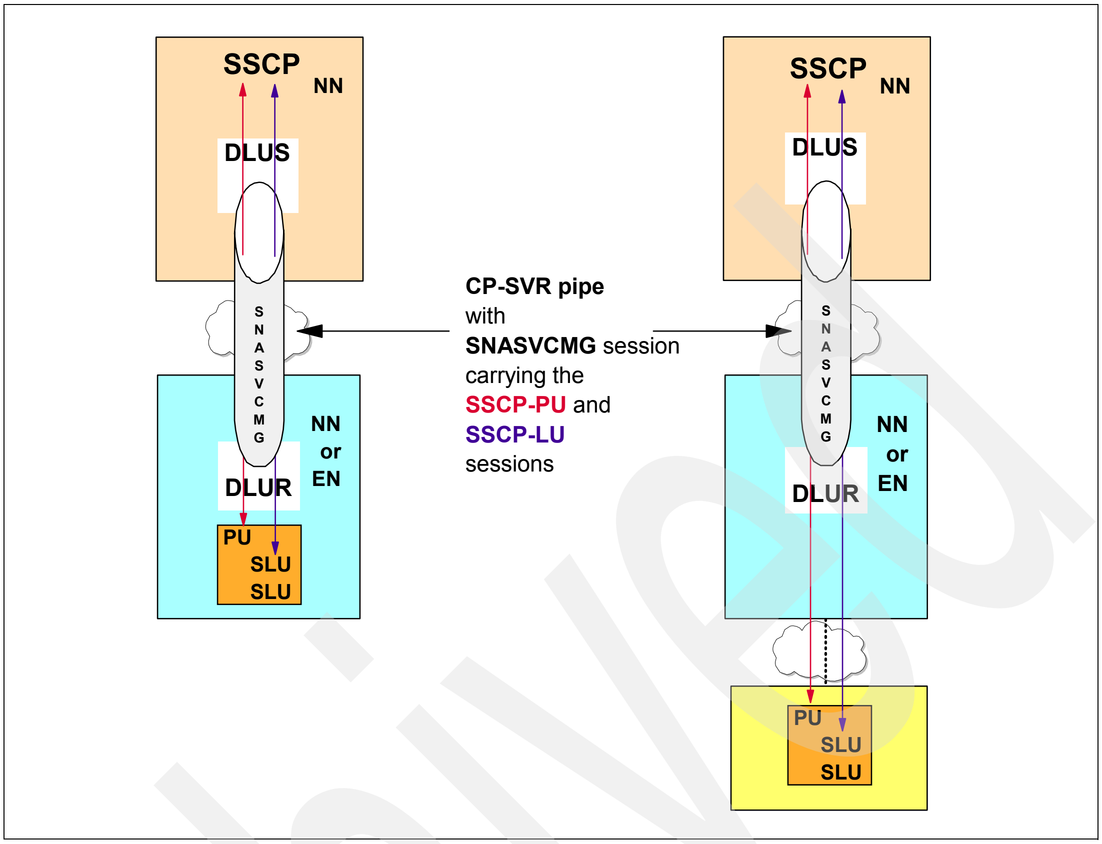


Figure 1-31 CP-SVR Pipe and Encapsulated SSCP-PU and SSCP-LU Sessions

The following statements apply to the CP-SVR pipe:

- Either the DLUS or the DLUR may initiate the CP-SVR pipe.
- CP-SVR sessions are only initiated when some form of PU activation is requested.
- The CP-SVR pipe is deactivated when it is no longer required.
- The CP-SVR pipe between the DLUR and DLUS may traverse APPN subnets.
- The CP-SVR pipe cannot cross through a subarea subnet.
- The resulting LU-LU sessions can be routed over a different path from the encapsulated SSCP flows.

Figure 1-32 shows the general positioning of a DLUR in the network, and how it can replace DLSw.

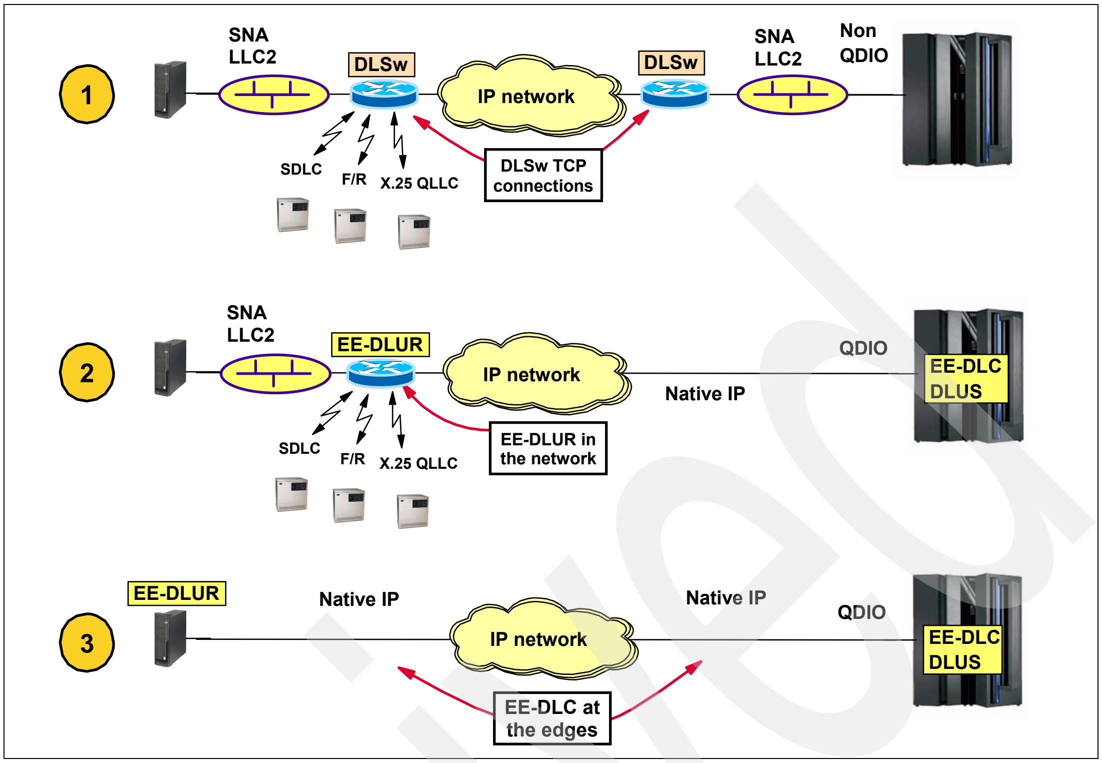


Figure 1-32 DLUR positioned to replace DLSw

Note the following explanation for Figure 1-32:

- 1: DLSw sits in the middle of the network with SNA LLC2 at the ends. The z/OS system uses an SNA non-QDIO interface.
- 2: Replacing the DLSw peering routers with EE DLUR/DLUS enables the z/OS system to use native IP QDIO interfaces with DLUS. The remote router then uses EE DLUR to continue to support the remote SNA LLC2 end devices.
- 3: By configuring EE DLUR on the remote device itself, EE DLUR can be removed from the intermediate router, and native IP can be supported end-to-end.

APPN connection networks

A Connection Network saves having to define links between every single pair of nodes that can communicate directly across a shared transport facility such as an IP network or a LAN. Within each node, you define just *two* connections in total instead of one for each per potential partner node. The *first* connection is required for CP-CP sessions (because these cannot flow across a connection network) and the *second* is to a virtual node that represents the shared transport facility. When the session path is calculated and the RSCV is presented to the primary end of the session, that node recognizes the virtual node in the RSCV and replaces it with a direct connection to its desired partner. For this to work, the address (IP address in the case of EE) of the partner must be present in the RSCV, and therefore in the topology database. Nodes that support connection networks include these addresses in their topology update reporting.

Purpose of a connection network

A *shared-access transport facility* (SATF), such as an Ethernet, makes it possible for direct connectivity between any pair of link stations attaching to the facility. Direct connectivity avoids session traffic being routed through intermediate network nodes but requires link definitions at a node for all partner nodes to which connectivity is required. Figure 1-33 illustrates the difference between direct routing and intermediate node routing with no SATF.

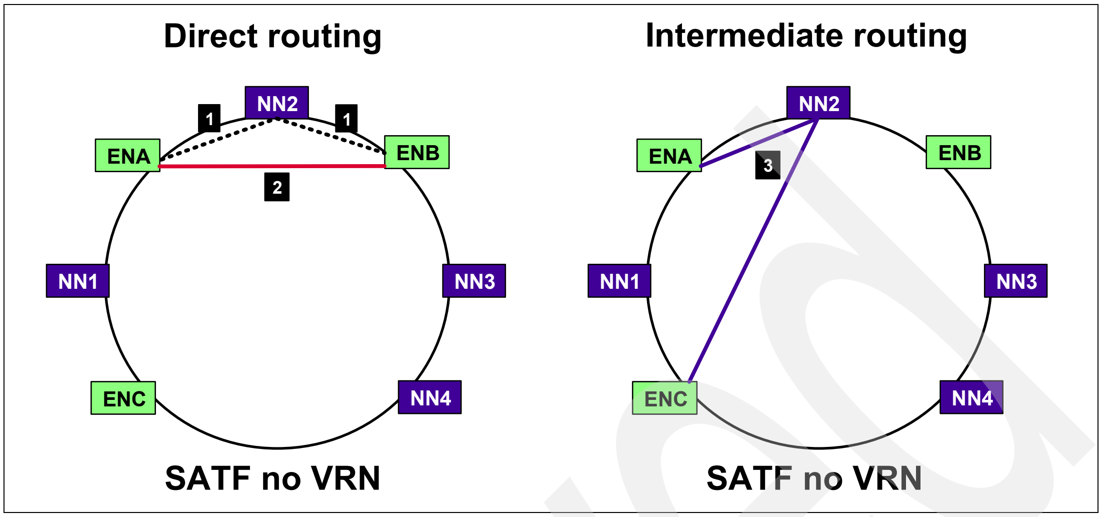


Figure 1-33 Direct versus Intermediate Routing with no SATF

Note the following explanation for Figure 1-33:

- 1: ENA and ENB both have a CP-CP session with the same Network Node Server (NN2).
- 2: ENA and ENB have a direct link established via explicit definitions. Although they need the assistance of the network node server to establish a session, the session data is exchanged directly between the two nodes.
- 3: However, no explicit direct link has been defined between ENA and ENC and, therefore, session data will always be routed through at least one intermediate network node, NN2 in this case.

Figure 1-34 shows any-to-any direct connections that are a result of each node explicitly defining every other node to which it connects.

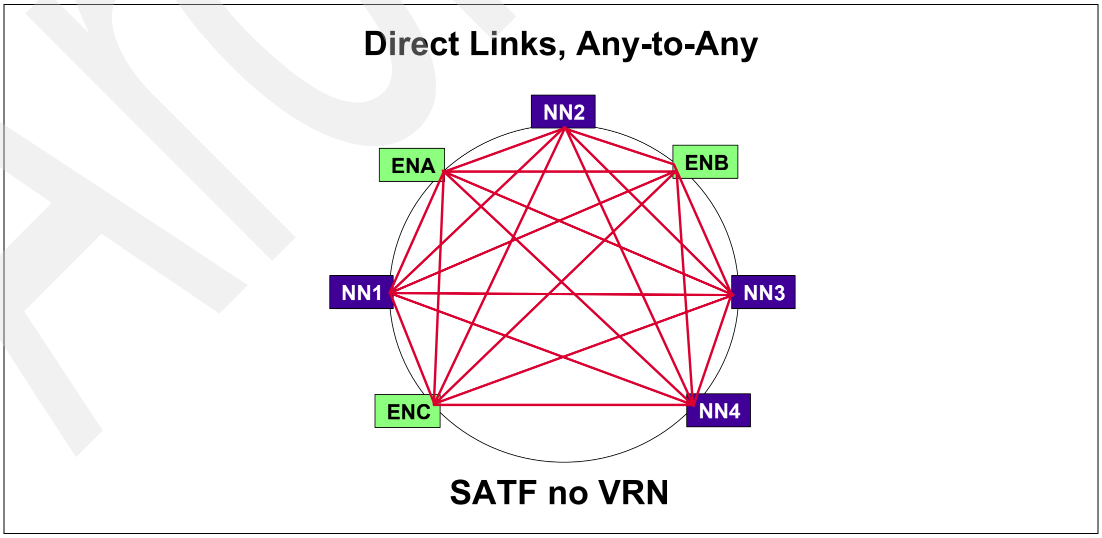


Figure 1-34 Direct Links, Any-to-Any

Note the following explanation for Figure 1-34:

- If any-to-any direct connectivity is required to avoid routing through intermediate network nodes and sending the same data more than once across the SATF, then the number of definitions required is proportional to the square of the number of nodes on the SATF. As the number of nodes grows, the number of definitions will become very high when there is no connection network defined.
- Another drawback of increasing the number of direct links between APPN network nodes is the number of *topology database updates* (TDUs) flowing in the network grows rapidly and may degrade the performance of the network.

Figure 1-35 depicts the TDU flows when any-to-any direct routes are all defined without the use of a connection network.

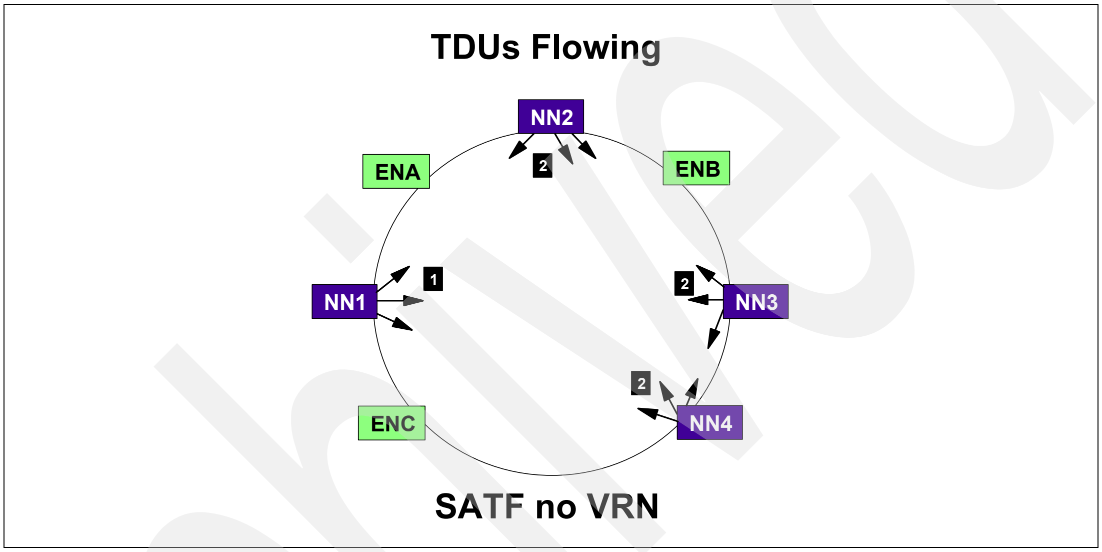


Figure 1-35 TDUs flowing

Note the following explanation for Figure 1-35:

- 1: An APPN network node (NN1) broadcasts TDUs to all adjacent network nodes.
- 2: After having received a TDU, an NN forwards the TDU to all adjacent APPN network nodes. So, instead of receiving one copy, NN2, NN3 and NN4 receive the TDU three times. Flow reduction mechanisms prevent the network nodes from continuing to forward the TDUs.

The above discussion concludes that defining any-to-any links on a SATF provides optimal session routing but requires a high number of definitions and results in high volumes of TDUs.

Benefits of a connection network

To alleviate the problems of defining any-to-any direct links, APPN allows nodes to define a *virtual routing node* (VRN) to represent their attachment to an SATF. Session traffic between two nodes that have defined the VRN can be routed through the VRN without passing through any real network node. TDUs will never be exchanged with a VRN.

The SATF and the set of all nodes defined as having a connection to a common virtual routing node representing the SATF are said to comprise a *connection network* (CN). The connection network is given a network-qualified name. The CN name is used as the CP name of the virtual routing node.

Note: It is important to understand that session setup data and TDUs are routed through an APPN environment using CP-CP sessions. Nodes cannot establish CP-CP sessions *with* nor *through* a VRN because it is not a real node. Two nodes can establish CP-CP sessions only if a direct link has been defined between them.

Figure 1-36 shows the minimal definition requirements for an APPN end node when using a connection network virtual routing node.

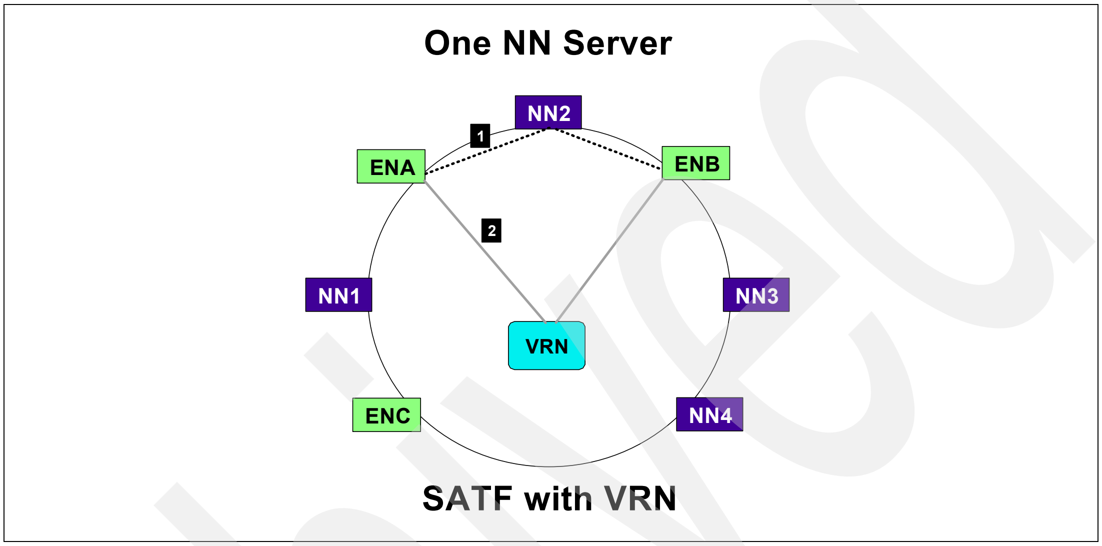


Figure 1-36 SATF with VRN: one NN server

Note the following explanation for Figure 1-36:

- Session establishment between LUs owned by APPN end nodes requires assistance from a network node server if no direct link has been defined between the APPN end nodes. Because the APPN end nodes cannot have CP-CP sessions with a network node server through a VRN, it is necessary for them to have defined CP-CP capable links to their respective network node servers, as well as defined connections to the VRN.
- ENA has defined two connections:
  - 1: one to its network node server, NN2
  - 2: one to the VRN

ENB has likewise defined two similar connections. When two End Nodes do not have a direct link to each other, APPN always requires the data to flow through an intermediate routing node. Without using the VRN, the data flow path between ENA and ENB is from ENA through NN2 to ENB (ENA-NN2-ENB). Similarly, when using the VRN, the data flow path between ENA and ENB is logically from ENA through the VRN to ENB (ENA-VRN-ENB). In the first case, the path is set up to include the real node NN2. In the second case, the path is set up to include the virtual node, and thereby, ends up being a direct connection over the connection network.

In the above discussion, the two end nodes used the same common Network Node Server. But what happens when they use different servers, yet share a common connection network? See Figure 1-37.

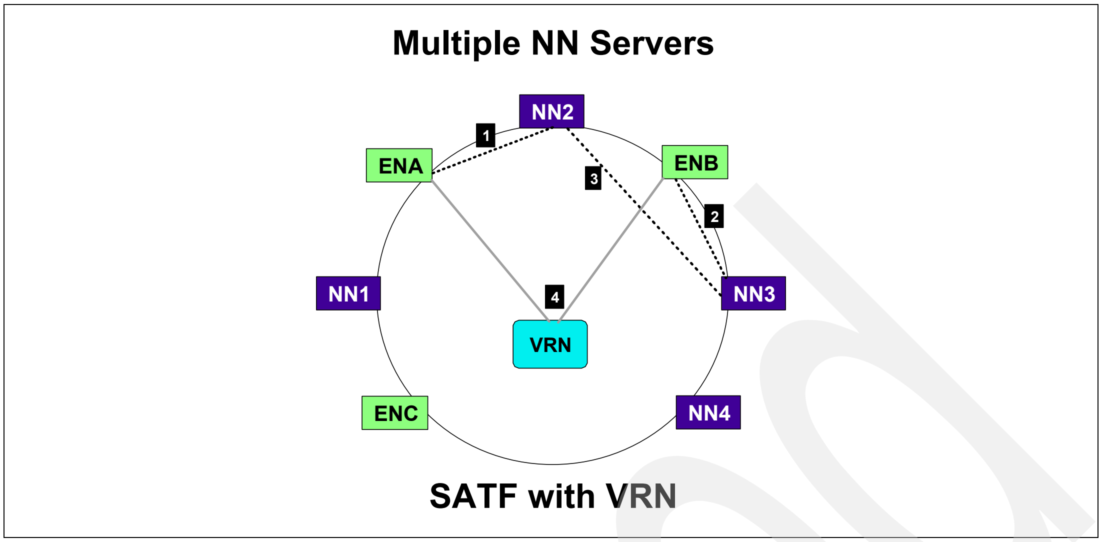


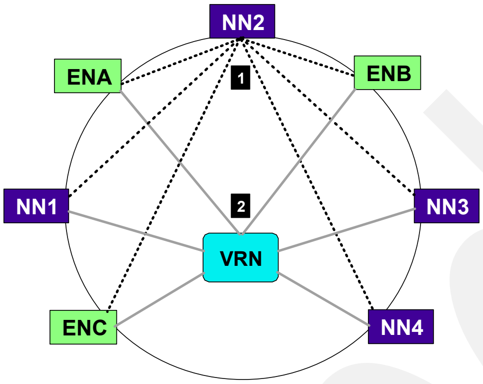
Figure 1-37 SATF with VRN: multiple NN servers

Note the following explanation for Figure 1-37:

- 1: ENA has a CP-CP session with its network node server, NN2.
- 2: ENB has a CP-CP session with its network node server, NN3.
- 3: Network nodes cannot establish CP-CP sessions through a VRN. Therefore, if two APPN end nodes (ENA and ENB) do not share the same network node server, session establishment between LUs on ENA and ENB is possible only if their network node servers have CP-CP connectivity to each other. This is also required to support session establishment between LUs on two network nodes. CP-CP connectivity between two network nodes requires that the two network nodes have defined a link between each other and CP-CP sessions have been established between the two nodes, or that the two network node servers can exchange data via one or more intermediate network nodes with active CP-CP sessions between each pair of adjacent network nodes.
- 4: By defining the connection network, VRN, to both ENA and ENB, the two network nodes can determine the preferred path to be through the VRN, and give ENA and ENB a direct path over which to connect.

The benefits of defining a VRN can be seen in Figure 1-38.

Any-to-Any via VRN



SATF with VRN

Figure 1-38 SATF with VRN: any-to-any via SATF

Note the following explanation for Figure 1-38:

- 1: To have any-to-any connectivity without session data being routed through real network nodes requires only two link definitions in each node: one to the VRN and one to a common network node as depicted in the figure as NN2. Because NN2 is the *primary* (preferred) NN server for the ENs, it is the only node that requires link definitions (for CP-CP) to all nodes. NN2 assists only in session setup; no session data will be routed through it.
- 2: Session data between any of the nodes that define the VRN will flow through the VRN. This means that they have a direct link between each other over which the data will flow.

  For performance and backup reasons, more than one common network node can be defined. So this discussion should be applied to the *backup* (secondary) NN server as well.

Because the number of predefined direct connections has been reduced, the number of TDUs flowing across the network has been reduced, as seen in Figure 1-39. TDUs are no longer sent to every defined node, but only to the NN server. Then it forwards the TDUs only to the NNs with which it has CP-CP sessions.

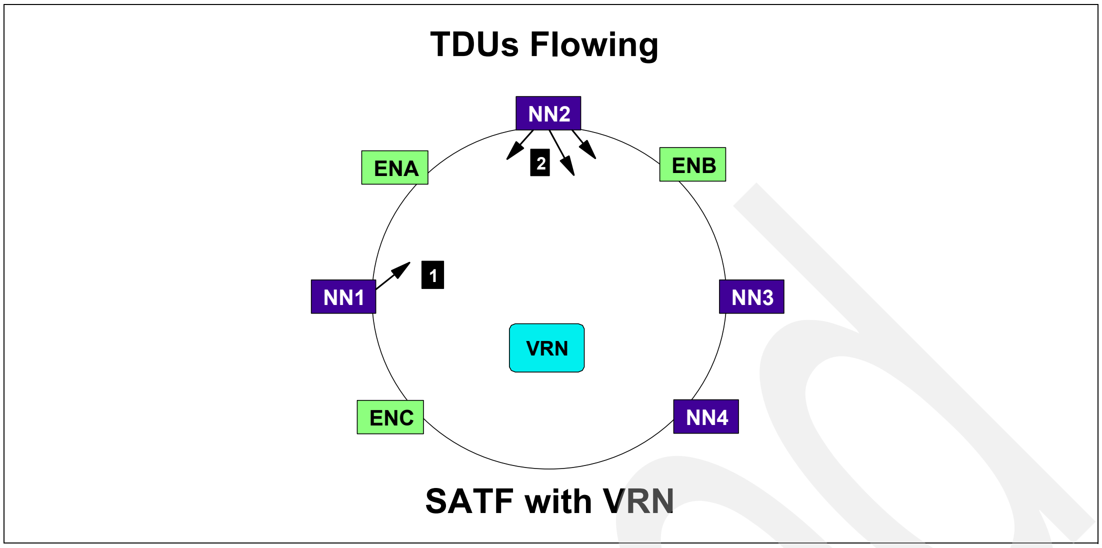


Figure 1-39 SATF with VRN: TDUs flowing

1.4.3 Functions of HPR/IP (EE)

This section discusses the following HPR/IP functions:

- "HPR/IP EE connection networks"
- "HPR/IP EE hardware interface functions"
- "HPR/IP EE network functions"
- "HPR/IP EE security functions"
- "EE with Multiple VIPAs"
- "EE with Extended Border Nodes"

HPR/IP EE connection networks

One extremely important aspect of EE is the ability to view the IP network as an APPN Connection Network. The benefit in the IP environment is the ability to dynamically establish a single one-hop HPR link to any host to which IP connectivity is enabled, provided that the host implements EE. This positions the routing function to be handled entirely within IP, because IP routers are serving as the only routing nodes (hosts) in the network and no ANR routing is done. It also minimizes the toll of routing in the network, because this function does not have to be duplicated by IP and HPR.

In some cases, however, the duplication of routing function is desirable when considering redundancy from an HPR point of view. If an EE link is available in some intermediate HPR node as well as some other native HPR link, the ANR routing function could recover from potential errors in the IP network. However, it is generally best to have IP primarily perform all routing functions.

Define as many local or global connection networks as your configuration requires. By configuring 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.

Local EE connection networks

All participants on a local connection network (VNTYPE=LOCAL) must be located within the same APPN topology subnetwork. 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).

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.

Global EE connection networks

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). In a global connection network 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.

HPR/IP EE hardware interface functions

EE can use any interface supported by the TCP/IP stack. Use OSA-Express or OSA-Express2 port in QDIO mode and HiperSockets for optimal performance and functionality. 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.

HPR/IP EE network functions

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 designing 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 z/OS, because EE can affect the way the hardware interfaces are defined.

HPR/IP EE security functions

There are several security techniques supported by EE. Included in these are:

- “EE with IPSec”
- “EE with Authentication”
- “EE with Encryption”
- “EE with Network Address Translation”

EE with 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 off-loaded to an attached router platform, as follows:

- By placing the IPSec endpoint on z/OS, you have end-to-end protection, but your System z processor will incur the cost of the encryption.

- By off loading the IPSec function to a router, you off load the encryption cost but you have an unprotected segment between z/OS and the router hosting the IPSec endpoint.

EE with Authentication

If you need to provide authentication, you can use IPSec, which includes an authentication scheme. You can use IPSec authentication in one of the three ways:

- Stand-alone
- In combination with IPSec encryption
- In combination with SNA session-level encryption

EE with 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. IPSec encryption can be used in the following ways:

- 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.

EE with Network Address Translation

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).

Important: Because EE has dependencies on port assignments and a one-to-one IP address mapping, NAPT is incompatible with EE.

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.

Important: Because EE requires a unique one-to-one IP address mapping, *dynamic* network address translation is generally incompatible.

The following statements apply when using EE on IP networks that implement NAT:

- For predefined EE connections, the remote IP address is the public address translated or mapped by the NAT device.
- To enable EE connection networks to coexist with NAT, the remote partner is defined using the HOSTNAME operand, not the IPADDR operand.

EE with Multiple VIPAs

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.

Enterprise Extender supports the use of multiple static VIPAs for establishing these parallel connections between EE nodes. These are some of the advantages to having multiple VIPAs:

- Different VIPAs can be provided to different vendors, which eases the tasks of network management, problem determination, and firewall administration.
- Multiple connection networks can be defined using multiple VIPAs.
- There is more potential for flexibility in making IP routing decisions.

It would be beneficial to define two different global VRNs; one global VRN that is defined only by nodes within one of your own subnetworks, and the other 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 permitting internal systems to directly connect to any of the other systems in your subnetworks.

The use of multiple VIPAs, and therefore multiple VRNs, can be used 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.

EE with Extended Border Nodes

Connecting networks with extended border node connections is quite simple, as compared to SNI. You get an extended border node connection by starting both VTAMs as border nodes at the connection endpoints.

Figure 1-40 shows how EE with its EBN capability, can replace the SNA/SNI functions.

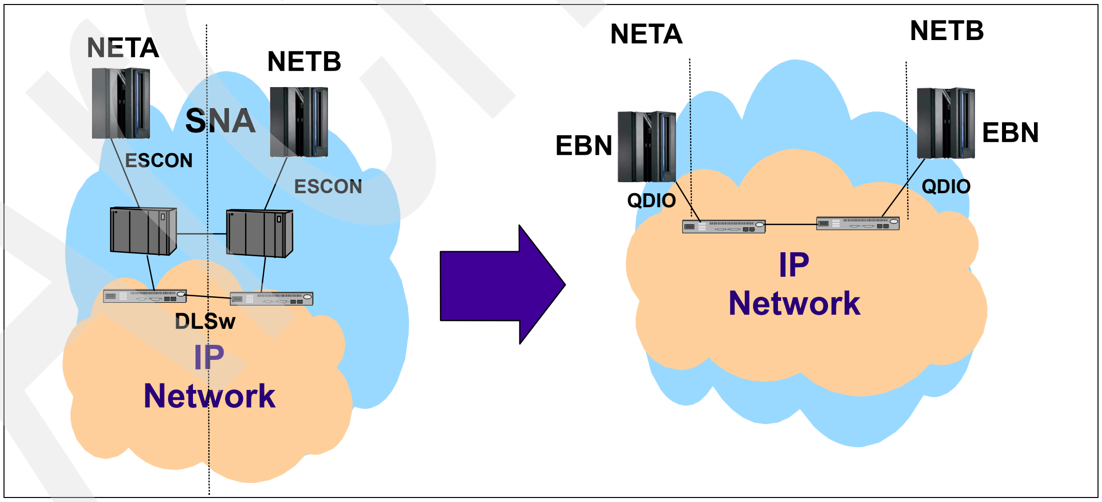


Figure 1-40 SNA/SNI versus EE/EBN

EE used in conjunction with the APPN extended border node function can replace SNA Network Interconnection (SNI) functionality in a way that does not require SNA hardware, such as the 3745. 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 the other partner company.

Figure 1-41 shows an EBN environment without a Global Connection Network.

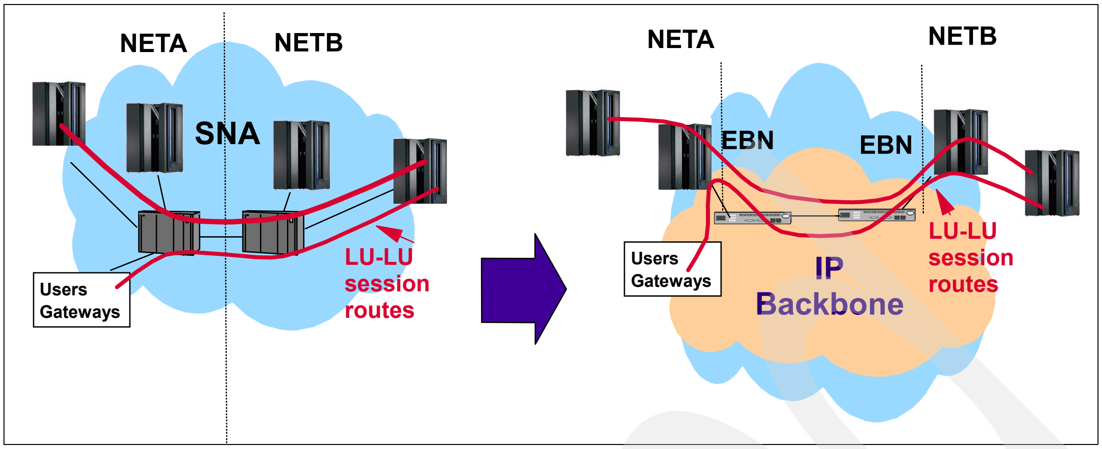


Figure 1-41 EE/EBN configuration without a Global Connection Network

Figure 1-42 shows the presence of a GVRN and the benefit of using it to reduce the path length between the two ENs. A direct link between the two ENs also reduces the traffic management burden on the two EBNs.

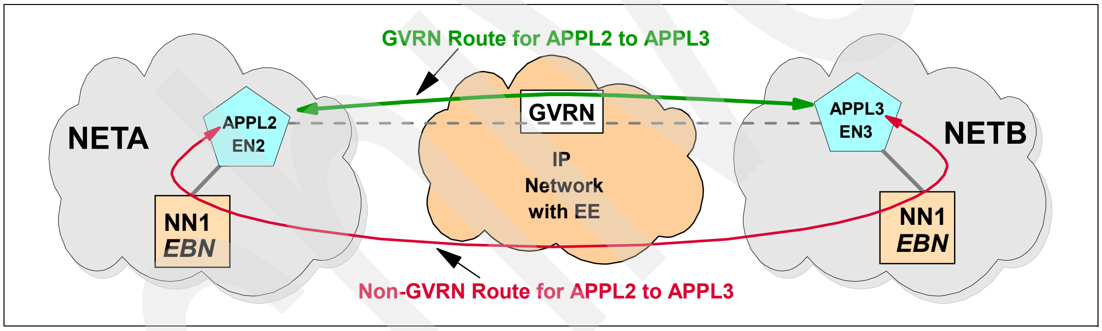

GVRN Route for APPL2 to APPL3
NETA
APPL2
EN2
GVRN
IP
Network
with EE
APPL3
EN3
NETB
NN1
EBN
NN1
EBN
Non-GVRN Route for APPL2 to APPL3

Figure 1-42 Global Connection Network benefit

A clear understanding of APPN search methods and VTAM's use of the Adjacent Cluster Table (ADJCLUST) is necessary when setting up EBN connections. Both of the following books have chapters that discuss APPN searching:

- *Inside APPN: The Essential Guide to the Next-Generation SNA*, SG24-3669
- *Subarea to APPN Migration: VTAM and APPN Implementation*, SG24-4656

2

Planning for EE

In this chapter we discuss the methodology used in planning our Enterprise Extender (EE) implementation, from a z/OS perspective.

For the purpose of this planning chapter, we assume APPN/HPR has been implemented on the z/OS platforms, and that TCP/IP and dynamic routing services of z/OS Communications Server are functioning correctly.

If you are not familiar with the APPN/HPR environment, we recommend reading the APPN discussions in Chapter 1.

> **Important:** Anynet was removed from z/OS Communications Server V1R8. Enterprise Extender is the successor to Anynet and is now available for most IBM platforms.

The topics covered in the chapter include:

- “Defining the objectives”
- “Determining the scope”
- “Creating a strategy”
- “Developing a high-level design”
- “Implementing the EE design”

2.1 Overview

EE requires a minimum of APPN and HPR enablement in VTAM as well as a functioning TCP/IP stack. EE uses UDP/IP as a transport mechanism, creating the possibility for direct connectivity between APPN nodes across an IP network. This is achieved by utilizing the EE Data Link Control (DLC) function in z/OS Communications Server.

With the EE DLC, VTAM can provide faster, simpler and more flexible network connectivity than is currently available in an SNA network. EE also provides an opportunity to exploit APPN/HPR functions as well as other System z networking features that may currently not be in use. For example:

- OSA-Express or OSA-Express2 1000BASE-T and Gigabit Ethernet features in conjunction with Queued Direct Input Output (QDIO) technology can provide high-speed bandwidth for network connections. In most cases, they offer increased bandwidth over other LAN technologies.
- HiperSockets provides high-speed connectivity between TCP/IP stacks running in different logical partitions (LPARs) in the same System z server. HiperSockets uses internal QDIO to pass data traffic between LPARs at memory speed.
- A direct UDP/IP connection between two APPN nodes eliminates the requirement for gateways or protocol encapsulation. HPR provides end-to-end session management for the LU-LU sessions.
- With an IP connection between two APPN nodes, the reach of the APPN environment is limited only by the reach of the IP network.
- By using IP to provide connections, a connection network or VRN support can simplify a configuration and allow any-to-any connectivity for EE endpoints.

Typically, the first EE implementation phase will not include all SNA resources in your network. Nonetheless, the initial EE implementation may be the first step in that direction. Therefore, it is necessary to determine what you want to achieve long term. For example, the implementation of Enterprise Extender may also present an opportunity to plan for token-ring to Ethernet migration, or use other APPN and IP networking functions that will increase availability.

It is important that the method involves reviewing your current environment to understand which resources are installed and how they are being used. There may also be an opportunity to consolidate some of the resources.

2.2 Defining the objectives

Objectives are what you plan to accomplish, they define the end result of your actions. For example:

- The objectives of our EE implementation scenarios were:
  - Implement four non-mainframe servers using EE DLC
  - Provide PU type 2 support for SNA applications running on System z
  - Provide EBN connectivity from System z servers to other networks using EE
- Some other goals for implementing EE might be to:
  - Implement new servers using EE DLC. This may occur because:
    - An SNA application is being installed on a server with no SNA network connectivity.
    - The hardware for an existing server is being upgraded.

- Provide business partner connectivity using EE:
  - IP connections are preferred over SDLC lines for new or existing business partners
- Make use of the EE DLC to provide better performance or availability through the IP network:
  - This may also include a hardware refresh for the server or a new installation.
- Extend the reach of the APPN environment to additional data centers
  - This can occur as part of corporate acquisitions.
- Create Branch Extender nodes using EE to the mainframes and LLC2 to downstream clients.
  - This can be to move devices off of IBM 374x equipment.

2.3 Determining the scope

Given the objectives identified in the previous step, it is now necessary to put a scope around the objectives. The scope controls which SNA/APPN connections will be moved to the EE environment and therefore, which devices will be affected. It is very important that you comply with the scope of the project, otherwise it could become difficult to manage.

The main categories in which EE connections are likely to be implemented are:

- Between mainframe LPARs, either within a site or across multiple sites
- Between peripheral devices and SNA applications hosted on the mainframe
- As a replacement for SNI connections with business partners

In this section we discuss the following:

- Identifying devices and connections
- LPAR-to-LPAR connections
- Non-VTAM application servers
- Peripheral devices
- EBN to replace SNI

Figure 2-1 on page 52 shows an environment with the most commonly used connection types. We use it to determine our scope. Note that APPN functions are already implemented.

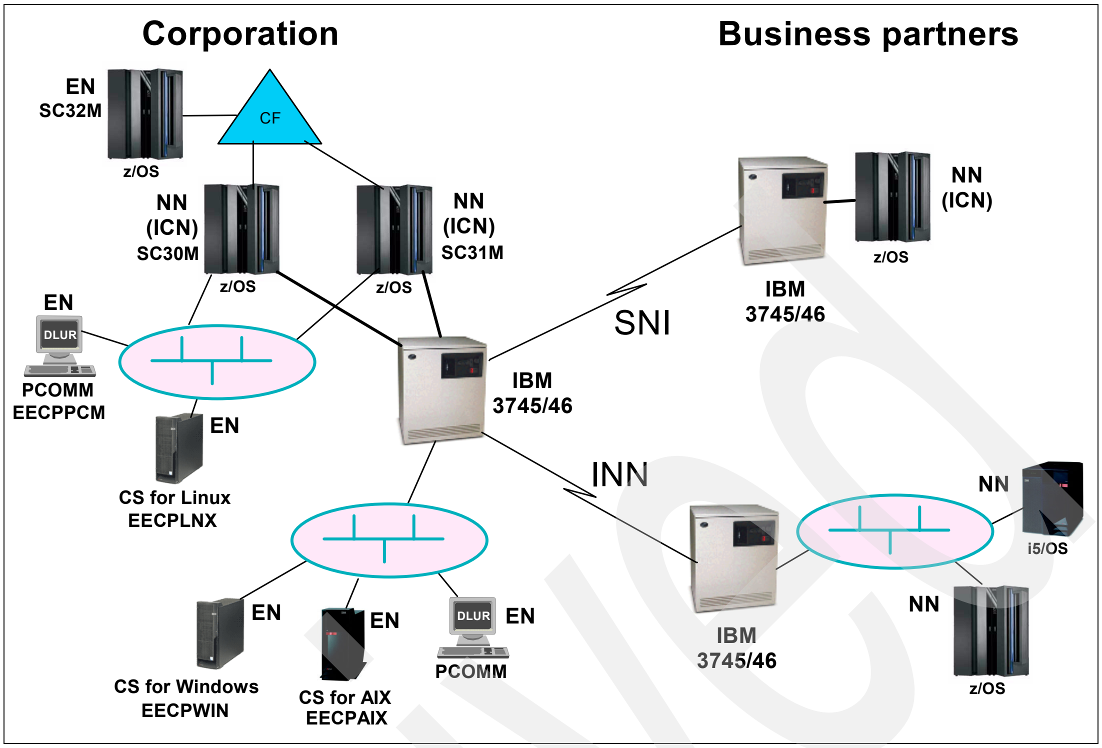


Figure 2-1 Determining the scope of the EE environment

As part of defining the scope, we identify which devices should participate in the EE network and to which systems they will connect. Tables in 2.3.1, "Identifying devices and connections" on page 52 list the devices used in our scenarios and the connections between the devices.

2.3.1 Identifying devices and connections

Now it is time to identify which devices should be converted to EE and what the possible implications are for those connections. We base this step on the environment shown in Figure 2-1.

First, list the devices that will migrate to EE as part of your project. Table 2-1 on page 53, shows a list of devices to be migrated to EE in our scenarios. It also summaries the SNA role and protocol capability for each device.

Note that the SNA protocols listed for z/OS Communications Server only show a subset of the protocols supported by VTAM. That is because, they are the only protocols supported by our hardware in our environment.

Table 2-1 Devices nominated for migration to EE

Node name	Software	Node type	Supported protocols
SC30M	z/OS V1R8.0 Communications Server	NN, EBN, Primary NNS	APPN, HPR, EE, XCF, MPC+
SC31M	z/OS V1R8.0 Communications Server	NN, EBN, Backup NNS	APPN, HPR, EE, XCF, MPC+
SC32M	z/OS V1R8.0 Communications Server	EN	APPN, HPR, EE, XCF, MPC+
EECPAIX	IBM Communications Server for AIX	EN, BEX	LLC2, APPN, HPR, EE
EECPLNX	IBM Communications Server for Linux	EN, BEX	LLC2, APPN, HPR, EE
EECPWIN	IBM Communications Server for Windows	EN, BEX	LLC2, APPN, HPR, EE
EECPPCM	IBM Personal Communications for Windows	EN	LLC2, APPN, HPR,EE

The capability of individual devices to support the EE DLC has been considered, but how these devices connect to their partners for SNA sessions must also be considered.

The Node and Partner Node entries in Table 2-2 show the connections we used in producing the scenarios in this book.

Table 2-2 Planned connections for EE devices

Node name	Node type	Connection description	Partner node	Node type	Supported protocols	Connection path
SC30M	NN	NN <-> NN	SC31M	NN	EE,XCF,MPC+	Direct
SC31M	NN	NN <-> NN	SC30M	NN	EE,XCF,MPC+	Direct
SC32M	EN	CP-CP to NNS	SC30M	NN	EE,XCF,MPC+	Direct
SC32M	EN	Backup NNS	SC31M	NN	EE,XCF,MPC+	Direct
SC32M	EN	Business partner application sessions	Other network	EN	EE	EBN
EECPAIX	EN	CP-CP for NNS	SC30M	NN	EE	Direct
EECPAIX	EN	Backup NNS	SC31M	NN	EE	Direct
EECPAIX	EN, DLUR	Mainframe application sessions	SC32M	EN	EE	Local CN
EECPLNX	EN	CP-CP for NNS	SC30M	NN	EE	Direct
EECPLNX	EN	Backup NNS	SC31M	NN	EE	Direct
EECPLNX	EN, DLUR	Mainframe application sessions	SC32M	EN	EE	VRN
EECPWIN	EN	CP-CP for NNS	SC30M	NN	EE	Direct
EECPWIN	EN	Backup NNS	SC31M	NN	EE	Direct
EECPWIN	EN, DLUR	Mainframe application sessions	SC32M	EN	CN	Local CN
EECPPCM	EN	CP-CP for NNS	SC30M	NN	EE	Direct
EECPPCM	EN	Backup NNS	SC31M	NN	EE	Direct
EECPPCM	EN, DLUR	Mainframe application sessions	SC32M	EN	CN	Local CN
EECPPCM	EN	AIX application access	EECPAIX	EN	EE	Local CN

As we look through the proposed connections, we conclude that EE can be implemented on all the devices and the partner endpoints in our environment. However, there may be APPN nodes that do not support EE in your configuration. In this case, a mix of Subarea SNA, APPN/HPR and EE connections is required. When planning for those connections, ensure that there are no more single-points of failure in your design than before the EE implementation.

2.3.2 LPAR-to-LPAR connections

The z/OS LPARs should be the first systems where EE is implemented. All End Node (EN) devices require a connection to a Network Node Server (NNS). The choice candidate for the NNS role is a z/OS system.

EE connections between LPARs are usually in addition to existing connections such as XCF for sysplex and MPC+. There may even be subarea connections between the LPARs.

We recommend establishing a connection in only one direction to set up the required CP-CP sessions. In an EE connection, we refer to the two connected nodes as the *local node* and the *remote node*. Remote nodes are defined on the local node by using a VTAM switched major node (SWNET). The EN should be the initiator of the connection to the NN Server (NNS). Assume the EN is the remote node for purposes of this discussion. A PATH statement is placed into the SWNET major node on the remote EN system. The PATH statement points to the local NNS system. The PATH information tells the remote EN how to access the local NNS. The PATH statement causes the remote EN to establish the connection to its NNS. Set the REDIAL parameter to FOREVER to ensure the connection to the NNS is established as soon as the NNS can be contacted. The SWNET major node on the local NN that defines the EN should *not* have a PATH statement.

Note: Avoid establishing a connection from both sides. Doing so introduces the possibility for a race condition that could hang the EE connection.

It is important to have reliable connections between Network Nodes. It is not uncommon to have fully meshed NN connections, allowing direct any-to-any communication. However, it is not necessary to fully mesh the NNs, but rather ensure that every NN can reach any other NN even if one of the links or NNs fails.

2.3.3 Non-VTAM application servers

There may be SNA applications running on servers other than z/OS and its predecessors. In some instances these applications are accessed from 3270 LU2 devices. Moving these servers to the EE environment may introduce availability issues when the clients are still in the subarea SNA environment. Some examples of these servers are:

- The System i™ servers can support SNA applications in most SNA formats as well as System i specific connections (5250). It operates as a PU 2.1 device. System i has only recently provided Enterprise Extender support, but has a long history of APPN implementation. APPN was implemented on the System i predecessors long before it was available with VTAM. You may also find the previous generation of AS/400® servers still around. AS/400 servers do not support EE.

Restriction: SNA Primary LU Services (SPLS) connections do not work over EE. Enterprise Extender only supports dependent LU sessions using DLUR, and DLUR does not support SPLS.

- System p servers are also capable of running SNA stacks and SNA applications.
- There are a number of midrange servers running SNA protocol stacks. A few support Enterprise Extender connections to the mainframe, often using a different name for the DLC.

  Some of these servers also support SPLS connections, so beware when migrating to EE as SPLS is not supported.

2.3.4 Peripheral devices

The peripheral devices migrating to EE are generally the simplest to plan. Each platform may have some mechanism to provide access through two LAN adapters. Subsequent chapters describe the implementation of EE on midrange and Intel® operating systems. There are descriptions of mechanisms that provide redundant LAN access for several IBM Communications server products.

Many, if not all, peripheral devices can be set up to automatically establish the connection to the its NNS and backup NNS. The number of retries can be limited or set to “forever”. Since the NNS and backup usually do not establish outbound connections to the peripheral device, it is advisable to configure the peripheral device to retry indefinitely (forever). If the EN cannot access the NNS, no sessions can be set up.

2.3.5 EBN to replace SNI

z/OS Communications Server is the only platform that can provide Session Services Extensions (SSE), which is required for EBN connections to other networks.

If the project uses EBN to replace SNI connections, the main consideration will be connectivity between the sites and security controls.

There are two user exits that provide security controls. The DSME user exit provides SNA security controls for all APPN sessions. The SME user exit provides security control for all subarea connections passing through the VTAM where the SME runs and also for all sessions with one endpoint in that VTAM.

The use of EBN requires a review of the security controls that are discussed in “Directory Services Management Exit (DSME)” on page 68.

2.4 Creating a strategy

A strategy is a high-level view of what you want to accomplish now and long term. A long term strategy may include plans outside the scope of the current project, but require consideration when developing the design for EE. We discussed the following steps:

- “Researching overall objectives of your networking environment”
- “Understanding the features and functions supporting EE”
- “Making strategic decisions”

2.4.1 Researching overall objectives of your networking environment

Determine the requirements of other networking strategies in the organization by investigating the following questions:

When developing the strategy, consider the following questions:

- Are there other components in the network that might benefit from an EE implementation?
  - For performance reasons
  - For availability reasons
  - Other platforms in the environment that support EE
- What functions of EE could enhance your environment that are not part of the current objectives or scope?
  - Automatic failover
  - Simple IP transport as opposed to protocol encapsulation
- Are there other ongoing activities in your environment that could be assisted by this or future EE implementations?
  - 3745 consolidation or removal
  - Avoidance of future costs relating to:
    - DLSw software
    - NCP

- How does EE fit in with other networking strategies in the organization?
  - Voice over IP
  - Migrating applications to native IP
  - Retaining existing SNA applications
  - Mergers or acquisitions of other companies and their network protocols

2.4.2 Understanding the features and functions supporting EE

Because Enterprise Extender is an extension of APPN/HPR and uses the TCP/IP stack, there are several important features and functions that you need to understand and consider while developing a strategy for EE. In this section we discuss these functional considerations, as follows:

- VTAM considerations
- SNA routing considerations
- APPN/HPR considerations
- Subarea SNA considerations
- TCP/IP (stack) considerations
- IP network considerations

VTAM considerations

Additional definitions are needed in the APPN-enabled VTAM for EE. The definitions are in the following distinct areas:

- Start options
- XCA major node
- Model major nodes
- Switched major nodes
- Static VIPAs
- Parallel TGs and multiple VIPAs
- Dedicated TG number for EE
- Class of Service (COS) and Type of Service (TOS)

Start options

There are start options applicable only to EE and others that deserve mentioning, because they have an influence on the EE environment. For example:

- IPADDR or HOSTNAME defines the IP static VIPA addresses used by EE. These parameters may also be defined in the EE XCA GROUP statement. If you plan to use EE across NATed firewalls with a connection network, then the use of HOSTNAMES is mandatory.
- TCPNAME specifies the started task name of the TCP/IP stack VTAM will use for EE connections. Even though z/OS can support multiple TCP/IP stacks, VTAM can only connect to a single stack.
- T1BUF, T2BUF, and TIBUF are buffer pools for EE. You should monitor these buffer pools and code their settings with values that minimize buffer pool expansion.
- HPRPST should be reviewed when implementing EE. HPRPST specifies the maximum time VTAM will wait for a path switch of an RTP pipe to complete.

Tip: Use HPRPST default values to ensure time-out values are consistent within your network and increase the probability they match the values in any partner networks with which you have connections.

- UNRCHTIM sets a time period during which a connection network is deemed to be unavailable after a failure on the connection network is reported to VTAM. No session setup or path switches will be tried across the connection network towards a specific destination node until the UNRCHTIM timer has popped.

XCA major node

It is generally expected that the main access from the mainframe to the LAN is via OSA ports. With EE it is expected that the OSA connection is high-speed Ethernet at 100 Mbps or faster.

To communicate with the TCP/IP stack, VTAM uses an XCA major node defined specifically for EE. The main considerations for the XCA major node are:

- There is only one XCA PORT active on z/OS Communications Server with MEDIUM=HPRIP.
  - Do NOT change the IPPORT and IPTOS values from the default unless you have a very good reason. A further discussion of IPPORT and IPTOS is in "Class of Service (COS) and Type of Service (TOS)" on page 61.
- There can be multiple groups under the XCA PORT, with each group defining a separate type of connection.
- Each Group defines:
  - One IP Address (IPADDR) or host name (HOSTNAME). IPADDR or HOSTNAME must refer to a static VIPA.
  - One Local Connection Network (LCN)) or one Global Connection Network (GCN) VRN name
  - A number of lines defined for each GROUP.
  - Whether inbound or outbound connections are permitted or both.
  - Dynamic PU definition capability is controlled at XCA GROUP level.

Attention: The DYNPU parameter of the XCA GROUP is ignored for connections coming in over a connection network. Dynamic PU definition is always permitted for these connections

How many groups do you need? A basic configuration would consist of one static VIPA and one local connection network and would allow CALL=INOUT. This requires only one group. If you require two VIPAs for multi-VIPA support, then there must be two GROUPs. If its two static VIPAs, one local connection network and one global connection network for each, then you need between two and four GROUPs. Add the CALL=IN and CALL=OUT options and the number of GROUPs can increase rapidly.

See 4.4.5, "Summary of VTAM major nodes" on page 145 for the XCA major node used in our scenarios.

Model major nodes

The model major node is very useful in an EE environment. If remote devices are connecting via a subarea connection, there is no CP-CP session set up. However converting to EE requires a CP-CP session and a DLUS/DLUR connection for dependent PU 2.0 device types. The original PU 2.0 switched major node will continue to work in the EE environment, but first it needs a CP-CP session to be established, using the following parameters:

- **DYNTYPE=EE**

  The model major node with DYNTYPE=EE defines a switched PU that has characteristics suitable for CP-CP sessions. As all of the PU 2.0 PU already have switched PU definitions the model major node does not impact them.

  The model major node for EE is used when DYNPU=YES is coded on the EE XCA major node and there is no predefined switched PU matching the incoming XID.

  Useful statements for the EE model major node type are:

  – DISCNT=NO to keep links up even when there is no HPR pipe over them. If you leave DISCNT=YES (default value), the stopping and restarting every couple of seconds as the remote end tries to recover the link will generate unnecessary processing and writes many messages to the network log.

  – TGP=*profile name* if you want to specify a specific TG profile.

  – CPCP=YES

  – DYNLU=YES if you want to allow dynamic ILU definition and the DYNLU start option is set to NO.

- **DYNTYPE=RTP**

  The model major node with DYNTYPE=RTP allows you to dynamically define a PU for the RTP pipe that has characteristics suitable for LU-LU sessions.

  Useful statements for the RTP model major node type are:

  – DISCNT=NO to keep RTP pipes up even when there is no session using them. If you leave DISCNT=YES (default value), the stopping and restarting of the RTP pipes every couple of seconds will generate unnecessary processing and writes many messages to the network log.

  – DYNLU=YES if you want to allow dynamic ILU and CP definition and the DYNLU start option is set to NO.

- **DYNTYPE=VN**

  The model major node with DYNTYPE=VN allows you to dynamically define a PU for connection requests coming in over a virtual node (connection network) that has characteristics suitable for EE sessions.

  Useful statements for the VN model major node type are:

  – DISCNT=NO to keep links up even when there is no HPR pipe over them. If you leave DISCNT=YES (default value), the links stopping and restarting every couple of seconds will generate unnecessary processing and writes many messages to the network log.

  – DYNLU=YES if you want to allow dynamic ILU and CP definition and the DYNLU start option is set to NO.

Switched major nodes

Remote devices usually have PU 2.0 definitions in VTAMLST and the model major node can provide support for the new CP-CP sessions required by peripheral devices. However, there are some connections that require switched major nodes with PATH statements using the REDIAL parameter. These are the following:

- NNs, for connections to their NN counterparts
- z/OS ENs, for their connection to their NNS and backup NNS.

The one thing these connections have in common is to establish an outbound connect to their partners. The PATH statement in the switched PU definition links the IP address of the partner system and the GROUP name in the EE XCA major node.

For devices other than z/OS LPARs, the link should be initiated by the peripheral device. If you do not want to permit dynamic CP definitions for the peripheral devices, you will have to define a switched PU for every EE node.

Useful statements on the switched PU are similar to those on the EE model major node type and include:

- DISCNT=NO to keep links up even when there is no HPR pipe over them. If you leave DISCNT=YES (default value), the links starting and stopping will generate unnecessary processing and many messages in the network log.
- TGP=*profile name* if you want to specify a specific TG profile.
- CPCP=YES
- DYNLU=YES if you want to allow dynamic ILU definition and you have DYNLU=NO in the VTAM stat options.

See 4.4.5, “Summary of VTAM major nodes” on page 145 for the switched major node definitions used in our scenarios.

Static VIPAs

EE requires one or more static VIPAs to be configured in the TCP/IP stack. We recommend that you allocate dedicated VIPAs to VTAM for its EE connections. The use of a dedicated VIPA allows for easier tracing and problem determination.

If there are firewalls between potential Connection Network partners and those firewalls perform a NAT function, HOSTNAMES must be used at each end to allow VTAM to correctly resolve addresses when NAT is used.

See 4.3.1, “Modifications to the TCP/IP PROFILE” on page 130 for the configuration of the single static EE VIPA scenario.

See 10.3, “Configuring multiple EE VIPAs” on page 328 for the configuration of the multiple static EE VIPA scenario.

Parallel TGs and multiple VIPAs

As of z/OS V1R8.0 Communications Server VTAM no longer initiates parallel EE sessions (TGs) between the same IP address pair using multiple SAPs. Multiple SAPs are no longer allowed between the same IP address pair.

Parallel EE sessions (TGs) can be configured by defining more than one static VIPA for VTAM. As VTAM then has more than one IP address, it can establish an EE link (TG) using each of the VIPAs. Multiple VIPAs are configured in separate EE XCA GROUP statements.

Multiple VIPAs can be set up so that they are in different subnets from the network perspective. This allows each VIPA’s address to be routed differently in the network. They can be set to allow different hardware to be used on each path, possibly providing higher availability.

Dedicated TG number for EE

To assist in problem determination it is useful to have all EE connections use a predefined TG number when setting up connections. When looking at topology it becomes immediately obvious what function a TG performs. The TG number can be defined on a switched PU definition for predefined definitions. For dynamically defined PUs, VTAM cannot specify the TG number, this must be done in the configuration of the peripheral device.

Class of Service (COS) and Type of Service (TOS)

Enterprise Extender uses five UDP ports to map APPN COS. In addition to the four COS levels for APPN, a fifth has been added for EE LDLC signalling.

Table 2-3 EE priorities

Priority	UDP Port	TOS	Example COS using this priority	Common Use
LDLC Signalling	12000 or x’2EE0’	x’C0 ’	n/a	EE only
Network	12001 or x’2EE1’	x’C0 ’	CPSVCMG	APPN environment
High	12002 or x’2EE2’	x’80’	#INTER	Interactive
Medium	12003 or x’2EE3’	x’40’	#CONNECT	Client-server
Low	12004 or x’2EE4’	x’20’	#BATCH	Batch

The default values set by VTAM and TCP/IP help distinguish the type of traffic being sent. However, in many sites, network support (meaning switch and router designers) will not accept a TOS value set by a “host” system. You may have to negotiate with the network designers to allow either the TOS value to be retained and mapped to the corporate QoS or have the network designers map the UDP ports to a suitable QoS value.

Attention: Even though it is possible to change the UDP ports and TOS values, we do not recommend doing so unless there is a very good reason. Enterprise Extender is a standard created by the Internet Engineering Task Force (IETF) and APPN Implementers Workshop (AIW). It is documented in RFC 2353. The UDP ports are Registered Ports assigned by the Internet Assigned Numbers Authority (IANA).

SNA routing considerations

When EE is introduced into a network, there is a possibility that the SNA routing topology may change. In this section we discuss the following related topics:

- VTAM as an SNA router
- Geography
- Capacity and performance
- Hardware and software

VTAM as an SNA router

In subarea or non-EE APPN environments, NCPs provide a high percentage of SNA routing. VTAM usually performs very little SNA routing. A data session path for an LU2 from a workstation to an application would be:

Workstation <-> NCP <-> NCP <-> **Application,** as shown in Figure 2-2 on page 62.

There is no VTAM in the routing path once the LU to LU session is set up. If the ICN is brought down for any reason, the data session continues (always for ILU and if ANS=CONT is coded for a dependent LU). The application could be running on a z/OS LPAR or it could be running on a midrange server.

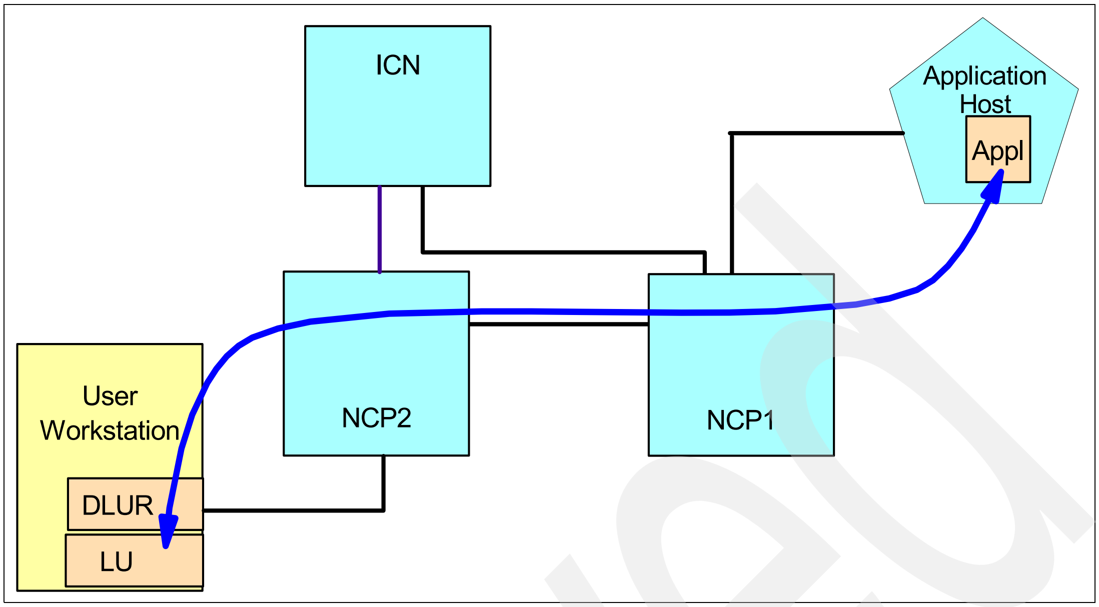


Figure 2-2 LU-LU data session path to application

If EE is implemented on the user workstation, the route to the server becomes more complex. Then the path would look like **Workstation** <-> NNS/DLUS <-> NCP <-> **Application**, as shown in Figure 2-3.

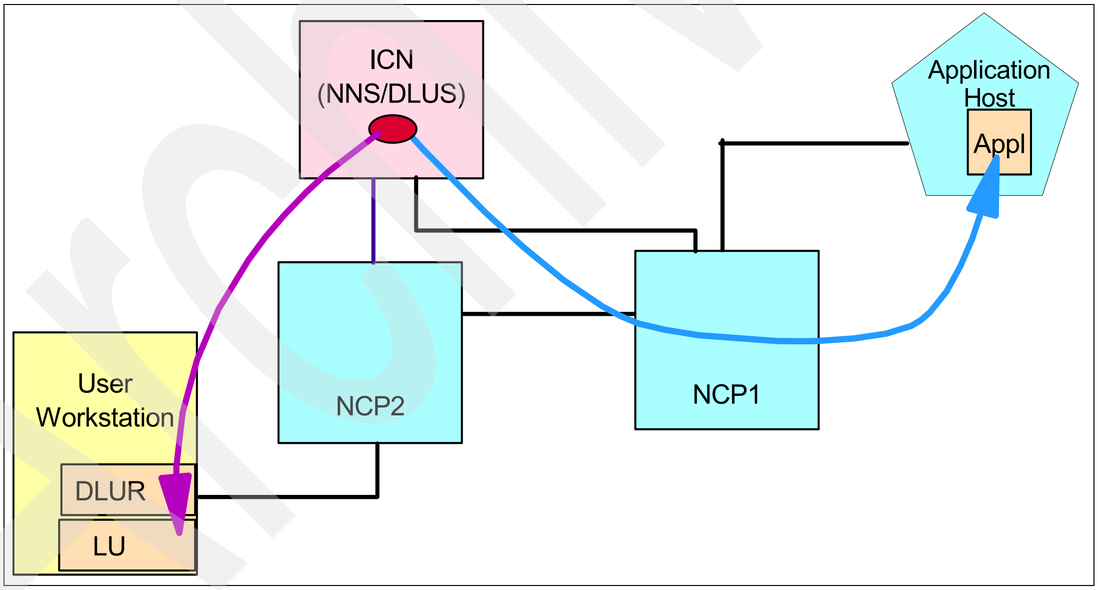


Figure 2-3 LU-LU data session path to application with EE

In this scenario, the ICN now becomes an SNA router for all data sessions from the workstation to the application on the server. If the ICN goes down for any reason, the data session path is broken. It can only be re-established if there is another ICN to take over ownership of the NCPs and provide session initiation. This is an undesirable configuration.

When planning an EE implementation it is important to consider the path that sessions will take. It may be that both ends of the session need to be convert to EE. If both session partners are EE enabled, then a connection network can be set up between the partners, eliminating the need for VTAM to provide routing services.

Geography

When EE is introduced, the geographic reach of the APPN environment can greatly increase. The IP network is usually much wider than the SNA network. DLSw has often been used to provide IP connectivity, but it required hardware and specific connections to carry SNA. Even then there are limitations in DLSw's capability to support HPR. EE provides reach to the entire IP network so that APPN connections that were once difficult or impossible to set up are now much easier.

Capacity and performance

The impact of EE on the mainframes can be quite varied depending upon your network environment.

- For existing APPN/HPR connections the changes in resource utilization will be very small. In fact under some conditions, EE will use less resources than conventional HPR. Between LPARs sharing the same physical hardware, HiperSockets (using iQDIO) will reduce processor utilization. The OSA (using QDIO) also improves VTAM to LAN performance.
- Response times and throughput may improve, especially within the data center. This is due to:
  - The faster speeds of the OSA adapters compared to ESCON® or parallel channels for IBM 3745.
  - LAN switching in the network is much faster than the processing associated with IBM 3745 (NCP) or DLSw.

> **Tip:** Some peripheral device configuration definitions may need to be reviewed in light of faster throughput.

- If devices are moving from Subarea SNA to EE, then there may be an increase in VTAM cycles. VTAM has to perform the boundary function previously performed by NCP. In addition, there is HPR support that was not available under Subarea SNA.

Hardware and software

Enterprise Extender requires some form of IP network attachment. Any of the many connection methods supported by Communications Server for z/OS is acceptable. However, we recommend using OSA ports and HiperSockets wherever possible.

All currently supported releases of z/OS Communications Server provide Enterprise Extender support.

Some of the features described in this book were not provided in z/OS Communications Server releases before V1.8.

APPN/HPR considerations

The implementation of EE may have a significant impact on the way your APPN environment operates. EE capability on the mainframes can permit the migration of a large number of devices from the Subarea SNA environment to an APPN environment. This will impact VTAM options such as SORDER, ALSLIST, APPNCOS, and buffers.

The implementation of EE allows you to exploit APPN/HPR features that you may not have used before.

If you are not familiar with the APPN environment, you can review the APPN sections of Chapter 1, "Introduction to APPN and Enterprise Extender" on page 1.

Network ownership and topology

In the Subarea SNA environments it was common to have a Communications Management Configuration (CMC) and a backup CMC to run the network. When the production CMC was down, the network was transferred to the backup CMC, often by a lengthy, manual process.

In an EE network there is generally one VTAM that provides NNS and DLUS services and another VTAM that can act as backup. These VTAMs can be considered as a CMC and backup CMC. The main difference between EE CMCs and Subarea SNA CMCs is that the switch to the backup CMC can be automatic for the EE components in the network. The switch is no longer performed from the VTAM side, but is performed when the peripheral device decides to switch DLUS. This results in faster network recovery for those devices and therefore improved network availability.

Important: LU-LU sessions can be non-disruptively switched provided the device supports ACTPU(ERP), ACTLU(ERP) and ANS=CONT is coded on the switched PU supporting the dependent LUs.

One other advantage offered by EE is that the backup CMC can now be anywhere on the IP network, even at your Disaster Recovery (DR) site. That means there can be automatic *network* disaster recovery for EE components.

Dynamics for CP-CP sessions and LU-LU sessions

The subarea SNA environment has traditionally been a very hierarchical environment. Everything was predefined. As VTAM's subarea SNA environments evolved, the capability for dynamic definitions was provided in line with the ever increasing numbers of SNA devices. A widely used example of this is dynamic CDRSCs.

APPN introduced a more dynamic environment. Paths between devices are dynamically calculated. The link definitions defined between nodes determined what paths could be used rather than the centrally generated path statements of subarea SNA. The mainframe VTAM still selected which path to take but it was often based upon information provided by the peripheral devices.

Additional dynamic capabilities for VTAM have been introduced to simplify the definition of devices. These include model PU and LU definition, dynamic PU definition, dynamic LU definition and dynamic adjacent CP definition.

Enterprise Extender *requires at least one pair of CP-CP session for each node.* If you are introducing a large number of EE connections, then expect similar growth in the number of CP-CP sessions and therefore an increase in the number of switched PU definitions. The existing definitions for dependent PUs and LUs should already be in place in your network. If the DLUS function remains on the same CMC, then the dependent PU and LU are unlikely to need any change.

The definitions for LEN based independent LUs *will* change after migration to EE. For this reason, the ILU will now be associated with the PUname of the switched PU used for the CP-CP connection rather than the PUname of the old PU 2.1 used by LEN.

You can review the dynamics used in your network as part of the EE project. The decision to provide dynamic capabilities or not is often based upon historical implementations or on a need to control the entire network environment.

Using dynamic definitions

VTAM's dynamic capabilities enable creation of dynamic VTAM definitions for PUs, LUs, CPs and Adjacent CPs. The CPs are, in fact, a special case of LU definitions. Creating the capability to use dynamic CP definitions will also allow dynamic definition for the following:

- Independent LUs
- CDRSCs
- LEN and other APPN PU 2.1s

Dynamic CP definitions can be configured using a Model Major Node. The use of a Model Major Node greatly simplifies the configuration process. It allows key values such as DISCNT and TGP to be inherited by dynamically defined CPs.

To dynamically create non-connection network dynamic CPs, it is necessary to

- Define DYNPU=YES on the EE XCA GROUP statement.
- Set DYNLU=YES in ATCSTRnn, or code DYNLU=YES in the model PU for DYNTYPE=EE or provide a model CDRSC definition.
  - CDRDYN=YES must be coded if DYNLU=YES is desired.
- Set DYNADJCP=YES in the VTAM start options.

Should you want to require predefined CP and switched PU definitions for some users and allow dynamic CP and PU definition for others, different groups may be set up in the EE XCA major node, each with its own static VIPA and DYNPU value.

Controlled static definitions

If you want to control the definition of EE CPs and ILUs, then predefined definitions are required. For EE, when the EE XCA major node has DYNPU=NO for all groups, there can be no dynamic definition, regardless of the value of other settings.

- To ensure only static definitions of non-connection network EE CPs or PUs, it is only necessary to
  - Define DYNPU=NO on *all* of the EE XCA GROUP statements.

    Code a switched PU with a CPNAME matching the EE node's CPNAME and NETID.

Important: The EE XCA Group allows dynamic PU definitions for CPs and PUs coming in via the Virtual Routing Node (connection network), even if you code DYNPU=NO on the XCA definition.

- To ensure all CPs and ILUs must be defined:
  - Set DYNLU=NO in ATCSTRnn or code DYNLU=NO in the PU used by the CP, that is, the PU statement with the CPNAME coded.
- To ensure that ILUs are associated only with predefined PUs for the Control Points code:
  - ALSREQ=YES in ATCSTRxx start options. ILUs may be associated with more than one specified PU or may be authorized to any PU in the APPN environment. This is a global parameter and affects all ILUs.
  - Code ALSREQ=NO in the VTAM start options and code ALSREQ=YES on the CDRSC definitions for specific ILUs you want to control.
- You may also choose to set DYNADJCP=NO in the VTAM start options, depending upon the level of control that you want to enforce. In this case all CPs must be predefined in adjacent CP table.
- Do not define connection networks on the End Nodes if you want to ensure there will be no dynamic links created to these nodes.

When choosing static definitions only, the following must be coded to allow CP and ILU sessions:

- A switched PU specifying the CPNAME of your connection
- A CDRSC for all ILUs using that CP. Specify all valid PUs of CPs that can own the ILU using the ALSLIST parameter.

VTAM start options

The VTAM start options for APPN should be reviewed when planning the EE implementation. The number of new APPN/HPR devices could require a change in existing APPN start option values, for the following:

- **SORDER**

  In a mixed subarea and APPN environment there are two networks to search whenever VTAM tries to locate a resource. The SORDER parameter tells VTAM how to conduct this search. Usually, the SORDER option is set based upon percentage of the LUs that are to be found in subarea compared to the percentage to be found in APPN. When EE is introduced, balance between subarea and APPN may dramatically change. If this is likely to happen, SORDER can be changed from SUBAREA to APPNFRST.

- **DYNADJCP**

  A large number of EE devices will result in the same number of CP-CP sessions. For this reason, you should consider allowing dynamic adjacent CP definitions in lieu of the configuration effort required. DYNADJCP defaults to YES.

Attention: To permit dynamic adjacent CP definitions, set CDRDYN=YES in the VTAM start options.

- **DYNLU**

  The DYNLU start option determines whether CDRSCs can be dynamically created on this VTAM for Independent LUs. If set to YES, VTAM will accept sessions from undefined ILUs. The DYNLU start option can be overridden by the DYNLU parameter on PU and ADJCP definition statements.

  If DYNLU=YES is desired then CDRDYN=YES must also be coded. DYNLU defaults to NO.

Logmodes and APPN COS

The introduction of Enterprise Extender increases the importance of SNA COS for LU-LU sessions. SNA COS was observed over IBM 3745 serial lines and mainframe OSA. However, many sessions were over NCP token-ring or DLSw over the wide area network. Neither of these had separate queues to prioritize traffic by SNA COS. The introduction of EE enhances the network's capability to prioritize SNA traffic according to SNA COS. This prioritization can be implemented at both APPN and IP levels for connections that are over both the LAN and the WAN.

EE has enhanced the capability to provide COS prioritization over the entire network. Therefore, it would be worthwhile to review the APPNCOS entries in the logmode tables to ensure all logmodes include APPNCOS and that applications are using the appropriate logmode. See 3.4, “Consolidating LOGMODEs” on page 114 for a discussion of logmodes, logmode tables and APPNCOS.

XCF, MPC+, and EE

Inter-LPAR communication can generally have three paths within a sysplex, XCF, MPC+ and EE and all three are frequently defined. By assigning each type of connection a different TG number it is easier to see which paths sessions are taking.

It is also possible to assign different classes of service to different logmodes. You may want to use XCF for interactive traffic, MPC+ for batch traffic and EE for everything else. See 12.1, “APPN route selection” on page 376 for a discussion on logmodes and TG weighting.

Connection Network

Provision of a connection network greatly simplifies the configuration of devices in the APPN EE network and ensures that VTAM does not act as an SNA router any more than absolutely necessary. The connection network is described in “APPN connection networks” on page 38 and “Virtual Routing Node (VRN)” on page 22.

Defining a connection network allows for fewer definitions on the End Node. For an End Node only three links need be configured:

1. EN to primary Network Node Server
2. EN to backup Network Node Server
3. EN to the VRN or connection network.

All other connections with other members of the connection network will be dynamically created.

There are two types of connection networks:

1. Local connection networks within one NETID
2. Global connection networks that span multiple NETIDs

It is possible to have more than one local or global connection network, but only one connection network can be defined in each group statement.

Capacity and Performance

Depending upon the number of new CP-CP sessions and HPR pipes for application access, there will be an increase in APPN/HPR storage usage. VTAM’s buffer utilization should be reviewed if that is not already an ongoing process. For example, ECSA, CSM, VTAM buffers.

Security through VTAM definitions

In an APPN environment it is still very common for third party connections to be run using SNI. It is possible to control which of your applications are accessible to SNI partners by coding only GWNAU elements in the NCP. This control does not have any effect for APPN sessions of any kind.

Session Management Exit (SME)

The SME moves the control from the NCP to VTAM and increases the span of control to all session requests passing through the VTAM running the SME. The SME is usually run on the CMC and backup CMC of a network. It can also be run on data hosts but in that case, controls only those sessions terminating on the data host.

APPN/HPR has different requirements from Subarea SNA for security controls. For an SNI connection, it is adequate to run a VTAM Session Management Exit (SME) to control session setup between LUs. The SME is traditionally run on the SNI Gateway VTAM. However, the SME on the SNI gateway VTAM may not have visibility of all session setup requests flowing over an HPR pipe. This could allow sessions to be set up without the SME session authorization.

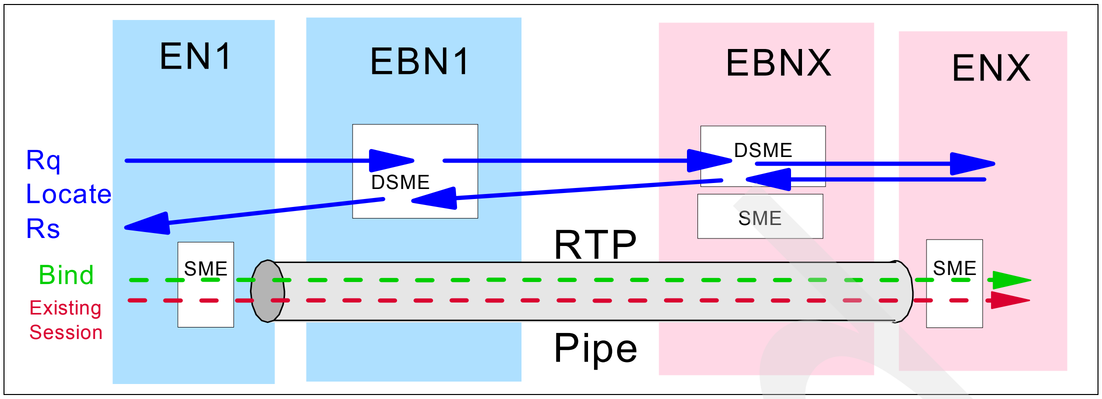


Figure 2-4 Locate and Bind processing for APPN/HPR

As shown in Figure 2-4, the SME cannot always control APPN/HPR sessions. The SME gets control during both the session initiation request and again at the session initiation response. In an APPN/HPR environment, the LOCATE is sent from the requestor (EN1) to its NNS-EBN (EBN1) and then to the partner EBN (EBNX) that is running an SME. EBNX sends the Locate response t back to the originating host (EN1) with path information. If there is an RTP pipe between host EN1 and your application host (ENX), the BIND will pass through your EBN CMC using ANR routing. The SME cannot see the BIND since ANR traffic is not visible in VTAM.

Tip: The SME on the destination node can allow or reject the session.

Directory Services Management Exit (DSME)

Another exit is available that will allow security checking in the APPN/HPR environment, VTAM can use a Directory Services Management Exit (DSME). A DSME has two functions of interest. As its name implies, the DSME can control directory searches within the APPN environment. It is also capable of providing session authorization for APPN sessions. You can write a DSME to provide the level of control you require in a similar manner to the SME.

The DSME receives control when VTAM receives a LOCATE request from another system or when a locate request is sent from VTAM. The DSME can examine the names of the OLU and DLU and determine whether to allow the search to continue. Because the DSME performs its test before the requesting node has enough information to send a bind, ANR routing cannot be used to bypass controls.

Tip: The RTPONLY parameter stops ANR routing from occurring through a Network Node for a specific Adjacent CP. This can be useful if the node is an Extended Border Node (EBN). When RTPONLY=YES is coded, there will be no ANR traffic from the Adjacent CP allowed through the EBN. The EBN will then have session awareness of all sessions, allowing the SME to control sessions in the same manner as was the case for SNI.

IPSEC

Since Enterprise Extender uses UDP/IP data flows, EE can make use of IPSEC functions for SNA traffic. Two key functions provided by IPSEC are:

1. Authentication
2. Encryption

These are important words when talking to security people and may assist you in integrating the SNA traffic into the IP world. In some environments, there may be a requirement to encrypt all traffic between security zones. IPSEC can provide this function to a degree acceptable to the security experts. In other cases there may be a requirement to authenticate between two ends of a connection. IPSEC also provides this capability in a manner that security practictioners will understand and accept.

Encrypted data

If there is a requirement for data encryption, EE connections can use two options:

1. IPSEC provides encryption and will encrypt all data on an EE *connection*. Generally that means all data between two host systems is encrypted.
2. SNA Session Level Encryption (SLE) is provided by the z/OS Cryptography Facility. Logmodes can be set up to require encryption so any application using those logmodes would encrypt session data. This allows data to and from specific *applications* to be encrypted rather than the connection level encryption provided by IPSEC.

Subarea SNA considerations

It is likely there will be some subarea SNA connections remaining in your network. The most likely subarea SNA functions will be:

- SNI
- PU type 2 support

SNI

In many networks the last of the subarea SNA functions to convert to APPN is SNI to business partners. EE simplifies the conversion from SNI to APPN EBN connections, requiring only that both partners implement EE to leverage IP connectivity.

The EBN connection to a business partner requires some form of SNA security. This has commonly been provided for SNI connections by the VTAM session management exit. The SME however may not be able to control APPN sessions that are started over an existing RTP pipe.

An SME can continue to control sessions provided RTPONLY is coded on the ADJCP definition statement of the Adjacent CP Major Node (VBUILD=ADJCP) of your border nodes.

The impact of RTPONLY=YES on an EBN is to provide session awareness for all sessions set up between the networks. No HPR pipe can be established that bypasses the EBN.

The other alternative is to implement VTAM’s DSME function that can provide session establishment control and also allow for RTP path switching.

PU type 2 support

The best aspect of PU 2.0 support in an EE APPN/HPR is that the definitions do not change. Since EE is an APPN/HPR function you must use DLUS/DLUR support for PU 2.0 functions, but actual PU 2.0 Switched Major Node members are unchanged.

TCP/IP (stack) considerations

Remember, EE always establishes affinity with only one TCP/IP stack. All local EE static VIPA addresses must be associated with a single TCP/IP stack, even in common-INET environments. Other considerations include:

- MULTIPATH
- Parallel OSA paths
- MTU size
- Dynamic routing protocol
- IUTSAMEH

MULTIPATH

Enabling IP MULTIPATH for TCP/IP applications frequently has benefits for TCP/IP performance when more than one LAN interface is available. However, IP MULTIPATH, either per connection or per packet, can have an adverse impact on EE traffic.

Since multipath is a global TCP/IP parameter, consider all aspects of the IP network before deciding whether MULTIPATH is used for EE or not. If MULTIPATH per PACKET provides good performance for the TCP/IP applications, then it is likely EE will also operate well.

For UDP applications, specifying MULTIPATH per CONNECTION cannot provide a *per connection* path since UDP is connectionless. The IP stack therefore sends groups of UDP packets on each of the equally weighted interfaces using a round robin algorithm.

If the two paths are slightly different in performance (for example, due to congestion), then it is quite possible the EE packets will arrive out of order. When this occurs, the receiving RTP will use additional CPU processing to queue and reorder the packets. If the delay between paths is too long RTP may even have to request retransmission of the NLPs causing a slowdown of the allowed ARB send rate.

Within a data center this will not occur frequently if both of the paths are similar in speed and configuration and the paths are unlikely to be slowed down due to congestion. Outside the data center it may be that parallel paths are not of the same delay.

If MULTIPATH per CONNECTION is run to support TCP/IP applications and you want to run EE without MULTIPATH, then a second TCP/IP stack running without MULTIPATH is a possible solution for EE and any other UDP applications.

Parallel OSA paths

It is recommended that at least two OSA ports be available for each LPAR running EE. Each OSA port should be on a separate physical OSA feature and each of those OSA ports be connected to a separate Ethernet switch. This provides an alternate path in the event of OSA port or switch hardware failure.

To make use of the two interfaces to the network a dynamic routing protocol is required. If static routes are used, multiple default_routes may be specified, but path protection is only available in the case of OSA failure. Failure of downstream components in the path will not be known to the TCP/IP stack and packets sent over that route will be lost. If dynamic routing is used, the entire route is checked for availability. One interface is used as the primary route but the second OSA path is available for backup should any component in the route fail.

MTU size

The selection of the correct MTU size for all interfaces is critical to the performance of any IP application, including Enterprise Extender. MTU is defined in OMPRoute and must be defined on *all* interfaces using the OMPRoute Interface, OSPF_Interface and RIP_Interface statements. Failing to define an interface to OMPRoute will result in the default MTU of 576 being used. The default value can significantly reduce throughput.

> **Important:** Note that an OSA-Express or OSA-Express2 Ethernet port has an MTU of 1492, by default.

Dynamic routing protocol

A dynamic routing protocol is recommended to support EE. The TCP/IP stack provides the application OMPRoute for dynamic routing. OMPRoute supports OSPF V2 and RIP V2. OSPF is more commonly used than RIP and is used in our scenarios.

When a VIPA is defined, a host address and a subnet are configured. OSPF can broadcast either or both of these values. It is recommended that OSPF broadcast only the host routes of EE VIPAs to the network. Setting the OSPF_INTERFACE parameter for each static VIPA to ADVERTISE_VIPA_ROUTES=HOST_ONLY ensures no subnet routes will be broadcast. Earlier releases of z/OS Communications Server used the parameter SUBNET=NOVIPASUBNET to achieve this.

Presenting only host routes to the network allows VIPA addresses to be assigned from a dedicated "VIPA" pool or subnet whose routing is not summarized anywhere in the network. This enables the VIPAs to be moved to other LPARs in your network, even to a different data center. In this way, network failover to an alternate site can be a standard, automatic procedure.

IUTSAMEH

IUTSAMEH, or "IUTSAME HOST", is used to provide the "communications link" between the TCP/IP stack used for EE and VTAM. There are two ways IUTSAMEH can be configured:

1. If DYNAMICXCF is configured for the TCP/IP stack, IUTSAMEH will be dynamically created and can be used by EE.

> **Tip:** If dynamic XCF *is* configured, it is recommended that no static IUTSAMEH be defined.

2. If DYNAMICXCF is not configured, then you must manually define the IUTSAMEH device and its associated link and then start the IUTSAMEH device.

> **Note:** Dynamic XCF can be specified for TCP/IP even if the LPAR is not part of a Sysplex. In this case, no XCF connections are be defined, but IUTSAMEH is dynamically created.

IP network considerations

Some functions provided by Enterprise Extender must be reflected in the rest of the network. If they are not, EE may perform poorly. The functions include:

- Quality of Service (QoS) in the IP network
- IP routing for SNA services
- IP security considerations

Quality of Service (QoS) in the IP network

EE is capable of setting the Type of Service (TOS) or Differentiated Services bits in the IP header. There are two points to consider concerning the QoS in your organization.

1. Is there QoS or prioritization available in your network?
   a. If QoS is available, work with the network designers to ensure EE traffic is considered in the network configuration and prioritization.
   b. If WAN router prioritization is available, work with your network designers to ensure EE is assigned an appropriate priority. Sometimes, EE will replace DLSw, but can be set more granularly than DLSw.
   c. If not, then care must be taken concerning EE access points and router paths.
2. Will the TOS or DiffServ settings set by TCP/IP be recognized by the network switches and routers?
   a. Frequently switches are set to discard any TOS or DiffServ settings sent by any application host. QoS marking is then provided by the network device.
      i. The network device may accept TOS or DSCP markings from some *trusted* hosts.
      ii. Work with the network people to have z/OS Communications Server added to the trusted server list.

b. QoS may be allocated by the switches based upon UDP port number. The mapping are usually organization specific, but some guidelines would be
   i. UDP ports 12000-12001 have the highest data priority available. They are signalling functions comparable to Voice over IP signalling and dynamic routing protocols.
   ii. UDP port 12002 is *SNA High Priority* for interactive sessions and requires a QoS comparable to that of Telnet or other response time sensitive applications.
   iii. UDP port 12003 is *SNA Medium Priority* and is often used for client server type sessions and should have a QoS comparable to HTTP or email. Response time is not as sensitive as 12002, but should not be relegated to the lowest priority.
   iv. UDP port 12004 is *SNA Low Priority* for batch sessions and should have a QoS comparable to FTP sessions.

IP routing for SNA services

The configuration of the mainframes for high availability is sometimes at odds with the normal network practices for servers. The mechanism z/OS Communications Server uses to provide redundant routing paths is the use of a dynamic routing protocol. Mainframes are probably the only servers in the network to actively use OSPF or RIP. Therefore expect to spend some time working with your network designers to get this implemented. They may consider the idea unusual and may take considerable persuasion to implement such a solution.

For other types of servers, it may be possible to use high availability function now that the SNA connection uses IP for transport. Consider the operating systems' implementation of high availability using multiple network adapters. There are many different schemes and most are platform specific. Most did not work for LLC2 sessions but can now be used for IP.

IP security considerations

When introducing Enterprise Extender into the network, there will be IP security considerations impacting the operation of EE, such as firewalls and Network Address Translation (NAT):

- **Firewalls**

  SNA traffic is traditionally LLC2 traffic over the LAN. LLC2 is transported between sites or zones of a network using DLSw. Where firewalls are in the path of SNA sessions it will be necessary to implement firewall rules allowing EE UDP traffic. This must be planned with your security architecture and implementation people.

  In discussions with the security administrator it would be worthwhile to show that the implementation of EE actually provides increased security compared to LLC2 traffic.

  – EE traffic is controlled using IP filters in a firewall rather than the layer 2 MAC address controls required for LLC2. In many organizations, the LLC2 traffic bypasses not only the firewalls, but all recognized security devices. Implementing EE actually brings the access rules within the scope of firewall control.

  – LLC2 rules are only at MAC Address level. There is no application or port controls. EE uses only specific ports that are easily identified to create specific filters.

  The risk profile for the organization is therefore improved.

  During the implementation phase of the project ensure the task of implementing firewall rules is included in the schedule and tested.

- **NAT**

  Current levels of z/OS Communications Server support Network Address Translation (NAT) devices.

  If an EE session path includes a NAT device, you must use HOSTNAME (*partner's host name*) in the PATH statement of the switched major node for the related EE connection. The reason for this is that only the partner's host name is passed when EE connections are established. For name resolution, the partner's host name must be defined in the *local* DNS with the IP address of the NAT device's *inbound interface*. The IP address of the inbound interface is before NAT occurs.

Important: Each EE VIPA should have an associated host name defined in its local DNS.

2.4.3 Making strategic decisions

As part of the strategy, evaluate the features and functions just reviewed, and determine which ones should be implemented in your network. Use the following list of questions as a guide in your decision making process. You may want to record your decisions in a format similar to that shown in Table 2-4 on page 76.

- Do the APPN node types change in my EE implementation?
  - EE increases the geographic reach of the APPN/HPR network. Review the number and placement of NNs in the network. The environment for APPN may have changed since it was originally planned. Can you reach the NNs and backup NNs from all nodes in the EE network?
  - If there are non-EE NNs in the network how does that impact SNA routing?
- Does the SNA naming convention need to be modified for EE resources?
  - For EE specific resources and major nodes, it would be useful to identify the EE specific nodes by name. In this book, we used the EE prefix for Enterprise Extender specific resources as follows:
    - EESW.... for switched major node
    - EECP.... for CPNAME of EE nodes.
    - EEPU.... for PUs matching CPNAME for an EE node.
- Is a dedicated stack required for EE?
  - From an EE perspective this is only likely if IP MULTIPATH is enabled on your present TCP/IP stack and you expect it will impact EE performance. There may be other, non-EE, reasons why you want to have multiple TCP/IP stacks. So those considerations must also be taken into account.
- Is a Local Connection Network (LCN) to be used?
  - It is recommended that a local connection network be used, if at all possible. The only reason not to use the local connection network would be to require all ILUs and CP sessions to be predefined.
- Is a Global Connection Network (GCN) to be used?
  - If you have EBN connections a GCN may be of benefit. We recommend a GCN if your organization has multiple NETIDs. If the only EBN connections are to business partners, the GCN may benefit efficiency, but remember every business partner connection must also be authorized through the firewalls. This may negate the benefits of GCN.
- Is additional SNA network security required?
  - If you already run a DSME for APPN session control, then the DSME need only be updated to reflect the new connections. There is no change in strategy.
  - If you do not run a DSME and EBN business partner connections are to be introduced, you may want to add the DSME function to your strategy. You may also control EBN connections to z/OS LPARs by running an SME on all LPARs as an alternative to the

DSME. If the target of the EBN connection is a non-VTAM SNA host, then the DSME is strongly recommended.

  - SNI connections can be controlled using the NCP GWNAU statements without defining a POOL. If you have used this option, it will be necessary to review the security controls and consider a DSME.

- Use HOSTNAMES only when defining EE addresses?
  - The use of hostnames rather than dotted decimal IP addresses is generally recommended in an IP network. In this sense, VTAM/EE is no different from any other user of the network. If you plan to NAT addresses and use any form of connection network, definition by host name is required.
- What type of dynamic routing protocol is to be used?
  - We recommend including a dynamic routing protocol in your EE strategy. OSPF or RIPV2 are the choices for z/OS Communications Server. Your selection will be based upon the dynamic routing protocol used in the rest of your network.
  - To run without a dynamic routing protocol is a restrictive option and reduces the redundancy and recovery capabilities of the EE network.
- Use fixed or dynamic TG number for EE?
  - Fixed TG numbers for EE make problem determination simpler. It also makes configuration of the peripheral devices slightly more complex. This is a strategy decision and should be decided for all APPN TGs.
- Will standard EE IPTOS and PORTs be used?
  - If you chose to change either the EE UDP port numbers or the IP TOS settings, you should clearly document this in your strategy.
  - You can code IPPORT only on the EE XCA PORT statement. This makes your choice of ports apply to all of your EE definitions. You will have to change the EE ports definitions on all peripheral devices.
  - Changing IPPORTs will impact your ability to connect to other networks.
- Will IP security be impacted in any way?
  - Will the EE connections traverse firewalls? If so, it will be necessary to add filters for the EE UDP ports to the firewall rules.
- Does QoS strategy and implementation include EE? How?
  - Work with the network designers to include QoS for the EE ports or TOS in your network. Document the mechanism used to provide QoS for EE connections and also make sure it is included in the overall QoS strategy documentation. QoS for EE should also be included in the standard QoS “build” for switches and routers so that you can be sure EE performance will be maintained.
- Will multiple EE VIPAs be used? Using multiple VIPAs enables:
  - Different values of DYNPU in different GROUPs
  - Separation of GLOBAL and LOCAL connection networks.
  - Parallel TGs for z/OS V1R8.0 Communications Server and later.
  - Different IP addresses for accounting.
  - Migration to a new addressing scheme.
  - Migration to IPV6.

- Are there multiple physical paths that could benefit from parallel TGs?
  - The IP network will usually allow for redundant use of all of the physical path between two points in a network.
  - However, it is possible that there may exist parallel paths between any two nodes that use totally independent hardware. In this case parallel TGs would allow both paths to be used between the EE nodes.
  - Parallel TGs require multiple EE VIPAs as of z/OS V1R8.0 Communications Server.
- Permit dynamic adjacent CPs?
  - If you want to have complete control of which adjacent CPs can connect to your network, then require that DYNADJCP=NO be coded in the start options.
  - If DYNADJCP=NO, you will require an ADJCP statement for every CP connecting to a VTAM.
  - If you code DYNADJCP=YES, you can still control dynamic definition of adjacent EE CPs using the DYNPU parameter of the EE XCA GROUP statement. Each adjacent CP will require a predefined switched PU definition to connect, even if DYNADJCP is set to YES.
- Will dynamic PU definitions for EE CP-CP sessions be permitted?
  - In general we recommend predefined PU definitions for CP-CP sessions between LPARs. You can create a separate EE XCA group and VIPA for these connections.
  - Between peripheral devices and the mainframe LPARs, dynamic definitions of PUs for CP-CP sessions can be used if your organization’s policies permit. To enable this, a separate EE XCA GROUP can be created with DYNPU=YES.
- Use model major nodes?
  - The use of model major nodes depends upon your organizations policies.
  - A model major node with DYNTYPE=EE can be used to dynamically create PUs for EE to carry CP-CP sessions to your NNs. It permits key parameters such as DISCNT and TG number to be inherited by dynamic definitions.
  - A model major node with DYNTYPE=RTP can be used to control the characteristics of RTPs and a model major node with DYNTYPE=VN can assist in the dynamic definitions of connections coming in from the connection networks.
- Permit dynamic CDRSCs?
  - Dynamic CDRSCs will be required if connection networks are used
  - Dynamic CDRSCs have commonly been used for many years in the subarea environment, in order to allow flexible connections within the SNA network. It is necessary to permit dynamic CDRSCs if dynamic ILU or dynamic CP definitions are to be allowed.
  - An alternative to dynamic CDRSCs is to create a CDRSC model based upon the LU name of the CPs or ILUs. This provides some level of control over which “dynamic CDRSCs” are to be permitted.
- Permit dynamic ILU definitions?
  - Dynamic ILU definitions of some form must be permitted if connection networks are used.
  - This may be provided globally by using the DYNLU=YES start option or by creating a model CDRSC definition to match the dynamic ILUs you want to permit.

- Require PU or CP association for each ILU? Code ALSREQ=YES as a start option.
  - If you want to maximize the control of ILU connections, you can code the ALSLIST=REQ start option. This will generally require the association of every ILU with one or more specified adjacent PUs. You can also code ALSLIST=ISTAPNPU which will allow any PU from the APPN environment to be associated with an ILU.
  - A predefined CDRSC is required for each ILU.
  - ALSLIST=REQ also controls ILUs from the subarea environment.

Table 2-4 shows the decisions employed in the scenarios for this book. Based on your environment, your decisions may be different.

Table 2-4 Our strategy decisions

Function or Facility	Decision	Comments
Do the APPN node types change in my EE implementation?	SC30M and SC31M are NNs and EBNs. All others connecting to the corporate are ENs. Business partner servers are NN or EBN.	
Does the SNA naming convention need to be modified for EE resources?	Prefix EE resources with “EE” will make it easier to identify those resources.	D NET,RSCLIST,ID=EE* provides a list of EE resources.
Is a dedicated TCP/IP stack for EE required?	Yes, in our case for separation from production network.	Yes, a new TCP/IP stack needs to be created
Is a Global Connection Network to be used?	Yes, there are multiple NETIDs in this organization and many sessions are EN to EN.	One EE XCA group must code VNNAME=W3IBMCOM.VRN, VNTYPE=GLOBAL
Is additional SNA network security required?	No	No third party connections
Use HOSTNAMES only when defining EE addresses?	Yes, only hostnames are to be used.	Hostnames must be added to DNS or IPNODES file.
What type of dynamic routing protocol be used?	OSPF running under OMPROUTE is to be used.	Must coordinate with network designers. Install OMPROUTE
Use fixed or dynamic TG number for EE?	Use fixed TG numbers to assist in problem determination.	Use TG 6 to identify EE connections. Use TG 7 for EBN EE connections.
Will standard EE IPTOS and PORTs be used?	Yes, the default ports and TOS settings will be used.	
Will IP security be impacted in any way.	No, firewalls are not traversed in our environment.	
Does QoS strategy and implementation include EE? How?	No, QoS in the test network	
Will multiple EE VIPAs be used?	Yes	Two XCA Groups, one DYNPU=NO, the other DYNPU=YES

Function or Facility	Decision	Comments
Are there multiple physical paths that could benefit from parallel TGs?	No	Parallel TGs configured in Chapter 9
Permit dynamic adjacent CPs?	Yes	DYNADJCP start option Defaults to YES
Will dynamic PU definitions for EE CP-CP sessions be permitted?	Use predefined PU for incoming z/OS LPARs, but allow dynamic CP for all others	Two EE XCA groups for each LPAR, one with DYNPU=YES and the other with DYNPU=NO. Define a switched PU with the CPNAME for every CP connecting over DYNPU=NO XCA.
Use model major nodes?	Yes, DYNTYPE=EE, RTP and VN	Model major nodes do not enable CONNECTOUT.
Permit dynamic CDRSCs	Yes	CDRDYN=YES start option
Permit dynamic ILU definition?	Yes	DYNLU=YES start option, also requires CDRDYN=YES.
Require PU or CP association for each ILU?	No	ALSREQ=NO default start option used.

2.5 Developing a high-level design

Now that all information has been gathered, reviewed, evaluated, and the strategy is clear, it is possible to create a design for the EE environment. This design may not be limited to EE, but could include considerations for other SNA and IP solutions.

The EE network used for this book is shown in Figure 2-5 on page 78. It is a single site network, with connections to business partners and downstream servers running IBM Communications Server on various platforms and IBM Personal Communications.

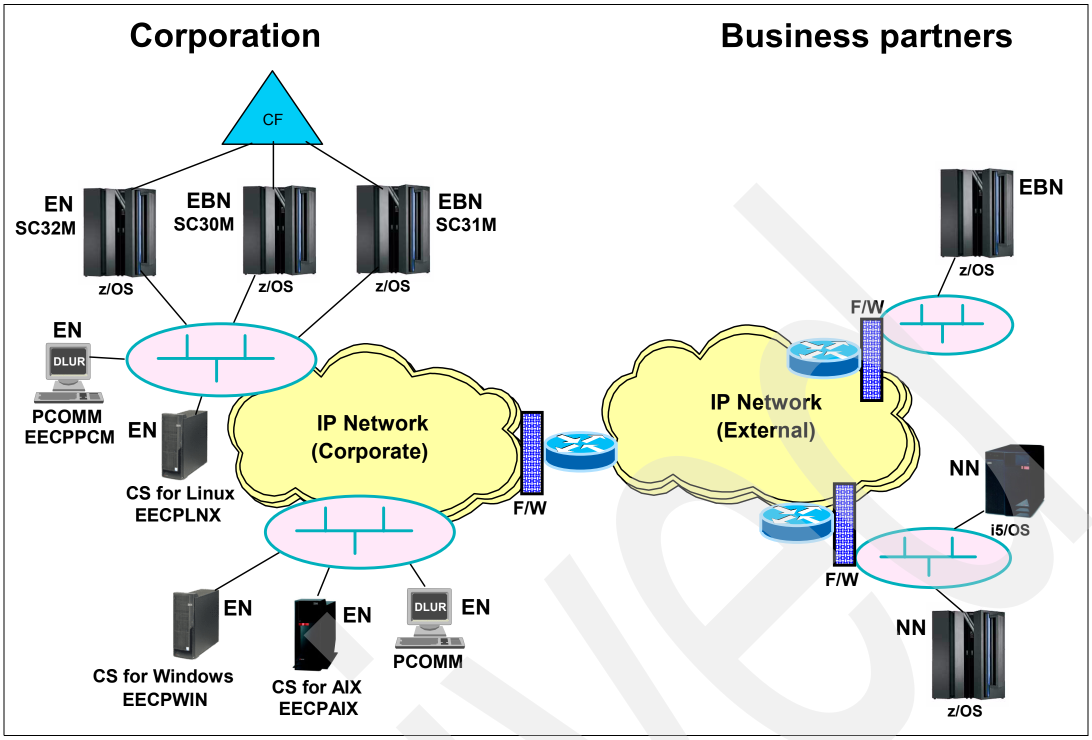


Figure 2-5 Our EE network

Our EE network is replacing the APPN environment shown in Figure 2-1 on page 52.

We discuss the following in this section:

- CMC or Network Node placement
- VTAM configuration
- TCP/IP stack configuration
- OMPRoute configuration
- Security options
- Server migration

2.5.1 CMC or Network Node placement

Determining and positioning the CMCs (NNs) is key in any design of an APPN environment. If you currently have an APPN environment implemented, you probably already have these assigned. However, this is a good opportunity to revisit the reasons previously used for assigning roles to the various systems. When determining where the CMC (NN) function should be implemented, a good starting point is to consider where the existing CMC functions are placed.

Will there be inter-network connections (EBN)? If so, are the CMCs correctly placed for the EBN function? Is there to be redundancy in the EBN configuration? If so, are there two CMCs that are suitable for WAN link connections? If there are several WAN links, do they all terminate in the same site? Will you use IP to route over failed WAN links or will you let HPR do the rerouting?

The RDBOOKEE network will also accommodate an EBN connection to network USIBMSC as is shown in Figure 2-6.

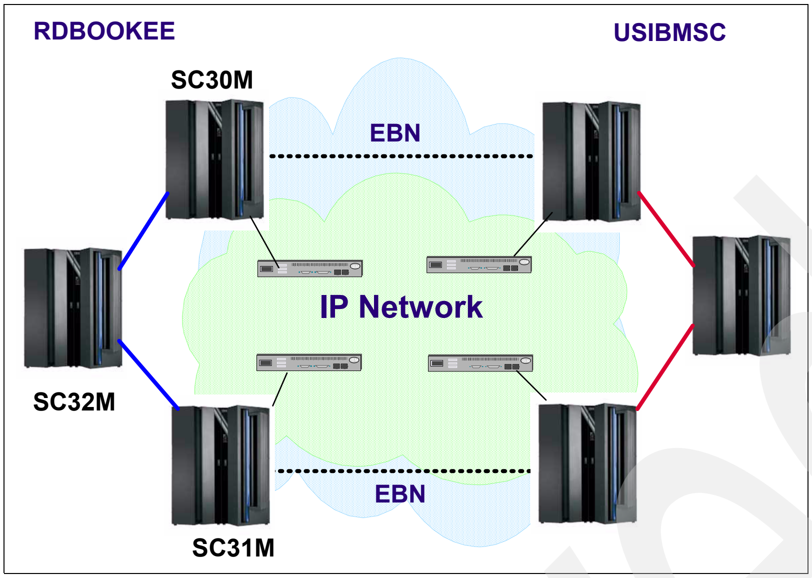


Figure 2-6 EBN connection between two networks

Table 2-5 lists the EBNs for our EE environment.

Table 2-5 Select the Network Nodes and identify their location

CMC Name	CMC Location	Role	Subnetwork
RDBOOKEE.SC30M	ITSO Data Center, Poughkeepsie	Primary NNS, EBN	Only
RDBOOKEE.SC31M	ITSO Data Center, Poughkeepsie	Backup NNS, EBN	Only

2.5.2 VTAM configuration

The basic VTAM definitions and start options will be based on the strategy you have chosen for EE implementation and upon the types of connections you will have.

Table 2-6 lists the VTAM start options and ATCCONxx members used in our scenarios.

Table 2-6 VTAM start options and VTAMLST members

VTAM Definition	Value or details	Comment
TCPNAME	TCPIPA	We used a second stack in order to access our test network
DYNADJCP	YES	Dynamics are allowed where possible
ALSREQ	NO	start option
CDRDYN	YES	start option
DYNLU	YES	start option

VTAM Definition	Value or details	Comment
XCA Group 1 ▸ DYNPU ▸ VNNAME ▸ VNTYPE ▸ HOSTNAME	XCA GROUP Values ▸ YES ▸ RDBOOKEE.VRNLOCAL ▸ LOCAL ▸ SCxxM-EE1.ITSO.IBM.COM	EE XCA major node, EEXCA xx = 30, 31 or 32
XCA Group 2 ▸ DYNPU ▸ VNNAME ▸ VNTYPE ▸ HOSTNAME	XCA GROUP Values ▸ NO ▸ W3IBMCOM.VRN ▸ GLOBAL ▸ SCxxM-EE2.ITSO.IBM.COM	EE XCA major node, EEXCA xx = 30, 31 or 32

Create Model Major Nodes	DYNTYPE=EE
	DYNTYPE=RTP
	DYNTYPE=VN
		EEMODLPU
		EEMODLRT
		EEMODLVN

Create Switched PUs	Inter-LPAR Servers	EESWNN30, EESWNN31, EESWEN32 EESWAIX, EESWLNX, EESWWIN, EESWPCOM

2.5.3 TCP/IP stack configuration

Table 2-7 lists the TCP/IP profile definitions used in our scenarios.

Table 2-7 TCP/IP profile

TCP/IP Definitions	Value or details	Comments
Number of EE VIPAs per LPAR	10.10.1.231 and 10.10.1.232 10.10.1.241 and 10.10.1.242 10.10.1.221 and 10.10.1.222	SC30M SC31M SC32M
Ports 12000-12004 allocated to VTAM	PORTRANGE 12000 5 UDP NET	Not required, but controls who can use the port
Assign DNS names to EE VIPAs	Request corporate DNS update	SCxxM-ee1.itso.ibm.com SCxxM-ee2.itso.ibm.com xx = 30, 31 or 32

2.5.4 OMPRoute configuration

Table 2-8 lists the OMPRoute definitions for the OSPF dynamic routing protocol used in our scenarios.

Table 2-8 OMPROUTE

TCP/IP Definitions	Value or details	Comments
OSPF_Interface for each VIPA	10.10.1.231 and 10.10.1.232 10.10.1.241 and 10.10.1.242 10.10.1.221 and 10.10.1.222	Advertise_VIPA_Routes=HOST_ONLY
MTU for each interface	MTU=1500	Gigabit Ethernet OSA-Et

2.5.5 Security options

The security options chosen when the strategy was created are to be detailed at design time. This will involve updates to existing security rules or perhaps the introduction of new security applications or appliances.

2.5.6 Server migration

Devices being added to the environment will be the easiest to implement, as they generally go straight into a test environment. This way, there is an opportunity to fully test and document the environment.

Migrating existing devices is a more difficult task. Most servers cannot simply reconfigure from LLC2 to EE. The business will want significant proof that the application will not be adversely impacted. This proof will come from test or development versions of the servers.

Migration is simpler for servers when it coincides with a hardware upgrade or refresh. There is generally a window for testing during the upgrade process.

Table 2-9 shows the information that is needed when configuring a connection to a server.

Table 2-9 Information for connecting to a server

EE Definitions	Value or details	Comments
Local CPNAME	EECPnnn	AIX, LNX, WIN, PCOM
Local CPNAME node ID	0EE EE000	Some servers require IDNUM and IDBLK even though it is not used. This is not the IDNUM IDBLK used for PU 2.0
Primary NNS host name	SC30M-EE1.itso.ibm.com	This is XCA group with DYNPU=YES
Backup NNS host name	SC31M-EE1.itso.ibm.com	This is XCA group with DYNPU=YES
Connection Network	RDBOOKEE.VRNLOCAL	Local connection network
DLUS	RDBOOKEE.SC30M RDBOOKEE.SC31M	

2.6 Implementing the EE design

Before EE is rolled out, what else do you have to do? How do you test the environment to make sure it is doing what is expected? We answer these questions in the subsequent sections.

The following topics are covered in this section:

- Operations training
- Logical partitions first
- Testing and verifying the EE configuration
- Deployment of EE into production

2.6.1 Operations training

Before you begin even the testing, consider how the operations staff will be able to support the EE environment. There will be many degrees of operational readiness. Operations capabilities (APPN, HPR and EE) can range from none to expert depending upon your SNA history.

If your network has implemented APPN to a wide extent, it is likely your operations and delivery personnel are familiar with APPN. The implementation of EE will simply be an additional DLC for operations.

If your operations staff have not operated in an APPN environment, then it will be necessary to provide training as part of the project.

There is one noticeable difference between the EE DLC and previous SNA connections over an IP network. It is now possible to test a connection end-to-end from the mainframe using a combination of SNA and TCP/IP facilities. Advise your operations staff of this capability rather than assuming they are aware of it. If the operations staff are “SNA people” then they will require training on the IP aspects of EE.

Training of operations should begin before any testing is set up.

2.6.2 Logical partitions first

The first candidates for EE implementation are almost always the z/OS Communications Server systems. The z/OS systems usually provide the NNS functions for the rest of the network. The z/OS systems are also the proving ground for the EE project. If all goes well during the z/OS upgrade, it gives the business a boost in confidence about the whole project.

Code all VTAM major nodes that are needed to support your design. Develop a test plan to implement the CMCs (NNs) first and prepare to verify them one at a time. Make sure each NN can connect to the other(s). Then, plan the implementation of the EN systems, one at a time, making sure they can connect to their appropriate NNS. If your design accommodates connections between the ENs, then plan to verify those connections last. Refer to the appropriate chapters in this book for detailed coding examples and testing procedures related to your scenario.

With the z/OS systems it may not be possible to follow the same testing and rollout process as will be used for the “application servers”. The reason for this is almost all applications have a presence on the z/OS LPARs and the testing opportunities are few.

Finally, create your implementation and testing plan for the remote servers and other network devices that are to use EE. Reference the appropriate chapters in this book for the platform involved in your situation. If this book does not discuss your specific platform, use the chapter which most closely resembles your situation.

2.6.3 Testing and verifying the EE configuration

We recommend building a test environment where you can install, test, and verify each device and connection type that will be implemented in your production environment. With that test environment you can also try out recovery scenarios, document operational procedures, and learn how to proceed with the implementation tasks.

Testing the EE implementation generally involves three stages.

1. Proof of Concept (PoC)

2. Pilot
3. Procedural verification

Although it would seem that the stages should run one after another, the stages may overlap considerably. For that reason, the three stages will apply to an application or network function (gateways, for example) and they probably will not all start at the same time. The PoC of the last application may only begin when the first application has completed its rollout.

Proof of Concept (PoC)

The proof of concept involves test devices configured as closely as possible to the production environment. The basic connectivity is tested and functional testing is performed at both system and applications level. The majority of testing is to be performed at application level and should exercise the application in a similar manner to testing for a new release of the application. The documented test cases for the last application upgrade may be useful. File transfer, if used, should be tested to ensure throughput is as good or better than in the old configuration.

If there are a number of different applications or types of services implementing EE, you will probably have to engage in a separate PoC for each application of function.

The implementation of EE has not had significant impact upon SNA connections. In some cases non-VTAM code used for EE has enforced SNA rules that have not been enforced in the past. The symptom of this is a connection failure or error, but it is usually presented with a sense code identifying the problem. Correct the problem rather than abandoning EE.

Some issues that have been seen are:

- Invalid Logmode that works with one gateway device, but not another.
- Data corruption due to OS stack sequencing errors (triggered by higher throughput).
- BIND issues based upon interpretation of BIND control vectors.
- When downstream devices move to a new gateway, you may need to review LLC2 acknowledgement handshaking as file transfer throughput can be impacted.

Pilot

The pilot verifies that the PoC testing covered all aspects of the application and that EE works well in a live environment. The objectives of the pilots are to:

- Test components of the deployment plan to ensure executability
- Gain experience and develop best practices to be utilized
- Provide value to the deployment process, avoiding unplanned events

In addition, the PoC stage determined whether the EE connections impact the applications running over them. We recommend you select a small number of devices or locations to use in a pilot. Again this will be on an application by application basis. The pilot approach includes:

- Identifying what is to be tested
- Conducting structured, controlled tests
- Gathering and assessing results
- Measuring those results against expectations and documented goals
- Building experience and refining the final implementation methodologies

When constructing pilots, consider these questions:

- What is the purpose of the pilot?
- What are the desired results?
- What does a successful pilot look like?
- How do we measure success?

- What criteria must be met in designing the pilot?
- What should be considered in selecting a candidate: machines, people?
- Who are the participants and what are their roles?
- How many pilots should be conducted?
- What are the lessons learned?

Procedural verification

Part of any well planned implementation should include becoming familiar with test procedures and the appropriate commands, in order to perform proper network verification. We recommend that you review the following publications :

- The *z/OS Communications Server SNA Network Implementation Guide*, SC31-8777, includes an Enterprise Extender chapter that provides a suggested procedure for troubleshooting EE problems.
- The *z/OS Communications Server IP System Administrator's Commands*, SC31-8781, describes the following IP commands which are helpful in an EE environment:
  - D TCPIP,,NETSTAT,HOME
  - D TCPIP,,NETSTAT,DEVLINKS
  - D TCPIP,,NETSTAT,ALLCONN,CLIENT=NET
  - D TCPIP,,NETSTAT,ALLCONN,PORT=1200N
  - D TCPIP,,NETSTAT,CONN,CLIENT=NET
  - D TCPIP,,NETSTAT,CONN,PORT=1200N
  - D TCPIP,,NETSTAT,SOCK,CLIENT=NET
  - D TCPIP,,NETSTAT,SOCK,PORT=1200N
  - D TCPIP,,NETSTAT,BYTE,CLIENT=NET
- The *z/OS Communications Server SNA Operation*, SC31-8779, describes the following SNA commands and their operands:
  - D NET,EE
  - D NET,EEDIAG
  - D NET,TOPO
  - D NET,RTPS
  - D NET,TRL
  - D NET,CPCP
  - D NET,NETSRVR
  - D NET,APING
  - D NET,ID=THEEEPUNAME
  - D NET,ID=THEEEXCANAME
  - D NET,ID=THEEEMODELNAME
  - D NET,ID=THEEESWNETNAME
  - D NET,ID=ISTADJCP,E

Throughout this book, you will find examples that use these commands.

2.6.4 Deployment of EE into production

In order to deploy Enterprise Extender successfully, you must have a thorough executable roll out plan. That plan should include such considerations as:

- Committed management leadership and backing
- Effective communications to all user groups affected
- Clarity of direction with measurable goals to be achieved by EE
- Disciplined project management to keep unplanned events under control
- A documented methodology for implementation
- A documented methodology for back out, if necessary

The implementation and back out considerations should answer key questions about the deployment, for example:

- For management alignment and backing, who is the advocate and have they approved deployment?
- For effective communications, what and how do we coordinate communication and education?
- For expectations to be met, have those expectations been documented and can the goals be achieved?
- For the project plan, is the project manager briefed on the personnel resource requirements?
- For the change management system, are all change records created, accurate, and approved?
- For the schedule of events, who will perform each of the deployment activities and by when?
- For implementation planning, is there a well defined script: who, what, when, where, how?
- For contingency planning, have we documented and shared the lessons learned from the pilot(s).
- For back out requirements, what must be done to deploy the contingency plans?

There are some additional areas to consider in preparing for EE deployment. Make sure that the implementation plan and contingency plan are both translated into executable steps. Sometimes impressive documents are created that use "business style" language to help the non-technical person understand the changes that are planned, but they fall short when it comes time for the technician to perform those changes. The actual commands and procedures are missing, and the technician has no script to follow.

We recommend that you create short, simple, and targeted procedures for each role involved to enable proper execution of assigned duties. Those procedures should accommodate roles such as operators, network engineers, firewall administrators, VTAM and TCP/IP systems personnel, application owners and verifiers, help desk personnel, DNS name server administrators, and perhaps even security systems.

In addition, verify that the education and communication materials are valid by "piloting" them in a dry-run mode to make sure they are ready for their intended audience. Have the relevant materials posted for easy access. Also, the input and output matrix should be defined and tracked, with management reporting and measurements in place. Lessons learned should be documented, shared with the team, and provided to the project manager. Prepare a project summary presentation for the management advocate who supported the project, showing how the deployment of EE met the goals of the project.

3

Preparing z/OS for EE

This chapter discusses the steps required to prepare your z/OS system to function as an Enterprise Extender (EE) capable Advanced Peer-to- Peer Networking (APPN) Node. We describe how the Communications Server must be defined to provide access to the z/OS environment and to participate in an APPN environment.

The following topics are discussed:

- “Description of our APPN environment”
- “Preparing TCP/IP to participate in an APPN/EE network”
- “Preparing VTAM to participate in an APPN/HPR network”
- “Consolidating LOGMODEs”
- “Verifying the APPN topology”

This chapter does not provide any guidelines for migrating from a pure subarea infrastructure or the potential problem areas that such a mixed subarea/APPN infrastructure imposes. If you are not familiar with APPN or EE, refer to Chapter 1, “Introduction to APPN and Enterprise Extender” on page 1 for details.

3.1 Description of our APPN environment

The steps in this chapter show how to prepare an APPN environment to support the z/OS EE environment. Figure 3-1 shows the desired APPN topology consisting of three VTAMs in a sysplex. Two are started as Network Nodes (NNs), and one is started as an End Node (EN). There are two Transmission Groups (TGs) between each VTAM: XCF links (TG21) and MPC links (TG2).

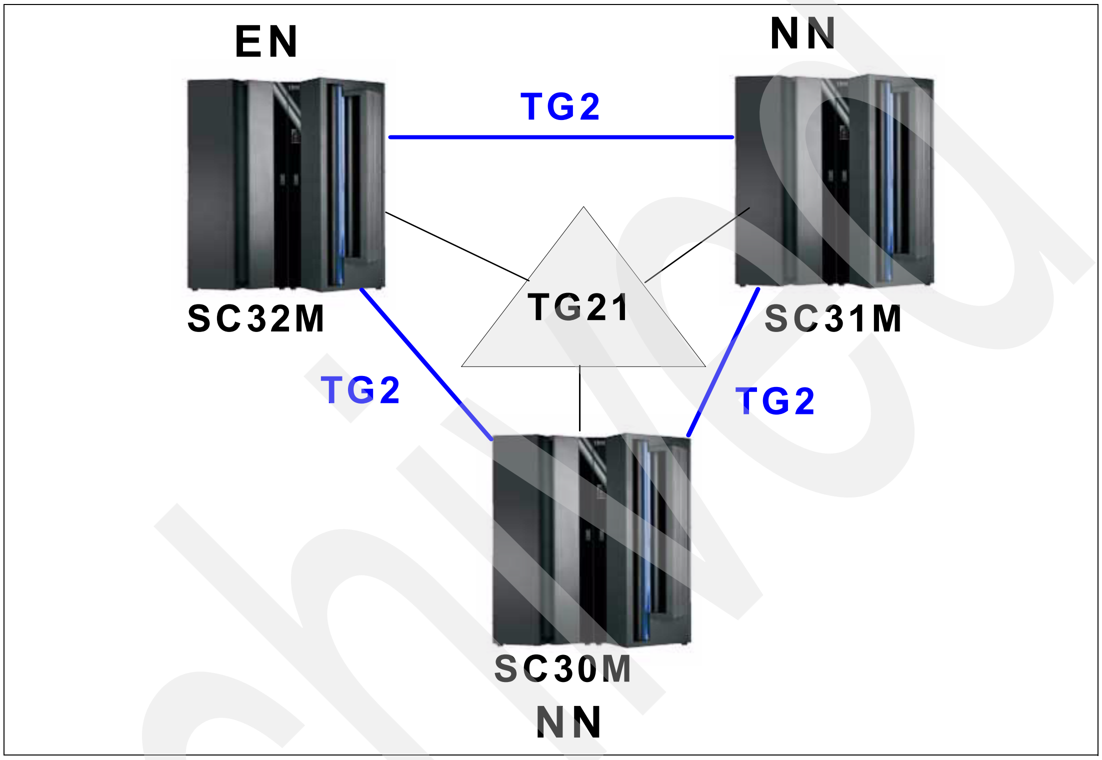


Figure 3-1 Our APPN topology

Highly available access to a z/OS system is provided by designing redundant IP interfaces into a redundant IP network infrastructure. This includes Virtual IP Addresses (VIPAs) and the use of a dynamic routing protocol. In our scenarios we used two OSA-Express2 1000BASE-T features (4 ports) in Queued Direct Input Output (QDIO) mode, one port from each OSA-Express2 feature was assigned to a separate VLAN, along with Open Shortest Path First (OSPF) as the dynamic routing protocol. A static VIPA was included in our setup to represent the stack itself.

For complete details on designing high availability for your z/OS networking systems, refer to *Communications Server for z/OS TCP/IP Implementation, Volume 3 - High Availability, Scalability, and Performance*, SG24-7341, and *Communications Server for z/OS V1R7 TCP/IP, Implementation Volume 3 - High Availability, Scalability, and Performance*, SG24-7171.

3.2 Preparing TCP/IP to participate in an APPN/EE network

The following discussion assumes that the TCP/IP stacks are already running and supporting IP traffic. However, your existing environment may not include use of Cross-system Coupling Facility (XCF), the latest Open Systems Adaptor (OSA) technologies, and OSPF dynamic routing. For this reason, we have included sample definitions for XCF, OSA (in QDIO mode), and OSPF.

In order to keep our focus on Enterprise Extender (EE) features instead of TCP/IP features, we kept our network design simple, only using OSA-Express2 ports.

Figure 3-2 depicts the TCP/IP definitions we considered in preparing our systems to support an EE environment.

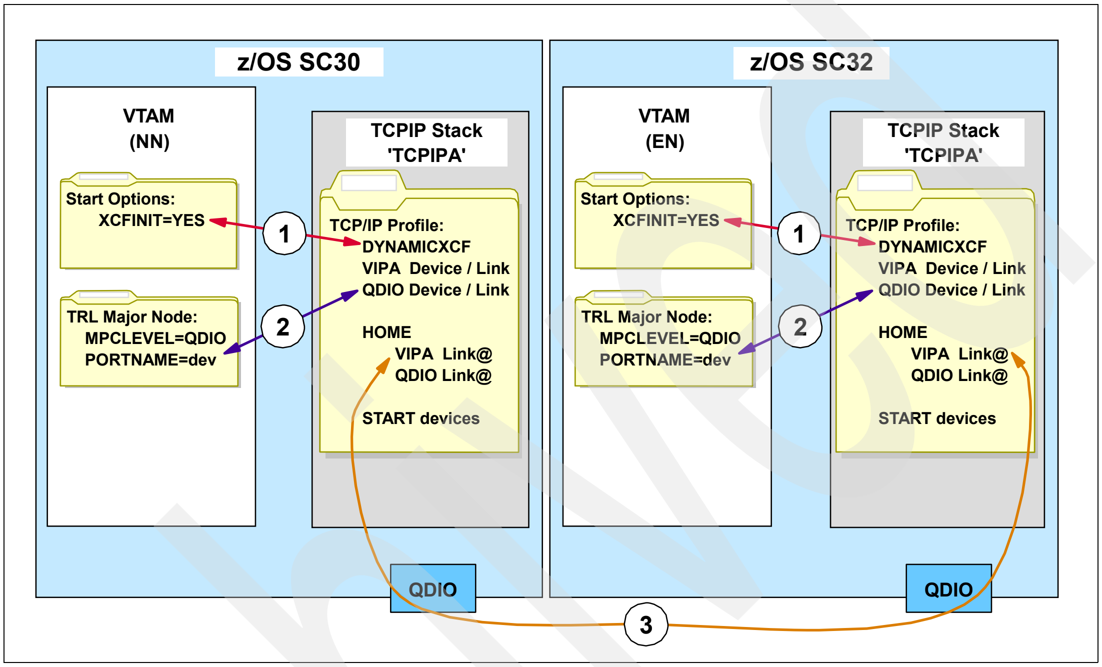


Figure 3-2 TCP/IP definitions to prepare for an EE environment

Note the following explanation for Figure 3-2:

1. DYNAMICXCF is the mechanism that TCP/IP uses to dynamically build the DEVICE, LINK, HOME, and START statements that support XCF inter-stack communications within the sysplex. When the DYNAMICXCF statement is used, manual coding of these statements can be avoided. VTAM must have the XCFINIT=YES start option specified in order for TCP/IP to dynamically create and allocate the XCF interfaces.
2. OSA QDIO interfaces are used in this illustration to represent the physical interfaces to be used later by EE.
3. Each stack should have at least one static VIPA defined that represents that stack to the network. Usually that VIPA is also treated as the stack's SOURCEVIPA.

The examples in this section are all based on LPAR SC30 and its TCP/IP stack (TCPIPA). The definitions and commands for the other z/OS systems are similar, except where names and addresses must be unique.

We discuss the following topics related to preparing the TCP/IP environment:

- "Modifying the TCP/IP PROFILE"
- "Modifying the OSPF environment"
- "Starting the TCP/IP task"
- "Starting the OMPROUTE task"
- "Verifying the TCP/IP environment"
- "Verifying the OSPF environment"

3.2.1 Modifying the TCP/IP PROFILE

The following items discuss the PROFILE considerations when preparing the TCP/IP stack for high availability and positioning it for a later EE environment:

- “DEVICE and LINK statements for XCF inter-stack communications”
- “DEVICE and LINK statements for the static VIPA”
- “DEVICE and LINK statements for the OSA devices”
- “HOME statements for the VIPA and OSAs”
- “STARTing the devices”

DEVICE and LINK statements for XCF inter-stack communications

If the TCP/IP stacks in the sysplex are to use inter-stack communication, the XCF interfaces must be defined in each participating stack. They can be defined dynamically by using the DYNAMICXCF operand on the IPCONFIG profile statement.

Example 3-1 shows our IPCONFIG statement with the DYNAMICXCF parameter added.

Example 3-1 DYNAMICXCF statement for XCF

```
;TCP/IP Dynamic definition:
   IPCONFIG DYNAMICXCF 10.10.20.100 255.255.255.0 1     1

;VTAM start option:
   XCFINIT=YES                                          2
```

Note the following explanation for Example 3-1:

- 1: The dynamic XCF devices are used for inter-stack communications within the sysplex. To simplify XCF definitions, define them using the IPCONFIG DYNAMICXCF statement. The DEVICE name defaults to the SSCPNAME of the partner system’s VTAM, and the LINK name defaults to EZAXCFxx, where xx is the partner system’s &SYSCLONE value.
- 2: The VTAM start option, XCFINIT=YES, must be specified in ATCSTRxx.

DEVICE and LINK statements for the static VIPA

Example 3-2 shows the statements we used to define our static VIPA.

Example 3-2 VIPA Device and Link statements

```
DEVICE VIPADEV  VIRTUAL 0          1
LINK VIPALINK VIRTUAL 0 VIPADEV
```

Note the following explanation for Example 3-2:

- 1: Define one static VIPA to represent the stack itself. The non-EE traffic can use that VIPA to access the stack.

  A Dynamic VIPA discussion and the reasons to use it are out of the scope of this book, refer to *Communications Server for z/OS TCP/IP Implementation, Volume 3 - High Availability, Scalability, and Performance*, SG24-7341.

DEVICE and LINK statements for the OSA devices

Example 3-3 shows the VTAM and TCP/IP statements we used to define *one* of our four OSAs.

Example 3-3   OSA Device and Link statements

```
       TCP/IP Profile statements
DEVICE OSA2080 MPCIPA NONROUTER AUTORESTART 1
LINK OSA2080LNK IPAQENET OSA2080            2


       VTAM TRL Major Node
OSA2080  VBUILD TYPE=TRL
OSA2080P TRLE  LNCTL=MPC,                                        *
               READ=2080,                                        *
               WRITE=2081,                                       *
               DATAPATH=(2082-2088),                             *
               PORTNAME=OSA2080,                                 * 1
               MPCLEVEL=QDIO                                       2
```

Note the following explanation for Example 3-3:

- 1: VTAM performs the DLC (I/O) functions for QDIO interfaces for TCP/IP. QDIO mode interfaces require TRLE definitions in VTAM. The TRL PORTNAME must match the TCP/IP DEVICE name.
- 2: QDIO mode (IPAQENET) is the preferred mode of operation for OSA adapters and is used in our examples.

Even though EE supports all interface types that the TCP/IP stack supports, the latest OSA technologies should be used in order to achieve the best throughput and performance.

HOME statements for the VIPA and OSAs

Example 3-4 shows the HOME statements for our devices.

Example 3-4   Home statements

```
HOME
  10.10.1.230    VIPALINK    1
  10.10.2.232    OSA2080LNK
  10.10.3.233    OSA20A0LNK
  10.10.2.234    OSA20C0LNK
  10.10.3.235    OSA20E0LNK
PRIMARYINTERFACE VIPALINK    2
```

Note the following explanation for Example 3-4:

- 1: The order in which the VIPA link interfaces are positioned in the HOME list is very important when SOURCEVIPA is in effect. Place a specific VIPA link immediately above those physical interfaces for which that VIPA is to be the SOURCEVIPA.
- 2: Use the PRIMARYINTERFACE statement to specify which link is to be designated as the default local host for use by the GETHOSTID() function. The PRIMARYINTERFACE statement’s link IP address is *not* used as the source IP address for any out-going datagrams, *unless* that *same* address is configured as the SOURCEVIPA address.

STARTing the devices

Example 3-5 shows the statements we used to start our devices.

Example 3-5 Starting the IP devices

```
START OSA2080
START OSA20A0
START OSA20C0
START OSA20E0
```

Note the following explanation for Example 3-5:

- We did not have to provide a START statement for the VIPAs. They are started automatically by the stack and do not require a START statement.
- We did not have to provide a START statement for the XCF devices. Using the DYNAMICXCF statement causes them to start automatically.

3.2.2 Modifying the OSPF environment

Some installations may not be running OSPF. However, many of the latest features of TCP/IP that support dynamic load balancing and multi-instance applications require the OSPF dynamic routing protocol support. In addition, when a static VIPA address is assigned from a subnet that is different from the subnet of the physical devices, a dynamic routing protocol is required. Because we assigned a unique subnet to our VIPAs, we included an OSPF setup in our scenarios.

The following items discuss OMPROUTE considerations when preparing for EE:

- “OMPROUTE started task procedure”
- “OMPROUTE environment variable control file”
- “OSPF miscellaneous statements”
- “OSPF_INTERFACE statements for VIPAs”
- “OSPF_INTERFACE statements for OSAs”
- “INTERFACE statement for XCF”

OMPROUTE started task procedure

Example 3-6 shows the statements we used to define our OMPROUTE started task (OMPA).

Example 3-6 OMPROUTE started task

```
//OMPA   PROC STDENV=STDENV&SYSCLONE
//OMPA   EXEC PGM=OMPROUTE,REGION=4096K,TIME=NOLIMIT,
//        PARM=('POSIX(ON) ALL31(ON)',
//             'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA"',   1
//             '"_CEE_ENVFILE=DD:STDENV")/')                 2
//*            '"_CEE_ENVFILE=DD:STDENV")/-t2 -d1')
//*
//STDENV   DD DISP=SHR,DSN=TCPIPA.OMPROUTE.&STDENV
//SYSPRINT DD SYSOUT=*                                       3
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

Note the following explanation for Example 3-6:

- 1: The ENVAR parameter on the EXEC statement contains a stack affinity transport assignment variable set to TCPIPA. Stack affinity can also be assigned in the resolver config file using the TCPIPJOBNAME statement. If both are used, the environment variable in the Job Control Language (JCL) takes precedence.

- 2: The ENVAR parameter on the EXEC statement also contains a pointer to the environment variable file Data Definition (DD) statement. The environment variable control file, containing the environment variables used by OMPROUTE, can be either a PDS member, a sequential MVS file, or an HFS file.
- 3: The SYSPRINT and SYSOUT DD statements can direct output to either the system output queues or to HFS files.

A sample start procedure is provided in SEZAINST(OMPROUTE).

User IDs must be RACF® authorized for starting OMPROUTE, including the user ID assigned to the started task procedure, in our case OMPA. To reduce the risk of an unauthorized user starting OMPROUTE and affecting the contents of the routing table, users who start OMPROUTE must be RACF-authorized to the entity MVS.ROUTEMGR.OMPROUTE and require a UID of zero. OMPROUTE requires UID=0 for correct route installation, configuration, and operation.

OMPROUTE environment variable control file

Example 3-7 shows the environment variables we used with OMPROUTE.These environment variables are used by OMPROUTE and can be tailored to a particular installation.

Example 3-7 Environment variables for OMPROUTE

```
RESOLVER_CONFIG=//'TCPIPA.TCPPARMS(DATAa30)'  1
OMPROUTE_FILE=//'TCPIPA.TCPPARMS(OMPA30)'     2
OMPROUTE_OPTIONS=hello_hi                     3
OMPROUTE_DEBUG_FILE=/tmp/syslog/debuga30      4
OMPROUTE_DEBUG_FILE_CONTROL=10000,5           5
```

Note the following explanation for Example 3-7:

- 1: The RESOLVER_CONFIG variable is used by OMPROUTE to specify the resolver configuration file (known as TCPIP.DATA or SYSTCPD in other applications). The resolver configuration file contains keywords (DATASETPREFIX and TCPIPJOBNAME) used by OMPROUTE. The value assigned to DATASETPREFIX will determine the high-level qualifier (HLQ). The HLQ is used in the search order for the OMPROUTE configuration file. If no DATASETPREFIX keyword is found, a default of TCP/IP is used. The value assigned to TCPIPJOBNAME will be used as the name of the TCP/IP stack with which OMPROUTE establishes a connection (stack affinity).
- 2: The OMPROUTE_FILE variable is used by OMPROUTE in the search order for the OMPROUTE configuration file. The configuration file contains all the definition statements to support OSPF.
- 3: The OMPROUTE_OPTIONS variable is used by OMPROUTE to set various controls for OMPROUTE processing. Currently only the *hello_hi* option is supported. The syntax of this variable is: **OMPROUTE_OPTIONS=hello_hi** ; specifying OMPROUTE_OPTIONS=hello_hi changes the way OMPROUTE processes the IPv4 OSPF hello packets. These packets are given a higher priority than other updates and processed by the first available OMPROUTE task ahead of other received IPv4 OSPF packets.
- 4: The OMPROUTE_DEBUG_FILE variable is used by OMPROUTE to establish the debug log output destination. Various logging and error messages are written to this log file.

- 5: The OMPROUTE_DEBUG_FILE_CONTROL variable is used by OMPROUTE to control the size and quantity of trace files created when OMPROUTE_DEBUG_FILE is specified. The syntax of this variable is: OMPROUTE_DEBUG_FILE_CONTROL=<size of file>,<num of files>

OSPF miscellaneous statements

The OMPROUTE configuration file provides information about the TCP/IP interfaces and how OSPF is to operate. For details on all the OMPROUTE configuration statements and their syntax, refer to *z/OS Communications Server: IP Configuration Reference*, SC31-8776.

A sample configuration file is provided in SEZAINST(EZAORCFG).

Example 3-8 shows miscellaneous OMPROUTE statements.

Example 3-8   OSPF miscellaneous statements

```
Area Area_Number=0.0.0.2
     Stub_Area=YES            1
     Authentication_type=None;
OSPF RouterID=10.10.1.230;    2
```

Note the following explanation for Example 3-8:

- 1: The OSPF area is defined as a totally stubby area.
- 2: This OMPROUTE instance has a router-id that is set to the stack’s static VIPA.

OSPF_INTERFACE statements for VIPAs

Example 3-9 shows static VIPA interface OMPROUTE statements.

Example 3-9   OSPF_INTERFACE statements for VIPAs

```
; Static vipa for the stack
ospf_interface ip_address=10.10.1.230      1
           subnet_mask=255.255.255.0
           name=VIPALINK                   2
           Advertise_VIPA_Routes=HOST_ONLY 3
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
```

Note the following explanation for Example 3-9:

- 1: The IP address of the VIPA must match the address in the HOME list.
- 2: The name must match the link name in the HOME list.
- 3: Only the full 32-bit host address of a VIPA should be advertised and *not* the subnet address of the VIPA.

OSPF_INTERFACE statements for OSAs

Example 3-10 shows OSA interface OMPROUTE statements. Only one of the four OSAs in our configuration is shown.

Example 3-10 OSPF_INTERFACE statements for an OSA

```
; OSA Qdio OSA2080LNK
ospf_interface ip_address=10.10.2.232  1
           subnet_mask=255.255.255.0
           name=OSA2080LNK             2
           ROUTER_PRIORITY=0           3
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500 ;
```

Note the following explanation for Example 3-10:

- 1: The IP address of the OSA must match the address in the HOME list.
- 2:The name must match the link name in the HOME list.
- 3: The router priority must be set to zero so that this instance of OMPROUTE does not become an OSPF Designated Router.

INTERFACE statement for XCF

Example 3-11 shows XCF interface OMPROUTE statements.

Example 3-11 INTERFACE statement for XCF

```
INTERFACE                          1
     IP_Address=10.10.20.*         2
     Subnet_Mask=255.255.255.0
     MTU=1500;
```

Note the following explanation for Example 3-11:

- 1: XCF interfaces are for use by the mainframes within the sysplex. Only those mainframes in the sysplex need to know about the XCF interfaces and IP addresses. No external router nor network device ever need to know about the XCF IP addresses. These XCF addresses should never be advertised. A quick way to accomplish this is to define them as a non-OSPF interface by using the INTERFACE statement instead of the OSPF_INTERFACE statement.
- 2: Because there is a potential for many XCF IP addresses (one for each stack participating in the sysplex), we assigned an entire subnet to XCF and wild-carded the IP address accordingly. By wild-carding, we avoided having to code *each* XCF IP address in the sysplex. A wild-carded entry does not require the *name* parameter.

3.2.3 Starting the TCP/IP task

We started our TCPIPA procedure using the MVS Start command: **S TCPIPA**

We were interested in seeing the following startup messages:

- `EZB6473I TCP/IP STACK FUNCTIONS INITIALIZATION COMPLETE.`
- `EZAIN11I ALL TCPIP SERVICES FOR PROC TCPIPA ARE AVAILABLE.`

3.2.4 Starting the OMPROUTE task

OMPROUTE can be started by using one of three methods:

- Using the MVS START command, specifying the proclib member name, `S OMPA`
- Using the z/OS OMVS shell **`omproute`** command specifying the program, **`omproute`**
- Using the AUTOLOG statement within the TCP/IP Profile, as we did:

```
AUTOLOG
   OMPA
ENDAUTOLOG
```

If the AUTOLOG statement is used, then do *not* reserve a port for OMPROUTE. OMPROUTE for OSPF does not listen on any port. If a port reservation *is* made for OMPROUTE supporting only OSPF *and* AUTOLOG is used to start OMPROUTE, then code NOAUTOLOG on the PORT reservation statement, as shown here:

```
PORT 520 UDP OMPROUTE NOAUTOLOG
```

OMPROUTE issues a number of startup messages. The important ones to look for are shown in Example 3-12.

Example 3-12 OMPROUTE startup messages

```
EZZ7800I OMPROUTE STARTING
EZZ7898I OMPROUTE INITIALIZATION COMPLETE
```

3.2.5 Verifying the TCP/IP environment

For a complete list of commands that can be used to manage the TCP/IP environment and their detailed syntax, refer to:

- *z/OS Communications Server: IP System Administrator's Commands*, SC31-8781
- *z/OS Communications Server: IP User's Guide and Commands*, SC31-8780

We used the following commands and tools to verify our TCP/IP environment:

- "Expected TCP/IP stack startup messages"
- "TSO PING command"
- "TSO TRACERTE command"
- "TSO NETSTAT command"
- "z/OS DISPLAY TCPIP NETSTAT command"
- "TELNET connections"
- "OSA takeover and takeback"

Expected TCP/IP stack startup messages

Successful TCP/IP startup can be verified by looking for the following groups of messages:

- "Startup messages for physical interfaces"
- "Startup messages for XCF interfaces"
- "Startup messages for the TCP/IP stack"

Startup messages for physical interfaces

Example 3-13 shows the expected successful startup messages for the OSAs.

Example 3-13 OSA startup messages

```
EZZ4313I INITIALIZATION COMPLETE FOR DEVICE OSA2080
EZZ4313I INITIALIZATION COMPLETE FOR DEVICE OSA20A0
EZZ4313I INITIALIZATION COMPLETE FOR DEVICE OSA20C0
EZZ4313I INITIALIZATION COMPLETE FOR DEVICE OSA20E0
```

Startup messages for XCF interfaces

Example 3-14 shows the expected successful startup messages for XCF (if CF is configured).

Example 3-14 XCF startup messages resulting from DYNAMICXCF

```
EZD1176I TCPIPA HAS SUCCESSFULLY JOINED THE TCP/IP SYSPLEX GROUP EZBTCPCS
EZZ4313I INITIALIZATION COMPLETE FOR DEVICE IUTIQDIO
```

Startup messages for the TCP/IP stack

Example 3-15 shows the expected successful startup messages for the TCPIPA stack.

Example 3-15 TCPIPA stack startup messages

```
EZB6473I TCP/IP STACK FUNCTIONS INITIALIZATION COMPLETE.
EZAIN11I ALL TCPIP SERVICES FOR PROC TCPIPA ARE AVAILABLE.
```

TSO PING command

Example 3-16 shows how we used the PING command.

Example 3-16 TSO PING to verify the IP environment

```
PING 10.10.1.230                          1
   CS V1R8: Pinging host 10.10.1.230
   Ping #1 response took 0.000 seconds.
   ***

PING 10.10.2.232                          2
   CS V1R8: Pinging host 10.10.2.232
   Ping #1 response took 0.000 seconds.
   ***
```

Note the following explanations for Example 3-16 (we pinged all of the addresses, but only show a couple here):

- 1: One of the VIPAs
- 2: One of the OSAs

TSO TRACERTE command

Example 3-17 shows how we used the TRACERTE command.

Example 3-17 TSO TRACERTE to verify the IP environment

```
TRACERTE 10.10.1.230                               1
   CS V1R8: Traceroute to 10.10.1.230 (10.10.1.230):
   1 10.10.1.230 (10.10.1.230)  0 ms  0 ms  0 ms
   ***
```

```
TRACERTE 10.10.3.233                              2
   CS V1R8: Traceroute to 10.10.3.233 (10.10.3.233):
   1 10.10.3.233 (10.10.3.233)  0 ms  0 ms  0 ms
   ***
```

Note the following explanation for Example 3-17 (we did a TRACERTE to all of the addresses, but only show a couple here):

- 1: One of the VIPAs
- 2: One of the OSAs

TSO NETSTAT command

Example 3-18 shows how we used the NETSTAT command.

Example 3-18 TSO NETSTAT to verify the IP environment

```
NETSTAT HOME                                                            1
   MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPIPA          05:07:22
   Home address list:
   LinkName:   VIPALINK                                                 2
     Address:  10.10.1.230
       Flags:  Primary
   LinkName:   OSA2080LNK                                               3
     Address:  10.10.2.232
       Flags:
   . . .
   LinkName:   EZAXCF31
     Address:  10.10.20.100
       Flags:
   . . .
NETSTAT DEVLINKS     4
NETSTAT CONFIG       5
NETSTAT CONN         6
NETSTAT ROUTE        7
NETSTAT PORTLIST     8
```

Note the following explanation for Example 3-18:

- 1: The NETSTAT HOME command generated output showing the following interfaces:
  - 2: The static VIPA for the stack
  - 3: The OSAs
  - 6: The XCF interface generated by DYNAMICXCF

We issued the other commands listed, but did not show the output:

- 4: DEVLINKS shows all the devices and their status
- 5: CONFIG shows miscellaneous stack settings
- 6: CONN shows all connections (clients and servers)
- 7: ROUTE shows all routes (defined and learned) including the OSPF default routes
- 8: PORTLIST shows or port reservations for EE and others if reserved

z/OS DISPLAY TCPIP NETSTAT command

Example 3-19 shows how we used the z/OS DISPLAY TCPIP command.

Example 3-19 z/OS DISPLAY TCPIP NETSTAT to verify the IP environment

```
D TCPIP,TCPIPA,N,HOME
```

```
D TCPIP,TCPIPA,N,ROUTE
D TCPIP,TCPIPA,N,STATS     1
```

Note the following explanation for Example 3-19:

- 1: The same NETSTAT commands can be entered as z/OS systems commands. One helpful command is NETSTAT STATS. It returns much information about the TCP/IP environment.

TELNET connections

Assuming a TN3270 server is actively listening on the mainframe, a workstation that has a 3270 emulator program and has connectivity to the mainframe can be used to connect to the TN3270 server. Connectivity can be confirmed when the expected TN3270 server *hello* screen is displayed on the workstation.

OSA takeover and takeback

An OSA can take over ARP assist responsibility for another failed OSA in the same subnet. This capability can be tested by *stopping* an OSA and watching for the takeover message issued by TCP/IP. When the *failed* OSA recovers, TCP/IP issues a message. The failure and recovery processes are discussed in:

- “OSA failure shows takeover”
- “OSA recovery shows takeback”

OSA failure shows takeover

Example 3-20 shows the OSA takeover message sequence.

Example 3-20 OSA takeover messages

```
V TCPIP,TCPIPA,STOP,OSA2080                                    1
   EZZ0060I PROCESSING COMMAND: VARY TCPIP,TCPIPA,STOP,OSA2080
   EZZ0053I COMMAND VARY STOP COMPLETED SUCCESSFULLY
   EZZ4329I LINK OSA20C0LNK HAS TAKEN OVER ARP RESPONSIBILITY FOR  2
            INACTIVE LINK OSA2080LNK
   EZZ4315I DEACTIVATION COMPLETE FOR DEVICE OSA2080

PING 10.10.2.232
   CS V1R8: Pinging host 10.10.2.232                           3
   Ping #1 response took 0.000 seconds.
   ***
```

Note the following explanation for Example 3-20:

- 1: The OSA2080 device is stopped, (a simulated failure).
- 2: Device OSA20C0 takes over the ARP responsibility for the failed device.
- 3: A ping from another system to the failed OSA’s IP address still gets a response.

OSA recovery shows takeback

Example 3-21 shows the recovery message.

Example 3-21 OSA recovery message

```
V TCPIP,TCPIPA,START,OSA2080                                               1
   EZZ0060I PROCESSING COMMAND: VARY TCPIP,TCPIPA,START,OSA2080
   EZZ0053I COMMAND VARY START COMPLETED SUCCESSFULLY
   IEF196I IEF237I 2081 ALLOCATED TO TP2081
   IEF196I IEF237I 2080 ALLOCATED TO TP2080
   IEF196I IEF237I 2082 ALLOCATED TO TP2082
   EZD0013I LINK OSA2080LNK HAS TAKEN BACK ARP RESPONSIBILITY FROM LINK 2
   OSA20C0LNK
   EZZ4313I INITIALIZATION COMPLETE FOR DEVICE OSA2080                     3
```

Note the following explanation for Example 3-21:

- When the OSA2080 device is restarted (1), it takes back Address Resolution Protocol (ARP) responsibility (2), and re-initializes (3).

3.2.6 Verifying the OSPF environment

For a complete list of commands that can be used to manage the OSPF environment and their detailed syntax, refer to:

- *z/OS Communications Server: IP System Administrator's Commands*, SC31-8781
- *z/OS Communications Server: IP User's Guide and Commands*, SC31-8780

We used the following commands and tools to verify our OSPF environment:

- "Expected OMPROUTE startup messages"
- "TSO NETSTAT ROUTE command"
- "z/OS DISPLAY TCPIP,OMP,RTTABLE command"
- "OSA takeover and takeback"
- "Sysplex autonomics"

Expected OMPROUTE startup messages

Example 3-22 shows the expected successful OMPROUTE startup messages.

Example 3-22 OMPROUTE startup messages

```
EZZ7800I OMPROUTE STARTING
EZZ7898I OMPROUTE INITIALIZATION COMPLETE
```

TSO NETSTAT ROUTE command

Example 3-23 shows how we used the TSO NETSTAT ROUTE command.

Example 3-23 TSO NETSTAT ROUTE to verify the OSPF environment

```
NETSTAT ROUTE                                                              1
   MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPIPA          06:10:32
   IPv4 Destinations
   Destination         Gateway          Flags     Refcnt  Interface
   -----------         -------          -----     ------  ---------
   Default             10.10.2.1        UGO       000000  OSA20C0LNK       2
   Default             10.10.3.2        UGO       000000  OSA20A0LNK
   Default             10.10.3.2        UGO       000000  OSA20E0LNK
   Default             10.10.2.1        UGO       000000  OSA2080LNK
   10.10.1.220/32      10.10.2.224      UGHO      000000  OSA20C0LNK
   10.10.1.220/32      10.10.2.222      UGHO      000000  OSA20C0LNK
........
```

Note the following explanation for Example 3-23:

- 1: The TSO NETSTAT ROUTE command shows route information.
- 2: OSPF has informed the stack of the four default routes, one for each OSA.

z/OS DISPLAY TCPIP,OMP,RTTABLE command

Example 3-24 shows how we used the z/OS DISPLAY TCPIP,OMPR,RTTABLE command.

Example 3-24 z/OS DISPLAY TCPIP OMP to verify the OSPF environment

```
D TCPIP,TCPIPA,OMP,RTTABLE                                                 1
   EZZ7847I ROUTING TABLE 592
   TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

   SPIA   0.0.0.0                 0  11      646     10.10.2.1         (4)  2
    DIR*  10.10.1.0        FFFFFF00  1       9901    10.10.1.230       (3)
    SPF   10.10.1.220      FFFFFFFF  20      646     10.10.2.224       (8)  3
    DIR*  10.10.1.230      FFFFFFFF  1       9901    VIPALINK               4
    SPF   10.10.1.240      FFFFFFFF  20      646     10.10.2.244       (8)  5
    SPF*  10.10.2.0        FFFFFF00  10      646     OSA2080LNK        (2)  6
    SPF*  10.10.3.0        FFFFFF00  10      9889    OSA20A0LNK        (2)  7
   STAT*  10.10.20.0       FFFFFF00  0       9902    10.10.20.100           8
   STAT*  10.10.20.101     FFFFFFFF  0       9306    10.10.20.100
   STAT*  10.10.20.102     FFFFFFFF  0       9202    10.10.20.100

   DEFAULT GATEWAY IN USE.

   TYPE COST     AGE       NEXT HOP
   SPIA 11       646       10.10.2.1        (4)
                          0 NETS DELETED, 1 NETS INACTIVE
```

The display of the OSPF Routing Table, RTTABLE, shows the supported OSPF routes. Note the following explanation for Example 3-24:

- 1: The display command
- 2: The four default routes through the four OSAs
- 3: Routes to SC32’s static VIPA
- 4: SC30’s own static VIPA

- 5: Routes to SC31’s static VIPA
- 6: OSA interfaces to VLAN10
- 7: OSA interfaces to VLAN20
- 8: XCF interfaces

OSA takeover and takeback

An OSA can take over ARP assist responsibility for another failed OSA in the same subnet. This capability can be tested by *stopping* an OSA and monitoring the OSPF routing table for missing routes through the inactive OSA. When the *failed* OSA recovers, OSPF learns of the active interface and reestablishes routes through it.

The failure and recovery processes are discussed in:

- “OSA failure shows absence of default route”
- “OSA recovery shows default route re-established”

OSA failure shows absence of default route

Example 3-25 shows the absence of the OSA2080LNK when OSA2080 has failed.

Example 3-25 OSA failure, default route missing

```
D TCPIP,TCPIPA,N,ROUTE
   EZD0101I NETSTAT CS V1R8 TCPIPA 564
   IPV4 DESTINATIONS
   DESTINATION        GATEWAY          FLAGS     REFCNT  INTERFACE
   DEFAULT            10.10.2.1        UGO       000000  OSA20C0LNK
   DEFAULT            10.10.3.2        UGO       000000  OSA20A0LNK
   DEFAULT            10.10.3.2        UGO       000000  OSA20E0LNK
   10.10.1.220/32     10.10.2.224      UGHO      000000  OSA20C0LNK
   10.10.1.220/32     10.10.2.222      UGHO      000000  OSA20C0LNK
........
```

OSA recovery shows default route re-established

Example 3-26 shows the OSA2080LNK recovered, and the default route re-established.

Example 3-26 OSA recovery, default route re-established

```
D TCPIP,TCPIPA,N,ROUTE
   EZD0101I NETSTAT CS V1R8 TCPIPA 611
   IPV4 DESTINATIONS
   DESTINATION        GATEWAY          FLAGS     REFCNT  INTERFACE
   DEFAULT            10.10.2.1        UGO       000000  OSA20C0LNK
   DEFAULT            10.10.3.2        UGO       000000  OSA20A0LNK
   DEFAULT            10.10.3.2        UGO       000000  OSA20E0LNK
   DEFAULT            10.10.2.1        UGO       000000  OSA2080LNK
   10.10.1.220/32     10.10.2.224      UGHO      000000  OSA20C0LNK
   10.10.1.220/32     10.10.2.222      UGHO      000000  OSA20C0LNK
........
```

Sysplex autonomics

If sysplex autonomics are enabled, it is very important that the WLM policy for the OMPROUTE address space receives sufficient resources in relationship to other work being managed on the system. Under high load conditions it is possible that OMPROUTE, if not properly classified, can trigger an autonomic response from the TCP/IP stack it has affinity with, resulting in the TCP/IP address space removing itself from the sysplex group.

3.3 Preparing VTAM to participate in an APPN/HPR network

> **Important:** It is recommended that the TCP/IP and OMPROUTE address spaces be placed in the SYSSTC service classification. Classification in another service class will leave the system vulnerable to a sysplex distributor outage.

This section shows how to configure VTAM as a pure APPN node in an APPN/HPR network with parallel *Transmission Groups (TG)* consisting of *Multi Path Channel (MPC)* connections and *Cross Coupling Facility (XCF)* connections.

We also describe the concepts of APPN route calculation based on *APPN Class of Service (APPNCOS)* and how *High Performance Routing (HPR)* pipes switch around failing entities without user session interruptions.

> **Note:** Many installations may already run APPN in their mainframe environment. However, we still recommend reading this section to get an overview of the options that are available. They could improve availability, manageability, or performance in your existing environment.

3.3.1 Modifying the VTAM start procedure

In preparation to the EE implementation, VTAM's start procedure needs some modifications. APPN NNs can checkpoint topology and directory databases to datasets, so that the information does not get lost during a VTAM restart.

APPN checkpoint dataset allocations

The APPN checkpoint datasets need to be allocated first.

Example 3-27 shows the Checkpoint Datasets used in our environment.

Example 3-27 APPN Checkpoint Datasets

```
SYS1.VTAM.SC30.DSDBCTRL
SYS1.VTAM.SC30.DSDB1
SYS1.VTAM.SC30.DSDB2
SYS1.VTAM.SC30.TRSDB
SYS1.VTAM.SC31.DSDBCTRL
SYS1.VTAM.SC31.DSDB1
SYS1.VTAM.SC31.DSDB2
SYS1.VTAM.SC31.TRSDB
SYS1.VTAM.SC32.DSDBCTRL
SYS1.VTAM.SC32.DSDB1
SYS1.VTAM.SC32.DSDB2
SYS1.VTAM.SC32.TRSDB
```

Note in Example 3-27 that the datasets are allocated for all system, even though SC32 is started as an EN. Doing so we are able to share the same start procedure for all VTAMs by specifying the &SYSNAME. symbolic in the DD cards.

New data definition (DD) statements in VTAM's procedure

Example 3-28 shows the new DD cards pointing to the checkpoint datasets.

Example 3-28 Modification to VTAM's start procedure

```
//* TRSDB     NEEDED FOR APPN CHECKPOINTING OF TRS DB
//*           RECFM=FB,LRECL=1000,BLKSIZE=NX1000,DSORG=PS
//TRSDB    DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..TRSDB 1
//* DSDBCTRL  NEEDED FOR APPN CHECKPOINTING OF DIRECTORY DB
//*           RECFM=FB,LRECL=20,BLKSIZE=20,DSORG=PS
//DSDBCTRL DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..DSDBCTRL 2
//DSDB1    DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..DSDB1 3
//DSDB2    DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..DSDB2
```

See the following notes to Example 3-28:

- **1**: TRSDB is the DD statement for the Topology Database. The DCBs of that dataset are:
  - Organization . . . : PS
  - Record format . . . : FB
  - Record length . . . : 1000
  - Block size . . . . : 27000
  - 1st extent tracks . : 100
- **2**: The DSDBCTRL dataset has the following allocation DCBs:
  - Organization . . . : PS
  - Record format . . . : FB
  - Record length . . . : 20
  - Block size . . . . : 20
  - 1st extent tracks . : 1
  - Secondary tracks . : 1
- **3**: The DSDB1 and DSDB2 datasets have following allocation DCBs:
  - Organization . . . : PS
  - Record format . . . : FB
  - Record length . . . : 1000
  - Block size . . . . : 27000
  - 1st extent tracks . : 100
  - Secondary tracks . : 10

3.3.2 Modifying the VTAM start options

This section describes the VTAM start options used in our scenarios. It is divided into three parts.

1. Common APPN parameters that are applicable to all VTAMs and are coded in ATCSTR00
2. Network Node (NN) specific parameters that are coded in ATCSTRNN
3. End Node (EN) specific parameters that are coded in ATCSTREN

We are using system symbolics wherever possible to be able to share common start lists between VTAMs.

All VTAMs will be started with **`S VTAM,,,LIST=`**`(nodetype)`. These are specific start options that will be read in first, followed by the general purpose start list (ATCSTR00), which apply to all VTAM configurations.

Common APPN parameters

Common APPN parameters are start options that can be coded on any APPN node (see “General APPN related start options” on page 105). They are coded in a general purpose start list (ATCSTR00), which is used to complete otherwise undefined start options after the more specific start (NN or EN) list has been processed.

Example 3-29 General APPN related start options

```
CONNTYPE=APPN,                                                         X
CPCP=YES,                                                              X
DIALRTRY=NO,                                                           X
DUPDEFS=NONE,                                                          X
DYNADJCP=YES,                                                          X
DYNLU=YES,                                                             X
HOSTSA=&SYSCLONE.,SACONNS=NO,                                          X
IOPURGE=120,                                                           X
NETID=RDBOOKEE,SSCPNAME=&SYSNAME.M,                                    X
SAVERSCV=YES,                                                          X
TRACE,TYPE=VTAM,MODE=INT,SIZE=999,DSPSIZE=5,OPT=(CIA,HPR),             X
XCFINIT=YES,                                                           X
```

CPCP

An APPN environment relies on CP-CP sessions across APPN TGs between the APPN nodes to perform searches and topology information exchange. This start option allows you to control whether you will globally allow CP-CP sessions to automatically setup whenever an APPN TG is activated. This parameter can be overridden at a PU level. In our environment we set CPCP=YES to allow for CPCP sessions on any available link.

DIALRTRY and DUPDEFS

The nature of APPN enables resources to be dynamically found. One of the consequences of this is that CPCP sessions may get very busy because of APPN broadcast searches occurring. We recommend setting DIALRTRY=NO and DUPDEFS=NONE to help reduce the amount of APPN broadcasts for connectable resources (applications and LUs in CONCT status) in the network. Only on the designated primary and backup DLUS we recommend setting DIALRTRY=YES to be able to have the SWNET major nodes active on both DLUS to simplify takeover scenarios.

DYNLU and DYNADJCP

One of the benefits of APPN is that resources can be dynamically found and do not have to be predefined. If for security reasons, you do not want this to happen, DYNLU and DYNADJCP can be set to prevent the creation of dynamic LUs and *Adjacent Control Points (ADJCP)*. Be aware that you have to predefine each LU and ADJCP in that case.

DYNMODTB

In order to select the desired priority for a session, it is important to assign the correct APPNCOS to a logmode. In an APPN environment, you must keep all your logmodes in one single table if you are using dynamic *Cross Domain Resources (CDRSC)*. DYNMODTB should point to that single logmode table so that VTAM can resolve the logmode to the desired APPNCOS. See 3.4, “Consolidating LOGMODEs” on page 114 for more information about LOGMODE and APPNCOS resolution.

ENHADDR

There is an architectural limit of 64 K element numbers per subarea. Even with VTAM running as a pure APPN node, internally all resources are represented using a FID4 subarea address format. Many types of resources can now make use of high element numbers in VTAM's subarea, thus freeing up space in the limited below 64 K element address space. We recommend setting ENHADDR=YES to provide for better scalability.

HOSTSA and SACONNS

If VTAM was already running as a subarea node with HOSTSA coded, adding the NODETYPE start option results in VTAM being a hybrid node with both subarea and APPN functionality, usually referred to as an *Interchange Node (ICN)* or *Migration Data Host (MDH)*.

For better manageability we recommend defining or keeping a unique subarea number using HOSTSA, and running VTAM as a pure EN/NN by coding SACONNS=NO.

IOPURGE

In the case of an APPN node not responding to APPN Locates, the IOPURGE parameter sets a threshold to time out outstanding session requests and fail the session rather than waiting forever. We recommend that IOPURGE be set to a reasonable value; for example, 120 through 129.

NETID and SSCPNAME

A node's identity in an APPN environment is uniquely determined by the so called *Fully Qualified Name (FQN)* a combination of the *Network Identifier (NETID)* and the *Control Point Name (CPNAME)*. Therefore, NETID.CPNAME has to be unique in the APPN environment, as an IP address must be unique within an IP network. NETID and SSCPNAME are the VTAM start options that define its identity in the APPN environment. We recommend using system symbolics to build the SSCPNAME so that it can be related to the system it is representing.

NOTE: APPN ENs can have a NETID different from their NN and still belong to the same APPN Subnet. In some cases this can be done instead of using a more complicated *Extended Border Node (EBN)* configuration.

SAVERSCV

The *Route Selection Control Vector (RSCV)* defines the path through an APPN environment that a BIND will take at session setup. We recommend setting SAVERSCV=YES to preserve this information for better diagnostics in case there is a problem with the session later on.

TRACE

The VTAM Internal Trace provides important information to document VTAM problems and in most cases it is required to report a problem to IBM support. It is usually tracing into a 4 MB storage area in ECSA with a number of default options including PSS and SMS. We recommend tracing into a 50 MB data space to provide for a larger time frame and to activate HPR and CIA options to capture IP packets and *Network Layer Packets (NLP)*. In case of a problem a CSDUMP should be taken that will automatically dump the VIT data space.

If you are concerned about CPU overhead, turn off PSS and SMS instead after VTAM start using the **F NET,NOTRACE,TYPE=VTAM,OPT=**(PSS,SMS). Those trace options are active by default to document internal VTAM processing and are very talkative.

XCFINIT

VTAMs can establish dynamic APPN connections to each other through XCF if they are running in the same s*ysplex*. We recommend activating dynamic XCF links between all VTAMs at startup by coding XCFINIT=YES. This way we automatically get APPN connectivity within the sysplex and do not have to configure any other major nodes.

Network Node parameters

Network Node parameters are start options that we used for our VTAM Network Nodes (see Example 3-30). They are all coded in a NN-specific start list (ATCSTRNN), which is selected in the LIST=NN parameter during VTAM start.

Example 3-30 Network Node specific start options

```
NODETYPE=NN,                                                           X
CDSERVR=YES,                                                           X
DIALRTRY=YES,                                                          X
INITDB=DIR,                                                            X
CSALIMIT=0
```

NODETYPE

VTAM can either be started as a *Network Node (NN)* or as an *End Node (EN)* getting services from a NN. As a general rule, the number of NNs in an APPN environment should be kept to a minimum to reduce the amount of APPN flows in the network, but still be high enough to provide the required redundancy to avoid any single points of failure. This means that every EN should have a backup NN in case the primary NN fails. NODETYPE (EN or NN) determines VTAM's role in the APPN environment. When you plan to implement *VTAM Generic Resources (GR)* we recommend having at least one NN per sysplex to provide the *Resource Selector* function.

Recommendation: Keep the number of Network Nodes (NN) in the APPN environment to a minimum. For high availability every sysplex should have at least two NNs. However, if there is only one NN in your sysplex and the backup NN is not part of that sysplex, then set ENBCAST=YES in the ENs' NETSRVR list.

CDSERVR

To improve efficiency in APPN searching, NNs can take on the role of a *Central Directory Server (CDS)*. If there is a CDS available in the APPN environment, all NNs must ask the CDS to find an unknown LU. The CDS will check its own database and if necessary, send out an APPN broadcast to find the resource. If successful, it will store the new information in its database to be reused in case another NN asks for the same LU later on. After some time, the CDS will have knowledge of all resources, that were being searched and found and therefore, will not have to broadcast for any existing resources. Since no other NN except the CDS is allowed to broadcast searches, the number of APPN broadcast searches will be kept to a minimum. An APPN environment can have multiple CDS, we recommend keeping the number of CDS reasonably small. In most cases two Central Directory Servers will be enough to provide for efficient searching in APPN.

INITDB

APPN requires two databases. The *APPN Topology and Routing Services (TRS) Database* contains all NNs and the links connecting those NNs. This information is required by every NN to calculate routes for sessions and every NN needs to have the same information. When a new APPN NN joins the network, it will receive the complete APPN topology from its neighbor NNs via *Topology Database Updates (TDU)* over the CP-CP sessions. In large networks this can take a considerable amount of time and consume networking bandwidth if the adjacent NNs are geographically dispersed.

The *APPN Directory Database* contains information about *Logical Units (LU)* and their location. The information is used to send directed *APPN LOCATE* requests during session setup and helps avoid *APPN BROADCAST* searches. Unlike the topology database, every NN keeps its own directory database, which means, it is different in every NN. As more and more sessions set up, the Directory Database gets populated with more and more resources and will therefore contain valuable information that would be worth keeping over a VTAM restart.

VTAM provides the possibility of check pointing both the topology and the directory database to datasets and reading them from disk again during start, thus making use of already learned information and reducing the number of TDU and BROADCAST flows in the network.The INITDB start option controls which databases will be pre-loaded during VTAM startup.

We recommend coding INITDB=DIR on the Central Directory Server(s) as the number of NNs is fairly small in our network and check pointing the topology database is therefore not beneficial.

CSALIMIT

The generally accepted concept of *Communication Management Configuration (CMC)* still applies to APPN environments and we recommend not running any business applications on Network Node Servers. As such, a NN VTAM does not have to compete for ECSA storage with other subsystems and you should not limit VTAM's ECSA usage there. Therefore we recommend coding CSALIMIT=0 on network owning VTAMs.

DIALRTRY

In a high availability network design Dependent LU Requestors (DLUR) should define a primary and backup Dependent LU Server (DLUS). The switched major nodes defining the dependent LUs should be active on both DLUS to simplify and speed up takeover scenarios. The PU and LUs on the backup DLUS show CONCT status as long as the primary DLUS is available. During this time it must still be possible to establish sessions, if a session request for a CONCT LU arrives at the backup DLUS. With DIALRTRY=YES coded, VTAM will send out an APPN Broadcast to find an active instance of the destination LU and the primary DLUS will receive the session request instead.

End Node specific parameters

End Node parameters are start options that we used for our VTAM End Nodes (see Example 3-31). They are all coded in an EN-specific start list (ATCSTREN), which will be selected via the LIST=NN parameter during VTAM start.

Example 3-31 End Node specific start options

```
NODETYPE=EN,                                                           X
NNSPREF=SC30M,                                                         X
CSALIMIT=200M
```

CSALIMIT

A VTAM EN typically hosts business applications that require enough ECSA storage to perform their tasks efficiently. We recommend limiting the amount of ECSA storage to be used by VTAM on ENs to avoid infecting other subsystems in case that VTAM suffers a storage problem. You should monitor the VTAM's usage regularly to determine the normal demand, especially during a migration to APPN and adjust the limit as needed. CSALIMIT can be modified dynamically.

Tip: The `D NET,BFRUSE,BUFFER=SUMMARY` command can be used to monitor VTAM's CSA demand.

NNSPREF

An EN must receive services from a Network Node server to successfully participate in the APPN environment. It therefore must establish a pair of LU6.2 sessions, referred to as CP-CP sessions to one NN. It can only have one single NN server at any point in time. If this NN fails, the EN tries to establish CP-CP sessions to another available NN. For high availability reasons an EN should always have connections to two Network Nodes, the primary and backup NN. For easier management, we recommend that an EN revert back to its primary NN after it has recovered from a failure.

3.3.3 Adding VTAM major nodes

In the previous section we described the start options, that enable VTAM's APPN function. Now we are defining the connections Transmission Groups (TG) between the VTAMs in our environment and assigning characteristics to them. At the end of this step we have two parallel TGs between each LPAR, XCF links using TG21 to be used for interactive sessions, and MPC connections using TG2 to be used for medium and low priority sessions. This is achieved by assigning different weights to the transmission groups, based on the APPN Class of Service.

Transmission Group Profiles

A *Transmission Group Profile (TGP)* defines a set of TG characteristics that can be assigned to a link, or multiple links for example, capacity, security properties, and user defined parameters. IBM supplies a list of commonly used characteristics in SYS1.SAMPLIB member IBMTGPS. The latest version of IBMTGPS should be copied into VTAMLST and Activated via ATCCON. D NET,TGPS lists all Transmission Group Profiles that VTAM knows about and can therefore be coded on the PU statements.

Example 3-32 shows the TGPS after activation of the latest IBMTGPS from SYS1.SAMLIB.

Example 3-32 Display of the active Transmission Group Profiles

```
V NET,ACT,ID=IBMTGPS
IST097I VARY ACCEPTED
IST093I IBMTGPS ACTIVE

D NET,TGPS
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TGPS 010
IST1107I TGP NAME TG CHARACTERISTICS
IST1108I ETHERNET 00800000000000000000014C00808080
IST1108I TOKNRING 00750000000000000000014C00808080
IST1108I ISDNNSWT 00450000000000000000017100808080
IST1108I ISDNSWTD 00300000000000808000017100808080
IST1108I SDLCNSWT 00300000000000000000017100808080
IST1108I SDLCSWTD 00300000000000808000017100808080
IST1108I X25      00300000000000808000209100808080
IST1108I TRING16M 00850000000000000000014C00808080
IST1108I CHANNEL  008E0000000000000000604C00808080
IST1108I ESCON    00A20000000000000000604C00808080
IST1108I XCF      008A0000000000000000604C00808080
...
IST1108I EEXTCAMP 00750000000000000000017100808080
IST1108I EEXTWAN  00430000000000000000209100808080
IST1108I FASTENET 009A0000000000000000014C00808080
IST1108I GIGENET  00B40000000000000000014C00808080
```

```
IST1108I FICON    00AF0000000000000000014C00808080
IST1108I FICONEXP 00B40000000000000000014C00808080
IST1108I HIPERSOC 00B40000000000000000014C00808080
IST1454I 26 TGP(S) DISPLAYED
IST314I END
```

In the display in Example 3-32, the second byte in the TG characteristics is the hexadecimal representation of the capacity. We now have a set of characteristics defined that we can later apply to our APPN link definitions by referring to a profile with the TGP parameter on the PU statement.

> **Important:** The Transmission Group Profile member must be active before any PU with a TGP parameter is activated. Make sure to place it to the top of your ATCCONxx member.

APPN Class of Service table

APPN route calculation is done based on a *Class of Service table (COSTAB)*. In the COSTAB, for every *APPN Class of Service (APPNCOS)* there is a number of lines that map a certain set of characteristics to a weight. During route calculation the weights of all nodes and TGs along a possible session path are added and the least weight route will be selected. For more information about this algorithm see 12.1, “APPN route selection” on page 376.

VTAM initially shipped a COSTAB that contained 8 lines per APPNCOS and treats every link with a capacity of 4M as the best choice. A new COSTAB is available in SYS1.SAMPLIB(ISTACST2) that provides more granularity as it supplies 12 rows per APPNCOS instead of 8. We recommend copying this table into VTAMLST and activating it via ATCCONxx.

Example 3-33 shows the activation of the new Class of Service table ISTACST2.

Example 3-33 Activating the new Class of Service table

```
V NET,ACT,ID=ISTACST2
IST097I VARY ACCEPTED
IST093I ISTACST2 ACTIVE

D NET,COS,TYPE=APPN
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = APPN COS 282
IST1782I ENTRY NAME     TABLE NAME     ACTIVATION TIME
IST1783I CPSVCMG        ISTACST2       10/20/06  15:18:49
IST1783I SNASVCMG       ISTACST2       10/20/06  15:18:49
IST1783I #CONNECT       ISTACST2       10/20/06  15:18:49
IST1783I #INTER         ISTACST2       10/20/06  15:18:49
IST1783I #INTERSC       ISTACST2       10/20/06  15:18:49
IST1783I #BATCH         ISTACST2       10/20/06  15:18:49
IST1783I #BATCHSC       ISTACST2       10/20/06  15:18:49
IST314I END
```

These APPNCOS entries are available on all APPN platforms. We recommend using these entries whenever possible instead of creating new ones. If new entries are created they have to be distributed to all other platforms.

Initial APPN topology: XCF connectivity only

Because of XCFINIT=YES in the start options, we already have implemented an APPN environment consisting of three nodes connected to each other through XCF links. A highly available APPN environment demands an additional link between the nodes. The most commonly used type of connection in such a scenario is a channel connection using Multi Path Channel (MPC) protocol.

Adding an additional link: MPC channel connections

MPC channels can group multiple READ and WRITE channels to a single transmission group between two systems and so provide a very robust and fast connection between VTAMs. A Transport Resource List Element (TRLE) defines the channel unit addresses for read and write, an additional local SNA major node defines the PU that refers to this TRLE.

TRL Major Node

Example 3-34 lists the Transport Resource List major node definition on SC30M.

Example 3-34 TRLE major node on SC30M

```
***********************************************************************
* TRANSPORT RESOURCE LIST MAJOR NODE FOR APPN CONNECTION ACROSS FCTC
***********************************************************************
TRL4SC30 VBUILD TYPE=TRL
TRL2SC31 TRLE  LNCTL=MPC,       MPC FICON channel TO SC31M             *
               MAXBFRU=16,                                             *
               READ=(5842,5843),                                       *
               WRITE=(4842,4843)
TRL2SC32 TRLE  LNCTL=MPC,       MPC FICON channel TO SC32M             *
               MAXBFRU=16,                                             *
               READ=(5852,5853),                                       *
               WRITE=(4852,4853)
```

The TRL major node contains the Transport Resource List Elements (TRLE) that define the READ and WRITE devices to be used for communication. The READ devices in this system must be connected to the WRITE devices of the adjacent system.

LOCAL SNA major node for MPC

Example 3-35 lists the local SNA major node referring to the TRLE on SC30M.

Example 3-35 Local SNA major node on SC30M

```
***********************************************************************
* LOCAL MAJOR NODE FOR MPC CONNETIONS TO SC31M AND SC32M
***********************************************************************
MPC4SC30 VBUILD TYPE=LOCAL
MPC2SC31 PU    PUTYPE=2,XID=YES,                                       *
               NETID=RDBOOKEE,CPNAME=SC31M,TGN=2,     * MPC = TG2      *
               TGP=FICON,                                              *
               TRLE=TRL2SC31
MPC2SC32 PU    PUTYPE=2,XID=YES,                                       *
               NETID=RDBOOKEE,CPNAME=SC32M,TGN=2,      * MPC = TG2     *
               TGP=FICON,                                              *
               TRLE=TRL2SC32
```

As shown Example 3-35, we define the PU with NETID, CPNAME, and TGN to assign a unique TG number (TGN=2) to the MPC connections. This helps to simplify network management in the APPN environment as we can already tell from a topology display, which DLC type the transmission group is using.

3.3.4 Verifying the APPN topology

We now have a redundant APPN environment consisting of one End Node (SC32M) and two Network Nodes (SC30M and SC31M). All systems have two parallel TGs to connect to each other. XCF links use Cross Coupling Facility connectivity and are assigned a TG21. MPC links use FICON® channels and are assigned a TG2 (see Figure 3-1 on page 88).

Display of the APPN topology with XCF and MPC links

Example 3-36 shows the commands that verify our connectivity between the LPARs.

Example 3-36 The APPN Topology after MPC activation

```
D NET,TOPO,ID=SC30M,LIST=ALL 1
IST1295I CP NAME          NODETYPE ROUTERES CONGESTION  CP-CP WEIGHT
IST1296I RDBOOKEE.SC30M    NN       128      NONE        *NA*  *NA*
IST1579I                   ------------------------------------------
IST1297I                   ICN/MDH  CDSERVR  RSN         HPR
IST1298I                   NO       YES      2           RTP
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP RDBOOKEE.SC30M
IST1357I                                                 CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.SC31M    2        OPER     INTERM      YES   *NA*
IST1301I RDBOOKEE.SC31M    21       OPER     INTERM      YES   *NA*
IST1301I RDBOOKEE.SC32M    21       OPER     ENDPT       YES   *NA*
IST1301I RDBOOKEE.SC32M    2        OPER     ENDPT       YES   *NA*

D NET,TOPO,ID=SC31M,LIST=ALL 2
IST350I DISPLAY TYPE = TOPOLOGY 555
IST1295I CP NAME          NODETYPE ROUTERES CONGESTION  CP-CP WEIGHT
IST1296I RDBOOKEE.SC31M    NN       128      NONE        *NA*  *NA*
IST1579I                   ------------------------------------------
IST1297I                   ICN/MDH  CDSERVR  RSN         HPR
IST1298I                   NO       YES      2           RTP
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP RDBOOKEE.SC31M
IST1357I                                                 CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.SC32M    2        OPER     ENDPT       YES   *NA*
IST1301I RDBOOKEE.SC32M    21       OPER     ENDPT       YES   *NA*
IST1301I RDBOOKEE.SC30M    2        OPER     INTERM      YES   *NA*
IST1301I RDBOOKEE.SC30M    21       OPER     INTERM      YES   *NA*
IST314I END
```

Note the following explanation for Example 3-36:

- 1: This display shows the connections originating from SC30M. We see that we have TG2 and TG21 operational to both SC31M and SC32M.
- 2: This display shows the connections originating from SC31M. We see that we have TG2 and TG21 operational to both SC32M and SC30M.

Starting sessions using APING

Using the APING command, we are now verifying the connections. Example 3-37 shows a session setup using an APPNCOS of #INTER.

Example 3-37 APING with LOGMODE=#INTER

```
D NET,APING,ID=SC32M,LOGMODE=#INTER
IST097I DISPLAY ACCEPTED
IST1489I APING SESSION INFORMATION 871
IST1490I DLU=RDBOOKEE.SC32M SID=DE1FD92BF340E016
IST933I LOGMODE=#INTER  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #INTER
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I  21  RDBOOKEE.SC32M     APPN         RTP
IST314I END
```

The #INTER session used the XCF link TG21.

Example 3-38 shows an APING with LOGMODE=ISTCOSDF to use a #CONNECT class of service.

Example 3-38 APING using ISTCOSDF LOGMODE using MPC link TG 2

```
D NET,APING,ID=SC32M,LOGMODE=ISTCOSDF 1
IST097I DISPLAY ACCEPTED
IST1489I APING SESSION INFORMATION 883
IST1490I DLU=RDBOOKEE.SC32M SID=DE1FD92BF340E018
IST933I LOGMODE=ISTCOSDF, COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #CONNECT
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC32M     APPN         RTP 2
IST314I END
```

Note the following explanation for Example 3-38:

- 1: As there is no #CONNECT logmode in ISTINCLM, ISTCOSDF can be used to set up a session over a #CONNECT HPR pipe.
- 2: The ISTCOSDF session uses the MPC link TG2.

Example 3-39 shows an APING with LOGMODE=#BATCH using a #BATCH class of service.

Example 3-39 APING using #BATCH LOGMODE

```
D NET,APING,ID=SC32M,LOGMODE=#BATCH
IST097I DISPLAY ACCEPTED
IST1489I APING SESSION INFORMATION 832
IST1490I DLU=RDBOOKEE.SC32M SID=DE1FD92BF3B33117
IST933I LOGMODE=#BATCH  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #BATCH
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC32M     APPN         RTP
IST314I END
```

The #BATCH session uses the MPC link TG2.

3.4 Consolidating LOGMODEs

This section describes why a logmode consolidation should be performed as one of the first steps in an APPN migration. We also provide a method for conducting the consolidation process with examples.

Consideration: If you decide to skip this step, sessions may use an APPN class of service with undesirable routes.

3.4.1 Priorities in APPN

SNA subarea has always had the capability of prioritizing traffic based on the type of user session. This way, high volume batch traffic cannot interfere with interactive traffic, causing response times on those sessions to increase. The priority was chosen by the COS operand in the LOGMODE used for the session.

APPN also provides a mechanism to use different priorities for different types of sessions. APPNCOS operand in the LOGMODE definition controls the priority used for a session. In APPN there are four priorities, NETWORK, HIGH, MEDIUM and LOW, which can be selected by assigning an APPNCOS value from the current COSTAB.

The standard APPNCOS entries available on any APPN platform are shown in Table 3-1.

Table 3-1 Standard APPNCOS entries and their priorities

APPNCOS	Priority	Usage
CPSVCMG	NETWORK	CP-CP sessions
SNASVCMG	NETWORK	DLUR and CNOS
#INTER #INTERSC	HIGH	interactive
#CONNECT	MEDIUM	
#BATCH #BATCHSC	LOW	Printers, File Transfer

Tip: If possible, use the available standard APPNCOS values.

3.4.2 Inventory of logmode tables

In most installations, historically a great number of logmode tables exist and are referred to using the MODETAB parameter on many APPL, LU, or CDRSC definitions. The same logmode entries can probably be found in multiple tables, sometimes with different session parameters. As we have seen above, APPN prioritization requires to add a desired APPNCOS keyword to the logmode entry. This poses a problem when many mode tables need to be updated.

Which mode tables are actually being used? Which ones can be remove, because they are no longer used? The only way to determine this is to do a search on ‘`MODETAB=`’ across all concatenated VTAMLSTs and issue a D NET,TABLE,ID=tablename,SCOPE=ALL. The results will show you what logmode tables are in use (see Example 3-40 on page 115).

Example 3-40   Who uses the logmode tables?

```
D NET,TABLE,ID=ISTINCLM,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST986I TABLE=ISTINCLM TYPE=MODETAB USE COUNT=2
IST987I THE RESOURCES THAT USE THE TABLE ARE:
IST988I TCP*     SC30B*
IST1454I 2 RESOURCE(S) DISPLAYED
D NET,TABLE,ID=MTAPPC,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST986I TABLE=MTAPPC TYPE=MODETAB USE COUNT=6
IST987I THE RESOURCES THAT USE THE TABLE ARE:
IST988I SC30APPC SC30SRV  SC30SGT
IST988I IOASERV  SC30OSA  SC3CSAR?
IST1454I 6 RESOURCE(S) DISPLAYED
IST314I END
D NET,TABLE,ID=AMODETAB,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST986I TABLE=AMODETAB TYPE=MODETAB USE COUNT=141
IST987I THE RESOURCES THAT USE THE TABLE ARE:
IST988I SC30N    SC30NPPT SC30NLUC
IST988I SC30N000 SC30N001 SC30N002
IST988I SC30N003 SC30N004 SC30N005
D NET,TABLE,ID=EMSMODE,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST986I TABLE=EMSMODE TYPE=MODETAB USE COUNT=30
IST987I THE RESOURCES THAT USE THE TABLE ARE:
IST988I SC30E001 SC30E002 SC30E003
...
IST1454I 30 RESOURCE(S) DISPLAYED
IST314I END
D NET,TABLE,ID=NEWMTAB,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST986I TABLE=NEWMTAB TYPE=MODETAB USE COUNT=8
IST987I THE RESOURCES THAT USE THE TABLE ARE:
IST988I L8E1     L8E2     L8E3
...
```

The results of our investigation are shown in Table 3-2.

Table 3-2   Mode tables in use and APPNCOS settings

MODETAB	Major node	Application	Logmodes	COS/APPNCOS
ISTINCLM	@TCPLUS TCP	Telnet LUs	SNX3270x 1	#CONNECT for all user logmodes
AGWTAB	APDBDC	DB2®		
MTAPPC	APAPPCXX	APPC/MVS	APPCHOST	none
AMODETAB	APNETVXX	NetView®	DSIL6MOD	none
EMSMODE	APEMSXX	NetView Access	DYNAMIC	none
NEWMTAB		Console		

Our results: In total there were 112 logmode entries in five tables. Sixteen entries had a subarea class of service defined, that means 96 were using the default COS and were not assigned a designated priority in subarea. The logmodes SNASVCMG and #INTER were found in two tables, even CPSVCMG was defined in one user mode table. Neither COS nor APPNCOS were defined in any of those logmodes.

3.4.3 Steps to consolidate your user defined logmode tables

Remember, with EE you are sharing the common IP infrastructure and competing for bandwidth with other IP traffic. Chapter 1 explained how SNA priorities are treated in the IP network if EE is used to transport SNA data. Here is what needs to be done to simplify and manage session priorities in SNA:

1. Copy all logmodes from all tables into a single table ALLMODES
2. Remove duplicate logmodes
3. Remove logmodes that also exist in ISTINCLM
4. Assign an APPNCOS=#INTER to all interactive logmodes.
5. Assign an APPNCOS=#BATCH to all other logmodes
6. Change all MODETAB= definitions to MODETAB=ALLMODES
7. Dynamically activate the change using V NET,ACT,ID=majnode,UPDATE=ALL
8. Code DYNMODTB=ALLMODES and issue F NET,VTAMOPTS,DYNMODTB=ALLMODES

Now we can be sure that all logmodes that we defined are using either #INTER or #BATCH as an APPNCOS. In APPN, MODE-to-COS resolution does not always occur at the SLU VTAM as it did with pure subarea.

Note: There is a problem for PLU initiated sessions in APPN environments. Given that the SLU is a dynamic cross domain resource, the PLU VTAM will not be able find the logmode, if it is not in ISTINCLM, and therefore will not be able to assign the desired class of service, even though the logmode is found in the SLU VTAM and defined there in the MODETAB operand of the secondary LU. To get around this problem, you could define the SLU as a CDRSC and use the MODETAB parameter to point to the desired table for a given logmode. This is however, is not viable and contradicts the principle of using dynamics in an APPN environment. Therefore, we recommend specifying a mode table for dynamic CDRSCs using the VTAM start option (DYNMODTB).

3.4.4 Changes to the default logmode table ISTINCLM

We generally recommend not to modify the default logmode table ISTINCLM. However, there are some good reasons to apply some minor changes to this table.

There is a potential problem that a session request specifies a logmode that is not in our single user defined mode table or that the SLU does not have a MODETAB coded at all. Normally those sessions will fail with a sense code (08210002). There is a special logmode called ISTCOSDF provided in ISTINCLM, which can be used to assign an APPNCOS for those sessions that are initiated with an unknown logmode. The APPNCOS of this LODMOGE is #CONNECT. We recommend changing this to APPNCOS=#BATCHSC in ISTINCLM, because this APPNCOS is not generally used very often. We can identify those sessions easily using D NET,RTPS,APPNCOS=#BATCHSC.

In addition we decided to create a new #CONNECT logmode in ISTINCLM that maps to an APPNCOS of #CONNECT. We can then use D NET,APING,ID=xxx,LOGMODE=#CONNECT to test connectivity across an #CONNECT HPR pipe later on.

Attention: The #CONNECT logmode does not exist on other platforms and must be defined before it can be used in the APING command. If another VTAM is the target of such an APING command, the #CONNECT logmode must be available in ISTINCLM. This is because there is not a VTAM APPL major node where you could code a MODETAB pointing to a user defined table.

Looking at the APPNCOS values assigned to the logmodes in ISTINCLM, we recognize that they do not follow our suggestion, putting interactive sessions on #INTER and printer or high volume traffic on #BATCH. The logmodes given in Table 3-3 were changed to correct the priorities to meet our objectives.

Table 3-3 Modified Logmodes in ISTINCLM

Logmode	Used by	Modified	Original	Comment
NSX3270x [1]	TN3270 sessions	#INTER	#CONNECT	TELNETDEVICE
SNX3270x [1]	TN3270E sessions	#INTER	#CONNECT	TELNETDEVICE
SCS	LU1 printer	#BATCH	#CONNECT	
D4C2K, DSC2K	LU3 printer	#BATCH	#CONNECT	
ISTCOSDF	logmode unknown	#BATCHSC	#CONNECT	See ISTCOSDF start option
#CONNECT	APING	#CONNECT	None	Newly created

[1] x represents the model type (1-5) based on the DEVICETYPE of the Telnet terminal.

The source of ISTINCLM can be found in SYS1.SAMPLIB.

F NET,TABLE,OPT=LOAD,NEWTAB=ISTINCLM must be issued to load the new table.

The APPNCOS start option can be used to assign a certain APPNCOS if a session request comes in with an unknown APPNCOS. The default is #CONNECT.

3.4.5 Summary: What we achieved with the consolidation task

The object of the logmode consolidation process, should be a single logmode table that contains all logmodes known to be used in the network. All sessions using any of those known logmodes should either use an APPNCOS of #INTER or #BATCH. If a session sets up without a logmode being specified or a logmode yet not known and defined, it will use an APPNCOS of #BATCHSC. So a regular check on HPR pipes with that APPNCOS will reveal the LUs involved in those sessions and measures can be taken to resolve this situation in the future. The APPNCOS of #CONNECT will be used if a session sets up with an unknown APPNCOS.

3.5 Modifying z/OS definitions

In this section we point out some general z/OS definitions that you must consider before you enable EE in your system.

3.5.1 Storage considerations

HPR is a protocol that was designed for high speed networks. The Adaptive Rate Based (ARB) flow control algorithm dynamically adjusts the allowed send rate based on the current capabilities of the network. With today’s network infrastructure, we can easily get into very high send rates on the HPR pipes. In fact, this is what we want to achieve with a migration to EE; use the high speed IP infrastructure for the SNA applications. The RTP function of HPR is responsible for detecting packets that get lost and retransmitting them. Therefore, it must hold on to the data buffer until it gets an acknowledgement from the receiving RTP. Already in steady state networks, there is always a certain amount of packets *in flight*, which means unacknowledged by the remote RTP. These unacknowledged buffers are kept in CSM storage and are freed if the acknowledgement arrives. If the acknowledgment does not arrive due to network problems, the storage cannot be freed. In fact, the number of unacknowledged buffers may increases dramatically. The CSM demand of a HPR enabled VTAM cannot be calculated with the golden rule of ‘steady state’ usage +x%. It might ramp up very fast, if a central network component causes many HPR pipes to stall at once.

ECSA definitions in IEASYS00

We recommend that you assign as much ECSA as you can on the RTP endpoints and monitor the usage constantly. Large installations may see a peak demand of 500 MB or more. So rather approach this limit from the top than from the bottom, VTAM does not use CSA below the 16 MB line (see Example 3-41).

Example 3-41 ECSA definitions in SYS1.PARMLIB

```
SYS1.PARMLIB(IEASYS00)
CSA=(2048,600M),
```

CSM definitions in IVTPRM00

The default CSM storage definitions are set to 120 MB. You should monitor the usage on a regular basis using the D NET,CSM,OWNERID=ALL and if necessary modify IVTPRM00 in SYS1.PARMLIB (see Example 3-42).

Example 3-42 CSM definitions in SYS1.PARMLIB

```
FIXED MAX(120M)
ECSA MAX(120M)
```

The F NET,CSM,ECSA=xxxM command can be used to dynamically change the definitions. Make sure you have enough ECSA available in your system.

3.5.2 System Authorization Facility (SAF) definitions

SAF is an interface defined by MVS that enables programs to use system authorization services in order to control access to resources, such as data sets and MVS commands. SAF either processes security authorization requests directly or works with RACF, or other security products, to process them.

If you plan to use host names in any VTAM definitions, VTAM must have an OMVS segment to be able to resolve host names to IP addresses. Like any other IP application making program calls to the TCP/IP stack, VTAM needs to ask the stack for DNS name resolution when you code a host name instead of an IP address. In order for VTAM to successfully submit a DNS query to the stack, the user ID under which the VTAM address space is running must have authority to do so. That authority is built and given by defining an OMVS segment for VTAM's user ID.

3.6 Verifying the high availability in the APPN environment

Now that we have created and activated the definitions necessary to provide our highly available APPN environment, as depicted in Figure 3-1 on page 88, let us verify those definitions.

3.6.1 Topology displays

Example 3-43 shows a display command that lists all NNs in the network.

Example 3-43 Display all NNs in the network

```
D NET,TOPO,LIST=NN
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TOPOLOGY
IST1295I CP NAME           NODETYPE ROUTERES CONGESTION  CP-CP WEIGHT
IST1296I RDBOOKEE.SC30M    NN       128      NONE        YES   *NA*
IST1296I RDBOOKEE.SC31M    NN       128      NONE        *NA*  *NA*
IST314I END
```

There are two NNs in the topology, SC30M and SC31M.

Example 3-44 shows a display of all TGs originating from a node (SC30M).

Example 3-44 Topology display showing all TGs connecting to SC30M

```
D NET,TOPO,ID=SC30M,LIST=ALL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TOPOLOGY 193
IST1295I CP NAME           NODETYPE ROUTERES CONGESTION  CP-CP WEIGHT
IST1296I RDBOOKEE.SC30M    NN       128      NONE        *NA*  *NA*
IST1579I                   ------------------------------------------
IST1297I                   ICN/MDH  CDSERVR  RSN         HPR
IST1298I                   NO       YES      2           RTP
IST1579I                   ------------------------------------------
IST1223I                   BN       NATIVE   TIME LEFT   LOCATE SIZE
IST1224I                   NO       YES      14          16K
IST1357I                                                 CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.SC31M    2        OPER     INTERM      YES   *NA*
IST1301I RDBOOKEE.SC31M    21       OPER     INTERM      YES   *NA*
IST1301I RDBOOKEE.SC32M    21       OPER     ENDPT       YES   *NA*
IST1301I RDBOOKEE.SC32M    2        OPER     ENDPT       YES   *NA*
```

SC30M has four operational TGs, two to each adjacent node.

3.6.2 Scenario 1 - MPC link terminating

In this scenario we simulate an outage of the MPC link between SC32M and SC31M.

Initial state

Using a Telnet LU on SC32M we logged on to SC31ECHO using an RTP pipe called #INTER via TG2 to SC31M. Example 3-45 shows the HPR pipe using the MPC link.

Example 3-45 Display all RTP pipes over the MPC link towards SC31M

```
D NET,RTPS,FIRSTTG=2,CPNAME=SC31M 1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = RTPS 844
IST1695I PU NAME       CP NAME      COSNAME SWITCH CONGEST STALL SESS
IST1960I CNR0005E RDBOOKEE.SC31M    #INTER    NO     NO     NO      2
IST1960I CNR00004 RDBOOKEE.SC31M    RSETUP    NO     NO     NO      0
IST2084I 2 OF 2 MATCHING RTP PIPES DISPLAYED
IST314I END

D NET,ID=CNR0005E,E 2
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR0005E, TYPE = PU_T2.1 847
IST1392I DISCNTIM = 00010 DEFINED AT PU FOR DISCONNECT
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = SC31M - CP NETID = RDBOOKEE - DYNAMIC LU = YES
IST1589I XNETALS = YES
IST2178I RPNCB ADDRESS 0E945018
IST1963I APPNCOS = #INTER - PRIORITY = HIGH
...
IST1855I NUMBER OF SESSIONS USING RTP = 2
IST1480I RTP END TO END ROUTE - RSCV PATH
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC31M     APPN         RTP
IST875I ALSNAME TOWARDS RTP = MPC2SC31
IST1738I ANR LABEL               TP           ER NUMBER
IST1739I 8001000A00000000        *NA*         *NA*
IST231I RTP MAJOR NODE = ISTRTPMN
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST355I LOGICAL UNITS:
IST080I SC31ECHO ACT/S----Y TN31L214 ACT/S----Y
IST314I END
```

Note the following explanation for Example 3-45:

- **1:** D NET,RTPS,FIRSTTG=2,CPNAME=SC31M can be used to show all HPR pipes that are using this TG. The display result shows two HPR pipes. The #INTER pipe carries two sessions, the RSETUP pipe is used for control flows between RTPs and never carries any session.
- 2: D NET,ID=CNR0005E,E shows the path via TG2 and two cross domain LUs having sessions over this HPR pipe: SC31ECHO is an application on SC31M, TN31L214 is a telnet client on SC31M logged on to an application on this host SC30M.

Terminating the MPC links

Next, we inactivated the MPC connection between SC31M and SC32M. Example 3-46 shows the command issued at the remote host SC31M and the resulting messages on the local host SC32M.

Example 3-46 Inactivation of TG2 caused HPR path switch to occur

```
Command on SC31M
V NET,INACT,ID=MPC2SC32,I 1
IST097I VARY ACCEPTED
IST1196I APPN CONNECTION FOR RDBOOKEE.SC32M INACTIVE - TGN = 2
```

```
IST105I MPC2SC32 NODE NOW INACTIVE

Messages on SC32M
IST1196I  APPN CONNECTION FOR RDBOOKEE.SC31M    INACTIVE - TGN =   2 2
IST1494I  PATH SWITCH STARTED   FOR RTP CNR0005E TO RDBOOKEE.SC31M
IST1819I  PATH SWITCH REASON: TG INOP
IST1494I  PATH SWITCH COMPLETED FOR RTP CNR0005E TO RDBOOKEE.SC31M 3
IST1480I  RTP END TO END ROUTE - RSCV PATH
IST1460I  TGN  CPNAME             TG TYPE      HPR
IST1461I   21  RDBOOKEE.SC31M     APPN         RTP

D NET,SESSIONS,SID=DE1FD59B70100D61 4
IST350I DISPLAY TYPE = SESSIONS 865
IST879I PLU/DLU REAL = RDBOOKEE.SC31ECHO ALIAS = ***NA***
IST879I SLU/OLU REAL = RDBOOKEE.TN32L109 ALIAS = ***NA***
IST880I SETUP STATUS = ACTIV
IST875I ADJSSCP TOWARDS PLU = ISTAPNCP
IST875I ALSNAME TOWARDS PLU = CNR0005E
IST933I LOGMODE=SNX32705, COS=*BLANK*
IST1710I RSCV FROM PLU SAVED AT SESSION ACTIVATION
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC32M     APPN         RTP
IST1713I RTP RSCV IN THE DIRECTION OF THE PLU
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I  21  RDBOOKEE.SC31M     APPN         RTP
IST314I END
```

Note the following explanation for Example 3-46:

- 1:The MPC link is inactivated at the remote host.
- 2: Message IST1196I indicates that the AAPN TG2 is now inactive.
- 3: Messages IST1494I indicate a path switch occurred and succeeded.
- 4 The display of the session shows that the path had changed. IST1710I shows the original path at session initiation, IST1713I shows the current path, which is now using the XCF connection TG21.

Both sessions survived the link outage, because HPR path switched to an alternate link.

3.6.3 Scenario 2 - MPC link recovering: F NET,VTAMOPTS,PSRTETRY

An HPR pipe had been switched away from its preferred route. This scenario shows how the HPR pipe switches back again once the MPC link is active.

Initial state

Example 3-47 shows the activation of the MPC link at the remote host.

Example 3-47 Activation of the MPC link on SC31M

```
On SC31M
V NET,ACT,ID=MPC2SC32
IST097I VARY ACCEPTED
IST1086I APPN CONNECTION FOR RDBOOKEE.SC32M IS ACTIVE - TGN = 2
IST093I MPC2SC32 ACTIVE
```

Activating the MPC link did not cause any path switch to occur. This is because the default for PSRETRY in ATCSTR00 does not check regularly for better routes. We temporarily activate this function to switch all HPR pipes back to their preferred route.

PSRETRY causing HPR pipe to switch back

Example 3-48, “F NET,VTAMOPTS,PSRETRY=(30,30,30,30): triggering path switch” on page 122 shows how the HPR pipe switches back to its original route.

Example 3-48 F NET,VTAMOPTS,PSRETRY=(30,30,30,30): triggering path switch

```
On SC31M
F NET,VTAMOPTS,PSRETRY=(30,30,30,30) 1
IST097I MODIFY ACCEPTED
IST223I MODIFY COMMAND COMPLETED

ON SC32M
IST1494I  PATH SWITCH STARTED   FOR RTP CNR0005E TO RDBOOKEE.SC31M 2
IST1937I  PATH SWITCH REASON: INITIATED BY REMOTE PARTNER
IST1494I  PATH SWITCH COMPLETED FOR RTP CNR0005E TO RDBOOKEE.SC31M
IST1480I  RTP END TO END ROUTE - RSCV PATH
IST1460I  TGN  CPNAME            TG TYPE      HPR
IST1461I    2  RDBOOKEE.SC31M     APPN         RTP

D NET,SESSIONS,SID=DE1FD59B70100D61 3
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SESSIONS 879
IST879I PLU/DLU REAL = RDBOOKEE.SC31ECHO ALIAS = ***NA***
IST879I SLU/OLU REAL = RDBOOKEE.TN32L109 ALIAS = ***NA***
IST880I SETUP STATUS = ACTIV
IST875I ADJSSCP TOWARDS PLU = ISTAPNCP
IST875I ALSNAME TOWARDS PLU = CNR0005E
...
IST1710I RSCV FROM PLU SAVED AT SESSION ACTIVATION
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC32M     APPN         RTP
IST1713I RTP RSCV IN THE DIRECTION OF THE PLU
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC31M     APPN         RTP
IST314I END

ON SC31M 4
F NET,VTAMOPTS,PSRETRY=(0,0,0,0)
IST097I MODIFY ACCEPTED
IST223I MODIFY COMMAND COMPLETED
```

Note the following explanation for Example 3-48, “F NET,VTAMOPTS,PSRETRY=(30,30,30,30): triggering path switch” on page 122:

- 1: At the remote host SC31M the F NET,VTAMOPTS,PSRETRY command is issued to immediately check for better routes and path switch all HPR pipes that are currently on non-optimal routes.
- 2: On the local host IST1494I indicates a path switch occurred and succeeded. Note the reason as seen on this side of the HPR pipe is `INITIATED BY REMOTE PARTNER`. The new path is using TG 2 again.
- 3: The display of the session shows the actual route information using TG2 also.

- 4: On SC31M the PSRETRY function is disabled again.

3.6.4 Scenario 3 - Primary NNs terminating and coming back

This scenario shows how the EN SC32M looses its primary NNS SC30M, activates CP-CP sessions to its backup NNs and switches back to its primary when it comes back.

Initial state

Example 3-49 lists the initial state, SC32M has CP-CP sessions with primary NNS SC30M.

Example 3-49 D NET,CPCP showing CP-CP sessions with primary NNS

```
On SC32M
D NET,CPCP
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = CP-CP SESSION STATUS 882
IST1765I ADJACENT CP        WINNER   LOSER   STATE      NODE ANDCB
IST1766I RDBOOKEE.SC30M     ACT      ACT     UP         NN   13786010 1
IST1766I RDBOOKEE.SC31M     INACT    INACT   BOTH DOWN  NN   13786128
IST1454I 2 ADJCP(S) DISPLAYED
IST314I END

D NET,RTPS,APPNCOS=CPSVCMG,FIRSTTG=21 2
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = RTPS 998
IST1695I PU NAME       CP NAME      COSNAME SWITCH CONGEST STALL SESS
IST1960I CNR00063 RDBOOKEE.SC30M    CPSVCMG   NO     NO     NO      2
IST2084I 1 OF 1 MATCHING RTP PIPES DISPLAYED
IST314I END
```

Note the following explanation for Example 3-49:

- 1: IST1766I shows that CP-CP sessions with SC30M are active.
- 2: The command D NET,RTPS,APPNCOS=CPSVCMG,FIRSTTG=21 proves that TG21 is used for CPSCVMG pipe.

The primary NN server is terminated

Example 3-50 shows the messages on the EN SC32M that occurred when the primary NNS went away.

Example 3-50 Message on SC32M when SC30M terminates

```
IST1196I  APPN CONNECTION FOR RDBOOKEE.SC30M    INACTIVE - TGN =  21 1
IST2187I  XCF SEND FAILURE ON TRLE ISTT3230   MESSAGE TYPE: DATA
IST1684I  RETURN CODE = 00000008 REASON CODE = 00000008
IST1494I  PATH SWITCH STARTED   FOR RTP CNR00063 TO RDBOOKEE.SC30M  2
IST1819I  PATH SWITCH REASON: TG INOP
IST1494I  PATH SWITCH FAILED    FOR RTP CNR00063 TO RDBOOKEE.SC30M
IST1495I  NO ALTERNATE ROUTE AVAILABLE
IST2187I  XCF SEND FAILURE ON TRLE ISTT3230   MESSAGE TYPE: DISC
IST1684I  RETURN CODE = 00000008 REASON CODE = 00000008
IST1097I  CP-CP SESSION WITH RDBOOKEE.SC30M    TERMINATED 3
IST1280I  SESSION TYPE = CONWINNER - SENSE = 08420001
IST1488I  INACTIVATION OF RTP CNR00063 AS ACTIVE  TO RDBOOKEE.SC30M
IST871I  RESOURCE CNR00063 DELETED
```

```
IST1097I  CP-CP SESSION WITH RDBOOKEE.SC30M    TERMINATED 3
IST1280I  SESSION TYPE = CONLOSER  - SENSE = 08420001
IST1488I  ACTIVATION   OF RTP CNR00064 AS ACTIVE  TO RDBOOKEE.SC31M 4
IST1705I  SORDER   = APPN     FROM START OPTION
IST1705I  SSCPORD  = PRIORITY FROM START OPTION
IST894I  ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
IST895I     ISTAPNCP       08420001
IST1488I  ACTIVATION   OF RTP CNR00065 AS PASSIVE TO RDBOOKEE.SC31M 5

D NET,CPCP 6
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = CP-CP SESSION STATUS 036
IST1765I ADJACENT CP        WINNER   LOSER   STATE      NODE ANDCB
IST1766I RDBOOKEE.SC31M     ACT      ACT     UP         NN   13786128
IST1454I 1 ADJCP(S) DISPLAYED
IST314I END

D NET,RTPS,APPNCOS=CPSVCMG 7
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = RTPS 078
IST1695I PU NAME       CP NAME      COSNAME SWITCH CONGEST STALL SESS
IST1960I CNR00065 RDBOOKEE.SC31M    CPSVCMG   NO     NO     NO      2
IST2084I 1 OF 1 MATCHING RTP PIPES DISPLAYED
IST314I END
```

Note the following explanation for Example 3-50:

- 1: XCF link that carrying the CPSVCMG pipe terminates.
- 2: Path switch of CPSVCMG pipe is starting but fails with ‘NO ALTERNATE ROUTE AVAILABLE’.
- 3: Both CP-CP sessions terminate with sense code 08420001.
- 4: The local host SC32M activates a new RSETUP HPR pipe towards SC31M.
- 5: The remote host SC31M activates a new CPSVCMG pipe towards SC32M.
- 6: CP-CP sessions are active to SC31M.
- 7: Both CP-CP sessions are using CNR00065.

Primary NNS comes back

When the primary NNS activates again, the EN SC32M terminates the CP-CP sessions with the backup NNS and starts new CP-CP sessions with the primary NNS. Example 3-51 shows the associated messages on the EN SC32M.

Example 3-51   Messages on SC32M when primary NNS comes back

```
On SC32M

IST1097I  CP-CP SESSION WITH RDBOOKEE.SC31M    TERMINATED 1
IST1280I  SESSION TYPE = CONWINNER - SENSE = 08B50000
IST1097I  CP-CP SESSION WITH RDBOOKEE.SC31M    TERMINATED 1
IST1280I  SESSION TYPE = CONLOSER  - SENSE = 08B50000
IST1488I  ACTIVATION   OF RTP CNR00068 AS ACTIVE  TO RDBOOKEE.SC30M 2
IST1096I  CP-CP SESSIONS WITH RDBOOKEE.SC30M    ACTIVATED 3
IST1673I  SWITCH TO PREFERRED NETWORK NODE SERVER IS COMPLETE
```

```
D NET,CPCP 4
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = CP-CP SESSION STATUS 093
IST1765I ADJACENT CP        WINNER   LOSER   STATE      NODE ANDCB
IST1766I RDBOOKEE.SC30M     ACT      ACT     UP         NN   13786010
IST1766I RDBOOKEE.SC31M     INACT    INACT   BOTH DOWN  NN   13786128
IST1454I 2 ADJCP(S) DISPLAYED
IST314I END

D NET,RTPS,APPNCOS=CPSVCMG 5
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = RTPS 096
IST1695I PU NAME       CP NAME      COSNAME SWITCH CONGEST STALL SESS
IST1960I CNR00068 RDBOOKEE.SC30M    CPSVCMG   NO     NO     NO      2
IST2084I 1 OF 1 MATCHING RTP PIPES DISPLAYED
IST314I END
```

Note the following explanation for Example 3-51:

- 1: Both CP-CP sessions with SC31M are terminating with sense code 08B50000.
- 2: SC32M activates a new CPSVCMG pipe towards SC30M.
- 3: CP-CP sessions are activated.
- 4: A display shows them both active.
- 5: CNR00068 is the HPR pipe holding both sessions.

SC32M (EN) successfully switched to preferred NNS.

4

Enabling EE in CS for z/OS

This chapter discusses the required steps to enable Virtual Telecommunications Access Method (VTAM) and TCP/IP to communicate with other Advanced Peer-to Peer Networking (APPN) nodes using Enterprise Extender (EE). It builds upon the environment that was created in Chapter 3, “Preparing z/OS for EE” on page 87.

This chapter contains the following topics:

- “Description of our EE environment”
- “Overview of EE in Communications Server for z/OS”
- “Enabling EE in the z/OS TCP/IP environment”
- “Enabling EE in the z/OS VTAM environment”

4.1 Description of our EE environment

In addition to the basic APPN transmission groups (TG 21 and TG 2) created in Chapter 3, we added EE definitions to the existing APPN environment to establish EE transmission groups (TG 6) between each system. The definitions in this chapter, results in the diagram in Figure 4-1, which includes the following transmission groups (TGs):

- Connections through XCF using TG 21
- Connections through MPC channels using TG 2
- Connections through EE using TG 6

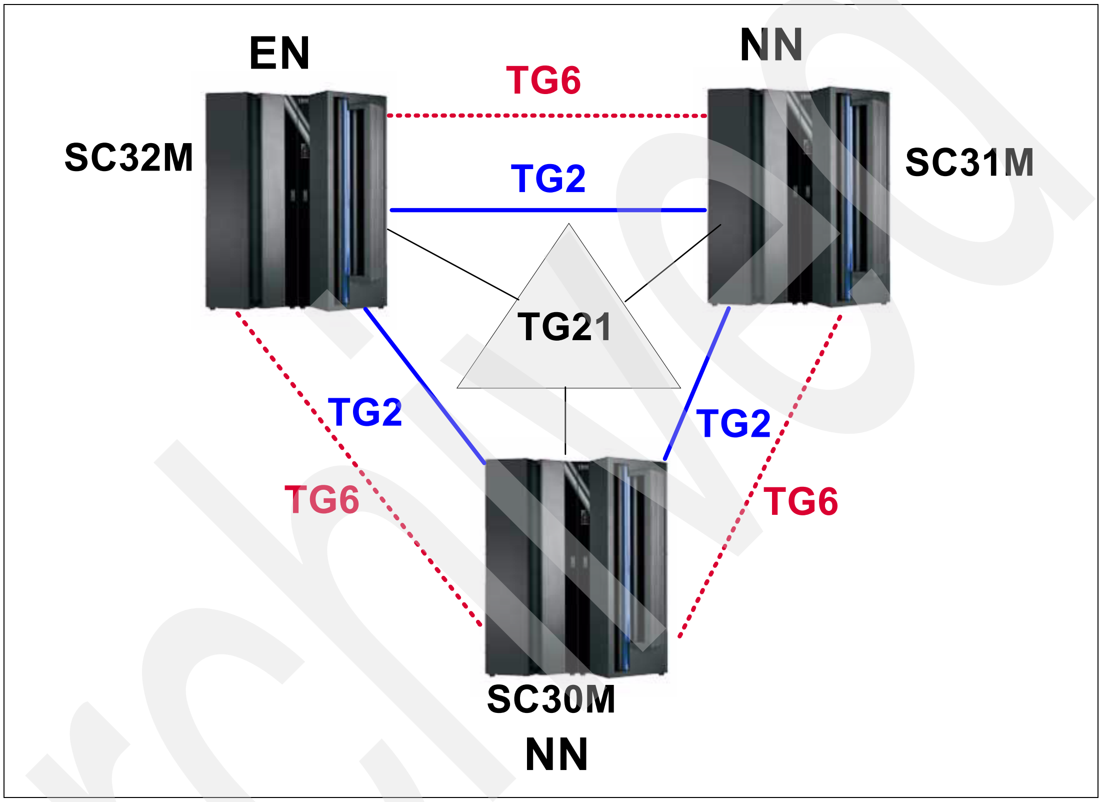


Figure 4-1 Our EE topology

A discussion of the TCP/IP definitions used to provide IP connectivity for our z/OS environment can be found in 3.2, “Preparing TCP/IP to participate in an APPN/EE network” on page 88.

4.2 Overview of EE in Communications Server for z/OS

Enterprise Extender requires VTAM to open five User Datagram Protocol (UDP) sockets on a single TCP/IP stack. In a multiple stack environment, you need to decide on which TCP/IP stack VTAM will open the EE sockets. The selected stack must allow VTAM to access the ports 12000 - 12004 for UDP protocol.

From a Data Link Control layer point of view, VTAM’s HPRIP XCA major node uses the SAMEHOST device (IUTSAMEH) to communicate with the TCP/IP stack. This special device is also used for communication between multiple TCP/IP stacks in the same logical partitions (LPAR) and must be started and available.

VTAM will issue specific BIND() calls to STATIC VIPA addresses only, no real interface addresses, no dynamic Virtual IP Addresses (VIPAs), and no distributed VIPAs. Those VIPA addresses must be defined in the TCP/IP stack's HOME statement.

Figure 4-2 illustrates the TCP/IP-to-VTAM cross-reference mappings of definitions required for the EE environment.

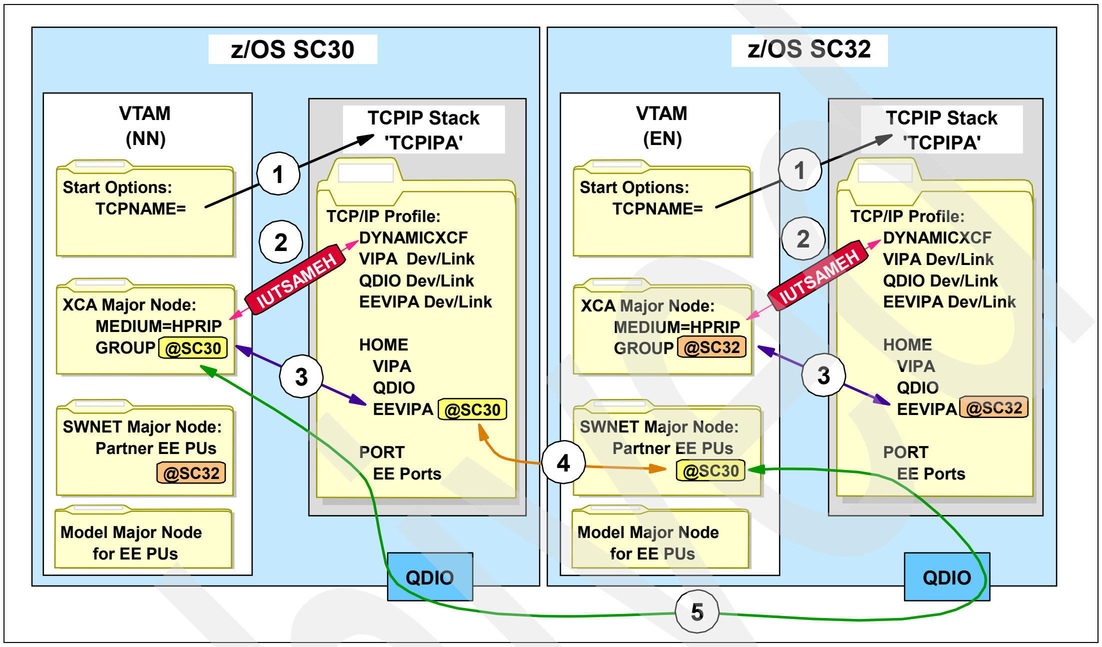


Figure 4-2 TCP/IP-to-VTAM cross-reference map for EE definitions

Note the following explanation for Figure 4-2:

1. VTAM must have affinity to only one TCP/IP stack. Assign the TCP/IP task name to VTAM by using the TCPNAME start option in VTAM's start list.
2. The IUTSAMEH device is the mechanism that VTAM and TCP/IP use to communicate with one another. Create the IUTSAMEH device by using the DYNAMICXCF statement in the TCP/IP profile.
3. The XCA major node is the mechanism used by VTAM to define VTAM's access to TCP/IP. The IPADDR or HOSTNAME parameter within the XCA major node points to the static VIPA address that has been defined by TCP/IP. The EE ports (12000 - 12004) are also defined in the XCA major node.
4. Any remote network partner wanting to establish an EE connection to the local node specifies the VIPA addresses of the local node in its SWNET major node. The remote partner uses its SWNET major node to access the local node through its TCP/IP stack.
5. Two EE partners communicate with each other through normal IP interfaces. OSA QDIO interfaces are used in this illustration.

The examples in this section are all based on our lab system SC30 and its TCPIPA stack. The definitions and commands for the other z/OS systems would be similar, except where names and addresses must be unique.

If you are not familiar with the APPN/HPR/EE environment, refer to the APPN and EE discussions in Chapter 1, "Introduction to APPN and Enterprise Extender" on page 1.

4.3 Enabling EE in the z/OS TCP/IP environment

The following discussion assumes that the TCP/IP stack that has been positioned to support EE is already running. The stack should contain definitions for XCF, QDIO, a static VIPA, and OSPF dynamic routing. If your z/OS TCP/IP system has not been prepared with this support, see Chapter 3, “Preparing z/OS for EE” on page 87.

For complete details in designing high availability into your z/OS networking systems, refer to *Communications Server for z/OS TCP/IP Implementation, Volume 3 - High Availability, Scalability, and Performance*, SG24-7341 and SG24-7171.

In our scenarios, multiple static VIPAs are included: one represents the stack itself and two more are dedicated to EE. The IUTSAMEH device is used by EE for communication between VTAM and TCP/IP.

> **Note:** VTAM’s EE function requires a STATIC VIPA to be used to communicate with TCP/IP. Starting with z/OS V1R5, multiple VIPA addresses can be used with EE for supporting various networking environments. These multiple VIPAs must belong to the same IP stack. VTAM does not communicate with more than one IP stack at a time in a multiple stack environment. We assigned two static VIPAs to our EE to help explain the use of multiple EE VIPAs.

We discuss the following topics related to enabling the TCP/IP environment for EE:

- “Modifications to the TCP/IP PROFILE”
- “Modifications to the OSPF environment”
- “Starting the TCP/IP task”
- “Starting the OMPROUTE task”
- “Verifying the TCP/IP environment”
- “Verifying the OSPF environment”

4.3.1 Modifications to the TCP/IP PROFILE

The following items discuss the PROFILE considerations when enabling the TCP/IP stack for EE:

- “UDP checksum considerations with EE”
- “DEVICE and LINK statements for IUTSAMEH for VTAM-to-TCP/IP”
- “DEVICE and LINK statements for the static EE VIPAs”
- “HOME statements for the VIPAs”
- “PORT reservations for EE”
- “STARTing the devices”
- “MULTIPATH considerations with EE”
- “MTU size considerations with EE”
- “RESOLVER setup considerations with EE”
- “NETMONITOR usage with EE”

UDP checksum considerations with EE

Performing UDP checksum has traditionally been considered optional. However, EE relies on the validity of the UDP checksum to provide data integrity for EE packets. Specify UDPCHKSUM in the TCP/IP profile as follows:

```
UDPCONFIG UDPCHKSUM
```

Important: Not only should UDPCHKSUM be specified in the z/OS TCP/IP profile, but every EE endpoint must implement UDP checksum in order to achieve data integrity for HPR connections. For details, see the checksum discussions in *z/OS Communications Server: IP Configuration Guide*, SC31-8775.

DEVICE and LINK statements for IUTSAMEH for VTAM-to-TCP/IP

The IUTSAMEH device is necessary for IPv4 support of VTAM-to-TCP/IP communication. EE uses this device to access TCP/IP.

If the DYNAMICXCF statement is not present in the TCP/IP profile, now is the time to add it. It is required so that the IUTSAMEH device can be dynamically defined. If it is not added, the IUTSAMEH device must be defined manually.

Example 4-1 shows both the methods.

Example 4-1 IUTSAMEH device definition for EE

```
Dynamic definition: 1
   IPCONFIG DYNAMICXCF 10.10.20.100 255.255.255.0 1

Manual definition: 2
   DEVICE IUTSAMEH MPCPTP
   LINK EZASAMEMVS MPCPTP IUTSAMEH
   HOME 10.10.20.100 EZASAMEMVS
   START IUTSAMEH
```

Note the following explanation for Example 4-1:

- 1: IUTSAMEH definitions are dynamically defined when dynamic XCF support is enabled using IPCONFIG DYNAMICXCF. For IPv4, the device name is IUTSAMEH and the link name is EZASAMEMVS.
- 2: To manually configure IUTSAMEH definitions for IPv4, define and start IUTSAMEH using the DEVICE, LINK, HOME, and START statements.

To simplify EE definitions, consider defining IUTSAMEH connections using the IPCONFIG DYNAMICXCF statement.

DEVICE and LINK statements for the static EE VIPAs

Example 4-2 shows the statements we used to define our EE VIPAs.

Example 4-2 VIPA Device and Link statements

```
DEVICE EEVIPA1  VIRTUAL 0         1
LINK EELINK1 VIRTUAL 0 EEVIPA1
;
DEVICE EEVIPA2  VIRTUAL 0         1
LINK EELINK2 VIRTUAL 0 EEVIPA2
```

Note the following explanation for Example 4-2:

- 1: Define one or more static VIPAs for EE. Define the static VIPA addresses for EE by using the DEVICE and LINK statements. Other non-EE applications can use the stack's VIPA to access the stack as well.

HOME statements for the VIPAs

Example 4-3 shows the HOME statements for our interfaces.

Example 4-3 Our HOME statements

```
HOME
  10.10.1.230    VIPALINK    1
  10.10.2.232    OSA2080LNK
  10.10.3.233    OSA20A0LNK
  10.10.2.234    OSA20C0LNK
  10.10.3.235    OSA20E0LNK
  10.10.1.231    EELINK1     2
  10.10.1.232    EELINK2

  PRIMARYINTERFACE VIPALINK  3
```

Note the following explanation for Example 4-3:

- The EE static VIPAs are always used as the source IP address for *all* EE UDP datagrams, *regardless* of the SOURCEVIPA setting. If you have NOSOURCEVIPA coded or defaulted, non-EE traffic uses the outbound interface's address as the source IP address, but EE traffic still uses the EE VIPA as the source IP address.
- The order in which the VIPA link interfaces are positioned in the HOME list is very important when SOURCEVIPA is in effect. When assigning dedicated VIPAs to EE, place them at the *bottom* of the list after all physical interfaces (2). Being placed at the end of the list will prevent their use as a source VIPA for *non-EE* traffic.
- Use the PRIMARYINTERFACE statement (3) to specify which link is to be designated as the default local host for use by the GETHOSTID() function. The PRIMARYINTERFACE statement's link IP address is *not* used as the source VIPA address for any out-going datagrams, *unless* that *same* address is configured as the SOURCEVIPA address (1).

PORT reservations for EE

Example 4-4 shows the statements we used to define our PORT reservations for EE.

Example 4-4 Port reservations for EE

```
Use of PORTRANGE
   PORTRANGE 12000 5 UDP NET ; RESERVE UDP PORTS 12000-12004 FOR EE 1

Use of PORT
   PORT 12000 UDP NET ; RESERVE UDP PORTS 12000-12004 FOR EE       2
        12001 UDP NET
        12002 UDP NET
        12003 UDP NET
        12004 UDP NET
```

Note the following explanation for Example 4-4:

- Reserve the UDP ports 12000-12004 for EE using the PORTRANGE statement (1) or PORT statements (2) to prevent another application from using one of the EE ports. If you do not reserve the UDP ports, you might have a conflict later when another application attempts to use one of the ports.

Use the MVS job name associated with the VTAM started task to reserve UDP ports that are to be used for EE connections. The MVS job name for a given started task can be determined in the following order:

- 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.
- The job name specified on the JOB JCL statement within VTAM's JCL.
- The proclib member name itself.

STARTing the devices

We did not have to provide a START statement for the VIPAs. They are started automatically by the stack and do not require a START statement.

We did not have to provide a START statement for the IUTSAMEH device. Using the DYNAMICXCF statement causes IUTSAMEH to also start automatically.

MULTIPATH considerations with EE

The IPCONFIG MULTIPATH parameters in the TCP/IP profile enables 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, it does not matter which option you choose. In either case, IP uses a *per batch of packets* approach to send EE UDP packets. Because there are no connections in the UDP protocol, true *per-connection* multipath is not possible with UDP and EE. *Per-packet* multipath routing is too granular, leading to resequencing overhead at the rapid transport protocol (RTP) receiving endpoint.

Note: If you already have multipath routing enabled for your TCP applications, you do not need to change it.

MTU size considerations with EE

To ensure optimal performance, the TCP/IP maximum transmission unit (MTU) size should be greater than the RTP network layer packet (NLP) size. VTAM queries TCP/IP for its MTU size when establishing an RTP connection and subtracts 31 bytes to allow for IP, UDP and LDLC headers. If the MTU size is less than 768 bytes, VTAM sets the maximum NLP size to 768 (this is the smallest maximum packet size allowed by VTAM for High Performance Routing (HPR) packets). This action by VTAM could cause TCP/IP to fragment if the real MTU size in the network is really less than 768 bytes. However, today's networks support an MTU size of at least 1492. If you see an NLP size of 768, it is probably because of a default MTU size of 576 bytes being chosen for the route to the destination IP address and indicative of a missing route definition.

Attention: IPSec adds additional ESP and AH headers to each IP packet, therefore enlarging the size of the original IP datagram in the WAN. If the resulting IPSec datagram exceeds the MTU size of the next hop, it will have to be fragmented. Some devices will not route fragmented packets for security reasons causing retransmissions at RTP layer. Unfortunately the retransmitted NLPs will suffer the same death and the HPR pipe will be stalled. If EE traffic is traversing IPSec tunnels, we recommend reducing the MTU size toward the destination host to 1420 to accommodate the increase of the packet size caused by IPSec. See the IPSec discussions in *z/OS Communications Server: IP Configuration Guide*, SC31-8775.

The PATHMTUDISCOVERY option in z/OS applies to TCP only and not to UDP. Other non-z/OS platforms can specify the option for either TCP or UDP or both. For details, refer to *z/OS Communications Server: IP Configuration Reference*, SC31-8776.

RESOLVER setup considerations with EE

The system RESOLVER is a separate address space used by the TCP/IP stack, servers, and clients, to resolve host name specifications to their associated IP addresses, and to dynamically locate a number of IP related service files. The main use of the RESOLVER is to resolve host names to their IP addresses. EE's access to the TCP/IP stack is defined in a VTAM XCA major node, and access to a partner node is defined in a VTAM SWNET major node. In both cases, XCA and SWNET, the definitions can be made with either IP addresses or host names.

When host names are used, VTAM must access the system RESOLVER to resolve the host names, local and remote, into their associated IP addresses. There could be a possibility that the official DNS servers within the organization do not have all name entries that EE might need to use. If the domain of the host name in question is owned by the organization and served by its DNS servers, then the host name can easily be added to the DSN servers.

> **Note:** There may be security setup considerations for both the RESOLVER and for VTAM when using the RESOLVER. For details on establishing access to RESOLVER services, see *z/OS Communications Server: IP Configuration Guide*, SC31-8775.

The use of host names is mandatory if you plan to use IP connection networks in conjunction with Network Address Translation (NAT) devices or with other distributed EE platforms with multiple IP interfaces. When the normal DNS server farm cannot accommodate the required host name, use of IPNODES is the only solution, however all z/OS LPARs must access the same IPNODES file or have the same contents in their IPNODES file.

The recommendation is to create a global TCPIP.DATA data set for a single stack environment, and a default TCPIP.DATA data set if a multiple stack environment is required. For details about customizing the RESOLVER and providing a local IPNODES file for resolving EE host names, refer to the following publications:

- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Communications Server: IP Configuration Reference*, SC31-8776
- *Communications Server for z/OS TCP/IP Implementation, Volume 1 - Base Functions, Connectivity, and Routing*, SG24-7339

NETMONITOR usage with EE

Use the NETMONITOR statement in the stack profile to activate or deactivate selected network management application programming interfaces (NMI). Different service functions can be specified on the NETMONITOR statement. One of the options is of particular interest in an EE environment. The PKTTRCSERVICE option enables the *packet trace service function* to run on this TCP/IP stack. The service enables network management applications to access trace data collected for any active packet traces or data traces.

An example of turning on the packet trace service function is shown here:

```
NETMONITOR PKTTRCSERVICE
```

For details about NETMONITOR options, refer to:

- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Communications Server: IP Configuration Reference*, SC31-8776

4.3.2 Modifications to the OSPF environment

Some installations may not be running OSPF. However, many of the latest features of TCP/IP that support dynamic load balancing and multi-instance applications require the OSPF dynamic routing protocol support. This discussion assumes OMPROUTE is running with OSPF support for XCF, OSAs, and a stack VIPA.

Example 4-5 shows static VIPA interface OMPROUTE statements we used.

Example 4-5 OSPF_INTERFACE statements for VIPAs

```
; Static vipa#1 for EE
ospf_interface ip_address=10.10.1.231      1
           subnet_mask=255.255.255.0
           name=EELINK1                    2
           Advertise_VIPA_Routes=HOST_ONLY 3
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; Static vipa#2 for EE
ospf_interface ip_address=10.10.1.232      1
           subnet_mask=255.255.255.0
           name=EELINK2                    2
           Advertise_VIPA_Routes=HOST_ONLY 3
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
```

Note the following explanation for Example 4-5:

- 1: The IP address of the VIPA must match the address in the HOME list.
- 2: The name must match the link name in the HOME list.
- 3: Only the full 32-bit host IP address of a VIPA should be advertised and *not* the subnet address of the VIPA.

4.3.3 Starting the TCP/IP task

The TCP/IP task must be restarted to pick up the additional VIPA definitions. We stopped and restarted our TCPIPA procedure using the MVS commands: P TCPIPA and S TCPIPA.

4.3.4 Starting the OMPROUTE task

The OMPROUTE task must be restarted to pick up the additional VIPA definitions. Our OMPA task was automatically shut down and restarted by the TCPIPA task. If you do not use the AUTOLOG method of starting OMPROUTE, use the MVS Start and Stop commands: P OMPA and S OMPA to recycle the task.

4.3.5 Verifying the TCP/IP environment

For a complete list of commands that can be used to manage the TCP/IP environment and their detailed syntax, refer to *z/OS Communications Server: IP System Administrator's Commands*, SC31-8781 and *z/OS Communications Server: IP User's Guide and Commands*, SC31-8780.

We used the following commands and tools to verify our TCP/IP environment:

- "Expected TCP/IP stack startup messages"
- "TSO PING command"

- “TSO TRACERTE command”
- “TSO NETSTAT command”
- “z/OS DISPLAY TCPIP NETSTAT command”

Expected TCP/IP stack startup messages

Successful TCP/IP startup can be verified by looking for the following groups of messages:

- “Startup messages for the IUTSAMEH device”
- “Startup messages for the TCP/IP stack”

Startup messages for the IUTSAMEH device

Example 4-6 shows the expected successful startup messages for IUTSAMEH.

Example 4-6 IUTSAMEH startup message

```
EZZ4313I INITIALIZATION COMPLETE FOR DEVICE IUTSAMEH                    1
EZZ4324I CONNECTION TO 10.10.1.231 ACTIVE FOR DEVICE IUTSAMEH           2
EZZ4324I CONNECTION TO 10.10.1.232 ACTIVE FOR DEVICE IUTSAMEH
```

Note the following explanation for Example 4-6:

- 1: The EZZ4313I and EZZ4324I messages may not appear immediately upon TCP/IP startup if the EE XCA major node is not active. A NETSTAT DEVLINKS display will show a status of *SENT SETUP* for the IUTSAMEH device until the EE XCA major node is activated on the VTAM side. When the EE XCA major node becomes active, the IUTSAMEH status changes to *READY*, and messages EZZ4313I and EZZ4324I are then issued.
- 2: Message EZZ4324I indicates which static VIPA has been selected by the EE XCA major node. It is a way to verify the existence and health of the static VIPAs intended for EE use. The IP addresses in the message will be those defined in the PROFILE for the EE VIPAs.

Startup messages for the TCP/IP stack

Example 4-7 shows the expected successful startup messages for the TCPIPA stack.

Example 4-7 TCPIPA stack startup messages

```
EZB6473I TCP/IP STACK FUNCTIONS INITIALIZATION COMPLETE.
EZAIN11I ALL TCPIP SERVICES FOR PROC TCPIPA ARE AVAILABLE.
```

TSO PING command

Example 4-8 shows how we used the PING command.

Example 4-8 TSO PING to verify the IP environment

```
PING 10.10.1.231                        1
   CS V1R8: Pinging host 10.10.1.231
   Ping #1 response took 0.000 seconds.
   ***

PING 10.10.1.232                        2
   CS V1R8: Pinging host 10.10.1.232
   Ping #1 response took 0.000 seconds.
   ***
```

TSO TRACERTE command

We did a TRACERTE to all the addresses, Example 4-9 shows how we used the TRACERTE command and the results.

Example 4-9 TSO TRACERTE to verify the IP environment

```
TRACERTE 10.10.1.232                                 1
   CS V1R8: Traceroute to 10.10.1.231 (10.10.1.231):
   1 10.10.1.231 (10.10.1.231)  0 ms  0 ms  0 ms
   ***
```

TSO NETSTAT command

Example 4-10 shows how we used the NETSTAT command and the results.

Example 4-10 TSO NETSTAT to verify the IP environment

```
NETSTAT HOME                                                           1
   MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPIPA          05:07:22
   Home address list:
   . . . .
   LinkName:   EELINK1                                                 2
     Address:  10.10.1.231
       Flags:
   LinkName:   EELINK2
     Address:  10.10.1.232
       Flags:
   LinkName:   EZASAMEMVS                                              3
     Address:  10.10.20.100
       Flags:
   . . . .
NETSTAT DEVLINKS      4
NETSTAT CONFIG        5
NETSTAT CONN         6
NETSTAT ROUTE        7
NETSTAT PORTLIST     8
```

Note the following explanation for Example 4-10:

- 1: The NETSTAT HOME command generated output showing the following interfaces:
  - 2: The static VIPAs for EE
  - 3: The EZASAMEMVS link (IUTSAMEH device) generated by DYNAMICXCF statement

We issued the other commands listed, but did not show the output:

- 4: DEVLINKS shows all the devices and their status
- 5: CONFIG shows miscellaneous stack settings
- 6: CONN shows all connections (clients and servers)
- 7: ROUTE shows all routes (defined and learned), including the OSPF default routes
- 8: PORTLIST shows port reservations for EE and others if reserved

z/OS DISPLAY TCPIP NETSTAT command

Example 4-11 shows how we used the z/OS DISPLAY TCPIP command.

Example 4-11   z/OS DISPLAY TCPIP NETSTAT to verify the IP environment

```
D TCPIP,TCPIPA,N,HOME
D TCPIP,TCPIPA,N,ROUTE
D TCPIP,TCPIPA,N,STATS
D TCPIP,TCPIPA,N,STATS,PROTOCOL=UDP       1
   EZD0101I NETSTAT CS V1R8 TCPIPA 538
   UDP STATISTICS
     DATAGRAMS RECEIVED    = 4074
     NO PORT ERRORS        = 624
     RECEIVE ERRORS        = 1
     DATAGRAMS SENT        = 4699
   END OF THE REPORT
```

The same NETSTAT commands can be entered as z/OS systems commands. One helpful command is NETSTAT STATS. It returns much information about the TCP/IP environment. The PROTOCOL=UDP filter can be specified to limit the output to UDP information. This could be very helpful when monitoring EE activity.

4.3.6 Verifying the OSPF environment

For a complete list of commands that can be used to manage the OSPF environment and their detailed syntax, refer to *z/OS Communications Server: IP System Administrator's Commands*, SC31-8781 and *z/OS Communications Server: IP User's Guide and Commands*, SC31-8780.

We used the following commands and tools to verify our OSPF environment:

- "Expected OMPROUTE startup messages"
- "TSO NETSTAT ROUTE command"
- "z/OS DISPLAY TCPIP,OMP,RTTABLE command"
- "Possible OMPROUTE error conditions"

Expected OMPROUTE startup messages

Example 4-12 shows the expected successful OMPROUTE startup messages.

Example 4-12   OMPROUTE startup messages

```
EZZ7800I OMPROUTE STARTING
EZZ7898I OMPROUTE INITIALIZATION COMPLETE
```

TSO NETSTAT ROUTE command

Example 4-13 shows how we used the TSO NETSTAT ROUTE command to verify that the EE VIPAs were represented.

Example 4-13   TSO NETSTAT ROUTE to verify the OSPF environment

```
NETSTAT ROUTE                                                            1
   MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPIPA         21:43:29
   IPv4 Destinations
   Destination         Gateway          Flags     Refcnt  Interface
   -----------         -------          -----     ------  ---------
   . . .
   10.10.1.222/32      10.10.3.223      UGHO      000000  OSA20A0LNK
   10.10.1.230/32      0.0.0.0          UH        000000  VIPALINK
   10.10.1.231/32      0.0.0.0          UH        000000  EELINK1        2
   10.10.1.232/32      0.0.0.0         UH        000000  EELINK2
   10.10.1.240/32      10.10.2.244     UGHO      000000  OSA2080LNK
........
```

Note the following explanation for Example 4-13 on page 138:

- 1: The TSO NETSTAT ROUTE command shows route information.
- 2: Verify that OSPF has informed the stack of the routes to the EE VIPAs.

z/OS DISPLAY TCPIP,OMP,RTTABLE command

Example 4-14 shows how we used the z/OS DISPLAY TCPIP,OMPR,RTTABLE command.

Example 4-14 z/OS DISPLAY TCPIP OMP to verify the OSPF environment

```
D TCPIP,TCPIPA,OMP,RTTABLE                                                 1
   EZZ7847I ROUTING TABLE 592
   TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

   SPIA   0.0.0.0                 0  11      646     10.10.2.1         (4)
    . . .
    DIR*  10.10.1.230      FFFFFFFF  1       9901    VIPALINK
    DIR*  10.10.1.231      FFFFFFFF  1       9901    EELINK1                2
    DIR*  10.10.1.232      FFFFFFFF  1       9901    EELINK2
    SPF   10.10.1.240      FFFFFFFF  20      646     10.10.2.244       (8)
    . . .
```

Note the following explanation for Example 4-14:

- 1: The display of the OSPF routing table, RTTABLE, shows the supported OSPF routes.
- 2: Verify that the EE VIPAs appear in the table.

Possible OMPROUTE error conditions

An active EE connection could unexpectedly fail with the messages:

- IST1411I INOP GENERATED FOR linename
- IST1430I REASON FOR INOP IS XID OR LDLC COMMAND TIMEOUT
- IST314I END

The EE connection inactivation is due to an LDLC time out. 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 the network
- There are possible OMPROUTE problems

4.4 Enabling EE in the z/OS VTAM environment

In VTAM the most important step of enablement of EE is defining the interface to TCP/IP through an XCA major node with MEDIUM=HPRIP. If multiple TCP/IP stacks are active, the stack affinity must be established through a new start option TCPNAME. Once this is done, VTAM is ready to accept connections through the EE DLC type. All EE connections are treated as switched PUs and can be either predefined in a SWNET major node or can be dynamically defined. A model PU type, DYNTYPE=EE, is available to define characteristics for dynamically defined PUs that connect through the EE DLC. Another model PU type, DYNTYPE=VN, is used to define dynamically created connections through a Connection Network or VRN.

We discuss the following topics:

- "Adding VTAM start options"
- "Defining the XCA HPRIP major node"
- "Defining model major nodes for EE connections and RTP Pipes"
- "Defining switched PUs for EE connections"

4.4.1 Adding VTAM start options

There are five start options specific to EE:

- "IPADDR" defines the default VIPA to be used unless otherwise specified
- "HOSTNAME" defines the default IP host name to be used unless otherwise specified
- "TCPNAME" defines the TCP/IP stack to be used for EE
- "T1BUF and T2BUF" are new buffer pools associated with EE connections

Other VTAM start options are HPR relevant and apply to all HPR pipes, not uniquely to EE, but should also be reviewed with respect to their appropriate settings in an IP-based infrastructure. These include the following:

- "HPRARB" defines the ARB mode that VTAM is preferring when a pipe sets up.
- "HPRPST" are the timers to wait until an HPR pipe will fail after a path switch does not succeed immediately.

IPADDR

IPADDR defines the IPV4 static VIPA address used by VTAM if it is not specified in a GROUP. Starting in Communications Server V1R5 VTAM can make use of multiple VIPAs. We recommend defining the IP addresses at the GROUP level, so that they can be visible in one place.

HOSTNAME

HOSTNAME can be used instead of IPADDR. Using system symbolics in combination with HOSTNAME enables you to share the same start list among multiple LPARs. However, VTAM requires that resolving host name to IP address already is available at VTAM startup if HOSTNAME is coded in ATCSTRxx.

TCPNAME

TCPNAME specifies the started task name of the TCP/IP stack VTAM will use to connect to when activating the XCA major node. Coding TCPNAME is required, if you want to synchronize the activation of the TCP/IP stack and the XCA major node.

T1BUF and T2BUF

T1BUF and T2BUF are new buffer pools for EE. These buffer pools take over *some* of the function previously provided by the TIBUF pools. Monitor usage of these three pools and set to minimize buffer pool expansion.

HPRARB

HPRARB is not new with Enterprise Extender, however, we strongly recommend to keep the default (HPRARB=RESPMODE), to ensure VTAM get its fair share of the available bandwidth in the IP network.

HPRPST

HPRPST determines how long VTAM will wait for a path switch of an RTP pipe to succeed. When this timer expires, VTAM will terminate its side of the HPR pipe. There is a specific timer for each priority enabling it to terminate high priority pipes sooner than low priority pipes. Problems may arise when the two RTP endpoints have different timers, especially if dependent LUs are using an HPR pipe. One side may start a new session because its pipe and sessions were already terminated, while the other side still sees them active waiting for a path switch to occur. We therefore recommend using the same timers on all RTP endpoints. If you keep the defaults, chances are high that all other RTP endpoints will operate on the same timer values. See Table 4-1 for the default values.

Table 4-1 Path switch timer default values

	Network	**High**	**Medium**	**Low**
VTAM	1 minute	2 minutes	4 minutes	8 minutes
CS/AIX	60 seconds	120 seconds	240 seconds	480 seconds
CS/Linux	60 seconds	120 seconds	240 seconds	480 seconds
CS/Windows	60 seconds	120 seconds	240 seconds	480 seconds
PCOMM	60 seconds	120 seconds	240 seconds	480 seconds
i5/OS	60 seconds	120 seconds	240 seconds	480 seconds
SNASw	60 seconds	120 seconds	240 seconds	480 seconds

Tip: Use HPRPST default values to ensure time-out values are consistent within your network and increase the probability they match the values in any partner networks with which you have connections.

PSRETRY and PSWEIGHT

PSRETRY can be used to regularly check for a better route and automatically switch to an alternate path. PSWEIGHT can be used to define whether a path switch should also occur to an equally weighted route. We recommend keeping the default disabling this function and activate it on demand, after a preferred link has recovered.

Summary of start options

Example 4-15 is a summary of the new or modified start options we used in our scenarios.

Example 4-15 Start options to enable EE

```
TCPNAME=TCPIPA,                                                       X
T1BUF=(2000,,0,,60,120),                                              X
T2BUF=(2000,,0,,60,120),                                              X
XHPRPST=(8M,4M,2M,1M),                                                X
HPRARB=RESPMODE,                                                      X
```

This completes the necessary start options and we can now proceed to the major node definitions in VTAM.

4.4.2 Defining the XCA HPRIP major node

The XCA major node defines the communication path to the TCP/IP stack. It consists of a single PORT statement with MEDIUM=HPRIP and one or more GROUPs defining the lines and PUs to be used for EE connections.

The following definitions are discussed in this section:

- "PORT" defining the global EE parameters to apply to all connections
- "GROUP" definitions defining IP addresses, Connection networks and other parameters

> **Tip:** Starting with Communications Server V1R8 the V NET,ACT,ID=xcamnode,UPDATE=ALL can be used to non-disruptively add new groups to an active XCA major node.

PORT

There can only be one XCA PORT MEDIUM=HPRIP active in VTAM. It defines the general parameters that EE operates on. Two EE connected hosts should use the same parameters, because it could have an adverse effect if the remote EE partners chose different parameters. We recommend not changing the defaults as those parameters can only be globally set, not on a per connection basis. If you change them, all your potential remote EE partners have to change theirs as well. While this may be still possible in your own company, you may not have that flexibility when connecting to a business partner.

IPPORT defines the UDP ports to be used. RFC2353 requires UDP ports to be set to 12000-12004. Do not change them, otherwise you can lose inter-operability with other EE implementations.

IPTOS defines the type of service bits that VTAM will set in the IP header of outbound IP packets. The defaults are IPTOS=(20, 40, 80, and C0) for low, medium, high, and network and LDLC priority. Those are the architected values and should not be changed. Be aware that those bits might be changed for each packet on its way to a destination, even within z/OS with Traffic Regulation Manager and Policy Agent.

IPRESOLV is a timer to wait for a successful host name resolution by DNS. The default is 0 meaning VTAM will wait indefinitely. We recommend coding a reasonable non-zero value to avoid hanging sessions across connection networks.

> **Note:** Sessions through Connection Networks across a NATed infrastructure require the use of host names. Therefore a successful host name lookup is mandatory to resolve the remote host name into a valid IP address in the local IP domain. This must be handled by the local DNS infrastructure and should not take too long. For non-NATed connection networks, VTAM can perfectly use the IP address presented in the APPN LOCATE flows and does not require host name resolution at all. Still host name resolution will occur as the use of host names has precedence over the use of IP address. To avoid unnecessary delays for those session setups, the IPRESOLV timer should be reasonably small.

HPREELIV controls the HPR Liveness Reduction for EE. The default is YES. When an RTP pipe is comprised of a single EE connection (which includes a single physical hop across a two-hop EE virtual routing node [VRN]), the HPR ALIVE timer will not be used, thereby relying on the EE LDLC layer for inactivity monitoring and saving the timer maintenance and keep-alive status request/reply overhead.

LIVTIME, **SRQTIME**, and **SRQRETRY** control, how fast VTAM will recognize the loss of connectivity on a link to a remote EE node and enter path switch state for all HPR pipes currently using this link. All EE nodes will constantly send LDLC TEST frames on idle links at regular intervals (liveness timer). If the remote side does not answer within an interval called the LDLC retry timer, the LDLC TEST frame will be will resent again until the LDLC retry count is exceeded. If the remote side still did not respond, a link will be considered dead and an INOP will finally cause a path switch to occur for all pipes eligible for path switching.

The formula to calculate the total time is LDLC_total=LIVTIME+((SRQRETRY+1)*SRQTIME). Ideally all EE nodes should inop their representation of a link at approximately the same time so that they do not get out of sync in terms of connection states, where one side regards a link as broken, while the other end still is in retry processing, keeping all HPR pipes and sessions using those pipes up.

LIVTIME specifies how often TEST frames are sent out to a remote EE partner on port 12000. VTAM can use an incremental liveness timer to reduce the number of test frames sent into the network on otherwise idle connections. You can specify an initial value and a maximum value. Especially in large EE networks, the amount of UDP port 12000 traffic can cause a high overhead in the network and increase CPU consumption in both VTAM and TCP/IP because these packets flow 7*24h. If you specify a maximum value, VTAM will increase the intervals on idle connections and therefore reduce the number of TEST frames sent into the network. The backside is a longer time to recognize an outage on the connection. But, as mentioned before, this is on idle connections only, when HPR traffic resumes over an EE connection, the current liveness window will reset to the initial setting.

Attention: LIVTIME=(min,max) can be used to control the interval between TEST frames sent on UDP port 12000. High values reduce the number of packets but add additional delay until a broken connection is detected on LDLC level.

All platforms provide their own LIVTIME values. A lower value will cause the TEST frames to be sent more frequently. Every node must react upon incoming TEST requests with a TEST reply.

Table 4-2 LDLC liveness timer, SRQ timer and SRQ retries and their defaults

Platform	Liveness timer(s)	SRQ timer (s)	SRQ retries	Total time (s)
VTAM	10	15	3	70
CS Windows	10	15	3	70
CS AIX	2	2	10	24
CS Linux	10	15	3	70
System i	10	15	3	70

Table 4-2 lists the default LDLC parameters and the resulting total time it takes to detect a loss of connectivity. After the TG INOP message, which triggers a path switch, HPRPST comes into play to determine how long an HPR pipe will stay in the path switch state until it is terminated.

Note: SRQTIME and SRQRETRY do not have any influence on the SRQ timers on HPR pipes. Those are maintained by the ARB algorithm and cannot be configured.

GROUP

The GROUP statement defines a set of lines and PUs that are to be used for EE connections. You can have multiple groups in the same XCA major node.

IPADDR and **HOSTNAME** defines the IP address that VTAM will use to open bind to. Those two parameters are mutually exclusive within the same group. If you are using HOSTNAME you need to have a functioning resolver process when VTAM activates the major node and the name resolution must return a static VIPA address of the local TCP/IP stack.

VNNAME, VNTYPE, and **TGP** are used to define a connection to a Virtual Routing Node (VRN). There are two types of Connection Networks (local and global), the TGP parameter can be specified, to assign TG characteristics to dynamically created links across a connection network.

UNRCHTIM can be used to control how long a connection network link should be marked unreachable after a dial failure.

KEEPACT should be set to YES, to recover the lines and PUs after the TCP/IP stack went away.

DYNPU controls, whether links can automatically be created or whether they have to be predefined.

Note: The DYNPU parameter of the XCA GROUP is ignored for connections coming in over the connection network. Dynamic PU definition is always permitted for these connections.

AUTOGEN defines the number of lines and PUs that will be created in this group.

4.4.3 Defining model major nodes for EE connections and RTP Pipes

A model major node can be used to provide characteristics of dynamically created PUs. There are 3 model PUs that can be used to influence the behavior of links, RTP pipes and LU6.2 sessions in an EE environment.

DYNTYPE=EE will be used for dynamic PUs that connect through a DYNPU=YES XCA group, when VTAM does not find a matching SWNET PU definition for the device. We recommend coding this model to assign realistic characteristics via a TGP and code DISCNT=NO to avoid link deactivation of unused links.

DYNTYPE=VN is used for dynamically created PUs that represent a link through a connection network or Virtual Node (VN). We recommend coding a delay on the DISCNT parameter to avoid links terminating too soon, in case they need to be re-established shortly after the termination. The TG characteristics of those links are defined at the GROUP in the XCA major node.

DYNTYPE=RTP is used to define characteristics for RTP PUs. This major node may be necessary to force LU6.2 sessions to stay up, even without a conversation. With the default DISCNT value, you will see sessions terminate because of the Limited Resource bit (LR) being set in the BIND. Some applications try to re-acquire lost sessions with the result that sessions will be activated and deactivated 7*24h causing increased overhead and CPU utilization. We recommend coding DISCNT=NO in this case. The down side is, that there will be RTP LULU pipes with no sessions.

4.4.4 Defining switched PUs for EE connections

If you do not allow dynamically created PUs or want to actively start a connection towards another node, you must predefine the PU in a SWNET major node. Optionally you can specify a path statement, that contains information about the destination address of the remote node.

PU

The PU statement is used to assign characteristics to the link.

NETID and **CPNAME** or **IDBLK** and **IDNUM** is a combination that uniquely identifies the remote node within this VTAM. We recommend using NETID and CPNAME as this enables you to also assign a TGN, which, if used consistently, can simplify network management considerably.

CPCP and **HPR** should be set to YES as this is required to participate in an APPN/HPR network.

DISCNT determines whether a link should be deactivated if it is not used by any session. We recommend coding DISCNT=NO for those links as the other end will most probably reactivate it immediately.

DWACT and **DWINOP** define whether a PU should be dialed when it is activated or after an inop occurred. We recommend deciding on one side only starting and recovering the link to avoid XID race conditions. On VTAM NNs, both parameters should be set to NO as the ENs will have a vital interest to keep their link to a NN server active and will activate and recover the links if they drop. Between VTAM NNs, you should also decide on an active or passive role.

TGP can be used to assign characteristics from a transmission group profile.

PATH

The PATH statement must be coded if VTAM wants to dial a PU. It contains information about the destination of the remote node.

HOSTNAME and **IPADDR** are mutually exclusive. Using host names requires name resolution to resolve the host name into an IP address. We recommend using IPADDR whenever the IP address is considered to be static and host names if IP addresses change frequently, for example when DHCP is used to assign IP addresses. HOSTNAME must be used if the EE session path includes a NAT device.

SAPADDR can be used to address a specific Service Access Point. The default SAPADDR is 8. We recommend changing this to 4 to be consistent with other EE platforms and to avoid the creation of parallel TGs between the same IP address pair. This is no longer supported in z/OS V1R8.

GRPNM specifies the XCA group to be used for dial out. The IP address belonging to this group will be chosen as the source IP address in outbound packets.

4.4.5 Summary of VTAM major nodes

The XCA major node defines the connection to the TCP/IP stack.

Example 4-16 shows the XCA major node used on all LPARs.

Example 4-16 XCA major node

```
EEXCA&SYSCLONE. VBUILD TYPE=XCA 1
EEPORT&SYSCLONE. PORT MEDIUM=HPRIP,                                              *
                 IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                         *
                 IPRESOLV=2,                                                     *
                 SRQTIME=15,SRQRETRY=3
*
```

```
EEGVL&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            * 2
               HOSTNAME=SC&SYSCLONE.M-EE1.ITSO.IBM.COM,                *
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,TGP=TGPVRN,       *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              *
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM&SYSCLONE.,EEN&SYSCLONE.)
*
EEGVG&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            * 2
               HOSTNAME=SC&SYSCLONE.M-EE2.ITSO.IBM.COM,                *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,TGP=EEXTWAN,          *
               UNRCHTIM=180,                                           *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               *
               AUTOGEN=(16,EEX&SYSCLONE.,EEY&SYSCLONE.)
```

Note the following explanation for Example 4-16:

- 1: In the XCA major node defining the EE DLC there is only one PORT statement defining the global EE parameters.
- 2: Multiple GROUP statements can be used to define different VIPA addresses, possibly using different connectivity options. We used two groups to define a link to two Connection Networks. The use of IP host names in combination with system symbolics enables us to reuse the same XCA major node on all LPARs.

There are three model PU definitions that should be considered in an EE environment.

Example 4-17 lists the three major nodes that we used.

Example 4-17 Model major nodes

```
EEMODLPU VBUILD TYPE=MODEL
EEPUMODL PU    DYNTYPE=EE,DISCNT=NO,CPCP=YES,TGP=EEXTCAMP 1

EEMODLVN VBUILD TYPE=MODEL
EEPUVRN  PU    DYNTYPE=VN,DISCNT=(DELAY,,180) 2

EEMODLRT VBUILD TYPE=MODEL
EEPURTP  PU    DYNTYPE=RTP,DISCNT=NO,CPCP=YES 3
```

Note the following explanation for Example 4-17:

- 1: This is the model for dynamic switched PUs connecting via an XCA GROUP with DYNPU=YES.
- 2 : This model is used for dynamic PUs that represent links through a connection network.
- 2 : This model can be used to modify the characteristics of RTP pipes.

The switched major node is used to predefine a connection. It is required when VTAM needs to start a link. Typically this is the case on VTAM ENs defining the link to a NNS or at EBNs defining a link to another non-native EBN.

Example 4-18 shows a switched major node defining a connection towards a NN. This switched major node contains two PU and PATH definitions towards the two NNSs.

Example 4-18 Switched major node on EN SC32M

```
EESWEN32 VBUILD TYPE=SWNET
*******************************************************************
*  Link to primary NNS SC30M
*******************************************************************
EEPUNN30 PU    CPNAME=SC30M,NETID=RDBOOKEE,TGN=06,                     * 1
               TGP=HIPERSOC,DISCNT=NO,                                 *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,                         *
               DWACT=YES,DWINOP=YES,                                   * 3
               ISTATUS=ACTIVE
EEPTNN30 PATH  HOSTNAME=SC30M-EE2.ITSO.IBM.COM,SAPADDR=4,              * 2
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
*******************************************************************
*  Link to backup  NNS SC31M
*******************************************************************
EEPUNN31 PU    CPNAME=SC31M,NETID=RDBOOKEE,TGN=06,                     *
               TGP=HIPERSOC,DISCNT=NO,                                 *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,                         *
               DWACT=YES,DWINOP=YES,                                   *
               ISTATUS=ACTIVE
EEPTNN31 PATH  HOSTNAME=SC31M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

Note the following explanation for Example 4-18:

- 1: The remote node is identified by NETID and CPNAME. We decided to assign a specific TG number for EE connections, so that they can be easily identified in the APPN topology.
- 2: The PATH statement contains the remote host name, alternatively the IP address can be specified here. We recommend coding the SAPADDR=4 to be compliant with other EE platforms and avoid unwanted parallel TGs.
- 3: DWACT and DWINOP define whether the link should be dialed upon activating of the PU and after an INOP. We recommend only one side taking responsibility of activation and recovery of the link to avoid race conditions.

5

Enabling EE in Personal Communications (PCOMM)

In this chapter, we discuss Enterprise Extender (EE) support for IBM Personal Communications (PCOMM) and the configuration steps necessary to enable it. The chapter also provides techniques that can be used for problem determination with PCOMM in an EE environment.

The topics covered in this chapter are:

- "Overview of Personal Communications (PCOMM)"
- "Implementation considerations"
- "Configuring EE in Personal Communications (PCOMM)"
- "Verifying and managing EE in Personal Communications (PCOMM)"
- "Diagnosing EE in Personal Communications (PCOMM)"

5.1 Overview of Personal Communications (PCOMM)

PCOMM is a key component of IBM host access products, providing access to applications and data residing on enterprise servers (mainframes). PCOMM fully implements Advanced Peer to Peer Networking (APPN), High Performance Routing (HPR), and Dependent LU Requester (DLUR) functions. With PCOMM, you can use various connectivity methods such as LLC2, Multipath Channel (MPC), in addition to Enterprise Extender (EE).

Through the use of EE in Personal Communications, you can maintain the integrity of SNA LU6.2, CPIC, and APPC applications over IP. This capability is an IBM exclusive, both on laptops and in communications servers that provide the gateway back into the SNA/APPN environment, over an IP network.

This chapter does not cover the installation considerations and steps of PCOMM. For more information about installation of PCOMM, refer to *Personal Communications for Windows Quick Beginnings*, GC31-8679.

Throughout this chapter we assume that you have a basic knowledge of APPN. If you are not familiar with APPN, refer to 1.2, “What is APPN and APPN/HPR?” on page 4.

5.2 Implementation considerations

In this section we discuss topics related to the implementation of PCOMM in an EE environment, such as:

- “Highly available mainframe applications”
- “Firewall considerations”

5.2.1 Highly available mainframe applications

PCOMM is installed on a personal computer (PC) and typically, those systems do not have multiple network interfaces. This means that if the network interface in the PC fails, connectivity to the mainframe application is lost. Whereas, if the mainframe loses its connectivity, all clients lose connectivity to the applications and this likely to have a major impact on their business.

To ensure high availability for mainframe applications, you must have a backup scheme in place for those applications, such as a parallel sysplex environment. However, if the appropriate configuration to support high availability is not implemented correctly, then the backup scheme becomes ineffective.

With EE, you can use various APPN features to ensure high availability to mainframe applications. These are described in the following sections:

- “Using a backup DLUS”
- “Using multiple connections”

Using a backup DLUS

EE supports dependent LUs through DLUS-DLUR sessions. If you are not familiar with DLUS and DLUR, refer to “Dependent LU Requester and Server (DLUR/DLUS)” on page 33.

If your primary DLUS fails, then a backup DLUS can takeover DLUS functions. Though defining a backup DLUS is not mandatory, we strongly recommend it.

Using multiple connections

Usually an End Node (EN) has a preferred network node server (NNS) and a backup network node server. For a preferred NNS and a backup NNS, you must define a static connection for each node. For other nodes, you can define more connections to each node or define a connection network to make a dynamic APPN link.

We strongly recommend using a connection network, because it is easier to implement and manage.

5.2.2 Firewall considerations

Network security almost always includes firewalls and rules that support certain security policies. As mentioned, EE uses User Datagram Protocol (UDP) ports to communicate between endpoints, therefore, changes will have to made to the firewalls along the path of the endpoints.

In addition, many PCs have a personal firewall to block unauthorized network access. By default, EE uses the UDP ports 12000 through 12004. You must ensure that these EE UDP ports are not blocked by the personal firewall in the PC.

In many cases, users fail to connect to the mainframe with EE, because firewalls along the path of the endpoints are not permitting EE packets to pass through.

5.2.3 Scenario used in this chapter

Figure 5-1 on page 152 illustrates the scenario used in this chapter. Here, SC30M, SC31M and SC32M are z/OS VTAMs and EECPPCOM is a PCOMM.

We configure the PCOMM as an End Node (EN). The Control Point (CP) name will be EECPPCOM. The preferred network node server of this EN will be SC30M, and the backup network node server will be SC31M. We define static link stations for these two network nodes because a static link station must exist for network node servers.

For connectivity to other nodes such as SC32M, we use dynamic link stations through connection network instead of using static link stations. In our scenario, the name of the connection network is RDBOOKEE.VRNLOCAL.

Throughout this chapter, we use host names instead of IP addresses. Using host names instead of IP addresses helps you to manage the EE environment when the IP network needs changes.

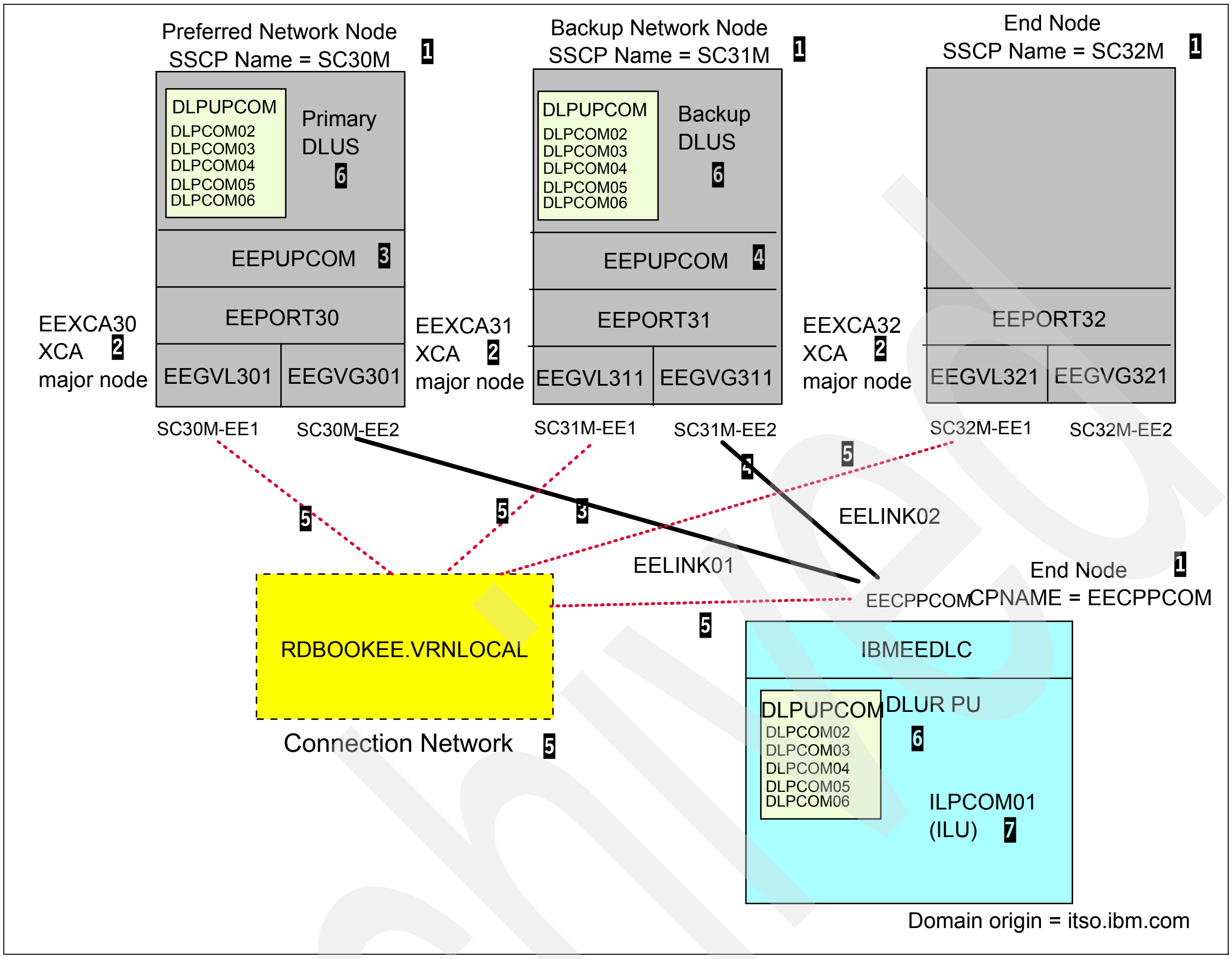


Figure 5-1 Scenario used in this chapter

Here are explanatory notes to Figure 5-1:

- 1: CP name of each node. For z/OS VTAM, it is an SSCP name. The preferred network node server for EECPPCOM is SC30M, and the backup network node server is SC31M.
- 2: For enabling EE in z/OS VTAM, an XCA major node is required. An XCA major node consists of port and group definitions. We define two groups in an XCA major node in each system.
- 3: Static link station to SC30M (preferred network node server). We use EEGVG3x1 group for this connectivity, PU definition for this link station should exist because DYNPU=NO for this group. We define EEPUPCOM for this purpose.
- 4: Static link station to SC31M (backup network node server). During an outage of the preferred network node server, this node will be used as a network node server. We use EEGVG3x1 group for this connectivity; a PU definition for this link station should exist because DYNPU=NO for this group. We define EEPUCOM for this purpose.
- 5: In each node, we define a connection network, RDBOOKEE.VRNLOCAL. Because every node participates in this same connection network, dynamic link stations will be generated without any definition, if needed. We use EEGVL3x1 group for this connection network in z/OS, and IBMEEDLC in PCOMM.

- 6: For dependent LUs, the DLUS/DLUR feature must be configured. In our scenario, SC30M will be a primary DLUS, and SC31M will be a backup DLUS. You must define a type 2.0 PU and all dependent LUs in each VTAM for accepting the DLUR's request. For PCOMM, the definition of DLUR PU and dependent LUs is required.
- 7: For independent LUs, there is nothing to be defined in z/OS if DYNLU=YES is used; otherwise a CDRSC definition is required for each independent LU in VTAM.

5.3 Configuring EE in Personal Communications (PCOMM)

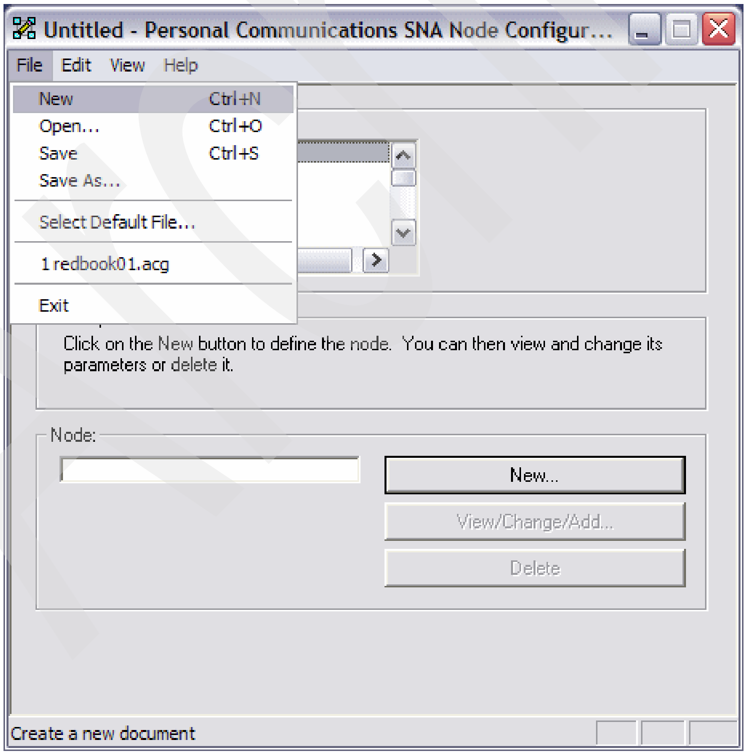
In this section we describe two methods to configure the PCOMM. First we describe a general method to configure PCOMM resources and then a quick method to define 3270 sessions using EE.

The section covers the following:

- "Configuring node"
- "Configuring connectivity"
- "Configuring dependent LUs with DLUR"
- "Configuring independent LUs"
- "Configuring LOGMODEs"
- "Configuring quick 3270 Sessions"

We use the SNA node configuration utility for configuring the PCOMM. Select **Start** → **All Programs** → **IBM Personal Communications** → **SNA Node Configuration**.

Create a new configuration file as shown in Figure 5-2.



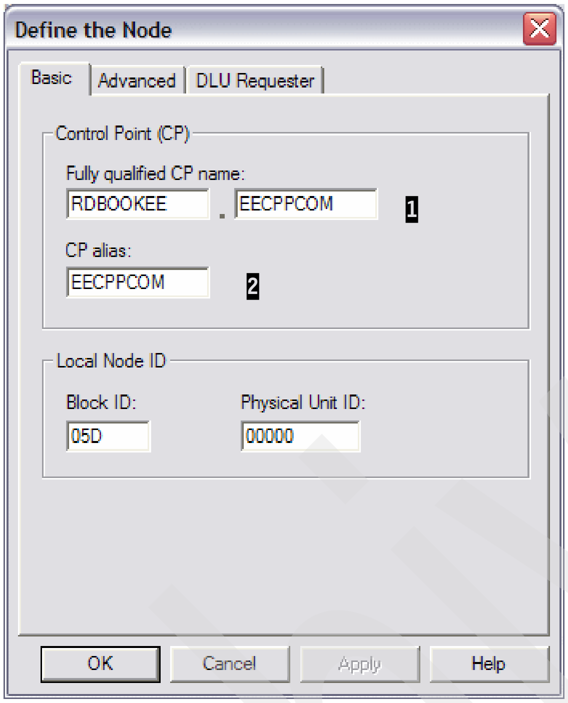
Click on the New button to define the node. You can then view and change its parameters or delete it.

Create a new document

Figure 5-2 Creating a new configuration file

5.3.1 Configuring node

A node is a base object in an APPN environment. A node in an APPN environment is similar to a host in a TCP/IP network. To define a node, select the **Configure node** menu in SNA node configuration utility. This opens Define the Node window as shown in Figure 5-3.



Control Point (CP)

Fully qualified CP name: RDBOOKEE . EECPPCOM

CP alias: EECPPCOM

Local Node ID

Block ID: 05D

Physical Unit ID: 00000

Figure 5-3 Configuring the node parameters

Here are explanatory notes to Figure 5-3:

- 1: This is the fully qualified control point (CP) name. It is divided into two parts. The first part is the NETID of an SNA network, and the second part is the CP name of that server. The CP name must be unique in one SNA network. This means that in a network with the same NETID, the CP name must be unique. Contact the z/OS administrator to get the correct NETID for your system. CP is a unique identity in an APPN environment, and represents this server in the same way as a host name is for TCP/IP. Therefore, a meaningful name should be given for maintenance purposes. In this Chapter, we use RDBOOKEE as the NETID.
- 2: This is just an alias of this control point.

Note: PCOMM does not have a network node or branch extender feature. PCOMM can join an SNA or APPN environment only as an End Node (EN).

5.3.2 Configuring connectivity

Next, we configure the connectivity resources for this PCOMM. Configuration steps will be in the following order:

1. “Configuring PCOMM for connectivity using EE”
2. “Configuring VTAM for connectivity using EE”

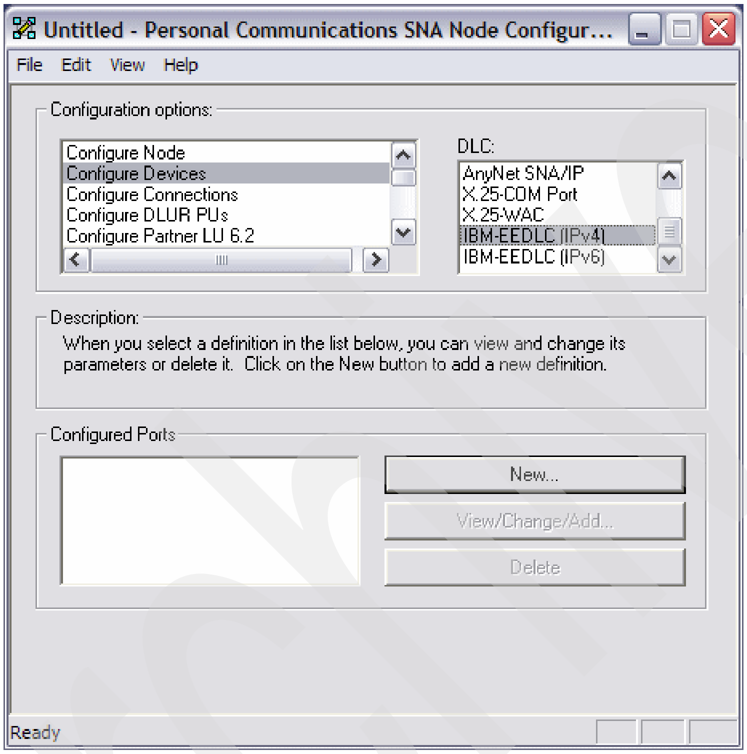
Configuring PCOMM for connectivity using EE

For EE, you should define:

- "Device"
- "Connections"
- "Configuring connection network"

Device

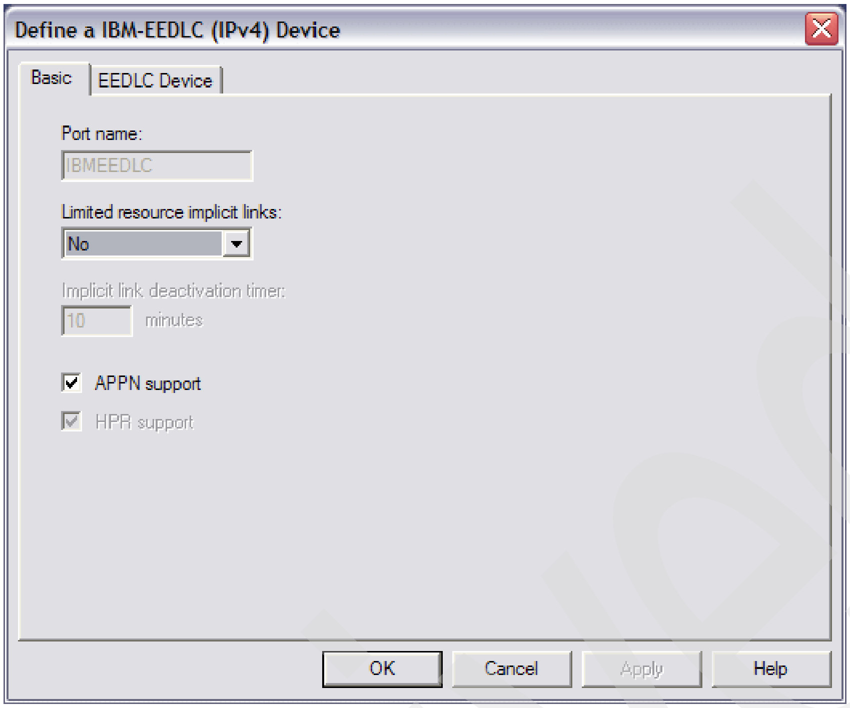
Select the **Configure Devices** menu in the SNA node configuration utility as shown in Figure 5-4.



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Figure 5-4 Configure devices menu

Figure 5-5 shows the selected EE devices.



Define a IBM-EEDLC (IPv4) Device

Port name: IBMEEDLC

Limited resource implicit links: No

Implicit link deactivation timer: 10 minutes

☑ APPN support

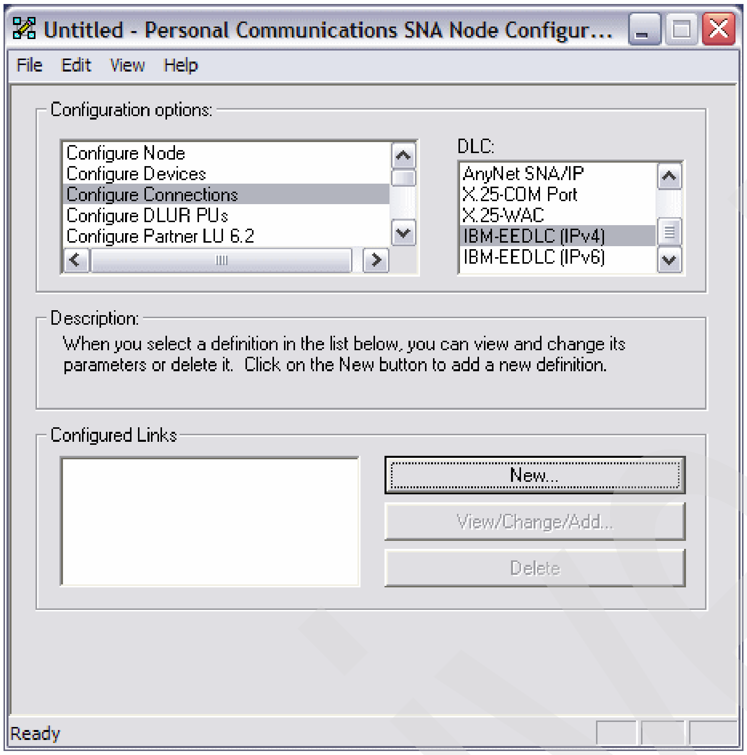
☑ HPR support

Figure 5-5 Configuring the EEDLC device

We use all the parameters as default.

Connections

A connection in a PCOMM is an APPN link to the other node. You can define a connection in the **Configure Connections menu** in the node configuration utility as in Figure 5-6. We make two link stations, one for SC30M (preferred network node server), and one for SC31M (backup network node server).



Configuration options:

- Configure Node
- Configure Devices
- Configure Connections
- Configure DLUR PUs
- Configure Partner LU 6.2

DLC:

- AnyNet SNA/IP
- X.25-COM Port
- X.25-WAC
- IBM-EEDLC (IPv4)
- IBM-EEDLC (IPv6)

Description:

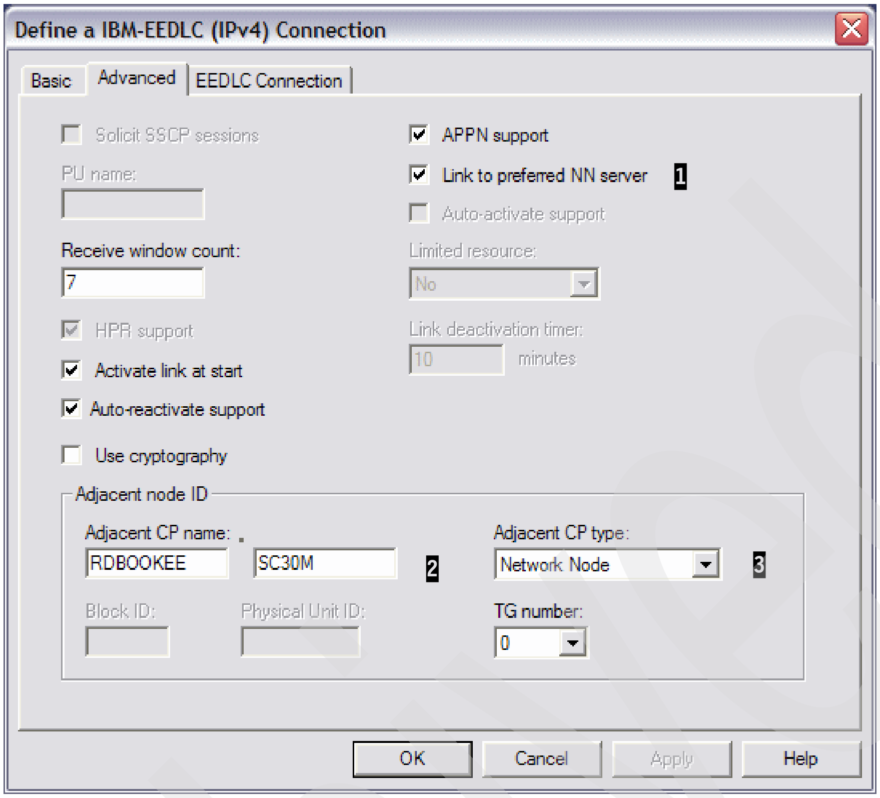
When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Configured Links

Figure 5-6 Configure connections menu

Click **New** to create a new connection.

We configure the Advanced tab of this new connection first. See Figure 5-7.



Define a IBM-EEDLC (IPv4) Connection

□ Solicit SSCP sessions

PU name:

Receive window count: 7

☑ HPR support

☑ Activate link at start

☑ Auto-reactivate support

□ Use cryptography

☑ APPN support

☑ Link to preferred NN server **1**

□ Auto-activate support

Limited resource: No

Link deactivation timer: 10 minutes

Adjacent node ID

Adjacent CP name: RDBOOKEE . SC30M **2**

Adjacent CP type: Network Node **3**

Block ID:

Physical Unit ID:

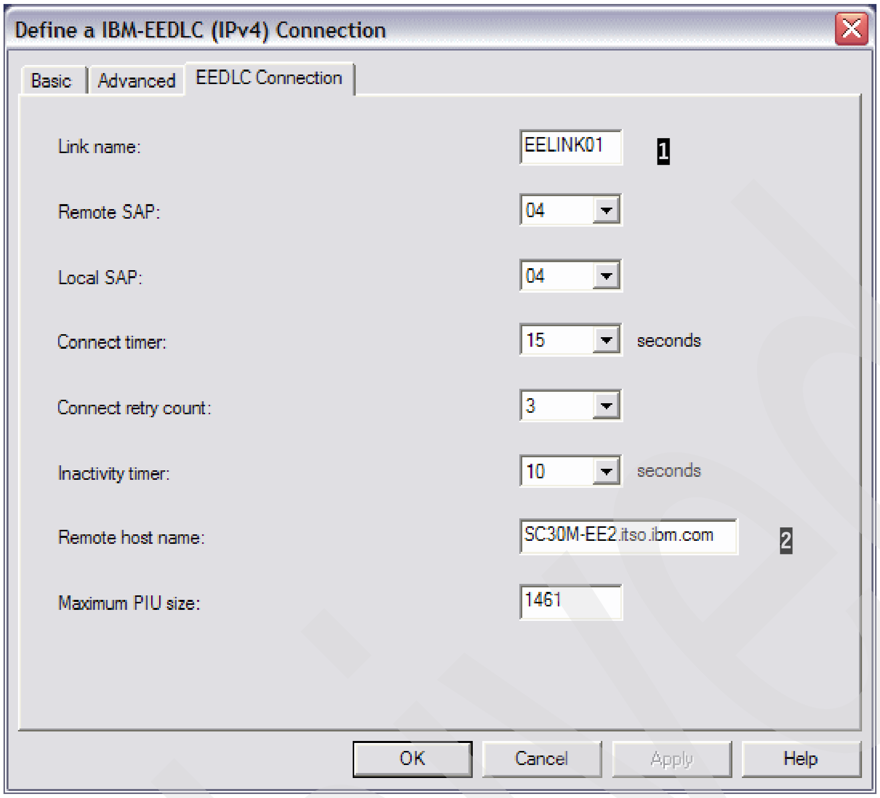
TG number: 0

Figure 5-7 Configuring Advanced tab of connection to SC30M

Here are explanatory notes to Figure 5-7:

- **1**: You can specify this remote node as the preferred network node server (NNS). The End node (EN) can make two CP-CP sessions with only one network node (NN) at one time. This NN is called the network node server of the EN. If you do not specify which node is the preferred NNS, then the first NN which makes CP-CP sessions with the EN is chosen as the NNS of that EN. In our scenario, SC30M is chosen to be the preferred network node server, and SC31M is the backup network node.
- **2**: The CP name of the remote node. Specify the fully qualified CP name (including NETID and CP name) here.
- **3**: The node type of the remote host. In our scenario, the remote node is a Network Node (NN).

Next, click the **EEDLC Connection** tab to proceed with the configuration. See Figure 5-8.



Link name: EELINK01

Remote SAP: 04

Local SAP: 04

Connect timer: 15 seconds

Connect retry count: 3

Inactivity timer: 10 seconds

Remote host name: SC30M-EE2.itso.ibm.com

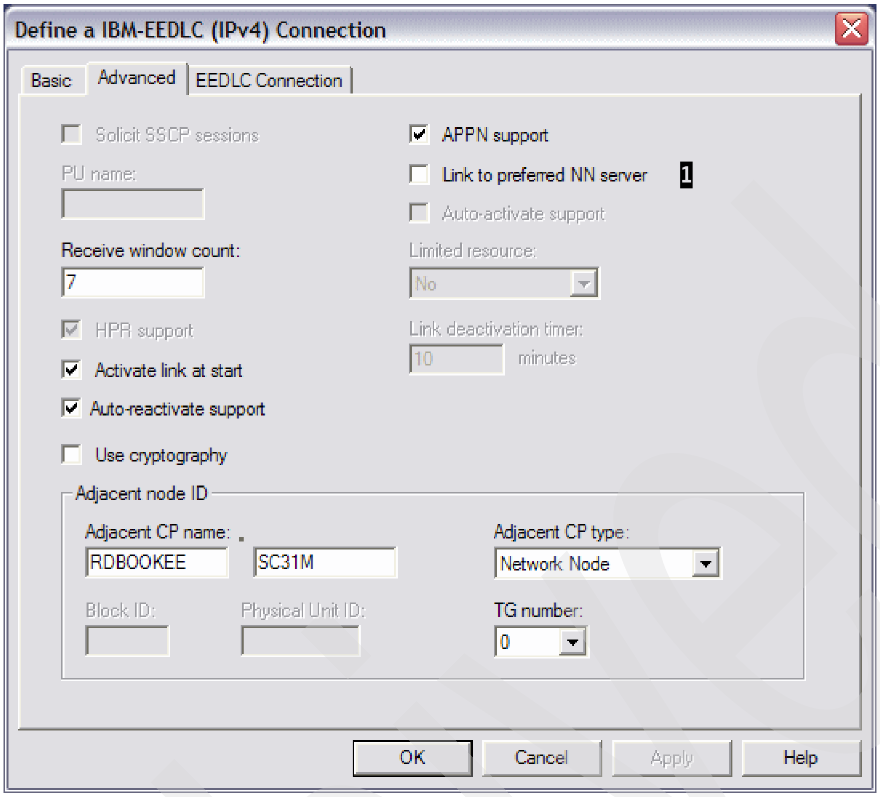
Maximum PIU size: 1461

Figure 5-8 Configuring EEDLC Connection tab for connection to SC30M

Here are explanatory notes to Figure 5-8:

- 1: This is the name of the connection (link station). This name is meaningful only to this server.
- 2: The IP address or host name of the remote host. In our scenario, we use the host name instead of the IP address of each system.

We define one more connection which will be connected to SC31M (see Figure 5-9).



Define a IBM-EEDLC (IPv4) Connection

☐ Solicit SSCP sessions

PU name:

Receive window count: 7

☑ HPR support

☑ Activate link at start

☑ Auto-reactivate support

☐ Use cryptography

☑ APPN support

☐ Link to preferred NN server **1**

☐ Auto-activate support

Limited resource: No

Link deactivation timer: 10 minutes

Adjacent node ID

Adjacent CP name: RDBOOKEE . SC31M

Adjacent CP type: Network Node

Block ID:

Physical Unit ID:

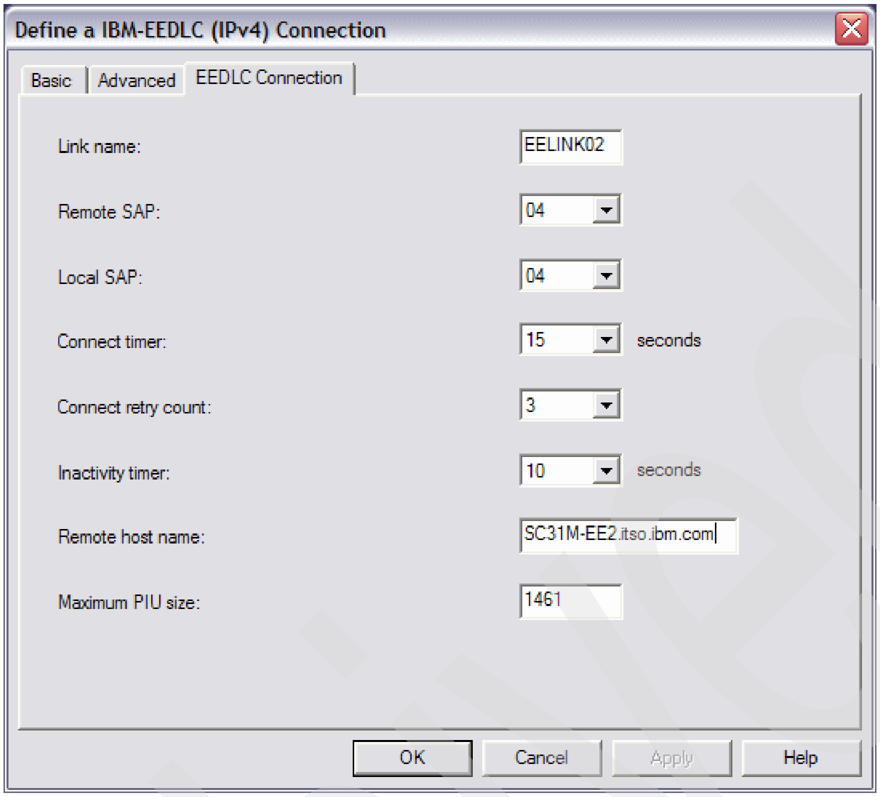
TG number: 0

Figure 5-9 Configuring Advanced tab of connection to SC31M

Here is an explanatory note to Figure 5-9:

- **1**: SC32M is a backup network node server

In Figure 5-9 and Figure 5-10, we define a static link station to SC31M.



Link name: EELINK02

Remote SAP: 04

Local SAP: 04

Connect timer: 15 seconds

Connect retry count: 3

Inactivity timer: 10 seconds

Remote host name: SC31M-EE2.itso.ibm.com

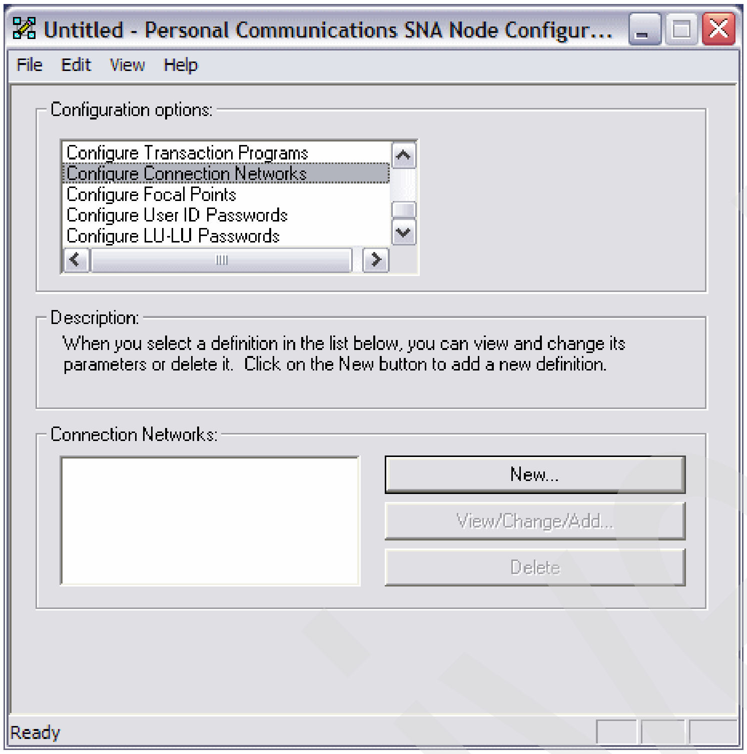
Maximum PIU size: 1461

Figure 5-10 Configuring EEDLC Connection tab for connection to SC31M

Configuring connection network

In our scenario, we use a connection network to communicate with the other nodes except the network node server and backup network node server. For more information about connection network, refer to “APPN connection networks” on page 38.

Select **Configure Connection Networks** in the SNA node operation utility to define a connection network as shown in Figure 5-11.



Configuration options:

- Configure Transaction Programs
- Configure Connection Networks
- Configure Focal Points
- Configure User ID Passwords
- Configure LU-LU Passwords

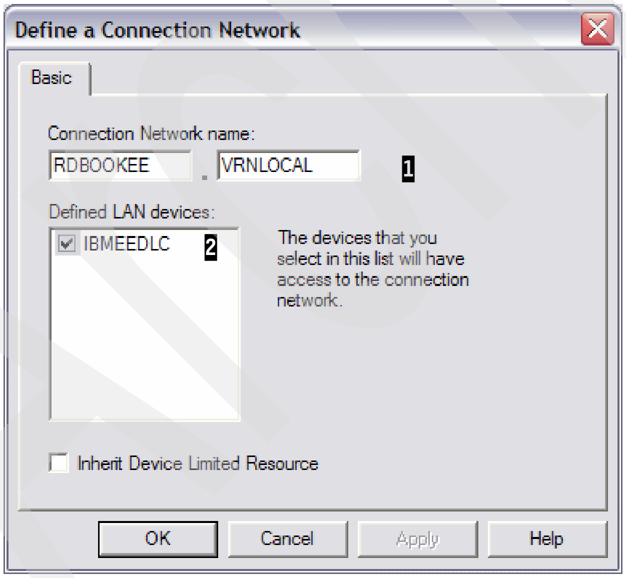
Description:

When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Connection Networks:

Figure 5-11 Configure Connection Networks menu

Click **New** to define a connection network and Figure 5-12 will appear.



Connection Network name: RDBOOKEE . VRNLOCAL

Defined LAN devices:

☑ IBMEEDLC

The devices that you select in this list will have access to the connection network.

☐ Inherit Device Limited Resource

Figure 5-12 Configuring a connection network

Here are explanatory notes to Figure 5-12:

- 1: The name of the connection network.
- 2: The port which is used for this connection network. In our scenario, IBMEEDLC will be used.

Configuring VTAM for connectivity using EE

The APPN environment supports dynamic definitions. This means you can connect to another APPN node without node-specific definitions. VTAM controls this behavior with the DYNPU parameter in the external communications adapter (XCA) major node definition. In our scenario, we do not use dynamic PU definitions, therefore we show the VTAM definitions for the PCOMM to connect to the z/OS LPARs in this section.

You must define the following in order to configure the EE connection for PCOMM:

- “Enterprise Extender XCA major node”
- “PU definition for PCOMM”

Enterprise Extender XCA major node

We use two IP addresses for EE in VTAM side (with multiple VIPA). In Example 5-1, we show the EE XCA major node for SC30M.

Example 5-1 EEXCA XCA major node for SC30M

```
EEXCA30         VBUILD TYPE=XCA
EEPORT30         PORT MEDIUM=HPRIP,                                  *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                *
               IPRESOLV=2,                                           *
               SRQTIME=15,SRQRETRY=3
*
EEGVL301       GROUP DIAL=YES,CALL=INOUT,                            *
               HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                      * 1
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                * 2
               TGP=EEXTCAMP,                                         *
               KEEPACT=YES,                                          *
               DYNPU=YES,                                            * 3
               UNRCHTIM=30,                                          *
               AUTOGEN=(16,EEM30,EEN30)
*
EEGVG301       GROUP DIAL=YES,CALL=INOUT,                            *
               HOSTNAME=SC30M-EE2.ITSO.IBM.COM,                      * 4
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                    * 5
               TGP=EEXTWAN,                                          *
               KEEPACT=YES,                                          *
               DYNPU=NO,                                             * 6
               UNRCHTIM=180,                                         *
               AUTOGEN=(16,EEX30,EEY30)
```

Here are explanatory notes to Example 5-1:

- 1: Host name for the first group.
- 2: The name of a connection network is RDBOOKEE.VRNLOCAL and its type is local. All participants of this connection network must be located within the same APPN topology subnetwork (the same NETID).
- 3: Dynamic definition of PU is allowed for this group.
- 4: Hostname for the second group.
- 5: The name of the connection network is W3IBMCOM.VRN and its type is global. Nodes located in different APPN topology subnetworks can share the same connection network.
- 6: Dynamic definition of PU is prohibited for this group, and a predefined PU is required.

We use the second group for the connection via a defined link stations. Since dynamic PU definition is prohibited for this group, we define a PU for PCOMM. For the connection network, we use the local connection network because PCOMM is in the same APPN subnetwork (use the same NETID) with other nodes. Using the connection network implies that DYNPU=YES; hence, no VTAM definition is required for PCOMM through connection network. For more information about VTAM definition and XCA major node, refer to 4.4.2, “Defining the XCA HPRIP major node” on page 142.

PU definition for PCOMM

Each system (SC30M, SC31M) should have the definition as shown in Example 5-2, if the DYNPU=NO in XCA major node of VTAM.

Example 5-2 PU Definition for PCOMM

```
EESWPCOM VBUILD TYPE=SWNET
EEPUPCOM PU    CPNAME=EECPPCOM,                        1          *
               NETID=RDBOOKEE,                         2          *
               TGN=06,                                 3          *
               CPCP=YES,                               4          *
               HPR=YES,                                           *
               DYNLU=YES,                              5          *
               ISTATUS=ACTIVE
```

Here are explanatory notes to Example 5-2:

- 1: The CP name of the PCOMM. This name must be the same as the name that was configured in node definition of PCOMM.
- 2: The NETID of this SNA or APPN environment.
- 3: The TG number which is used for this link. We defined the TG number as 0 in the IBM Communications Server for Windows side. With this configuration, the TG number will be decided when the XID is exchanged. The TG number will be 6.
- 4: The CP-CP sessions capability. Set this parameter to YES.
- 5: The dynamic LU definition feature. If this parameter is set to YES, then all the independent LUs defined in the PCOMM can be used in this z/OS VTAM without a CDRSC definition in VTAM. Otherwise, you have to define a CDRSC entry for each independent LU.

5.3.3 Configuring dependent LUs with DLUR

We define dependent LUs and DLUR PUs in this section. For further information about the dependent LU and the DLUR, refer to “Dependent LU Requester and Server (DLUR/DLUS)” on page 33. The topics covered are:

- “DLUR definition in IBM Communications Server for Windows”
- “DLUR definition in VTAM”

DLUR definition in IBM Communications Server for Windows

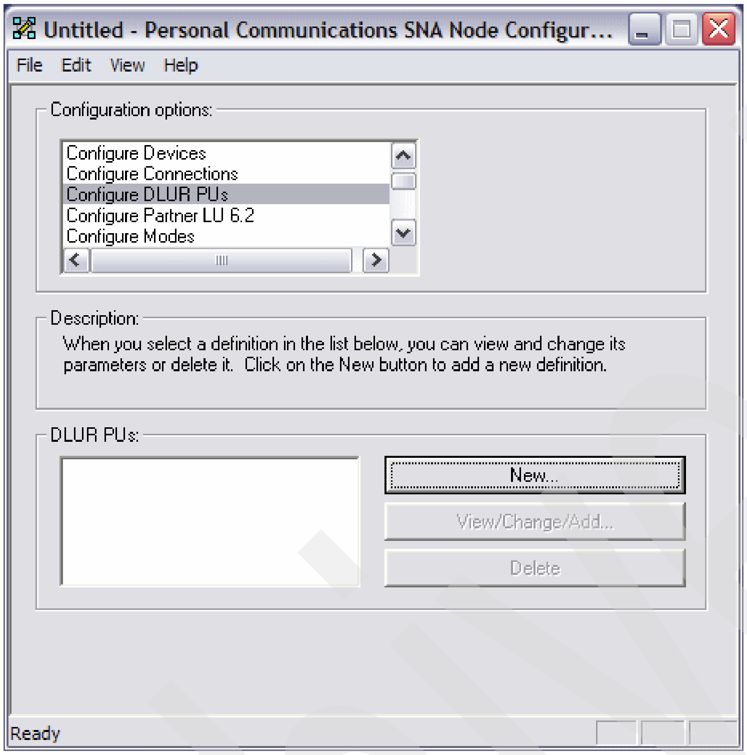
For using the dependent LUs, you must define the following in PCOMM:

- “DLUR PU”
- “Dependent LUs”

DLUR PU

DLUR PU is the type 2.0 PU which provides PU services to the dependent LUs.

To define the DLUR PU in PCOMM, select **Configure DLUR PUs** in the SNA node configuration utility as shown in Figure 5-13.



Configuration options:

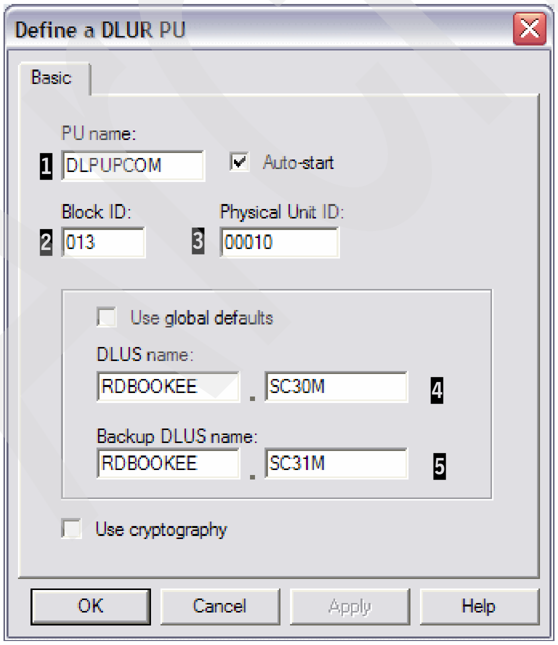
- Configure Devices
- Configure Connections
- Configure DLUR PUs
- Configure Partner LU 6.2
- Configure Modes

Description:
When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

DLUR PUs:

Figure 5-13   Configure DLUR PUs menu

Click **New** to create a DLUR PU and Figure 5-13 will appear.



PU name: 1 DLPUPCOM   ☑ Auto-start

Block ID: 2 013   3 Physical Unit ID: 00010

☐ Use global defaults

DLUS name: RDBOOKEE . SC30M 4

Backup DLUS name: RDBOOKEE . SC31M 5

☐ Use cryptography

Figure 5-14   Configuring a DLUR PU

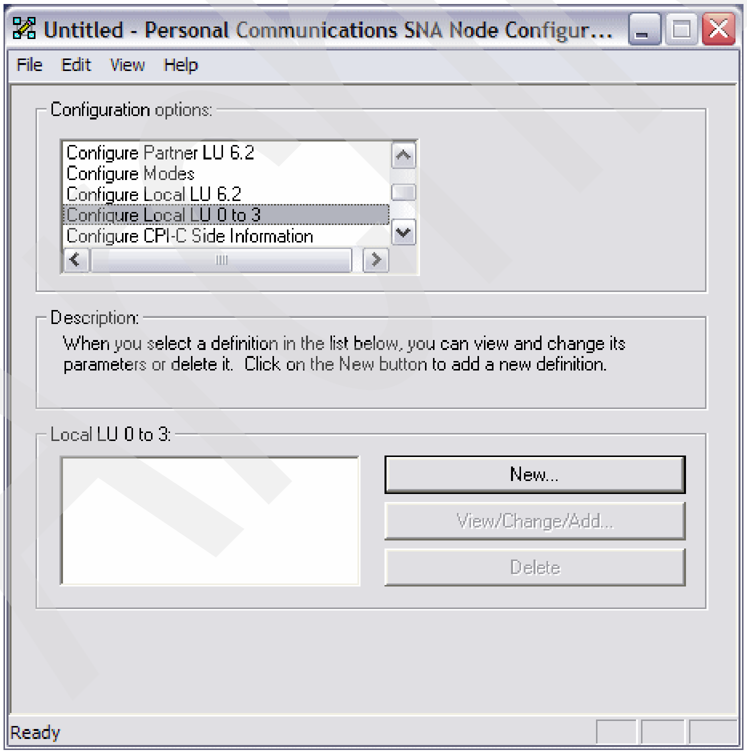
Here are explanatory notes to Figure 5-14:

- 1: This is the PU name for this DLUR PU for dependent LUs. This name is meaningful only in this PCOMM, so you can use a different name from the PU name in the VTAM definition. But for management purpose, we recommend using the same name as your VTAM definition.
- 2~3: This is the PU identification number. VTAM has IDBLK (ID block) and IDNUM (ID number) parameters in its PU definition for this client, and the two definitions must be the same. In this case, IDBLK is `013`, and IDNUM is `00010`. This value should be unique in a VTAM.
- 4: This is the SSCP name of the primary Dependent LU Server (DLUS) name. PCOMM will try to establish a CPSVRMGR pipe (DLUS-DLUR sessions) with this DLUS. When this attempt fails, it will try the backup DLUS if it is available.
- 5: This is the SSCP name of the backup DLUS. If you want to use the backup DLUS, then the same PU and LU definitions must be configured in the VTAM of the backup DLUS.

Note: You can define multiple DLUR PUs in a node. Because a PU can support only 255 dependent LUs, if you want more LUs you must define more DLUR PUs to the same primary/backup DLUS pair. Moreover, it is possible to define more DLUR PUs to the other primary/backup DLUS pair.

Dependent LUs

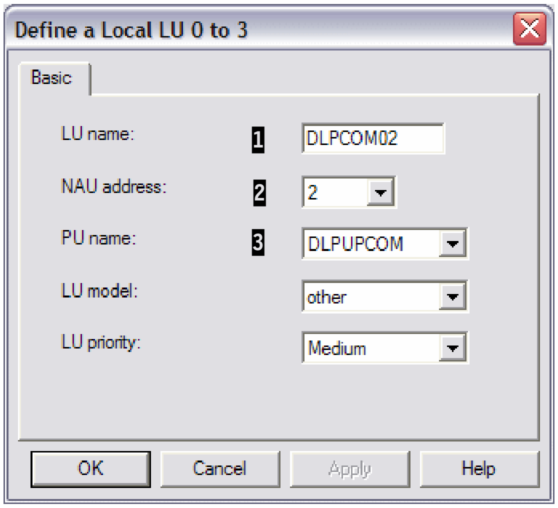
To define dependent LUs select **Configure Local LU 0 to 3** in the SNA node configuration utility as shown in Figure 5-15.



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Figure 5-15 Configure Local LU 0 to 3 menu

Click **New** to define a local dependent LU and Figure 5-16 will appear.



LU name: 1 DLPCOM02

NAU address: 2 2

PU name: 3 DLPUPCOM

LU model: other

LU priority: Medium

Figure 5-16 Configuring a local dependent LU

Here are explanatory notes to Figure 5-16:

- 1: The name of the dependent LU. This LU name need not be the same as the LU name in VTAM definition. But for maintenance purpose, we recommend you use the same LU name on both the sides.
- 2: The address of this local LU. This number must match the LOCADDR parameter in a VTAM definition for this dependent LU.
- 3: The name of the DLUR PU already defined.

Note: The LU name defined here is only meaningful for this PCOMM. In other words, there is no need to match these LU names with the LU names in the VTAM definition. Only the LU number (LOCADDR) must be the same on both the sides. But for management purpose, we recommend using the same name on both the sides.

DLUR definition in VTAM

In VTAM, you should define the type 2.0 PU, and the dependent LUs in a switched major node. In our scenario, we make a switched major node as shown in Example 5-3.

Example 5-3 Switched major node definition for DLUR

```
DLSWPCOM VBUILD TYPE=SWNET
DLPUPCOM PU     PUTYPE=2,IDBLK=013,IDNUM=00010,USSTAB=USSSNAEE,          * 1
                MODETAB=ALLMODES,DLOGMOD=DYNHIGH,ANS=CONT                  2
DLPCOM02 LU     LOCADDR=2                                                  3
DLPCOM03 LU     LOCADDR=3
DLPCOM04 LU     LOCADDR=4
DLPCOM05 LU     LOCADDR=5
DLPCOM06 LU     LOCADDR=6
```

Here are explanatory notes to Example 5-3:

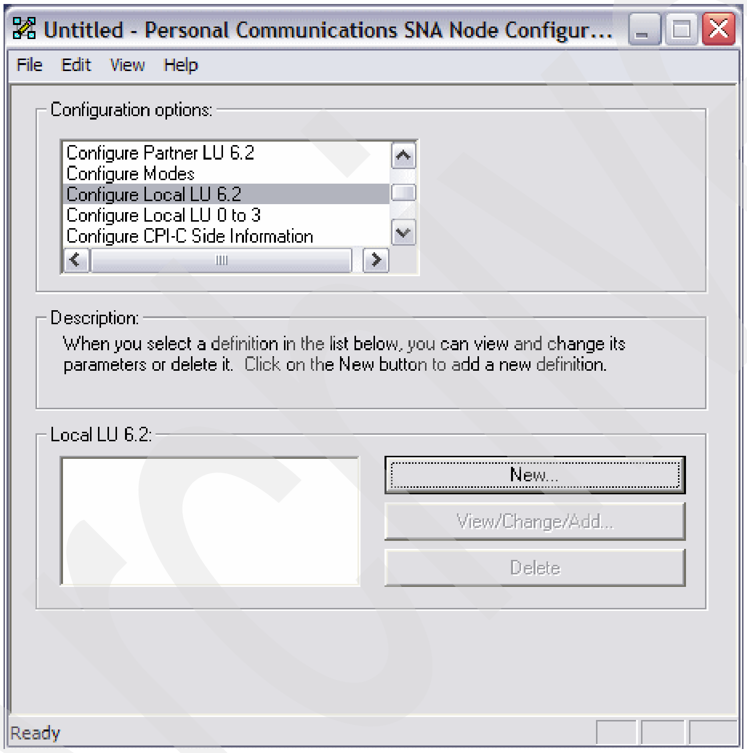
- 1: IDBLK and IDNUM must match the PU ID in the DLUR PU definition in PCOMM.
- 2: ANS=CONT is required for DLUR PUs. If the default value (ANS=STOP) is used, SSCP-PU session and SSCP-LU sessions will be broken in case the primary DLUS is lost. This results in session outages during takeover to backup DLUS.

- 2: The name of the LU may be different from the one in PCOMM, but the LOCADDR must match the LU number in DLUR PU definition of PCOMM. However, we recommend using the same names for management purposes.

5.3.4 Configuring independent LUs

An independent LU has the capability to communicate with another independent LU without the SSCP (VTAM). So, independent LU is not a resource of VTAM, but a resource of the node in which the independent LU is defined. There are two ways of handling these independent LUs whether the DYNLU parameter in VTAM is *Yes* or *No*. If DYNLU is *Yes*, then no definition is required for VTAM, and VTAM will generate a dynamic definition of a CDRSC entry for that independent LU. If DYNLU is *No*, then there must be a predefined CDRSC entry in the VTAM for that independent LU.

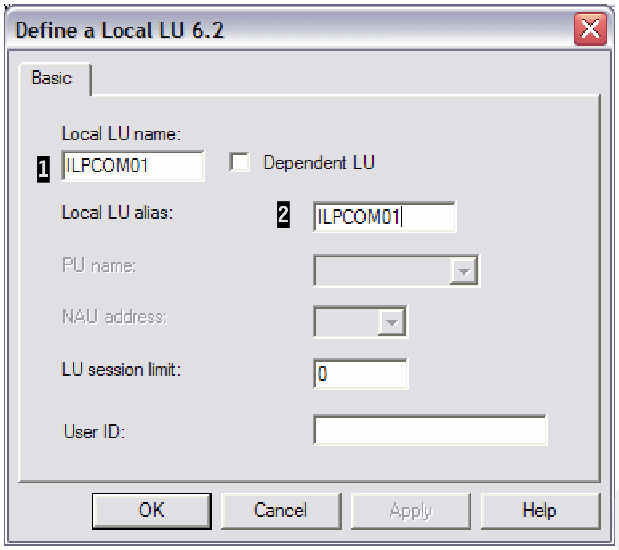
Select **Configure Local LU 6.2** in SNA node configuration utility as shown in Figure 5-17.



When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Figure 5-17 Configure Local LU 6.2 menu

Click **New** to define a new local LU 6.2 (see Figure 5-18).



Local LU name: ILPCOM01

□ Dependent LU

Local LU alias: ILPCOM01

PU name:

NAU address:

LU session limit: 0

User ID:

Figure 5-18 Configuring independent LU

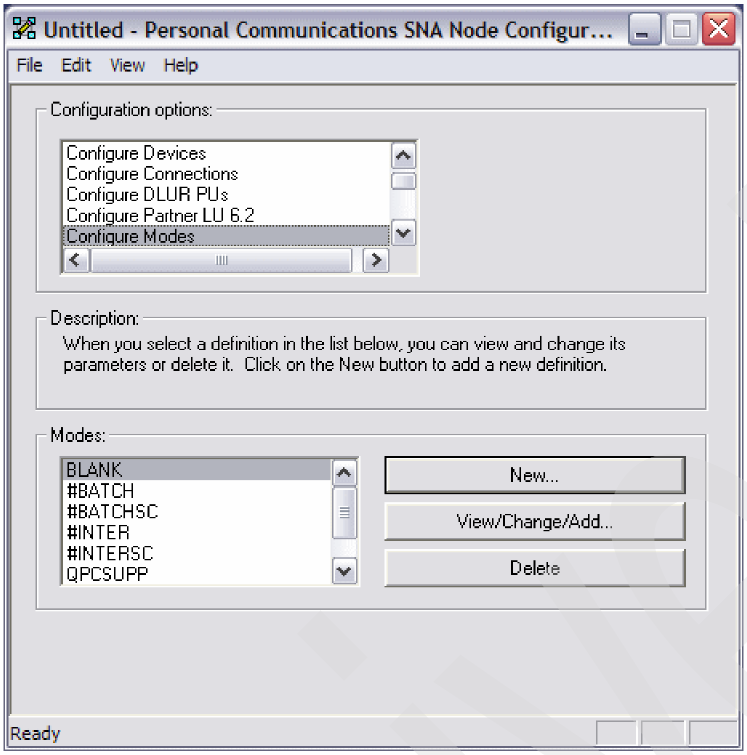
Here are explanatory notes to Figure 5-18:

- 1: This is the name of the LU. This LU name will also be used in z/OS.
- 2: This is the alias of the LU. This value is meaningful only to this PCOMM.

5.3.5 Configuring LOGMODEs

Sometimes, your application can use the customized LOGMODE, which is not shipped by IBM. In this case you can define a customized LOGMODE in the IBM Communications Server for Windows. You can define logmode for type 6.2 LUs only.

Select **Configure Modes** in SNA node configuration utility as shown in Figure 5-19.



Configuration options:

- Configure Devices
- Configure Connections
- Configure DLUR PUs
- Configure Partner LU 6.2
- Configure Modes

Description:

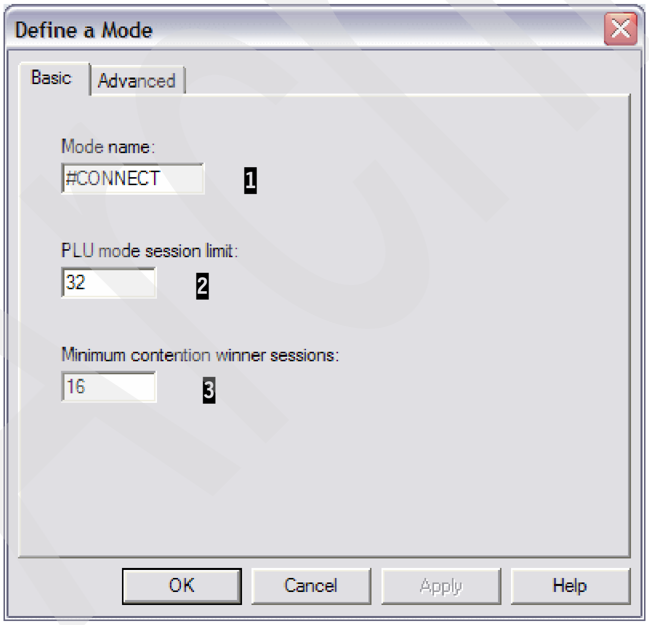
When you select a definition in the list below, you can view and change its parameters or delete it. Click on the New button to add a new definition.

Modes:

- BLANK
- #BATCH
- #BATCHSC
- #INTER
- #INTERSC
- QPCSUPP

Figure 5-19 Configure Modes menu

Click **New** to create a mode (see Figure 5-20).



Mode name: #CONNECT 1

PLU mode session limit: 32 2

Minimum contention winner sessions: 16 3

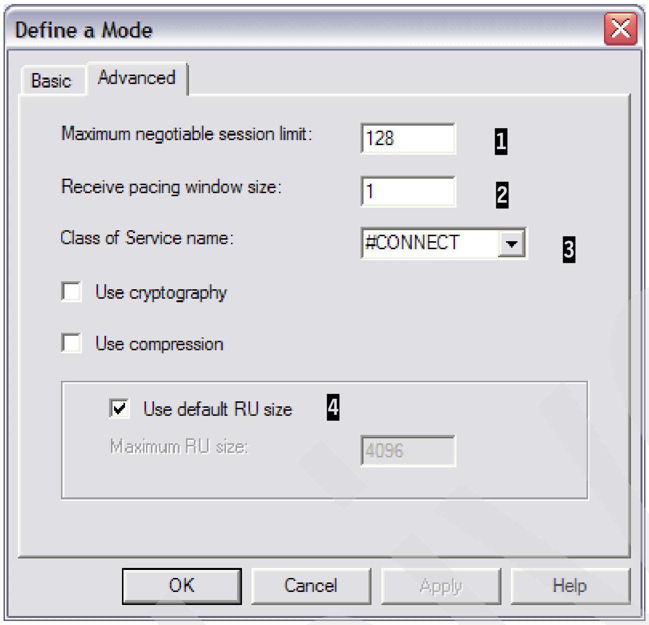
Figure 5-20 Configuring Basic tab of mode

Here are explanatory notes to Figure 5-20:

- 1: The name of this logmode.
- 2: The default session limit of this logmode. This is the maximum number of sessions permitted between a pair of LUs using this mode, even with CNOS negotiation.

- 3: The minimum contention winner sessions. The number of sessions (up to the session limit) that PCOMM must reserve for use by the local LU as the contention winner. Specify 0 if you do not want to reserve the contention winner sessions.

Select the **Advanced** tab to proceed with the configuration (see Figure 5-21).



Maximum negotiable session limit: 128

Receive pacing window size: 1

Class of Service name: #CONNECT

□ Use cryptography

□ Use compression

☑ Use default RU size

Maximum RU size: 4096

Figure 5-21 Configuring the Advanced tab of mode

Here are explanatory notes to Figure 5-21:

- 1: The maximum number of sessions which will be supported by this logmode. In many cases this value is set to the same value as the initial session limit. Through CNOS (Change Number of Sessions), the maximum number of sessions can be changed.
- 2: The initial pacing window size. The initial number of request units (RUs) that the local LU can receive before it must send a pacing response to the remote LU in adaptive pacing. If fixed pacing is used, this value specifies the receive pacing window.
- 3: You can specify how many sessions to activate automatically for each pair of LUs that use this mode. This value is used when CNOS exchange is initiated implicitly.
- 4: Name of the class of service (COS) to request when activating sessions on this mode. This must be defined in APPNCOS in VTAM.
- 5: You can specify whether IBM Communications Server for Windows uses the maximum RU size upper limit parameter to define the maximum RU size. If you select this check box then you can specify the upper bound for the maximum RU size. If you do not restrict the maximum RU size, then IBM Communications Server for Windows sets the upper bound for the maximum RU size to the largest value that can be accommodated in the link BTU size.

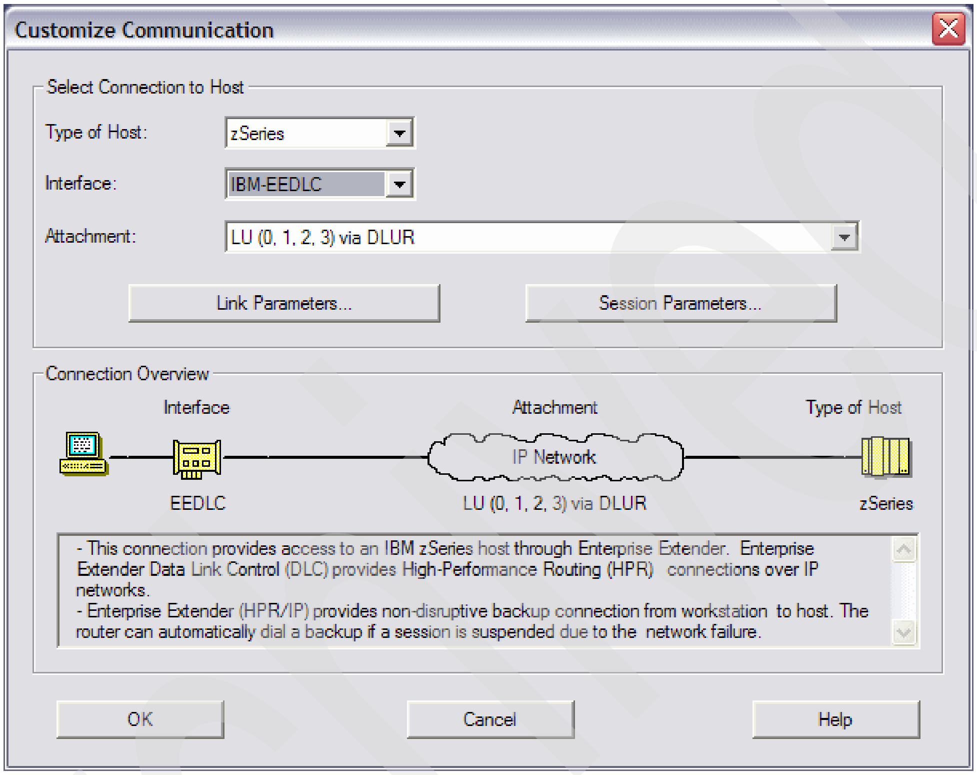
In Figure 5-20 and Figure 5-21, we define a new logmode named #CONNECT which uses the #CONNECT as its class of service.

5.3.6 Configuring quick 3270 Sessions

PCOMM provides a quick way to define the mainframe session using EE and DLUR. A dependent LU will be used for this 3270 session, so the VTAM definition is also needed to connect. We use the same configuration as defined in the previous sections in this chapter. For more information about each parameter refer to 5.3, “Configuring EE in Personal Communications (PCOMM)” on page 153.

PCOMM definition

To configure a new connection profile, select **Start** → **All Programs** → **IBM Personal Communications** → **Start or Configure Sessions**.



Customize Communication

Select Connection to Host

Type of Host: zSeries

Interface: IBM-EEDLC

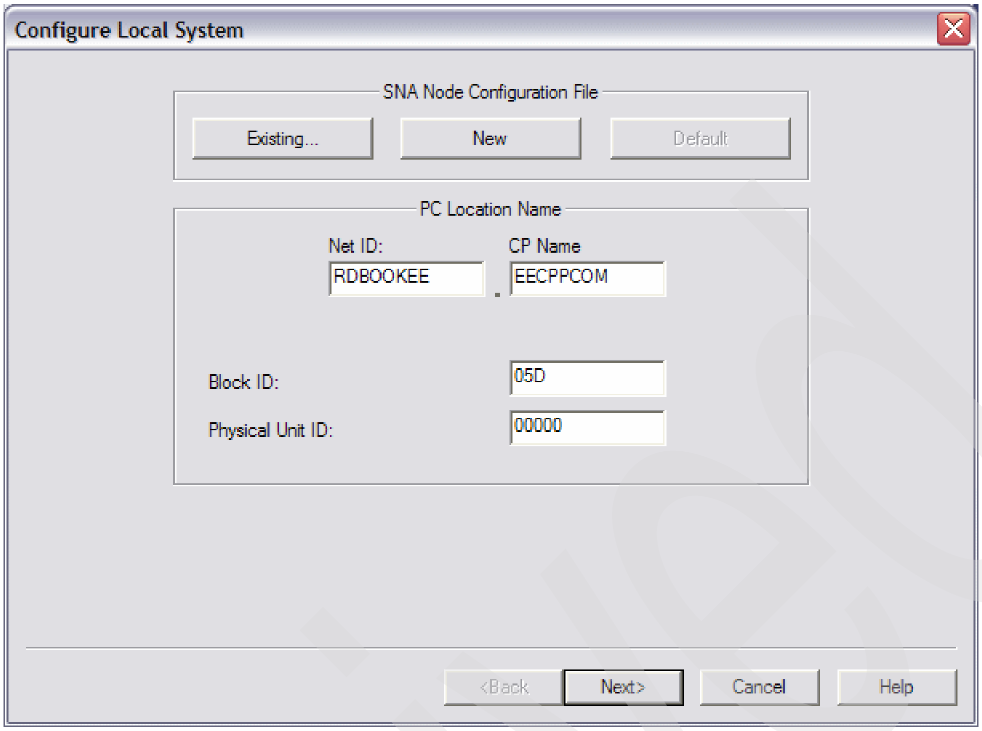
Attachment: LU (0, 1, 2, 3) via DLUR

Connection Overview

- This connection provides access to an IBM zSeries host through Enterprise Extender. Enterprise Extender Data Link Control (DLC) provides High-Performance Routing (HPR) connections over IP networks.
- Enterprise Extender (HPR/IP) provides non-disruptive backup connection from workstation to host. The router can automatically dial a backup if a session is suspended due to the network failure.

Figure 5-22 Configuring a new session

In Figure 5-22, select the `IBM-EEDLC` for Interface, and then click **OK**. PCOMM will generate the device for this EE connection automatically.



Net ID: RDBOOKEE

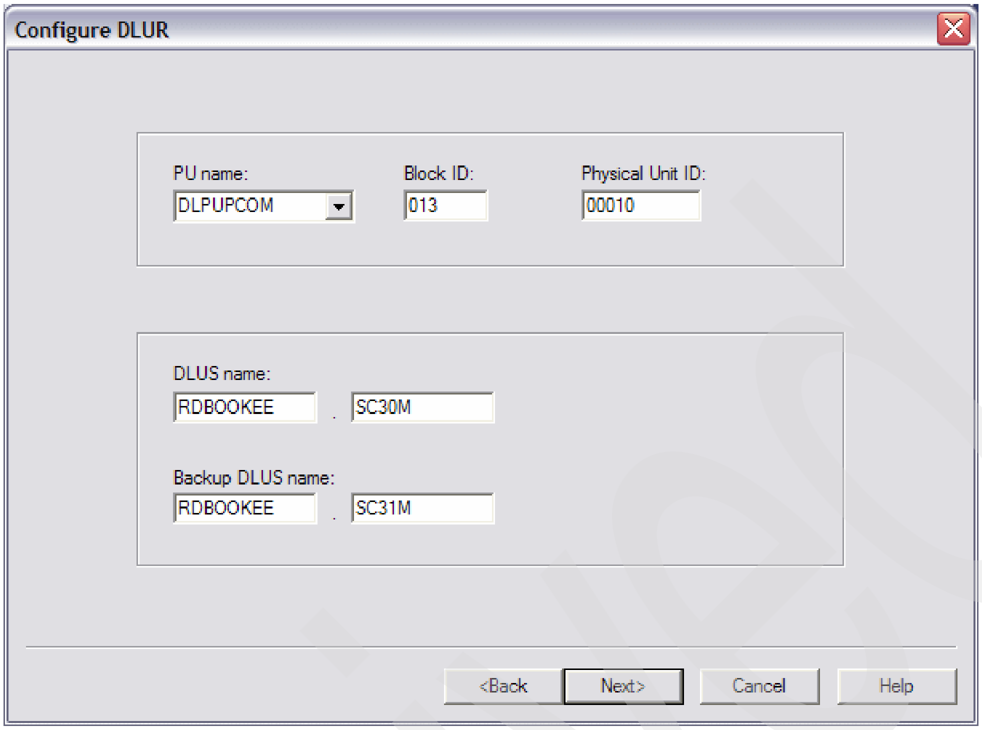
CP Name: EECPPCOM

Block ID: 05D

Physical Unit ID: 00000

Figure 5-23 Configuring the node parameters

Figure 5-23 is the node configuration window. Fill the Net ID and CP Name field which will be used for this node. Net ID is the network ID for this SNA or APPN environment, and CP Name is the name of this PCOMM. CP Name must be unique in an SNA or APPN environment, so it will be decided with caution. Click **Next**. PCOMM will generate the node definition automatically.



PU name: DLPUPCOM

Block ID: 013

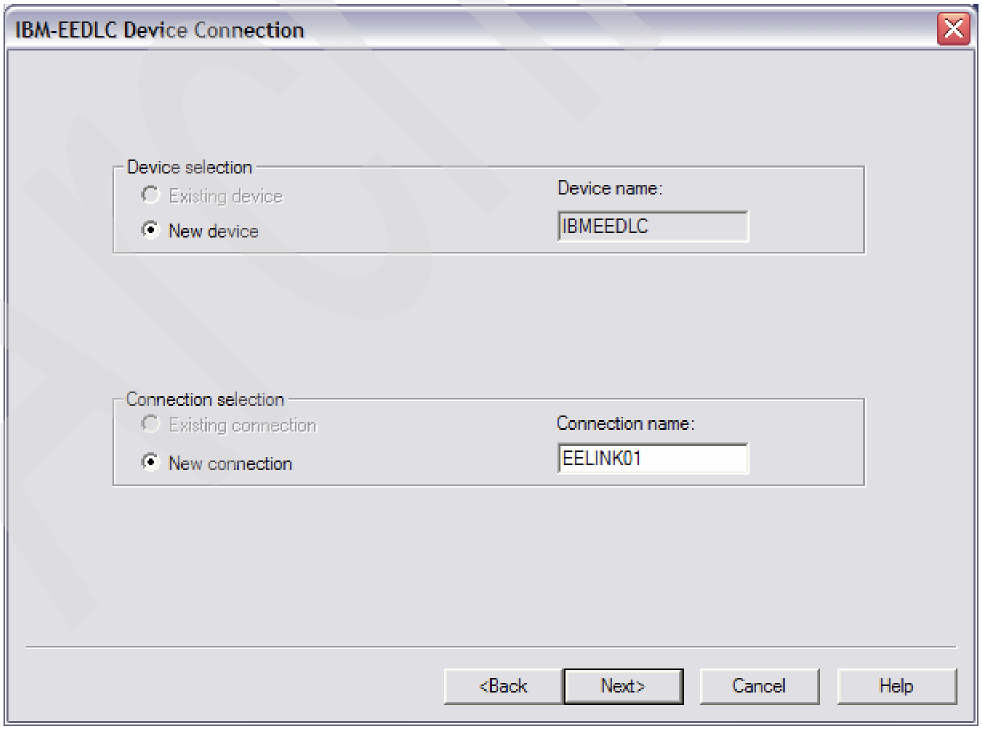
Physical Unit ID: 00010

DLUS name: RDBOOKEE . SC30M

Backup DLUS name: RDBOOKEE . SC31M

Figure 5-24 Configuring the DLUR PU

Figure 5-24 is the definition of DLUR PU. We use the same definitions as Figure 5-14 on page 165.



○ Existing device

● New device

Device name: IBMEEDLC

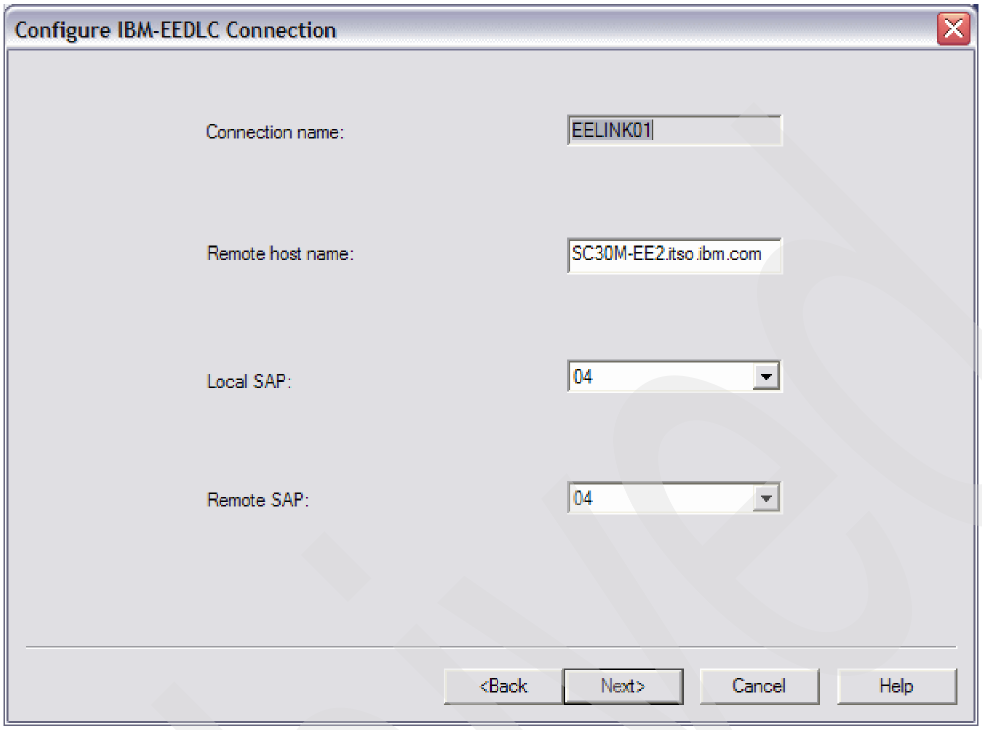
○ Existing connection

● New connection

Connection name: EELINK01

Figure 5-25 Configuring device and connection

In Figure 5-25, specify which connection is used for this session. If you specify a new connection name, it will be created.



Connection name: EELINK01

Remote host name: SC30M-EE2.itso.ibm.com

Local SAP: 04

Remote SAP: 04

Figure 5-26 Configuring connection

Figure 5-26 is the definition for a connection. We fill the Remote host name field as `SC30M-EE2.itso.ibm.com`. Then click **Finish**.

Configuration is completed successfully. Now, you can save this configuration and connect to the mainframe.

5.4 Verifying and managing EE in Personal Communications (PCOMM)

Most administration work will be done on the SNA Node Operations utility. You can start the SNA Node Operations utility by selecting **Start** → **All programs** → **IBM Personal Communications** → **Administrative and PD Aids** → **SNA Node Operations**.

The topics covered in this section are:

- "Starting resources"
- "Verifying resources"
- "Stopping resources"
- "Backing up your configuration"

5.4.1 Starting resources

All resources of PCOMM can be started by **start** right-click menu of them. But we recommend that you use the "initially active" options of each resource, because it simplifies management. For LU-LU sessions, it is better to use batch type scripts.

For starting the node, select **Operations** → **Start Node** as shown in Figure 5-27.

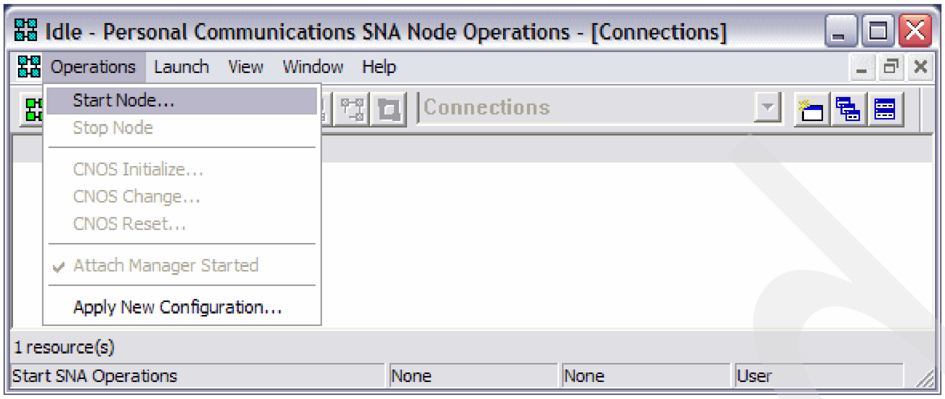


Figure 5-27 Starting node

5.4.2 Verifying resources

You can easily monitor SNA/APPN activities with SNA Node Operations. In this section we describe the following:

- "Verifying connectivity to other nodes"
- "Verifying DLUR PU and dependent LUs"
- "Verifying HPR Path Switch"
- "Verifying the backup DLUS"

Verifying connectivity to other nodes

You can verify connectivity by selecting **Connections** in the SNA node operations utility as shown in Figure 5-28.

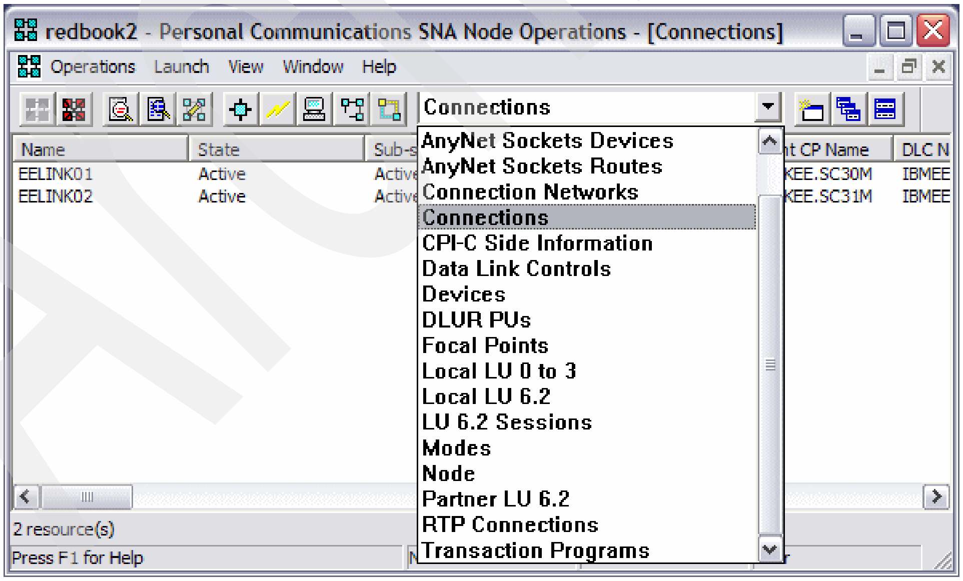
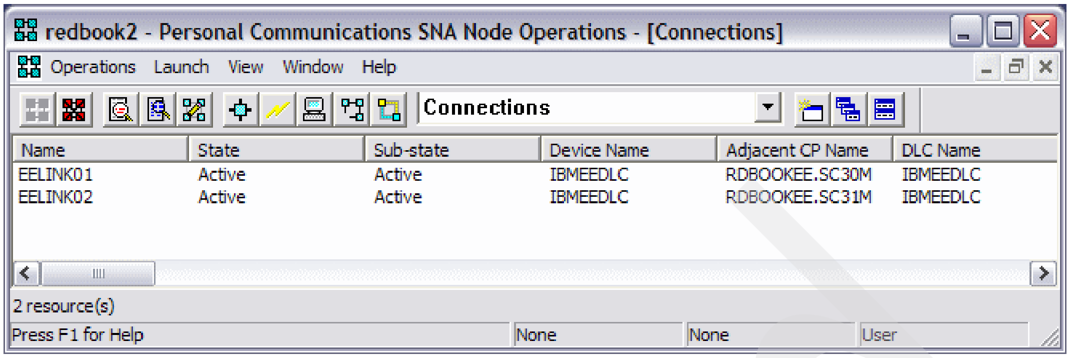


Figure 5-28 Connections menu in SNA node operations


redbook2 - Personal Communications SNA Node Operations - [Connections]
Operations Launch View Window Help
Connections

Name	State	Sub-state	Device Name	Adjacent CP Name	DLC Name
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	IBMEEDLC

2 resource(s)

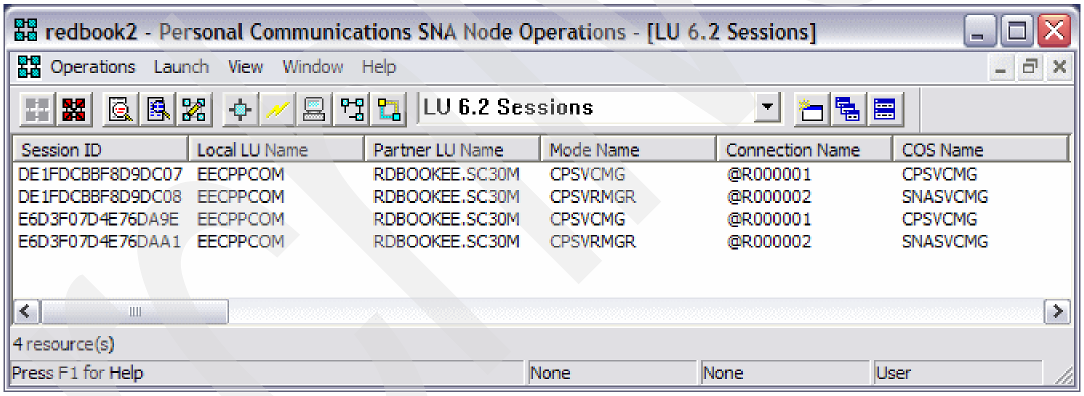
Press F1 for Help | None | None | User

Figure 5-29 Verifying connections

In Figure 5-29, you can see that two connections are `Active` under State. EELINK01 is connected to the SC30M, and EELINK02 to the SC31M by IBMEEDLC DLC.

If an end node is connected to its network node server, then two CP-CP sessions are established between the two nodes. In our scenario, the preferred network node server is SC30M. With these CP-CP sessions, an end node can get topology and directory information from the network node server.

For verifying the CP-CP sessions, select **Local LU 6.2** in the SNA node operations utility.


redbook2 - Personal Communications SNA Node Operations - [LU 6.2 Sessions]
Operations Launch View Window Help
LU 6.2 Sessions

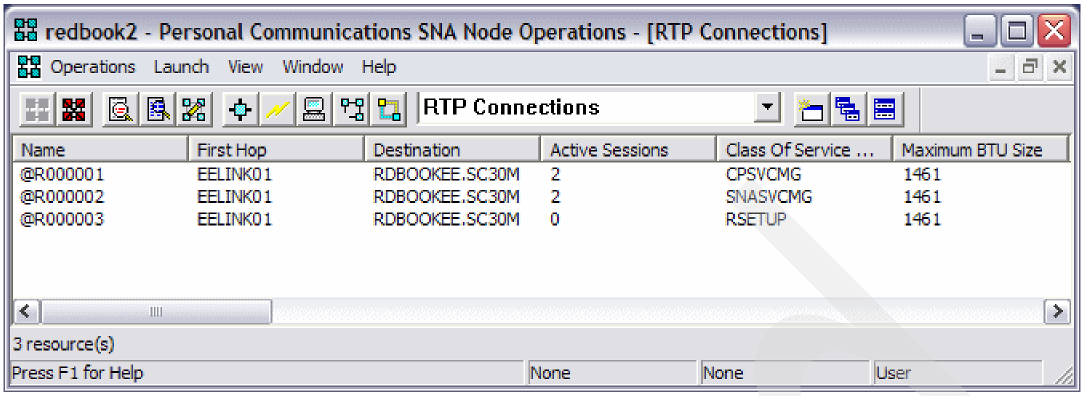
Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FDCBBF8D9DC07	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG
DE1FDCBBF8D9DC08	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
E6D3F07D4E76DA9E	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG
E6D3F07D4E76DAA1	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG

4 resource(s)

Press F1 for Help | None | None | User

Figure 5-30 Verifying the CP-CP sessions

Figure 5-30 shows two CP-CP sessions with the SC30M (preferred network node server) through @R000001 RTP connection. Two LU 6.2 sessions which use the CPSVCMG as their mode are the CP-CP sessions. An RTP connection is a logical link station to the opposite endpoint which uses a specific class of service (COS). To can get information about the RTP connections by selecting **RTP Connections** in the SNA node operations utility.



redbook2 - Personal Communications SNA Node Operations - [RTP Connections]

RTP Connections

Name	First Hop	Destination	Active Sessions	Class Of Service ...	Maximum BTU Size
@R000001	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG	1461
@R000002	EELINK01	RDBOOKEE.SC30M	2	SNASVCMG	1461
@R000003	EELINK01	RDBOOKEE.SC30M	0	RSETUP	1461

3 resource(s)

Press F1 for Help | None | None | User

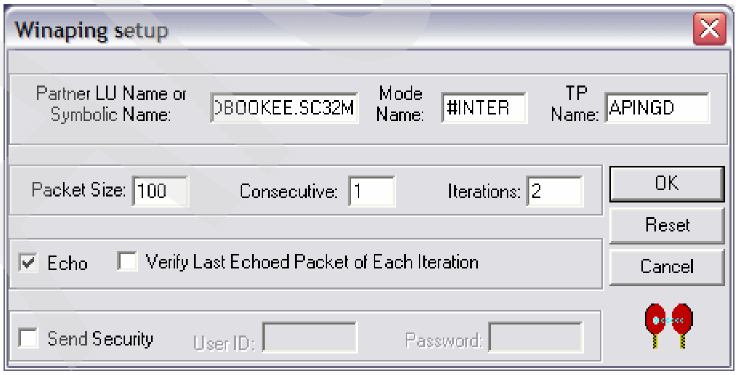
Figure 5-31 Verifying the RTP connections

Figure 5-31 shows three RTP Connections in this PCOMM. The first connection, @R000001 is an RTP connection for the CP-CP sessions with SC30M and the first hop of this RTP path is EELINK01 connection. This path of the RTP Connection may change when the topology is changed. We be describing this HPR Path Switch feature later. For more information about HPR and RTP, refer to 1.2.1, “Positioning APPN and HPR” on page 4.

> **Note:** An RTP connection is created between two session endpoints. There may be several nodes between two endpoints. The SNA node operations utility shows the next hop of that RTP connection only.

Next, we verify if the connection network works properly. The connection network does not support the CP-CP sessions, so you should have predefined link stations to the network node server and backup network node server, if it exists. We have another z/OS LPAR in our scenario, SC32M. We did not define a static connection to SC32M, but a dynamic connection to SC32M can be generated if needed, because SC32M is connected to the same connection network, RDBOOKEE.VRNLOCAL, as this PCOMM.

The ping program for APPC, called *winaping* is provided for reachability test. Open a command prompt and type `winaping` or select **Start** → **All Programs** → **IBM Personal Communications** → **Utilities** → **Check Connection APING**.



Winaping setup

Partner LU Name or Symbolic Name:)BOOKEE.SC32M
Mode Name: #INTER
TP Name: APINGD

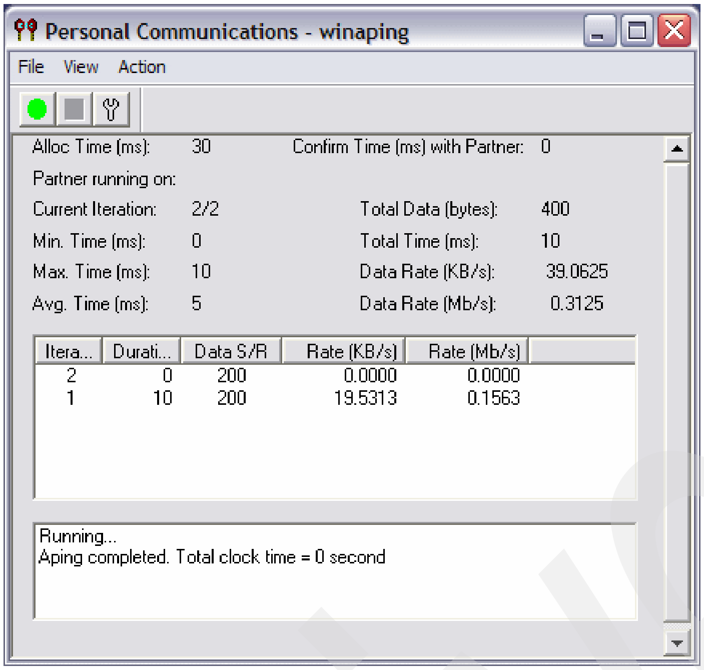
Packet Size: 100
Consecutive: 1
Iterations: 2

☑ Echo
□ Verify Last Echoed Packet of Each Iteration

□ Send Security
User ID:
Password:

Figure 5-32 Winaping setup

Type `RDBOOKEE.SC32M` (the fully qualified CP name of SC32M) in the partner LU Name field, and then click **OK**. Click the start button (the left-most one), then Figure 5-33 appears.



Alloc Time (ms): 30 Confirm Time (ms) with Partner: 0

Partner running on:

Current Iteration: 2/2 Total Data (bytes): 400

Min. Time (ms): 0 Total Time (ms): 10

Max. Time (ms): 10 Data Rate (KB/s): 39.0625

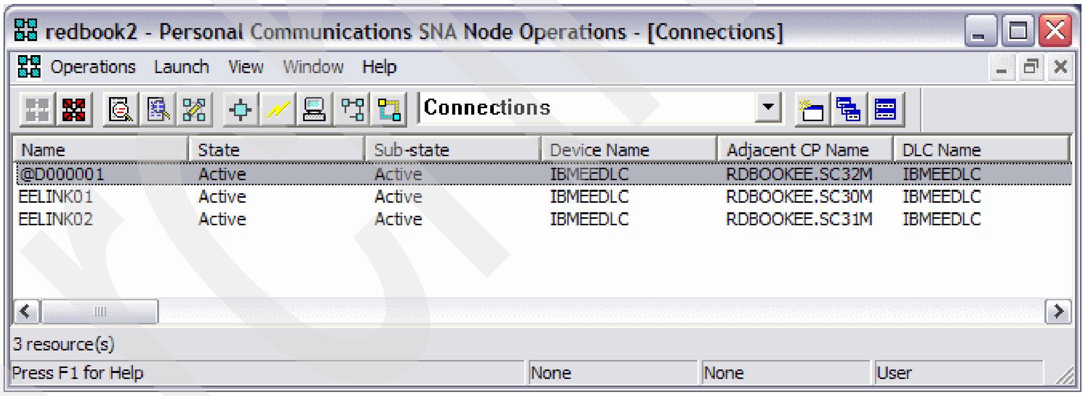
Avg. Time (ms): 5 Data Rate (Mb/s): 0.3125

Itera...	Durati...	Data S/R	Rate (KB/s)	Rate (Mb/s)
2	0	200	0.0000	0.0000
1	10	200	19.5313	0.1563

```
Running...
Aping completed. Total clock time = 0 second
```

Figure 5-33 The result of APING

Figure 5-33 shows that APING is successfully executed and the connection to SC32M is successful. Now check the connection status in the SNA node operations utility.


redbook2 - Personal Communications SNA Node Operations - [Connections]
Operations Launch View Window Help
Connections

Name	State	Sub-state	Device Name	Adjacent CP Name	DLC Name
@D000001	Active	Active	IBMEEDLC	RDBOOKEE.SC32M	IBMEEDLC
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	IBMEEDLC

3 resource(s)

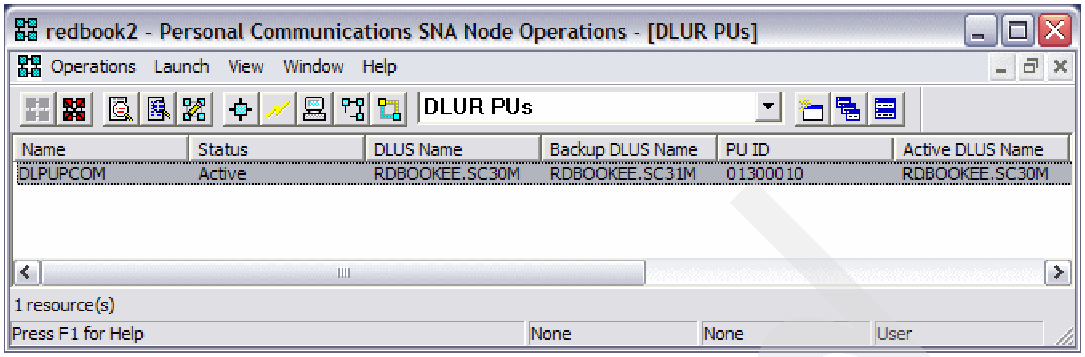
Press F1 for Help None None User

Figure 5-34 Dynamic connection to SC32M created through connection network

Figure 5-34 shows one newly-created connection named @D000001. This connection is created through connection network. If connection network is not used in this node, the APING utility will use two hop paths, via SC30M to SC32M or SC31M to SC32M.

Verifying DLUR PU and dependent LUs

You can verify the current status of DLUR PU by selecting **DLUR PUs** in the SNA node operations utility as shown in Figure 5-35.


redbook2 - Personal Communications SNA Node Operations - [DLUR PUs]
Operations
Launch
View
Window
Help
DLUR PUs

Name	Status	DLUS Name	Backup DLUS Name	PU ID	Active DLUS Name
DLPUPCOM	Active	RDBOOKEE.SC30M	RDBOOKEE.SC31M	01300010	RDBOOKEE.SC30M

1 resource(s)

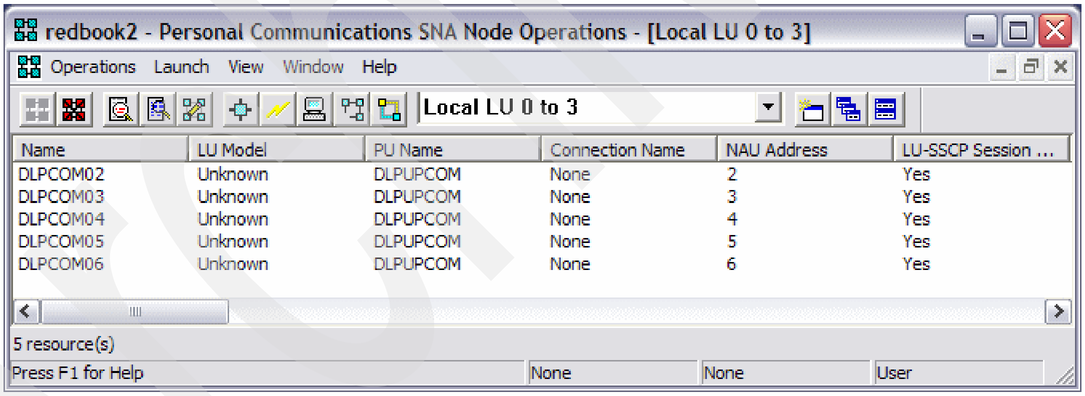
Press F1 for Help | None | None | User

Figure 5-35 Verifying DLUR PU

Figure 5-35 shows that only one DLUR PU is `Active` under Status. The primary DLUS is SC30M, and the backup DLUS is SC31M. Active DLUS is the primary DLUS, SC30M at this time. We simulate the problem of primary DLUS later.

Figure 5-30 on page 177 shows that there are two sessions which are established with the SC30M using the CPSVRMGR mode. These two sessions are called CPSVRMGR pipe, and through this pipe, SSCP-PU and SSCP-LU sessions are set up. These two sessions are using @R000002 as their RTP connection, which is using the SNASVCMG class of service (refer to Figure 5-31 on page 178).

Now we verify the status of the dependent LUs. Select **Local LU 0 to 3** in the SNA node operations utility.


redbook2 - Personal Communications SNA Node Operations - [Local LU 0 to 3]
Operations
Launch
View
Window
Help
Local LU 0 to 3

Name	LU Model	PU Name	Connection Name	NAU Address	LU-SSCP Session ...
DLPCOM02	Unknown	DLPUPCOM	None	2	Yes
DLPCOM03	Unknown	DLPUPCOM	None	3	Yes
DLPCOM04	Unknown	DLPUPCOM	None	4	Yes
DLPCOM05	Unknown	DLPUPCOM	None	5	Yes
DLPCOM06	Unknown	DLPUPCOM	None	6	Yes

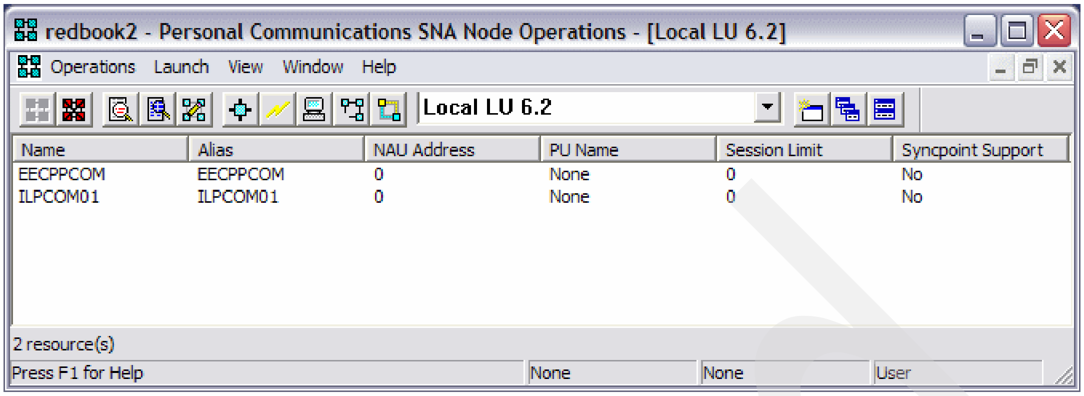
5 resource(s)

Press F1 for Help | None | None | User

Figure 5-36 Verifying dependent LUs

Figure 5-36 shows five dependent LUs which are successfully connected to z/OS (SC30M). LU-SSCP sessions are already established. Applications can now use these LUs for their purpose. When the LU-LU sessions are set up, a new RTP connection will be generated for the LU-LU sessions.

For independent LUs, select Local LU 6.2 menu in SNA node operations utility.


redbook2 - Personal Communications SNA Node Operations - [Local LU 6.2]
Operations Launch View Window Help
Local LU 6.2
2 resource(s)
Press F1 for Help
None
None
User

Name	Alias	NAU Address	PU Name	Session Limit	Syncpoint Support
EECPPCOM	EECPPCOM	0	None	0	No
ILPCOM01	ILPCOM01	0	None	0	No

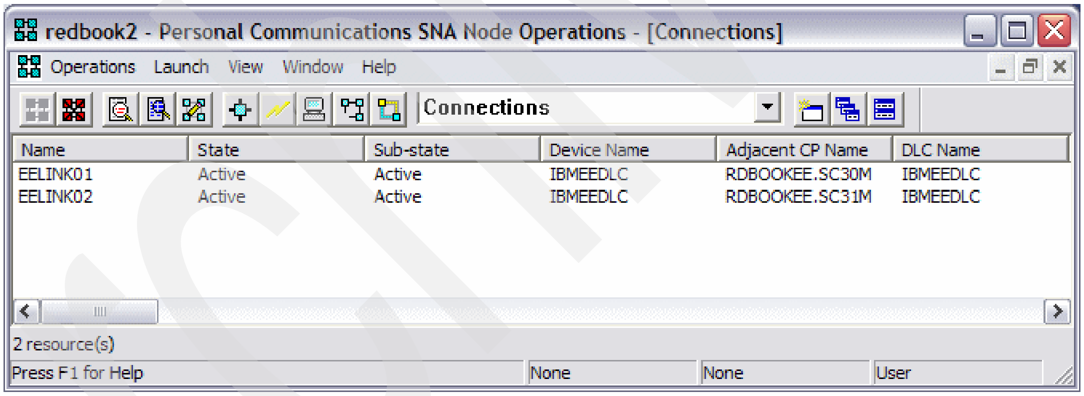
Figure 5-37 Verifying independent LU

Figure 5-37 shows two independent LUs in this node. EECPPCOM is the default LU for this node, and ILPCOM01 is the defined LU in the previous section, in this chapter.

Verifying HPR Path Switch

The RTP path can be changed if the topology is changed. We show this HPR Path Switch.

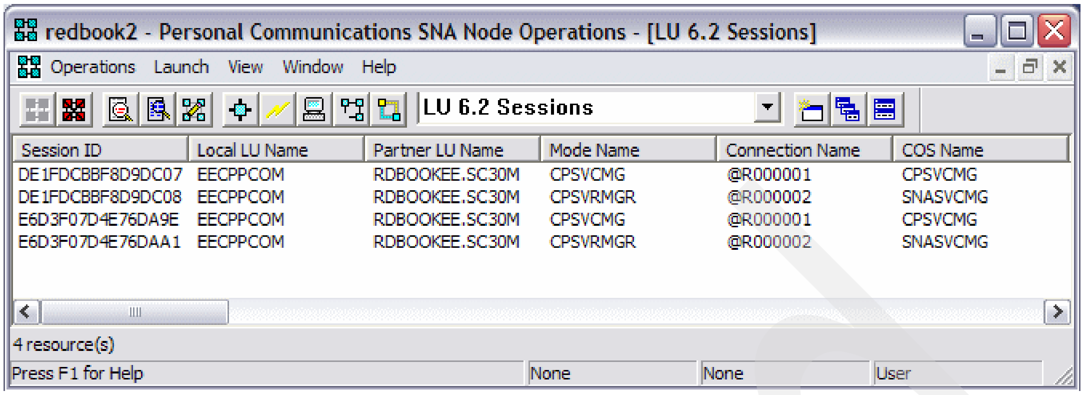
First, check the current status of each resource such as LU 6.2 sessions and RTP connections.


redbook2 - Personal Communications SNA Node Operations - [Connections]
Operations Launch View Window Help
Connections
2 resource(s)
Press F1 for Help
None
None
User

Name	State	Sub-state	Device Name	Adjacent CP Name	DLC Name
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	IBMEEDLC

Figure 5-38 connection status before the HPR path switch

Figure 5-38 shows that two connections are `Active`. EELINK01 is for SC30M, and EELINK02 for SC31M.


redbook2 - Personal Communications SNA Node Operations - [LU 6.2 Sessions]
Operations Launch View Window Help
LU 6.2 Sessions

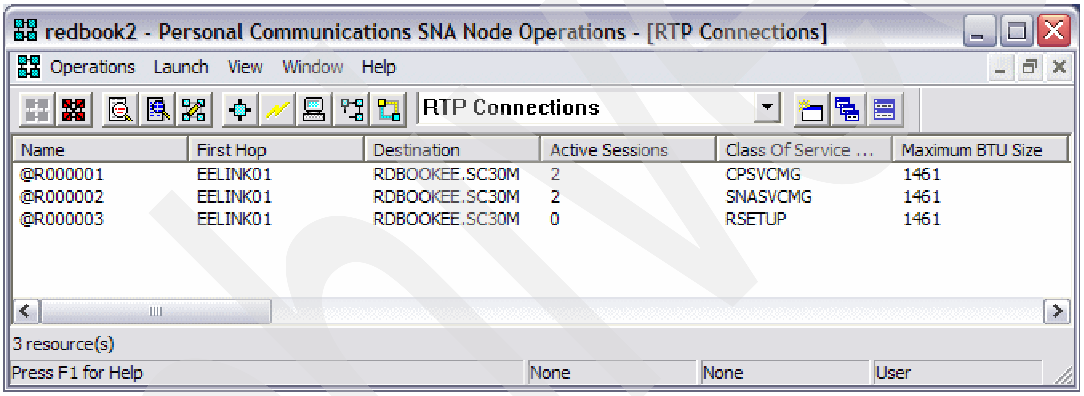
Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FDCBBF8D9DC07	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG
DE1FDCBBF8D9DC08	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
E6D3F07D4E76DA9E	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG
E6D3F07D4E76DAA1	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG

4 resource(s)

Press F1 for Help | None | None | User

Figure 5-39 LU 6.2 Session status before the HPR path switch

Figure 5-39 shows four LU 6.2 sessions. Two sessions which use CPSVCMG as their mode are CP-CP sessions, and other sessions are CPSVRMGR pipe for DLUR PU.


redbook2 - Personal Communications SNA Node Operations - [RTP Connections]
Operations Launch View Window Help
RTP Connections

Name	First Hop	Destination	Active Sessions	Class Of Service ...	Maximum BTU Size
@R000001	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG	1461
@R000002	EELINK01	RDBOOKEE.SC30M	2	SNASVCMG	1461
@R000003	EELINK01	RDBOOKEE.SC30M	0	RSETUP	1461

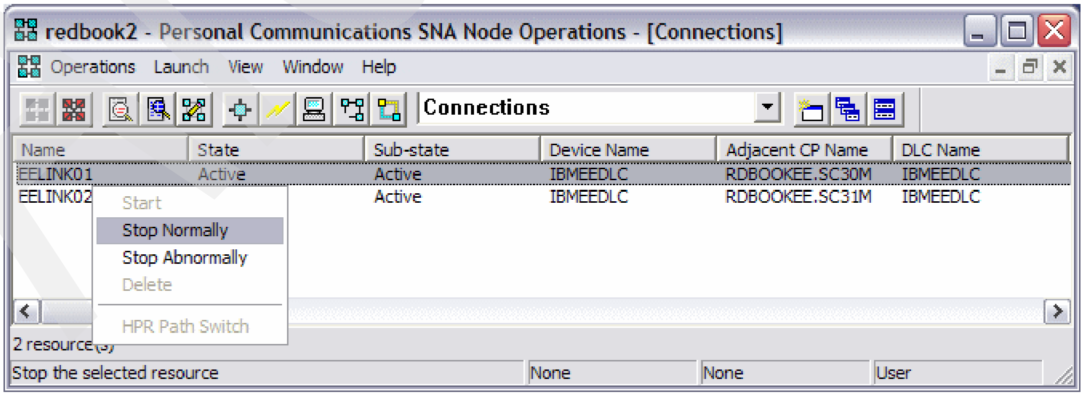
3 resource(s)

Press F1 for Help | None | None | User

Figure 5-40 RTP connection status before the HPR path switch

Figure 5-40shows RTP connections. @R000001 is for CP-CP sessions and @R000002 is for CPSVRMGR pipe.

Now, we shut down the EELINK01 link station.


redbook2 - Personal Communications SNA Node Operations - [Connections]
Operations Launch View Window Help
Connections
Start
Stop Normally
Stop Abnormally
Delete
HPR Path Switch

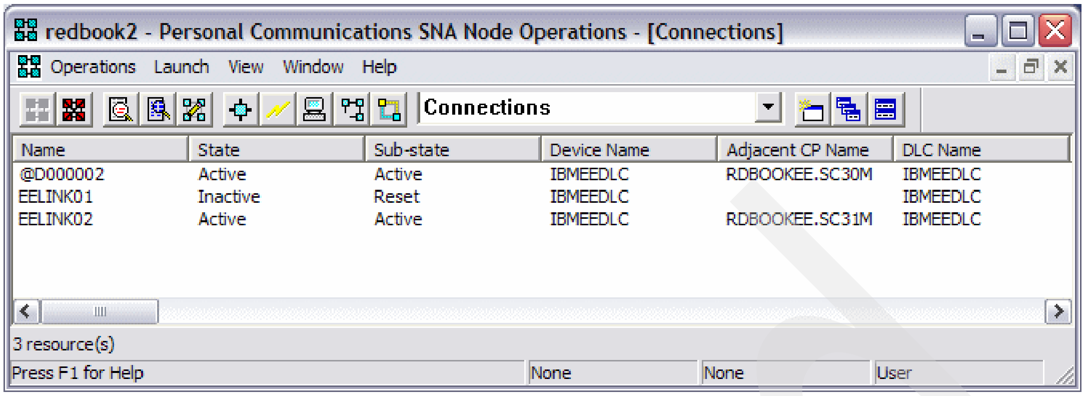
Name	State	Sub-state	Device Name	Adjacent CP Name	DLC Name
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK02		Active	IBMEEDLC	RDBOOKEE.SC31M	IBMEEDLC

2 resource(s)

Stop the selected resource | None | None | User

Figure 5-41 Stopping the EELINK01 connection

In Figure 5-41, we shut down the EELINK01 connection.


redbook2 - Personal Communications SNA Node Operations - [Connections]
Operations Launch View Window Help
Connections

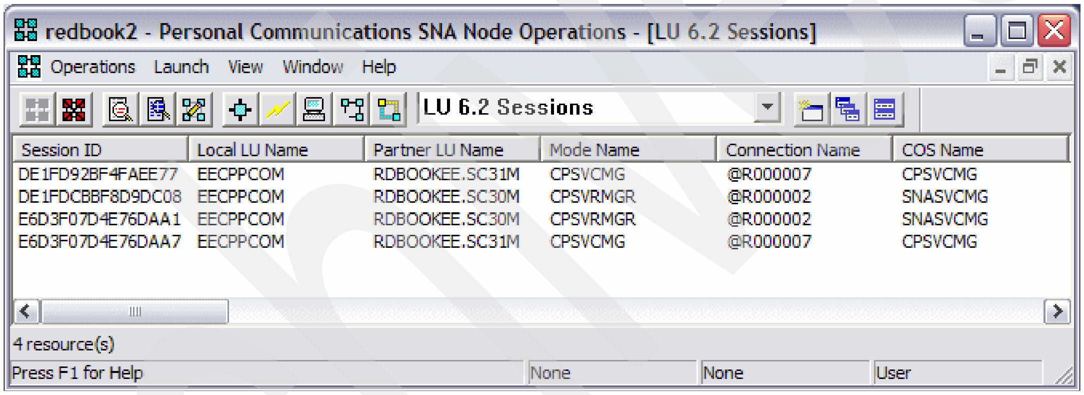
Name	State	Sub-state	Device Name	Adjacent CP Name	DLC Name
@D000002	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK01	Inactive	Reset	IBMEEDLC		IBMEEDLC
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	IBMEEDLC

3 resource(s)

Press F1 for Help | None | None | User

Figure 5-42 EELINK01 connection is now shutdown

Figure 5-42 shows that EELINK01 connection is in an `Inactive` state, and a new dynamic connection @D000002 is created for serving DLUR-DLUS sessions. Now check the status of LU 6.2 connections after shutdown of EELINK01 connection.


redbook2 - Personal Communications SNA Node Operations - [LU 6.2 Sessions]
Operations Launch View Window Help
LU 6.2 Sessions

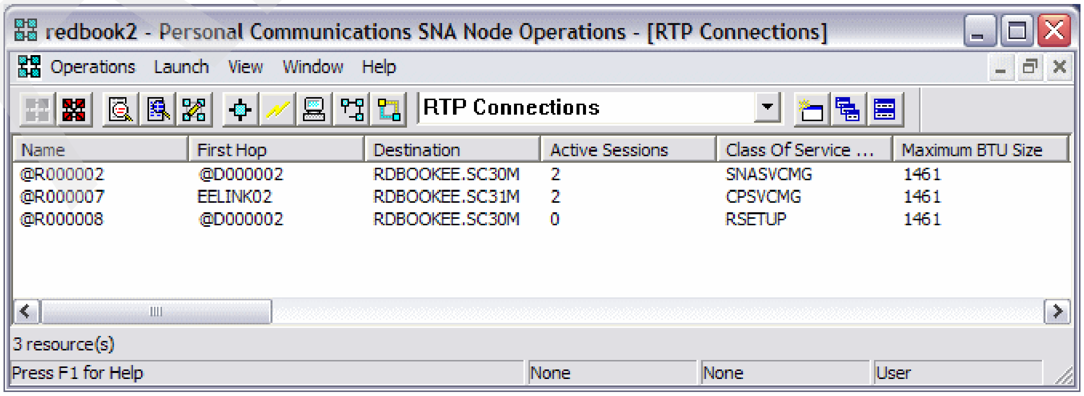
Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FD92BF4FAEE77	EECPPCOM	RDBOOKEE.SC31M	CPSVCMG	@R000007	CPSVCMG
DE1FDCBBF8D9DC08	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
E6D3F07D4E76DAA1	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
E6D3F07D4E76DAA7	EECPPCOM	RDBOOKEE.SC31M	CPSVCMG	@R000007	CPSVCMG

4 resource(s)

Press F1 for Help | None | None | User

Figure 5-43 LU 6.2 session status after shutdown of EELINK01

In Figure 5-43, you will see that CP-CP sessions are established with the SC31M (backup network node server) through new RTP connection, @R000007. But DLUR-DLUS sessions still use the @R000002 for their RTP connection. CP-CP sessions are moved to the SC31M even though there is a dynamic connection to the SC30M. This is because the CP-CP sessions cannot use the dynamic connection through the connection network. Moreover, the CP-CP sessions can be established with only the adjacent node. That means, the CP-CP sessions cannot use the path which has more than two hops.


redbook2 - Personal Communications SNA Node Operations - [RTP Connections]
Operations Launch View Window Help
RTP Connections

Name	First Hop	Destination	Active Sessions	Class Of Service ...	Maximum BTU Size
@R000002	@D000002	RDBOOKEE.SC30M	2	SNASVCMG	1461
@R000007	EELINK02	RDBOOKEE.SC31M	2	CPSVCMG	1461
@R000008	@D000002	RDBOOKEE.SC30M	0	RSETUP	1461

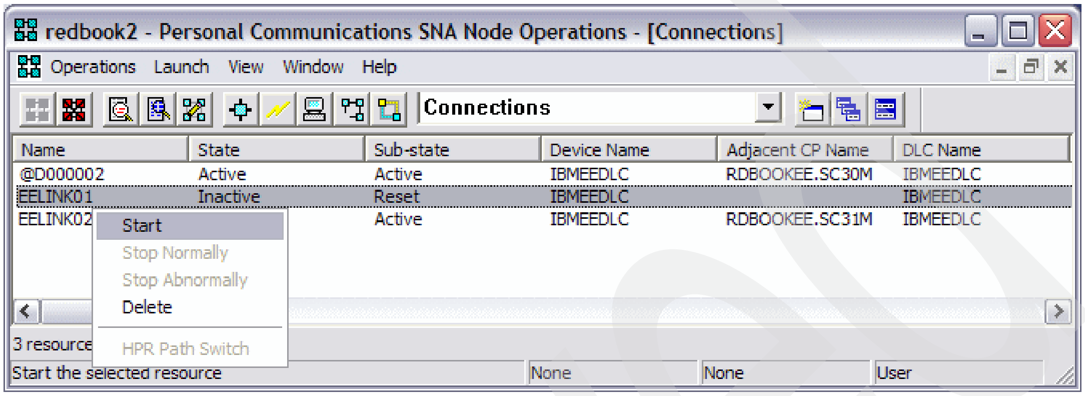
3 resource(s)

Press F1 for Help | None | None | User

Figure 5-44 Status of RTP connection after shutdown of EELINK01

In Figure 5-44 you can find that @R000002 now uses the @D000002 as its next hop, but the sessions using this RTP connection are intact. This is the feature of the HPR Path Switch. All sessions using an RTP connection are intact, but the path used by the RTP connection can be changed dynamically.

Now, we re-start the EELINK01 connection (see Figure 5-45 and Figure 5-46).

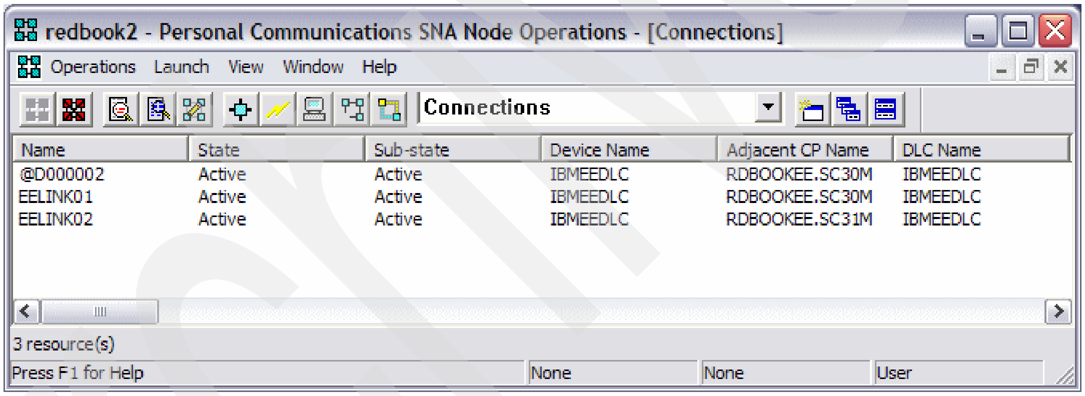

redbook2 - Personal Communications SNA Node Operations - [Connections]
Operations Launch View Window Help
Connections
Start
Stop Normally
Stop Abnormally
Delete
HPR Path Switch

Name	State	Sub-state	Device Name	Adjacent CP Name	DLC Name
@D000002	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK01	Inactive	Reset	IBMEEDLC		IBMEEDLC
EELINK02		Active	IBMEEDLC	RDBOOKEE.SC31M	IBMEEDLC

3 resource

Start the selected resource None None User

Figure 5-45 Re-start of EELINK01 connection


redbook2 - Personal Communications SNA Node Operations - [Connections]
Operations Launch View Window Help
Connections

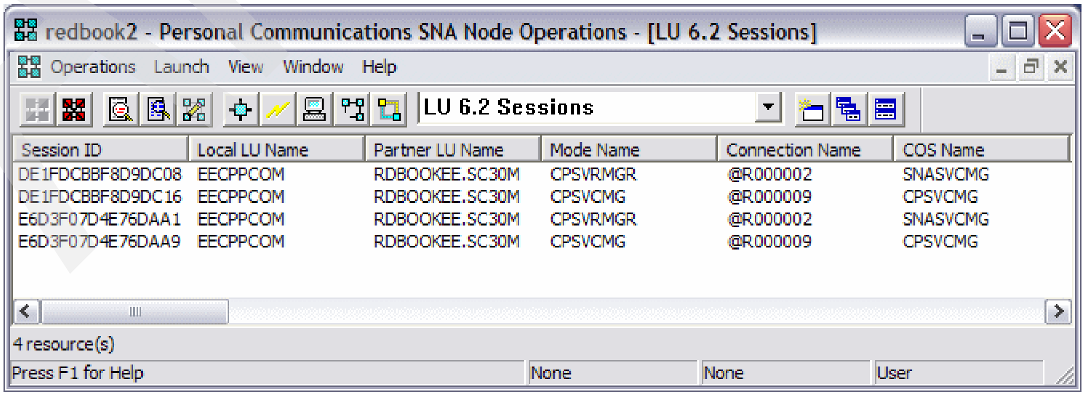
Name	State	Sub-state	Device Name	Adjacent CP Name	DLC Name
@D000002	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	IBMEEDLC
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	IBMEEDLC

3 resource(s)

Press F1 for Help None None User

Figure 5-46 EELINK01 connection is restored

After the EELINK01 connection is restored, check the status of LU 6.2 sessions and RTP connections (see Figure 5-47).


redbook2 - Personal Communications SNA Node Operations - [LU 6.2 Sessions]
Operations Launch View Window Help
LU 6.2 Sessions

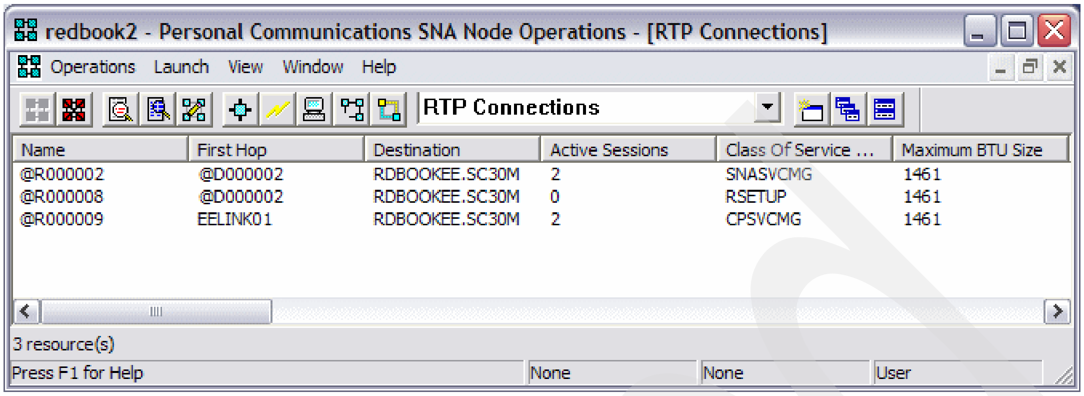
Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FDCBBF8D9DC08	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
DE1FDCBBF8D9DC16	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000009	CPSVCMG
E6D3F07D4E76DAA1	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
E6D3F07D4E76DAA9	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000009	CPSVCMG

4 resource(s)

Press F1 for Help None None User

Figure 5-47 Status of LU 6.2 sessions after restoring EELINK01 connection

Figure 5-47 shows two CP-CP sessions are established with SC30M now, because it is the preferred network node server. DLUR-DLUS sessions are not changed.

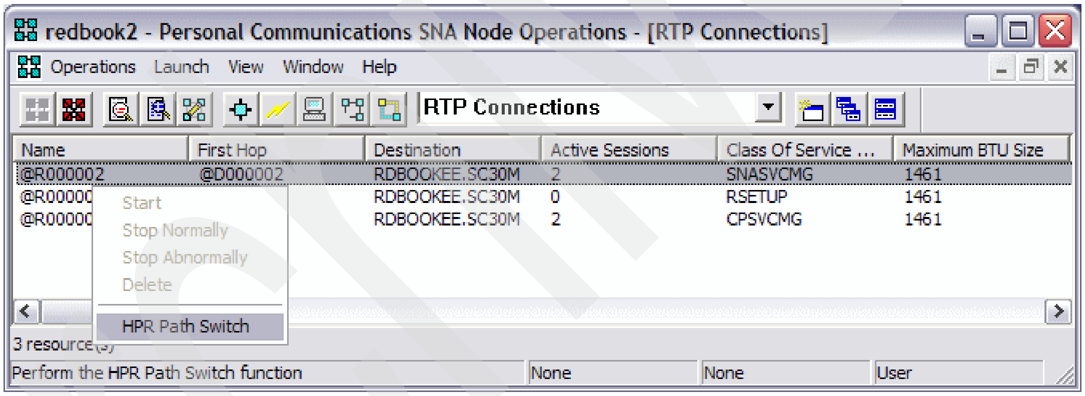

redbook2 - Personal Communications SNA Node Operations - [RTP Connections]
Operations Launch View Window Help
RTP Connections

Name	First Hop	Destination	Active Sessions	Class Of Service ...	Maximum BTU Size
@R000002	@D000002	RDBOOKEE.SC30M	2	SNASVCMG	1461
@R000008	@D000002	RDBOOKEE.SC30M	0	RSETUP	1461
@R000009	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG	1461

3 resource(s)
Press F1 for Help
None
None
User

Figure 5-48   Status of RTP connections after restoring EELINK01 connection

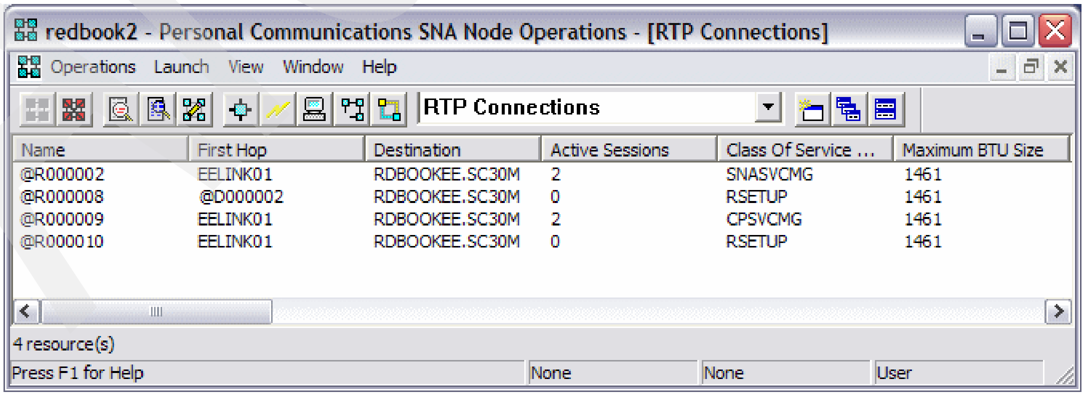
Figure 5-48 shows the status of RTP connection after restoring. Now, CP-CP sessions are restored to SC30M and DLUS-DLUR sessions are now using the @D000002 as their next hop. We can manually start the HPR Path Switch process to a more efficient path. We cannot specify the next hop which will be used. The efficient path will be found dynamically.


redbook2 - Personal Communications SNA Node Operations - [RTP Connections]
Operations Launch View Window Help
RTP Connections

Name	First Hop	Destination	Active Sessions	Class Of Service ...	Maximum BTU Size
@R000002	@D000002	RDBOOKEE.SC30M	2	SNASVCMG	1461
@R00000		RDBOOKEE.SC30M	0	RSETUP	1461
@R00000		RDBOOKEE.SC30M	2	CPSVCMG	1461

Start
Stop Normally
Stop Abnormally
Delete
HPR Path Switch
3 resource
Perform the HPR Path Switch function
None
None
User

Figure 5-49   HPR path switch


redbook2 - Personal Communications SNA Node Operations - [RTP Connections]
Operations Launch View Window Help
RTP Connections

Name	First Hop	Destination	Active Sessions	Class Of Service ...	Maximum BTU Size
@R000002	EELINK01	RDBOOKEE.SC30M	2	SNASVCMG	1461
@R000008	@D000002	RDBOOKEE.SC30M	0	RSETUP	1461
@R000009	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG	1461
@R000010	EELINK01	RDBOOKEE.SC30M	0	RSETUP	1461

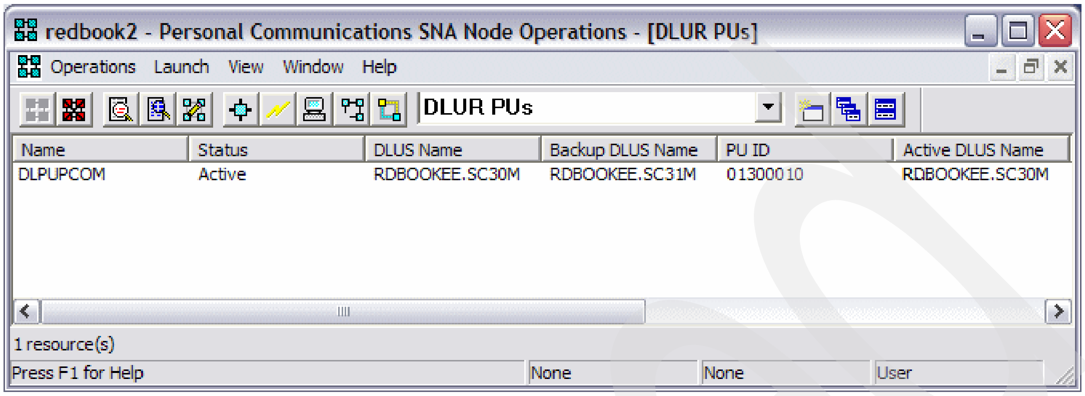
4 resource(s)
Press F1 for Help
None
None
User

Figure 5-50   After HPR path switch

In Figure 5-50, the path for @R000002 is changed. The first hop of the RTP connection is changed to EELINK01 now.

Verifying the backup DLUS

If the DLUS has a problem, the DLUS-DLUR sessions (CPSVRMGR) pipe might be broken. If you define the backup DLUS, then the backup DLUS can takeover the DLUS function in that case. We simulate this situation by deactivating the PU on the VTAM side.


redbook2 - Personal Communications SNA Node Operations - [DLUR PUs]
Operations
Launch
View
Window
Help
DLUR PUs

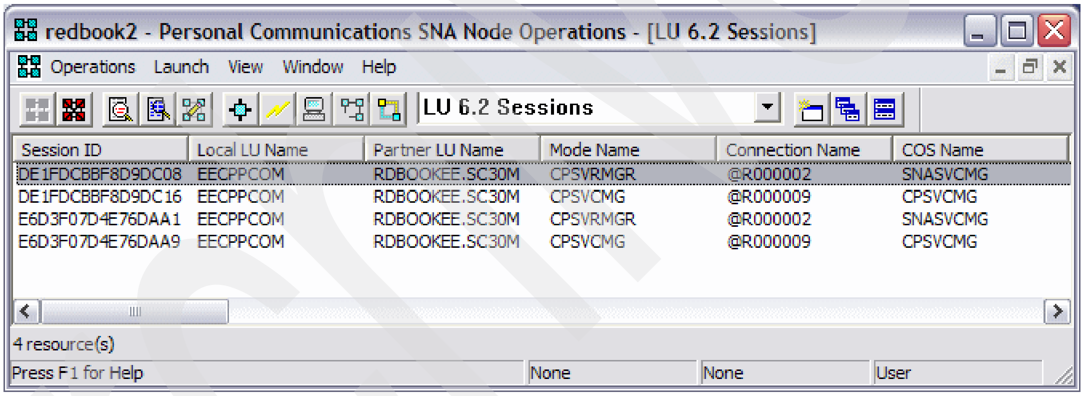
Name	Status	DLUS Name	Backup DLUS Name	PU ID	Active DLUS Name
DLPUPCOM	Active	RDBOOKEE.SC30M	RDBOOKEE.SC31M	01300010	RDBOOKEE.SC30M

1 resource(s)

Press F1 for Help | None | None | User

Figure 5-51 Status of DLUR PU before deactivation of PU

Figure 5-51 and Figure 5-52 show the current state.


redbook2 - Personal Communications SNA Node Operations - [LU 6.2 Sessions]
Operations
Launch
View
Window
Help
LU 6.2 Sessions

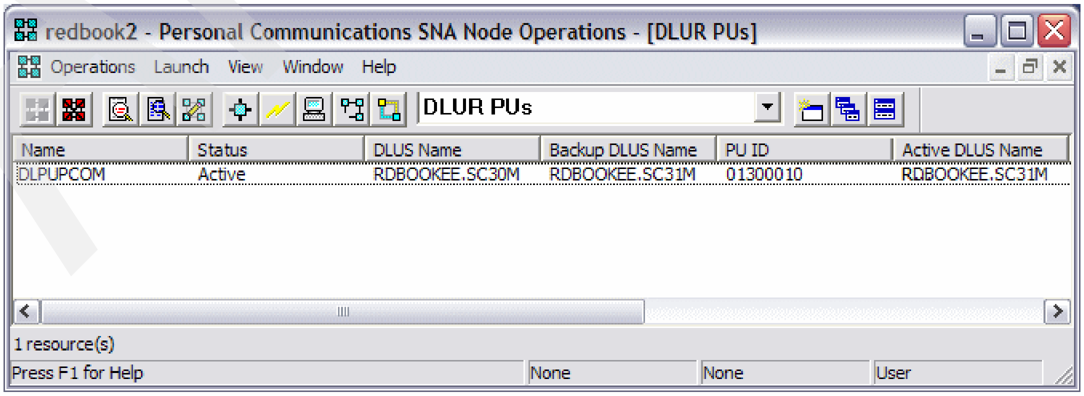
Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FDCBBF8D9DC08	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
DE1FDCBBF8D9DC16	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000009	CPSVCMG
E6D3F07D4E76DAA1	EECPPCOM	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
E6D3F07D4E76DAA9	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000009	CPSVCMG

4 resource(s)

Press F1 for Help | None | None | User

Figure 5-52 Status of LU6.2 sessions before deactivation of PU

Before deactivating the PU, the active DLUS is the SC30M and two DLUS-DLUR sessions are established with SC30M. Now, we deactivate the PU in VTAM.


redbook2 - Personal Communications SNA Node Operations - [DLUR PUs]
Operations
Launch
View
Window
Help
DLUR PUs

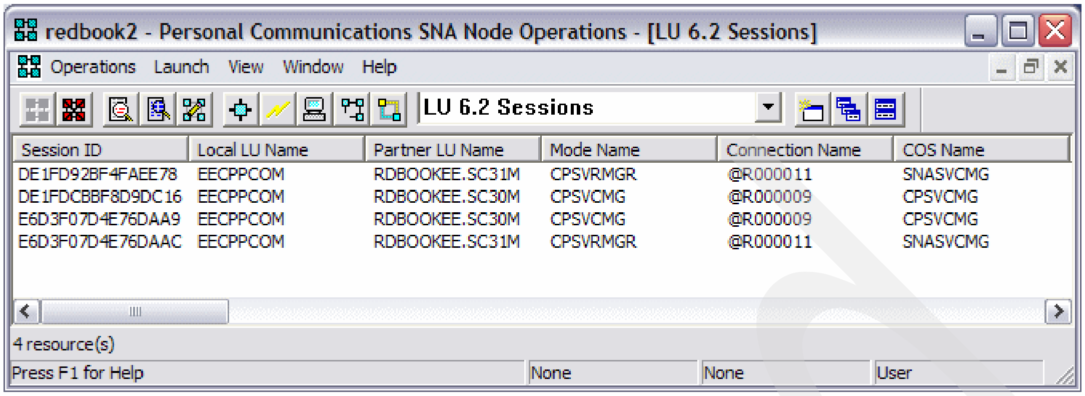
Name	Status	DLUS Name	Backup DLUS Name	PU ID	Active DLUS Name
DLPUPCOM	Active	RDBOOKEE.SC30M	RDBOOKEE.SC31M	01300010	RDBOOKEE.SC31M

1 resource(s)

Press F1 for Help | None | None | User

Figure 5-53 Status of DLUR PU after deactivation of PU

Figure 5-53 and Figure 5-54 show the state after deactivation.


redbook2 - Personal Communications SNA Node Operations - [LU 6.2 Sessions]
Operations Launch View Window Help
LU 6.2 Sessions

Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FD92BF4FAEE78	EECPPCOM	RDBOOKEE.SC31M	CPSVRMGR	@R000011	SNASVCMG
DE1FDCBBF8D9DC16	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000009	CPSVCMG
E6D3F07D4E76DAA9	EECPPCOM	RDBOOKEE.SC30M	CPSVCMG	@R000009	CPSVCMG
E6D3F07D4E76DAAC	EECPPCOM	RDBOOKEE.SC31M	CPSVRMGR	@R000011	SNASVCMG

4 resource(s)
Press F1 for Help
None
None
User

Figure 5-54 Status of LU 6.2 sessions after deactivation of PU

In Figure 5-53, the active DLUS is changed to SC31M because it cannot contact the DLUS in the SC30M. Though the primary DLUS (SC30M) becomes available in this situation, the active DLUS will not be restored to the SC30M. You should manually restore it to the SC30M by issuing the following command in the SC31M:

```
/V NET,INACT,ID=DLPUPCOM,GIVEBACK
```

5.4.3 Stopping resources

All resources of PCOMM can be stopped by right-clicking **stop**. We recommend you stop the LU-LU sessions first using the batch file, and stop the node by selecting **Operations** → **Stop Node** in the SNA node operations window as shown in Figure 5-55.

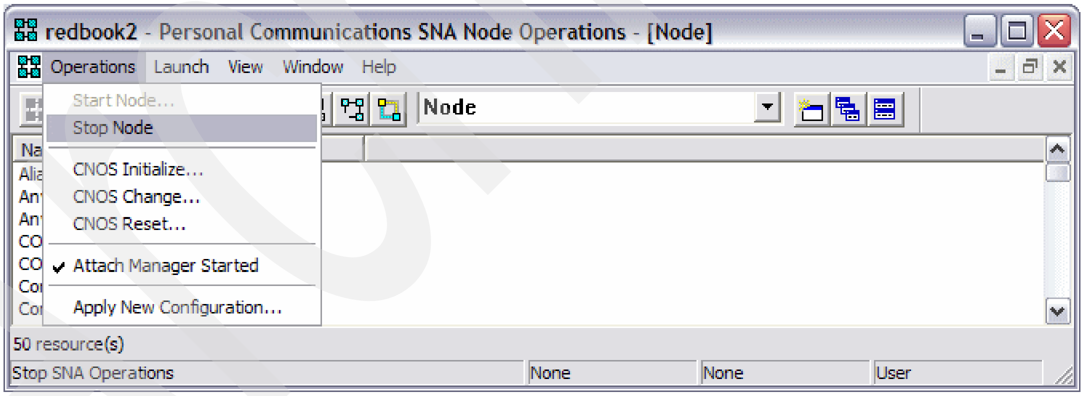


Figure 5-55 Stopping node

5.4.4 Backing up your configuration

You have to save your configuration in the *.acg file in the Program Files\IBM\Personal Communications\private directory of your system disk (by default). Using a couple of acg files, you can easily switch from one configuration to another. Always make a good backup of these acg files.

5.5 Diagnosing EE in Personal Communications (PCOMM)

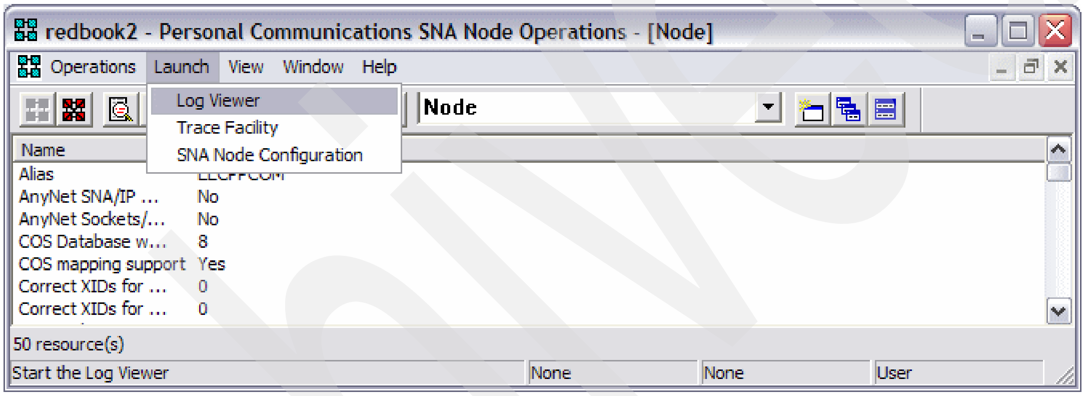
IBM Communications Server for Windows provides logging and tracing facility. In this section, we briefly describe how you can get log files and trace information.

The topics covered in this section are:

- “Reviewing the log files”
- “Tracing Personal Communications (PCOMM) for EE”
- “Information bundler”
- “Sample batch file for EE tracing”

5.5.1 Reviewing the log files

PCOMM provides the Log viewer utility. You can start the log viewer with the SNA node operations window as shown in Figure 5-56.



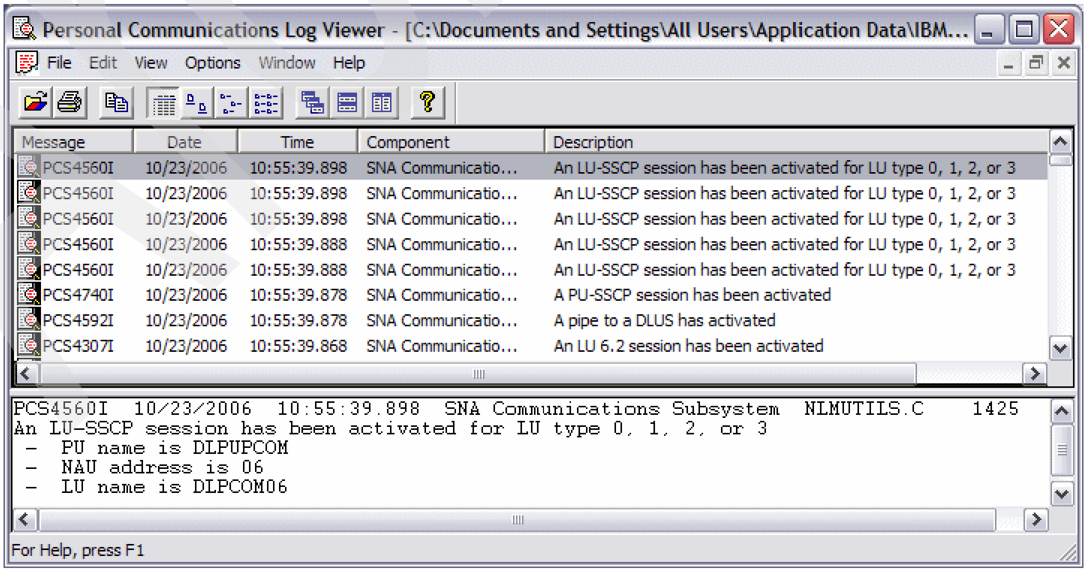
Name	
Alias	LLCPFCOM
AnyNet SNA/IP ...	No
AnyNet Sockets/...	No
COS Database w...	8
COS mapping support	Yes
Correct XIDs for ...	0
Correct XIDs for ...	0

50 resource(s)

Start the Log Viewer | None | None | User

Figure 5-56 Starting the log viewer

With this log viewer, you can easily detect the various unusual events on the PCOMM. This is the first step to debug your PCOMM problems including the EE function.


Personal Communications Log Viewer - [C:\Documents and Settings\All Users\Application Data\IBM...
File
Edit
View
Options
Window
Help
For Help, press F1

Message	Date	Time	Component	Description
PCS4560I	10/23/2006	10:55:39.898	SNA Communicatio...	An LU-SSCP session has been activated for LU type 0, 1, 2, or 3
PCS4560I	10/23/2006	10:55:39.898	SNA Communicatio...	An LU-SSCP session has been activated for LU type 0, 1, 2, or 3
PCS4560I	10/23/2006	10:55:39.898	SNA Communicatio...	An LU-SSCP session has been activated for LU type 0, 1, 2, or 3
PCS4560I	10/23/2006	10:55:39.888	SNA Communicatio...	An LU-SSCP session has been activated for LU type 0, 1, 2, or 3
PCS4560I	10/23/2006	10:55:39.888	SNA Communicatio...	An LU-SSCP session has been activated for LU type 0, 1, 2, or 3
PCS4740I	10/23/2006	10:55:39.878	SNA Communicatio...	A PU-SSCP session has been activated
PCS4592I	10/23/2006	10:55:39.878	SNA Communicatio...	A pipe to a DLUS has activated
PCS4307I	10/23/2006	10:55:39.868	SNA Communicatio...	An LU 6.2 session has been activated

```
PCS4560I  10/23/2006  10:55:39.898  SNA Communications Subsystem  NLMUTILS.C    1425
An LU-SSCP session has been activated for LU type 0, 1, 2, or 3
 -  PU name is DLPUPCOM
 -  NAU address is 06
 -  LU name is DLPCOM06
```

Figure 5-57 Log viewer

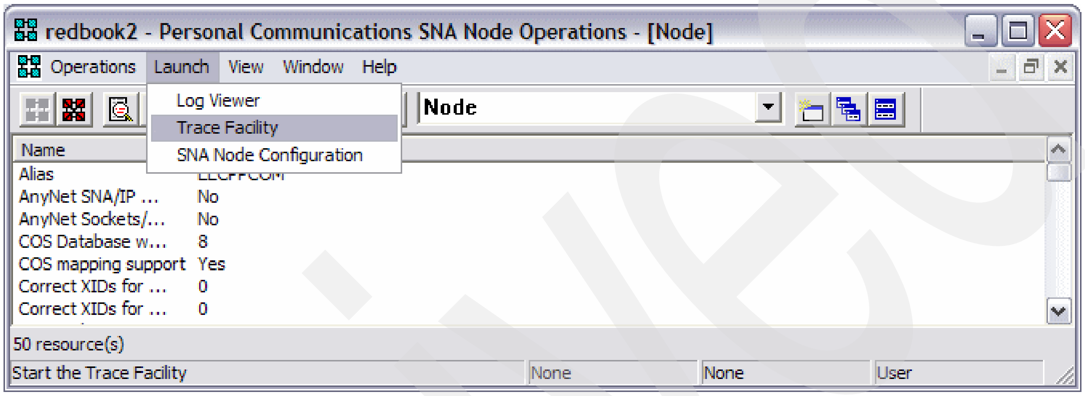
5.5.2 Tracing Personal Communications (PCOMM) for EE

We explain the trace facility of IBM Communications Server for Windows, and also explain using the IP packet trace briefly in this section. The topics covered in this section are:

- "Trace facility of PCOMM"
- "Getting the IP packet trace"

Trace facility of PCOMM

The IBM Communications Server for Windows provides the trace facility on many events. You can start the trace facility using the SNA node operations utility as in Figure 5-58.



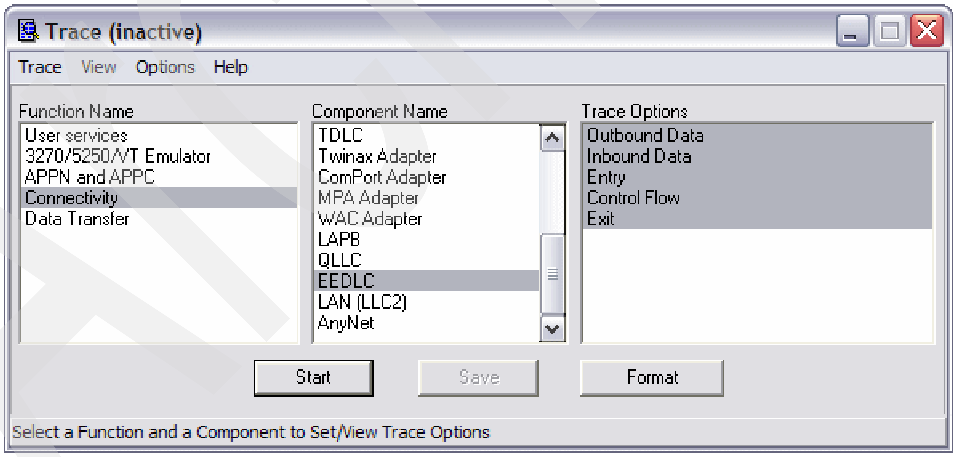
Name	
Alias	LLCPPCOM
AnyNet SNA/IP ...	No
AnyNet Sockets/...	No
COS Database w...	8
COS mapping support	Yes
Correct XIDs for ...	0
Correct XIDs for ...	0

50 resource(s)

Start the Trace Facility | None | None | User

Figure 5-58 Starting trace facility

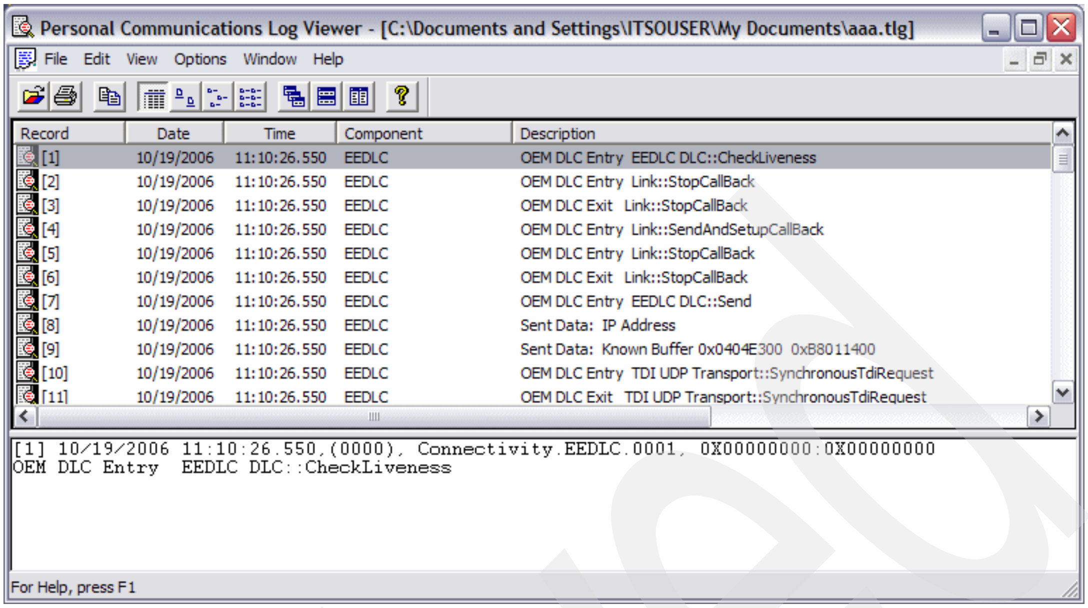
You can setup various types of traces with the trace facility. For more information about the trace facility of PCOMM, refer to *Personal Communications for Windows Administrator's Guide and Reference*, SC31-8840. In Figure 5-59, we set up the trace of all the activities of EEDLC.



Select a Function and a Component to Set/View Trace Options

Figure 5-59 Tracing all activities of EEDLC

After getting the traces, you can easily format the trace files to the log file formats with the trace facility, and you can also view them with the log viewer (see Figure 5-60).



Personal Communications Log Viewer - [C:\Documents and Settings\ITSOUSER\My Documents\aaa.tlg]
File
Edit
View
Options
Window
Help

Record	Date	Time	Component	Description
[1]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Entry EEDLC DLC::CheckLiveness
[2]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Entry Link::StopCallBack
[3]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Exit Link::StopCallBack
[4]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Entry Link::SendAndSetupCallBack
[5]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Entry Link::StopCallBack
[6]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Exit Link::StopCallBack
[7]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Entry EEDLC DLC::Send
[8]	10/19/2006	11:10:26.550	EEDLC	Sent Data: IP Address
[9]	10/19/2006	11:10:26.550	EEDLC	Sent Data: Known Buffer 0x0404E300 0xB8011400
[10]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Entry TDI UDP Transport::SynchronousTdiRequest
[11]	10/19/2006	11:10:26.550	EEDLC	OEM DLC Exit TDI UDP Transport::SynchronousTdiRequest

```
[1] 10/19/2006 11:10:26.550,(0000), Connectivity.EEDLC.0001, 0X00000000:0X00000000
OEM DLC Entry  EEDLC DLC::CheckLiveness
```

Figure 5-60 Reviewing the traces

We provide a simple batch program to collect the traces of EE activities, and to format those trace files into the log viewer format.

Getting the IP packet trace

EE uses UDP for its transport. Sometimes, the problem may be related to UDP/IP and not the SNA/APPN. In this case packet trace can be used to solve the problem.

There exist a number of commercial and open source packet capture utilities and protocol analyzers. If you have a special packet trace utility and packet analyzer, then just use that tool for debugging EE problems. But if you do not have any packet trace utility or packet analyzer, try to use the Wireshark program (formerly Ethereal) for this purpose. Wireshark is an open source software which provides a good set of protocol analyzer functions including packet capturing with winpcap library. You can easily capture all the frames which pass your Ethernet adapter, and review all the frames captured with an easy GUI of Ethereal. For more information and downloading the application, visit the following Web site:

`http://www.wireshark.org`

5.5.3 Information bundler

Information bundler utility gathers system files, trace and log files, and registry information and creates a self-extracting .EXE file. This utility should be executed immediately after the trace is complete to ensure that the correct information is gathered. To use the information bundler, do the following:

- Select **Start** → **IBM Personal Communications** → **Administrative and PD Aids** → **Information Bundler**.
- If you are in an active session with PCOMM, select **Launch** → **Information Bundler** from the **Actions** menu.

A .EXE file containing system and PCOMM information is created in the Personal Communications system-class application data directory. By default this file is called X12345.exe.

Note: PCOMM system-class application data directory is hidden. You should enable "show hidden files and folders" in your Explorer setting.

5.5.4 Sample batch file gathering EE trace data

We provide a simple batch file to gather EE traces in PCOMM. You have to create two batch files as in Example 5-4, one for starting the traces, and one for stopping the traces.

Example 5-4 Sample batch file for EE tracing

```
** batch file for turning on trace

cstrace reset
cstrace  start /f 4 /c 22  /o 10 20 /r
cstrace  apply /f 4 /c 33  /o 1
cstrace  apply /f 3 /c 7  /o 4
cstrace status

** batch file for turning off trace

cstrace stop
cstrace save rdbookee.trc
cstrace format rdbookee.trc
```

After the execution of the first batch file, re-generate the situation which you want to trace. After re-generation of the problem, execute the second file, turning off the batch file. You can find a redbookee.tlg log file in the same directory as the batch files, this tlg file can be read with a log viewer of PCOMM.

6

Enabling EE in CS for AIX

In this chapter, we describe the general configuration steps for Enterprise Extender (EE) on AIX and provide useful information about Communications Server for AIX.

The topics covered in this chapter are as follows:

- "Overview of Communications Server for AIX"
- "Implementation considerations"
- "Configuring EE with Communications Server for AIX"
- "Verifying and managing EE with CS for AIX"
- "Diagnosing EE with Communications Server for AIX"

6.1 Overview of Communications Server for AIX

Communications Server for AIX (CS/AIX) fully implements Advanced Peer-to-Peer Networking (APPN), High Performance Routing (HPR), and DLUR Dependent LU Requestor (DLUR) functions. You can use various connectivity methods such as LLC2 and MPC+ in addition to EE. With EE, you can use Advanced Peer-to-Peer Communication (APPC) applications on top of LU6.2, and you can also use applications on dependent LUs such as terminals, printers, and LU type 0 applications.

Although AIX has IPv6 support, CS/AIX currently only supports IPv4.

We do not provide any information about installing CS/AIX in this book. For information about installing CS/AIX, refer to *IBM Communications Server for AIX Quick Beginnings*, GC31-8583.

Throughout this chapter, we assume that you have a basic knowledge of APPN. If you are not familiar with APPN, refer to 1.2.1, “Positioning APPN and HPR” on page 4.

6.2 Implementation considerations

The topics covered in this section are:

- “High availability considerations”
- “UDP consideration”
- “Scenario used in this chapter”

6.2.1 High availability considerations

With HPR, you can create a very flexible and intelligent SNA network. For more information about HPR, refer to *Subarea to APPN Migration: HPR and DLUR Implementation*, SG24-5204.

You must check the following in order to create a good APPN/HPR network with EE in CS/AIX:

- “The number of Ethernet adapters and switches”
- “The number of connections to z/OS Logical Partitions (LPARs)”

The number of Ethernet adapters and switches

If you require high network availability, then your AIX server must have multiple Ethernet adapters installed. Though you can use VIPA interfaces to achieve this, we recommend using the EtherChannel feature in AIX.

EtherChannel with EE

EtherChannel or port aggregation was first developed as a function of a switch. By aggregating two or more ports as one logical port, users can get more bandwidth, and higher availability. AIX accepts these concepts as a base operating system function. For AIX, there are several different operation modes for EtherChannel:

- Standard
- Round_robin
- EtherChannel backup (netif_backup)
- IEEE802.3ad Link Aggregation

From the above modes, we use the EtherChannel backup (netif_backup) mode. Also remember, load balancing mode is not very useful in an EE environment, it could generate out-of-order packets, which might degrade the performance of EE.

For more information about the AIX EtherChannel feature, refer to *Managing communications and networks*, SC23-5203.

With EtherChannel backup mode, you can have one active adapter and one backup adapter. Moreover, you can connect each adapter to a different switch. If the active adapter fails, the backup adapter becomes the active adapter without delay. These two adapters consist of one IP interface, so you can handle this interface as though it were only one adapter.

The number of connections to z/OS Logical Partitions (LPARs)

Usually an end node (EN) has a preferred network node server (NNS) and a backup network node server. For a preferred NNS and a backup NNS, you must define a connection for each. For other nodes, you must define additional connections to each node or you can define a Connection Network to support dynamic APPN links.

We strongly recommend using a Connection Network, because it provides a more simple way to implement nodes (see “Connection Network” on page 67).

6.2.2 UDP consideration

EE uses User Datagram Protocol (UDP) for its transport. It relies on UDP checksum for data integrity for High Performance Routing (HPR). But in most cases, the UDP checksum feature is optional. AIX can turn this UDP checksum feature on or off, and for performance reasons you can turn off the UDP checksum feature. But in an EE environment, always turn on the UDP checksum feature for proper operation of HPR. By default, the *udpcksum no* option is set to 1 (UDP checksum feature is used).

6.2.3 Scenario used in this chapter

Figure 6-1 on page 196 illustrates the scenario used in this book. SC30M, SC31M, and SC32M are z/OS VTAMs and EECPAIX is a CS/AIX.

We configure the CS/AIX as an End Node (EN). The CP name will be EECPAIX. The preferred network node server of this EN is SC30M, and the backup network node server is SC31M. We define static link stations for these two network nodes, because a static link station should exist for the network node server.

For connectivity to other nodes such as SC32M, we use dynamic link stations through connection network instead of using static link stations. In our scenario, the name of the connection network is RDBOOKEE.VRNLOCAL.

In general, a CS/AIX EN can use a Local VRN or a Global VRN, but a CS/AIX NN can only use Local VRNs.

Throughout this chapter we use host names instead of IP addresses.

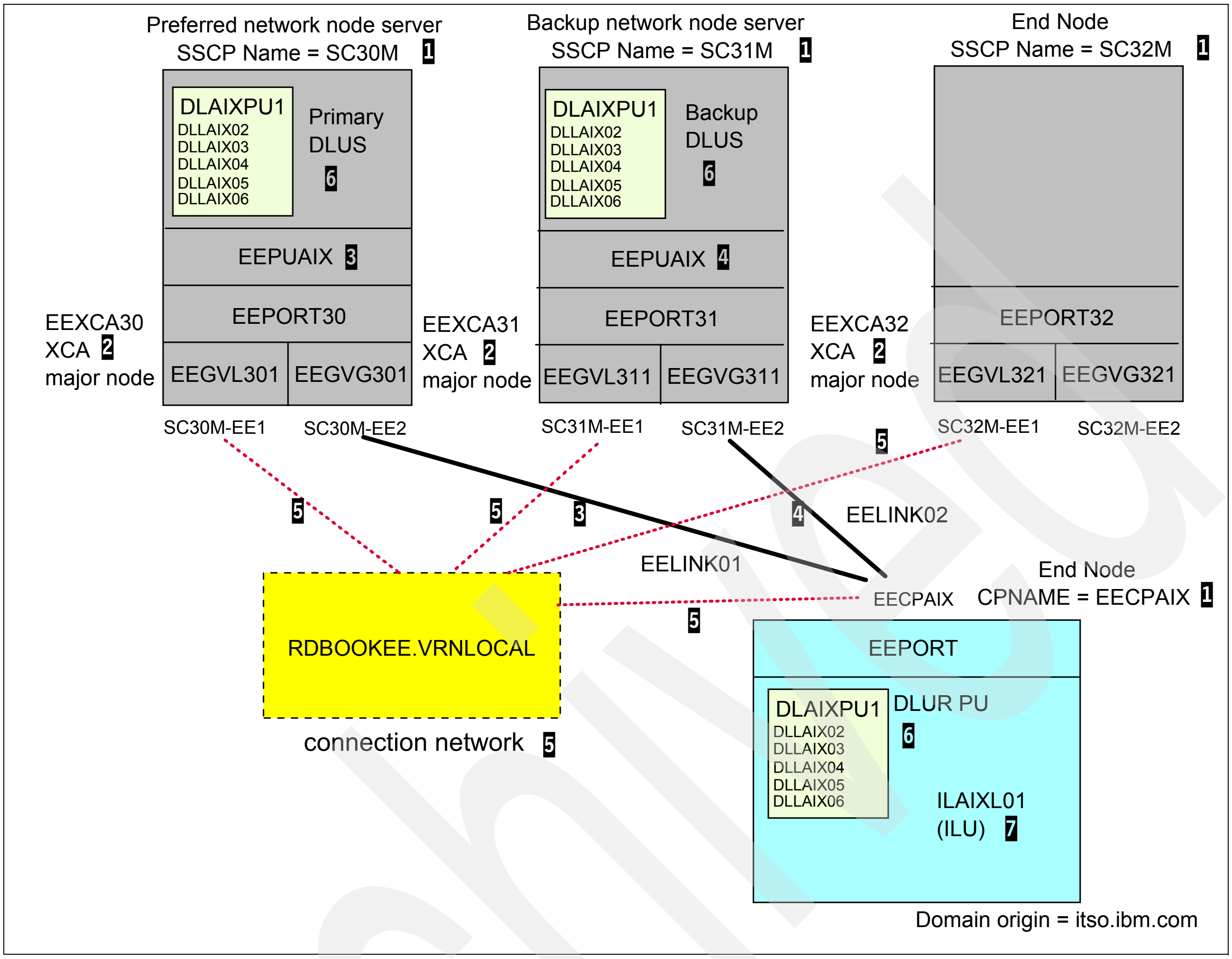


Figure 6-1 Scenario used in this chapter

Here are explanatory notes to Figure 6-1:

- 1: CP name of each node. For z/OS VTAM, it is an SSCP name. The preferred network node server for EECPAIX is SC30M, and the backup network node server is SC31M.
- 2: For enabling EE in z/OS VTAM, an XCA major node is required. An XCA major node consists of port and group definitions. We define two groups in an XCA major node in each system.
- 3: Static link station to SC30M (preferred network node server). We use EEGVG3x1 group for this connectivity; a PU definition for this link station should exist because DYNPU=NO for this group. We define EEPUAIX for this purpose.
- 4: Static link station to SC31M (backup network node server). During an outage of the preferred network node server, this node will be used as a network node server. We use EEGVG3x1 group for this connectivity; a PU definition for this link station must exist because DYNPU=NO for this group. We define EEPUAIX for this purpose.
- 5: In each node, we define a connection network, RDBOOKEE.VRNLOCAL. Because every node participates in this same connection network, dynamic link stations will be generated without any definition if needed. We use EEGVL3x1 group for this connection network in z/OS, and EEPORT will be used in CS/AIX.

- 6: For dependent LUs, the DLUS/DLUR feature should be configured. In our scenario, SC30M will be a primary DLUS, and SC31M will be a backup DLUS. You must define a type 2.0 PU and all dependent LUs in each VTAM for accepting the DLUR's request. For CS/AIXCS/AIX, a definition of DLUR PU and dependent LUs is required.
- 7: For independent LUs, there is nothing to be defined in z/OS if DYNLU=YES is used; otherwise a CDRSC definition is required for each independent LU in VTAM.

6.3 Configuring EE with Communications Server for AIX

Configuring CS/AIX involves many steps. In this section we describe the configuration steps in the following order:

- "Configuring node"
- "Configuring connectivity"
- "Configuring the dependent LUs with DLUR"
- "Configuring independent LUs"
- "Configuring LOGMODEs"

There are five methods of configuring CS/AIX:

- **`smit`** command
- **`snaadmin`** command
- **`xsnaadmin`** using X windows
- Modifying /etc/sna_node.cfg file with text editor such as vi
- Web based administration tool

Throughout this chapter we use the **`smit`** command and the **`snaadmin`** command for our configuration. For using xsnaadmin, refer to Chapter 7, "Enabling EE in CS for Linux" on page 229, because definition procedures are similar for CS/AIX.

Important: Editing /etc/sna/sna_node.cfg directly bypasses the syntax and consistency checking done by the smit/snaadmin/xsnaadmin methods and is therefore not recommended.

If you want to modify the /etc/sna/sna_node.cfg file directly, always *stop* the sna services first by **`sna stop`** command. CS/AIX has a copy of the configuration in its memory when the sna services are running. When a change is needed while sna services are running, first stop the resource for which you want to change the parameter, and use the **`snaadmin`** command instead of modifying the /etc/sna/sna_node.cfg file directly. Every parameter you see in the /etc/sna/sna_node.cfg file can be modified using the **`snaadmin`** command line utility though it cannot be modified using **`smit`** or **`xsnaadmin`**.

6.3.1 Configuring node

You must configure the properties of the node first. You can go to the node configuration panel with the **`smit`** command as follows:

smit sna → Configure SNA Resources → Local Node Resources → Node Definition.

You can then see the configuration screen as shown in Example 6-1 on page 198.

Example 6-1 Configuring the properties of node

```
                                 Node Definition

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* Control Point alias                                 [EECPAIX]         1
  Description                                         []
* Control Point name                                  [RDBOOKEE.EECPAIX]  2
  APPN Support                                         END_NODE         3       +
  Node ID                                             [07100000]                X
  NM-API Style                                         NORMAL                   +

  If BACK_LEVEL,

        Queue NMVTs?                                   NO                       +
```

Here are explanatory notes to Example 6-1:

- 1: This is just an alias of this control point. Control point is a unique identity in an APPN environment, and represents this server and is the same as a host name for TCP/IP. Give a meaningful name for maintenance purpose.

  > **Note:** An * (asterisk) just before the name of a parameter indicates that the particular parameter is mandatory. You must, therefore, define all parameters which have an asterisk before them.

- 2: This is the fully qualified control point name. It is divided into two parts separated by a dot. The first part is called NETID of an SNA network, and the second part is the CP name of that server. The CP name must be unique in an SNA network. Which means that in a network with the same NETID, the CP name must be unique. You must contact the z/OS administrator to get the correct NETID that your system will use. In this book, we use RDBOOKEE as a NETID.
- 3: Usually, CS/AIX participates in an SNA network as an End Node (EN). But CS/AIX can also act as a Network Node (NN) or branch network node in special cases. If there is no need to route SNA sessions through this server, then set this server as an EN. If a node is set to NN, it will participate in the locate search process, and this will use more system resources.

With **snaadmin** command, use:

```
snaadmin define_node, cp_alias=EECPAIX, node_type=END_NODE,
fqcp_name=RDBOOKEE.EECPAIX
```

snaadmin -hd command is very useful to get the parameter list of each subcommand.

Example 6-2 snaadmin -hd command

```
# snaadmin -hd define_node

This command defines a new node, or modifies an existing node.
It must be issued to a server where the node is not running.  It cannot be
issued to a running node.
```

```
Name                        Type         Length/Range
----                        ----         ------------
Mandatory fields:
cp_alias                    character   8
fqcp_name                   character   17

Defaulted fields (default in parentheses):
description                 character   31  (null string)
node_type                   constant    LEN_NODE, NETWORK_NODE, END_NODE,
                                        BRANCH_NETWORK_NODE  (END_NODE)
mode_to_cos_map_supp        constant    NO, YES  (YES)
mds_supported               constant    NO, YES  (YES)
node_id                     hex         4  (0x7)
.........................................................................
```

Example 6-2 shows the partial output of **snaadmin -hd define_node** command. You can find the mandatory fields easily.

6.3.2 Configuring connectivity

Now, we configure the connectivity resources for this node. We describe the configuration steps in the following order:

- “Configuring CS/AIX for connectivity using EE”
- “Configuring VTAM for connectivity of CS/AIX using EE”

Configuring CS/AIX for connectivity using EE

You must define the following resources in order to connect to z/OS VTAM:

- “Data Link Control (DLC)”
- “Port”
- “Link station”
- “Configuring connection network”

Data Link Control (DLC)

You must specify to the CS/AIX which “type of physical connection” you would be using. DLC controls the most low level in communicating with partners. There are various types of DLC but we use Enterprise Extender (HPR/IP) as our DLC. You can go to the DLC configuration panel as follows:

smit sna → **Configure SNA Resources** → **Local Node Resources** → **Connectivity** → **DLCs, Ports and Link Stations** → **Add Connectivity Resources** → **Add Enterprise Extender (HPR over IP) Resources** → **Add Enterprise Extender DLC**

You can then see the configuration panel as shown in Example 6-3.

Example 6-3 Configuring the properties of EE DLC

```
                            Add Enterprise Extender DLC

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* DLC name                                            [EEDLC] 1
```

```
  Description                                         []
  Initially active?                                    YES    2                  +
  LLC UDP port number                                 [12000] 3                  #
  Network UDP port number                             [12001] 4                  #
  High UDP port number                                [12002] 5                  #
  Medium UDP port number                              [12003] 6                  #
  Low UDP port number                                 [12004] 7                  #
```

Here are explanatory notes to Example 6-3:

- 1: This is the name of this DLC. For EE only one port is sufficient for all EE connections, therefore, you should define only one DLC for this server. This name is meaningful only to this server, so the same name can be used for the other server in the same APPN environment.
- 2: If you make this DLC as initially-active, then the DLC is started automatically when the node is started. We recommend configuring the port to be initially-active for convenience.
- 3~7: These UDP ports are used for various purposes, and all these ports are required for EE. Take care to change these port numbers because the z/OS must also have the same definitions. The default is 12000~12004.

With **snaadmin** command, use:

```
snaadmin define_ip_dlc, dlc_name=EEDLC, initially_active=YES
```

Port

You must define the port for each interface that you will use. To do this, go to the appropriate configuration panel as follows:

smit sna → Configure SNA Resources → Local Node Resources → Connectivity → DLCs, Ports and Link Stations → Add Connectivity Resources → Add Enterprise Extender (HPR over IP) Resources → Add Enterprise Extender Port

This takes you to the panel as shown in Example 6-4.

Example 6-4 Configuring the properties of the port

```
                                    Add Enterprise Extender Port

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* Port name                                           [EEPORT]              1
  Description                                         []
* DLC name                                            [EEDLC]               2    +
  Initially active?                                    YES                  3  +
  Maximum BTU size allowed                            [1500]                4    #
  Maximum number of active links allowed              [4096]                     #
  Implicit links to end nodes are uplinks?             NO                        +
  Local IP interface                                  [en1]                 5
```

Here are explanatory notes to Example 6-4:

- 1: This is the name of the port. This name is meaningful only for this node.
- 2: This is the name of the DLC that this port will use. Press **F4**, to see the list of DLCs already defined in this node, and select **EEDLC** in this case.

- 3: Keep it as “Initially active” for your convenience.
- 4: Maximum BTU size is the SNA packet size including the Transmission Header (TH) and the Request/Response Header (RH). This value is negotiated when the link station is connected to the other node (when the XID is exchanged). Leave it at the default value, 1500 bytes, if you do not have any problem on it. Usually, this default value is enough in most cases. But if you use NAT or IPSec across to the partner node, then you might have to decrease this maximum BTU size because of fragmentation issues. This value is overridden by the maximum BTU size of the link station that uses this port.
- 5: You can specify the interface that you want to use for EE. Local IP interface will determine the local IP address on which CS/AIX will listen for UDP ports 12000-12004. This field should contain the VIPA, or EtherChannel address, or a non-default interface (if there are multiple interfaces installed). The default interface is determined by reverse-lookup of the AIX host name.

> **Note:** You have to be careful with the default route of AIX. Even if you specify the interface which you want to use for EE traffics, all EE traffics may be sent through the interface which has the default route, if you do not specify any static route (when the partner node is in a different subnet from this node). Define the host route to be able to predict the path of EE communication if you want to use a specific interface for physical transfer.

You can do the same thing using the **snaadmin** command as follows:

```
snaadmin define_ip_port, port_name=EEPORT, dlc_name=EEDLC, local_ip_interface=en1
```

Link station

You can use the link stations as your “real connection” to the other node. With defined DLCs and ports, you can make as many different link stations as you want. In our scenario, we make two link stations for SC30M and SC31M, so we get one TG to SC30M and another TG to SC31M. SC30M and SC31M are connected by various methods (EE, MPC+, and XCF) so you can reach each system by using two paths (one for direct connection, and another for using another z/OS system). In fact, Virtual Telecommunications Access Method (VTAM) PU definition (in z/OS) and this link station definition are a one-to-one match.

You can go to the link station definition panel as follows:

smit sna → Configure SNA Resources → Local Node Resources → Connectivity → DLCs, Ports and Link Stations → Add Connectivity Resources → Add Enterprise Extender (HPR over IP) Resources → Add Enterprise Extender Link Station

This takes you to the panel as shown in Example 6-5.

Example 6-5 Configuring the properties of EELINK01

```
                       Add Enterprise Extender Link Station

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* Link station name                                 [EELINK01]              1
  Description                                       []
* Port name                                         [EEPORT]                2  +
  Activation                                         ON_NODE_STARTUP        3 +
  Contact Information:
```

```
* Remote IP host                                      [SC30M-EE2.itso.ibm.com] 4
  Independent LU Traffic:
  Remote node Control Point name                      [RDBOOKEE.SC30M]         5
  Remote node type                                     NETWORK_NODE            6  +
  Advanced:
  Maximum BTU size to be sent                         [1500]                      #
  Request CP-CP sessions?                              YES                        +
  Remote node is a network node server                 YES                     7  +
  TG number                                           [0]                      8  #
  Branch link type                                     NONE                       +
  Reactivate link station after failure                YES                        +
  Restart on normal deactivation                       NO                         +
  Acknowledgement timeout                             [10000]                     #
  Maximum retry count                                 [4]                         #
  Liveness timeout                                    [10000]                     #
```

Her are explanatory notes to Example 6-5:

- 1: This is the name of the link station. This name is meaningful to only this server.
- 2: Press **F4**, and choose the appropriate port name in your environment. In our case, the name of the port is EEPORT.
- 3: There are three methods of activation.
  - BY_ADMINISTRATOR: operator should start this link station manually.
  - ON_NODE_STARTUP: started automatically when the node is started.
  - ON_DEMAND: started when the first request is received by this link station.

  We recommend choosing `ON_NODE_STARTUP` for your convenience.
- 4: The IP address or host name of the remote host. In our scenario, we use the host name instead of IP address of each system.
- 5: The cp name of the remote node. Specify the fully qualified CP name (including NETID and CP name) here.
- 6: The node type of the remote host. In this case, the remote node is a Network Node (NN).
- 7: You can specify that this remote node is the preferred network node server (NNS). End Node (EN) can make two CP-CP sessions with only one NN at one time. This NN is called the network node server of the EN. If you do not specify which node is the NNS, then the first NN which makes CP-CP sessions with the EN is chosen to be the NNS of that EN. In our scenario, SC30M is chosen to be the preferred network node server, and SC31M will be a backup network node.
- 8: Transmission Group (TG) is a communication path between two communication partners in an SNA network. Two communication partners can have multiple TGs at the same time, and each TG has a different TG number. We use the TG number as 0 (default). If TG number is set at 0 on both sides, then the default value (21) will be used.

> **Note:** You cannot assign the same TG number to more than two TGs between the same communication partners. For one pair of communication, the TG name must be unique. So if you do not specify the TG number explicitly (0, by default), and there are two parallel TGs between one pair, then the TG number will be assigned as 21, 22 for each (up to 255).

With **snaadmin** command, use:

```
snaadmin define_ip_ls, ls_name=EELINK01, port_name=EEPORT,
adj_cp_type=NETWORK_NODE, default_nn_server=YES, initially_active=YES,
remote_ip_host=SC30M-EE2.itso.ibm.com, adj_cp_name=RDBOOKEE.SC30M
```

For the second link station, we configure as shown in Example 6-6.

Example 6-6 Configuring the properties of EELINK02

```
                         Add Enterprise Extender Link Station

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                           [Entry Fields]
* Link station name                                     [EELINK01]
  Description                                           []
* Port name                                             [EEPORT]             +
  Activation                                             ON_NODE_STARTUP     +
  Contact Information:
* Remote IP host                                        [SC31M-EE2.itso.ibm.com]
  Independent LU Traffic:
  Remote node Control Point name                        [RDBOOKEE.SC31M]
  Remote node type                                       NETWORK_NODE         +
  Advanced:
  Maximum BTU size to be sent                           [1500]                #
  Request CP-CP sessions?                                YES                  +
  Remote node is a network node server                   NO                   +
  TG number                                             [0]                   #
  Branch link type                                       NONE                 +
  Reactivate link station after failure                  YES                  +
  Restart on normal deactivation                         NO                   +
  Acknowledgement timeout                               [10000]               #
  Maximum retry count                                   [4]                   #
  Liveness timeout                                      [10000]               #
```

You can use the following **snaadmin** command:

```
snaadmin define_ip_ls, ls_name=EELINK02, port_name=EEPORT,
adj_cp_type=NETWORK_NODE, initially_active=YES,
remote_ip_host=SC31M-EE2.itso.ibm.com, adj_cp_name=RDBOOKEE.SC31M
```

Configuring connection network

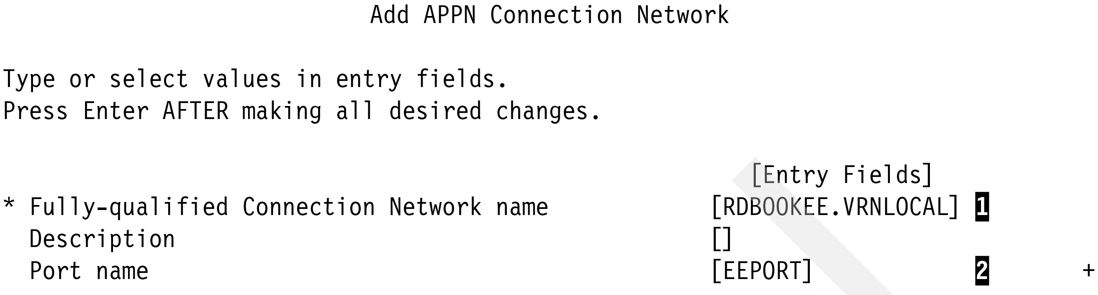
In our scenario, we use a connection network to communicate with other nodes, except the network node server and the backup network node server. For more information about connection network, refer to “APPN connection networks” on page 38

In CS/AIX, you can define the connection network in the following smit panel:

smit sna → Configure SNA Resources → Local Node Resources → Connectivity → APPN Connection Networks → Add APPN Connection Network

You can create multiple connection networks in the CS/AIX. Example 6-7 on page 204 shows the connection network setup panel.

Example 6-7   Defining Connection Network



```
                             Add APPN Connection Network

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* Fully-qualified Connection Network name           [RDBOOKEE.VRNLOCAL] 1
  Description                                       []
  Port name                                         [EEPORT]            2      +
```

Here are explanatory notes to Example 6-7:

- 1: This is the name of the virtual node which acts as a connection network, and all other nodes must also use this virtual node to participate in the same connection network.
- 2: This is the name of the port through which CS/AIX contacts the connection network.

With **snaadmin** command, use:

```
snaadmin define_cn, fqcn_name=RDBOOKEE.VRNLOCAL, port_name=EEPORT
```

Configuring VTAM for connectivity of CS/AIX using EE

The APPN environment supports dynamic definitions. This means you can connect to another APPN node without node-specific definitions. VTAM controls this behavior with the DYNPU parameter in the external communications adapter (XCA) major node definition. In our scenario, we do not use dynamic PU definitions, therefore we show the VTAM definitions for the CS/AIX to connect to the z/OS LPARs in this section.

You must define the following to configure the EE connection of CS/AIX:

- “Enterprise Extender XCA major node” on page 204
- “PU definition for the CS/AIX node”

Enterprise Extender XCA major node

We use two IP addresses for EE on VTAM side (with multiple VIPA). In Example 6-8, we show the XCA major node of SC30M.

> **Note:** VTAM defaults to an SAP® address of 08 for the remote node when MEDIUM=HPRIP is specified in the XCA major node. However, CS/AIX has a local SAP address of 04, by default. This can cause a problem when VTAM does a dial-out over a VRN to CS/AIX, therefore CALL=IN must be specified on the GROUP.
>
> With CS/AIX V6.3.0.3 or later, you can have multiple define_ip_ports with different lsap_address values and the same local_ip_interface value, which point to different define_ip_dlc definitions.

Example 6-8   XCA major node for SC30M

```
EEXCA30         VBUILD TYPE=XCA
EEPORT30          PORT MEDIUM=HPRIP,                                 *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),               *
               IPRESOLV=2,                                           *
               SRQTIME=15,SRQRETRY=3
*
EEGVL301       GROUP DIAL=YES,CALL=IN,                               *
```

```
               HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                        * 1
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                  * 2
               TGP=EEXTCAMP,                                           *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              * 3
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM30,EEN30)
*
EEGVG301       GROUP DIAL=YES,CALL=IN,                                 *
               HOSTNAME=SC30M-EE2.ITSO.IBM.COM,                        * 4
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      * 5
               TGP=EEXTWAN,                                            *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               * 6
               UNRCHTIM=180,                                           *
               AUTOGEN=(16,EEX30,EEY30)
```

Here are explanatory notes to Example 6-8:

- 1: Hostname for the first group.
- 2: The name of a connection network is RDBOOKEE.VRNLOCAL and its type is local. All participants of this connection network must be located within the same APPN topology subnetwork (the same NETID).
- 3: Dynamic definition of PU is allowed for this group.
- 4: Hostname for the second group
- 5: The name of a connection network is W3IBMCOM.VRN and its type is global. Nodes located in different APPN topology subnetworks can share the same connection network.
- 6: Dynamic definition of PU is prohibited for this group, predefined PU is required.

We use the second group for connection via predefined link stations. Dynamic PU definition is prohibited for this group, so we must define a PU for CS/AIX node. For connection network, we use the local connection network because CS/AIX node is in the same APPN subnetwork (use the same NETID) with other nodes. Using the connection network implies that DYNPU=YES, no VTAM definition is required for CS/AIX node through connection network. For more information about VTAM definition and XCA major node, refer to 4.4.2, “Defining the XCA HPRIP major node” on page 142.

PU definition for the CS/AIX node

Each system (SC30M and SC31M) should have the definition, as shown in Example 6-9, if the DYNPU=NO in XCA major node of VTAM.

Example 6-9 PU definition for CS/AIX

```
EESWAIXO VBUILD TYPE=SWNET
EEPUAIX  PU    CPNAME=EECPAIX,                             1           *
               NETID=RDBOOKEE,                             2           *
               TGN=06,                                     3           *
               CPCP=YES,                                   4           *
               HPR=YES,                                                *
               DYNLU=YES,                                  5           *
               ISTATUS=ACTIVE
```

These are the explanatory notes to Example 6-9:

- 1: The CP name of the CS/AIX node. This name must be the same as the name you configured in node definition of CS/AIX.
- 2: The NETID of this SNA network.
- 3: The TG number which is used for this link. We defined the TG number as 0 on the IBM Communications Server for AIX side (see Example 6-5 on page 201). With this configuration, the TG number will be decided when the XID is exchanged, and the TG number will be 6.
- 4: The CP-CP sessions capability. Set this parameter to YES.
- 5: The dynamic LU definition feature. If this parameter is set to YES, then all the independent LUs defined in the CS/AIX can be used in this z/OS VTAM without a CDRSC definition in VTAM. Otherwise, you have to define a CDRSC entry for each independent LU.

6.3.3 Configuring the dependent LUs with DLUR

We define dependent LUs and DLUR PUs in this section. For further information about the dependent LU and the DLUR, refer to “Dependent LU Requester and Server (DLUR/DLUS)” on page 33.

The following topics are covered:

- “DLUR definition in CS/AIX”
- “DLUR definition in VTAM”

DLUR definition in CS/AIX

For using the dependent LUs, you have to define in CS/AIX:

- “DLUR PU”
- “Dependent LUs”

DLUR PU

DLUR PU is the type 2.0 PU which provides PU services to the dependent LUs.

You can go to the DLUR PU definition panel as follows:

smit sna → Configure SNA resources → Local Node Resources → Connectivity → DLUR PUs → Add DLUR PU

This takes you to the DLUR PU definition panel, as shown in Example 6-10.

Example 6-10   Configuring the properties of DLUR PU

```
                                  Add DLUR PU

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* PU name                                           [DLAIXPU1]            1
  Description                                       []
  DLUS server name                                  [RDBOOKEE.SC30M]      2
  Backup DLUS server name                           [RDBOOKEE.SC31M]      3
* PU id                                             [01000001]            4   X
  Initially active?                                  YES                  5   +
  Retry contacting DLUS indefinitely?               [65535]                 6   #
          (0=No, 65535=Yes, else number of retries)
  Compression Supported                              NO                          +
```

Here are explanatory notes to Example 6-10:

- 1: This is the PU name for this DLUR PU for dependent LUs. This name is meaningful only in this CS/AIX, so you can use a different name from the PU name in the VTAM definition. But for management purposes, we recommend using the same name as your VTAM definition.
- 2: This is the SSCP name of the primary Dependent LU Server (DLUS) name. CS/AIX will try to establish a CPSVRMGR pipe (DLUS-DLUR sessions) with this DLUS. When this attempt fails, it will try to the backup DLUS if it is available.
- 3: This is the SSCP name of the backup DLUS. If you will use the backup DLUS, then the same PU and LU definitions must be configured in the VTAM of the backup DLUS.
- 4: This is the PU identification number. VTAM has IDBLK (ID block) and IDNUM (ID number) parameters in its PU definition for this client, and the two definitions must be the same. In this case, the IDBLK is 010, and IDNUM is 00001. This value should be unique in a VTAM.
- 5: We define this PU as initially_active for convenience of management.
- 6: If the attempt to establish CPSVRMGR pipe fails, this DLUR PU will retry as configured in this parameter. If you specify 0 (by default), then it will not retry. We define 65535, then this DLUR PU will always retry when it cannot contact DLUS appropriately.

With **snaadmin** command, use:

```
snaadmin define_internal_pu, pu_name=DLAIXPU1, pu_id=0x01000001,
dlus_name=RDBOOKEE.SC30M, bkup_dlus_name=RDBOOKEE.SC31M, initially_active=yes,
dlus_retry_limit=65535
```

Note: You can define multiple DLUR PUs in a node. Because a PU can support only 255 dependent LUs, if you want more LUs you have to define more DLUR PUs to the same primary/backup DLUS pair. Moreover it is possible to define more DLUR PUs to the other primary/backup DLUS pair.

Dependent LUs

Now, we define the dependent LUs on the defined DLUR PU. In most cases, it is easier to define the dependent LUs using a range definition because all the properties of those LUs are the same. You can define a range of dependent LUs using following **smit** sequence:

smit sna → Configure SNA Resources → Local Node Resources → Local LUs → LU 0-3 → Add Range of LUs Type 0-3

Next you can define a range of LUs in the **smit** panel, such as Example 6-11.

Example 6-11 Configuring a range of dependent LUs

```
                            Add Range Of LUs Type 0-3

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* Base name for LUs                                   [DLAIX]       1
  Description                                         []
```

```
* Host LS/DLUR PU                                   [DLAIXPU1]    2          +
* LU number of first LU                             [2]           3          #
* LU number of last LU                              [6]           4          #
  Model type                                         UNKNOWN                 +
  Pool name (if pooled)                             []
  Inactivity timeout                                [0]                      #
  Restrict SSCP to SSCP id                          [0]
  Name attributes                                    NONE                    +
  Base number                                       [2]           5          #
```

Here are explanatory notes to Example 6-11:

- 1: This is the base name of dependent LUs. You can use up to 5 characters here, and the names of all the dependent LUs defined here will be started with this base name.
- 2: You can assign all these dependent LUs to some link station or DLUR PU. In this case, we use the DLUR PU for these dependent LUs.
- 3: This is the local address of the first LU of this range of LUs. The local address of LU must match the LOCADDR parameter in VTAM definition for each LU.

  > **Note:** LU name defined here is only meaningful for this CS/AIX. In other words, there is no need to match these LU names with the LU names in the VTAM definition. Only the LU number (LOCADDR) must be the same on both the sides. But for management purpose, we recommend using the same name on both the sides.

- 4: This is the local address of the last LU of this range of LUs. The number of LUs is decided at this phase.
- 5: You can specify a number from which to start naming the LUs in the range.

With **snaadmin** command, use:

```
snaadmin define_lu_0_to_3_range, base_name=DLAIX, pu_name=DLAIXPU1, min_nau=2,
max_nau=6, base_number=2
```

After this configuration, we have five dependent LUs from DLAIX001 through DLAIX005 in the CS/AIX.

DLUR definition in VTAM

In VTAM, you should define the type 2.0 PU, and the dependent LUs in a switched major node. In our scenario, we make a switched major node as shown in Example 6-12:

Example 6-12 Switched major node definition for DLUR

```
DLSWAIX  VBUILD TYPE=SWNET
DLAIXPU1 PU    PUTYPE=2,USSTAB=USSSNAEE,                              *
               IDBLK=010,IDNUM=00001,                                 * 1
               MODETAB=ALLMODES,DLOGMOD=DYNHIGH,ANS=CONT                2
DLLAIX02 LU    LOCADDR=2                                                3
DLLAIX03 LU    LOCADDR=3
DLLAIX04 LU    LOCADDR=4
DLLAIX05 LU    LOCADDR=5
DLLAIX06 LU    LOCADDR=6
```

Here are explanatory notes to Example 6-12:

- 1: IDBLK and IDNUM must match the PU ID in the DLUR PU definition in CS/AIX.

- 2: ANS=CONT is required for DLUR PUs. If the default value (ANS=STOP) is used, SSCP-PU session and SSCP-LU sessions will be broken in case the primary DLUS is lost. This results in session outages during takeover to backup DLUS.
- 2: The name of the LU may be different from that of CS/AIX, but the LOCADDR must match with the LU number in the DLUR PU definition of CS/AIX. However, we recommend using the same names for management purposes.

6.3.4 Configuring independent LUs

An independent LU has the capability of communication with other independent LUs without the SSCP (VTAM). Therefore, an independent LU is not a resource of VTAM, but a resource of the node in which the independent LU is defined. There are two ways of handling these independent LUs whether DYNLU parameter in VTAM is *yes* or *no*. If DYNLU is *yes*, then no definition is required for VTAM, and VTAM will generate a dynamic definition of a CDRSC entry for that independent LU. If DYNLU is *no*, then there must be a predefined CDRSC entry in the VTAM for that independent LU.

You can define the independent LU with the following **smit** commands:

smit sna → Configure SNA Resources → Local Node Resources → LU 6.2 Configuration → LU 6.2 → Add Independent LU Type 6.2

This will take you to the panel shown in Example 6-13.

Example 6-13 Defining an independent LU

```
                              Add Independent LU Type 6.2

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* LU alias                                              [ILAIXL01]     1
  Security Access List name                             []                  +
  Description                                           []
* LU name                                               [ILAIXL01]     2
  Support Syncpoint?                                     NO                 +
  Additional LU properties                               NONE               +
  Computer                                              []
  Timeout                                               [60]                #
```

Here are explanatory notes to Example 6-13:

- 1: This is the alias of the LU. This value is meaningful to only this CS/AIX node.
- 2: This is the name of the LU. This LU name will be used in z/OS also.

With **snaadmin** command, use:

```
snaadmin define_local_lu, lu_name=ILAIXL01, lu_alias=ILAIX01, lu_session_limit=0
```

If your DYNLU setting is *no*, then the CDRSC definition is needed in VTAM as shown in Example 6-14.

Example 6-14 CDRSC major node for an independent LU

```
EECDRSC  VBUILD TYPE=CDRSC
ILAIXL01 CDRSC
```

You have to define this CDRSC entry in all VTAMs which will use this independent LU.

6.3.5 Configuring LOGMODEs

Sometimes, your application can use the customized LOGMODE which is not shipped by IBM. In this case you can define a customized LOGMODE in the CS/AIX. You can define logmode for type 6.2 LUs only. As shown in Example 6-15, you can define the new LOGMODEs with the following `smit` commands:

smit sna → Configure SNA Resources → Local Node Resources → LU 6.2 Configuration → LU 6.2 Mode → Add Mode

Example 6-15 Define a mode

```
                                     Add Mode

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* Name                                                [#CONNECT]        1
  Description                                         []

  Session limits

        Maximum number of sessions                    [32767]           2        #
        Initial session limit                         [2]               3        #
        Min con. winner sessions                      [1]               4        #
        Min con. loser sessions                       [0]               5        #
        Auto-activate sessions                        [0]               6        #

  Receive pacing window

        Initial                                       [4]               7        #
        Maximum                                       [0]               8        #


  Use default RU sizes?                                YES              9        +

  If NO,

        Maximum RU size upper limit                   [1024]            10       #
        Maximum RU size lower limit                   [0]               11       #


  COS name                                            [#CONNECT]        12


  Use compression                                      PROHIBITED                +

  If REQUESTED,

        Maximum compression level                      NONE                      +
        Maximum decompression level                    NONE                      +
```

Here are explanatory notes to Example 6-15 on page 210:

- 1: The name of this logmode.
- 2: The maximum number of sessions which will be supported by this logmode. In many cases this value is set to the same value as the initial session limit. Through CNOS (Change Number of Sessions), this maximum number of sessions can be changed.
- 3: The default session limit of this logmode. This is the maximum number of sessions permitted between a pair of LUs using this mode, even with CNOS negotiation.
- 4: The minimum contention winner sessions. The number of sessions (up to the session limit) that CS/AIX must reserve for use by the local LU as the contention winner. Specify 0 if you do not want to reserve the contention winner sessions.
- 5: The minimum contention loser sessions. The minimum number of sessions that CS/AIX must reserve for use by the local LU as the contention loser. The sum of the minimum contention winner sessions and the minimum contention loser sessions must not exceed the initial session limit. Specify 0 if you do not want to reserve the contention loser sessions.
- 6: You can specify how many sessions to activate automatically for each pair of LUs that use this mode. This value is used when CNOS exchange is initiated implicitly.
- 7: The initial pacing window size. The initial number of request units (RUs) that the local LU can receive before it must send a pacing response to the remote LU.
- 8: The maximum pacing window size. The maximum number of request units (RUs) that the local LU can receive before it must send a pacing response to the remote LU. This value is optional. If it is not supplied, the maximum receive pacing window is unlimited. If a value is supplied, it is used to limit the size of the receive pacing window for adaptive pacing. If adaptive pacing is not used, this value is ignored.
- 9: You can specify whether CS/AIX uses the maximum RU size upper limit and maximum RU size lower limit parameters to define the maximum RU size. If you specify YES, CS/AIX ignores the following RU size parameters, and sets the upper bound for the maximum RU size to the largest value that can be accommodated in the link BTU size.
- 10: Upper bound for the maximum size of RUs sent and received on sessions in this mode. This value is used when the maximum RU size is negotiated during session activation.
- 11: Lower bound for the maximum size of RUs sent and received on sessions in this mode. The value is used when the maximum RU size is negotiated during session activation.
- 12: Name of the class of service (COS) to request when activating sessions on this mode. This must be defined in APPNCOS in VTAM.

In Example 6-15 on page 210, we define a new logmode named `#CONNECT` which uses the `#CONNECT` as its class of service. You can define the same logmode with the following command:

```
snaadmin define_mode, mode_name=#CONNECT, cos_name=#CONNECT
```

6.4 Verifying and managing EE with CS for AIX

Most administration work of CS/AIX is done by **sna** and **snaadmin** commands. In this section, we cover the following:

- “Starting resources”
- “Verifying resources”
- “Stopping resources”
- “Backing up your configuration”

6.4.1 Starting resources

Typically, the resources must be activated in the following order:

1. “Starting the SNA service”
2. “Starting the SNA node”
3. “Starting the DLC, port, and link stations”
4. “Starting the DLUR PUs”
5. “Starting the sessions”

Starting the SNA service

First, you must activate the SNA service by issuing the **`sna start`** command or by using the following smit panel:

smit sna → **Manage SNA Resources** → **Start SNA Resources** → **Start SNA**

Example 6-16 Starting the SNA services

```
# sna start
SNA software is initializing...
SNA software has been initialized.
```

You can see the messages as shown in Example 6-16 if your CS/AIX is working properly.

Starting the SNA node

You must start the SNA node for participating in the SNA network. You can use the **`snaadmin`** command as shown in Example 6-17, or use the smit panel. You can use the `Manage SNA Resources` smit panel for all operations of the resources of this CS/AIX node.

Example 6-17 Starting the SNA node

```
# snaadmin init_node
------------------------------------------------------------------------
init_node command completed successfully
------------------------------------------------------------------------
```

If you define all the connectivity resources such as DLCs, ports, link stations, and DLUR PUs as initially_active, then all these resources will be activated in this stage. You do not have to activate these resources manually in that case.

Starting the DLC, port, and link stations

You can also use the **`snaadmin`** commands as shown in Example 6-18, or the smit panel for starting the connectivity resources.

Example 6-18 Starting the DLC, port, and link stations

```
# snaadmin start_dlc, dlc_name=EEDLC
------------------------------------------------------------------------
start_dlc command completed successfully
------------------------------------------------------------------------
# snaadmin start_port, port_name=EEPORT
------------------------------------------------------------------------
start_port command completed successfully
------------------------------------------------------------------------
# snaadmin start_ls, ls_name=EELINK01
------------------------------------------------------------------------
```

```
start_ls command completed successfully
------------------------------------------------------------------------
# snaadmin start_ls, ls_name=EELINK02
------------------------------------------------------------------------
start_ls command completed successfully
------------------------------------------------------------------------
```

Starting the DLUR PUs

Starting the DLUR PUs is the same procedure as starting the other connectivity resources such as, DLC, port, and link stations. We show only the **snaadmin** command. See Example 6-19.

Example 6-19 Starting the DLUR PU

```
# snaadmin start_internal_pu, pu_name=DLAIXPU1
------------------------------------------------------------------------
start_internal_pu command completed successfully
------------------------------------------------------------------------
```

Starting the sessions

Now that all the resources are activated, you can start the sessions you need. Usually, making scripts for activating and deactivating sessions is a good idea for management convenience. You can use the **snaadmin activate_session** command.

6.4.2 Verifying resources

In this section, we show you how you to verify various resources in CS/AIX. You can use several tools to monitor CS/AIX such as smit, **sna** command, **snnadmin** command, xsnaadmin, and the Web administration tool. From these tools, we use the **sna** and **snaadmin** commands only. For other tools and detailed information, refer to:

- *IBM Communications Server for AIX Administration Command Reference*, SC31-8587
- *IBM Communications Server for AIX Administration Guide*, SC31-8586

Note that "snaadmin query_ls" and "snaadmin query_session" show more details and should be used instead of "sna -d l" and "sna -d s" with CS/AIX V4.2 and later.

This section will cover the following topics:

- “Verifying the connectivity to other nodes”
- “Verifying the DLUR PUs and LUs”
- “Verifying HPR path switch”
- “Verifying backup DLUS”

If you are not familiar with HPR, RTP and DLUS/DLUR, refer to 1.2.1, “Positioning APPN and HPR” on page 4.

Verifying the connectivity to other nodes

You can verify the connectivity to other nodes with **sna -d l** command.

Example 6-20 sna -d l command

```
# sna -d l

    Link           Adjacent      Node    Device             # of local  In
   station         CP name       type     name      State    sessions   use
```

```
-------------- ----------------- ----- --------- ---------- ---------- -----
EELINK01       RDBOOKEE.SC30M    NN    hpr/ip    Active     0          No
EELINK02       RDBOOKEE.SC31M    NN    hpr/ip    Active     0          No
DLAIXPU1       RDBOOKEE.SC30M    DLUS  n/a       Active     5          Yes
```

In Example 6-20 on page 213, the output of the **sna -d l** command shows that two link stations are in an **Active** State. EELINK01 is connected to the SC30M, and EELINK02 is connected to SC31M by hpr/ip.

If an end node is connected to its network node server, then two CP-CP sessions are established between the two nodes. In our scenario, the preferred network node server is SC30M. With these CP-CP sessions, an end node can get topology and directory information from the network node server.

Example 6-21 sna -d s command

```
# sna -d s
      Local            Partner        Mode      Link
     LU name           LU name        name     station     State
----------------- ----------------- -------- ----------- ----------
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
```

In “sna -d s command” on page 214, the **sna-d s** command shows that two CP-CP sessions are established with SC30M through the @R000001 link station using CPSVCMG logmode because SC30M is the preferred network node server of the EECPAIX node. Recall the definition panel of the two link stations, EELINK01 and EELINK02 in Example 6-5 on page 201 and Example 6-6 on page 203. We defined the Remote node is a network node server field as `yes` only in EELINK01 (towards SC30M) link station definition panel.

The @R000001 is a dynamically created RTP pipe. If you want to know which physical link station is the first hop for this @R000001 RTP pipe, you can use the **snaadmin query_rtp_connection command** as shown in Example 6-22.

Example 6-22 snaadmin query_rtp_connection command

```
# snaadmin query_rtp_connection, rtp_name=@R000001

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000001
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2
------------------------------------------------------------------------
```

Now, you can see that EELINK01 is the first hop for RTP pipe @R000001.

Next, we check the connection network (see Example 6-20 on page 213). The connection network does not support the CP-CP sessions, so you must predefine link stations to the network node server and the backup network node server, if it exists. We have another z/OS LPAR in our scenario, SC32M. SC32M is connected to the same connection network, RDBOOKEE.VRNLOCAL.

Example 6-23 Verifying the connection network

```
# sna -d l

    Link           Adjacent      Node    Device              # of local  In
   station         CP name       type     name      State    sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
EELINK01       RDBOOKEE.SC30M    NN    hpr/ip    Active     0          No
EELINK02       RDBOOKEE.SC31M    NN    hpr/ip    Active     0          No
DLAIXPU1       RDBOOKEE.SC30M    DLUS  n/a       Active     5          Yes

# aping RDBOOKEE.SC32M

IBM aping version 2.44 APPC echo test with timings.
Licensed Materials - Property of IBM
(C) Copyright 1994,1995 by IBM Corp. All rights reserved.

Allocate duration:      50 ms
Program startup and Confirm duration:       0 ms

Connected to a partner running on: (UNKNOWN operating system)
        Duration        Data Sent        Data Rate        Data Rate
        (msec)          (bytes)          (KB/s)           (Mb/s)
        --------        ---------        ---------        ---------
               0              200
               0              200
Totals:        0              400
Duration statistics: Min = 0 Ave = 0 Max = 0
# sna -d l
    Link           Adjacent      Node    Device              # of local  In
   station         CP name       type     name      State    sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
@D000001       RDBOOKEE.SC32M    VN    hpr/ip    Active     0          No
EELINK01       RDBOOKEE.SC30M    NN    hpr/ip    Active     0          No
EELINK02       RDBOOKEE.SC31M    NN    hpr/ip    Active     0          No
DLAIXPU1       RDBOOKEE.SC30M    DLUS  n/a       Active     5          Yes

# snaadmin query_ls, ls_name=@D000001

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

ls_name = @D000001
description = ""
dlc_type = HPRIP
state = ACTIVE
act_sess_count = 0
det_adj_cp_name = RDBOOKEE.SC32M
det_adj_cp_type = VRN
port_name = EEPORT
```

```
adj_cp_name = RDBOOKEE.SC32M
adj_cp_type = VRN
------------------------------------------------------------------------
```

In Example 6-23 on page 215, a new dynamic link station named @D000001 is created through connection network. You will observe that the adjacent CP type is VRN.

Verifying the DLUR PUs and LUs

If DLUR PU is started, two DLUR-DLUS sessions (CPSVRMGR pipe) are established between DLUR and DLUS.

Example 6-24 Verifying DLUR PU

```
# sna -d l

    Link           Adjacent      Node    Device              # of local  In
   station         CP name       type     name      State     sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
EELINK01       RDBOOKEE.SC30M    NN    hpr/ip    Active     0          No
EELINK02       RDBOOKEE.SC31M    NN    hpr/ip    Active     0          No
DLAIXPU1       RDBOOKEE.SC30M    DLUS  n/a       Active     5          Yes
```

Example 6-24 shows a DLUR PU, DLAIXPU1 in the link station fields. This DLUR PU is connected to SC30M because it is the primary DLUS for this node. See `the number of local sessions` field. Five sessions are established through DLAIXPU1. These are SSCP-LU sessions serviced by this DLUR PU, DLAIXPU1.

Example 6-25 Verifying DLUR-DLUS sessions (CPSVRMGR pipe)

```
# sna -d s

      Local              Partner         Mode       Link
     LU name             LU name         name     station      State
----------------- ----------------- -------- ----------- ----------
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
```

Example 6-25 shows that two sessions using CPSVRMGR as their logmode are established with SC30M. These two sessions are called DLUS-DLUR sessions or CPSVRMGR pipe. They use the @R000002 RTP connection.

Example 6-26 snaadmin status_dlur command

```
# snaadmin status_dlur

--------------------------------------------------------------------------------
DLUR PU  LU       Status         DLUS              PLU              Description
--------------------------------------------------------------------------------
DLAIXPU1          Active         RDBOOKEE.SC30M
         DLAIX002 SSCP           RDBOOKEE.SC30M
         DLAIX003 SSCP           RDBOOKEE.SC30M
         DLAIX004 SSCP           RDBOOKEE.SC30M
         DLAIX005 SSCP           RDBOOKEE.SC30M
         DLAIX006 SSCP           RDBOOKEE.SC30M
--------------------------------------------------------------------------------
```

Example 6-26 shows the output of **snaadmin status_dlur** command. The status of five dependent LUs is **SSCP**. SSCP status means that the SSCP-LU session is successfully established. If an LU successfully establishes an LU-LU session with another LU, the status changes to `Active` as shown in Example 6-27.

Example 6-27 LU-LU session is established

```
# snaadmin status_dlur

--------------------------------------------------------------------------------
DLUR PU  LU       Status         DLUS              PLU              Description
--------------------------------------------------------------------------------
DLAIXPU1          Active         RDBOOKEE.SC30M
         DLAIX002 Active    1    RDBOOKEE.SC30M    RDBOOKEE.SC30TS02   2
         DLAIX003 SSCP           RDBOOKEE.SC30M
         DLAIX004 SSCP           RDBOOKEE.SC30M
         DLAIX005 SSCP           RDBOOKEE.SC30M
         DLAIX006 SSCP           RDBOOKEE.SC30M
--------------------------------------------------------------------------------

# sna -d l

    Link          Adjacent      Node    Device                # of local  In
   station        CP name       type     name       State      sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
EELINK01      RDBOOKEE.SC30M    NN    hpr/ip    Active     0          No
EELINK02      RDBOOKEE.SC31M    NN    hpr/ip    Active     0          No
DLAIXPU1      RDBOOKEE.SC30M    DLUS  n/a       Active     6   3      Yes

# snaadmin query_rtp_connection

------------------------------------------------------------------------
list_options = SUMMARY + FIRST_IN_LIST
.........................................................................
rtp_name = @R000004   4
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = #INTER   5
num_sess_active = 1   6
```

Here are explanatory notes to Example 6-27:

- 1~2: The status of DLAIX002 LU is now **Active** because now this LU has an LU-LU session with SC30TSO2.
- 3: The number of sessions is now 6 because an LU-LU session is established using DLAIX002 LU.
- 4~6: This RTP connection is created for delivering traffics for the LU-LU session between DLAIX002 and SC30TSO2. Its class of service is #INTER and the number of sessions is 1.

There is one independent LU which is defined in the previous step. In Example 6-26, the status of the ILAIXL01 is `inactive`. But, there is no need to activate this LU. You can just start the LU-LU sessions with other LUs. We try to *aping* from this ILAIXL01 LU to SC30M.

Example 6-28 aping from ILAIXL01 to SC30M

```
# snaadmin aping, lu_name=ILAIXL01, fqplu_name=RDBOOKEE.SC30M, mode_name=#CONNECT,
iterations=4
------------------------------------------------------------------------
alloc_time = 8
min_time = 0
avg_time = 1
max_time = 1
partner_ver_len = 0
------------------------------------------------------------------------

# snaadmin status_lu62

--------------------------------------------------------------------------------
LU        LU alias  Machine   Partner LU          Mode      Session Count
--------------------------------------------------------------------------------
EECPAIX   EECPAIX             RDBOOKEE.SC30M      CPSVCMG   2 Sessions
                              RDBOOKEE.SC30M      CPSVRMGR  2 Sessions
ILAIXL01  ILAIX01             RDBOOKEE.SC30M      #CONNECT  1 Session
--------------------------------------------------------------------------------
```

In Example 6-28, we used the **snaadmin aping** command to aping to SC30M. You can specify the logmode also; we tried with `#CONNECT` because a number of real LU-LU sessions use this logmode.

Verifying HPR path switch

The RTP path can be changed if the topology is changed. We show this HPR Path Switch. We simulate a network problem by deactivating the EELINK01 link station in CS/AIX.

First, see Example 6-29 for verifying the status *before* the HPR path switch occurs.

Example 6-29 Before HPR path switch

```
# sna -d l

    Link          Adjacent      Node    Device             # of local  In
   station         CP name      type     name      State    sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
EELINK01       RDBOOKEE.SC30M    NN    hpr/ip    Active     0          No
EELINK02       RDBOOKEE.SC31M    NN    hpr/ip    Active     0          No
DLAIXPU1       RDBOOKEE.SC30M    DLUS  n/a       Active     6          Yes
```

```
# sna -d s

      Local            Partner        Mode      Link
     LU name           LU name        name     station     State
----------------- ----------------- -------- ----------- ----------
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available

# snaadmin query_rtp_connection,rtp_name=@R000001

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000001
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2

------------------------------------------------------------------------

# snaadmin query_rtp_connection,rtp_name=@R000002

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000002
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------

# snaadmin query_rtp_connection,rtp_name=@R000004

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000004
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = #INTER
num_sess_active = 1
------------------------------------------------------------------------
```

See Example 6-29. Before shutdown of the EELINK01, three RTP connections are using the EELINK01 as their next hop as follows:

- @R000001 is for CP-CP sessions

- @R000002 is for CPSVRMGR pipe
- @R000003 is for the dependent LU-LU session.

Now, we deactivate the EELINK01 link station.

Example 6-30 After HPR path switch

```
# snaadmin stop_ls, ls_name=EELINK01

------------------------------------------------------------------------
stop_ls command completed successfully
------------------------------------------------------------------------

# sna -d l
    Link           Adjacent      Node    Device             # of local  In
   station         CP name       type     name      State    sessions   use
-------------- ----------------- ----- --------- ---------- ---------- -----
@D000001       RDBOOKEE.SC30M    VN    hpr/ip    Active     0          No    1
EELINK02       RDBOOKEE.SC31M    NN    hpr/ip    Active     0          No
DLAIXPU1       RDBOOKEE.SC30M    DLUS  n/a       Active     6          Yes

# sna -d s

      Local            Partner        Mode      Link
     LU name           LU name        name     station     State
----------------- ----------------- -------- ----------- ----------
RDBOOKEE.EECPAIX  RDBOOKEE.SC31M    CPSVCMG  @R000005 2  Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC31M    CPSVCMG  @R000005 2  Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available

# snaadmin query_rtp_connection, rtp_name=@R000005

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000005
first_hop_ls_name = EELINK02   3
dest_node_name = RDBOOKEE.SC31M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2
------------------------------------------------------------------------

# snaadmin query_rtp_connection, rtp_name=@R000002

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000002
first_hop_ls_name = @D000001   4
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------
```

```
# snaadmin query_rtp_connection, rtp_name=@R000004

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000004
first_hop_ls_name = @D000001   4
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = #INTER
num_sess_active = 1
------------------------------------------------------------------------
```

Here are explanatory notes to Example 6-30:

- 1: A dynamic link station, @D000001 is created towards SC30M through connection network during HPR path switch.
- 2: CP-CP sessions used to be established through @R000001, but are now established through @R000005. The new RTP connection is created. CP-CP sessions must be established through a predefined link station and cannot be established through a dynamic link station through connection network. When the EELINK01 is shut down, two CP-CP sessions towards SC30M are broken, and two new CP-CP sessions towards SC31M are established with the new RTP connection.
- 3: The next hop of @R000005 is EELINK02.
- 4: Two DLUR-DLUS sessions (CPSVRMGR pipe) are still established with SC30M, but the path has been changed to @D000001. This is done by HPR path switch. When EELINK01 is shut down, the @R000002 RTP connection detects that the path is now unavailable, so it tries to path switch. In this case, making a dynamic link station to SC30M is the most efficient way to path switch. The @R000004 RTP connection also does the same. The dependent LU-LU session is established between DLAIX002 LU and SC30TSO2 LU in the SC30M, so this RTP connection also does a path switch with dynamic link station towards SC30M.

Now, we re-activate the EELINK01 link station.

Example 6-31 After reactivation of EELINK01

```
# snaadmin start_ls,ls_name=EELINK01

------------------------------------------------------------------------
start_ls command completed successfully
------------------------------------------------------------------------

# sna -d l

    Link          Adjacent      Node   Device              # of local  In
   station        CP name       type    name      State     sessions   use
-------------- ---------------- ----- --------- ---------- ---------- -----
@D000002       RDBOOKEE.SC30M   VN    hpr/ip    Active     0          No
EELINK01       RDBOOKEE.SC30M   NN    hpr/ip    Active     0          No   1
EELINK02       RDBOOKEE.SC31M   NN    hpr/ip    Active     0          No
DLAIXPU1       RDBOOKEE.SC30M   DLUS  n/a       Active     6          Yes

# sna -d s
```

```
     Local             Partner        Mode      Link
    LU name            LU name        name     station     State
----------------- ----------------- -------- ----------- ----------
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000007  2 Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000007  2 Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available

# snaadmin query_rtp_connection, rtp_name=@R000007

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000007
first_hop_ls_name = EELINK01   3
dest_node_name = RDBOOKEE.SC30M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2
------------------------------------------------------------------------

# snaadmin query_rtp_connection, rtp_name=@R000002

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000002
first_hop_ls_name = @D000002   4
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------
```

Here are explanatory notes to Example 6-31:

- 1: EELINK01 is now up.
- 2: CP-CP sessions are moved to SC30M again because SC30M is a preferred network node server.
- 3: This RTP connection is created for new CP-CP sessions.
- 4: The path of this RTP connection is not changed because there is no topology change.

We can manually change the path of an RTP connection. Here, the path of an RTP connection will be changed to the most efficient one.

Example 6-32 Manual HPR path switch

```
# snaadmin query_rtp_connection, rtp_name=@R000002

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000002
first_hop_ls_name = @D000002
dest_node_name = RDBOOKEE.SC30M
```

```
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------

# snaadmin path_switch, rtp_connection_name=@R000002

------------------------------------------------------------------------
path_switch command completed successfully
------------------------------------------------------------------------

# snaadmin query_rtp_connection, rtp_name=@R000002

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000002
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------
```

In Example 6-32, the first hop of @R000002 is changed from @D000002 to EELINK01 after issuing the HPR path switch command.

Verifying backup DLUS

If the DLUS has a problem, the DLUS-DLUR sessions (CPSVRMGR) pipe may be broken. If you define the backup DLUS, then the backup DLUS can takeover the DLUS function in that case. We simulate this situation with the z/OS command in SC30M:

V NET,INACT,ID=DLAIXPU1,F

Example 6-33 Status of DLUR before shutdown of PU in SC30M

```
# snaadmin status_dlur

--------------------------------------------------------------------------------
DLUR PU  LU      Status         DLUS              PLU               Description
--------------------------------------------------------------------------------
DLAIXPU1         Active         RDBOOKEE.SC30M
         DLAIX002 SSCP          RDBOOKEE.SC30M
         DLAIX003 SSCP          RDBOOKEE.SC30M
         DLAIX004 SSCP          RDBOOKEE.SC30M
         DLAIX005 SSCP          RDBOOKEE.SC30M
         DLAIX006 SSCP          RDBOOKEE.SC30M
--------------------------------------------------------------------------------

# sna -d s

      Local            Partner        Mode      Link
     LU name           LU name        name     station     State
----------------- ----------------- -------- ----------- ----------
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVRMGR @R000002    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
```

Example 6-33 shows that the DLUR-DLUS sessions are established with SC30M. Now, we shut down the PU in SC30M.

Example 6-34 Status of DLUR after shutdown of PU in SC30M

```
# snaadmin status_dlur

--------------------------------------------------------------------------------
DLUR PU  LU       Status         DLUS              PLU               Description
--------------------------------------------------------------------------------
DLAIXPU1          Active         RDBOOKEE.SC31M
         DLAIX002 SSCP           RDBOOKEE.SC31M
         DLAIX003 SSCP           RDBOOKEE.SC31M
         DLAIX004 SSCP           RDBOOKEE.SC31M
         DLAIX005 SSCP           RDBOOKEE.SC31M
         DLAIX006 SSCP           RDBOOKEE.SC31M
--------------------------------------------------------------------------------

# sna -d s

      Local             Partner        Mode      Link
     LU name            LU name        name     station     State
----------------- ----------------- -------- ----------- ----------
RDBOOKEE.EECPAIX  RDBOOKEE.SC31M    CPSVRMGR @R000006    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC31M    CPSVRMGR @R000006    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
RDBOOKEE.EECPAIX  RDBOOKEE.SC30M    CPSVCMG  @R000001    Available
```

Example 6-34 shows that SC31M becomes the active DLUS for this node. Two DLUS-DLUR sessions are now established with SC31M with new RTP connection.

6.4.3 Stopping resources

Stopping resources is very similar to starting resources. You can use various methods to stop each resource of CS/AIX. You can use the smit panel, **snaadmin** command, xsnaadmin window, and Web based administration tool.

But in most cases, **snaadmin term_node** and **stop sna** commands are used for stopping all activities of a node.

With smit, go to **smit sna** → **Manage SNA Resources** → **Stop SNA Resources** panel.

If you want to stop an individual resource by issuing the **snaadmin** command, use the following commands:

- **snaadmin deactivate_session** : deactivate the LU-LU sessions
- **snaadmin stop_internal_pu** : stopping the DLUR PU
- **snaadmin stop_ls** : stopping the link station
- **snaadmin stop_port** : stopping the port
- **snaadmin stop_dlc** : stopping the dlc

For your convenience, always make a script for activate and deactivate sessions because usually, there are a lot of sessions to be stopped in a node.

6.4.4 Backing up your configuration

The sna configuration file contains all the definitions you have done. You can find the node configuration file at: `/etc/sna/sna_node.cfg`.

If you want to make a backup file for your configurations, all you have to do is to copy this file and keep it as a backup file. You can switch among many configuration files by renaming the backup configuration file to this file name and restarting the node.

6.5 Diagnosing EE with Communications Server for AIX

CS/AIX provides logging and tracing facility. In this section, we briefly show you how to get log and trace information. The topics covered in this section are:

- "Logs in Communications server for AIX (CS/AIX)"
- "Tracing Communications server for AIX (CS/AIX)"
- "Information bundler (snagetpd)"

6.5.1 Logs in Communications server for AIX (CS/AIX)

All log files of CS/AIX are located in the /var/sna directory. The most important files in this directory are normally sna.err and bak.err files. These two files contain all the special activities on CS/AIX.

You can change the logging policy in the following smit panel:

smit sna → Problem Determination Aids → SNA Logging

You can also change the logging policy with the following commands:

```
snaadmin set_log_file
snaadmin set_log_type
```

You can set the file size of each file (sna.err, bak.err) and the types of events which are logged. For more information about logging, refer to *IBM Communications Server for AIX Diagnostics Guide*, SC31-8588.

6.5.2 Tracing Communications server for AIX (CS/AIX)

Sometimes, CS/AIX may have a problem. You can find the reason for that problem in the log files, but there are many cases which need traces. Refer to the following Web site for more information about getting traces in CS/AIX.

http://www.ibm.com/support/docview.wss?rs=1005&uid=swg21210597

The topics covered in this section are:

- "Trace CS/AIX for EE"
- "Packet Trace"

Trace CS/AIX for EE

For EE, the line traces should be captured to debug. The line traces contain all the SNA traffics through specific connectivity resources of CS/AIX such as DLCs, ports, link stations and sessions. Use the following scripts (see Example 6-35) to get the line traces in your CS/AIX.

Example 6-35 Getting line traces

```
snaadmin set_trace_file, trace_file_type=IPS, trace_file_size=10000000
snaadmin set_global_log_type, audit=yes, exception=YES
snaadmin set_global_log_type, succinct_audits=NO, succinct_errors=NO
snaadmin add_dlc_trace
snaadmin set_trace_type, trace_flags=NONE, api_flags=NONE
```

To stop the line trace, issue the **snaadmin remove_dlc_trace** command.

Packet Trace

Additionally, packet traces may be very helpful in some cases because EE uses the UDP protocol for transport. AIX provides two packet trace utilities.

- “iptrace”
- “tcpdump”

After you get the packet traces, you can open those traces with several protocol analyzers such as Wireshark. Wireshark is a good protocol analyzer based on open source projects. To get more information about Wireshark or to download Wireshark go to the following Web site:

http://www.wireshark.org

iptrace

iptrace is a utility for capturing the packet traces in AIX. Usually iptrace is started by a **startsrc** command and stopped by a **stopsrc** command. For capturing UDP data for EE see Example 6-36.

Example 6-36 Getting packet trace for EE with iptrace

```
# startsrc -s iptrace -a "-b /tmp/iptrace.bin"

0513-059 The iptrace Subsystem has been started. Subsystem PID is 487482.

...........doing what you want to trace...........

# stopsrc -s iptrace
0513-044 The iptrace Subsystem was requested to stop.
```

Example 6-36 shows how to capture the iptraces with the **iptrace** command. In the example -b flag means “bidirectional”. Always get the packet trace without a capture filter because the problem may not be on the UDP ports used by EE. Moreover, you can also filter the traces after you get them.

For more information about **iptrace** program, refer to the main page of **iptrace**.

tcpdump

tcpdump is a packet sniffing utility which is widely used. Many operating systems have this utility and AIX is not an exception. With **tcpdump**, online packet monitoring is possible. You can see the packets which pass Ethernet adapters when you get traces. It is also possible to save traces to a binary format.

Example 6-37 Online checking with tcpdump

```
# tcpdump -i en2 -n 'host SC30M-EE2.itso.ibm.com and ip proto \udp and (port 12000
 or 12001 or 12002 or 12003 or 12004)'

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on en2, link-type 1, capture size 96 bytes
14:55:49.678508 IP 10.20.4.203.12002 > 10.10.1.232.12002: udp 132
14:55:49.679095 IP 10.10.1.232.12000 > 10.20.4.203.12000: udp 3
14:55:49.679141 IP 10.10.1.232.12002 > 10.20.4.203.12002: udp 60
................................................................
^C
28 packets received by filter
0 packets dropped by kernel
```

In Example 6-37, we capture the EE traffics only. `-n` flag is used for prohibiting name resolution in traces for our presentation purpose. You do not need `-n` flag in most cases.

For a simple reachability check (as in Example 6-37), is sufficient and useful, especially if there is a firewall between the two nodes. But to debug a more complicated problem, we recommend getting traces to a binary format (see Example 6-38) and importing them to a protocol analyzer.

Example 6-38 Getting packet trace for post-processing with tcpdump

```
# tcpdump -i en2 -w /tmp/tcpdump.bin

tcpdump: listening on en2, link-type 1, capture size 96 bytes
^C
36 packets received by filter
0 packets dropped by kernel
```

For more information about the **tcpdump** program, refer to the main page of **tcpdump**.

6.5.3 Information bundler (snagetpd)

CS/AIX provides the information bundler called **snagetpd**. With the **snagetpd** program, you can gather a lot of information regarding SNA and TCP/IP, such as the output of **snaadmin** command, output of **netstat** command, sna configuration file, network options, and more.

Issue `snagetpd -q` for gathering information in CS/AIX.

6.5.4 Sample scripts to gather PD data in Communications server for AIX (CS/AIX)

We provide sample scripts to gather PD data in CS/AIX. These scripts start the line traces and IP traces, and gather system information regarding SNA. Two scripts are provided here, one for starting traces and another for stopping traces and gathering information. See Example 6-39.

Example 6-39 Script which starts the traces

```
echo "Now, starting the iptrace.\n"
```

```
startsrc -s iptrace -a "-b /tmp/iptrace.bin"
echo "Now, starting the sna line trace.\n"
snaadmin set_trace_file, trace_file_type=IPS, trace_file_size=100000000,
file_name=/tmp/trace1.trc, file_name_2=/tmp/trace2.trc                       1
snaadmin set_global_log_type, audit=yes, exception=YES
snaadmin set_global_log_type, succinct_audits=NO, succinct_errors=NO
snaadmin add_dlc_trace
snaadmin set_trace_type, trace_flags=NONE, api_flags=NONE
echo "Now, all traces are successfully started. Please regenerate problems!\n"
```

Here are explanatory note to Example 6-39:

- 1: You can change the size and name of trace files.

The script which is described in Example 6-39 starts the iptrace (packet trace) and line trace. After running this script, you should regenerate the problem situation that will be investigated. After regeneration of the problem, run the script, which stops all traces and gathers the information.

Example 6-40 Script which stops the traces and gathers information

```
echo "Now, stopping the iptraces\n"
stopsrc -s iptrace
echo "Now, gathering the sna informations\n"
snagetpd -q
echo "Now, formatting the sna traces.\n"
snatrcfmt -l -f /tmp/trace1.trc -o /tmp/trace1.fmt
snatrcfmt -l -f /tmp/trace2.trc -o /tmp/trace2.fmt
cp /tmp/trace1.trc /tmp/trace2.trc /var/sna/sna.err /var/sna/bak.err .
mv /tmp/trace1.fmt* /tmp/trace2.fmt* /tmp/iptrace.bin .
tar -cvf eepddata.tar ./iptrace.bin ./pd.tar.gz ./trace1.trc ./trace2.trc
./sna.err ./bak.err ./trace1.fmt* ./trace2.fmt*
gzip eepddata.tar
rm -f ./iptrace.bin ./pd.tar.gz ./trace1.trc ./trace2.trc ./sna.err ./bak.err
./trace1.fmt* ./trace2.fmt*
echo "The pd file eepddata.tar.gz is successfully generated.\n"
```

The script described in Example 6-40 stops all the traces and gathers information including system environment and trace data. It also formats the line trace data. After running this script, you can get a eepddata.tar.gz file which contains all the data.

7

Enabling EE in CS for Linux

This chapter describes the general configuration steps for Enterprise Extender (EE) on Linux and also provides useful information about Communications Server for Linux (CS Linux).

The topics covered in this chapter are as follows:

- "Overview of Communications Server for Linux"
- "Implementation considerations"
- "Configuring EE with CS Linux"
- "Verifying and managing EE with CS Linux"
- "Diagnosing EE with CS for Linux"

7.1 Overview of Communications Server for Linux

Communications Server for Linux (CS Linux) offers a full package of enterprise networking solutions. IBM Communications Server for Linux fully implements Advanced Peer to Peer Networking (APPN), High Performance Routing (HPR), and Dependent LU Requester (DLUR) functions. With IBM Communications Server for Linux, you can use various connectivity methods such as LLC2 and Multipath Channel (MPC), in addition to Enterprise Extender (EE).

Although Linux has IPv6 support, CS Linux currently only supports IPv4.

We do not cover the installation considerations and steps for IBM Communications Server for Linux. For more information about installing IBM Communications Server for Linux, refer to *IBM Communications Server for Linux Quick Beginnings*, GC31-6768 and *IBM Communications Server for Linux on System z Quick Beginnings*, GC31-6769.

Throughout this chapter, we assume that you have a basic knowledge of APPN. If you are not familiar with APPN, refer to 1.2.1, “Positioning APPN and HPR” on page 4.

7.2 Implementation considerations

In designing a Systems Network Architecture (SNA) network, the most important thing is availability. You have to design a reliable and fault-tolerant SNA network for your mission-critical applications. In this section, we cover the design considerations for a high availability APPN environment.

The section covers the following topics:

- “High availability considerations”
- “Scenario used in this chapter”

7.2.1 High availability considerations

With High Performance Routing (HPR), you can make a flexible and intelligent SNA network. For more information about HPR refer to 1.2.1, “Positioning APPN and HPR” on page 4. Because EE uses the HPR feature, you must consider the reliability of your APPN environment.

You must consider the following issues:

- “The number of Ethernet adapters and switches”
- “The number of connections to z/OS Logical Partitions (LPARs)”

For CS Linux on System z you can use a HiperSockets interface to connect to z/OS LPARs for EE, if the LPARs are on the same System z server.

The number of Ethernet adapters and switches

If you require high network availability, then your Linux server must have multiple Ethernet adapters installed. Though you can use VIPA interfaces to achieve this, we recommend using the link aggregation feature called *bonding*.

It is also more reliable to use two separate switches for redundancy even though load balancing mode cannot be used. Remember that load balancing is not very useful in an EE environment, it could generate out-of-order packets, which might degrade the performance of

EE. Therefore, instead of load balancing, the active/backup style of bonding is more suitable for EE.

The number of connections to z/OS Logical Partitions (LPARs)

Usually an end node (EN) has a preferred network node server (NNS) and a backup network node server. For a preferred NNS and a backup NNS, you must define a connection for each. For other nodes, you must define additional connections to each node or you can define a Connection Network to support dynamic APPN links.

We strongly recommend using a Connection Network, because it provides a more simple way to implement nodes (see “Connection Network” on page 67).

7.2.2 Scenario used in this chapter

The scenario used in this book is illustrated in Figure 7-1 on page 232. Here, SC30M, SC31M, and SC32M are z/OS VTAMs and EECPLNX is a IBM Communications Server for Linux.

We configure the CS Linux as an End Node (EN). The Control Point (CP) name will be EECPLNX. The preferred network node server of this EN is SC30M, and the backup network node server is SC31M. We define static link stations for these two network nodes because a static link station must exist for the network node server (NNS).

For connectivity to other nodes such as SC32M, we use dynamic link stations through connection network instead of using static link stations. In our scenario, the name of the connection network is RDBOOKEE.VRNLOCAL.

In general, a CS Linux EN can use a Local VRN or a Global VRN, but a CS Linux NN can only use Local VRNs.

Throughout this chapter we use host names instead of IP addresses.

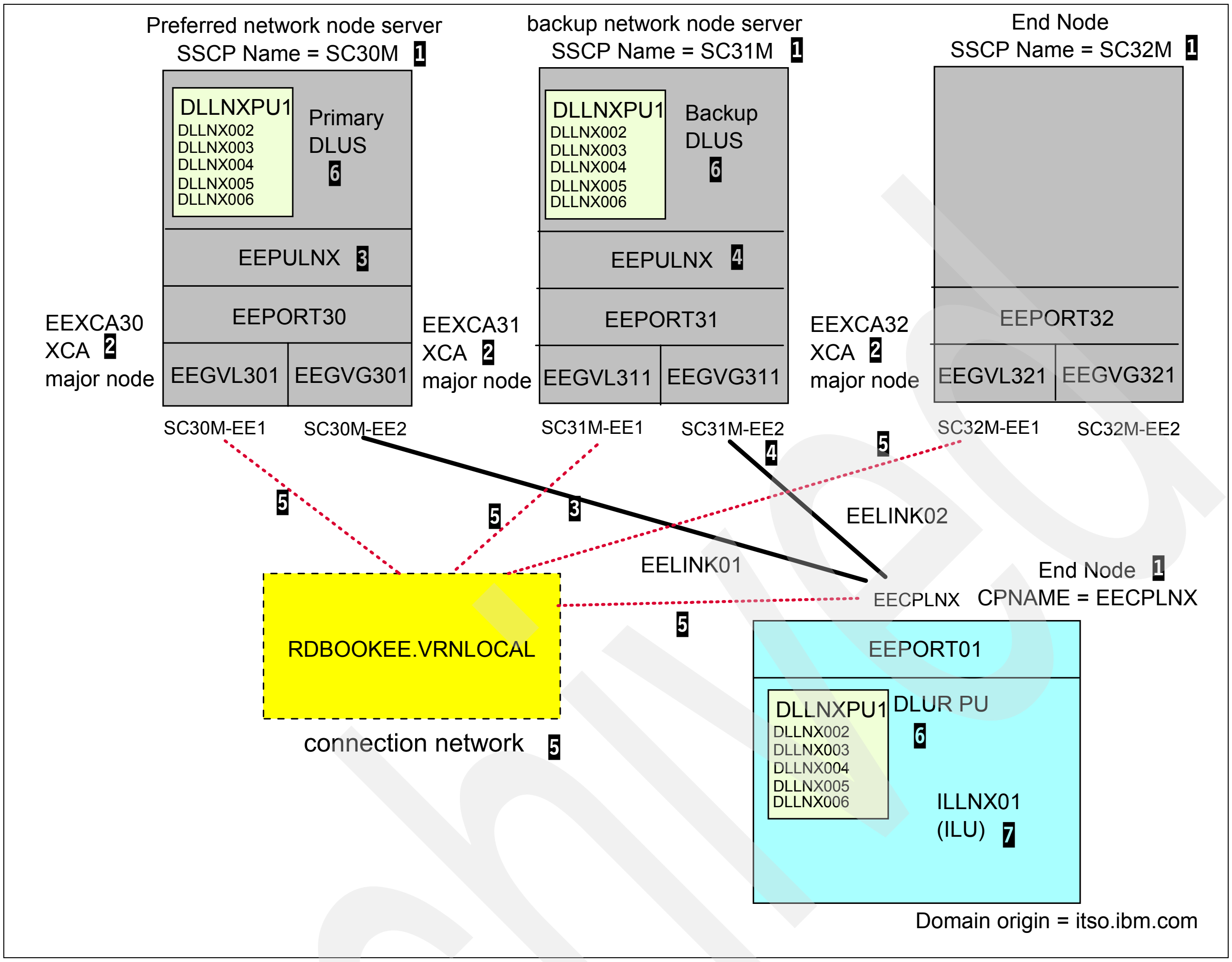


Figure 7-1 Scenario used in this chapter

Note the following explanations for Figure 7-1:

- 1: CP name of each node. For z/OS VTAM, it is an SSCP name. The preferred network node server for EECPLNX is SC30M, and the backup network node server is SC31M.
- 2: For enabling EE in z/OS VTAM, an external communications adapter (XCA) major node is required. An XCA major node consists of port and group definitions. We define two groups in an XCA major node in each system.
- 3: Static link station to SC30M (preferred network node server). We use EEGVG3x1 group for this connectivity. A PU definition for this link station should exist because DYNPU=NO for this group. We define EEPULNX for this purpose.
- 4: Static link station to SC31M (backup network node server). During an outage of the preferred network node server, this node will be used as a network node server. We use EEGVG3x1 group for this connectivity; a PU definition for this link station should exist because DYNPU=NO for this group. We define EEPULNX for this purpose.
- 5: In each node, we define a connection network, RDBOOKEE.VRNLOCAL. Because every node participates in this same connection network, dynamic link stations will be generated without any definition if needed. We use EEGVL3x1 group for this Connection Network in z/OS, and EEPORT will be used in IBM Communications Server for Linux.

- 6: For dependent LUs, the DLUS/DLUR feature should be configured. In our scenario, SC30M will be a primary DLUS, and SC31M will be a backup DLUS. You should define a type 2.0 PU and all dependent LUs in each VTAM for accepting DLUR's request. For IBM Communications Server for Linux, definition of DLUR PU and dependent LUs is required.
- 7: For independent LUs, there is nothing to be defined in z/OS if DYNLU=YES is used, otherwise a Cross Domain Resources (CDRSC) definition is required for each independent LU in VTAM.

7.3 Configuring EE with CS Linux

Configuring CS Linux includes the following:

- "Configuring node"
- "Configuring connectivity"
- "Configuring dependent LUs with DLUR"
- "Configuring independent LUs"
- "Configuring LOGMODEs"

There are four methods for configuring CS Linux:

- snaadmin command
- xsnaadmin program (X windows)
- Web-based administration tool
- Modifying the configuration file directly

We use the xsnaadmin program and the **snaadmin** command for configuring CS Linux in our scenario. To learn about other methods of configuring CS Linux refer to *IBM Communications Server for Linux Administration Guide*, SC31-6771.

You can find the snaadmin and xsnaadmin program in the following directory:

- snaadmin program: /opt/ibm/sna/bin/
- xsnaadmin program: /opt/ibm/sna/bin/X11

We advise that you add this directory to your PATH environment variable.

Important: Editing /etc/opt/ibm/sna/sna_node.cfg directly bypasses the syntax and consistency checking done by the snaadmin/xsnaadmin methods and is therefore not recommended.

If you want to modify the /etc/opt/ibm/sna/sna_node.cfg file directly, always *stop* the sna services first by **sna stop** command. IBM Communications Server for Linux has a copy of the configuration in its memory when the sna services are running. When a change is needed while sna services are running, first stop the resource whose parameter you want to change, and use the **snaadmin** command, instead of modifying the /etc/sna/sna_node.cfg file directly. Every parameter that you can see in the /etc/sna/sna_node.cfg file can be modified using the **snaadmin** command line utility, though it cannot be modified through **xsnaadmin**.

7.3.1 Configuring node

A node is a base object in an APPN environment. A node in an APPN environment is similar to a host in a TCP/IP network. So, the name of a node, that is, the Control Point (CP) name must be selected carefully.

Complete the following steps to configure the node:

1. If you start the xsnaadmin program without a configuration file, a pop-up window opens with a message stating that there is no configuration file. In such a situation activate the SNA services by issuing the **`sna start`** command and retry. A window opens asking you whether you want to use a step-by-step guide. Click no. This opens the main window of **`xsnaadmin`** as shown in Figure 7-2.

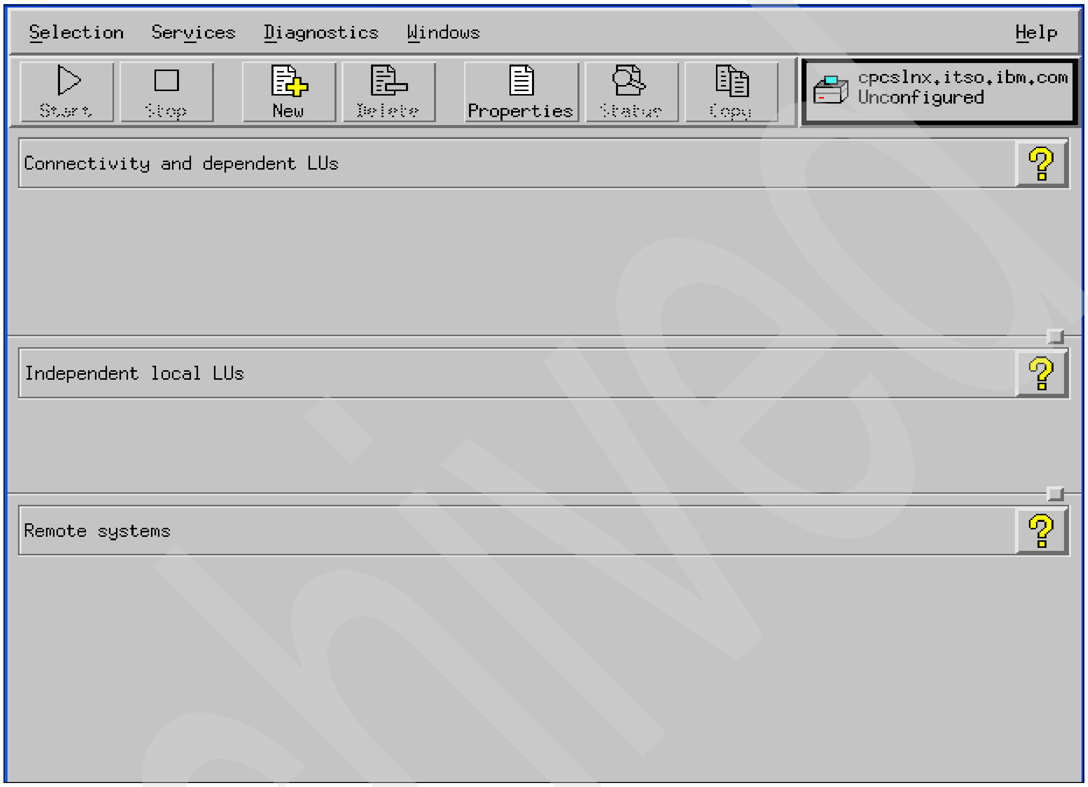
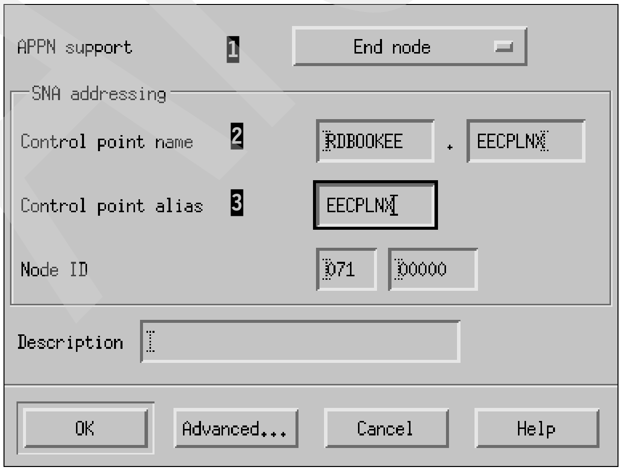


Figure 7-2 Main window of xsnaadmin

2. In the main window of xsnaadmin, you can go to the node definition window to use the Services tab.

   Select **Services** → **Configure node parameters**. See Figure 7-3.



APPN support: End node

Control point name: RDBOOKEE . EECPLNX

Control point alias: EECPLNX

Node ID: 071 00000

Description:

Figure 7-3 Configure node parameters

Note the following explanations for Figure 7-3:

- 1: Usually, CS Linux participates in an SNA network as an End node. But CS Linux can also act as a Network Node (NN) or branch network node in special cases. If it is not necessary to route SNA sessions through this server, then set this server as an End Node.
- 2: This is the fully qualified control point name. It is divided into two parts. The first part is called NETID of an SNA network, and the second part is the CP name of that server. The CP name in any SNA network must be unique. That is, in a network with the same NETID, the CP name must be unique. You must contact your z/OS administrator to get the correct NETID for your system. CP is a unique identity in an APPN environment, and represents this server in the same way that a host name stands for TCP/IP. It is advisable to give a meaningful name to facilitate maintenance. In this book, we use RDBOOKEE as the NETID.
- 3: This is just an alias of this CP. In Figure 7-3, we have defined it as EECPLNX.

Note: If a node is set to NN, it will participate in the locate search process, which uses more system resources. Because of this, it is advisable to set the node as an EN if there is no requirement for session routing.

3. Click **OK**. The default LU is automatically defined for this node, as shown in Figure 7-4. The name of the default LU is also the same as the name of the CP.

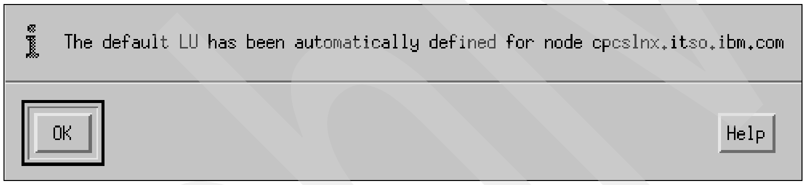

The default LU has been automatically defined for node cpcslnx.itso.ibm.com

Figure 7-4 Default LU is automatically defined

With **snaadmin** command, use:

```
snaadmin define_node, cp_alias=EECPLNX, node_type=END_NODE,
fqcp_name=RDBOOKEE.EECPLNX
```

The **snaadmin -hd** command is very useful to get the parameter list of each subcommand.

Example 7-1 shows the partial output of the **snaadmin -hd define_node** command. You can also see the mandatory fields in the example.

Example 7-1 snaadmin -hd command

```
EECPLNX:~ # snaadmin -hd define_node

This command defines a new node, or modifies an existing node.
It must be issued to a server where the node is not running.  It cannot be issued
to a running node.


Name                        Type         Length/Range
----                        ----         ------------
Mandatory fields:
cp_alias                    character    8
fqcp_name                   character    17
```

```
Defaulted fields (default in parentheses):
description                character   31  (null string)
node_type                  constant    LEN_NODE, NETWORK_NODE, END_NODE,
                                       BRANCH_NETWORK_NODE  (END_NODE)
mode_to_cos_map_supp       constant    NO, YES  (YES)
mds_supported              constant    NO, YES  (YES)
node_id                    hex         4  (0x7)
.........................................................................
```

7.3.2 Configuring connectivity

Next, we configure the connectivity resources for this node. The configuration steps will be in the following order:

1. “Configuring CS Linux for connectivity using EE”
2. “Configuring VTAM for connectivity of CS Linux using EE”

Configuring CS Linux for connectivity using EE

You must define the following resources to connect to the z/OS VTAM:

- “Port”
- “Link station”
- “Configuring connection network”

Port

The port is defined as follows:

1. You must define the Enterprise Extender Port for EE. In the main window of xsnaadmin select **Services** → **Connectivity** → **New port.**
2. In Figure 7-5, select **Enterprise Extender (HPR/IP)** port for EE and click **OK**.

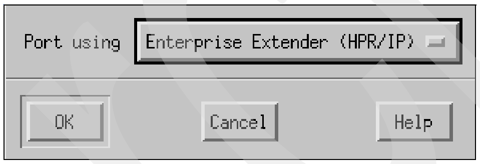
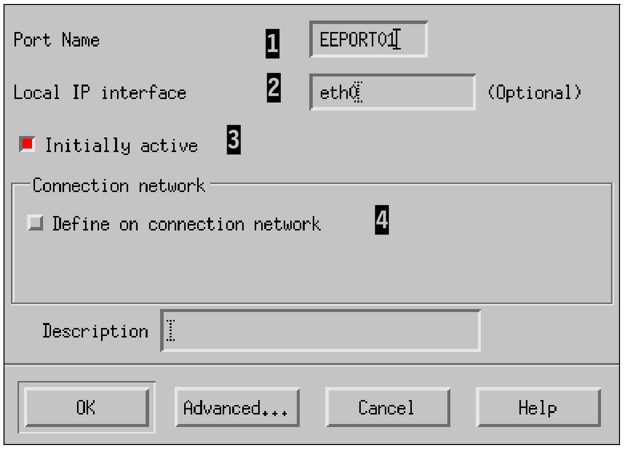

Port using Enterprise Extender (HPR/IP)

Figure 7-5 Configuring new port

This opens the window seen in Figure 7-6



Port Name: EEPORT01

Local IP interface: eth0 (Optional)

☒ Initially active

Connection network

□ Define on connection network

Description:

Figure 7-6 Configuring port

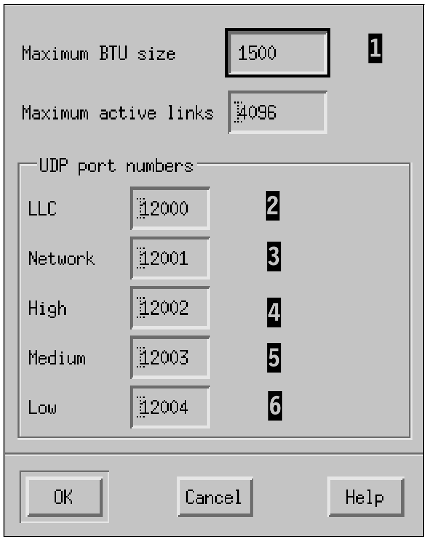
Note the following explanation for Figure 7-6:

- 1: The name of the port. It is only meaningful in this node.
- 2: The Local IP interface will determine the local IP address on which CS Linux will listen for UDP ports 12000-12004. This field should contain the VIPA or a non-default interface (if there are multiple interfaces installed). The default interface is determined by reverse-lookup of the Linux host name.

> **Note:** You must be careful with the default route of Linux. Even if you specify the interface which you want to use for EE traffics, all EE traffics might be sent through the interface which has the default route, if you do not specify any static route (when the partner node is in a different subnet from this node). Define the host route in order to be able to predict the path of EE communication if you want to use a specific interface for physical transfer.

- 3: You can start this port automatically when the node is initialized. If you select this check box, then you must start this port manually after node initialization is completed.
- 4: You can define a connection network on this port. We define the connection network later in this section.

3. Click **Advanced** to see the additional parameters on this port, shown in Figure 7-7.



Maximum BTU size 1500 1

Maximum active links 4096

UDP port numbers

LLC 12000 2

Network 12001 3

High 12002 4

Medium 12003 5

Low 12004 6

Figure 7-7 Advanced configuration on port

Note the following explanation for Figure 7-7:

- 1: Maximum BTU size is the SNA packet size including the Transmission Header (TH) and the Request/Response Header (RH). This value is negotiated when the link station is connected to the other node (when the XID is exchanged). Leave it the default value, 1500 bytes, if you do not have any problem on it. Usually, this default value is enough in most cases. But if you use NAT or IPSec across to the partner node, then you may have to decrease this maximum BTU size because of fragmentation issues. This value is overridden by the maximum BTU size of the link station which uses this port.
- 2 though 6: These UDP ports are used for various purposes and all these ports are required for EE. Be careful to change these port numbers because z/OS also must have the same definitions. The default is 12000~12004.

With the **snaadmin** command, use:

```
snaadmin define_ip_dlc, dlc_name=EEDLC, initially_active=YES
snaadmin define_ip_port, port_name=EEPORT, dlc_name=EEDLC, local_ip_interface=eth0
```

Note: With xsnaadmin, you cannot define the DLC individually. DLC is automatically defined when you define the port, and the name of the DLC will be given by xsnaadmin.

Link station

You can use the link stations as your “real connection” to the other node. With defined ports, you can make different link stations as you want. In our scenario, we make two link stations for SC30M and SC31M, so that we get one TG to SC30M and another TG to SC31M. SC30M and SC31M are connected by different methods (EE, MPC+, and XCF) so you can reach each system by using two paths (one for the direct connection, and the other for using another z/OS system). Actually, VTAM PU definition (in z/OS) and this link station definition are one-to-one match.

Proceed as follows:

1. Select **Services** → **Connectivity** → **New link station** in the main window of xsnaadmin. See Figure 7-8.

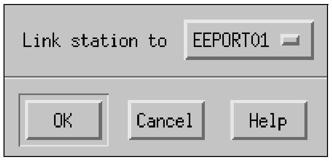
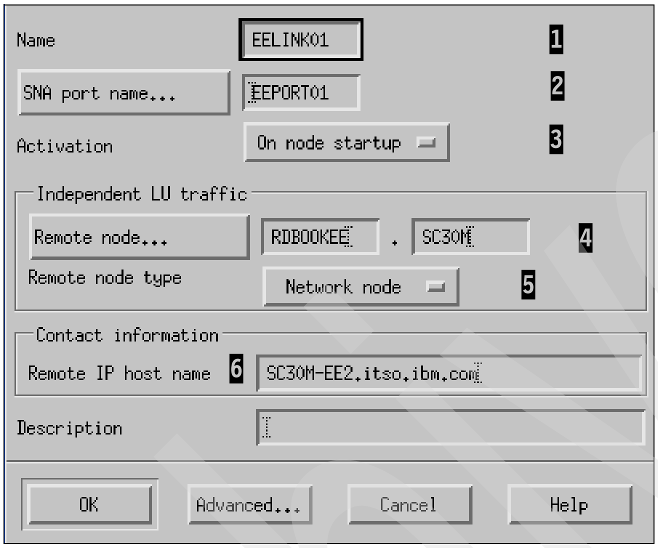

Link station to EEPORT01

Figure 7-8 Add a link station

2. Select **EEPORT01** as the port which is used by the link station, which is newly created, and click **OK.** See Figure 7-9.


Name EELINK01

SNA port name... EEPORT01

Activation On node startup

Independent LU traffic

Remote node... RDBOOKEE . SC30M

Remote node type Network node

Contact information

Remote IP host name SC30M-EE2.itso.ibm.com

Description

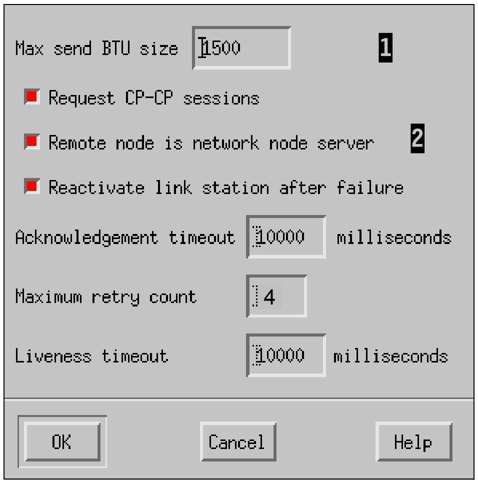
Figure 7-9 Configure the link station (EELINK01)

Note the following explanation for Figure 7-9:

- 1: This is the name of the link station. This name is meaningful only to this server.
- 2: The name of the port which is used by this link station.
- 3: There are three methods of activation:
  - BY_ADMINISTRATOR: The operator should start this link station manually.
  - ON_NODE_STARTUP: Starts automatically when the node is started.
  - ON_DEMAND: Starts when the first request is received by the link station.

    We recommend you to choose **ON_NODE_STARTUP** for your convenience.
- 4: The CP name of the remote node. Specify the fully qualified CP name (including NETID and CP name) here.
- 5: The node type of the remote host. In our scenario, the remote node is a NN.
- 6: The IP address or host name of the remote host. In our scenario, we use the host name instead of the IP address of each system.

3. Click **Advanced** to see the advanced parameters on this link station, as in Figure 7-10.



Max send BTU size: 1500 **1**

☑ Request CP-CP sessions

☑ Remote node is network node server **2**

☑ Reactivate link station after failure

Acknowledgement timeout: 10000 milliseconds

Maximum retry count: 4

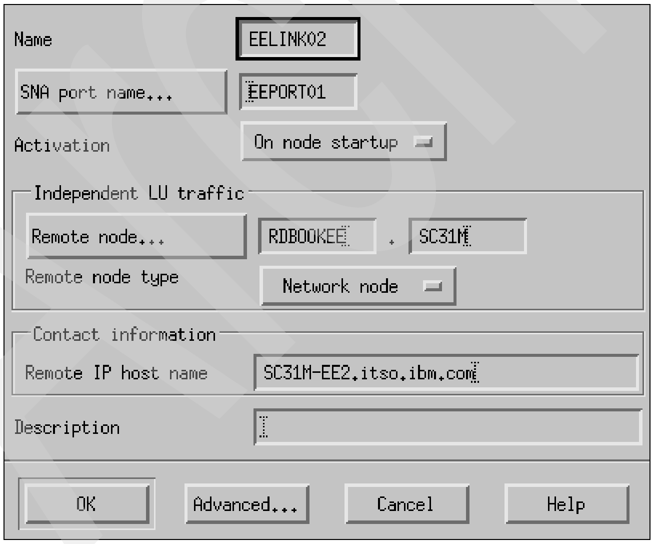
Liveness timeout: 10000 milliseconds

Figure 7-10 Advanced parameters on the link station

Note the following explanation for Figure 7-10:

- **1**: This value overrides the maximum BTU size that is configured on the port.
- **2**: You can specify this remote node as the preferred NNS. An EN can make two CP-CP sessions with only one NN at one time. This NN is called the NNS of the EN. If you do not specify which node is the NNS, then the first NN which makes CP-CP sessions with the EN is chosen to be the NNS of that EN. In our scenario, SC30M is chosen to be the preferred NNS, and SC31M as the backup network node.

4. In the same way, we configure another link station towards SC32M as shown in Figure 7-11 on page 240 and Figure 7-12 on page 241. In this link station, we selected **Remote node is network node server** check box because the remote node (SC31M) is a backup NNS, and not a preferred NNS.



Name: EELINK02

SNA port name...: EEPORT01

Activation: On node startup

Independent LU traffic

Remote node...: RDBOOKEE . SC31M

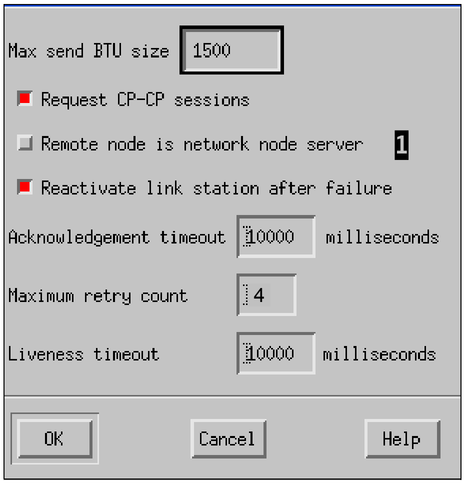
Remote node type: Network node

Contact information

Remote IP host name: SC31M-EE2.itso.ibm.com

Description:

Figure 7-11 Configure EELINK02



Max send BTU size 1500

☑ Request CP-CP sessions

☐ Remote node is network node server 1

☑ Reactivate link station after failure

Acknowledgement timeout 10000 milliseconds

Maximum retry count 4

Liveness timeout 10000 milliseconds

Figure 7-12 Advanced parameters of EELINK02

Note the following explanation for Figure 7-12:

- 1: SC31M is not a preferred NNS. But if this link station is the only one which is connected to an NN, SC31M will be the NNS for this node. Once the link station to the preferred NNS is recovered, the preferred NNS becomes the NNS immediately.

With **snaadmin** command, use:

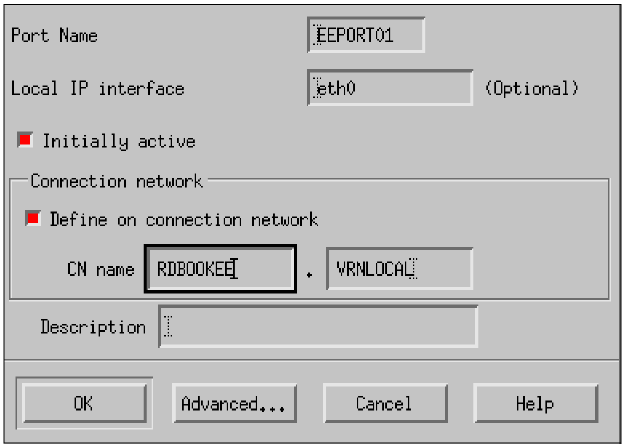
```
snaadmin define_ip_ls, ls_name=EELINK01, port_name=EEPORT,
adj_cp_type=NETWORK_NODE, default_nn_server=YES, initially_active=YES,
remote_ip_host=SC30M-EE2.itso.ibm.com, adj_cp_name=RDBOOKEE.SC30M
```

```
snaadmin define_ip_ls, ls_name=EELINK02, port_name=EEPORT,
adj_cp_type=NETWORK_NODE, initially_active=YES,
remote_ip_host=SC31M-EE2.itso.ibm.com, adj_cp_name=RDBOOKEE.SC31M
```

Configuring connection network

In our scenario, we use a connection network to communicate with other nodes except the NNS and the backup NNS, as follows:

1. When we defined the port for EE, we saw a connection network check box. Double-click the defined port (**EEPORT01**) shown in the Connectivity and dependent LUs area of the main window of xsnaadmin.
2. In Figure 7-13, we define a global connection network named RDBOOKEE.VRNLOCAL. This is the local connection network.


Port Name: EEPORT01

Local IP interface: eth0 (Optional)

☒ Initially active

Connection network

☒ Define on connection network

CN name: RDBOOKEE . VRNLOCAL

Description:

Figure 7-13 Define connection network

With **snaadmin** command, use:

```
snaadmin define_cn, fqcn_name=RDBOOKEE.VRNLOCAL, port_name=EEPORT01
```

For more information about connection network, refer to “APPN connection networks” on page 38.

Configuring VTAM for connectivity of CS Linux using EE

The APPN environment supports dynamic definitions. This means you can connect to another APPN node without node-specific definitions. VTAM controls this behavior with the DYNPU parameter in the external communications adapter (XCA) major node definition. In our scenario, we do not use dynamic PU definitions, therefore we show the VTAM definitions for the CS linux to connect to the z/OS LPARs in this section.

To configure the EE connection of CS Linux, you must define the following:

- “Enterprise Extender XCA major node”
- “PU definition for CS Linux”

Enterprise Extender XCA major node

We use two IP addresses for EE in VTAM side (with multiple VIPA). Example 7-2, shows the XCA major node of SC30M.

Note: VTAM defaults to an SAP address of 08 for the remote node when MEDIUM=HPRIP is specified in the XCA major node. However, CS Linux has a local SAP address of 04, by default. This can cause a problem when VTAM does a dial-out over a VRN to CS/AIX, therefore CALL=IN must be specified on the GROUP.

With CS Linux V6.2.2.1 or later, you can have multiple define_ip_ports with different lsap_address values and the same local_ip_interface value, which point to different define_ip_dlc definitions.

Example 7-2 EEXCA XCA major node for SC30M

```
EEXCA30         VBUILD TYPE=XCA
EEPORT30         PORT MEDIUM=HPRIP,                                    *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                 *
               IPRESOLV=2,                                             *
               SRQTIME=15,SRQRETRY=3
```

```
*
EEGVL301       GROUP DIAL=YES,CALL=IN,                                  *
               HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                         * 1
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                   * 2
               TGP=EEXTCAMP,                                            *
               KEEPACT=YES,                                             *
               DYNPU=YES,                                               * 3
               UNRCHTIM=30,                                             *
               AUTOGEN=(16,EEM30,EEN30)
*
EEGVG301       GROUP DIAL=YES,CALL=IN,                                  *
               HOSTNAME=SC30M-EE2.ITSO.IBM.COM,                         * 4
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                       * 5
               TGP=EEXTWAN,                                             *
               KEEPACT=YES,                                             *
               DYNPU=NO,                                                * 6
               UNRCHTIM=180,                                            *
               AUTOGEN=(16,EEX30,EEY30)
```

Note the following explanation for Example 7-2:

- 1: Host name for the first group.
- 2: The name of a connection network is RDBOOKEE.VRNLOCAL and its type is local. All participants of this connection network must be located within the same APPN topology subnetwork (the same NETID).
- 3: Dynamic definition of PU is allowed for this group.
- 4: Host name for the second group
- 5: The name of a connection network is W3IBMCOM.VRN and its type is global. Nodes located in different APPN topology subnetworks can share the same connection network.
- 6: Dynamic definition of PU is prohibited for this group, and a predefined PU is required.

We use the second group for connection via predefined link stations. Since dynamic PU definition is prohibited for this group, we must define a PU for IBM Communications Server for Linux node. For the connection network, we use the local connection network because IBM Communications Server for Linux node is in the same APPN subnetwork (use the same NETID) with other nodes. Using the connection network implies that DYNPU=YES, and no VTAM definition is required for IBM Communications Server for Linux node through connection network. For more information about VTAM definition and XCA major node, refer to 4.4.2, “Defining the XCA HPRIP major node” on page 142.

PU definition for CS Linux

Each system (SC30M and SC31M) should have the definition shown in Example 7-3, if the DYNPU=NO in XCA major node of VTAM.

Example 7-3 PU Definition for CS Linux

```
EESWLNXO VBUILD TYPE=SWNET
EEPULNX PU     CPNAME=EECPLNX,                           1            *
               NETID=RDBOOKEE,                           2            *
               TGN=06,                                   3            *
               CPCP=YES,                                 4            *
               HPR=YES,                                               *
               DYNLU=YES,                                5            *
```

```
                 ISTATUS=ACTIVE
```

Note the following explanation for Example 7-3:

- 1: The CP name of CS Linux. This name must be the same as the name you configured in the node definition of CS Linux.
- 2: The NETID of this SNA network.
- 3: The TG number which is used for this link. The default value of the TG number in the IBM Communications Server for Linux side is 0. With this configuration, the TG number will be decided when the XID is exchanged, and the TG number will be 6.
- 4: The CP-CP sessions capability. Set this parameter to YES.
- 5: The dynamic LU definition feature. If this parameter is set to YES, then all the independent LUs defined in CS Linux can be used in this z/OS VTAM without a CDRSC definition in VTAM. Otherwise, you must define a CDRSC entry for each independent LU.

7.3.3 Configuring dependent LUs with DLUR

We define dependent LUs and DLUR PUs in this section. For further information about the dependent LU and the DLUR, refer to “Dependent LU Requester and Server (DLUR/DLUS)” on page 33.

The following topics are covered in this section:

- “DLUR definition in IBM Communications Server for Linux”
- “DLUR definition in VTAM”

DLUR definition in IBM Communications Server for Linux

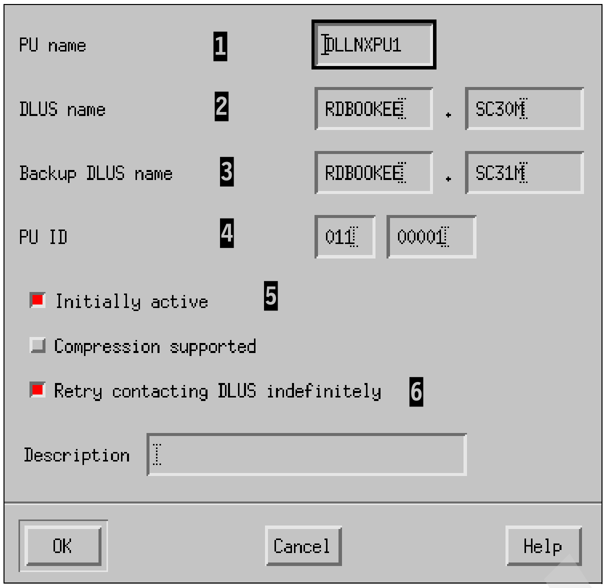
For using the dependent LUs, you must define the following in CS Linux:

- “DLUR PU”
- “Dependents LUs”

DLUR PU

DLUR PU is the type 2.0 PU which provides PU services to the dependent LUs.

To define the DLUR PU, select **Services** → **Connectivity** → **New DLUR PU** in the main window of xsnaadmin. See Figure 7-14.



PU name 1 DLLNXPU1

DLUS name 2 RDBOOKEE . SC30M

Backup DLUS name 3 RDBOOKEE . SC31M

PU ID 4 011 00001

☒ Initially active 5

□ Compression supported

☒ Retry contacting DLUS indefinitely 6

Description

Figure 7-14 Define DLUR PU

Note the following explanation for Figure 7-14:

- 1: This is the PU name for this DLUR PU for dependent LUs. This name is meaningful only in this IBM Communications Server for Linux, and you can use a different name from the PU name in the VTAM definition. But for management purposes, we recommend using the same name as your VTAM definition.
- 2: This is the SSCP name of the primary Dependent LU Server (DLUS). IBM Communications Server for Linux will try to establish a CPSVRMGR pipe (DLUS-DLUR sessions) with this DLUS. If this attempt fails, it will try to backup DLUS if it is available.
- 3: This is the SSCP name of the backup DLUS. If you use the backup DLUS, then the same PU and LU definitions must be configured in the VTAM of the backup DLUS.
- 4: This is the PU identification number. VTAM has IDBLK (ID block) and IDNUM (ID number) paramters in its PU definition for this client. These two definitions must be the same. In this case, the IDBLK is 011, and IDNUM is 00001. This value should be unique in a VTAM.
- 5: We define this PU as initially_active for convenience of management.
- 6: If the attempt to establish CPSVRMGR pipe fails, this DLUR PU will retry as configured in this parameter. We make this DLUR PU retry indefinitely when it fails to connect to DLUS.

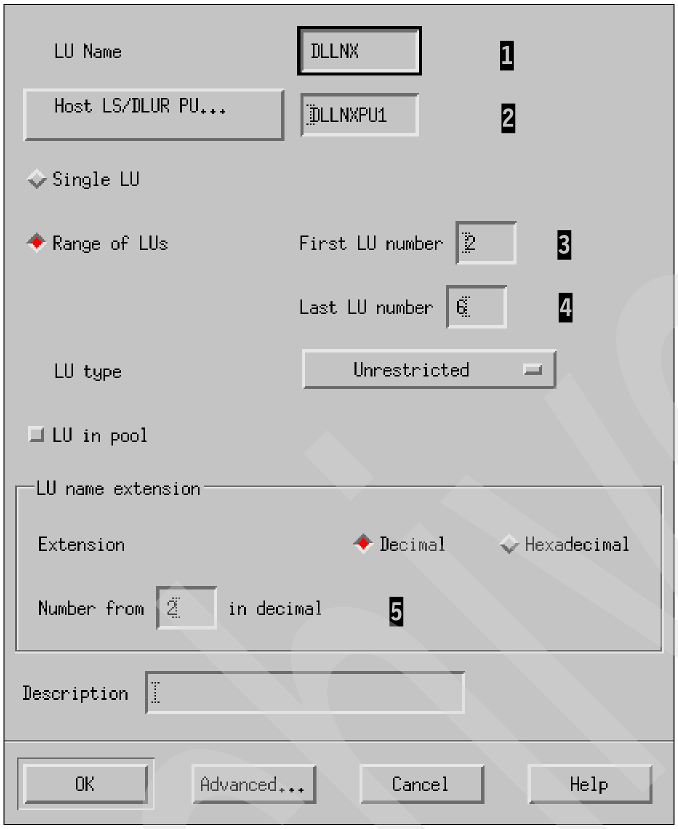
With snaadmin command, use:

```
snaadmin define_internal_pu, pu_name=DLLNXPU1, pu_id=0x01100001,
dlus_name=RDBOOKEE.SC30M, bkup_dlus_name=RDBOOKEE.SC31M, initially_active=yes,
dlus_retry_limit=65535
```

Note: You can define multiple DLUR PUs in a node. Because a PU can support only 255 dependent LUs, if you want more LUs you must define more DLUR PUs to the same primary/backup DLUS pair. Moreover, it is possible to define more DLUR PUs to the other primary/backup DLUS pair.

Dependents LUs

Here, we define the dependent LUs on the defined DLUR PU. In most cases, it is easier to define the dependent LUs using a range definition because all the properties of those LUs are the same. Select **Services** → **LUA** → **New LUA LU** in the main window of xsnaadmin. See Figure 7-15.



LU Name: DLLNX

Host LS/DLUR PU...: DLLNXPU1

○ Single LU

● Range of LUs — First LU number: 2 — Last LU number: 6

LU type: Unrestricted

□ LU in pool

LU name extension

Extension: ● Decimal ○ Hexadecimal

Number from: 2 in decimal

Description:

Figure 7-15 Define dependent LUs

Note the following explanation for Figure 7-15:

- 1: This is the base name of dependent LUs. You can use up to 6 characters here, and the names of all the dependent LUs defined here will be started with this base name.
- 2: You can assign all these dependent LUs to some link station or DLUR PU. In this case, we use the DLUR PU for these dependent LUs.
- 3: This is the local address of the first LU of this range of LUs. The local address of the LU must match the LOCADDR parameter in VTAM definition for each LU.

  Note: The LU name defined here is only meaningful for this CS Linux. In other words, there is no need to match these LU names with the LU names in the VTAM definition. Only the LU number (LOCADDR) must be the same on both the sides. But for management purpose, we recommend using the same name on both the sides.

- 4: This is the local address of the last LU of this range of LUs. The number of LUs is decided at this phase.
- 5: You can specify a number from which to start naming the LUs in the range.

With **snaadmin** command, use:

```
snaadmin define_lu_0_to_3_range, base_name=DLLNX, pu_name=DLLNXPU1, min_nau=2,
max_nau=6, base_number=2
```

DLUR definition in VTAM

In VTAM, you must define the type 2.0 PU, and the dependent LUs in a switched major node. In our scenario, we make a switched major node as shown in Example 7-4.

Example 7-4 Switched major node definition for DLUR

```
DLSWLNX  VBUILD TYPE=SWNET
DLLNXPU1 PU    PUTYPE=2,USSTAB=USSSNAEE,                                  *
               IDBLK=011,IDNUM=00001,                                     * 1
               MODETAB=ALLMODES,DLOGMOD=DYNEEHIG,ANS=CONT                   2
DLLNX002 LU    LOCADDR=2                                                    3
DLLNX003 LU    LOCADDR=3
DLLNX004 LU    LOCADDR=4
DLLNX005 LU    LOCADDR=5
DLLNX006 LU    LOCADDR=6
```

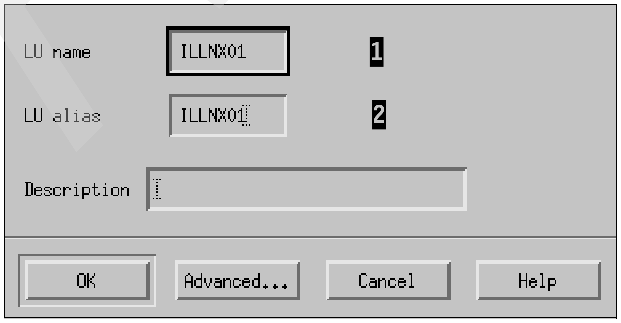
Note the following explanation for Example 7-4:

- 1: IDBLK and IDNUM must match the PU ID in the DLUR PU definition in CS Linux.
- 2: ANS=CONT is required for DLUR PUs. If the default value (ANS=STOP) is used, the SSCP-PU and SSCP-LU sessions will be broken in case the primary DLUS is lost. This results in session outages during takeover to backup DLUS.
- 3: The name of the LU may be different from that of CS Linux, but the LOCADDR must match with the LU number in DLUR PU definition of CS Linux. However, we recommend using the same names for management purposes.

7.3.4 Configuring independent LUs

An independent LU has the capability of communication with other independent LUs without the SSCP (VTAM). So, an independent LU is not a resource of VTAM, but a resource of the node in which the independent LU is defined. There are two ways of handling these independent LUs whether the DYNLU parameter in VTAM is *yes* or *no*. If DYNLU is *yes*, then no definition is required for VTAM, and VTAM will generate a dynamic definition of a CDRSC entry for that independent LU. But, if DYNLU is *no*, then there must be a predefined CDRSC entry in the VTAM for that independent LU.

Select **Services** → **APPC** → **New independent local lu** to define an independent LU. See Figure 7-16.



LU name: ILLNX01 1

LU alias: ILLNX01 2

Description:

Figure 7-16 Define an independent LU

Note the following explanation for Figure 7-16 on page 247:

- 1: This is the name of the LU. This name will be used in z/OS also.
- 2: This is the alias of the LU. This value is meaningful only to this CS Linux.

With **snaadmin** command, use:

```
snaadmin define_local_lu, lu_name=ILLNX01, lu_alias=ILLNX01, lu_session_limit=0
```

7.3.5 Configuring LOGMODEs

Sometimes, your application can use the customized LOGMODE which is not shipped by IBM. In this case you can define a customized LOGMODE in CS Linux. But, you can define logmode for type 6.2 LUs only.

1. Select **Services** → **APPC** → **Modes** in the main window of xsnaadmin. The the window is as shown in Figure 7-17.

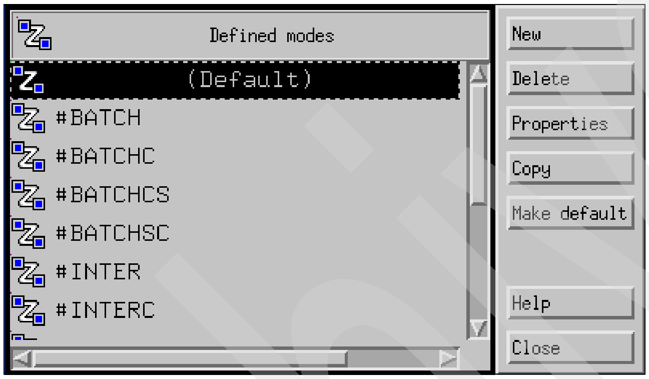
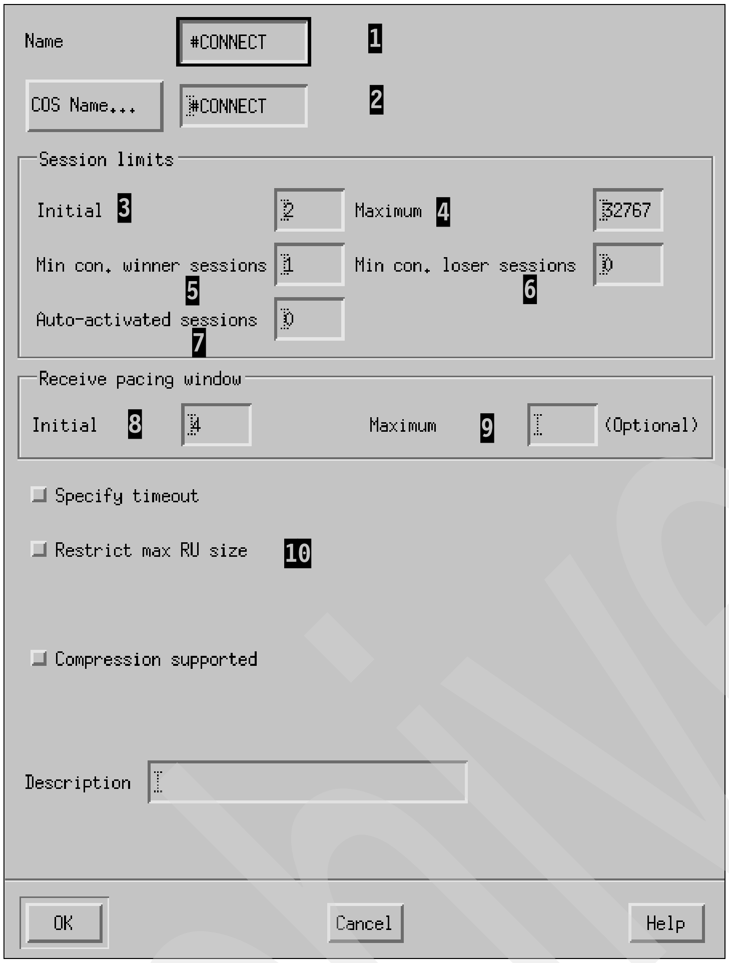


Figure 7-17 Modes

You can now see all the logmodes defined in this node.

2. Click **New** to proceed. This opens the window shown in Figure 7-18.



Name #CONNECT

COS Name... #CONNECT

Session limits

Initial 2 Maximum 32767

Min con. winner sessions 1 Min con. loser sessions 0

Auto-activated sessions 0

Receive pacing window

Initial 4 Maximum (Optional)

□ Specify timeout

□ Restrict max RU size

□ Compression supported

Description

Figure 7-18 Define a logmode

Note the following explanations for Figure 7-18:

- 1: The name of this logmode.
- 2: Name of the class of service (COS) to request when activating sessions on this mode. This must be defined in APPNCOS in VTAM.
- 3: The default session limit of this logmode. This is the maximum number of sessions permitted between a pair of LUs using this mode, even with CNOS negotiation.
- 4: The maximum number of sessions which will be supported by this logmode. In many cases this value is set to the same value as the initial session limit. Through CNOS (Change Number of Sessions), the maximum number of sessions can be changed.
- 5: The minimum contention winner sessions. The number of sessions (up to the session limit) that CS Linux must reserve for use by the local LU as the contention winner. Specify 0 if you do not want to reserve the contention winner sessions.
- 6: The minimum contention loser sessions. The minimum number of sessions that CS Linux must reserve for use by the local LU as the contention loser. The sum of the minimum contention winner sessions and the minimum contention loser sessions must not exceed the initial session limit. Specify 0 if you do not want to reserve the contention loser sessions.
- 7: You can specify the number of sessions to activate automatically for each pair of LUs that use this mode. This value is used when CNOS exchange is initiated implicitly.

- 8: The initial pacing window size. The initial number of request units (RUs) that the local LU can receive before it must send a pacing response to the remote LU.
- 9: The maximum pacing window size. The maximum number of RUs that the local LU can receive before it must send a pacing response to the remote LU. This value is optional. If it is not supplied, the maximum receive pacing window is unlimited. If a value is supplied, it is used to limit the size of the receive pacing window for adaptive pacing. If adaptive pacing is not used, this value is ignored.
- 10: You can specify whether CS Linux uses the maximum RU size upper limit and maximum RU size lower limit parameters to define the maximum RU size. If you select this check box then you can specify the upper bound for the maximum RU size and the lower limit for the maximum RU size. If you do not restrict the maximum RU size, then CS Linux sets the upper bound for the maximum RU size to the largest value that can be accommodated in the link BTU size.

In Figure 7-18, we define a new logmode named #CONNECT which uses the #CONNECT as its class of service.

7.4 Verifying and managing EE with CS Linux

In this section, we show how to start, verify, and stop the various SNA resources in CS Linux. We also show how you can backup your configuration file. The topics covered here are:

- "Starting resources"
- "Verifying resources"
- "Stopping resources"
- "Backing up your configuration"

In addition, a package called Web Administration provides the scripts and instructions for allowing remote Web browser access to administration functions for Communications Server for Linux. The Web pages of that package provide the interfaces to display link, PU, and node conditions. You also can start or stop the node, links, and PUs as needed.

For more information and to download the Web Administration package, go to:

http://www-1.ibm.com/support/docview.wss?rs=1006&uid=swg24008320

7.4.1 Starting resources

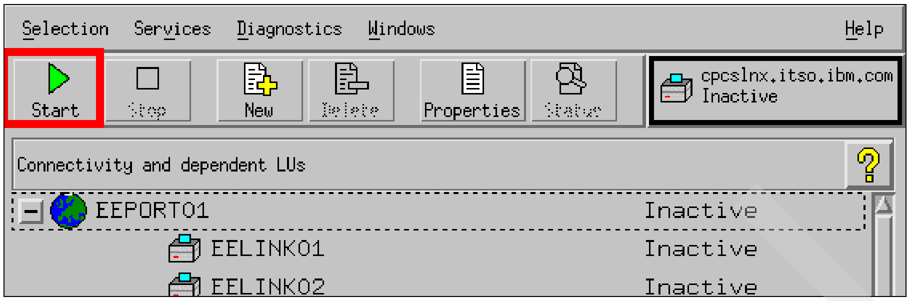
Starting each resource is very simple with xsnaadmin program. You have only to highlight the resource which you want to start, and click the start button. By default, the node is already started when the xsnaadmin program is up and running.

The topics covered here are:

- "Starting the SNA node"
- "Starting the ports, link stations and DLUR PUs"
- "Starting the sessions"

Starting the SNA node

To start the SNA service, you have only to highlight the node section of the main window of xsnaadmin and click **Start**. See Figure 7-19.



cpcslnx.itso.ibm.com
Inactive

Connectivity and dependent LUs

```
EEPORT01                    Inactive
      EELINK01              Inactive
      EELINK02              Inactive
```

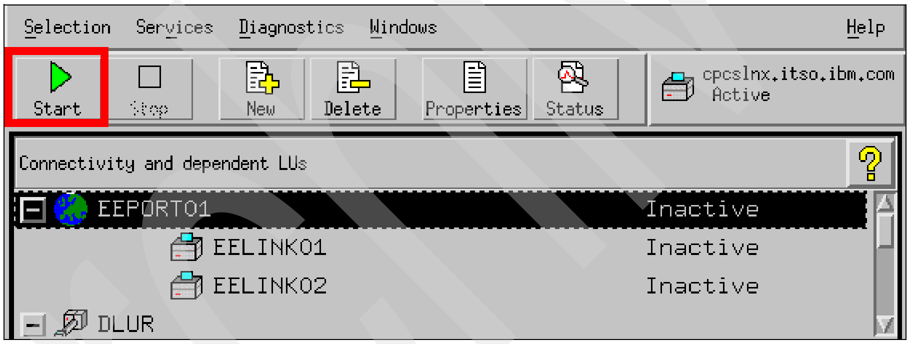
Figure 7-19 Starting the node

The box just below the help menu is for the node. If you click this node box, it is highlighted with a black border to show that the box is selected. Click **Start** to start the node**.** The node status changes to “Active” from “Inactive” in the node box.

If you have defined all the ports, link stations, and DLUR PUs as initially_active, then all the resources will be started during this node-startup.

Starting the ports, link stations and DLUR PUs

Starting each connectivity resource is also an easy task. Highlight the resource which you want to start with a single click and then click **Start**, as shown in Figure 7-20. In this case, we want to start the port, EEPORT1. The first step is to select **EEPORT1**, and then click **Start**.



cpcslnx.itso.ibm.com
Active

Connectivity and dependent LUs

```
EEPORT01                    Inactive
      EELINK01              Inactive
      EELINK02              Inactive
DLUR
```

Figure 7-20 Starting the connectivity resources

Other resources such as Link station and DLUR can be started in the same manner.

Starting the sessions

Now, all the resources are activated, and you can start the sessions that you need. Usually, making scripts for activating and deactivating sessions is a good idea for management. You can use the **`snaadmin activate_session`** command for this purpose.

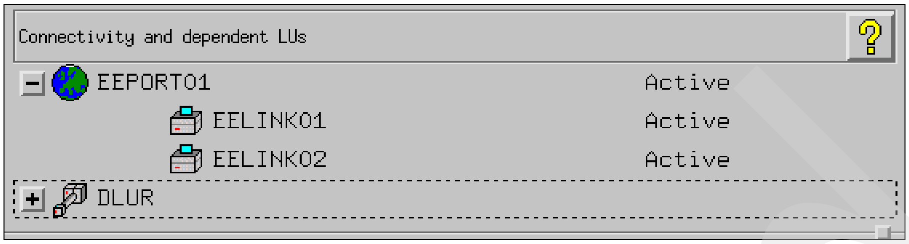
7.4.2 Verifying resources

You can easily monitor SNA activities in the xsnaadmin main window or by using the **`snaadmin`** command. In this section we describe the following:

- “Verifying the connectivity to other nodes”
- “Verifying the DLUR PUs and LUs”
- “Verifying RTP path switch”
- “Verifying backup DLUS”

Verifying the connectivity to other nodes

The Connectivity and dependent LUs area in Figure 7-21, shows the connectivity resources to other nodes.



Connectivity and dependent LUs

EEPORT01	Active
EELINK01	Active
EELINK02	Active
DLUR	

Figure 7-21 Verify the connectivity resources

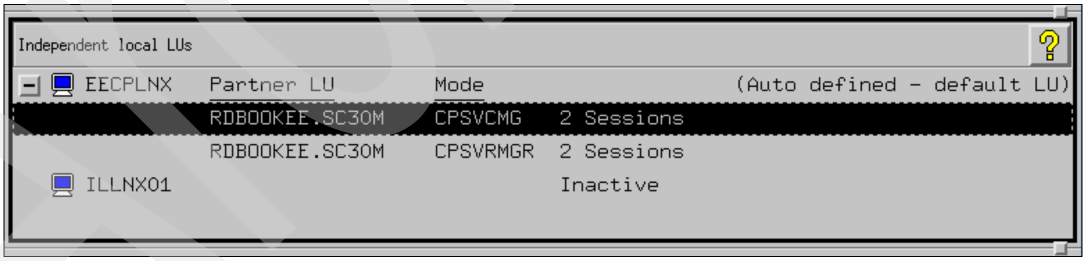
Figure 7-21, shows that the port (`EEPORT01`) and two link stations(`EELINK01n` and `EELINK02`) are in the `Active` state. But you can get the same information about connectivity with **snaadmin status_connectivity** command as shown in Example 7-5.

Example 7-5 Verify the connectivity with snaadmin command

```
cpcslnx:~ # snaadmin status_connectivity

--------------------------------------------------------------------------------
DLC       Port      LS        PU        Type Status         Description
--------------------------------------------------------------------------------
IPO                                     IP   Active
          EEPORT01                      IP   Active
                    EELINK01            IP   Active
                    EELINK02            IP   Active
--------------------------------------------------------------------------------
```

If an EN is connected to its NNS, then two CP-CP sessions are established between the two nodes. In our scenario, the preferred NNS is SC30M. With these CP-CP sessions, an EN can get topology and directory information from the NNS.



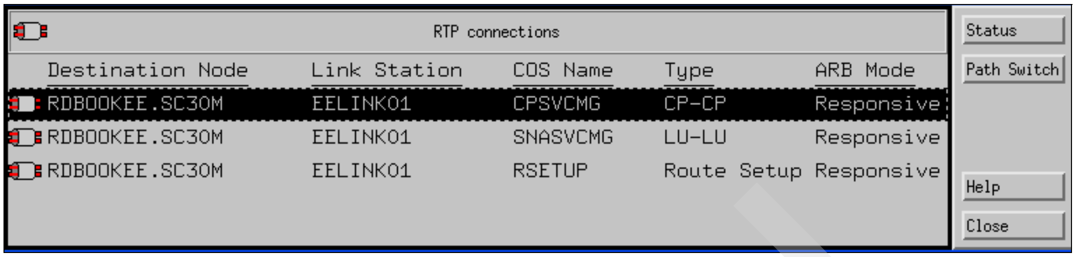
Independent local LUs

```
EECPLNX   Partner LU       Mode                 (Auto defined - default LU)
          RDBOOKEE.SC30M   CPSVCMG    2 Sessions
          RDBOOKEE.SC30M   CPSVRMGR   2 Sessions
ILLNX01                               Inactive
```

Figure 7-22 CP-CP sessions

Figure 7-22 shows the Independent local LUs area of the xsnaadmin window. The local LU, EECPLNX (default LU) has two CP-CP sessions with RDBOOKEE.SC30M LU (the Partner LU). These two sessions are CP-CP sessions because the used logmode is CPSVCMG.

When you are using EE, then all sessions are established via the RTP pipe because EE basically depends on HPR. The CP-CP sessions are also established through RTP pipe. You can see all the RTP pipes in CS Linux with xsnaadmin and snaadmin command. Select **Services** → **HPR** in your main xsnaadmin window.



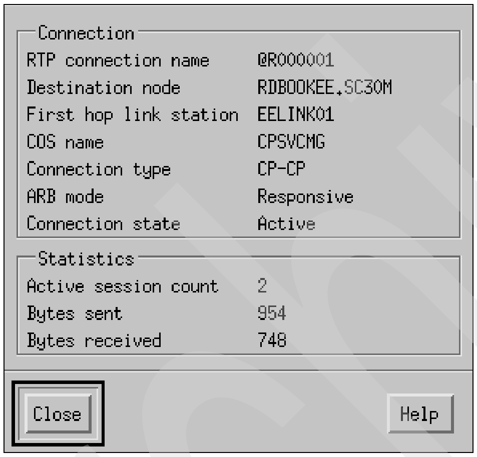
RTP connections

Destination Node	Link Station	COS Name	Type	ARB Mode
RDBOOKEE.SC30M	EELINK01	CPSVCMG	CP-CP	Responsive
RDBOOKEE.SC30M	EELINK01	SNASVCMG	LU-LU	Responsive
RDBOOKEE.SC30M	EELINK01	RSETUP	Route Setup	Responsive

Figure 7-23 Verify RTP sessions

Figure 7-23 shows three RTP pipes in this node. The highlighted pipe is for COS (Class of Service) CPSVCMG. The CP-CP sessions shown in Figure 7-22 on page 252 are established through this RTP pipe. You can also observe that EELINK01 is used for this RTP pipe. To get more information about this RTP pipe, click **Status** or double-click the appropriate RTP connection.

Figure 7-24 shows more detailed information about this RTP pipe.



Connection

RTP connection name	@R000001
Destination node	RDBOOKEE.SC30M
First hop link station	EELINK01
COS name	CPSVCMG
Connection type	CP-CP
ARB mode	Responsive
Connection state	Active

Statistics

Active session count	2
Bytes sent	954
Bytes received	748

Figure 7-24 RTP pipe for CP-CP sessions

You can also verify the RTP pipe of CP-CP sessions with the **snaadmin** command as in Example 7-6.

Example 7-6 Verify RTP Connection with snaadmin command

```
cpcslnx:/opt/ibm/sna/bin # snaadmin status_lu62

--------------------------------------------------------------------------------
LU        LU alias  Machine   Partner LU          Mode      Session Count
--------------------------------------------------------------------------------
EECPLNX   EECPLNX             RDBOOKEE.SC30M      CPSVCMG   2 Sessions
                              RDBOOKEE.SC30M      CPSVRMGR  2 Sessions
ILLNX01   ILLNX01                                           Inactive
--------------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection

------------------------------------------------------------------------
list_options = SUMMARY + FIRST_IN_LIST

rtp_name = @R000001
```

```
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2

rtp_name = @R000002
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2

rtp_name = @R000003
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = ROUTE_SETUP
cos_name = RSETUP
num_sess_active = 0
------------------------------------------------------------------------
```

For more information about HPR and RTP pipes, refer to 1.2.1, “Positioning APPN and HPR” on page 4.

Next, we check the connection network defined in Figure 7-13 on page 242. The connection network does not support the CP-CP sessions, so you must predefine link stations to the NNS and the backup NNS, if it exists. We have another z/OS LPAR in our scenario, SC32M. SC32M is connected to the same connection network, RDBOOKEE.VRNLOCAL.

Example 7-7 Verifying the connection network

```
cpcslnx:/opt/ibm/sna/bin # snaadmin status_connectivity

--------------------------------------------------------------------------------
DLC       Port      LS        PU        Type Status         Description
--------------------------------------------------------------------------------
IP0                                     IP   Active
          EEPORT01                      IP   Active
                    EELINK01            IP   Active
                    EELINK02            IP   Active
--------------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # aping RDBOOKEE.SC32M
IBM aping version 2.44 APPC echo test with timings.
Licensed Materials - Property of IBM
(C) Copyright 1994,1995 by IBM Corp. All rights reserved.

Allocate duration:       80 ms
Program startup and Confirm duration:        0 ms

Connected to a partner running on: (UNKNOWN operating system)
        Duration        Data Sent        Data Rate        Data Rate
        (msec)          (bytes)          (KB/s)           (Mb/s)
        --------        ---------        ---------        ---------
               0              200
               0              200
```

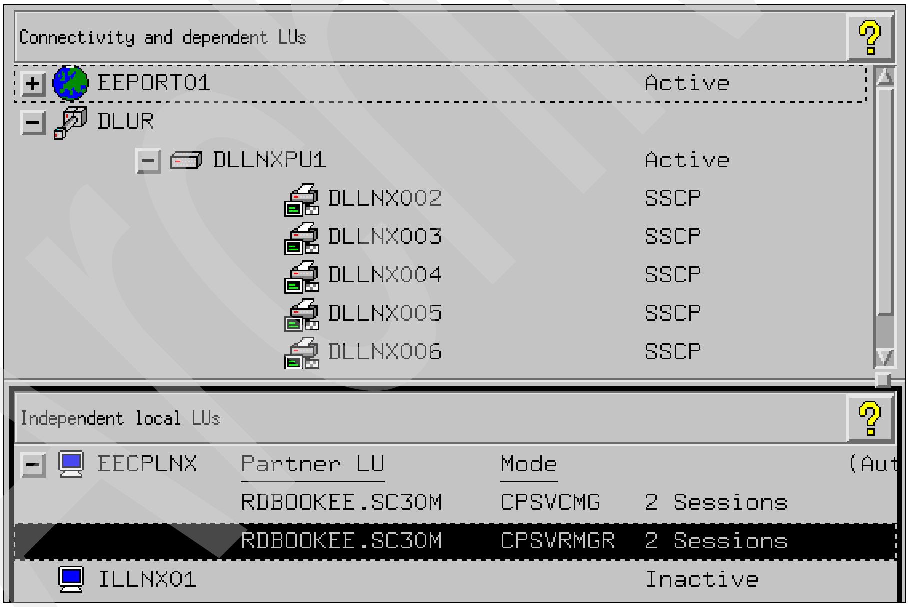
```
Totals:        0              400
Duration statistics: Min = 0 Ave = 0 Max = 0

cpcslnx:/opt/ibm/sna/bin # snaadmin status_connectivity
--------------------------------------------------------------------------------
DLC       Port      LS        PU        Type Status         Description
--------------------------------------------------------------------------------
IP0                                     IP   Active
          EEPORT01                      IP   Active
                    @D000003            IP   Dynamic        RDBOOKEE.SC32M
                    EELINK01            IP   Active
                    EELINK02            IP   Active
--------------------------------------------------------------------------------
```

See Figure 7-25. At first, we have only two link stations which are defined. After we issue an aping to SC32M, the new dynamic link station is created to SC32M. Two nodes are connected to the same connection network, so the dynamic link station can be created when it needed. This dynamic link station will disappear if there is not an existing LU-LU session which is using it.

Verifying the DLUR PUs and LUs

The DLUR PU is a type 2.0 PU and carries only the dependent LU traffics. In our scenario, there are five dependent LUs in a DLUR PU. All these dependent LU traffics are delivered through CPSVRMGR pipe. CPSVRMGR pipe is a pair of LU6.2 sessions between DLUR PU and DLUS with CPSVRMGR logmode.



```
Connectivity and dependent LUs
+ EEPORT01                              Active
- DLUR
    - DLLNXPU1                          Active
          DLLNX002                      SSCP
          DLLNX003                      SSCP
          DLLNX004                      SSCP
          DLLNX005                      SSCP
          DLLNX006                      SSCP

Independent local LUs
- EECPLNX   Partner LU       Mode                  (Aut
            RDBOOKEE.SC30M   CPSVCMG    2 Sessions
            RDBOOKEE.SC30M   CPSVRMGR   2 Sessions
  ILLNX01                               Inactive
```

Figure 7-25 DLUR PU and dependent LUs

Figure 7-25 shows that DLLNXPU1, a DLUR PU is `Active` and all the five dependent LUs are in an SSCP state. SSCP state means that the SSCP-LU session is established, but the LU-LU session is not established yet. Observe the highlighted entry in the Independent local LUs area. Here, `2 Sessions` with `CPSVRMGR` logmode are established with RDBOOKEE.SC30M (primary DLUS).

You can also verify the status of DLUR PU and dependent LUs with the **snaadmin** command, as shown in Example 7-8.

Example 7-8 Verify the DLUR PU and dependent LUs with snaadmin command

```
cpcslnx:/opt/ibm/sna/bin # snaadmin status_dlur
--------------------------------------------------------------------------------
DLUR PU  LU       Status         DLUS             PLU              Description
--------------------------------------------------------------------------------
DLLNXPU1          Active         RDBOOKEE.SC30M
         DLLNX002 SSCP           RDBOOKEE.SC30M
         DLLNX003 SSCP           RDBOOKEE.SC30M
         DLLNX004 SSCP           RDBOOKEE.SC30M
         DLLNX005 SSCP           RDBOOKEE.SC30M
         DLLNX006 SSCP           RDBOOKEE.SC30M
--------------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin status_lu62
--------------------------------------------------------------------------------
LU        LU alias  Machine   Partner LU          Mode      Session Count
--------------------------------------------------------------------------------
EECPLNX   EECPLNX             RDBOOKEE.SC30M      CPSVCMG   2 Sessions
                              RDBOOKEE.SC30M      CPSVRMGR  2 Sessions
ILLNX01   ILLNX01                                           Inactive
--------------------------------------------------------------------------------
```

Now observe the independent LU defined in the previous step. See Figure 7-25 on page 255. The ILLNX01 LU is defined but its session count is zero because no LU-LU session is activated using this LU. If it is correctly defined, we should be able to aping to the other LU with this local LU, as shown in Example 7-9.

Example 7-9 Verify the independent LU

```
cpcslnx:/opt/ibm/sna/bin # snaadmin status_lu62
--------------------------------------------------------------------------------
LU        LU alias  Machine   Partner LU          Mode      Session Count
--------------------------------------------------------------------------------
EECPLNX   EECPLNX             RDBOOKEE.SC30M      CPSVCMG   2 Sessions
                              RDBOOKEE.SC30M      CPSVRMGR  2 Sessions
ILLNX01   ILLNX01                                           Inactive
--------------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin aping, lu_name=ILLNX01,
fqplu_name=RDBOOKEE.SC30M

------------------------------------------------------------------------
alloc_time = 10
min_time = 0
avg_time = 0
max_time = 0
partner_ver_len = 0

------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin status_lu62
```

```
--------------------------------------------------------------------------------
LU        LU alias  Machine   Partner LU          Mode      Session Count
--------------------------------------------------------------------------------
EECPLNX   EECPLNX             RDBOOKEE.SC30M      CPSVCMG   2 Sessions
                              RDBOOKEE.SC30M      CPSVRMGR  2 Sessions
ILLNX01   ILLNX01             RDBOOKEE.SC30M                1 Session
                              RDBOOKEE.SC30M      SNASVCMG  1 Session
--------------------------------------------------------------------------------
```

Verifying RTP path switch

The RTP path can be changed if the topology is changed. We show this behavior briefly. See Example 7-10.

Example 7-10 RTP path switch

```
cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection, rtp_name=@R000003     1

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000003
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection, rtp_name=@R000002     2

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000002
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin stop_ls, ls_name=EELINK01                   3

------------------------------------------------------------------------
stop_ls command completed successfully
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection, rtp_name=@R000003     4

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000003
first_hop_ls_name = @D000005
dest_node_name = RDBOOKEE.SC30M
```

```
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection, rtp_name=@R000014     5

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000014
first_hop_ls_name = EELINK02
dest_node_name = RDBOOKEE.SC31M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin start_ls, ls_name=EELINK01                  6

------------------------------------------------------------------------
start_ls command completed successfully
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection, rtp_name=@R000003     7

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000003
first_hop_ls_name = @D000005
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin path_switch, rtp_connection_name=@R000003   8

------------------------------------------------------------------------
path_switch command completed successfully
------------------------------------------------------------------------

cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection, rtp_name=@R000003     9

------------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000003
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = LU_LU_SESSION
cos_name = SNASVCMG
num_sess_active = 2
------------------------------------------------------------------------
```

```
cpcslnx:/opt/ibm/sna/bin # snaadmin query_rtp_connection, rtp_name=@R000016      10

-----------------------------------------------------------------------
list_options = SUMMARY + LIST_INCLUSIVE

rtp_name = @R000016
first_hop_ls_name = EELINK01
dest_node_name = RDBOOKEE.SC30M
connection_type = CP_CP_SESSION
cos_name = CPSVCMG
num_sess_active = 2
-----------------------------------------------------------------------
```

Here are explanatory notes to Example 7-10:

- 1: This RTP pipe is for DLUR-DLUS sessions (CPSVRMGR pipe). This pipe is created to SC30M, and it uses EELINK01 as its physical path.
- 2: This RTP pipe is for CP-CP sessions. This pipe is created to SC30M, and it uses EELINK01 as its physical path
- 3: We stop the EELINK01 link station.
- 4: This RTP pipe has not disappeared, but the physical path is changed to @D000005, which is connected to SC30M. If there is no connection network, then it will be changed to EELINK02.
- 5: This RTP pipe is newly created because CP-CP sessions can be created only between adjacent nodes. Moreover, CP-CP sessions cannot be created on dynamic link created by connection network. The CP-CP sessions should be established with SC31M, the backup NNS, because the link station which is connected to the preferred NNS is now unavailable.
- 6: We re-start the EELINK01 link station.
- 7: This RTP pipe is not to be changed as there is no need to switch the RTP pipe in this case, because the path which is used now has not changed.
- 8: But we can change the path of an RTP pipe manually by command. You can also change the path of an RTP using the xsnaadmin window (see Figure 7-23 on page 253). You can find a Path Switch button in this window. If we change the path of an RTP, then the most efficient path will be chosen.
- 9: Now, the path of this RTP pipe is changed to EELINK01.
- 10: This RTP pipe is newly created because CP-CP sessions should be established to the primary NNS. In our scenario, the preferred NNS is SC30M system. So when the predefined link to the SC30M is up, then CP-CP sessions are terminated with SC31M, and established with SC30M.

Verifying backup DLUS

If the DLUS has a problem, the DLUS-DLUR sessions (CPSVRMGR) pipe may be broken. If you define the backup DLUS, then the backup DLUS can takeover the DLUS function in that case. We simulate this situation with the z/OS command in SC30M:

```
/V NET,INACT,ID=DLLNXPU1,F
```

Example 7-11 Status of DLUR before shutdown of PU in SC30M

```
EECPLNX:~ # snaadmin status_dlur

-----------------------------------------------------------------------------
```

```
DLUR PU  LU       Status         DLUS              PLU               Description
--------------------------------------------------------------------------------
DLLNXPU1          Active         RDBOOKEE.SC30M
         DLLNX002 SSCP           RDBOOKEE.SC30M
         DLLNX003 SSCP           RDBOOKEE.SC30M
         DLLNX004 SSCP           RDBOOKEE.SC30M
         DLLNX005 SSCP           RDBOOKEE.SC30M
         DLLNX006 SSCP           RDBOOKEE.SC30M
--------------------------------------------------------------------------------

EECPLNX:~ # snaadmin status_lu62

--------------------------------------------------------------------------------
LU        LU alias  Machine   Partner LU          Mode      Session Count
--------------------------------------------------------------------------------
EECPLNX   EECPLNX             RDBOOKEE.SC30M      CPSVCMG   2 Sessions
                              RDBOOKEE.SC30M      CPSVRMGR  2 Sessions
ILLNX01   ILLNX01                                           Inactive
--------------------------------------------------------------------------------
```

Example 7-11 shows that the DLUR-DLUS sessions are established with SC30M. Now, we shut down the PU in SC30M.

Example 7-12 Status of DLUR after shutdown of PU in SC30M

```
EECPLNX:~ # snaadmin status_dlur

--------------------------------------------------------------------------------
DLUR PU  LU       Status         DLUS              PLU               Description
--------------------------------------------------------------------------------
DLLNXPU1          Active         RDBOOKEE.SC31M
         DLLNX002 SSCP           RDBOOKEE.SC31M
         DLLNX003 SSCP           RDBOOKEE.SC31M
         DLLNX004 SSCP           RDBOOKEE.SC31M
         DLLNX005 SSCP           RDBOOKEE.SC31M
         DLLNX006 SSCP           RDBOOKEE.SC31M
--------------------------------------------------------------------------------

EECPLNX:~ # snaadmin status_lu62

--------------------------------------------------------------------------------
LU        LU alias  Machine   Partner LU          Mode      Session Count
--------------------------------------------------------------------------------
EECPLNX   EECPLNX             RDBOOKEE.SC30M      CPSVCMG   2 Sessions
                              RDBOOKEE.SC31M      CPSVRMGR  2 Sessions
ILLNX01   ILLNX01                                           Inactive
--------------------------------------------------------------------------------
```

Example 7-12 shows that SC31M becomes the active DLUS for this node. Two DLUS-DLUR sessions are now established with SC31M with the new RTP connection.

7.4.3 Stopping resources

Stopping resources is very similar to starting resources. You can use various methods to stop each resource of CS Linux. You can use the **snaadmin** command, the xsnaadmin window, and a Web-based administration tool.

But in most cases, `snaadmin term_node` and `stop sna` command is used for stopping all activities of a node.

In xsnaadmin, you can use the **Stop** button in the main window. For example, if you want to stop the node, select the node box and click **Stop** as shown in Figure 7-26.

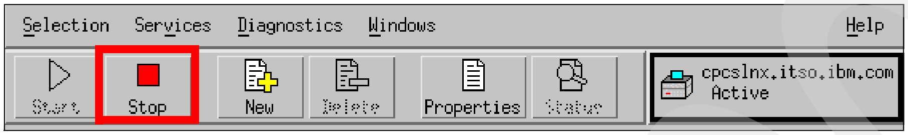


Figure 7-26 Stopping the node

If you want to stop the individual resource by using the **snaadmin** command, use the following commands:

- `snaadmin deactivate_session`: deactivate the LU-LU sessions
- `snaadmin stop_internal_pu`: stopping the DLUR PU
- `snaadmin stop_ls`: stopping the link station
- `snaadmin stop_port`: stopping the port
- `snaadmin stop_dlc`: stopping the dlc

For your convenience, always make a script for activate and deactivate sessions because usually, there are a lot of sessions to be stopped in a node.

7.4.4 Backing up your configuration

The sna configuration file contains all the definitions you have done. You can find the node configuration file at:

```
/etc/opt/ibm/sna/sna_node.cfg
```

If you want to make a backup file for your configurations, all you have to do is copy the above file and keep it as a backup file. You can switch among many configuration files by renaming the backup configuration file to the above file name and restarting the node.

7.5 Diagnosing EE with CS for Linux

IBM Communications Server for Linux provides logging and tracing facilities. In this section, we briefly explain how you can get log and trace information. The topics covered in this section are:

- “Log files in CS for Linux”
- “Tracing CS for Linux”
- “Information bundler (snagetpd)”
- “Sample scripts to gather PD data in IBM Communications Server for Linux”

7.5.1 Log files in CS for Linux

All log files of IBM Communications Server for Linux are located in the `/var/opt/ibm/sna` directory. The most important files in this directory are normally `sna.err` and `bak.err` file. These two files contain all special activities on IBM Communications Server for Linux.

With the following commands, you can set the properties of the log file:

```
snaadmin set_log_file
snaadmin set_log_type
```

You can set the file size and types of events which is logged. For more information about logging, refer to *IBM Communications Server for Linux Diagnostics Guide*, GC31-6779.

7.5.2 Tracing CS for Linux

Sometimes, IBM Communications Server for Linux may have a problem. You can identify the reason for that problem in the log files. But, there are many cases which need traces.

The topics covered in this section are:

- “Trace IBM Communications Server for Linux for EE”
- “Packet trace”
- “Information bundler (snagetpd)”

Trace IBM Communications Server for Linux for EE

For EE, the line traces should be captured to debug. The line traces contains all the SNA traffics through specific connectivity resources of IBM Communications Server for Linux such as DLCs, ports, link stations and sessions. Use the following scripts to get the line traces in your IBM Communications Server for Linux such as Example 7-13.

Example 7-13 getting line traces

```
snaadmin set_trace_file, trace_file_type=IPS, trace_file_size=10000000
snaadmin set_global_log_type, audit=yes, exception=YES
snaadmin set_global_log_type, succinct_audits=NO, succinct_errors=NO
snaadmin add_dlc_trace
snaadmin set_trace_type, trace_flags=NONE, api_flags=NONE
```

With the **`snaadmin set_trace_file`** command shown in Example 7-13, you can change the properties of the trace files such as file size and file name.

To stop the line trace, issue the **`snaadmin remove_dlc_trace`** command.

Packet trace

Additionally, packet traces may be very helpful in some cases because EE uses the UDP protocol for transport. Linux provides the tcpdump utility for packet tracing.

After you get the packet traces, you can open those traces with several protocol analyzers such as Wireshark. Wireshark is a good protocol analyzer based on open source projects. You can get more information or download Wireshark from the following Web site:

http://www.wireshark.org

tcpdump

tcpdump is a packet sniffing utility that is used widely. Many operating systems have this utility. With tcpdump, online packet monitoring is possible. You can see the packets which pass Ethernet adapters when you get traces. Also, it is also possible to save traces to a binary format.

Example 7-14 Online checking with tcpdump

```
EECPLNX:/var/opt/ibm/sna # tcpdump -i eth0 -n 'host SC30M-EE2.itso.ibm.com and ip
proto \udp and (port 12000 or 12001 or 12002 or 12003 or 12004)'

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
00:42:20.061436 IP 10.20.4.202.12000 > 10.10.1.232.12000: UDP, length 3
00:42:20.061840 IP 10.10.1.232.12000 > 10.20.4.202.12000: UDP, length 3
00:42:30.062044 IP 10.20.4.202.12000 > 10.10.1.232.12000: UDP, length 3
00:42:30.062509 IP 10.10.1.232.12000 > 10.20.4.202.12000: UDP, length 3

4 packets captured
8 packets received by filter
0 packets dropped by kernel
```

In Example 7-14, we capture the EE traffics only. `-n` flag is used for prohibiting name resolution in traces for our presentation purpose. You do not need `-n` flag in most cases.

For a simple reachability check, eye-checking as shown in Example 7-14 is sufficient and useful, especially if there is a firewall between the two nodes. But to debug a more complicated problem, we recommend getting traces (see Example 7-15) to a binary format and importing to a protocol analyzer.

Example 7-15 Getting packet trace for post-processing with tcpdump

```
EECPLNX:/var/opt/ibm/sna # tcpdump -i eth0 -w /tmp/tcpdump.bin

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
7 packets captured
14 packets received by filter
0 packets dropped by kernel
```

For more information about the tcpdump program, refer to the tcpdump man page.

7.5.3 Information bundler (snagetpd)

IBM Communications Server for Linux provides the information bundler called snagetpd. With the snagetpd program, you can gather a lot of information regarding SNA and TCP/IP such as the output of the **snaadmin** command, output of the **netstat** command, sna configuration file, network options, and more.

Issue snagetpd -q to gather information in IBM Communications Server for Linux.

7.5.4 Sample scripts to gather PD data in IBM Communications Server for Linux

We provide sample scripts to gather PD data in IBM Communications Server for Linux. These scripts starts the line traces and IP traces, and gather system information regarding SNA. Two scripts are provided in Example 7-16, one for starting traces and one for stopping traces and gathering information.

Example 7-16 Script which starts the traces

```
echo "Now, starting the tcpdump."
tcpdump -i any -w /tmp/tcpdump.dmp &
echo "Now, starting the sna line trace."
snaadmin set_trace_file, trace_file_type=IPS, trace_file_size=100000000,
file_name=/tmp/trace1.trc, file_name_2=/tmp/trace2.trc                        1
snaadmin set_global_log_type, audit=yes, exception=YES
snaadmin set_global_log_type, succinct_audits=NO, succinct_errors=NO
snaadmin add_dlc_trace
snaadmin set_trace_type, trace_flags=NONE, api_flags=NONE
echo "Now, all traces are successfully started. Please regenerate problems!"
```

Here is an explanatory note to Example 7-16:

- 1: You can change the size and name of the trace files.

The script which is described in Example 7-16 starts the packet trace and line trace. After running this script, you should regenerate the problem situation which will be investigated. After regeneration of the problem, run the script, which stops all traces and gathers the information.

Example 7-17 Script which stops the traces and gathers information

```
echo "Now, stopping the tcpdump"
kill -9 `ps -aef | grep -v grep | grep tcpdump | awk '{print $2}'`
echo "Now, gathering the sna informations"
snagetpd -q
echo "Now, formatting the sna traces."
snatrcfmt -l -f /tmp/trace1.trc -o /tmp/trace1.fmt
snatrcfmt -l -f /tmp/trace2.trc -o /tmp/trace2.fmt
cp /tmp/trace1.trc /tmp/trace2.trc /var/opt/ibm/sna/sna.err
/var/opt/ibm/sna/bak.err .
mv /tmp/trace1.fmt* /tmp/trace2.fmt* /tmp/tcpdump.dmp .
tar -cvf eepddata.tar ./tcpdump.dmp ./pd.tar.gz ./trace1.trc ./trace2.trc
./sna.err ./bak.err ./trace1.fmt* ./trace2.fmt*
gzip eepddata.tar
rm -f ./tcpdump.dmp ./pd.tar.gz ./trace1.trc ./trace2.trc ./sna.err ./bak.err
./trace1.fmt* ./trace2.fmt*
echo "The pd file eepddata.tar.gz is successfully generated."
```

The script described in Example 7-17 stops all the traces and gathers information, including system environments and trace data. It also formats the line trace data. After running this script, you can get a eepddata.tar.gz file which contains all the data.

8

Enabling EE in CS for Windows

In this chapter, we describe the general configuration steps for Enterprise Extender (EE) on Windows and provide useful information about Communications Server for Windows.

The topics covered in this chapter are:

- “Overview of Communications Server for Windows”
- “Implementation considerations”
- “Configuring EE with CS for Windows”
- “Verifying and managing EE with CS for Windows”
- “Diagnosing EE with Communications Server for Windows”

8.1 Overview of Communications Server for Windows

Communications Server for Windows (CS Windows) offers a full package of enterprise networking solutions. CS Windows fully implements Advanced Peer to Peer Networking (APPN), High Performance Routing (HPR), and Dependent LU Requester (DLUR) functions. With CS Windows, you can use various connectivity methods such as LLC2, Multipath Channel (MPC), in addition to Enterprise Extender (EE). We do not cover the installation considerations and steps of IBM Communications Server for Windows in this book. If you want more information about installation of IBM Communications Server for Windows,refer to *IBM Communications Server for Windows Quick Beginnings*, GC31-8424.

Throughout this chapter, we assume that you have a basic knowledge of APPN. If you are not familiar with APPN, refer to 1.2.1, “Positioning APPN and HPR” on page 4.

8.2 Implementation considerations

In designing a Systems Network Architecture (SNA) network, the most important thing is availability. You have to design a reliable and fault-tolerant SNA network for your mission-critical applications. In this section, we cover the design considerations for a high availability APPN environment.

The topics covered in this section are:

- “High availability considerations”
- “Scenario used in this chapter”

8.2.1 High availability considerations

With HPR, you can make a flexible and intelligent SNA network. For more information about HPR, refer to 1.2.1, “Positioning APPN and HPR” on page 4. Because EE uses the HPR feature, you must consider the reliability of your APPN environment.

You have to consider the following issues:

- “The number of Ethernet adapters and switches”
- “The number of connections to z/OS Logical Partitions (LPARs)”

The number of Ethernet adapters and switches

If you require high network availability, then your Windows server must have multiple Ethernet adapters installed. We recommend using the link aggregation feature called *teaming*.

It is also more reliable to use two separate switches for redundancy, even though load balancing mode cannot be used. Remember that load balancing is not very useful in an EE environment, it could generate out-of-order packets, which might degrade the performance of EE. Therefore, instead of load balancing, the active/backup style of teaming is more suitable for EE.

The number of connections to z/OS Logical Partitions (LPARs)

Usually an end node (EN) has a preferred network node server (NNS) and a backup network node server. For a preferred NNS and a backup NNS, you must define a connection for each. For other nodes, you must define additional connections to each node or you can define a Connection Network to support dynamic APPN links.

We strongly recommend using a Connection Network, because it provides a more simple way to implement nodes (see “Connection Network” on page 67).

8.2.2 Scenario used in this chapter

Figure 8-1 illustrates the scenario used in this book. Here, SC30M, SC31M and SC32M are z/OS VTAMs and EECPWIN is a IBM Communications Server for Windows.

We configure the IBM Communications Server for Windows as an End Node (EN). The Control Point (CP) name will be EECPWIN. The preferred network node server of this EN is SC30M, and the backup network node server is SC31M. We define static link stations for these two network nodes because a static link station should be exist for a network node server (NNS).

For connectivity to other nodes such as SC32M, we use dynamic link stations through connection network instead of using static link stations. In our scenario, the name of the connection network is RDBOOKEE.VRNLOCAL.

Throughout this chapter, we use host names instead of IP addresses.

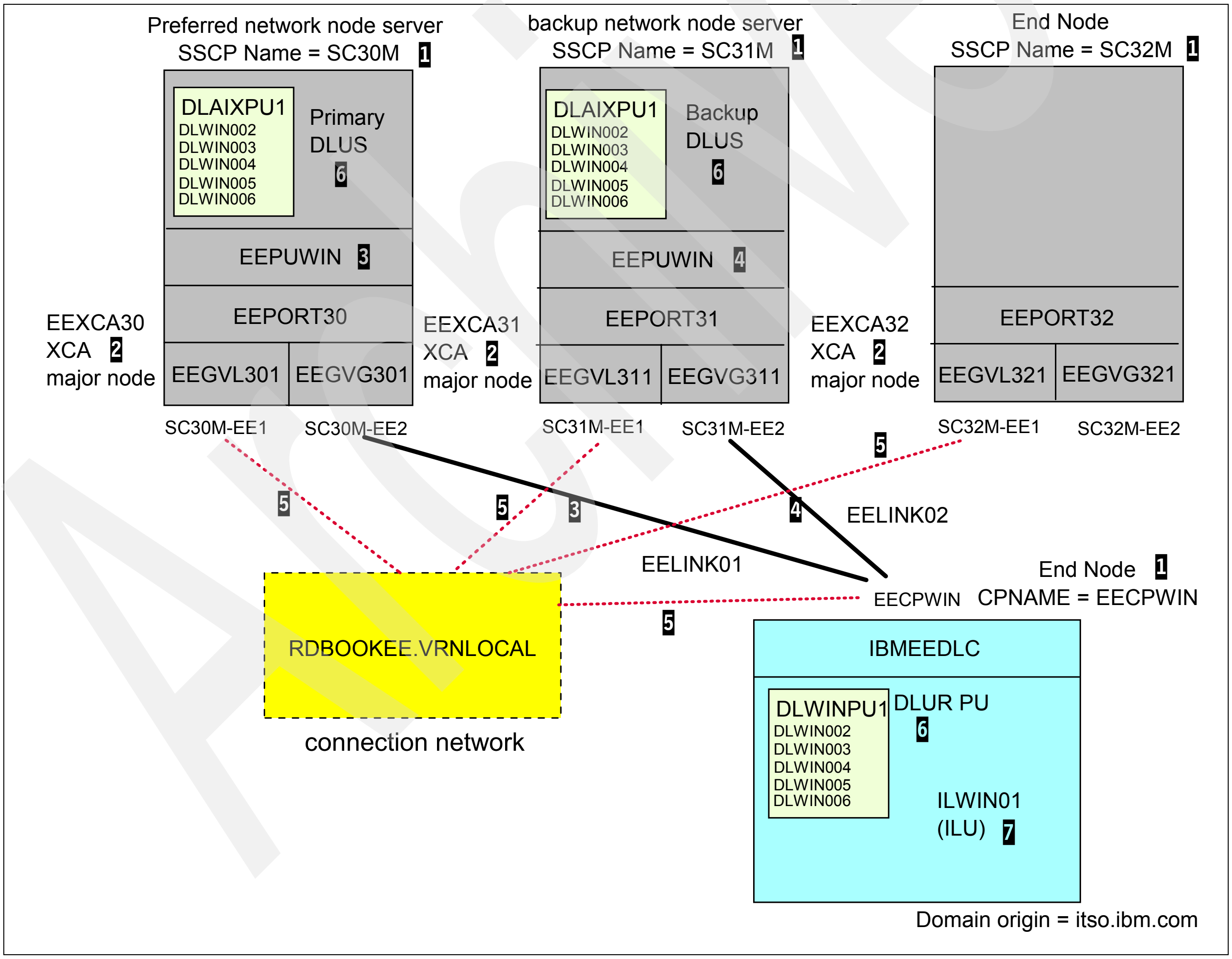


Figure 8-1 Scenario used in this chapter

Note the following explanation for Figure 8-1:

- 1: CP name of each node. For z/OS VTAM, it is an SSCP name. The preferred network node server for EECPWIN is SC30M, and the backup network node server is SC31M.
- 2: For enabling EE in z/OS VTAM, an external communications adapter (XCA) major node is required. An XCA major node consists of port and group definitions. We define two groups in an XCA major node in each system.
- 3: Static link station to SC30M (preferred network node server). We use EEGVG3x1 group for this connectivity. APU definition for this link station should exist because DYNPU=NO for this group. We define EEPUWIN for this purpose.
- 4: Static link station to SC31M (backup network node server). During outage of preferred network node server, this node will be used as a network node server. We use EEGVG3x1 group for this connectivity; PU definition for this link station should exist because DYNPU=NO for this group. We define EEPUWIN for this purpose.
- 5: In each node, we define a connection network, RDBOOKEE.VRNLOCAL. Because every node participates in this same connection network, dynamic link stations will be generated without any definition if needed. We use EEGVL3x1 group for this connection network in z/OS, and EEPORT will be used in IBM Communications Server for Windows.
- 6: For dependent LUs, the DLUS/DLUR feature should be configured. In our scenario, SC30M will be a primary DLUS, and SC31M will be a backup DLUS. You should define a type 2.0 PU and all dependent LUs in each VTAM for accepting DLUR's request. For IBM Communications Server for Windows, definition of DLUR PU and dependent LUs is required.
- 7: For independent LUs, there is nothing to be defined in z/OS if DYNLU=YES is used; otherwise a CDRSC definition is required for each independent LU in VTAM.

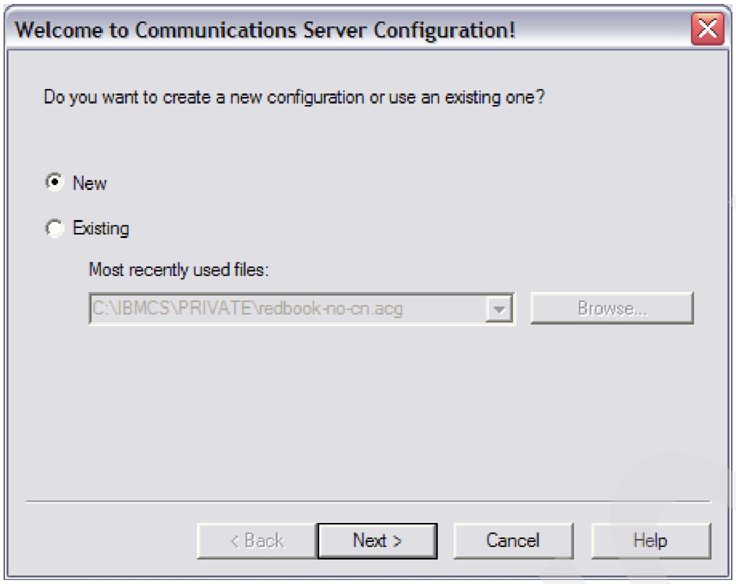
8.3 Configuring EE with CS for Windows

Configuring CS Windows includes the following:

- "Configuring node"
- "Configuring connectivity"
- "Configuring dependent LUs with DLUR"
- "Configuring independent LUs"
- "Configuring LOGMODEs"

We can configure the IBM Communications Server for Windows in two ways. The common way to configure the IBM Communications Server for Windows is using the GUI tool for configuration. You can also configure the IBM Communications Server for Windows by modifying the configuration file directly. Here, we use the GUI method for configuring our scenario environment.

For creating the new configuration file, start the SNA Node Configuration by clicking **Start → All programs → IBM Communications Server → SNA Node Configuration**.



Welcome to Communications Server Configuration!

Do you want to create a new configuration or use an existing one?

● New

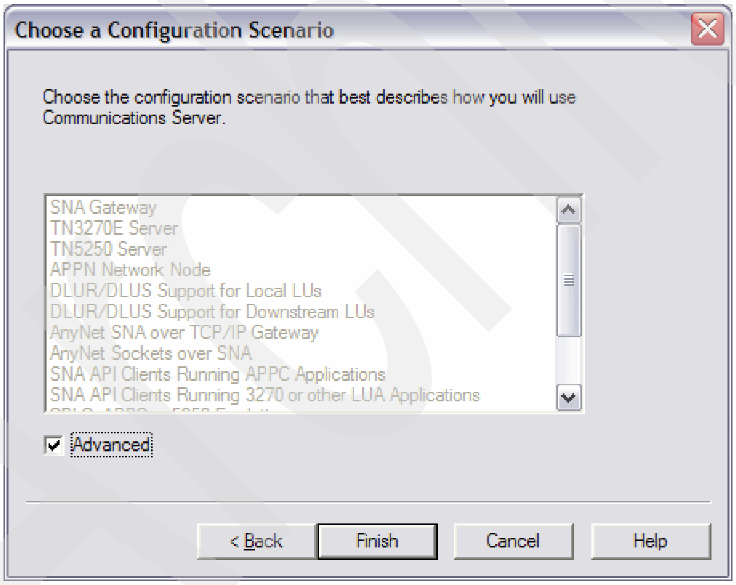
○ Existing

Most recently used files:

C:\IBMCS\PRIVATE\redbook-no-cn.acg

Figure 8-2 Creating new configuration file

In Figure 8-2, select **New** and click **Next**. This opens the Choose a Configuration Scenario window, as shown in Figure 8-3. In this window, select the **Advanced** check box, and then click **Finish**.



Choose a Configuration Scenario

Choose the configuration scenario that best describes how you will use Communications Server.

SNA Gateway
TN3270E Server
TN5250 Server
APPN Network Node
DLUR/DLUS Support for Local LUs
DLUR/DLUS Support for Downstream LUs
AnyNet SNA over TCP/IP Gateway
AnyNet Sockets over SNA
SNA API Clients Running APPC Applications
SNA API Clients Running 3270 or other LUA Applications

☑ Advanced

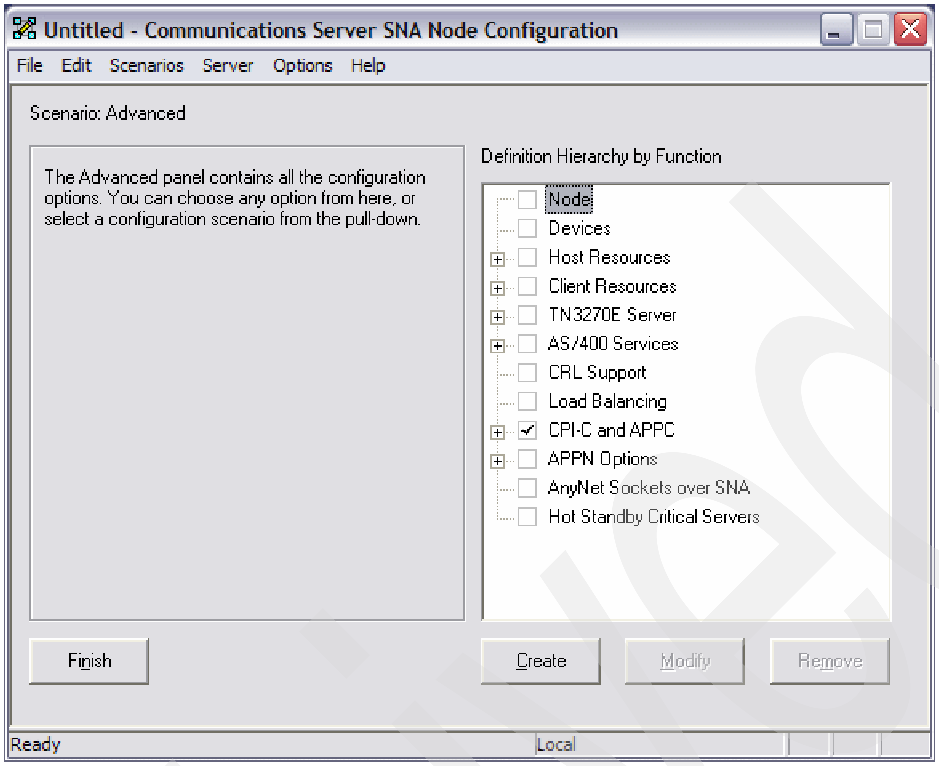
Figure 8-3 Choose a Configuration Scenario

This opens the main window of SNA node configuration, as shown in Figure 8-4.

8.3.1 Configuring node

The node is a base object in an APPN environment. A node in an APPN environment is similar to a host in a TCP/IP network.

In Figure 8-4, under the Definition Hierarchy by Function area double-click **Node**.


Untitled - Communications Server SNA Node Configuration
File Edit Scenarios Server Options Help
Finish
Create
Modify
Remove

Scenario: Advanced

The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

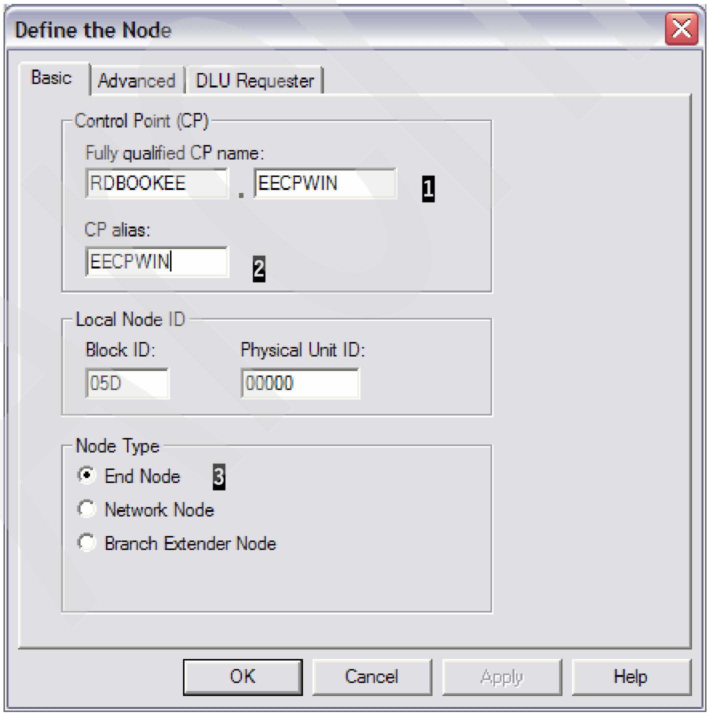
Definition Hierarchy by Function

- □ Node
- □ Devices
- □ Host Resources
- □ Client Resources
- □ TN3270E Server
- □ AS/400 Services
- □ CRL Support
- □ Load Balancing
- ☑ CPI-C and APPC
- □ APPN Options
- □ AnyNet Sockets over SNA
- □ Hot Standby Critical Servers

Ready Local

Figure 8-4 SNA Node Configuration main window

This opens Define the Node window as shown in Figure 8-5.



Control Point (CP)

Fully qualified CP name: RDBOOKEE . EECPWIN

CP alias: EECPWIN

Local Node ID

Block ID: 05D Physical Unit ID: 00000

Node Type

- ● End Node
- ○ Network Node
- ○ Branch Extender Node

Figure 8-5 Configuring node parameters

Note the following explanation for Figure 8-5:

- 1: This is the fully qualified control point (CP) name. It is divided into two parts. The first part is called NETID of an SNA network, and the second part is the CP name of that server. The CP name must be unique in one SNA network, that is, in a network with the same NETID, the CP name must be unique. You must contact the z/OS administrator to get the correct NETID for your system. Control point is a unique identity in an APPN environment, and represents this server in the same way as a host name is for TCP/IP. Give a meaningful name for maintenance purpose. In this book, we use `RDBOOKEE` as a NETID.
- 2: This is just an alias of this control point.
- 3: Usually, IBM Communications Server for Windows participates in an SNA network as an End Node (EN). But IBM Communications Server for Windows can also act as a Network Node (NN) or a branch network node in special cases. If there is no need to route SNA sessions through this server, then set this server as an EN.

> **Note:** If a node is set to Network Node (NN), it will participate in the locate search process; this will use more system resources. Set the node as an End Node (EN) if there is no requirement for session routing.

In Figure 8-5, we configure this node as an **END Node**, and the fully qualified CP name is `RDBOOKEE.EECPWIN`.

8.3.2 Configuring connectivity

Now, we configure the connectivity resources for this node in the following order:

- "Configuring CS Windows for connectivity using EE"
- "Configuring VTAM for connectivity using EE"

Configuring CS Windows for connectivity using EE

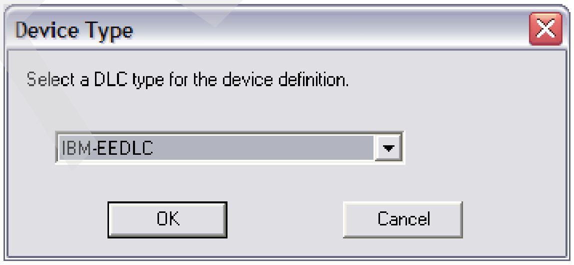
You should define the following resources to connect to the z/OS VTAM:

- "Device"
- "Peer Connection"
- "Configuring connection network"

Device

Define the device for EE by completing the following steps:

1. In Figure 8-4 on page 270, under the Definition Hierarchy by Function area double-click **Devices.** This opens the Device Type window as shown in Figure 8-6.



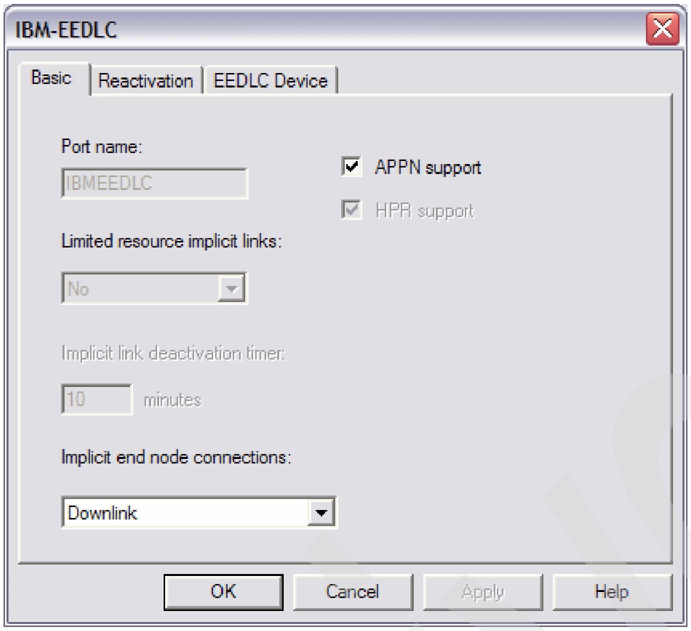
Device Type

Select a DLC type for the device definition.

IBM-EEDLC

Figure 8-6 Selecting a DLC type for EE

2. Select IBM-EEDLC, and then click **OK**. This opens the IBM_EEDLC window as shown in Figure 8-7.



Port name: IBMEEDLC

☑ APPN support

☑ HPR support

Limited resource implicit links: No

Implicit link deactivation timer: 10 minutes

Implicit end node connections: Downlink

Figure 8-7 Configuring properties of port

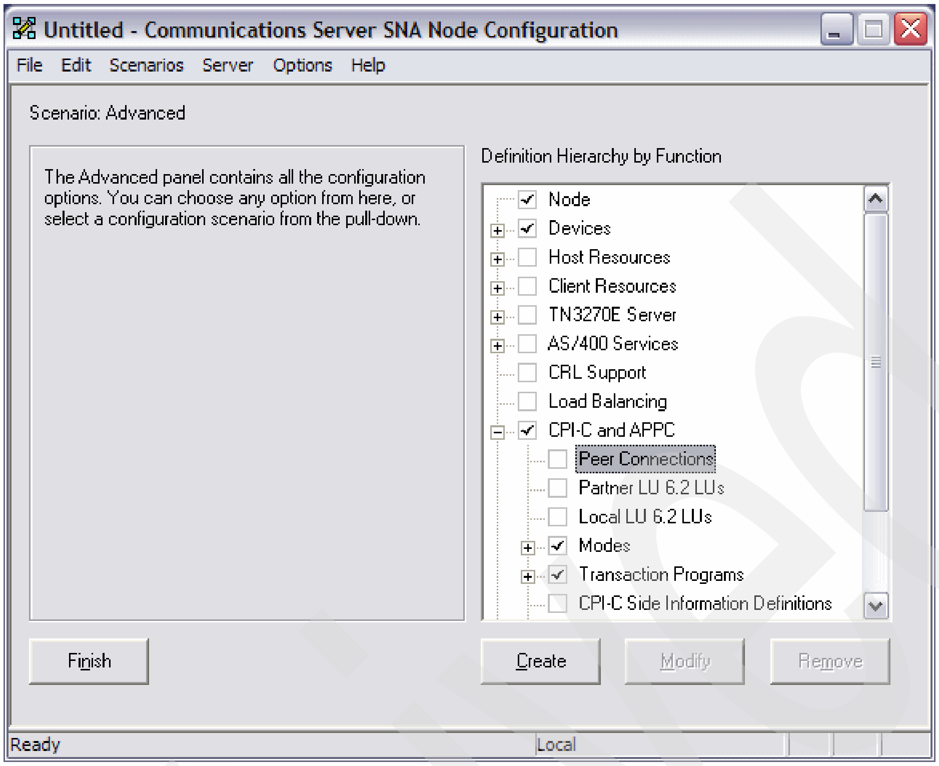
3. You can use the default values for the port in most cases.

Peer Connection

A peer connection in the IBM Communications Server for Windows is an APPN link to the other node. You can define a peer connection in **CPI-C and APPC** menu in the SNA node configuration utility as shown in Figure 8-8 on page 273. We make two link stations, one for SC30M (preferred network node server), and another for SC31M (backup network node server).

Complete the following steps:

1. In Figure 8-8 under the Definition Hierarchy by Function click **Peer Connections** and click **Create**.



Scenario: Advanced

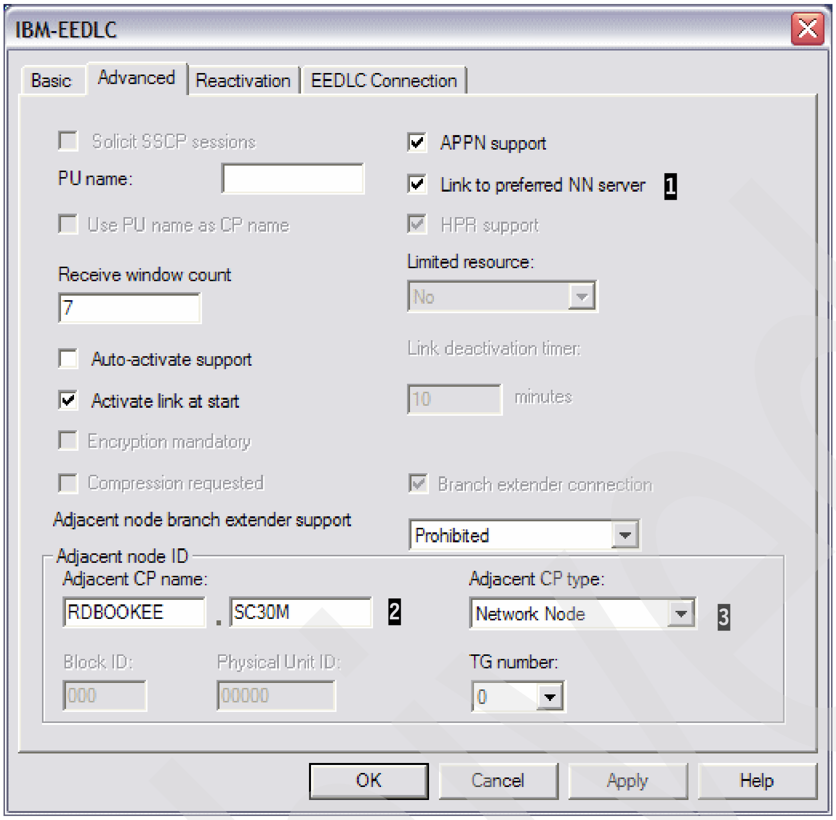
The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

Definition Hierarchy by Function

- ☑ Node
- ☑ Devices
- ☐ Host Resources
- ☐ Client Resources
- ☐ TN3270E Server
- ☐ AS/400 Services
- ☐ CRL Support
- ☐ Load Balancing
- ☑ CPI-C and APPC
  - ☐ Peer Connections
  - ☐ Partner LU 6.2 LUs
  - ☐ Local LU 6.2 LUs
  - ☑ Modes
  - ☑ Transaction Programs
  - ☐ CPI-C Side Information Definitions

Figure 8-8 Peer connection menu

2. In the peer connections definition window, select the **Advanced** tab for configuring the general properties of this peer connection, as shown in Figure 8-9.



□ Solicit SSCP sessions

PU name:

□ Use PU name as CP name

Receive window count: 7

□ Auto-activate support

☑ Activate link at start

□ Encryption mandatory

□ Compression requested

Adjacent node branch extender support: Prohibited

☑ APPN support

☑ Link to preferred NN server 1

☑ HPR support

Limited resource: No

Link deactivation timer: 10 minutes

☑ Branch extender connection

Adjacent node ID

Adjacent CP name: RDBOOKEE . SC30M 2

Adjacent CP type: Network Node 3

Block ID: 000

Physical Unit ID: 00000

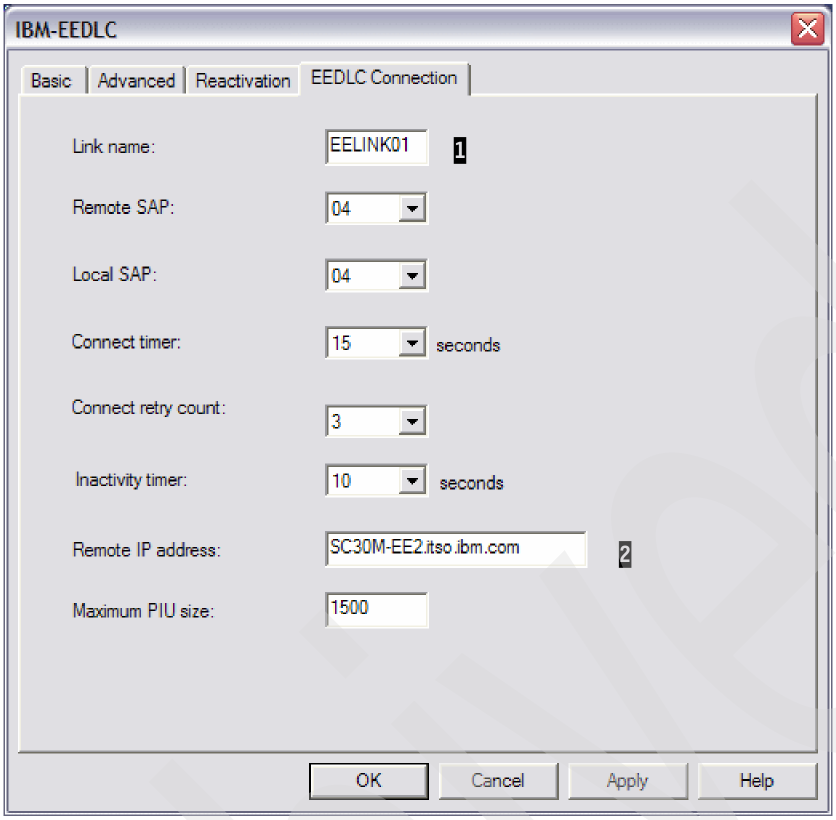
TG number: 0

Figure 8-9 Configuring the Advanced tab for peer connection to SC30M

Note the following explanation for Figure 8-9:

- 1: You can specify that this remote node is the preferred network node server (NNS). An End Node (EN) can make two CP-CP sessions with only one network node (NN) at one time. This NN is called the network node server of the EN. If you do not specify which node is the NNS, then the first NN which makes CP-CP sessions with the EN is chosen to be the NNS of that EN. In our scenario, `SC30M` is chosen to be the preferred network node server, and SC31M will be a backup network node.
- 2: The CP name of the remote node. Specify the fully qualified Control Point (CP) name (including NETID and CP name) here.
- 3: The node type of the remote host. In our scenario, the remote node is a `Network Node (NN)`.

3. Next, select the **EEDLC Connection** tab to proceed with the configuration. This opens the window shown in Figure 8-10.



Link name: EELINK01

Remote SAP: 04

Local SAP: 04

Connect timer: 15 seconds

Connect retry count: 3

Inactivity timer: 10 seconds

Remote IP address: SC30M-EE2.itso.ibm.com

Maximum PIU size: 1500

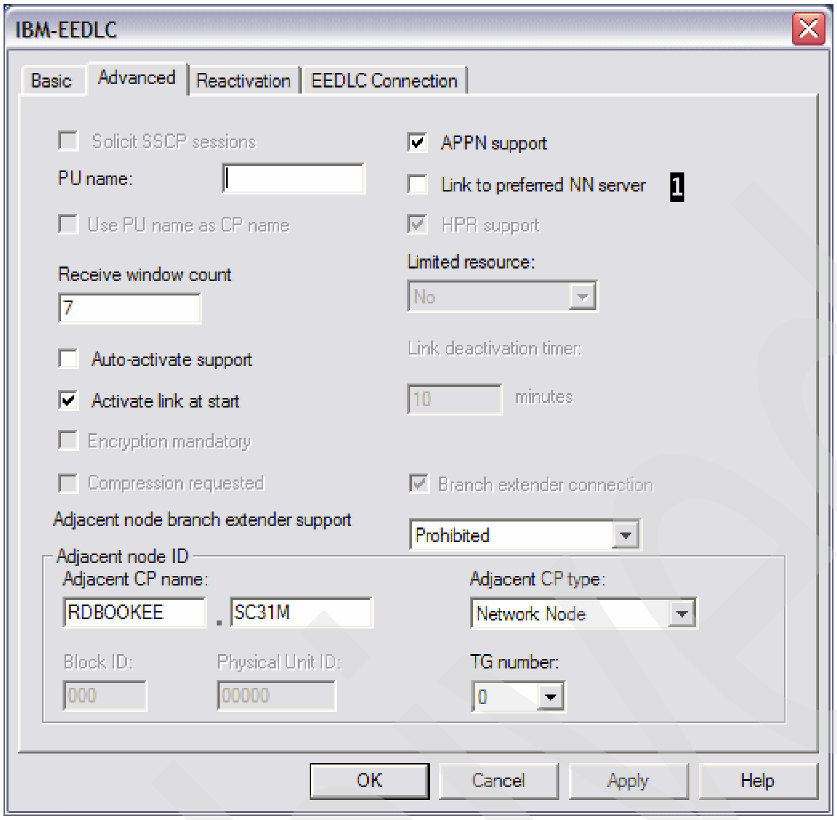
Figure 8-10 Configuring the EEDLC connection tab for peer connection to SC30M

Note the following explanation for Figure 8-10:

– 1: This is the name of the peer connection (Link station). This name is meaningful to only this server.

– 2: The IP address or host name of the remote host. In our scenario, we use the host name instead of the IP address of each system.

4. Next, we define one more peer connection to the SC31M (backup network node). See Figure 8-11.



IBM-EEDLC

Basic | Advanced | Reactivation | EEDLC Connection

□ Solicit SSCP sessions

PU name:

□ Use PU name as CP name

Receive window count: 7

□ Auto-activate support

☑ Activate link at start

□ Encryption mandatory

□ Compression requested

Adjacent node branch extender support: Prohibited

☑ APPN support

□ Link to preferred NN server **1**

☑ HPR support

Limited resource: No

Link deactivation timer: 10 minutes

☑ Branch extender connection

Adjacent node ID

Adjacent CP name: RDBOOKEE . SC31M

Adjacent CP type: Network Node

Block ID: 000

Physical Unit ID: 00000

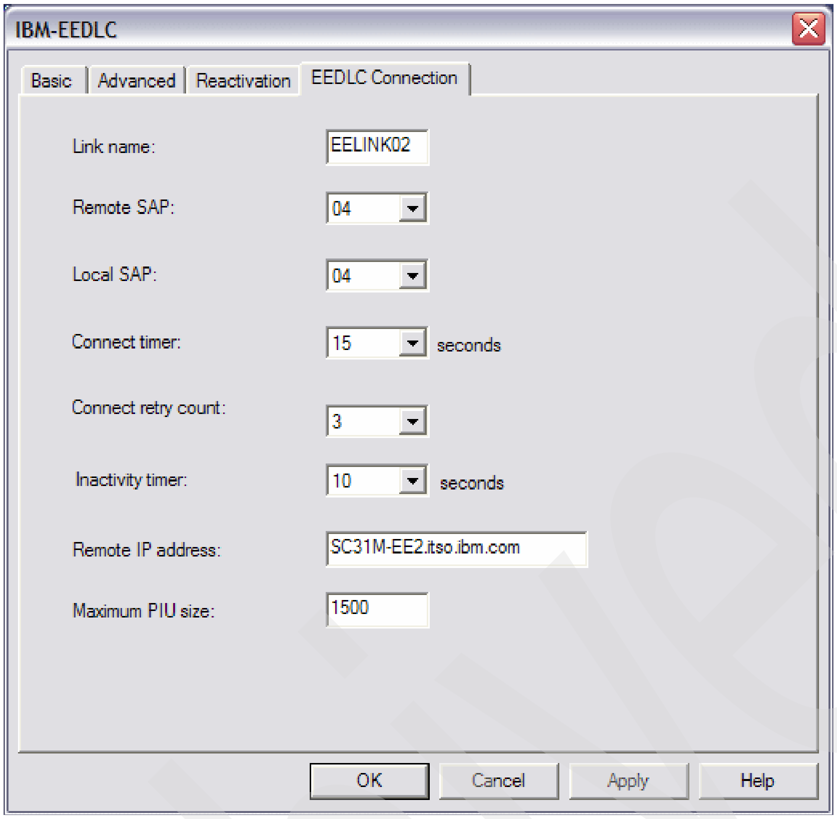
TG number: 0

Figure 8-11 Configuring Advanced tab for peer connection to SC31M

Note the following explanation for Figure 8-11:

– **1**: SC31M is not a preferred network node server.

5. Configure the IBM-EEDLC tab for peer connection to SC31MSC, as shown in Figure 8-12.



Link name: EELINK02

Remote SAP: 04

Local SAP: 04

Connect timer: 15 seconds

Connect retry count: 3

Inactivity timer: 10 seconds

Remote IP address: SC31M-EE2.itso.ibm.com

Maximum PIU size: 1500

Figure 8-12 Configuring IBM-EEDLC tab for peer connection to SC31M

Configuring connection network

In our scenario, we use a connection network to communicate with other nodes except the network node server and backup network node server. For more information about connection network, refer to “APPN connection networks” on page 38.

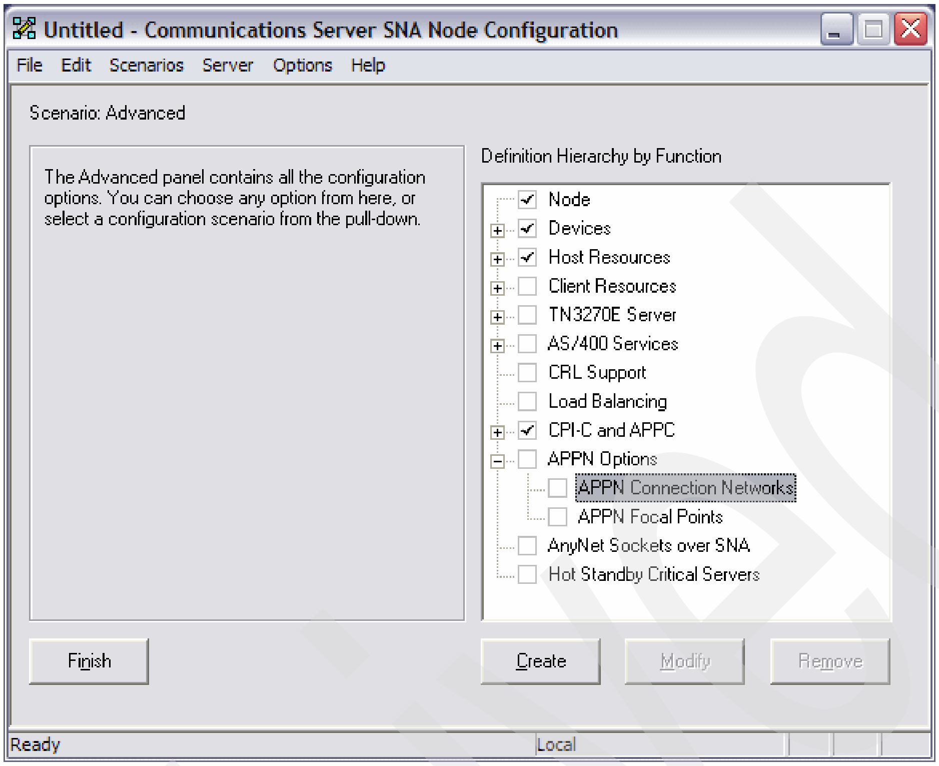
You can define a connection network as follows:

1. In Figure 8-13, under the Definition Hierarchy by Function area do one of the following:

   – Select **APPN Options** → **APPN Connection Networks.** Click **Create.**

   – Select **APPN Options** and then double-click **APPN Connection Networks.**

   This opens Define a Network window, as shown in Figure 8-14.


Untitled - Communications Server SNA Node Configuration
File Edit Scenarios Server Options Help
Finish
Create
Modify
Remove
Ready
Local

Scenario: Advanced

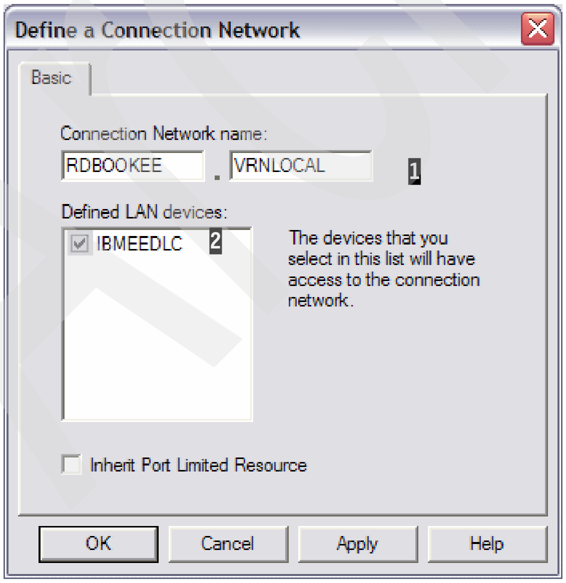
The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

Definition Hierarchy by Function

- ☑ Node
- ☑ Devices
- ☑ Host Resources
- □ Client Resources
- □ TN3270E Server
- □ AS/400 Services
- □ CRL Support
- □ Load Balancing
- ☑ CPI-C and APPC
- □ APPN Options
  - □ APPN Connection Networks
  - □ APPN Focal Points
- □ AnyNet Sockets over SNA
- □ Hot Standby Critical Servers

Figure 8-13 APPN Connection Networks menu

2. Configure a connection network as follows (as shown in Figure 8-14):
   - 1: The name of connection network.
   - 2: The port which is used for this connection network. In our scenario, `IBMEEDLC` will be used.



Connection Network name: RDBOOKEE . VRNLOCAL

Defined LAN devices:

☑ IBMEEDLC

The devices that you select in this list will have access to the connection network.

□ Inherit Port Limited Resource

Figure 8-14 Configuring a connection network

2

> **Note:** Be careful to define a connection network in IBM Communications Server for Windows if you have multiple network interfaces and each adapter is connected to a different subnet. We cannot specify which adapter (port) will be used for the connection network. If your connection network can be reached through Ethernet adapter #2, but IBM Communications Server for Windows picks Ethernet adapter #1 for connection network, it will not work properly. IBM Communications Server for Windows always picks the first Ethernet adapter for connection network. This behavior will be changed to the way other communications servers are using. Communications server for other platforms such as AIX and Linux can specify ports used by connection network.

Configuring VTAM for connectivity using EE

The APPN environment supports dynamic definitions. This means you can connect to another APPN node without node-specific definitions. VTAM controls this behavior with the DYNPU parameter in the external communications adapter (XCA) major node definition. In our scenario, we do not use dynamic PU definitions, therefore we show the VTAM definitions for the IBM Communications Server for Windowsto connect to the z/OS LPARs in this section.

You must define the following to configure the EE connection of the IBM Communications Server for Windows.

- “Enterprise Extender XCA major node”
- “PU definition for CS Windows”

Enterprise Extender XCA major node

We use two IP addresses for EE in VTAM side (with multiple VIPA). In Example 8-1, we show the XCA major node of SC30M.

Example 8-1 EEXCA XCA major node for SC30M

```
EEXCA30         VBUILD TYPE=XCA
EEPORT30         PORT MEDIUM=HPRIP,                                   *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                *
               IPRESOLV=2,                                            *
               SRQTIME=15,SRQRETRY=3
*
EEGVL301       GROUP DIAL=YES,CALL=INOUT,                             *
               HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                       * 1
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                 * 2
               TGP=EEXTCAMP,                                          *
               KEEPACT=YES,                                           *
               DYNPU=YES,                                             * 3
               UNRCHTIM=30,                                           *
               AUTOGEN=(16,EEM30,EEN30)
*
EEGVG301       GROUP DIAL=YES,CALL=INOUT,                             *
               HOSTNAME=SC30M-EE2.ITSO.IBM.COM,                       * 4
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                     * 5
               TGP=EEXTWAN,                                           *
               KEEPACT=YES,                                           *
               DYNPU=NO,                                              * 6
               UNRCHTIM=180,                                          *
               AUTOGEN=(16,EEX30,EEY30)
```

Note the following explanation for Example 8-1:

- 1: Host name for the first group.
- 2: The name of a connection network is RDBOOKEE.VRNLOCAL and its type is local. All participants of this connection network must be located within the same APPN topology subnetwork (the same NETID).
- 3: Dynamic definition of PU is allowed for this group.
- 4: Host name for the second group.
- 5: The name of a connection network is W3IBMCOM.VRN and its type is global. Nodes located in different APPN topology subnetworks can share the same connection network.
- 6: Dynamic definition of PU is prohibited for this group, a predefined PU is required.

We use the second group for connection via defined link stations. Dynamic PU definition is prohibited for this group, so we have to define a PU for IBM Communications Server for Windows node. For connection network, we use the local connection network because IBM Communications Server for Windows node is in the same APPN subnetwork (use the same NETID) with other nodes. Using the connection network implies that DYNPU=YES, and no VTAM definition is required for IBM Communications Server for Windows node through connection network. For more information about VTAM definition and XCA major node, refer to 4.4.2, “Defining the XCA HPRIP major node” on page 142.

PU definition for CS Windows

Each system (SC30M and SC31M) should have the definition such as shown in Example 8-2, if the DYNPU=NO in XCA major node of VTAM.

Example 8-2 PU Definition for CS Windows

```
EESWWINO VBUILD TYPE=SWNET
EEPUWIN  PU    CPNAME=EECPWIN,                             1           *
               NETID=RDBOOKEE,                             2           *
               TGN=06,                                     3           *
               CPCP=YES,                                   4           *
               HPR=YES,                                                *
               DYNLU=YES,                                  5           *
               ISTATUS=ACTIVE
```

Note the following explanation for Example 8-2:

- 1: The CP name of the IBM Communications Server for Windows node. This name must be the same as the name you configured in the node definition of CS Windows.
- 2: The NETID of this SNA network.
- 3: The TG number which is used for this link. We defined the TG number as 0 in the IBM Communications Server for Windows side (see Figure 8-9 on page 274). With this configuration, TG number will be decided when the XID is exchanged, and the TG number will be 6.
- 4: The CP-CP sessions capability. Set this parameter to YES.
- 5:The dynamic LU definition feature. If this parameter is set to YES, then all the independent LUs defined in the IBM Communications Server for Windows can be used in this z/OS VTAM without a CDRSC definition in VTAM. Otherwise, you have to define a CDRSC entry for each independent LU.

8.3.3 Configuring dependent LUs with DLUR

We define dependent LUs and DLUR PUs in this section. For further information about dependent LU and the DLUR, refer to “Dependent LU Requester and Server (DLUR/DLUS)” on page 33. The following topics are covered:

- “DLUR definition in IBM Communications Server for Windows”
- “DLUR definition in VTAM”

DLUR definition in IBM Communications Server for Windows

For using the dependent LUs, you must define the following in IBM Communications Server for Windows:

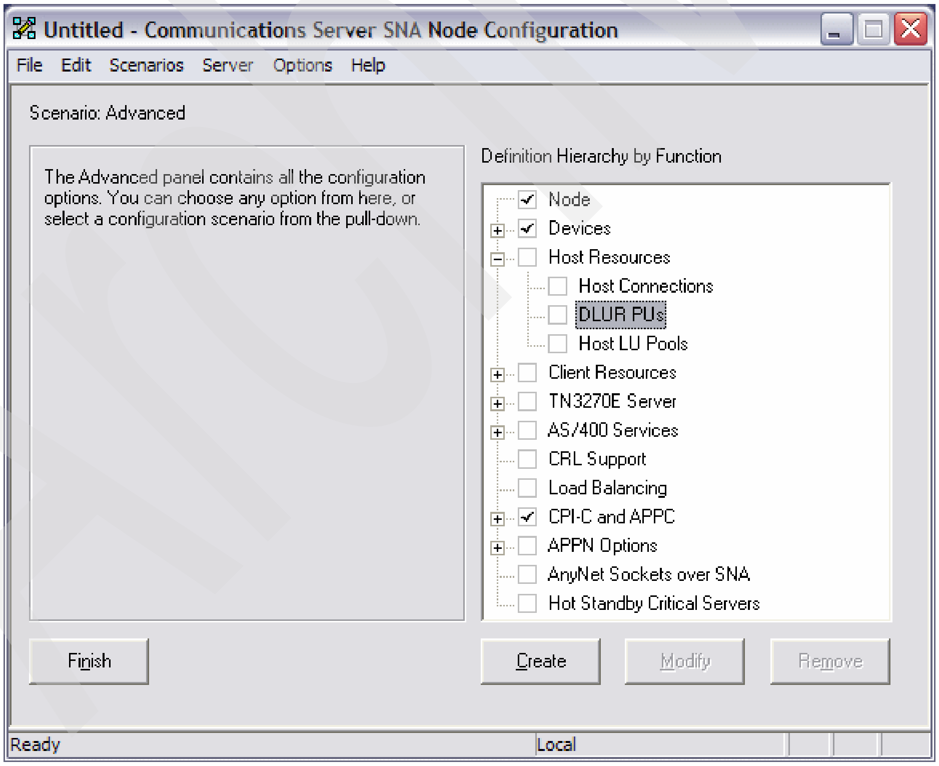
- “DLUR PU”
- “Dependent LUs”

DLUR PU

DLUR PU is a PU2.0 type, which provides PU services to the dependent LUs. It is configured as follows:

1. To define the DLUR PU in IBM Communications Server for Windows, in Figure 8-15, under the Definition Hierarchy by Function area do one of the following:
   - – Select **Host Connections** → **DLUR PUs**. Click **Create**.
   - – Select **Host Connections** and then double-click **DLUR PUs.**

   This opens the Define a DLUR PU window, as shown in Figure 8-16.



Scenario: Advanced

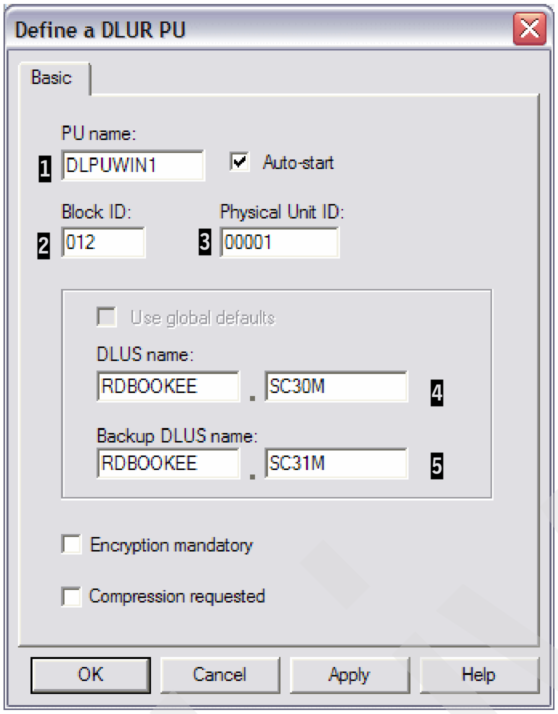
The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

Definition Hierarchy by Function

- ☑ Node
- ☑ Devices
- □ Host Resources
  - □ Host Connections
  - □ DLUR PUs
  - □ Host LU Pools
- □ Client Resources
- □ TN3270E Server
- □ AS/400 Services
- □ CRL Support
- □ Load Balancing
- ☑ CPI-C and APPC
- □ APPN Options
- □ AnyNet Sockets over SNA
- □ Hot Standby Critical Servers

Figure 8-15 DLUR PU menu

2. Configure DLUR PU as shown in Figure 8-16.



Define a DLUR PU

Basic

1 PU name: DLPUWIN1

☑ Auto-start

2 Block ID: 012

3 Physical Unit ID: 00001

□ Use global defaults

4 DLUS name: RDBOOKEE . SC30M

5 Backup DLUS name: RDBOOKEE . SC31M

□ Encryption mandatory

□ Compression requested

Figure 8-16 Configuring the DLUR PU

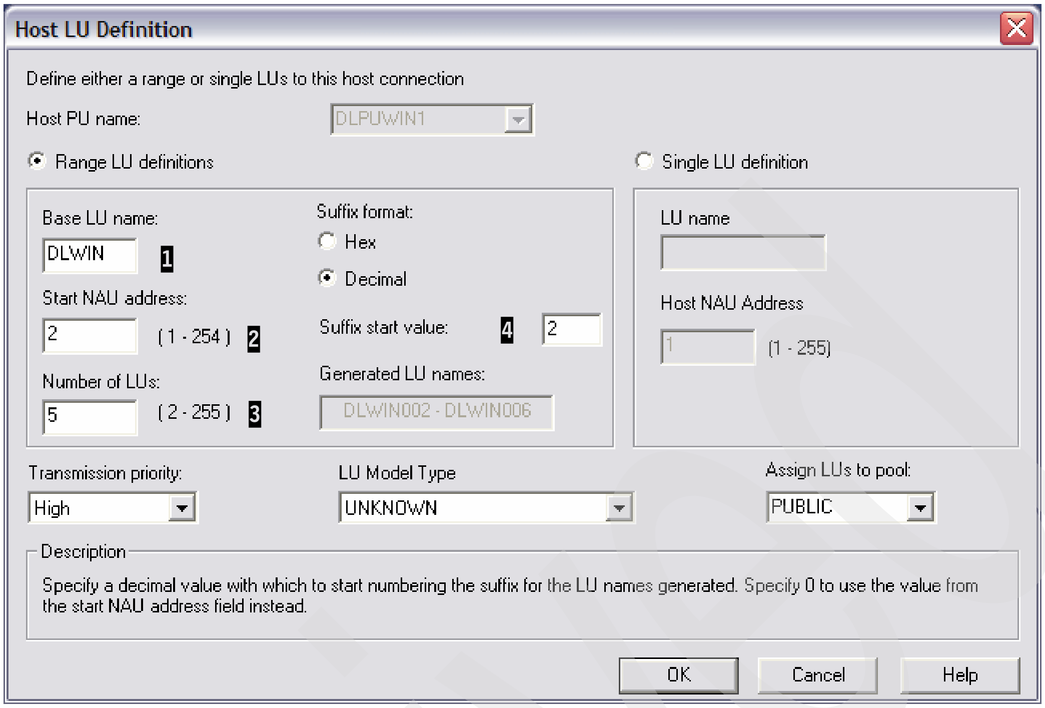
Note the following explanation for Figure 8-16:

- 1: This is the PU name for this DLUR PU for dependent LUs. This name is meaningful only in this IBM Communications Server for Windows, so you can use a different name from the PU name in the VTAM definition. But for management purposes, we recommend using the same name as your VTAM definition.
- 2~3: This is the PU identification number. VTAM has IDBLK (ID block) and IDNUM (ID number) parameters in its PU definition for this client, and the two definitions must be the same. In this case, the IDBLK is `012`, and IDNUM is `00001`. This value should be unique in a VTAM.
- 4: This is the SSCP name of the primary Dependent LU Server (DLUS) name. IBM Communications Server for Windows will try to establish a CPSVRMGR pipe (DLUS-DLUR sessions) with this DLUS. When this attempt fails, it will try to the backup DLUS if it is available.
- 5: This is the SSCP name of the backup DLUS. if you use the backup DLUS, then the same PU and LU definitions must be configured in the VTAM of backup DLUS.

Note: You can define multiple DLUR PUs in a node. Because a PU can support only 255 dependent LUs, if you want more LUs you have to define more DLUR PUs to the same primary/backup DLUS pair. Moreover it is possible to define more DLUR PUs to the other primary/backup DLUS pair.

Dependent LUs

When you click **OK** in Figure 8-16, IBM Communications Server for Windows will ask you whether you will define the dependent LUs on this DLUR PU. Click **Yes**. This opens the dependent LUs definition window as shown in Figure 8-17.



Host LU Definition

Define either a range or single LUs to this host connection

Host PU name: DLPUWIN1

● Range LU definitions ○ Single LU definition

Base LU name: DLWIN 1

Start NAU address: 2 (1 - 254) 2

Number of LUs: 5 (2 - 255) 3

Suffix format: ○ Hex ● Decimal

Suffix start value: 4 2

Generated LU names: DLWIN002 - DLWIN006

LU name:

Host NAU Address: 1 (1 - 255)

Transmission priority: High

LU Model Type: UNKNOWN

Assign LUs to pool: PUBLIC

Description

Specify a decimal value with which to start numbering the suffix for the LU names generated. Specify 0 to use the value from the start NAU address field instead.

Figure 8-17 Configuring dependent LUs on DLUR PU

Note the following explanation for Figure 8-17:

- 1: This is the base name of dependent LUs. You can use up to 6 characters here, and the names of all the dependent LUs defined here will be started with this base name.
- 2: This is the local address of the first LU of this range of LUs. The local address of LU must match the LOCADDR parameter in VTAM definition for each LU.

> **Note:** LU name defined here is only meaningful for this CS Windows. In other words, There is no need for matching these LU names with the LU names in the VTAM definition. Only the LU number (LOCADDR) must be the same on the both sides. But for management purpose, we recommend using the same name on both sides.

- 3: The number of LUs which will be defined.
- 4: You can specify a number from which to start naming the LUs in the range.

DLUR definition in VTAM

In VTAM, you should define the type 2.0 PU, and the dependent LUs in a switched major node. In our scenario, we make a switched major node such as in Example 8-3.

Example 8-3 Switched major node definition for DLUR

```
DLSWWIN VBUILD TYPE=SWNET
DLPUWIN1 PU    PUTYPE=2,USSTAB=USSSNAEE,                           *
               IDBLK=012,IDNUM=00001,                              * 1
               MODETAB=ALLMODES,DLOGMOD=DYNEEHIG,ANS=CONT            2
DLWIN002 LU    LOCADDR=2                                             3
DLWIN003 LU    LOCADDR=3
DLWIN004 LU    LOCADDR=4
DLWIN005 LU    LOCADDR=5
DLWIN006 LU    LOCADDR=6
```

Note the following explanation for Example 8-3:

- 1: IDBLK and IDNUM must match PU ID in the DLUR PU definition in IBM Communications Server for Windows.
- 2: ANS=CONT is required for DLUR PUs. If the default value (ANS=STOP) is used, SSCP-PU session and SSCP-LU sessions will be broken in case the primary DLUS is lost. This results in session outages during takeover to backup DLUS.
- 3: The name of the LU can be different from the one for IBM Communications Server for Windows, but the LOCADDR must match with the LU number in DLUR PU definition of IBM Communications Server for Windows. But we recommend using the same names for management purposes.

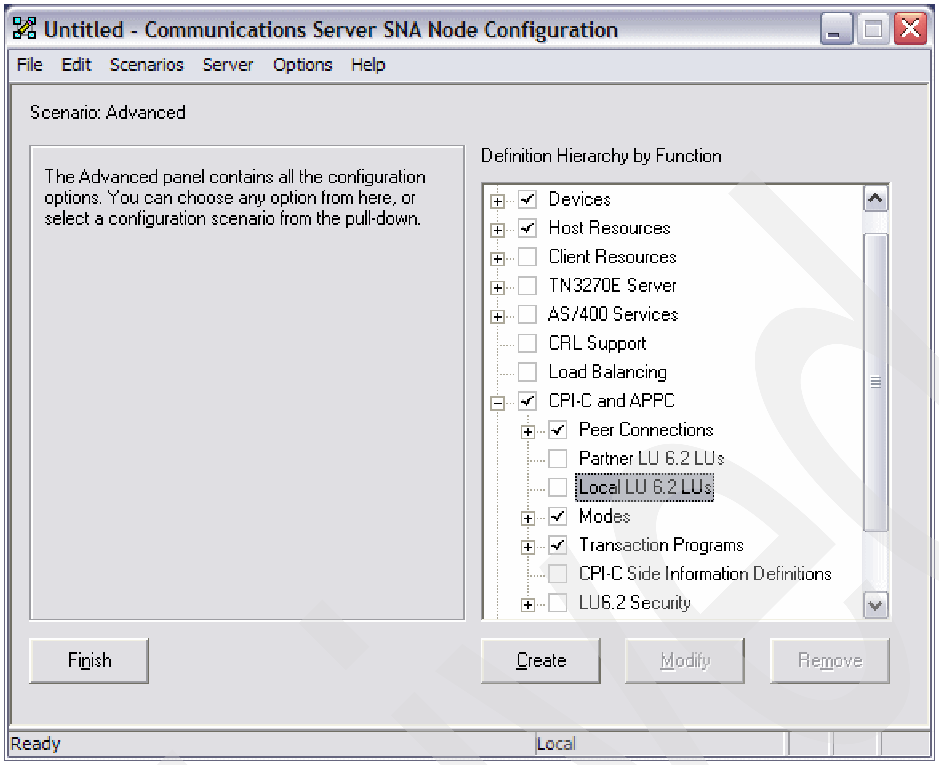
8.3.4 Configuring independent LUs

An independent LU has the capability of communication with another independent LU without the SSCP (VTAM). Thus, the independent LU is not a resource of VTAM, but a resource of the node in which the independent LU is defined. There are two ways of handling these independent LUs whether DYNLU parameter in VTAM is *yes* or *no*. If DYNLU is *yes*, then no definition is required for VTAM, and VTAM will generate a dynamic definition of a CDRSC entry for that independent LU. If DYNLU is *no*, then there must be a predefined CDRSC entry in the VTAM for that independent LU.

To define an independent LU complete the following:

1. Under Definition Hierarchy by Function (in Figure 8-18), Select **CPI-C and APPC.**
2. Do one of the following:
   – Select **Local LU 6.2 LUs,** and then click **Create.**
   – Double-click **Local LU 6.2 LUs.**

   This opens the Define a Local LU 6.2 window as shown in Figure 8-18.



Scenario: Advanced

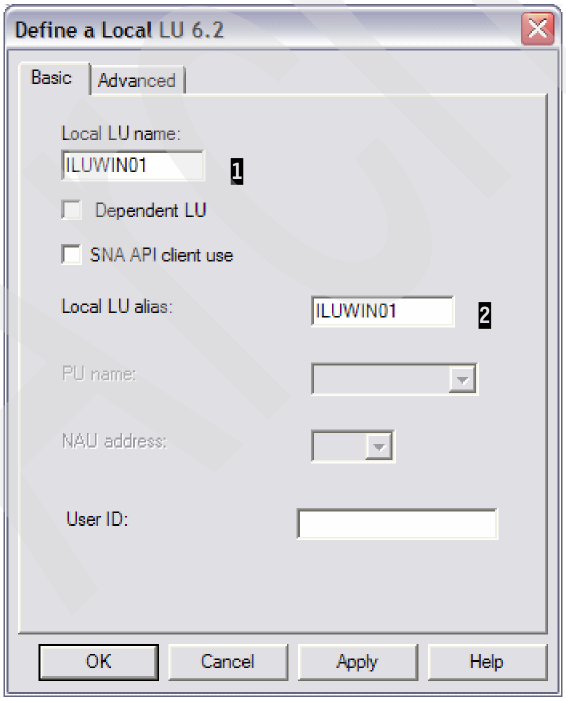
The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

Definition Hierarchy by Function

- ☑ Devices
- ☑ Host Resources
- □ Client Resources
- □ TN3270E Server
- □ AS/400 Services
- □ CRL Support
- □ Load Balancing
- ☑ CPI-C and APPC
  - ☑ Peer Connections
  - □ Partner LU 6.2 LUs
  - □ Local LU 6.2 LUs
  - ☑ Modes
  - ☑ Transaction Programs
  - □ CPI-C Side Information Definitions
  - □ LU6.2 Security

Figure 8-18 Local LU 6.2 LUs menu

3. Configure the independent LU as shown in Figure 8-19.



Local LU name: ILUWIN01

□ Dependent LU

□ SNA API client use

Local LU alias: ILUWIN01

PU name:

NAU address:

User ID:

Figure 8-19 Configuring an independent LU

Note the following explanation for Figure 8-19:

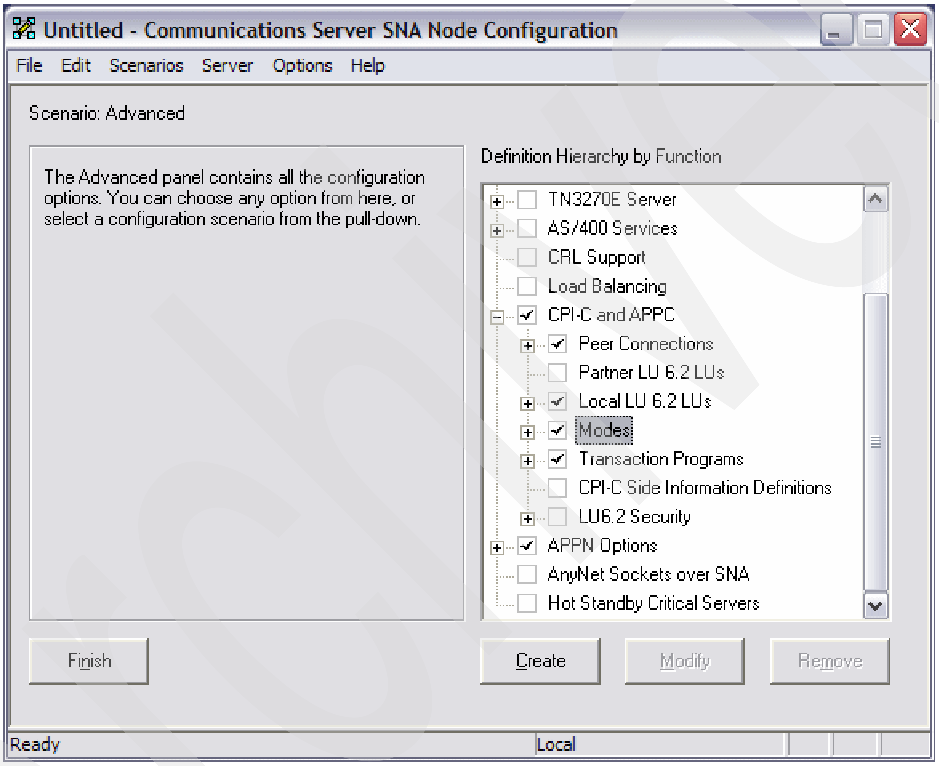
- 1: This is the name of the LU. This LU name will be used in z/OS also.
- 2: This is the alias of the LU. This value is meaningful to only this IBM Communications Server for Windows node.

8.3.5 Configuring LOGMODEs

Sometimes, your application can use the customized LOGMODE which is not shipped by IBM. In this case you can define a customized LOGMODE in the IBM Communications Server for Windows. You can define logmode for type 6.2 LUs only.

To define a mode complete the following steps:

1. Under the Definition Hierarchy by Function area (in Figure 8-20), select **CPI-C and APPC**.



Scenario: Advanced

The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

Definition Hierarchy by Function

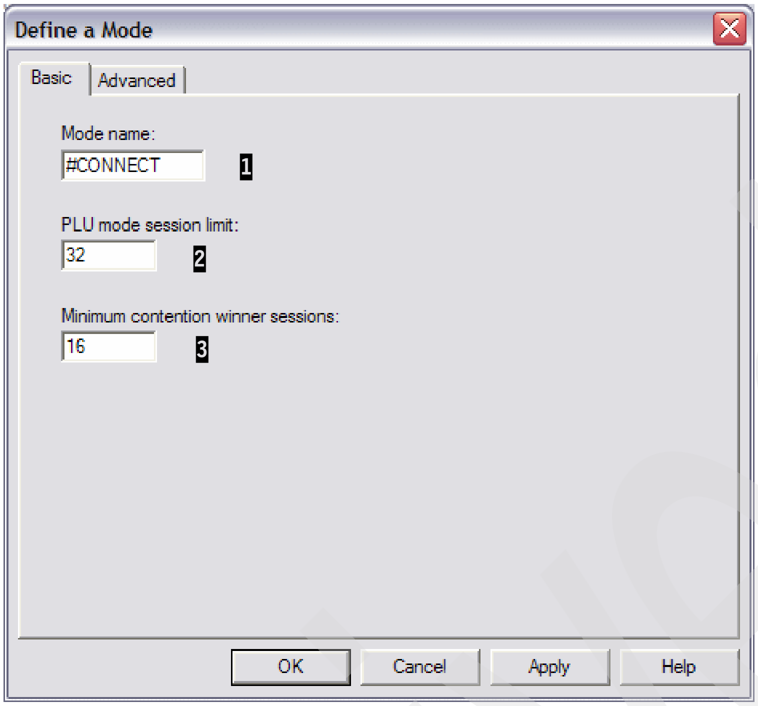
- □ TN3270E Server
- □ AS/400 Services
- □ CRL Support
- □ Load Balancing
- ☑ CPI-C and APPC
  - ☑ Peer Connections
  - □ Partner LU 6.2 LUs
  - ☑ Local LU 6.2 LUs
  - ☑ Modes
  - ☑ Transaction Programs
  - □ CPI-C Side Information Definitions
  - □ LU6.2 Security
- ☑ APPN Options
- □ AnyNet Sockets over SNA
- □ Hot Standby Critical Servers

Figure 8-20 Mode menu

2. Do one of the following:
   - Select **Modes** and click **Create.**
   - Double-click **Modes**.

   This opens the Define a Mode window, as shown in Figure 8-21.

3. Configure the Basic tab of this logmode, as shown in Figure 8-21.



Mode name: #CONNECT 1

PLU mode session limit: 32 2

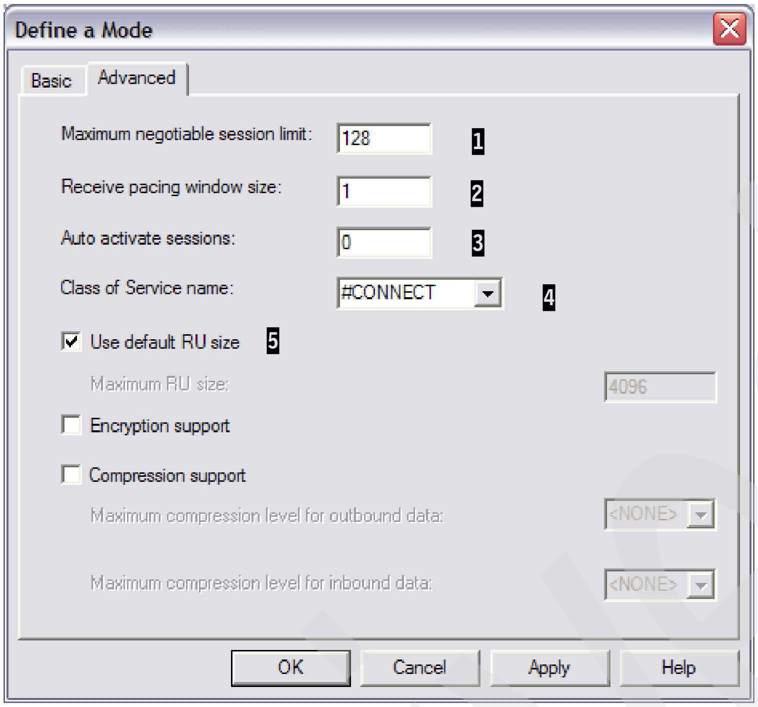
Minimum contention winner sessions: 16 3

Figure 8-21 Configuring Basic tab of mode

Note the following explanation for Figure 8-21:

- 1: The name of this logmode.
- 2: The default session limit of this logmode. This is the maximum number of sessions permitted between a pair of LUs using this mode, even with CNOS negotiation.
- 3: The minimum contention winner sessions. The number of sessions (up to the session limit) that IBM Communications Server for Windows must reserve for use by the local LU as the contention winner. Specify 0 if you do not want to reserve the contention winner sessions.

4. Next, configure the Advanced tab of this logmode, as shown in Figure 8-22.


Define a Mode

Maximum negotiable session limit: 128

Receive pacing window size: 1

Auto activate sessions: 0

Class of Service name: #CONNECT

☑ Use default RU size

Maximum RU size: 4096

☐ Encryption support

☐ Compression support

Maximum compression level for outbound data: <NONE>

Maximum compression level for inbound data: <NONE>

Figure 8-22 Configure the Advanced tab of mode

Note the following explanation for Figure 8-22:

- 1: The maximum number of the sessions which will be supported by this logmode. In many cases this value is set to the same value as the initial session limit. Through CNOS (Change Number of Sessions), this maximum number of sessions can be changed.
- 2: The initial pacing window size. The initial number of request units (RUs) that the local LU can receive before it must send a pacing response to the remote LU in adaptive pacing. If fixed pacing is used, this value specifies the receive pacing window.
- 3: You can specify how many sessions to activate automatically for each pair of LUs that use this mode. This value is used when CNOS exchange is initiated implicitly.
- 4: Name of the class of service (COS) to request when activating sessions on this mode. This must be defined in APPNCOS in VTAM.
- 5: You can specify whether IBM Communications Server for Windows uses the maximum RU size upper limit parameter to define the maximum RU size. If you select this check box then you can specify the upper bound for the maximum RU size. If you do not restrict the maximum RU size, then IBM Communications Server for Windows sets the upper bound for the maximum RU size to the largest value that can be accommodated in the link BTU size.

In Figure 8-21 and Figure 8-22, we define a new logmode named `#CONNECT` which uses the `#CONNECT` as its class of service.

8.4 Verifying and managing EE with CS for Windows

Most administration work will be done in the SNA Node Operations utility. You can start the SNA Node Operations utility by selecting **Start** → **All programs** → **IBM Communications Server** → **SNA Node Operations**.

The topics covered in this section are:

- “Starting resources”
- “Verifying resources”
- “Stopping resources”
- “Backing up your configuration”

8.4.1 Starting resources

We recommend that you use the “initially active” options of each resource, because it is easier to manage the resources. For LU-LU sessions, it is better to use batch-type scripts.

For starting a node, select **Operations** → **Start Node,** as shown in Figure 8-23.

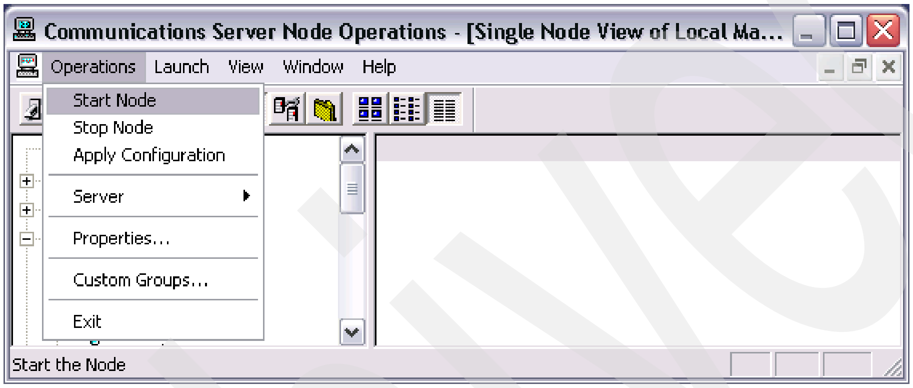


Figure 8-23 Starting node

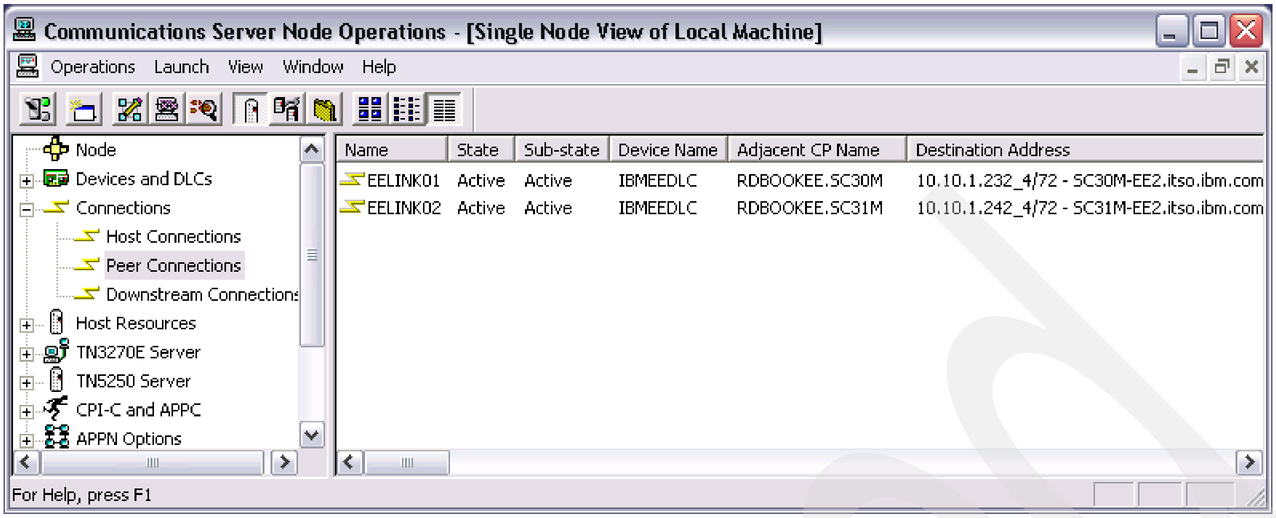
8.4.2 Verifying resources

You can easily monitor SNA activities with SNA Node Operations. In this section we explain the following:

- “Verifying connectivity to other nodes”
- “Verifying DLUR PU and dependent LUs”
- “Verifying HPR Path Switch”
- “Verifying the backup DLUS”

Verifying connectivity to other nodes

You can verify the connectivity to other nodes in the Connections menu (**Peer Connection** menu) in the SNA node operations utility.



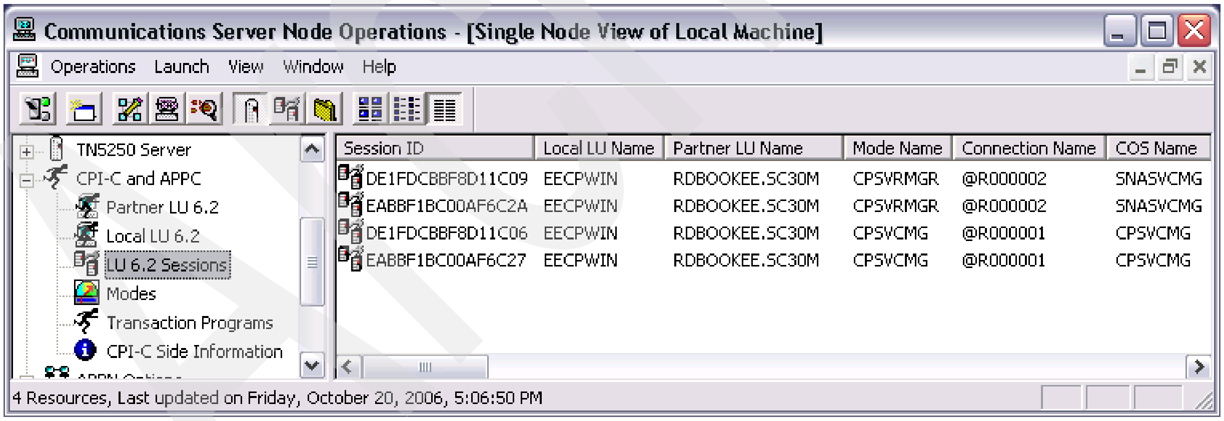
Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Address
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	10.10.1.232_4/72 - SC30M-EE2.itso.ibm.com
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	10.10.1.242_4/72 - SC31M-EE2.itso.ibm.com

Figure 8-24 Verifying the connectivity

In Figure 8-24, you can see that two peer connections are connected to SC30M and SC31M. Their State is `Active`, that is, those two peer connections are connected without any problem.

If an EN is connected to its network node server, then two CP-CP sessions are established between the two nodes. In our scenario, the preferred network node server is SC30M. With these CP-CP sessions, an EN can get topology and directory information from the network node server.

For verifying the CP-CP sessions, select **CPI-C and APPC** → **LU 6.2 Sessions** menu in the SNA node operations utility.



Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FDCBBF8D11C09	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
EABBF1BC00AF6C2A	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
DE1FDCBBF8D11C06	EECPWIN	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG
EABBF1BC00AF6C27	EECPWIN	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG

4 Resources, Last updated on Friday, October 20, 2006, 5:06:50 PM

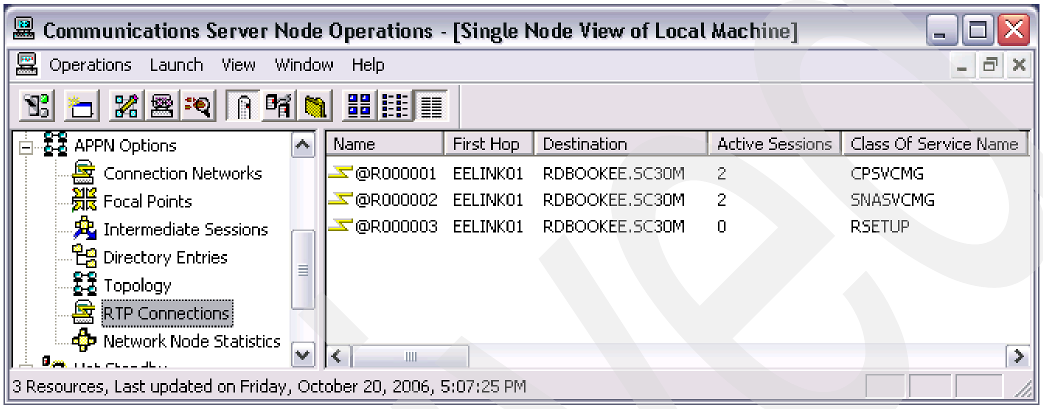
Figure 8-25 Verifying the CP-CP sessions

In Figure 8-25, you can see the four LU 6.2 Sessions. Among the four sessions, two sessions which are using CPSVCMG as their mode are the CP-CP sessions. Those two sessions are established through an RTP connection, @R000001. An RTP connection is a logical link station to the opposite endpoint which uses a specific class of service (COS). You can get information of the RTP connections by selecting **APPN Options** → **RTP Connections** menu in the SNA node operations utility.

In Figure 8-26, you can find three RTP connections. The first RTP connection, @R000001, uses the CPSVCMG for its class of service (COS), and this RTP connection is used for two

CP-CP sessions with the SC30M. Every RTP connection has a defined path to the other endpoint, and in this case, the next hop is EELINK01. This path of the RTP connection may change when the topology is changed. We show this HPR Path Switch feature shortly. For more information about HPR and RTP, refer to 1.2.1, “Positioning APPN and HPR” on page 4.

> **Note:** An RTP connection is created between two session endpoints. There may be several nodes between two endpoints. SNA node operations utility shows the next hop of that RTP connection only.



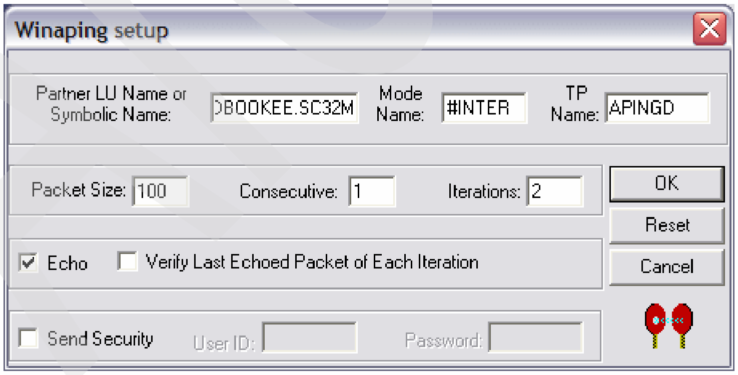
Name	First Hop	Destination	Active Sessions	Class Of Service Name
@R000001	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG
@R000002	EELINK01	RDBOOKEE.SC30M	2	SNASVCMG
@R000003	EELINK01	RDBOOKEE.SC30M	0	RSETUP

3 Resources, Last updated on Friday, October 20, 2006, 5:07:25 PM

Figure 8-26 Verifying the RTP connections

Next, we verify whether the connection network is working properly. The connection network does not support the CP-CP sessions, so you should have predefined link stations to the network node server and backup network node server, if it exists. We have another z/OS LPAR in our scenario, SC32M. We did not define a static connection to SC32M, but a dynamic connection to SC32M can be generated if needed, because SC32M is connected to the same connection network, RDBOOKEE.VRNLOCAL, in this IBM Communications Server for Windows.

The ping program for APPC, called “**winaping**” is provided for a reachability test. Open a command prompt and type the **winaping**. Figure 8-27 will appear.



Partner LU Name or Symbolic Name:)BOOKEE.SC32M
Mode Name: #INTER
TP Name: APINGD

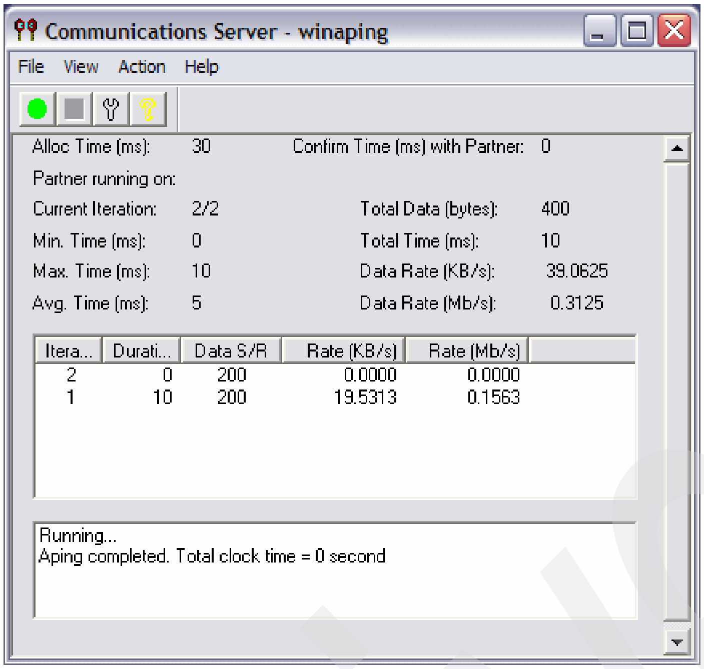
Packet Size: 100
Consecutive: 1
Iterations: 2

☑ Echo ☐ Verify Last Echoed Packet of Each Iteration

☐ Send Security User ID: Password:

Figure 8-27 Set up a winaping program

Fill the Partner LU Name field as the fully qualified name of the SC32M and click **OK**. And click the start button (the left-most one). Figure 8-28 will appear.



```
Alloc Time (ms):      30        Confirm Time (ms) with Partner:   0
Partner running on:
Current Iteration:    2/2       Total Data (bytes):   400
Min. Time (ms):       0         Total Time (ms):      10
Max. Time (ms):       10        Data Rate (KB/s):     39.0625
Avg. Time (ms):       5         Data Rate (Mb/s):     0.3125
```

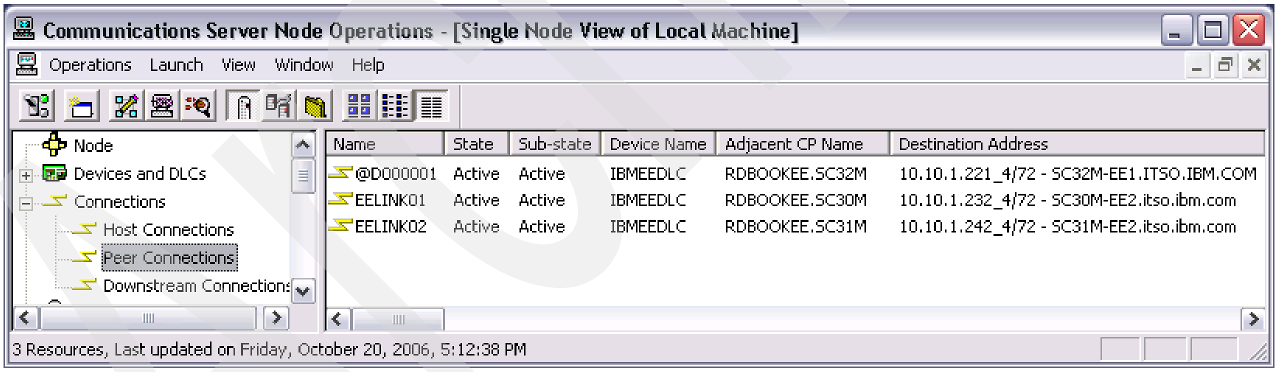
Itera...	Durati...	Data S/R	Rate (KB/s)	Rate (Mb/s)
2	0	200	0.0000	0.0000
1	10	200	19.5313	0.1563

```
Running...
Aping completed. Total clock time = 0 second
```

Figure 8-28   The result of aping to SC32M

Now, check the peer connection status.

In Figure 8-29, the new dynamic peer connection, `@D000001` is generated through connection network. If connection network is not used in this node, the APING utility will use two hop paths via the SC30M to SC32M or SC31M to SC32M.


Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
Node
Devices and DLCs
Connections
Host Connections
Peer Connections
Downstream Connections

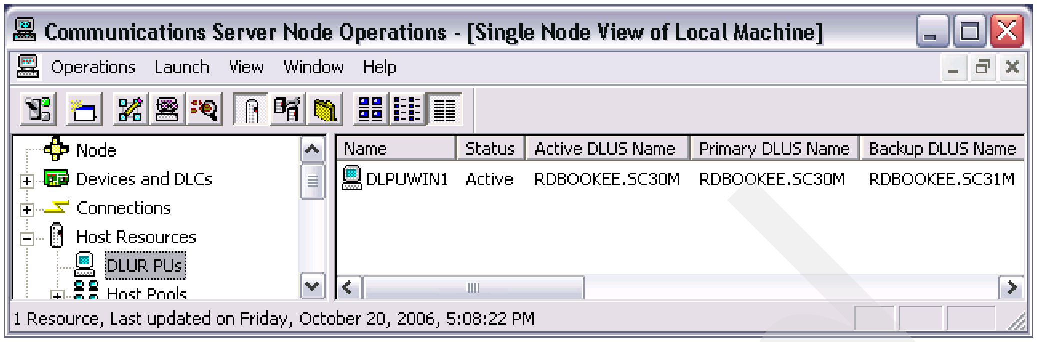
Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Address
@D000001	Active	Active	IBMEEDLC	RDBOOKEE.SC32M	10.10.1.221_4/72 - SC32M-EE1.ITSO.IBM.COM
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	10.10.1.232_4/72 - SC30M-EE2.itso.ibm.com
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	10.10.1.242_4/72 - SC31M-EE2.itso.ibm.com

3 Resources, Last updated on Friday, October 20, 2006, 5:12:38 PM

Figure 8-29   Dynamic peer connection through connection network

Verifying DLUR PU and dependent LUs

You can verify the DLUR PU by selecting **Host Resources** → **DLUR PUs** menu, as shown Figure 8-30 on page 293.


Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
Node
Devices and DLCs
Connections
Host Resources
DLUR PUs
Host Pools

Name	Status	Active DLUS Name	Primary DLUS Name	Backup DLUS Name
DLPUWIN1	Active	RDBOOKEE.SC30M	RDBOOKEE.SC30M	RDBOOKEE.SC31M

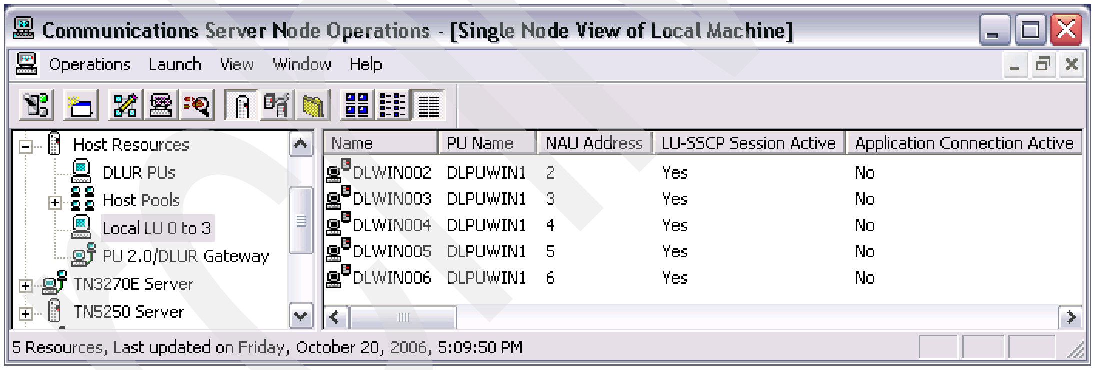
1 Resource, Last updated on Friday, October 20, 2006, 5:08:22 PM

Figure 8-30 Verifying the DLUR PU

In Figure 8-30, a DLUR PU, DLPUWIN1 is connected to SC30M because SC30M is primary the DLUS. If you see Figure 8-25 on page 290, there are two sessions which are established with the SC30M using the CPSVRMGR mode. These two sessions are called CPSVRMGR pipe, and through this pipe, SSCP-PU and SSCP-LU sessions are set up. These two sessions are using @R000002 as their RTP connection, which is using the SNASVCMG class of service (refer to Figure 8-26 on page 291).

Next, we verify the status of the dependent LUs.

In Figure 8-31, you can find five dependent LUs. The checkpoint is LU-SSCP Session Active field. If this field is `Yes`, then SSCP-LU session is established successfully.

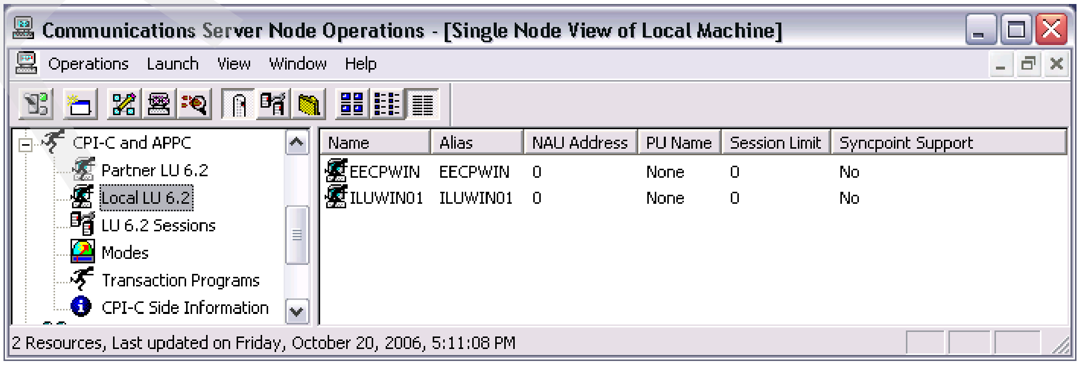

Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
Host Resources
DLUR PUs
Host Pools
Local LU 0 to 3
PU 2.0/DLUR Gateway
TN3270E Server
TN5250 Server

Name	PU Name	NAU Address	LU-SSCP Session Active	Application Connection Active
DLWIN002	DLPUWIN1	2	Yes	No
DLWIN003	DLPUWIN1	3	Yes	No
DLWIN004	DLPUWIN1	4	Yes	No
DLWIN005	DLPUWIN1	5	Yes	No
DLWIN006	DLPUWIN1	6	Yes	No

5 Resources, Last updated on Friday, October 20, 2006, 5:09:50 PM

Figure 8-31 Verifying the dependent LUs

For the independent LU, select the **CPI-C and APPC** → **Local LU 6.2** menu (see Figure 8-32).


Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
CPI-C and APPC
Partner LU 6.2
Local LU 6.2
LU 6.2 Sessions
Modes
Transaction Programs
CPI-C Side Information

Name	Alias	NAU Address	PU Name	Session Limit	Syncpoint Support
EECPWIN	EECPWIN	0	None	0	No
ILUWIN01	ILUWIN01	0	None	0	No

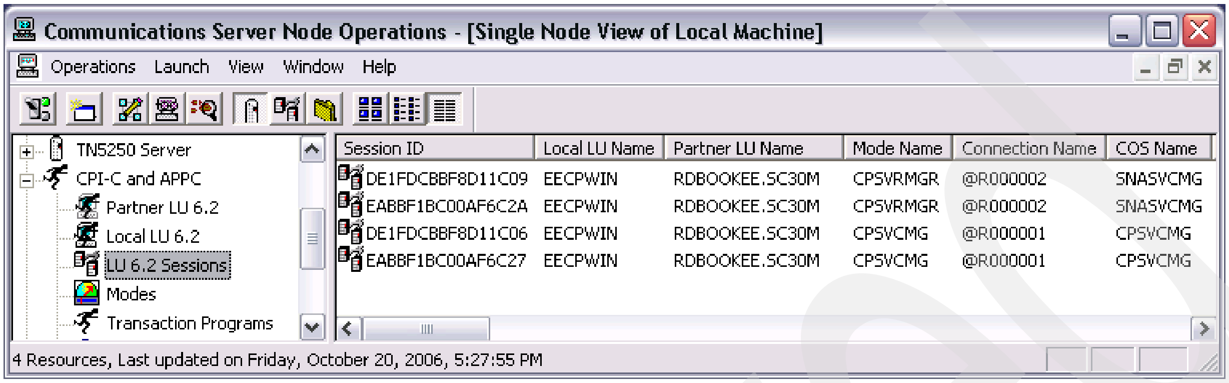
2 Resources, Last updated on Friday, October 20, 2006, 5:11:08 PM

Figure 8-32 Verifying the independent LU

Verifying HPR Path Switch

The RTP path can be changed if the topology is changed. We show this HPR Path Switch.

First, check the current status of each resource such as LU 6.2 sessions and RTP connections.

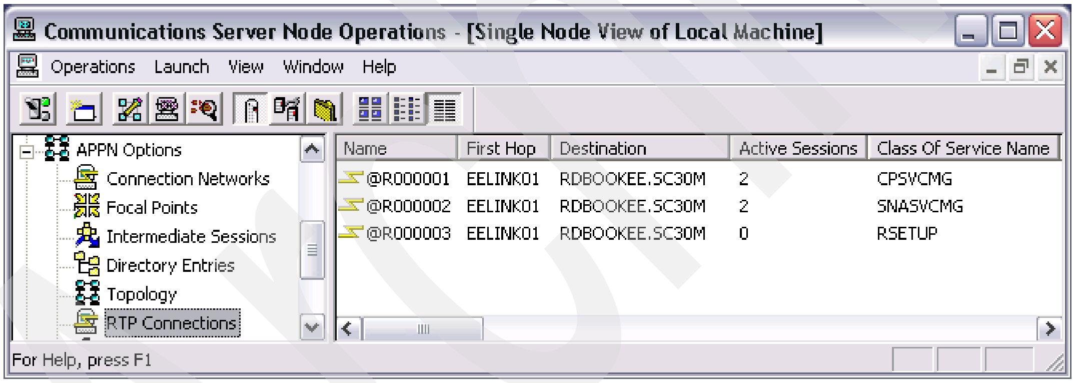


Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FDCBBF8D11C09	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
EABBF1BC00AF6C2A	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
DE1FDCBBF8D11C06	EECPWIN	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG
EABBF1BC00AF6C27	EECPWIN	RDBOOKEE.SC30M	CPSVCMG	@R000001	CPSVCMG

4 Resources, Last updated on Friday, October 20, 2006, 5:27:55 PM

Figure 8-33 Before the path switch, LU 6.2 sessions

Figure 8-33 shows four LU 6.2 sessions. Two sessions which use CPSVCMG as their mode are CP-CP sessions with SC30M, and two sessions which use CPSVRMGR as their mode are CPSVRMGR pipe (DLUS-DLUR sessions). CP-CP sessions are established through @R000001 RTP connection and CPSVRMGR pipe is established through @R000002 RTP connection.



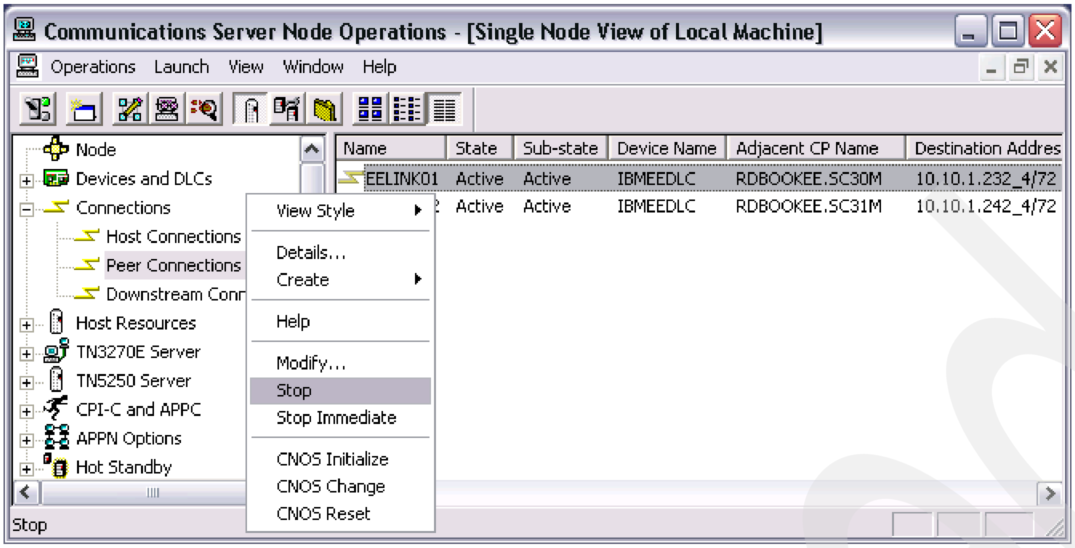
Name	First Hop	Destination	Active Sessions	Class Of Service Name
@R000001	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG
@R000002	EELINK01	RDBOOKEE.SC30M	2	SNASVCMG
@R000003	EELINK01	RDBOOKEE.SC30M	0	RSETUP

For Help, press F1

Figure 8-34 Before the path switch, RTP Connections

Figure 8-34 shows three RTP connections, where @R000001 is for CP-CP sessions and @R000002 is for CPSVRMGR pipe.

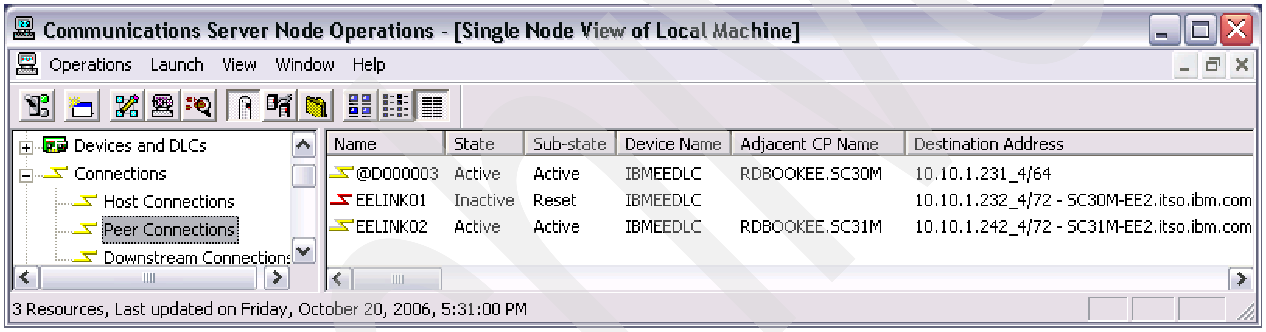
Now, shut down the EELINK01 peer connection (see Figure 8-35).



Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Addres
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	10.10.1.232_4/72
2	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	10.10.1.242_4/72

Figure 8-35 Shutdown the EELINK01 peer connection

You will observe that EELINK01 peer connection is broken and is now in an `Inactive` State.

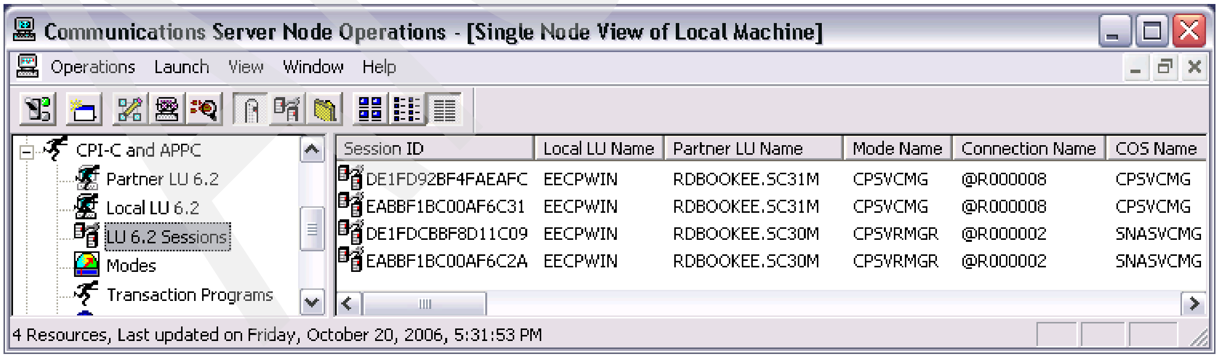


Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Address
@D000003	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	10.10.1.231_4/64
EELINK01	Inactive	Reset	IBMEEDLC		10.10.1.232_4/72 - SC30M-EE2.itso.ibm.com
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	10.10.1.242_4/72 - SC31M-EE2.itso.ibm.com

3 Resources, Last updated on Friday, October 20, 2006, 5:31:00 PM

Figure 8-36 EELINK01 in a inactive state

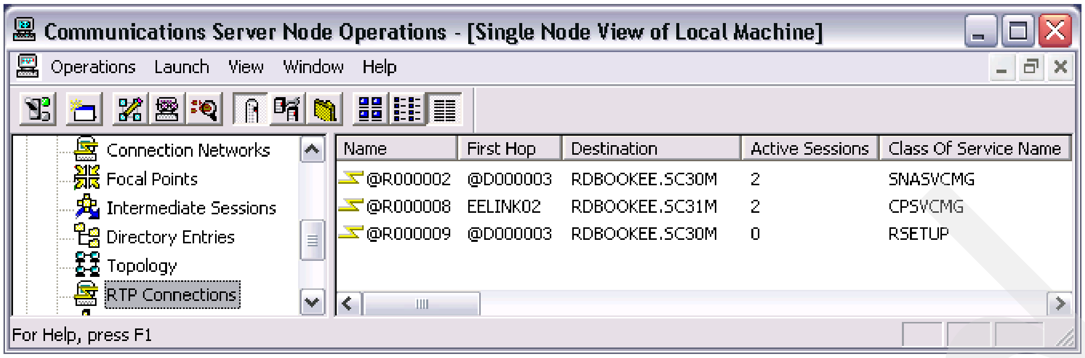
In Figure 8-36, the new dynamic peer connection, @D000003 is created. We explain this behavior shortly. Now, look at the LU 6.2 sessions and the RTP connections.



Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FD92BF4FAEAFC	EECPWIN	RDBOOKEE.SC31M	CPSVCMG	@R000008	CPSVCMG
EABBF1BC00AF6C31	EECPWIN	RDBOOKEE.SC31M	CPSVCMG	@R000008	CPSVCMG
DE1FDCBBF8D11C09	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
EABBF1BC00AF6C2A	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG

4 Resources, Last updated on Friday, October 20, 2006, 5:31:53 PM

Figure 8-37 LU 6.2 sessions after the path switch



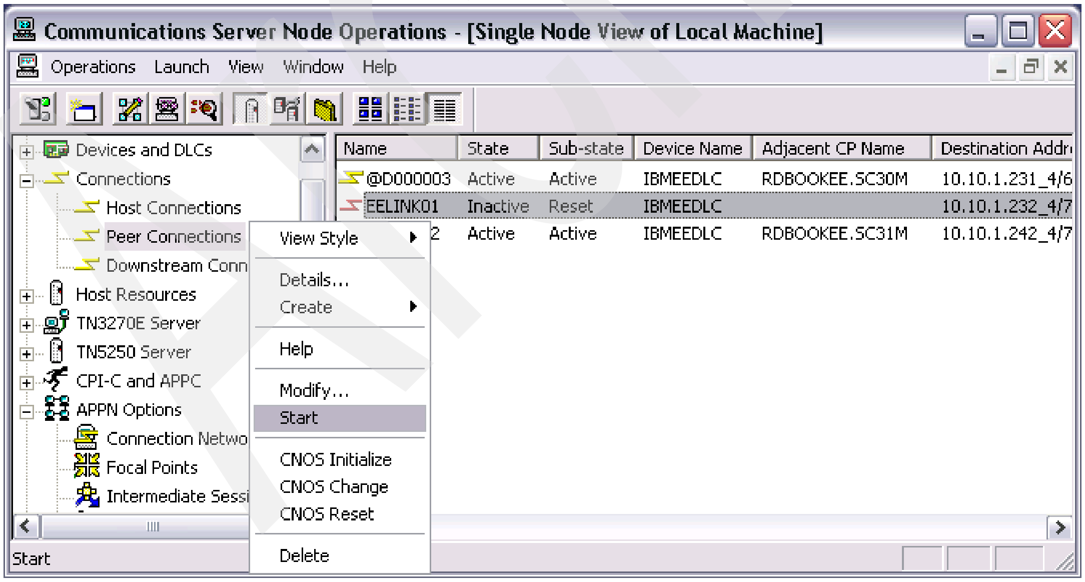
Name	First Hop	Destination	Active Sessions	Class Of Service Name
@R000002	@D000003	RDBOOKEE.SC30M	2	SNASVCMG
@R000008	EELINK02	RDBOOKEE.SC31M	2	CPSVCMG
@R000009	@D000003	RDBOOKEE.SC30M	0	RSETUP

Figure 8-38 RTP connections after the path switch

In Figure 8-37 on page 295, four sessions are still shown but now they are different from the sessions in Figure 8-33 on page 294. Two CP-CP sessions are now established with the SC31M system through @R000008 RTP connection (see Figure 8-38). You can find this RTP connection uses the EELINK02 peer connection as its first hop. So, the RTP connection @R000001 is deleted, and a new RTP connection is built for new CP-CP sessions with SC31M. This is because CP-CP sessions can be established with the adjacent CP only, and cannot be established through a dynamic peer connection which is created by connection network. Now, SC31 is the network node server for this IBM Communications Server for Windows.

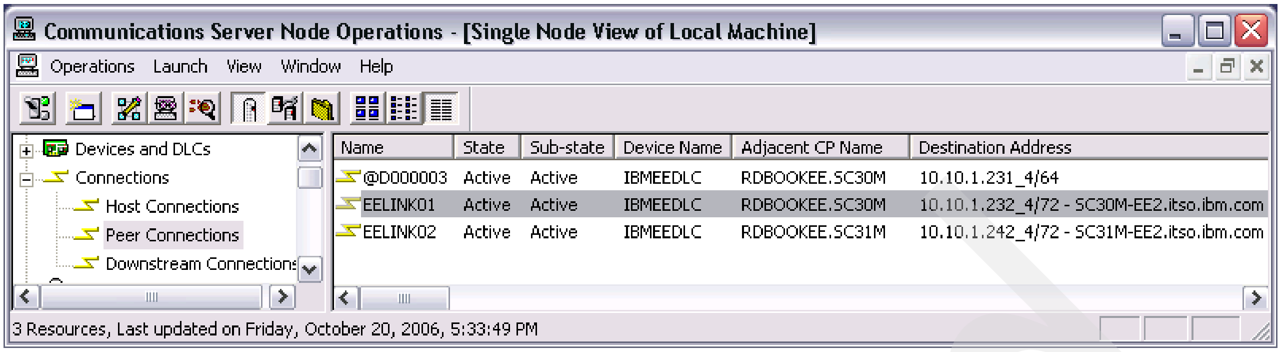
The CPSVRMGR pipe is still there, and still uses the @R000002 RTP connection, but the next hop of @R000002 is now changed to @D000003 dynamic peer connection. This is because SC30M and this IBM Communications Server for Windows (EECPWIN) are connected to the same connection network. If one of the two nodes does not participate in the same connection network, the path will be switched to EELINK02 peer connection.

Now, we reactivate the EELINK01 peer connection (see Figure 8-39 and Figure 8-40).



Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Addr
@D000003	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	10.10.1.231_4/6
EELINK01	Inactive	Reset	IBMEEDLC		10.10.1.232_4/7
2	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	10.10.1.242_4/7

Figure 8-39 Start the EELINK01 peer connection

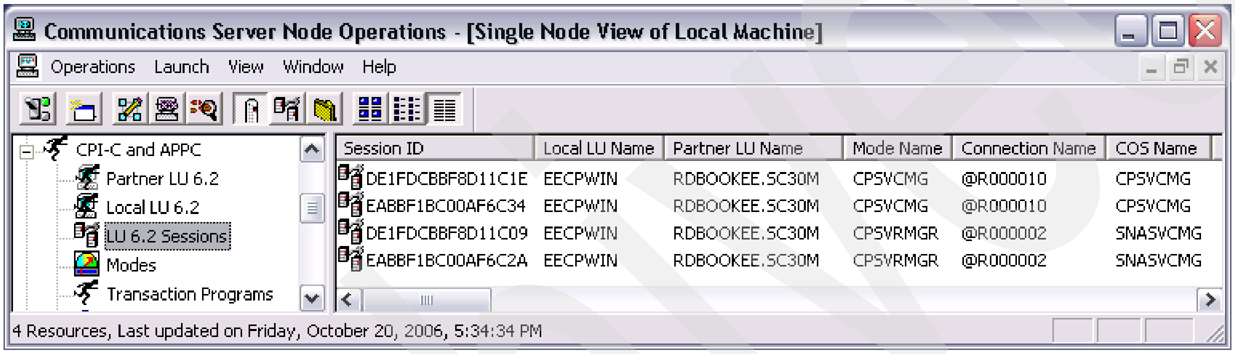

Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
Devices and DLCs
Connections
Host Connections
Peer Connections
Downstream Connections

Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Address
@D000003	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	10.10.1.231_4/64
EELINK01	Active	Active	IBMEEDLC	RDBOOKEE.SC30M	10.10.1.232_4/72 - SC30M-EE2.itso.ibm.com
EELINK02	Active	Active	IBMEEDLC	RDBOOKEE.SC31M	10.10.1.242_4/72 - SC31M-EE2.itso.ibm.com

3 Resources, Last updated on Friday, October 20, 2006, 5:33:49 PM

Figure 8-40 EELINK02 in a active state

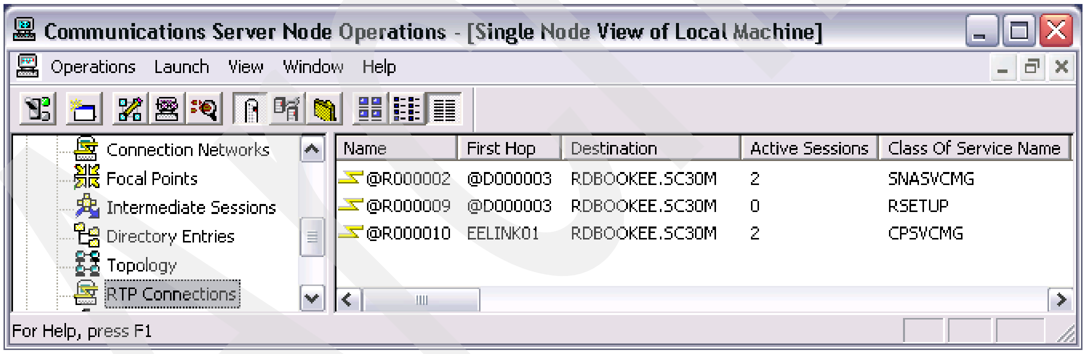
Now check the LU 6.2 sessions and the RTP connections (see Figure 8-41 and Figure 8-42).


Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
CPI-C and APPC
Partner LU 6.2
Local LU 6.2
LU 6.2 Sessions
Modes
Transaction Programs

Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name
DE1FDCBBF8D11C1E	EECPWIN	RDBOOKEE.SC30M	CPSVCMG	@R000010	CPSVCMG
EABBF1BC00AF6C34	EECPWIN	RDBOOKEE.SC30M	CPSVCMG	@R000010	CPSVCMG
DE1FDCBBF8D11C09	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG
EABBF1BC00AF6C2A	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR	@R000002	SNASVCMG

4 Resources, Last updated on Friday, October 20, 2006, 5:34:34 PM

Figure 8-41 LU 6.2 sessions after the reactivation of the EELINK01

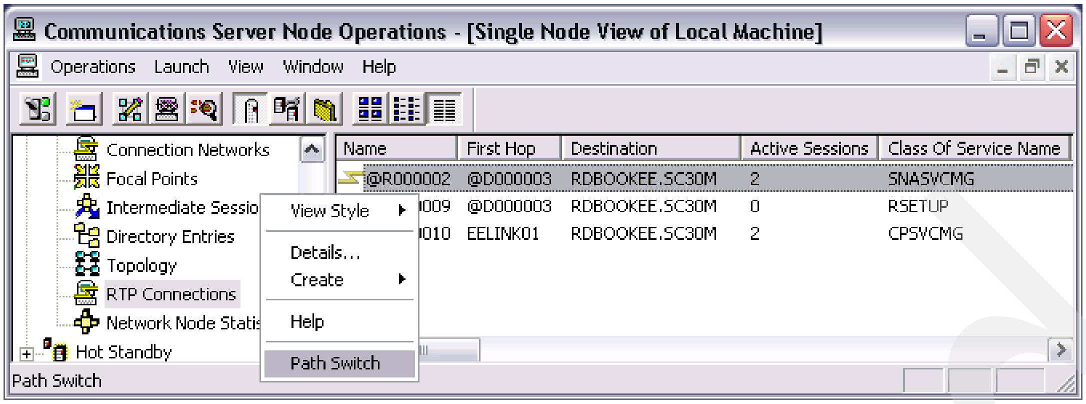

Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
Connection Networks
Focal Points
Intermediate Sessions
Directory Entries
Topology
RTP Connections

Name	First Hop	Destination	Active Sessions	Class Of Service Name
@R000002	@D000003	RDBOOKEE.SC30M	2	SNASVCMG
@R000009	@D000003	RDBOOKEE.SC30M	0	RSETUP
@R000010	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG

For Help, press F1

Figure 8-42 RTP connections after the reactivation of the EELINK02

Now, CP-CP sessions are established with SC30M because SC30M is the “preferred network node server”. The CPSVRMGR pipe shows no change because there is no path change on a @R000002 connection. You can change the RTP path to the most efficient path with the path switch command (see Figure 8-43).

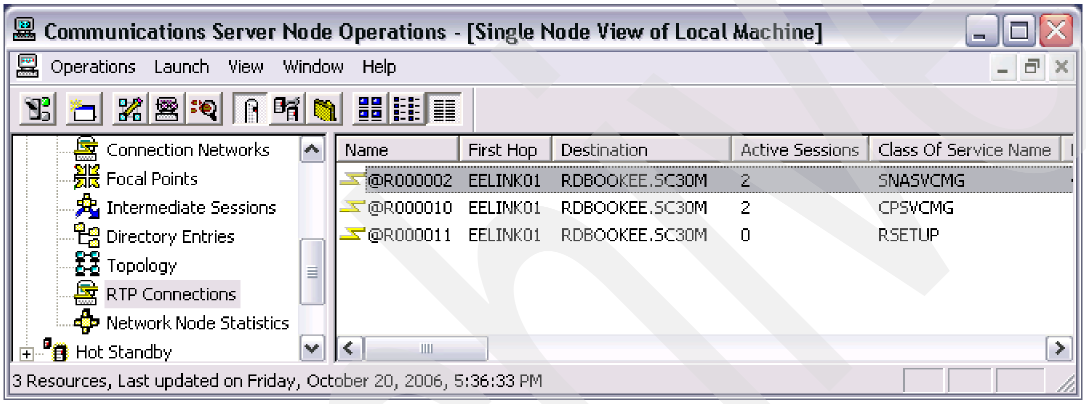


Name	First Hop	Destination	Active Sessions	Class Of Service Name
@R000002	@D000003	RDBOOKEE.SC30M	2	SNASVCMG
009	@D000003	RDBOOKEE.SC30M	0	RSETUP
010	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG

Path Switch

Figure 8-43 Path switch command

The path for the RTP connection (@R000002) is changed to the EELINK01 (see Figure 8-44).



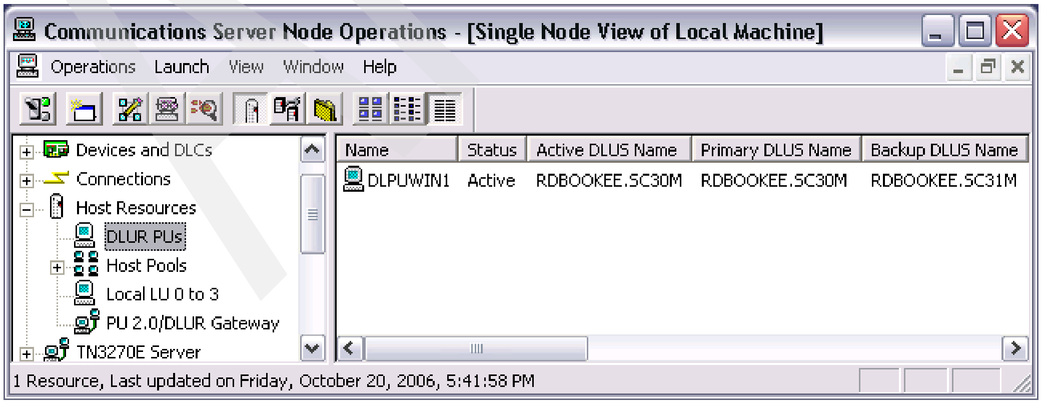
Name	First Hop	Destination	Active Sessions	Class Of Service Name
@R000002	EELINK01	RDBOOKEE.SC30M	2	SNASVCMG
@R000010	EELINK01	RDBOOKEE.SC30M	2	CPSVCMG
@R000011	EELINK01	RDBOOKEE.SC30M	0	RSETUP

3 Resources, Last updated on Friday, October 20, 2006, 5:36:33 PM

Figure 8-44 RTP connection is changed to the EELINK01

Verifying the backup DLUS

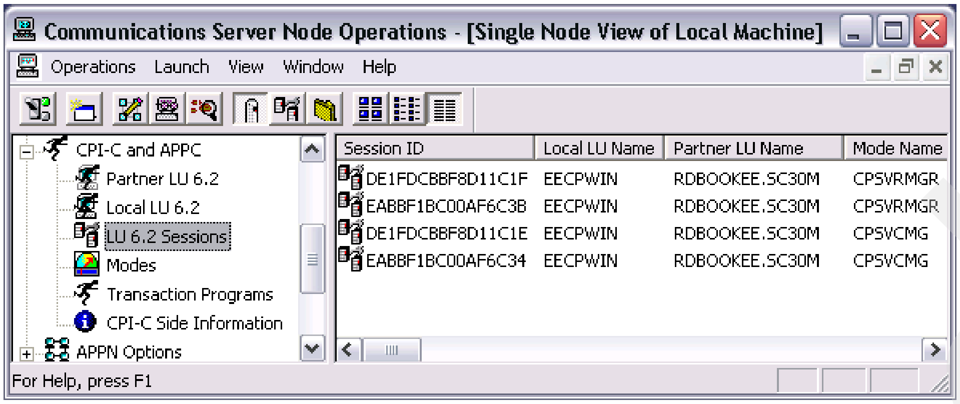
If the DLUS has a problem, the DLUS-DLUR sessions (CPSVRMGR) pipe might be broken. If you define the backup DLUS, then the backup DLUS can takeover the DLUS function in that case. We simulate this situation with the deactivation of the PU in the VTAM side.


Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
Devices and DLCs
Connections
Host Resources
DLUR PUs
Host Pools
Local LU 0 to 3
PU 2.0/DLUR Gateway
TN3270E Server

Name	Status	Active DLUS Name	Primary DLUS Name	Backup DLUS Name
DLPUWIN1	Active	RDBOOKEE.SC30M	RDBOOKEE.SC30M	RDBOOKEE.SC31M

1 Resource, Last updated on Friday, October 20, 2006, 5:41:58 PM

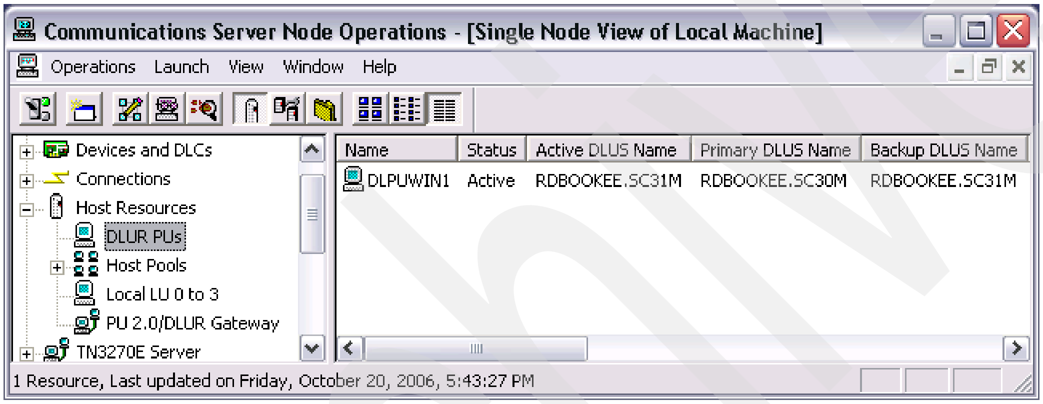
Figure 8-45 Active DLUS is the SC30M before the outage



Session ID	Local LU Name	Partner LU Name	Mode Name
DE1FDCBBF8D11C1F	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR
EABBF1BC00AF6C3B	EECPWIN	RDBOOKEE.SC30M	CPSVRMGR
DE1FDCBBF8D11C1E	EECPWIN	RDBOOKEE.SC30M	CPSVCMG
EABBF1BC00AF6C34	EECPWIN	RDBOOKEE.SC30M	CPSVCMG

Figure 8-46 DLUS-DLUR session is established with the SC30M before the outage

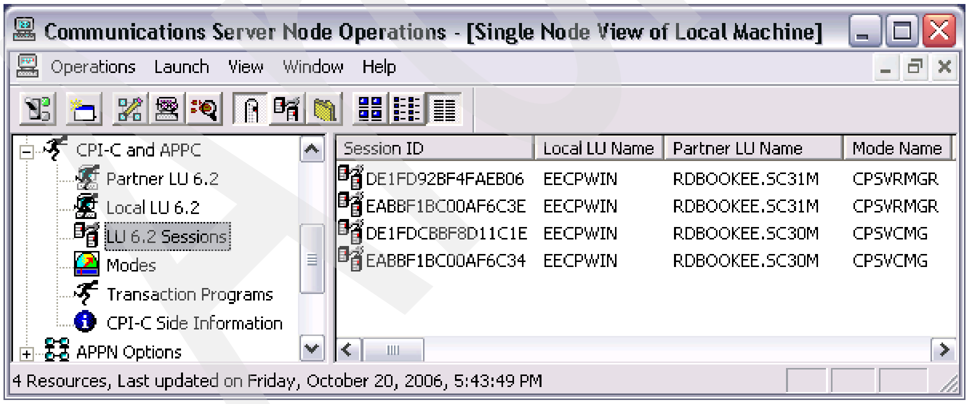
Figure 8-45 and Figure 8-46 show that the CPSVRMGR pipe is connected to the primary DLUS, SC30M. Now, we deactivate the PU in the SC30M.


Communications Server Node Operations - [Single Node View of Local Machine]
Operations Launch View Window Help
Devices and DLCs
Connections
Host Resources
DLUR PUs
Host Pools
Local LU 0 to 3
PU 2.0/DLUR Gateway
TN3270E Server

Name	Status	Active DLUS Name	Primary DLUS Name	Backup DLUS Name
DLPUWIN1	Active	RDBOOKEE.SC31M	RDBOOKEE.SC30M	RDBOOKEE.SC31M

1 Resource, Last updated on Friday, October 20, 2006, 5:43:27 PM

Figure 8-47 Active DLUS is the SC31M after the outage



Session ID	Local LU Name	Partner LU Name	Mode Name
DE1FD92BF4FAEB06	EECPWIN	RDBOOKEE.SC31M	CPSVRMGR
EABBF1BC00AF6C3E	EECPWIN	RDBOOKEE.SC31M	CPSVRMGR
DE1FDCBBF8D11C1E	EECPWIN	RDBOOKEE.SC30M	CPSVCMG
EABBF1BC00AF6C34	EECPWIN	RDBOOKEE.SC30M	CPSVCMG

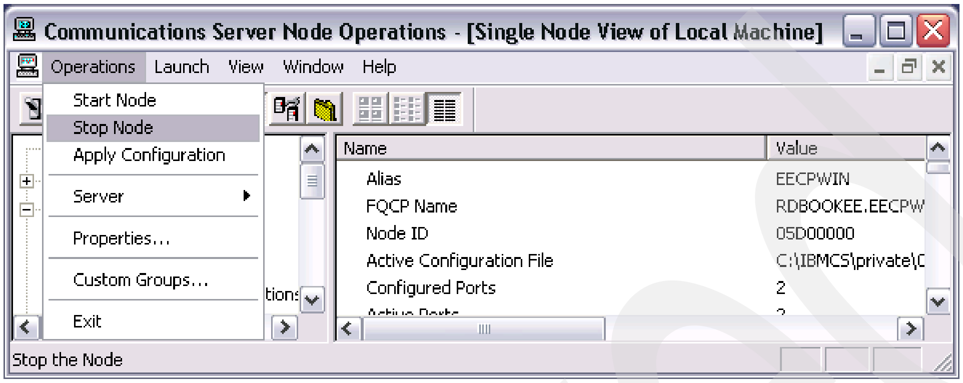
4 Resources, Last updated on Friday, October 20, 2006, 5:43:49 PM

Figure 8-48 DLUS-DLUR session is established with the SC31M after the outage

Now, SC31M is the active DLUS, that means all the dependent LUs now have SSCP-LU sessions with SC31M (see Figure 8-47 and Figure 8-48).

8.4.3 Stopping resources

We recommend that you stop the LU-LU sessions first using the batch file, and stop the node using **Operations** → **Stop node** menu in the SNA node operations window, as shown in Figure 8-49.



Name	Value
Alias	EECPWIN
FQCP Name	RDBOOKEE.EECPW
Node ID	05D00000
Active Configuration File	C:\IBMCS\private\C
Configured Ports	2

Figure 8-49 Stopping the node

8.4.4 Backing up your configuration

You must save your configuration in the *.acg file in the PRIVATE directory of your IBM Communications Server for Windows install directory (by default). Using a couple of acg files, You can easily switch from one configuration to the other. Always make a good backup of these acg files.

8.5 Diagnosing EE with Communications Server for Windows

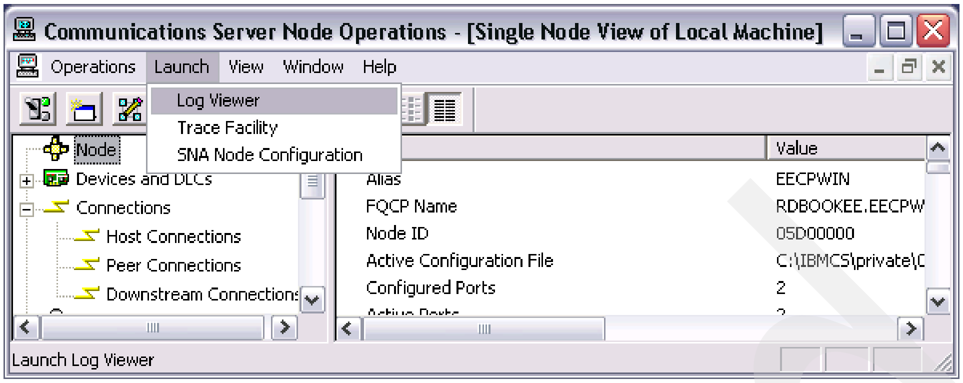
IBM Communications Server for Windows provides logging and tracing facility. In this section, we briefly explain how you can get log and trace information briefly. For further information about diagnosing IBM Communications Server for Windows, refer to *IBM Communications Server for Windows Quick Beginnings*, GC31-8424.

The topics covered in this section are:

- "Reviewing the log files"
- "Tracing CS for Windows"
- "Information bundler"
- Example 8-4 on page 303

8.5.1 Reviewing the log files

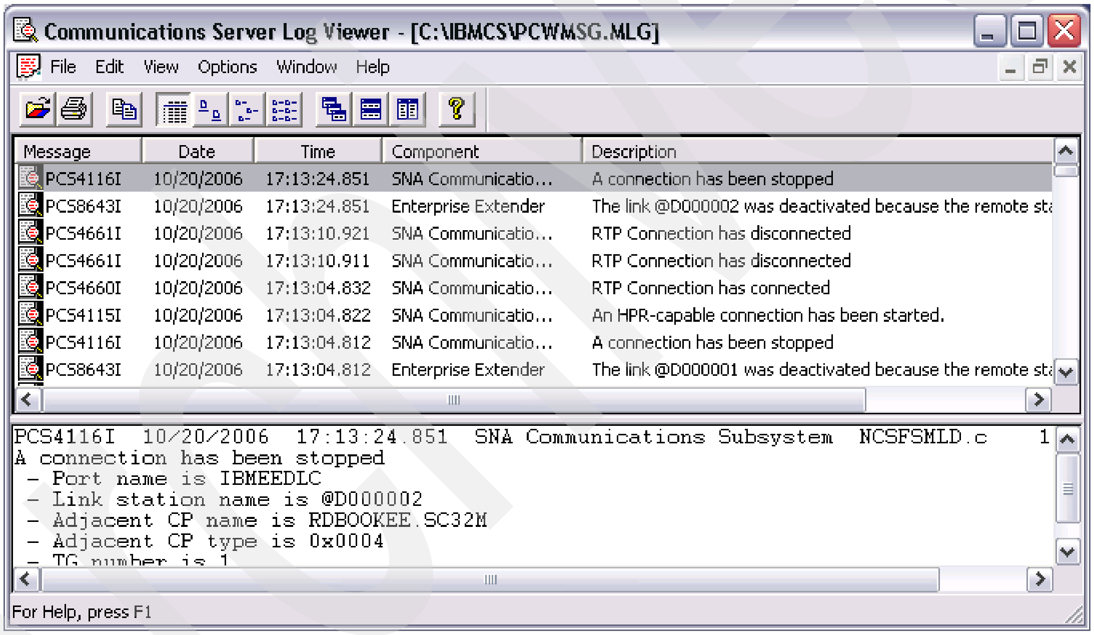
IBM Communications Server for Windows provides the Log viewer utility. You can start the log viewer with SNA node operations window such as shown in Figure 8-50.



	Value
Alias	EECPWIN
FQCP Name	RDBOOKEE.EECPW
Node ID	05D00000
Active Configuration File	C:\IBMCS\private\C
Configured Ports	2

Figure 8-50 Starting the Log viewer

With this log viewer, you can easily detect the various unusual events on the IBM Communications Server for Windows. This is the first step to debug issues for your IBM Communications Server for Windows (see Figure 8-51).



Message	Date	Time	Component	Description
PCS4116I	10/20/2006	17:13:24.851	SNA Communicatio...	A connection has been stopped
PCS8643I	10/20/2006	17:13:24.851	Enterprise Extender	The link @D000002 was deactivated because the remote sta
PCS4661I	10/20/2006	17:13:10.921	SNA Communicatio...	RTP Connection has disconnected
PCS4661I	10/20/2006	17:13:10.911	SNA Communicatio...	RTP Connection has disconnected
PCS4660I	10/20/2006	17:13:04.832	SNA Communicatio...	RTP Connection has connected
PCS4115I	10/20/2006	17:13:04.822	SNA Communicatio...	An HPR-capable connection has been started.
PCS4116I	10/20/2006	17:13:04.812	SNA Communicatio...	A connection has been stopped
PCS8643I	10/20/2006	17:13:04.812	Enterprise Extender	The link @D000001 was deactivated because the remote sta

```
PCS4116I  10/20/2006  17:13:24.851  SNA Communications Subsystem  NCSFSMLD.c    1
A connection has been stopped
 - Port name is IBMEEDLC
 - Link station name is @D000002
 - Adjacent CP name is RDBOOKEE.SC32M
 - Adjacent CP type is 0x0004
 - TG number is 1
```

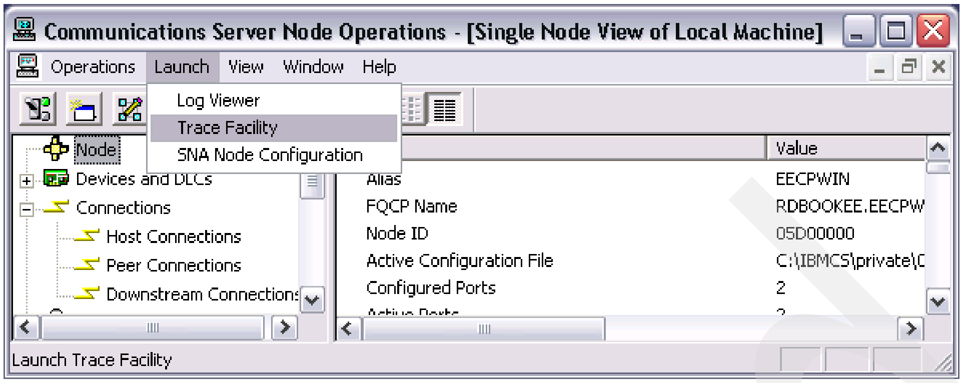
Figure 8-51 The Log viewer

8.5.2 Tracing CS for Windows

We explain the trace facility of IBM Communications Server for Windows and also explain using the IP packet trace briefly in this section.

Trace facility of IBM Communications Server for Windows

The IBM Communications Server for Windows provides the trace facility on many events. You can start the trace facility using SNA node operations, as shown in Figure 8-52.



	Value
Alias	EECPWIN
FQCP Name	RDBOOKEE.EECPW
Node ID	05D00000
Active Configuration File	C:\IBMCS\private\C
Configured Ports	2

Figure 8-52 Starting the trace facility

You can set up different kinds of traces with the trace facility. For more information about the trace facility of IBM Communications Server for Windows, refer to *IBM Communications Server for Windows Quick Beginnings*, GC31-8424. In Figure 8-53, we set up the trace of all the activities of EEDLC.

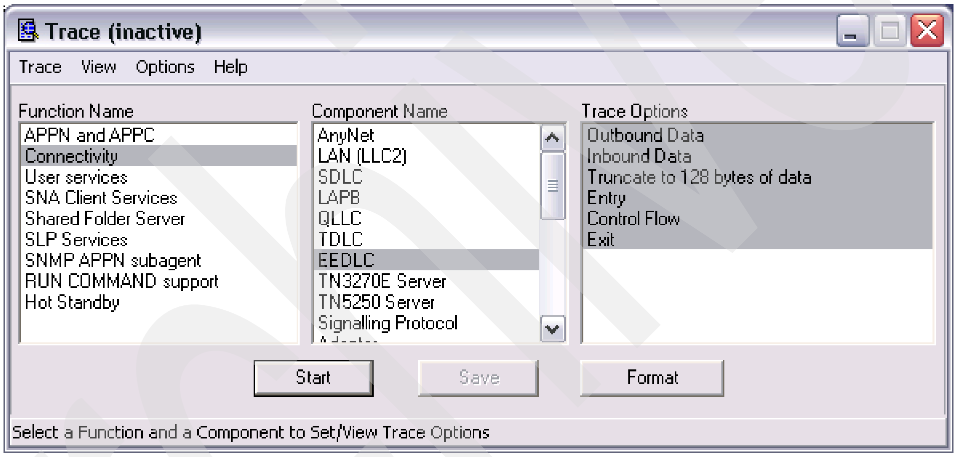


Figure 8-53 Tracing all the activities of EEDLC

Getting the IP packet trace

EE uses UDP for its transport. Sometimes, the problem may be on the UDP/IP side and not on the SNA/APPN side. In this case packet trace can provide a key to solving the problem.

There exist a lot of commercial and open source packet capture utilities and protocol analyzers. If you have a special packet trace utility and packet analyzer, then just use that tool for debugging EE problems. But if you do not have any packet trace utility or packet analyzer, try to use the Wireshark program (formerly Ethereal) for this purpose. Wireshark is an open source software which provides a good set of protocol analyzer functions including packet capturing with winpcap library. You can easily capture all the frames which pass your Ethernet adapter, and review all the frames which are captured with an easy GUI of Ethereal. For more information and downloading the application, visit the Web site:

http://www.wireshark.org

8.5.3 Information bundler

Information bundler gathers system files and specific trace and log files, as well as registry information such as the software installed or running on a machine.

To run the utility:

Select **Start** → **All Programs** → **IBM Communications Server** → **Problem Determination** → **Information Bundler**. A window will open containing information about the utility's progress.

The information bundler creates a file named cspdata.exe in the Communications Server installation subdirectory, which by default is `C:\IBMCS`.

8.5.4 Sample batch file gathering EE trace data

We provide a simple batch file to gather EE traces in IBM Communications Server for Windows. You have to create two batch files as shown in Example 8-4, one for starting the traces, and one for stopping the traces.

Example 8-4 Sample batch file for EE tracing

```
** batch file for turning on trace

cstrace reset
cstrace  start /f 4 /c 22  /o 10 20 /r
cstrace  apply /f 4 /c 33  /o 1
cstrace  apply /f 3 /c 7  /o 4
cstrace status

** batch file for turning off trace

cstrace stop
cstrace save rdbookee.trc
cstrace format rdbookee.trc
```

After execution of the first batch file, re-generate the situation that you want to trace. After re-generation of the problem, execute the second one, turning-off the batch file. You can find a redbookee.tlg log file in the same directory as batch files, this tlg file can be read with log viewer of IBM Communications Server for Windows.

9

Enabling EE in i5/OS

In this chapter, we explain the general configuration steps to enable EE in the System i operating system. i5/OS (V5R4M0 or later) provides the HPR/IP Data Link Control (DLC) to connect to other Enterprise Extender (EE) capable hosts over an IP infrastructure.

The topics covered in this chapter are as follows:

- “Overview of the Network Communications functions”
- “Implementation considerations”
- “Configuring EE in i5/OS”
- “Verifying and managing EE with i5/OS”
- “Diagnosing EE with i5/OS”

9.1 Overview of the Network Communications functions

EE support for System i was added to i5/OS V5R4, and does not exist in earlier releases of the i5/OS or OS/400® operating systems.

The i5/OS network communications also provides the following APPN functions:

- APPN End Node (APPN EN)
- APPN Network Node (APPN NN)
- APPN Branch Extender (APPN BE)
- Dependent LU Requestor (DLUR)
- Connection to APPN: Connection Network/Virtual Routing Node

In combination with DLUR the following functions are supported:

- Host Devices, including 3270 Emulation (*EML), Remote Job Entry (*RJE), and program-to-program communications (*PGM)
- SNA Passthrough (SNPT) upstream devices
- Distributed Host Command Facility (DHCF) devices (3270 pass through)
- Network Routing Facility (NRF) display and printer devices
- SNA Upline Facility (SNUF) devices, commonly used for VM/MVS Bridge (called "NJE" on mainframe hosts), DSNX, and user-written applications

The following functions are not supported with DLUR and, therefore, are not supported in an EE/DLUR configuration:

- SNA Primary LU2 Support (SPLS)
- Remote workstation controllers (5294, 5394, 5494)
- Retail (4690, 468x, 3641, 3684)
- Finance (FBBS or 470x)
- Any operation involving direct connection to remote hosts

9.2 Implementation considerations

The topics covered in this section are as follows:

- “APPN Node role and NETID”
- “High availability considerations”
- “Connection Network”

9.2.1 APPN Node role and NETID

Even though, the APPN NN function is available, the System i should be configured as an APPN EN or if downstream APPN ENs must be served, as an APPN BE Node. APPN NNs must participate in APPN Broadcast searches and Topology Database Updates and those node roles should be concentrated in the APPN backbone only.

9.2.2 High availability considerations

In order to achieve highly available connectivity, you must consider defining two connections to z/OS logical partitions (LPARs) acting as the primary and backup network node server (NNS). For dependent LU support, a primary and a backup Dependent LU Server (DLUS) should be configured.

Connections to primary and backup Network Node Server

Usually an End Node (EN) has a preferred network node server (NNS) and a backup NNS. You must define a connection for each, the preferred NNS and the backup NNS.

Primary and backup Dependent LU Server

For dependent LU support over EE, the DLUR function must be used in System i. For high availability reasons, two DLURs must be configured so that a backup DLUS can take over in case the primary DLUS fails.

9.2.3 Connection Network

In order to use the shortest available path between the System i and any other APPN node, a connection to a virtual routing node (VRN) must be configured. This allows for single hop High Performance Routing (HPR) routes over dynamically created links between two rapid transport protocol (RTP) endpoints instead of using two hop routes via intermediate automatic network routing (ANR) nodes.

9.2.4 Scenario used in this chapter

In our scenario, two z/OS V1R8 systems act as the primary and backup NNs for the System i APPN EN. Both z/OS Systems also provide DLUS services for the DLUR function in System i. The connectivity between System i and z/OS is through an IP WAN network and two firewalls. The IP WAN network to which the System i and the lab network used in the other chapters connect separate TCP/IP stacks in z/OS. Direct IP routing between the two IP stacks is prohibited for security reasons. As Virtual Telecommunications Access Method (VTAM) in z/OS cannot connect to two IP stacks at a time, we had to use a product called EEPROXY that forwards EE User Datagram Protocol (UDP) packets from one remote IP address in the external TCP/IP stack to VTAM's Virtual IP Address (VIPA) in the local secured TCP/IP stack. Because of these security restrictions, we were not able to use the EE connection networks in conjunction with the System i server, as we did with all the other platforms. To show that System i also supports connection networks, we defined a different VRN here.

Figure 9-1 shows the network with the System i connecting to the System z through firewalls.

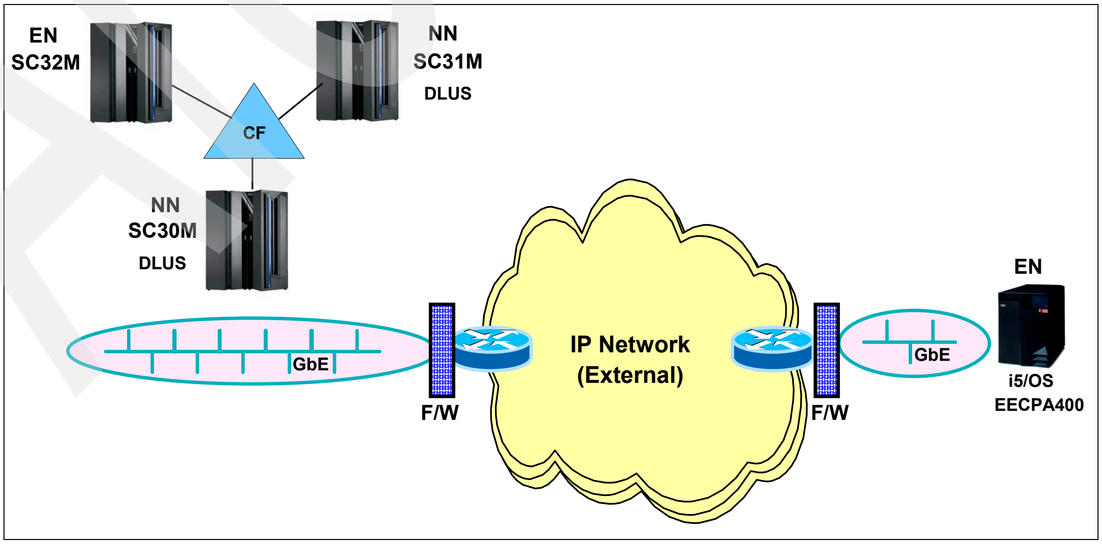


Figure 9-1 Scenario to connect System i to the EE network

9.3 Configuring EE in i5/OS

Configuring CS Linux includes many steps. The order of this section is:

- “Configuring the network attributes”
- “Configuring connectivity”
- “Configuring dependent LUs with DLUR”
- “Configuring LOGMODEs”

9.3.1 Configuring the network attributes

The general APPN parameters are listed under network attributes (dspneta). Example 9-1 shows the relevant parameters for EE.

Example 9-1 Display Network Attributes

```
Display Network Attributes
                                                          System:   MCEAS2L3
Current system name  . . . . . . . . . . . . . . :   MCEAS2L3
  Pending system name  . . . . . . . . . . . . . :
Local network ID . . . . . . . . . . . . . . . . :   RDBOOKEE 1
Local control point name . . . . . . . . . . . . :   EECPA400 1
Default local location . . . . . . . . . . . . . :   MCEAS2L3
Default mode . . . . . . . . . . . . . . . . . . :   BLANK
APPN node type . . . . . . . . . . . . . . . . . :   *ENDNODE 2
Data compression . . . . . . . . . . . . . . . . :   *NONE
Intermediate data compression  . . . . . . . . . :   *NONE
Maximum number of intermediate sessions  . . . . :   200
Route addition resistance  . . . . . . . . . . . :   128
Server network ID/control point name . . . . . . :   *LCLNETID   *ANY
...
Allow APPN virtual support . . . . . . . . . . . :   *YES 3
Allow HPR transport tower support  . . . . . . . :   *YES 4
Virtual controller autocreate APPC device limit  :   100
HPR path switch timers:                                   5
  Network priority . . . . . . . . . . . . . . . :   1
  High priority  . . . . . . . . . . . . . . . . :   2
  Medium priority  . . . . . . . . . . . . . . . :   4
  Low priority . . . . . . . . . . . . . . . . . :   8
```

The **dspneta** command can be used to display the network attribute of the System i.

Note the following explanations for Example 9-1:

- 1: This is the fully qualified control point (CP) name. It is divided into two parts. The first part is called NETID of a Systems Network Architecture (SNA) network, and the second part is the CP name of this node. In this book, we use `RDBOOKEE` as the NETID; and, CPNAME as `EECPA400`.
- 2: The APPN Node type to be configured. In this scenario, the System i will be an APPN EN.
- 3 and 4: Both parameters must be set to YES to support EE.
- 5: The HPR path switch timers define how long the RTP endpoint will wait for a successful path switch before terminating the HPR pipe.

9.3.2 Configuring connectivity

Now, we configure the connectivity resources for this node. Configuration steps will include:

- “Configuring APPC controllers to connect to VTAM NNs”
- “Configuring VTAM for connectivity to System i using EE”

Configuring APPC controllers to connect to VTAM NNs

You should define two Advanced Peer-to-Peer Communication (APPC) controllers to connect to the z/OS VTAM NNs.

Example 9-2 shows the APPC controller to the primary NNS SC30M.

Example 9-2 APPC Controller to define the connection to the primary NNs

```
Controller description . . . . . . :   DLPUSC30
Option . . . . . . . . . . . . . . :   *BASIC
Category of controller . . . . . . :   *APPC

Link type  . . . . . . . . . . . . :   *HPRIP
Online at IPL  . . . . . . . . . . :   *NO
Remote internet address  . . . . . :   9.12.4.212 1
Local internet address . . . . . . :   9.155.34.181
LDLC timers:
  LDLC retry count . . . . . . . . :   3
  LDLC retry timer . . . . . . . . :   15
  LDLC liveness timer  . . . . . . :   10

  LDLC link speed  . . . . . . . . . :   *CAMPUS
```

Note the following explanations for Example 9-2:

- **1** Remote internet address is the IP address that VTAM uses to communicate via EE. Local internet address is the IP address that i5/OS uses to communicate via EE.

> **Attention:** If firewalls are between the two systems, the rules must allow UDP traffic between both IP addresses for ports 12000 - 120004 in both directions. We recommend that you also allow ICMP ECHO traffic between the two IP addresses in order to use the **ping** command to test connectivity.

Configuring VTAM for connectivity to System i using EE

Given that VTAM is already EE enabled, it should have 5 UDP sockets at a static VIPA. This configuration is done in the external communications adapter (XCA) major node. Depending on the DYNPU setting in the XCA major node, you may need to code switched major nodes to define the PU statements that represent the APPC controllers in the System i.

- “Enterprise Extender XCA major node”
- “Switched PU definition of the i5/OS for the CPCP capable link via EE”

Enterprise Extender XCA major node

Example 9-3 shows how the XCA major node in VTAM would look like, if both VTAM and System i connected through the same TCP/IP stack.

Example 9-3 XCA major node for SC30M

```
EEXCA30         VBUILD TYPE=XCA
EEPORT30         PORT MEDIUM=HPRIP,                                    *
```

```
               IPTOS=(20,40,80,C0,C0),                                  *
               SRQTIME=15,SRQRETRY=3,LIVTIME=(10,60),
EEGVL301       GROUP DIAL=YES,CALL=INOUT,                               *
               IPADDR=9.12.4.212,                                       * 1
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                   * 2
               TGP=EEXTCAMP,                                            *
               KEEPACT=YES,                                             *
               DYNPU=YES,                                               *
               UNRCHTIM=30,                                             *
               AUTOGEN=(16,EEM30,EEN30)
```

Note the following explanations for Example 9-3:

- 1: The IP address in the GROUP matches the Remote Internet Address in the *APPC controller.
- 2: The VNNAME specifies the Connection Network Name to be used.

Switched PU definition of the i5/OS for the CPCP capable link via EE

Each NN system (SC30M, SC31M) has a switched PU definition for the System i. Example 9-4 shows the SWNET major node.

Example 9-4 PU Definition for link to i5/OS

```
EESWA400 VBUILD TYPE=SWNET
EEPUA400 PU    CPNAME=EECPA400,NETID=RDBOOKEE,TGN=6, 1 2                *
               CPCP=YES,HPR=YES,DISCNT=NO,DWACT=NO,DWINOP=NO,           *
               TGP=TGPAS400 3
EEPTA400 PATH  IPADDR=9.155.34.181,                                     *
               SAPADDR=4,                                               *
               GRPNM=EEGVL&SYSCLONE.1
```

Note the following explanation for Example 9-4:

- 1: The NETID and CP name of the System i. This name must be the same as the name you configured in node definition of i5/OS.
- 2: The TG number which is used for this link. It must match on both the sides, if it is predefined. In our environment we assigned TGN 6 for EE connections.
- 3: The TG Profile TGPAS400 contains a set of TG characteristics that are the same as in the System i. It must be available in VTAM's TGP table before the switched major node is activated.

9.3.3 Configuring dependent LUs with DLUR

We define dependent LUs and DLUR PUs in this section. The following topics are covered in this section:

- "DLUR definition in i5/OS"
- "DLUR definition in VTAM"

DLUR definition in i5/OS

For using the dependent LUs, you must define the following in CS Linux:

- "Host controller"
- "Dependents LUs"

Host controller

The DLUR PU is a PU type 2.0 which provides PU services to the dependent LUs. Example 9-5 shows the PU definitions in i5/OS.

Example 9-5 DLUR configuration in i5/OS

```
Display Controller Description

Controller description . . . . . . :   DLPUA400 1
Option . . . . . . . . . . . . . . :   *BASIC
Category of controller . . . . . . :   *HOST

Link type  . . . . . . . . . . . . :   *DLUR
Online at IPL  . . . . . . . . . . :   *NO
Switched connection  . . . . . . . :   *YES
Maximum frame size . . . . . . . . :   *LINKTYPE
Local exchange identifier  . . . . :   056A4001 2
Initial connection . . . . . . . . :   *DIAL
Dial initiation  . . . . . . . . . :   *LINKTYPE
Switched disconnect  . . . . . . . :   *NO
System job . . . . . . . . . . . . :   QCMNARB04
Message queue  . . . . . . . . . . :   *SYSVAL
Current message queue  . . . . . . :   QSYSOPR
  Library  . . . . . . . . . . . . :     QSYS
Controller description . . . . . . :   DLPUA400
Option . . . . . . . . . . . . . . :   *DEV
Category of controller . . . . . . :   *HOST



-------------------Attached Devices--------------------
DLLA4002      DLLA4003      DLLA4004      DLLA4005
```

Note the following explanations for Example 9-5:

- 1: This is the PU name for this DLUR PU for dependent LUs. We recommend that you use the same name as in VTAM because it simplifies problem diagnosis.
- 2: This is the PU identification number. VTAM has IDBLK (ID block) and IDNUM (ID number) parameters in its PU definition for this client, and the two definitions must be the same. In this case, the IDBLK is `056`, and IDNUM is `A4001`. This value must be unique in VTAM.

Dependents LUs

Now, we define the dependent LUs to the DLUR PU (see Example 9-6).

Example 9-6 HOST device defining the dependent LUs

```
Device description . . . . . . . . :   DLLA4002 1
Option . . . . . . . . . . . . . . :   *BASIC
Category of device . . . . . . . . :   *HOST

Local location address . . . . . . :   02 2
Remote location  . . . . . . . . . :   DLR3270
Online at IPL  . . . . . . . . . . :   *NO
Attached controller  . . . . . . . :   DLPUA400 3
Application type . . . . . . . . . :   *EML
```

```
Maximum length of request unit . . :   *CALC
Emulated device  . . . . . . . . . :   3278
Emulated keyboard  . . . . . . . . :   *UPPER
Emulated numeric lock  . . . . . . :   *NO
Emulation work station . . . . . . :   *ANY
End session with host  . . . . . . :   *UNBIND
Dependent location name  . . . . . :   *NONE
```

Note the following explanations for Example 9-6:

- 1: This name should be the same as the LU definition in VTAM.
- 2: This address must match the LOCADDR= definition in VTAM's SWNET major node.
- 3: This is the *host controller that we defined previously.

DLUR definition in VTAM

In VTAM, you should define the type 2.0 PU, and the dependent LUs in a switched major node. In our scenario, we make a switched major node as shown in Example 9-7.

Example 9-7 Switched major node definition for DLUR

```
DLSWA400 VBUILD TYPE=SWNET
DLPUA400 PU   PUTYPE=2,USSTAB=USSSNAEE,ANS=CONT, 1                   *
               IDBLK=056,IDNUM=A4001, 2                              *
               MODETAB=ALLMODES,DLOGMOD=DYNHIGH
DLLA4002 LU   LOCADDR=2 3
DLLA4003 LU   LOCADDR=3
DLLA4004 LU   LOCADDR=4
DLLA4005 LU   LOCADDR=5
DLLA4006 LU   LOCADDR=6
```

Note the following explanations for Example 9-7:

- 1: ANS=CONT is required for DLUR PUs. If the default value (ANS=STOP) is used, SSCP-PU session and SSCP-LU sessions will be broken in case the primary DLUS is lost. This results in session outages during takeover to backup DLUS.
- 2: IDBLK and IDNUM must match PU ID on the host controller in i5/OS.
- 3: The name of the LU may be different from the one in i5/OS, but the LOCADDR must be the same as the Local Location Address on the host device definition in i5/OS.

9.3.4 Configuring LOGMODEs

We recommend that you define a new logmode #CONNECT that maps to an APPNCOS of #CONNECT. This logmode does not exist in most platforms, but proved to be helpful in combination with the APING command so that an #CONNECT HPR pipe can be used for APING sessions also.

Example 9-8 shows the new logmode #CONNECT created with **wrkmodd.**

Example 9-8 #CONNECT logmode definition

```
Mode description . . . . . . . . . :   #CONNECT

Class-of-service . . . . . . . . . :   #CONNECT
Maximum sessions . . . . . . . . . :   8
Maximum conversations  . . . . . . :   8
```

```
Locally controlled sessions  . . . :   4
Pre-established sessions . . . . . :   0
Maximum inbound pacing value . . . :   *CALC
Inbound pacing value . . . . . . . :   3
Outbound pacing value  . . . . . . :   3
Maximum length of request unit . . :   *CALC
Data compression . . . . . . . . . :   *NETATR
Inbound data compression . . . . . :   *RLE
Outbound data compression  . . . . :   *RLE
Session level encryption . . . . . :   *NONE
Text . . . . . . . . . . . . . . . :   This Mode is to be used for APING
```

9.4 Verifying and managing EE with i5/OS

In this section, we explain how to start and verify the various SNA resources in i5/OS.

The section covers the following topics:

- “Starting resources”
- “Verifying resources”

9.4.1 Starting resources

Starting each resource is very simple with xsnaadmin program. You only have to highlight the resource which you want to start, and click the start button. By default, the node is already started when the xsnaadmin program is up and running.

The topics covered here are:

- Starting the APPC controllers
- Starting the host controller

Starting the APPC Controllers

This section describes how to start the APPC controller and the associated messages.

Example 9-9 shows where the APPC controllers are activated.

Example 9-9 Starting APPC controllers in System i

```
Work with Configuration Status                MCEAS2L3
                                                            11/22/06  15:51:11
Position to  . . . . .               Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status                -------------Job--------------
1    DLPUSC30          VARIED OFF
1    DLPUSC31          VARIED OFF
```

Example 9-10 shows the resulting messages.

Example 9-10 Messages when APPC controller is activated successfully

```
Controller DLPUSC30 contacted on line *N.
New alert focal point is RDBOOKEE.SC30M.

IST590I  CONNECTIN  ESTABLISHED FOR PU EEPUA400 ON LINE EEM3000E
IST1086I  APPN CONNECTION FOR RDBOOKEE.EECPA400 IS ACTIVE - TGN =   6
IST1488I  ACTIVATION   OF RTP CNR000DB AS PASSIVE TO RDBOOKEE.EECPA400
IST1096I  CP-CP SESSIONS WITH RDBOOKEE.EECPA400 ACTIVATED
```

Starting Host controller

This section describes how to start the host controller and the associated messages.

Example 9-11shows where the host controllers are activated.

Example 9-11 Starting Host controller

```
Work with Configuration Status                MCEAS2L3
                                                            11/22/06  15:59:13
Position to  . . . . .              Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status                ------------Job--------------
1    DLPUA400          VARIED OFF
       DLLA4002        VARIED OFF
       DLLA4003        VARIED OFF
       DLLA4004        VARIED OFF
       DLLA4005        VARIED OFF

Vary on completed for controller DLPUA400
```

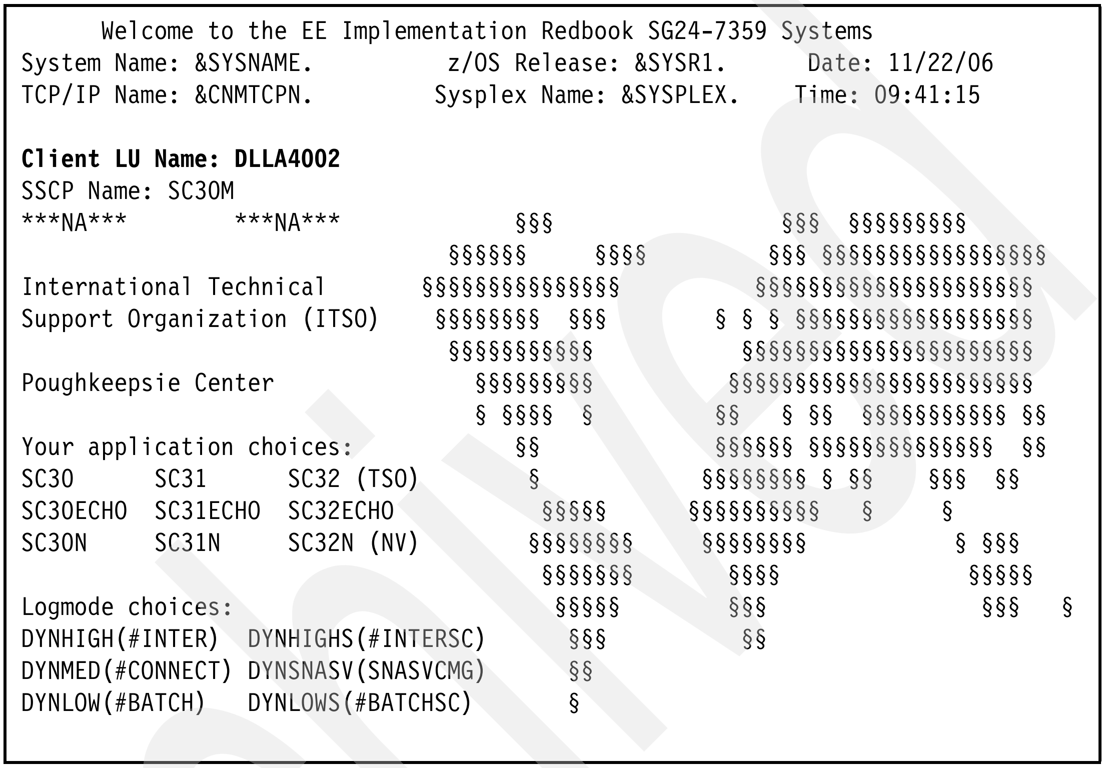
Example 9-12 shows the resulting VTAM messages.

Example 9-12 Resulting VTAM messages

```
IST1488I  ACTIVATION   OF RTP CNR000DC AS PASSIVE TO RDBOOKEE.EECPA400
IST1488I  ACTIVATION   OF RTP CNR000DD AS PASSIVE TO RDBOOKEE.EECPA400
IST1883I  SESSION ESTABLISHED WITH DLPUA400 - DLUR RDBOOKEE.EECPA400
IST093I  DLPUA400 ACTIVE
D NET,ID=DLPUA400,E
IST097I DISPLAY ACCEPTED
IST075I NAME = DLPUA400, TYPE = PU_T2 663
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1043I CP NAME = ***NA*** - CP NETID = RDBOOKEE - DYNAMIC LU = YES
IST1589I XNETALS = YES
IST1354I DLUR NAME = EECPA400          MAJNODE = DLSWA400
IST136I SWITCHED SNA MAJOR NODE = DLSWA400
IST1934I IDBLK = 056 IDNUM = A4001
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
IST1657I MAJOR NODE VTAMTOPO = REPORT
IST355I LOGICAL UNITS:
```

```
IST080I DLLA4002 ACTIV      DLLA4003 ACTIV      DLLA4004 ACTIV
IST080I DLLA4005 ACTIV      DLLA4006 CONCT
IST314I END
```

Starting a 3270 Emulation using an LU from the Host controller displays the VTAM welcome screen USSMSG10 on one of the System i LUs (see Figure 9-2).



```
      Welcome to the EE Implementation Redbook SG24-7359 Systems
System Name: &SYSNAME.          z/OS Release: &SYSR1.      Date: 11/22/06
TCP/IP Name: &CNMTCPN.         Sysplex Name: &SYSPLEX.    Time: 09:41:15

Client LU Name: DLLA4002
SSCP Name: SC30M
***NA***        ***NA***             §§§                 §§§  §§§§§§§§§
                                §§§§§§     §§§§         §§§ §§§§§§§§§§§§§§§§§
International Technical       §§§§§§§§§§§§§§§          §§§§§§§§§§§§§§§§§§§§§
Support Organization (ITSO)    §§§§§§§§  §§§        § § § §§§§§§§§§§§§§§§§§§
                                §§§§§§§§§§§           §§§§§§§§§§§§§§§§§§§§§§
Poughkeepsie Center               §§§§§§§§§          §§§§§§§§§§§§§§§§§§§§§§§
                                  § §§§§  §         §§   § §§  §§§§§§§§§§§ §§
Your application choices:            §§             §§§§§§ §§§§§§§§§§§§§§  §§
SC30      SC31      SC32 (TSO)        §            §§§§§§§§ § §§    §§§  §§
SC30ECHO  SC31ECHO  SC32ECHO           §§§§§      §§§§§§§§§§   §     §
SC30N     SC31N     SC32N (NV)        §§§§§§§§     §§§§§§§§           § §§§
                                       §§§§§§§       §§§§              §§§§§
Logmode choices:                        §§§§§       §§§                §§§   §
DYNHIGH(#INTER)  DYNHIGHS(#INTERSC)      §§§         §§
DYNMED(#CONNECT) DYNSNASV(SNASVCMG)      §§
DYNLOW(#BATCH)   DYNLOWS(#BATCHSC)       §
```

Figure 9-2 STREML3270 EMLCTL(DLPUA400) yields USS Message 10

9.4.2 Verifying resources

You can easily monitor SNA activities with xsnaadmin main window or **snaadmin** command.

In this section we explain the following:

- "Verifying the connectivity"
- "Verifying the sessions"
- "RTP path switch"

Verifying the connectivity

Example 9-13 shows the WORK Configuration STATUS panel.

Example 9-13 Verify the connectivity

```
Work with Configuration Status                MCEAS2L3
                                                            11/22/06  16:49:33
Position to  . . . . .                Starting characters


Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...
```

```
Opt  Description       Status                 -------------Job--------------
     DLPUA400          ACTIVE     2
       DLLA4002        ACTIVE                 QPADEV000B  MBURKHAR    227808
       DLLA4003        VARIED ON
       DLLA4004        VARIED ON
       DLLA4005        VARIED ON
     DLPUSC30          VARIED ON  1
     DLPUSC31          VARIED ON


                                                                        Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys
```

Note the following explanations for Example 9-13:

- 1: Both APPC controllers are VARIED ON.
- 2:The Host controller is ACTIVE and the first LU is also ACTIVE.

Verifying the sessions

Example 9-14 shows the active sessions.

Example 9-14 Active sessions

```
Work with APPN Locations
                                                            System:   MCEAS2L3
Type options, press Enter.
  5=Work with sessions   8=Work with RTP connections
  12=Work with configuration status

                     -------Remote-------    -------Local--------
                     Location    Network     Location    Network     Number of
Opt    Controller    Name        ID          Name        ID          Sessions
1      DLPUSC31
                     DLLA4002    RDBOOKEE    SC32TS01    RDBOOKEE     1
                     SC31M       RDBOOKEE    EECPA400    RDBOOKEE     2

Opt    Controller    Name        ID          Name        ID          Sessions
2      DLPUSC30
                     SC30M       RDBOOKEE    EECPA400    RDBOOKEE     2
```

Note the following explanation for Example 9-14:

- 1: APPC controller DLPUSC31 shows 1 session between DLLA4002 and SC32TSO1. This is an LU-LU session to TSO running on SC32. Over the same controller there are also 2 sessions to SC31M.
- 2: APPC controller DLPUSC30 shows 2 sessions to SC30M.

Example 9-15 shows the logmodes of those sessions.

Example 9-15 Active sessions and their logmodes

```
Work with Sessions for APPN Locations
                                                            System:   MCEAS2L3
```

```
Type options, press Enter.
  5=Work with job   8=Display session details   9=Display mode status
  12=Work with configuration status   14=Display APPN information

                   ------Remote------  ------Local-------
                   Location  Network   Location  Network
Opt  Description   Name      ID        Name      ID        Mode
     DLPUSC31
1      SC31M       SC31M     RDBOOKEE  EECPA400  RDBOOKEE  CPSVCMG
       SC31M       SC31M     RDBOOKEE  EECPA400  RDBOOKEE  CPSVCMG
------Remote------  ------Local-------
                   Location  Network   Location  Network
Opt  Description   Name      ID        Name      ID        Mode
     DLPUSC30
2      SC30M       SC30M     RDBOOKEE  EECPA400  RDBOOKEE  CPSVRMGR
       SC30M       SC30M     RDBOOKEE  EECPA400  RDBOOKEE  CPSVRMGR
```

Note the following explanation for Example 9-15:

- 1: The logmode CPSVCMG is used by the CP-CP sessions between the local CP and the NNS. In this case, SC31M is the current NNS.
- 2: The logmode CPSVRMGR is used by the DLUS/DLUR sessions. Those carry the SSCP-PU and SSCP-LU flows between the DLUR and DLUS. In this case SC30M is the current DLUS.

You will observe that, DLUS and NNS can be different.

RTP path switch

In case of a failure HPR pipes will switch around failures. We simulate this by stopping the APPC controller which is used by the #INTER pipe used for the TSO session.

Example 9-16 shows the existing pipe TCID 000050 towards SC32M using DLPUSC31.

Example 9-16 RTP path switch of TCID 000050

```
                          Work with RTP Connections
                                                              System:   MCEAS2L3
Type options, press Enter.
  5=Work with sessions   8=Work with APPN locations   9=Path switch
  10=End connection      12=Work with configuration status ...


                     ----RTP Partner-----
                     Control     Network                Class of    Number of
Opt    Controller    Point       ID         TCID        Service     Sessions
1      DLPUSC31
                     SC31M       RDBOOKEE   000030      -CPSVCMG      2
                     SC32M       RDBOOKEE   000050      #INTER        1
```

Note the following explanation for Example 9-16:

- 1: The RTP pipe with TCID 000050 was using DLPUSC31 to get to SC32M. On SC31M the link was terminated.

Example 9-17 shows how the link was terminated from VTAM.

Example 9-17 Terminating the link from SC31M to System i

```
V NET,INACT,ID=EEPUA400,I
IST097I VARY ACCEPTED
IST1196I APPN CONNECTION FOR RDBOOKEE.EECPA400 INACTIVE - TGN = 6    2
IST1097I  CP-CP SESSION WITH RDBOOKEE.EECPA400 TERMINATED
IST1280I  SESSION TYPE = CONWINNER - SENSE = 08420001
IST1097I  CP-CP SESSION WITH RDBOOKEE.EECPA400 TERMINATED
IST1280I  SESSION TYPE = CONLOSER  - SENSE = 08420001
IST1488I  ACTIVATION   OF RTP CNR000DE AS PASSIVE TO RDBOOKEE.EECPA400
IST1096I  CP-CP SESSIONS WITH RDBOOKEE.EECPA400 ACTIVATED  3
```

Note the following explanation for Example 9-17:

- 2: The link became inactive.
- 3: CP-CP sessions were activated with SC30M immediately.

Example 9-18 shows the VTAM messages indicating a path switch occurred.

Example 9-18 HPR pipe on SC32M switched to alternate route

```
IST1494I  PATH SWITCH STARTED   FOR RTP CNR0004D TO RDBOOKEE.EECPA400
IST1937I  PATH SWITCH REASON: INITIATED BY REMOTE PARTNER
IST1494I  PATH SWITCH COMPLETED FOR RTP CNR0004D TO RDBOOKEE.EECPA400
IST1480I  RTP END TO END ROUTE - RSCV PATH
IST1460I  TGN  CPNAME            TG TYPE      HPR
IST1461I    6  RDBOOKEE.SC30M     APPN         RTP
IST1461I    6  RDBOOKEE.EECPA400  APPN         RTP
```

The HPR pipe switched to the alternate route.

Example 9-19 shows how the RTP pipe TCID 000050 survived the link failure and is now using the APPC controller DLPUSC30.

Example 9-19 TCID 000050 using alternate APPC controller

```
                          Work with RTP Connections
                                                          System:   MCEAS2L3
Type options, press Enter.
  5=Work with sessions   8=Work with APPN locations   9=Path switch
  10=End connection      12=Work with configuration status ...


                   ----RTP Partner-----
                   Control     Network                Class of    Number of
Opt    Controller  Point       ID          TCID       Service     Sessions
       DLPUSC30
                   SC30M       RDBOOKEE    000020     -CPSVCMG     2
                   SC30M       RDBOOKEE    000040     SNASVCMG     2
                   SC32M       RDBOOKEE    000050     #INTER       1
```

This display shows that the TCID 00050 is now using DLPUSC30, so the pipe was successfully switched away from the failing APPC controller DLPUSC31.

The LU-LU session did not notice the path switch.

9.5 Diagnosing EE with i5/OS

Most EE problems require that you take an IP packet trace on both the sides of the EE endpoints. Here is how such a trace is taken in System i.

9.5.1 Tracing in i5/OS

The following steps are required to gather an IP trace:

1. DLTCMNTRC CFGOBJ(PETHLIN) CFGTYPE(*LIN)
2. DLTCMNTRC CFGOBJ(PETHLIN) CFGTYPE(*LIN) to delete old traces
3. STRCMNTRC CFGOBJ(PETHLIN) CFGTYPE(*LIN) MAXSTG(32M) to start a trace
4. ENDCMNTRC CFGOBJ(PETHLIN) CFGTYPE(*LIN) to stop a trace
5. PRTCMNTRC CFGOBJ(PETHLIN) CFGTYPE(*LIN) CODE(*ASCII) FMTTCP(*YES) FMTHPRIP(*YES) FMTLDLCIP(*YES) to format the trace

10

Implementing multiple VIPAs on z/OS

The purpose of this chapter is to assist in planning, implementing, and managing a z/OS EE environment with multiple Virtual IP Addresses (VIPAs). It explains how static VIPAs represent EE endpoints to z/OS Communications Server, where multiple VIPAs might apply, how to use multiple VIPAs to your advantage, and rules to consider when configuring them.

Sample definitions for both single VIPA and multiple VIPA EE configurations are depicted, describing the differences between the two approaches and considerations for each. Procedures used for verification and diagnosis in a multiple EE VIPAs environment are included, showing some strategic commands and sample output. Some examples show the use of either the host name or IP address when referencing a VIPA.

There is no difference between a static VIPA defined for non-EE applications and a static VIPA used for Enterprise Extender. All static VIPAs are defined the same way with a common set of statements regardless of their intended use. In this chapter, we use the terms *EE VIPA*, *local EE VIPA*, and *remote EE VIPA* to simply indicate that the VIPA is considered dedicated to Enterprise Extender's use. These terms do not imply a specially defined VIPA in any other respect.

The following topics are discussed in this chapter:

- "Overview of Enterprise Extender VIPA requirements"
- "Configuring a single EE VIPA"
- "Configuring multiple EE VIPAs"
- "Activation of multiple EE VIPAs"
- "Verification of the multiple EE VIPAs environment"
- "Diagnosis of the multiple EE VIPAs environment"

10.1 Overview of Enterprise Extender VIPA requirements

This section includes the following EE connectivity topics:

- “The Static VIPA requirement for EE”
- “Rules when using multiple EE VIPAs”
- “Configurations where multiple EE VIPAs might be used”

10.1.1 The Static VIPA requirement for EE

EE supports static VIPAs only, not dynamic VIPAs. The preferred method of managing most IP related applications is through the combined mechanisms of Dynamic VIPA and Sysplex Distribution, where each application can be assigned its own dedicated dynamic VIPA. However, any VIPA defined for use with EE, must be a static VIPA.

> **Note:** EE always uses its assigned static VIPA as the source IP address for all outbound EE traffic, regardless of the SOURCEVIPA setting. EE ignores and does not use any stack SOURCEVIPA that might be configured.

10.1.2 Rules when using multiple EE VIPAs

There are some important rules to be considered when using multiple EE VIPAs. They include the following:

- Every EE connection must use a unique IP address pair. 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. You can define these parallel EE connections by using a different local static VIPA address or remote IP address (or both) for each connection.
- Do not code different SAP values to define parallel TGs. The use of different SAP values to achieve parallel TGs is not supported. Use a different IP address pair for each of the TGs instead of different SAP values. This restriction applies only to parallel EE connections that are initiated by VTAM. An inbound parallel EE connection using the same IP address pair and different SAP values, will be accepted by VTAM.
- There is no limit to the number of EE connections that can be established using unique IP address pairs.
- The multiple local EE 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 does not use multiple stacks concurrently.
- If the use of the host name function is planned, it is crucial for consistent and predictable EE connection establishment that a given host name resolve to a single IP address. When multiple EE VIPAs are used, *each* EE VIPA must be associated with its own unique host name. DNS services must return one and only one EE VIPA address for any given EE host name.
- The use of multiple EE VIPAs dictates that the IPADDR or HOSTNAME of the VIPAs be coded on the GROUP statements within the EE external communications adapter (XCA) major node instead of the ATCSTRxx start list. The ATCSTRxx start list accommodates only one address or host name, not multiples. To be consistent then, simply code this information in the XCA major node regardless of how many EE VIPAs are being used.

10.1.3 Configurations where multiple EE VIPAs might be used

There are a number of design scenarios where multiple EE VIPAs could be used, such as:

- Different VIPAs to support different business partners
- Use of multiple connection networks
- Concurrent IPv4 and IPv6 connections terminating in the same VTAM
- Support of multiple security zones
- Utilize different routes through a redundant IP infrastructure
- Parallel TGs between two endpoints, use of DSAP is not supported
- Migration to a new IP addressing scheme, old and new in parallel
- Statistics and accounting based on COS and TGNs

Figure 10-1 depicts the strategic definitions required for establishing a multiple EE VIPA environment and the relationship between the VTAM and TCP/IP parameters used.

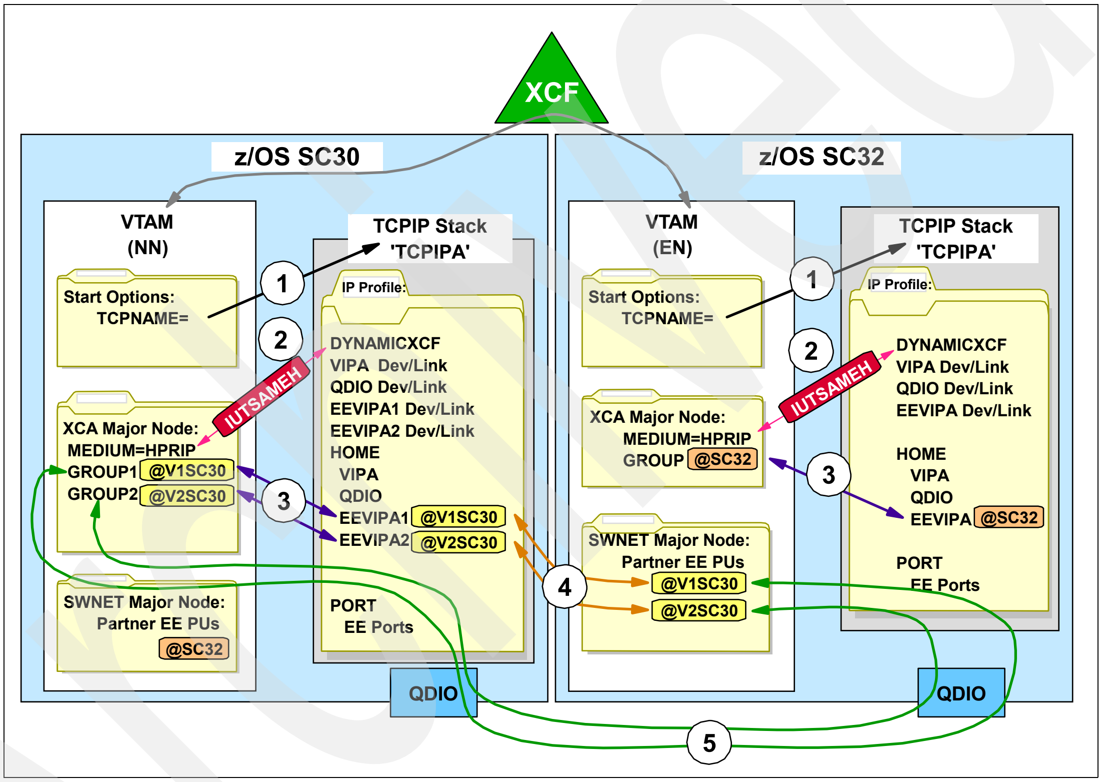


Figure 10-1 Relationship between VTAM and TCP/IP for multiple VIPA support

Note the following explanation for Figure 10-1:

1. Establish stack affinity between VTAM and TCP/IP using the TCPNAME start option.
2. Use the DYNAMICXCF statement to create the IUTSAMEH device used for VTAM-to-TCIP/IP communication.
3. The EE VIPAs are defined in TCP/IP using the DEVICE, LINK, and HOME statements. The HOME statement assigns an IP address to each EE VIPA. These IP addresses are then specified on the GROUP statements within the EE XCA major node.
4. Any remote network partner wanting to establish an EE connection to your local node specifies the VIPA addresses of your local node in its SWNET major node. The remote partner uses its SWNET major node to access your node through its TCP/IP stack.

5. The EE connection is established over the IP network between an IP address pair: one IP address on the remote system to one IP address on your local system. For each EE connection, a unique IP address pair must be used. For multiple parallel connections, multiple EE VIPAs must be defined.

10.2 Configuring a single EE VIPA

In order to understand the implementation of multiple EE VIPAs, a discussion on setting up a single EE VIPA is necessary. The single EE VIPA configuration is then used as a base on which to build the multiple EE VIPA environment. Figure 10-2 shows the necessary definitions in your local VTAM, TCP/IP, and OMPROUTE to define a single EE VIPA that can be used to support an EE connection. The figure is followed by configuration examples showing how to set up a single local EE VIPA. The examples include a SWNET major node on your system that points to a remote partner's EE VIPA.

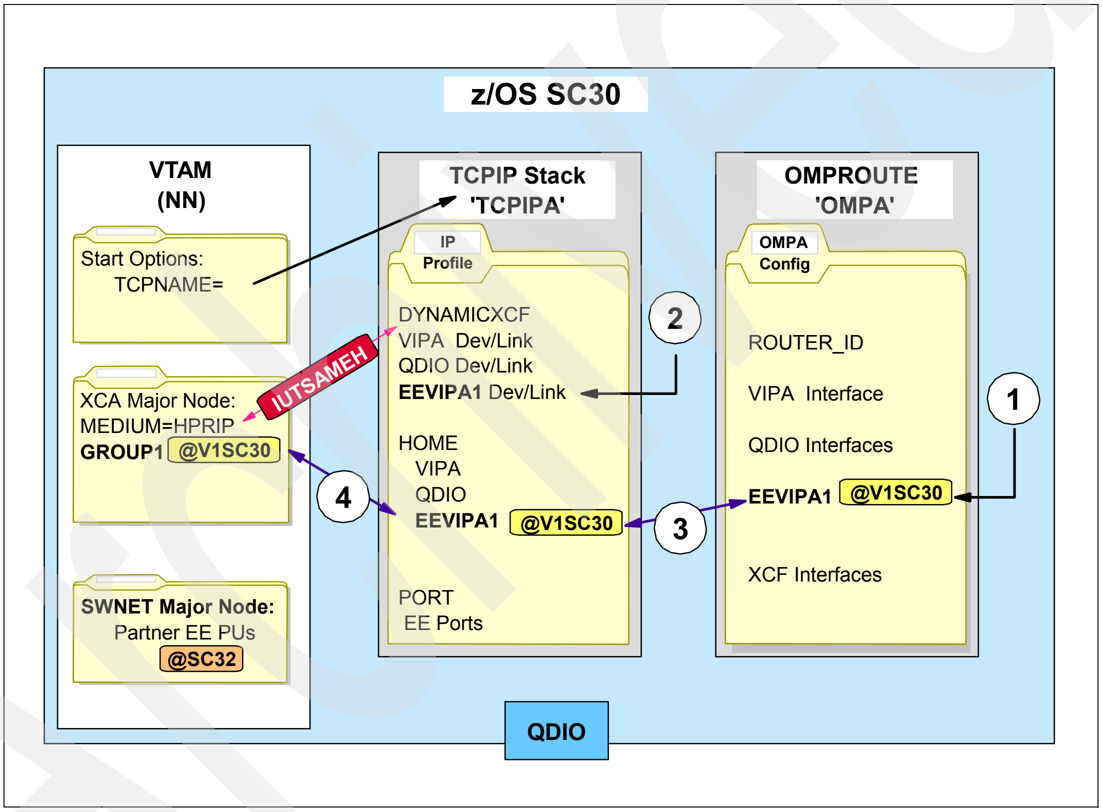


Figure 10-2 Defining a single EE VIPA on your local system

Note the following explanation for Figure 10-2:

- 1: Define the EE VIPA to OMPROUTE using the OSPF-INTERFACE statement, specifying the Link name and assigned IP address.
- 2: Define the EE VIPA to TCP/IP using the DEVICE, LINK, and HOME statements.
- 3: The Link name and IP address used in OMPROUTE must match those used in the HOME statement.
- 4: Specify the EE VIPA's IP address (or host name) on a GROUP statement within the EE XCA major node in VTAM. The IP address must be the one assigned in the HOME list in TCP/IP.

Do not forget to assign VTAM's stack affinity by using the VTAM TCPNAME start option. The DYNAMICXCF statement in TCP/IP creates the IUTSAMEH device used by VTAM-to-TCP/IP communication.

This section includes the following topics for setting up a single local EE VIPA:

- "Defining a local EE VIPA in the TCP/IP profile"
- "Defining a local EE VIPA in OMPROUTE configuration file"
- "Specifying a local VIPA in the VTAM ATCSTRxx start list"
- "Specifying a local VIPA in the EE XCA major node"
- "Specifying a remote VIPA in the EE SWNET major node"

10.2.1 Defining a local EE VIPA in the TCP/IP profile

Example 10-1 shows how to define an EE VIPA in TCP/IP.

Example 10-1 Defining an EE VIPA in TCP/IP

```
DEVICE EEVIPA1  VIRTUAL 0         1
LINK EELINK1 VIRTUAL 0 EEVIPA1    2
;
HOME
  10.10.1.230    VIPALINK         3
  10.10.2.232    OSA2080LNK       4
  10.10.3.233    OSA20A0LNK
  10.10.2.234    OSA20C0LNK
  10.10.3.235    OSA20E0LNK
  10.10.1.231    EELINK1          5

  PRIMARYINTERFACE VIPALINK       6
```

Note the following explanation for Example 10-1:

- 1: A DEVICE statement defines the EE VIPA as a virtual device.
- 2: A LINK statement defines the EE VIPA as a virtual link and associates it to the DEVICE.
- 3: Other static VIPAs (not related to EE) can be defined. Usually each stack has at least one VIPA defined that represents the stack or system on which it is running. Other applications, clients, and servers can use the stack's VIPA if they do not require association with a specific VIPA. Most of the manually defined interfaces are to be listed in the HOME list.
- 4: The physical interfaces of the stack are usually placed following the stack's VIPA. When physical interfaces follow a static VIPA in the HOME list and SOURCEVIPA is in effect, the static VIPA preceding the physical interfaces is used as the SOURCEVIPA in outbound packets from all applications except Enterprise Extender. EE uses its own VIPA as the SOURCEVIPA regardless of the stack's setting.
- 5: The order in which the VIPA link interfaces are positioned in the HOME list is very important when SOURCEVIPA is in effect. When assigning dedicated VIPAs to EE, place them at the *bottom* of the list after all the physical interfaces. Being placed at the end of the list will prevent their use as a source VIPA for *non-EE* traffic.
- 6: Use the PRIMARYINTERFACE statement to specify which link is to be designated as the default local host for use by the GETHOSTID() function. The PRIMARYINTERFACE statement's link IP address is *not* used as the source IP address for any out-going datagrams, *unless* that *same* address happens to be configured as the SOURCEVIPA address.

10.2.2 Defining a local EE VIPA in OMPROUTE configuration file

Example 10-2 shows how to define an EE VIPA in OMPROUTE.

Example 10-2 Defining an EE VIPA in OMPROUTE

```
; Static EE vipa
ospf_interface ip_address=10.10.1.231       1
          subnet_mask=255.255.255.0
          name=EELINK1                      2
          Advertise_VIPA_Routes=HOST_ONLY   3
          attaches_to_area=0.0.0.2
          cost0=10
          mtu=1500;
```

Note the following explanation for Example 10-2:

- 1: The VIPA's IP address must match the one specified in the HOME list of the stack profile (see Example 10-1 on page 325).
- 2: The VIPA's name must match the link name specified in the LINK statement and the HOME list of the stack profile (see Example 10-1 on page 325).
- 3: OSPF should not advertise VIPA subnets. Only the 32-bit host address is to be advertised.

10.2.3 Specifying a local VIPA in the VTAM ATCSTRxx start list

Example 10-3 shows how to specify the EE VIPA address in VTAM's start list.

Example 10-3 Specifying an EE VIPA address in ATCSTRxx

```
TCPNAME=TCPIPA,                  X   1
IPADDR=10.10.1.231,              X   2
```

Note the following explanations for Example 10-3:

- 1: VTAM associates with only one stack at a time. It establishes stack affinity through the setting of the TCPNAME start option. This is necessary only when there are multiple stacks running on the system.

  Note: Although specifying TCPNAME is optional in a single stack environment, you should specify a name for the TCP/IP stack that defines the local static VIPA for EE, because it ensures more reliable line activation.

- 2: When only one VIPA is being used by EE, the IP address of that VIPA can be specified by the IPADDR startup option. This becomes a global (default) setting for any EE XCA major node that activates without having a specific IP address assigned. Optionally, this single IP address can be omitted in the start list and coded in the EE XCA definition instead. VTAM accepts the IP address at either location. The HOSTNAME can be used instead of IPADDR, as shown here:

  ```
  HOSTNAME=SC30M-EE1.ITSO.IBM.COM
  ```

10.2.4 Specifying a local VIPA in the EE XCA major node

> **Note:** We no longer recommend coding the IPADDR option in the ATCSTRxx start list. For consistency, we recommend always coding it in the EE XCA major node where it is required when multiple VIPAs are implemented. It can be coded in the EE XCA whether single or multiple VIPAs are in use.

Example 10-4 shows how to specify an EE VIPA in the EE XCA major node.

Example 10-4 Specifying an EE VIPA in the EE XCA

```
EEXCA30  VBUILD TYPE=XCA
EEPORT30 PORT MEDIUM=HPRIP,                                            *
               . . . .
EEGVL301 GROUP DIAL=YES,CALL=INOUT,                                    *
               IPADDR=10.10.1.231,                                     *   1
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                  *   2
               . . . .

EEGVG301 GROUP DIAL=YES,CALL=INOUT,                                    *
               IPADDR=10.10.1.231,                                     *   1
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      *   3
               . . . .
```

Note the following explanation for Example 10-4:

- 1: A single VIPA can be used for EE. Multiple groups can be associated with a single common VIPA. When using only one single VIPA for all EE groups, it can be specified with the IPADDR keyword in either the VTAM ATCSTRxx start list or in the EE XCA GROUP. If IPADDR is specified in both locations, the value provided in the XCA is used by EE. HOSTNAME can be used *instead* of IPADDR, as shown here:

  ```
  HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                        *
  ```

- 2: The single VIPA address can be associated with a Local Connection Network (CN). A Local Connection Network is represented by a Virtual Routing Node (VRN), and is the mechanism whereby nodes belonging to the same CN (or VRN) can directly connect to each other without having to pass through an intermediate node. See the introduction chapter for a description of a Connection Network and its Virtual Routing Node.
- 3: The single VIPA address can be associated with a Global Connection Network.

10.2.5 Specifying a remote VIPA in the EE SWNET major node

Example 10-5 shows how to specify a VIPA in an EE SWNET major node.

Example 10-5 Specifying an EE VIPA in an EE SWNET

```
EESWEB30 VBUILD TYPE=SWNET
EEP63047 PU    CPNAME=SC47M,NETID=USIBMSC,TGN=06,                      *1
               . . . .
EEPT6T47 PATH  IPADDR=10.10.1.131,SAPADDR=4,                           *2
               . . . .
```

Note the following explanation for Example 10-5:

- 1: This definition is for remote partner SC47M. Assume it has only one EE VIPA defined, and is reachable only via that single EE VIPA.
- 2: The single EE VIPA IP address is specified with the IPADDR keyword. The HOSTNAME can be used instead of the IPADDR, as shown here:

```
EEPT6T47 PATH  HOSTNAME=SC47M-EE1.ITSO.IBM.COM,SAPADDR=4,                *
```

10.3 Configuring multiple EE VIPAs

We assigned two static VIPAs to our EE to help explain the use of multiple EE VIPAs. The single EE VIPA configuration created in the previous section is used as a base on which to build the multiple EE VIPA environment, in this section. Figure 10-3 shows the necessary definitions in your local VTAM, TCP/IP, and OMPROUTE for supporting two EE VIPAs that can be used to support multiple EE connections. The figure is followed by configuration examples showing how to add a local EE VIPA to your system. The discussion includes setting up SWNET major nodes on your system that point to remote partners.

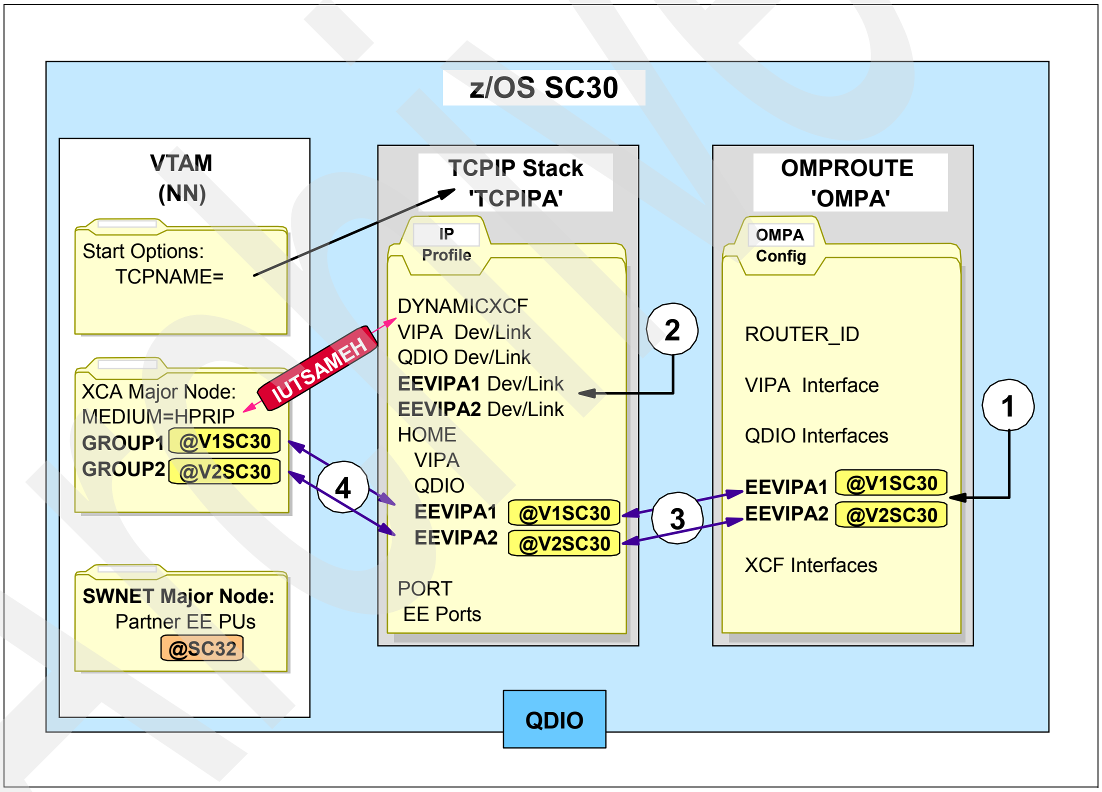


Figure 10-3 Defining multiple EE VIPAs on your local system

Note the following explanation for Figure 10-3:

- 1: Add the additional EE VIPA to OMPROUTE using the OSPF_INTERFACE statement, specifying the Link name and its IP address.
- 2: Add the additional EE VIPA to TCP/IP using the DEVICE, LINK, and HOME statements.
- 3: The Link name and IP address used in OMPROUTE must match those used in the HOME statement.

- 4: Add a second GROUP statement to the EE XCA major node, specifying the additional EE VIPA's IP address (or host name). The IP address must be the one assigned in the HOME list in TCP/IP.

Do not forget to have VTAM's stack affinity defined by using the VTAM TCPNAME start option. The DYNAMICXCF statement in TCP/IP has created the IUTSAMEH device used for VTAM-to-TCP/IP communication.

This section includes the following topics for defining an additional local EE VIPA:

- "Defining multiple local EE VIPAs in TCP/IP profile"
- "Defining multiple local EE VIPAs in OMPROUTE"
- "Omitting IPADDR in VTAM ATCSTRxx start list"
- "Specifying multiple local VIPAs in EE XCA major node"
- "Specifying multiple remote VIPAs in EE SWNET major node"

10.3.1 Defining multiple local EE VIPAs in TCP/IP profile

Example 10-6 shows how to define multiple EE VIPAs in TCP/IP.

Example 10-6 Defining multiple EE VIPAs in TCP/IP

```
DEVICE EEVIPA1  VIRTUAL 0        1
LINK EELINK1 VIRTUAL 0 EEVIPA1
;
DEVICE EEVIPA2  VIRTUAL 0        2
LINK EELINK2 VIRTUAL 0 EEVIPA2
;
HOME
  10.10.1.230    VIPALINK
  10.10.2.232    OSA2080LNK
  10.10.3.233    OSA20A0LNK
  10.10.2.234    OSA20C0LNK
  10.10.3.235    OSA20E0LNK
  10.10.1.231    EELINK1         3
  10.10.1.232    EELINK2

  PRIMARYINTERFACE VIPALINK      4
```

Note the following explanation for Example 10-6:

- 1: DEVICE and LINK statements define the EE VIPA#1 as a virtual device.
- 2: DEVICE and LINK statements define the EE VIPA#2 as a virtual device.
- 3: The order in which the VIPA link interfaces are positioned in the HOME list is very important when SOURCEVIPA is in effect. When assigning dedicated VIPAs to EE, place them at the *bottom* of the list after all the physical interfaces. Being placed at the end of the list will prevent their use as a source VIPA for *non-EE* traffic.
- 4: Use the PRIMARYINTERFACE statement to specify which link is to be designated as the default local host for use by the GETHOSTID() function. The PRIMARYINTERFACE statement's link IP address is *not* used as the source IP address for any out-going datagrams, *unless* that *same* address happens to be configured as the SOURCEVIPA address.

10.3.2 Defining multiple local EE VIPAs in OMPROUTE

Example 10-7 shows how to define multiple VIPAs in the OMPROUTE configuration file.

Example 10-7 Defining multiple EE VIPAs in OMPROUTE

```
; Static EE vipa #1
ospf_interface ip_address=10.10.1.231       1
           subnet_mask=255.255.255.0
           name=EELINK1                     2
           Advertise_VIPA_Routes=HOST_ONLY  5
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; Static EE vipa #2
ospf_interface ip_address=10.10.1.232       3
           subnet_mask=255.255.255.0
           name=EELINK2                     4
           Advertise_VIPA_Routes=HOST_ONLY  5
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
```

Note the following explanation for Example 10-7:

- 1: The VIPA#1’s IP address must match the one specified in the HOME list of the stack profile (see Example 10-6 on page 329).
- 2: The VIPA#1’s name must match the link name specified in the LINK statement and the HOME list of the stack profile (see Example 10-6 on page 329).
- 3: The VIPA#2’s IP address must match the one specified in the HOME list of the stack profile (see Example 10-6 on page 329).
- 4: The VIPA#2’s name must match the link name specified in the LINK statement and the HOME list of the stack profile (see Example 10-6 on page 329).
- 5: Remember to advertise VIPA host routes only. VIPA subnets are not to be advertised.

10.3.3 Omitting IPADDR in VTAM ATCSTRxx start list

When using multiple EE VIPAs, the IPADDR and HOSTNAME start options must be omitted from the VTAM start list, and one of them specified within the EE XCA major node. However the TCPNAME start option may still be required to establish stack affinity for VTAM.

10.3.4 Specifying multiple local VIPAs in EE XCA major node

Example 10-8 shows how to specify multiple VIPAs in the EE XCA.

Example 10-8 Specifying multiple VIPAs in the EE XCA

```
*       EEXCA MAJOR NODE
EEXCA30 VBUILD TYPE=XCA
EEPORT30 PORT MEDIUM=HPRIP,                                           *
               . . . .
EEGVL301 GROUP DIAL=YES,CALL=INOUT,                                  *
               HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                      *1
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                  *
               . . . .
EEGVG301 GROUP DIAL=YES,CALL=INOUT,                                    *
               IPADDR=10.10.1.232,                                     *2
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      *
               . . . .
```

Note the following explanation for Example 10-8:

- 1: EE VIPA#1 can be specified with *either* the HOSTNAME or IPADDR keyword.
- 2: EE VIPA#2 can be specified with *either* the HOSTNAME or IPADDR keyword.

10.3.5 Specifying multiple remote VIPAs in EE SWNET major node

Example 10-9 shows how to specify multiple VIPAs in an EE SWNET.

Example 10-9 Specifying multiple VIPAs in an EE SWNET

```
EESWEB30 VBUILD TYPE=SWNET
************************************************************************
* SWITCHED MAJORNODE FOR ENTERPRISE EXTENDER TO SC47M USING            *
* PARALLEL TGS TO USE SEPARATE IP INFRASTRUCTRE                        *
*   CPCP AND SENSITIVE SESSION ONLY VIA SLOWER BUT SECURE LINK         *
*              COMMON PREFIX FOR EE PU                                 *
*              |  TGN                                                  *
*              |  |SYSCLONE_LEFT                                       *
* 10.10.1.231  |  || SYSCLONE_RIGHT                   10.1.1.131       *
*  _____   |   |  || |                                 |   _____       *
* |     |-EE1--EEP63047-TG6--(PUBLIC IP)----2M--------EE1-|     |      *
* |SC30M|                                                 |SC47M|      *
* |_____|-EE2--EEP73047-TG7--(SECURE IP)---256K-------EE2-|_____|      *
*          |                                           |               *
* 10.10.1.232                                         10.1.1.132       *
*******************************************************************
*******************************************************************
*  TG6 = FAST BUT UNSECURE LINK TO SC47M
*******************************************************************
EEP63047 PU    CPNAME=SC47M,NETID=USIBMSC,TGN=06,                      *
               . . . .
EEPT6T47 PATH  HOSTNAME=SC47M-EE1.ITSO.IBM.COM,                        * 1
. . . .
EEP73047 PU    CPNAME=SC47M,NETID=USIBMSC,TGN=07,                      *
               . . . .
EEPT7T47 PATH  IPADDR=10.10.1.232,                                     * 2
. . . .
```

Note the following explanation for Example 10-9:

- 1: EE VIPA#1 can be specified with *either* the HOSTNAME or IPADDR keyword.
- 2: EE VIPA#2 can be specified with *either* the HOSTNAME or IPADDR keyword.

10.4 Activation of multiple EE VIPAs

This section includes the following activation topics:

- “Activating VIPAs in OMPROUTE”
- “Activating VIPAs in TCP/IP”
- “Activating the EE XCA major node”
- “Activating the EE SWNET major node”

10.4.1 Activating VIPAs in OMPROUTE

This section shows the following activities related to OMPROUTE:

- “Initial activation of OMPROUTE configuration file”
- “Updating and refreshing of OMPROUTE configuration file”

Initial activation of OMPROUTE configuration file

When OMPROUTE starts, it establishes affinity with the TCP/IP stack specified by the RESOLV_CONFIG environment variable, matches the link interfaces in the TCP/IP stack with those defined in its configuration file, creates route entries in its routing table based on the defined interfaces, and then shares the route information with the stack. The VIPAs are defined to OMPROUTE in such a way that OMPROUTE does not advertise their subnet, but only their full host address. Example 10-10 shows how to activate the OMPROUTE configuration file.

Example 10-10 Activating OMPROUTE configuration file containing multiple EE VIPAs

```
S OMPA        1

omproute      2

AUTOLOG
   OMPA       3
ENDAUTOLOG
```

Note the following explanation for Example 10-10:

- OMPROUTE can be started by using *one* of the following methods:
  - 1: Using the MVS Start command: `S OMPA`
  - 2: Using the OMVS shell command: **`omproute`**
  - 3: Using AUTOLOG within the TCP/IP profile

Updating and refreshing of OMPROUTE configuration file

If OMPROUTE is already running, a dynamic update can be made to its configuration by using the system MODIFY (F) command specifying a RECONFIG of the file. However, only new (non-existent) interfaces can be added to the configuration and dynamically refreshed. Changes to existing interfaces cannot be made dynamically. If a refresh to an existing interface is attempted, OMPROUTE ignores the attempt.

Adding a second EE VIPA to OMPROUTE can be accomplished by first adding the additional definition statements to the configuration file. Then activating them by using the RECONFIG command or by recycling the OMPROUTE started task. Example 10-11 on page 333 shows how to dynamically update the OMPROUTE configuration with an additional EE VIPA.

Example 10-11 Updating the OMPROUTE configuration with an additional EE VIPA

```
ro sc30,f ompa,reconfig                                                    1
EZZ7866I OMPROUTE MODIFY COMMAND ACCEPTED                                  2
EZZ8073I DYNAMICALLY ADDED OSPF_INTERFACE EELINK2 TO OMPROUTE CONFIGURATION  3

or

EZZ7821I IGNORING DUPLICATE OSPF_INTERFACE STATEMENT FOR 10.10.1.232       4
```

Note the following explanation for Example 10-11:

- 1: The Modify command requesting OMPROUTE to reread the configuration file and add any new interfaces not yet defined.
- 2: OMPROUTE always acknowledges the request with message EZZ7866I.
- 3: If a new interface definition is encountered during the reconfig scan, OMPROUTE issues message EZZ8073I to confirm the addition of the interface to the configuration.
- 4: If the interface definition has previously been added to the configuration, OMPROUTE issues message EZZ7821I to indicate it is ignoring any updates to the interface.

Important: These new configuration statements must be reread from the configuration file through this reconfig command *before* the interface is configured to the TCP/IP stack.

10.4.2 Activating VIPAs in TCP/IP

This section shows how to add and activate a new VIPA to the TCP/IP profile.

Initial activation of the TCP/IP profile

The TCP/IP profile can always be reinitialized by simply restarting the task with the MVS system Start command: **S TCPIPA**.

Updating of the TCP/IP profile

If the TCP/IP started task cannot be recycled due to production schedules, then it can be updated dynamically. An obey file can be created containing the necessary definition statements for the new VIPA, and be used to update the profile dynamically. The obey file can be a sequential data set or a member of a data set.

Example 10-12 shows how to dynamically update the TCP/IP profile with a new EE VIPA.

Example 10-12 Updating the TCP/IP profile with an additional EE VIPA using member 'ADDVIPA'

```
DEVICE EEVIPA2 VIRTUAL 0                                                   1
LINK EELINK2 VIRTUAL 0 EEVIPA2                                             2
;
HOME
  10.10.1.230    VIPALINK
  10.10.2.232    OSA2080LNK
  10.10.3.233    OSA20A0LNK
  10.10.2.234    OSA20C0LNK
  10.10.3.235    OSA20E0LNK
  10.10.1.231    EELINK1
  10.10.1.232    EELINK2                                                   3
PRIMARYINTERFACE VIPALINK
```

```
;      The OBEYFILE command
ro sc30,v tcpip,tcpipa,o,tcpipa.tcpparms(addvipa)                             4
EZZ0060I PROCESSING COMMAND: VARY TCPIP,TCPIPA,O,TCPIPA.TCPPARMS(ADDVIPA)
EZZ0300I OPENED OBEYFILE FILE 'TCPIPA.TCPPARMS(ADDVIPA)'
EZZ0309I PROFILE PROCESSING BEGINNING FOR 'TCPIPA.TCPPARMS(ADDVIPA)
EZZ0316I PROFILE PROCESSING COMPLETE FOR FILE 'TCPIPA.TCPPARMS(ADDVIPA)
EZZ0053I COMMAND VARY OBEY COMPLETED SUCCESSFULLY

ro sc30,d tcpip,tcpipa,n,home                                                 5
EZD0101I NETSTAT CS V1R8 TCPIPA 669
HOME ADDRESS LIST:
LINKNAME:   VIPALINK
  ADDRESS:  10.10.1.230
    FLAGS:  PRIMARY
. . . .
LINKNAME:   EELINK1
  ADDRESS:  10.10.1.231
    FLAGS:
LINKNAME:   EELINK2                                                           6
  ADDRESS:  10.10.1.232
    FLAGS:
. . . .
```

Note the following explanation for Example 10-12:

- 1 and 2: DEVICE and LINK statements for the new VIPA define it to the stack.
- 3: The HOME list includes the new VIPA. The entire HOME list must be coded in order to retain all current entries.
- 4: The OBEYFILE command tells the stack to read the ADDVIPA file and act upon the statements within. This action adds the new EELINK2 VIPA to the stack and home list, and activates it, placing it into a status of READY.
- 5: A NETSTAT HOME command displays the HOME list.
- 6: The new EELINK2 VIPA is now in the updated HOME list.

10.4.3 Activating the EE XCA major node

This section shows the following activities related to the EE XCA major node:

- "Initial activation of the EE XCA major node"
- "Updating and refresh of the EE XCA major node"

Initial activation of the EE XCA major node

The EE XCA major node can be updated with the necessary statements to define the additional EE VIPA. If the XCA node is not currently active, it can be activated using the VTAM `VARY ACT` command, and the new definitions will be processed. If it is already active, then it can be recycled by using the VTAM VARY INACT and VARY ACT commands. However, this will stop *all* EE connections and is usually not scheduled until a planned network outage.

Example 10-13 shows how to activate the XCA major node that contains a new GROUP associated with the new EE VIPA.

Example 10-13 Activating the XCA major node containing a new VIPA

```
ro sc30,v net,act,id=eexca                                    1
IST097I VARY ACCEPTED
IST093I EEXCA ACTIVE                                          2
IST1168I VIRTUAL NODE W3IBMCOM.VRN CONNECTION ACTIVE          3
IST1168I VIRTUAL NODE RDBOOKEE.VRNLOCAL CONNECTION ACTIVE     4
```

Note the following explanation for Example 10-13:

- 1 Activating the EEXCA node generates a set of messages.
- 2: VTAM confirms a successful activation.
- 3:The new VIPA is associated with the Global VRN group.
- 4:The original VIPA is associated with the Local VRN group.

Updating and refresh of the EE XCA major node

You can dynamically change a number of EE XCA major node operands by editing the VTAMLST member and then issuing the VARY NET,ID=eexca,ACT,UPDATE=ALL command. You can specify a new operand value on a higher-level definition statement for sifting, if applicable. To dynamically change an operand on the GROUP statement, the resource to which it applies must be inactive and all LINE subnodes must be inactive.

A GROUP can be added, updated, or removed. An IPADDR or HOSTNAME can be changed. An IPADDR can be replaced by a HOSTNAME, and vice verse.

Example 10-14 shows how to dynamically update the EE XCA major node with a new GROUP associated with the new EE VIPA.

Example 10-14 Updating the EE XCA major node with an additional EE VIPA

```
* Original source of EEXCA major node includes only one GROUP
EEXCA30  VBUILD TYPE=XCA                                                1
EEPORT30 PORT MEDIUM=HPRIP,                                    *
               . . . .
EEGVL301 GROUP DIAL=YES,CALL=INOUT,                                    *
               HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                        *
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                  *
               . . . .
* Updated source of EEXCA major node includes the additional second GROUP
EEXCA30  VBUILD TYPE=XCA                                                2
EEPORT30 PORT MEDIUM=HPRIP,                                            *
               . . . .
EEGVL301  GROUP DIAL=YES,CALL=INOUT,                                    *
               HOSTNAME=SC30M-EE1.ITSO.IBM.COM,                        *
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                  *
               . . . .
EEGVG301 GROUP DIAL=YES,CALL=INOUT,                                    *3
               HOSTNAME=SC30M-EE2.ITSO.IBM.COM,                        *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      *
               . . . .
ro sc30,v net,act,id=eexca,update=all                                   4
IST097I VARY ACCEPTED
IST886I VARY ACT EEXCA CHANGE EEGVL301 FAILED                           5
IST523I REASON = INVALID RESOURCE CURRENT STATE
```

```
IST314I END
IST093I EEXCA ACTIVE                                                       6
IST1168I VIRTUAL NODE W3IBMCOM.VRN CONNECTION ACTIVE                       7
```

Note the following explanation for Example 10-14:

- 1: The original source of EEXCA shows only the Local GROUP defined.
- 2: The updated source of EEXCA includes the additional second GROUP.
- 3: The new second GROUP defines the Global GROUP.
- 4: Activating the EEXCA major node with the UPDATE=ALL option generates a set of messages.
- 5: Messages IST886I and IST523I indicate the *existing* GROUP cannot be altered while it is active. We did not change anything on its definition and did not want to update it.
- 6: Message IST093I indicates that the remaining statements (our new ones) were accepted.
- 7: Message IST1168I shows the Global VRN GROUP associated with our new VIPA *did* activate and connect successfully as a VRN.

10.4.4 Activating the EE SWNET major node

This section discusses the following VTAM activities related to EE SWNET nodes:

- “Activating multiple EE SWNET nodes containing one PU each”
- “Activating one EE SWNET node containing multiple PUs”

Activating multiple EE SWNET nodes containing one PU each

Defining only one PU in a SWNET gives more flexibility in managing that PU definition and any connections it may establish. If modifications are required to its definition, the SWNET can be recycled without adversely affecting any other connected PUs. The manual effort in maintaining multiple SWNETs is more demanding compared to defining a number of PUs within one SWNET.

Activating one EE SWNET node containing multiple PUs

Defining multiple PUs within a SWNET eases the clerical burden in maintaining the definitions, but reduces the flexibility in managing the PUs as a group. If a modification is required to one, they may all be adversely affected if the owning SWNET node must be recycled to pick up the modification.

10.5 Verification of the multiple EE VIPAs environment

This section includes the following verification topics:

- “Verifying VIPAs in OMPROUTE”
- “Verifying VIPAs in TCP/IP”
- “Verifying VIPAs in VTAM EE XCA”
- “Verifying VIPAs in VTAM EE SWNET”

10.5.1 Verifying VIPAs in OMPROUTE

OMPROUTE VIPA definitions can be verified using the following procedures:

- "Verify the configured OSPF interface information for each EE VIPA"
- "Verify the runtime OSPF interface information for each EE VIPA"

Verify the configured OSPF interface information for each EE VIPA

Example 10-15 shows how to display the *configured* interface information for EE VIPAs in OMPROUTE.

Example 10-15 Displaying configured interfaces in OMPROUTE

```
ro sc30,d tcpip,tcpipa,omp,ospf,list,interfaces                              1
EZZ7833I INTERFACE CONFIGURATION 299
IP ADDRESS      AREA            COST RTRNS TRDLY PRI HELLO  DEAD DB_EX
10.10.1.232     0.0.0.2           10   N/A   N/A N/A   N/A   N/A   N/A       2
10.10.1.231     0.0.0.2           10   N/A   N/A N/A   N/A   N/A   N/A       3
10.10.3.235     0.0.0.2           10     5     1   0    10    40    40
10.10.2.234     0.0.0.2           10     5     1   0    10    40    40
10.10.3.233     0.0.0.2           10     5     1   0    10    40    40
10.10.2.232     0.0.0.2           10     5     1   0    10    40    40
10.10.1.230     0.0.0.2           10   N/A   N/A N/A   N/A   N/A   N/A

             ADVERTISED VIPA ROUTES                                          4
10.10.1.232    /255.255.255.255   10.10.1.231    /255.255.255.255            5
10.10.1.230    /255.255.255.255
```

Note the following explanation for Example 10-15:

- 1: The display command to show configured interface information responds with one line for each interface configured. Notice that the interfaces are listed in reverse order in which they are defined in the configuration file.
- 2: The EELINK2 interface is identified by its VIPA address.
- 3: The EELINK1 interface is identified by its VIPA address.
- 4: The section called ADVERTISED VIPA ROUTES indicates which VIPAs are being advertised and what type of advertisements they are.
- 5: We specified HOST_ONLY for the type of VIPA advertisements, and therefore the subnet mask for each of the VIPAs is set to the full 32-bit mask. The IP addresses of our EE VIPAs should be in the list, as well as any other VIPAs that may be defined.

Verify the runtime OSPF interface information for each EE VIPA

Example 10-16 shows how to display the *runtime* interface information for EE VIPAs in OMPROUTE.

Example 10-16 Displaying interface runtime information in OMPROUTE

```
ro sc30,d tcpip,tcpipa,omp,ospf,interface                                    1
EZZ7849I INTERFACES 302
IFC ADDRESS     PHYS         ASSOC. AREA     TYPE   STATE  #NBRS  #ADJS
10.10.1.232     EELINK2      0.0.0.2         VIPA    N/A    N/A    N/A       2
10.10.1.231     EELINK1      0.0.0.2         VIPA    N/A    N/A    N/A       3
10.10.3.235     OSA20E0LNK   0.0.0.2         BRDCST   32      5      1
10.10.2.234     OSA20C0LNK   0.0.0.2         BRDCST   32      5      1
10.10.3.233     OSA20A0LNK   0.0.0.2         BRDCST    2      0      0
```

```
10.10.2.232     OSA2080LNK   0.0.0.2         BRDCST    2      0      0
10.10.1.230     VIPALINK     0.0.0.2         VIPA    N/A    N/A    N/A

ro sc30,d tcpip,tcpipa,omp,ospf,interface,name=eelink1                          4
EZZ7850I INTERFACE DETAILS 305
                INTERFACE ADDRESS:      10.10.1.231                             5
                ATTACHED AREA:          0.0.0.2
                PHYSICAL INTERFACE:     EELINK1                                 6
                INTERFACE MASK:         255.255.255.0
                INTERFACE TYPE:         VIPA
                TOS 0 COST:             10

ro sc30,d tcpip,tcpipa,omp,ospf,interface,name=eelink2                          4
EZZ7850I INTERFACE DETAILS 308
                INTERFACE ADDRESS:      10.10.1.232                             5
                ATTACHED AREA:          0.0.0.2
                PHYSICAL INTERFACE:     EELINK2                                 6
                INTERFACE MASK:         255.255.255.0
                INTERFACE TYPE:         VIPA
                TOS 0 COST:             10
```

Note the following explanation for Example 10-16:

1: The display command to show runtime interface information responds with one line for each interface configured. Notice that the interfaces are listed in reverse order in which they are defined in the configuration file.

2: The EELINK2 interface is identified by its VIPA address and name.

3: The EELINK1 interface is identified by its VIPA address and name.

4: The display command that shows detailed information for a specific interface name responds with message EZZ7850I.

5 and 6: The VIPA interface is identified by its IP address and its name.

10.5.2 Verifying VIPAs in TCP/IP

TCP/IP VIPA definitions can be verified using the following procedures:

- “Verify the HOME list entries for the EE VIPAs”
- “Verify the DEVLINKS status of the EE VIPAs”
- “Verify the ROUTE table entries for the EE VIPAs”

Verify the HOME list entries for the EE VIPAs

Example 10-17 shows how to verify the presence of EE VIPAs in the HOME list.

Example 10-17 Displaying the contents of the HOME list

```
ro sc30,d tcpip,tcpipa,n,home                    1

ro sc30,d tcpip,tcpipa,n,home,intfname=eelink1   2
EZD0101I NETSTAT CS V1R8 TCPIPA 443
HOME ADDRESS LIST:
LINKNAME:   EELINK1                              3
  ADDRESS:  10.10.1.231                          4
```

```
    FLAGS:
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT

ro sc30,d tcpip,tcpipa,n,home,intfname=eelink2   2
EZD0101I NETSTAT CS V1R8 TCPIPA 446
HOME ADDRESS LIST:
LINKNAME:   EELINK2                              3
  ADDRESS:  10.10.1.232                          4
    FLAGS:
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT
```

Note the following explanation for Example 10-17:

- 1: The basic Netstat home command lists the entire home list, which could be very long.
- 2: The output of the Home command can be limited by filtering on the interface name, the Link name, as defined in the TCP/IP profile.
- 3: The VIPA interface is identified by its Link name from the TCP/IP profile LINK statement.
- 4:The VIPA interface is identified by its IP address from the TCP/IP profile HOME statement.

Verify the DEVLINKS status of the EE VIPAs

Example 10-18 shows how to display Device and Link status information for EE VIPAs.

Example 10-18 Displaying the Device and Link status information for EE VIPAs

```
ro sc30,d tcpip,tcpipa,n,dev                                                 1

ro sc30,d tcpip,tcpipa,n,dev,intfname=eelink1                                2
EZD0101I NETSTAT CS V1R8 TCPIPA 457
DEVNAME: EEVIPA1           DEVTYPE: VIPA                                     3
  DEVSTATUS: READY                                                           4
  LNKNAME: EELINK1           LNKTYPE: VIPA       LNKSTATUS: READY            5
    NETNUM: N/A  QUESIZE: N/A
  . . . .
ro sc30,d tcpip,tcpipa,n,dev,intfname=eelink2                                2
EZD0101I NETSTAT CS V1R8 TCPIPA 460
DEVNAME: EEVIPA2           DEVTYPE: VIPA                                     3
  DEVSTATUS: READY                                                           4
  LNKNAME: EELINK2           LNKTYPE: VIPA       LNKSTATUS: READY            5
    NETNUM: N/A  QUESIZE: N/A
  . . . .
```

Note the following explanation for Example 10-18:

- 1: The basic Netstat Devlinks command lists information for *all* devices, which could be more information than desired. The command output can be limited by filtering it.
- 2: The output of the Devlinks command can be limited by filtering on the interface name, the Link name, as defined in the TCP/IP profile.

- 3 and 5: The VIPA interface is identified by its Device name and Link name from the TCP/IP profile DEVICE and LINK statements.
- 4 and 5: The Device status and the Link status of a VIPA should always be READY. If they are not, there is a problem with the definitions.

Verify the ROUTE table entries for the EE VIPAs

Example 10-19 shows how to verify the presence of EE VIPAs in the stack's routing table.

Example 10-19 Displaying the stack's routing table entries for EE VIPAs

```
tso netstat route                                                    1
MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPIPA         21:43:29
IPv4 Destinations
Destination         Gateway          Flags     Refcnt  Interface
-----------         -------          -----     ------  ---------
. . . .
10.10.1.222/32      10.10.3.223      UGHO      000000  OSA20A0LNK
10.10.1.230/32      0.0.0.0          UH        000000  VIPALINK
10.10.1.231/32      0.0.0.0          UH        000000  EELINK1           2
10.10.1.232/32      0.0.0.0          UH        000000  EELINK2           3
10.10.1.240/32      10.10.2.244      UGHO      000000  OSA2080LNK
. . . .
ro sc30,d tcpip,tcpipa,n,route                                       4

ro sc30,d tcpip,tcpipa,n,route,ipaddr=10.10.1.231                    5
EZD0101I NETSTAT CS V1R8 TCPIPA 463
IPV4 DESTINATIONS
DESTINATION         GATEWAY          FLAGS     REFCNT  INTERFACE
10.10.1.231/32      0.0.0.0          UH        000000  EELINK1           6
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT

ro sc30,d tcpip,tcpipa,n,route,ipaddr=10.10.1.232                    7
EZD0101I NETSTAT CS V1R8 TCPIPA 466
IPV4 DESTINATIONS
DESTINATION         GATEWAY          FLAGS     REFCNT  INTERFACE
10.10.1.232/32      0.0.0.0          UH        000000  EELINK2           8
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT
```

Note the following explanation for Example 10-19:

- 1 and 4; The basic Netstat Route command lists information for *all* routes, which could be more information than desired. The command output can be limited by filtering it.
- 2 and 3: Route entries for EE VIPAs are located and identified in the routing table by the IP address and Link name.
- 5 and 7: The output of the Route command can be limited by filtering on the IP address of the VIPA as defined by the TCP/IP profile HOME statement.
- 6 and 8: The routing table entry for the requested VIPA IP address is returned as part of the filtered response.

10.5.3 Verifying VIPAs in VTAM EE XCA

EE XCA VIPA definitions can be verified using the following procedures:

- "Verify the EE environment"
- "Verify the EE XCA status and contents"

Verify the EE environment

Example 10-20 shows how to determine if EE is active, and how to determine the name of the the EE XCA major node.

Example 10-20 Determine the status of EE, is it active or not?

```
ro sc30,d net,ee                                                     1
IST2045I ENTERPRISE EXTENDER XCA MAJOR NODE NOT ACTIVE               2
                 or
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2000I ENTERPRISE EXTENDER GENERAL INFORMATION
IST1685I TCP/IP JOB NAME = TCPIPA                                    3
IST2003I ENTERPRISE EXTENDER XCA MAJOR NODE NAME = EEXCA             4
. . . .
ro sc30,d tcpip,tcpipa,n,conn,client=net                             5
EZD0101I NETSTAT CS V1R8 TCPIPA 369
USER ID  CONN     STATE
0 OF 0 RECORDS DISPLAYED                                             6
END OF THE REPORT

ro sc30,d tcpip,tcpipa,n,conn,ipaddr=10.10.1.231                     7
EZD0101I NETSTAT CS V1R8 TCPIPA 810
USER ID  CONN     STATE
0 OF 0 RECORDS DISPLAYED                                             8
END OF THE REPORT

ro sc30,d tcpip,tcpipa,n,conn,ipaddr=10.10.1.232                     7
EZD0101I NETSTAT CS V1R8 TCPIPA 810
USER ID  CONN     STATE
0 OF 0 RECORDS DISPLAYED                                             8
END OF THE REPORT
```

Note the following explanation for Example 10-20:

- 1: The VTAM Display EE command provides general EE information, and the output can consist of a number of messages, depending on the EE environment. A few strategic messages near the beginning of the output indicate the status of EE.
- 2: Message IST2045I indicates that there is no EE XCA major node active. Either it has never been activated, or its activation failed.
- 3: Message IST1685I is confirmation that the EE XCA major node *is* active and EE has established stack affinity with the indicated TCP/IP stack.
- 4: Message IST2003I indicates the EE XCA major node name that is currently active. Having determined the name of the EE XCA major node, you can display it to get more information about the general status of EE.
- 5: A display command requesting a list of connections that the VTAM started task name has with the stack can be used to determine if EE is active.

- 6: An indication that EE is not successfully active to the TCP/IP stack is that VTAM does not have any EE UDP ports (12000-12004) open on the stack.
- 7: A similar display command requesting a list of connections acquired by an EE VIPA address can be used to determine if that EE VIPA is active.
- 8: An indication that EE is not successfully active to the TCP/IP stack is that the EE VIPA is not be associated with any EE UDP ports.

Verify the EE XCA status and contents

Example 10-21 shows the status of the EE XCA and its associated EE VIPAs.

Example 10-21 Display the status of the EE XCA major node and its EE VIPAs

```
ro sc30,d net,id=eexca                                                 1
IST097I DISPLAY ACCEPTED
IST075I NAME = EEXCA, TYPE = XCA MAJOR NODE
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1679I MEDIUM = HPRIP
IST1685I TCP/IP JOB NAME = TCPIPA                                      2
IST924I --------------------------------------------------------------
IST089I EEGVL301 TYPE = LINE GROUP        , ACTIV                      3
IST1324I VNNAME = RDBOOKEE.VRNLOCAL  VNGROUP = EEGVL301  (LOCAL)
IST1680I LOCAL IP ADDRESS 10.10.1.231                                  4
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM                         5
IST2182I UNRCHTIM = 30
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
IST1106I EEGVL301 AC/R    21 NO    90750000000000000000017100808080
IST924I --------------------------------------------------------------
IST089I EEGVG301 TYPE = LINE GROUP        , ACTIV                      6
IST1324I VNNAME = W3IBMCOM.VRN       VNGROUP = EEGVG301  (GLOBAL)
IST1680I LOCAL IP ADDRESS 10.10.1.232                                  7
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM                         8
IST2182I UNRCHTIM = 180
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
IST1106I EEGVG301 AC/R    21 NO    92430000000000000000209100808080
IST924I --------------------------------------------------------------
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
IST314I END
```

Note the following explanation for Example 10-21:

- 1: A VTAM Display of the EE XCA major node name (obtained from the D NET,EE command), issues a set of messages that are helpful in determining the validity of each EE VIPA definition within the EE XCA major node.
- 2: Message IST1685I is confirmation that the EE XCA major node *is* active and EE has established stack affinity with the indicated TCP/IP stack.
- 3, 4, and 5: Messages IST085I, IST1680I, and IST1910I indicate the HOSTNAME and IP address of the EE VIPA#1 (EELINK1) and to which GROUP it is assigned.
- 6, 7, and 8: Messages IST085I, IST1680I, and IST1910I indicate the HOSTNAME and IP address of the EE VIPA#2 (EELINK2) and to which GROUP it is assigned.

The absence of IST1324I, IST1105, and IST1106 in this group of messages would imply that the group is not part of a Connection Network. The *AC/R* status indicates that the VRN is Active and Reported to APPN topology and routing services.

10.5.4 Verifying VIPAs in VTAM EE SWNET

EE SWNET VIPA definitions can be verified using the following procedures:

- “Verify the IP address status of EE VIPAs”
- “Verify the host name status of EE VIPAs”

Verify the IP address status of EE VIPAs

Example 10-22 shows how to determine if a local EE VIPA is available for connections. The set of response messages also gives an indication of the number of connections to the EE VIPA and the volume of traffic it is supporting.

Example 10-22 Displaying availability status of a local EE VIPA

```
ro sc30,d net,ee,ip=10.10.1.231,sum,max=*                             1
(ro sc30,d net,ee,ip=10.10.1.232,sum,max=*)

IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2002I ENTERPRISE EXTENDER AGGREGATE CONNECTION INFORMATION
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.231                                 2
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM                        2
IST2009I RTP PIPES =          4      LU-LU SESSIONS      =          3
IST2010I INOPS DUE TO SRQRETRY EXPIRATION                =        351
IST1324I VNNAME = RDBOOKEE.VRNLOCAL  VNGROUP = EEGVL301  (LOCAL)
IST2011I          AVAILABLE LINES FOR THIS EE VRN        =         13
IST2012I          ACTIVE CONNECTIONS USING THIS EE VRN   =          2
IST2013I AVAILABLE LINES FOR PREDEFINED EE CONNECTIONS   =          0
IST2014I ACTIVE PREDEFINED EE CONNECTIONS                =          1
IST2015I ACTIVE LOCAL  VRN EE CONNECTIONS                =          2
IST2016I ACTIVE GLOBAL VRN EE CONNECTIONS                =          0
IST2044I TOTAL ACTIVE EE CONNECTIONS FOR LOCAL IPADDR    =          3
IST924I -------------------------------------------------------------
IST2035I TOTALS FOR ALL PORT PRIORITIES
IST2036I   NLPS SENT           =                 3551  ( 003K )
IST2037I   BYTES SENT          =                16316  ( 016K )
IST2038I   NLPS RETRANSMITTED  =                    0  ( 000K )
IST2039I   BYTES RETRANSMITTED =                    0  ( 000K )
IST2040I   NLPS RECEIVED       =                 3571  ( 003K )
IST2041I   BYTES RECEIVED      =                17984  ( 017K )
IST314I END
```

Note the following explanation for Example 10-22:

- 1: The Display EE command is requesting summary information for a specific EE VIPA (responses for EELINK1 are shown).
- 2: Messages IST1680I and IST1910I confirm that the displayed information is associated with the specified EE VIPA from the original display request. The remaining messages provide connection and traffic volume statistics.

Verify the host name status of EE VIPAs

Example 10-23 shows how to determine if a local EE host name is available for remote connections via an EE SWNET.

Example 10-23 Display availability status of a local EE host name

```
ro sc30,d net,ee,hostname=sc30m-ee1.itso.ibm.com                     1
(ro sc30,d net,ee,hostname=sc30m-ee2.itso.ibm.com)

IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001            2
IST2120I HOSTNAME RESOLUTION IN PROGRESS                             3
IST314I END

IST350I DISPLAY TYPE = EE
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001            2
IST2121I HOSTNAME RESOLUTION COMPLETE                                3
IST1680I LOCAL IP ADDRESS 10.10.1.231                                4
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM
IST314I END

IST350I DISPLAY TYPE = EE
IST2002I ENTERPRISE EXTENDER AGGREGATE CONNECTION INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001            2
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.231                                4
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM
IST2009I RTP PIPES =         4      LU-LU SESSIONS      =         3  5
. . . .
IST314I END
```

Note the following explanation for Example 10-23:

- 1: The Display EE command is requesting summary information for a specific EE VIPA host name (responses for EELINK1 are shown).
- 2: Message IST2119I identifies the display command's correlator value. Each display command involving host name resolution is assigned a correlator value when the command is issued. Whenever VTAM resolves a host name to an IP address, it issues groups of messages during name resolution to keep the requestor informed of the progress. These message groups are not displayed all together as one multi-line response, and therefore, might become separated during the display process. Each message group that is associated (correlated) with the original request carries the correlator value of the original request.
- 3: Messages IST2120I and IST2121I indicate the status of the name resolution process.
- 4: Messages IST1680I and IST1910I confirm that the displayed information is associated with the specified EE VIPA from the original display request.
- 5: The remaining messages provide connection and traffic volume statistics, and are identical to the set of messages issued when an IP address is specified for the EE VIPA, as in Example 10-22 on page 343.

10.6 Diagnosis of the multiple EE VIPAs environment

This section includes the following diagnostic topics:

- “Diagnosing EE VIPA problems in OMPROUTE”
- “Diagnosing EE VIPA problems in TCP/IP”
- “Diagnosing EE VIPA problems in VTAM EE XCA”
- “Diagnosing EE VIPA problems in an EE SWNET”

10.6.1 Diagnosing EE VIPA problems in OMPROUTE

VIPA problems in OMPROUTE can be diagnosed using the following procedures:

- “Does OSPF have a routing table entry for each EE VIPA?”
- “Does OSPF have a valid routing table?”

Does OSPF have a routing table entry for each EE VIPA?

Example 10-24 shows how to determine if there is an OSPF routing table entry for each EE VIPA.

Example 10-24 Displaying OSPF routing table entries for EE VIPAs

```
ro sc30,d tcpip,tcpipa,omp,ospf,interface,name=eelink1                          1
ro sc30,d tcpip,tcpipa,omp,ospf,interface,name=eelink2                          2
EZZ7850I INTERFACE DETAILS 308
                INTERFACE ADDRESS:      10.10.1.232                             3
                ATTACHED AREA:          0.0.0.2
                PHYSICAL INTERFACE:     EELINK2                                 4
                INTERFACE MASK:         255.255.255.0
                INTERFACE TYPE:         VIPA
```

Note the following explanation for Example 10-24:

- **1** and **2**: Request a display of the OSPF routing table entry for the EE VIPAs.
- **3**: Each EE VIPA must have its own valid table entry.
- **4**: Each EE VIPA can be identified by its name and IP address. These must match what is defined in the TCP/IP profile on the DEVICE, LINK, and HOME statements.

Does OSPF have a valid routing table?

Example 10-25 shows how to determine whether OSPF itself is working properly by having a valid routing table. All the route entries should be inspected to confirm they are as expected.

Example 10-25 Displaying the OSPF routing table (RTTABLE)

```
ro sc30,d tcpip,tcpipa,omp,rttable                                              1
EZZ7847I ROUTING TABLE 293
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SPIA   0.0.0.0                 0  11      136250  10.10.2.1        (4)          2
 DIR*  10.10.1.0        FFFFFF00  1       136260  10.10.1.230      (3)
 SPF   10.10.1.130      FFFFFFFF  20      32739   10.10.2.134      (8)
. . . .
 DIR*  10.10.1.230      FFFFFFFF  1       136260  VIPALINK                      3
 DIR*  10.10.1.231      FFFFFFFF  1       136260  EELINK1                       4
 DIR*  10.10.1.232      FFFFFFFF  1       136260  EELINK2                       5
 SPF   10.10.1.240      FFFFFFFF  20      29749   10.10.2.244      (8)
```

```
. . . .
 SPF   10.10.1.242      FFFFFFFF  20      29749   10.10.2.244      (8)
 SPF*  10.10.2.0        FFFFFF00  10      136255  OSA2080LNK       (2)        6
 SPF*  10.10.3.0        FFFFFF00  10      136250  OSA20A0LNK       (2)        7
STAT*  10.10.20.0       FFFFFF00  0       136261  10.10.20.100
STAT*  10.10.20.101     FFFFFFFF  0       29766   10.10.20.100
STAT*  10.10.20.102     FFFFFFFF  0       136210  10.10.20.100
. . . . - more
```

Note the following explanation for Example 10-25:

- 1: Display the entire OSPF routing table. This could be a very large table.
- 2: Make sure all the DEFAULT routes are represented.
- 3: The stack's static VIPA must be present.
- 4 and 5: All EE VIPAs must be present.
- 6 and 7: All physical interfaces must be present

The remaining entries are routes to partner systems and learned routes that OSPF manages.

10.6.2 Diagnosing EE VIPA problems in TCP/IP

EE VIPA problems in TCP/IP can be diagnosed using the following procedures:

- "Is IUTSAMEH active between VTAM and TCP/IP?"
- "Does VTAM have active UDP ports in TCP/IP?"
- "Does each EE VIPA have active UDP ports in TCP/IP?"
- "Does each EE VIPA answer PING?"
- "Is each EE VIPA reachable via tracerte?"

Is IUTSAMEH active between VTAM and TCP/IP?

Example 10-26 shows how to determine if VTAM is active to TCP/IP.

Example 10-26 Displaying the status of IUTSAMEH and EZASAMEMVS

```
ro sc30,d tcpip,tcpipa,n,de,intfname=ezasamemvs                  1
EZZ2500I NETSTAT CS V1R8 TCPIPA 546
DEVNAME: IUTSAMEH         DEVTYPE: MPCPTP                        2
  DEVSTATUS: READY                                               3
  LNKNAME: EZASAMEMVS       LNKTYPE: MPCPTP      LNKSTATUS: READY 4
    NETNUM: N/A  QUESIZE: N/A
. . . .

ro sc30,d net,ee                                                 5
IST1685I TCP/IP JOB NAME = TCPIPA                                6
IST2003I ENTERPRISE EXTENDER XCA MAJOR NODE NAME = EEXCA         7
```

Note the following explanations for Example 10-26:

- 1: Display Netstat Devlinks for link EZASAMEMVS to check the status of the VTAM-to-TCP/IP connection. Remember, the EE XCA is not the only user of IUTSAMEH. In a CINET environment, multiple TCP/IP stacks on the same z/OS system might be communicating with each other through IUTSAMEH. If so, they also have activated IUTSAMEH and the VTAM-to-TCP/IP connection might not be the only user of it, see notes 5 - 7.

- 2, 3, and 4: Device IUTSAMEH and Link EZASAMEMVS show READY when VTAM has successfully connected to TCP/IP through IUTSAMEH.
- 5, 6, and 7: To be certain that there is an active VTAM-to-TCP/IP connection through the EE XCA and IUTSAMEH, use the Display EE command and look for messages IST1685I and IST2003I.

Does VTAM have active UDP ports in TCP/IP?

If VTAM does not have any UDP sockets open in TCP/IP, then the EE XCA major node has not been activated or has failed activation. See Example 10-20 on page 341 and Example 10-21 on page 342 to determine if the EE UDP ports (12000-12004) are in use by VTAM's started task. The following command can be used:

```
d tcpip,tcpipa,n,conn,client=net
```

Does each EE VIPA have active UDP ports in TCP/IP?

Example 10-27 shows how to determine if each EE VIPA has active local UDP sockets to TCP/IP.

Example 10-27 Displaying VTAM UDP sockets for a specific EE VIPA address

```
ro sc30,d tcpip,tcpipa,n,conn,client=net
ro sc30,d tcpip,tcpipa,n,conn,ipaddr=10.10.1.231
ro sc30,d tcpip,tcpipa,n,conn,ipaddr=10.10.1.232

EZD0101I NETSTAT CS V1R8 TCPIPA 812                       1
USER ID  CONN     STATE
0 OF 0 RECORDS DISPLAYED
END OF THE REPORT

EZD0101I NETSTAT CS V1R8 TCPIPA 484                       2
USER ID  CONN     STATE
NET      0000001D UDP
  LOCAL SOCKET:   10.10.1.231..12001
  FOREIGN SOCKET: *..*
NET      0000001C UDP
  LOCAL SOCKET:   10.10.1.231..12000
  FOREIGN SOCKET: *..*
NET      0000001E UDP
  LOCAL SOCKET:   10.10.1.231..12002
  FOREIGN SOCKET: *..*
NET      00000020 UDP
  LOCAL SOCKET:   10.10.1.231..12004
  FOREIGN SOCKET: *..*
NET      0000001F UDP
  LOCAL SOCKET:   10.10.1.231..12003
  FOREIGN SOCKET: *..*
NET      00000022 UDP
  LOCAL SOCKET:   10.10.1.232..12001
  FOREIGN SOCKET: *..*
NET      00000021 UDP
  LOCAL SOCKET:   10.10.1.232..12000
  FOREIGN SOCKET: *..*
NET      00000023 UDP
  LOCAL SOCKET:   10.10.1.232..12002
  FOREIGN SOCKET: *..*
```

```
NET      00000025 UDP
  LOCAL SOCKET:   10.10.1.232..12004
  FOREIGN SOCKET: *..*
NET      00000024 UDP
  LOCAL SOCKET:   10.10.1.232..12003
  FOREIGN SOCKET: *..*
. . . .
```

Note the following explanation for Example 10-27:

- 1: If no EE UDP ports are active to VTAM, then the EE XCA has not been activated.
- 2: If VTAM does have active EE UDP ports, they are associated with the EE VIPAs.

Does each EE VIPA answer PING?

Example 10-28 shows how to determine if EE VIPA addresses and host names respond to pings. We deleted the EELINK1 definition just for this example to show the PING failure when the VIPA is not defined or not active.

Example 10-28 Issuing a Ping to EE VIPA addresses and host names

```
ping 10.10.1.231                                           1
CS V1R8: Pinging host 10.10.1.231
Ping #1 timed out
***

ping sc30m-ee1.itso.ibm.com                                1
CS V1R8: Pinging host SC30M-EE1.ITSO.IBM.COM (10.10.1.231)
Ping #1 timed out
***

ping 10.10.1.232                                           2
CS V1R8: Pinging host 10.10.1.232
Ping #1 response took 0.000 seconds.
***

ping sc30m-ee2.itso.ibm.com                                2
CS V1R8: Pinging host SC30M-EE2.ITSO.IBM.COM (10.10.1.232)
Ping #1 response took 0.000 seconds.
***
```

Note the following explanation for Example 10-28:

- 1: If the VIPAs are not active or not defined properly, the PING will time out.
- 2: Otherwise, they will answer a Ping request.

Is each EE VIPA reachable via tracerte?

Example 10-29 shows how to determine if EE VIPA addresses and host names are reachable via tracerte.

Example 10-29 Issuing a tracerte to EE VIPAs and host names

```
tracerte 10.10.1.231                                1
CS V1R8: Traceroute to 10.10.1.231 (10.10.1.231):
1 10.10.3.2 (10.10.3.2)  1 ms  0 ms  0 ms
2 10.10.3.2 (10.10.3.2)  0 ms !H *  0 ms !H
3 * 10.10.3.2 (10.10.3.2)  0 ms !H *
```

```
4 10.10.3.2 (10.10.3.2)  0 ms !H *  0 ms !H
5 * 10.10.3.2 (10.10.3.2)  0 ms !H *
***

tracerte sc30m-ee1.itso.ibm.com                     1
CS V1R8: Traceroute to SC30M-EE1.ITSO.IBM.COM (10.10.1.231):
1 10.10.3.2 (10.10.3.2)  1 ms  0 ms  0 ms
2 10.10.3.2 (10.10.3.2)  0 ms !H *  0 ms !H
3 * 10.10.3.2 (10.10.3.2)  0 ms !H *
4 10.10.3.2 (10.10.3.2)  0 ms !H *  0 ms !H
***

tracerte 10.10.1.232                                2
CS V1R8: Traceroute to 10.10.1.232 (10.10.1.232):
1 10.10.1.232 (10.10.1.232)  0 ms  0 ms  0 ms
***

tracerte sc30m-ee2.itso.ibm.com                     2
CS V1R8: Traceroute to SC30M-EE2.ITSO.IBM.COM (10.10.1.232):
1 SC30M-EE2.ITSO.IBM.COM (10.10.1.232)  0 ms  0 ms  0 ms
***
```

Note the following explanation for Example 10-29:

- 1: If the VIPAs are not active or not defined properly, the TRACERTE will time out.
- 2: Otherwise, TRACERTE finds a route to them.

10.6.3 Diagnosing EE VIPA problems in VTAM EE XCA

VIPA problems in an EE XCA can be diagnosed using the following procedures:

- “Is each local EE VIPA’s address active in the EE environment”
- “Is each local EE VIPA’s hostname active in the EE environment”

Is each local EE VIPA’s address active in the EE environment

Example 10-30 shows how to determine if the EE VIPA address is active under the XCA.

Example 10-30 Displaying a local EE VIPA address

```
ro sc30,d net,ee,ip=10.10.1.231,sum                                   1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2002I ENTERPRISE EXTENDER AGGREGATE CONNECTION INFORMATION
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM
IST2009I RTP PIPES =          4      LU-LU SESSIONS      =          3 2
. . . .
ro sc30,d net,ee,ipaddr=10.10.1.232,sum                               1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2002I ENTERPRISE EXTENDER AGGREGATE CONNECTION INFORMATION
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM
IST2009I RTP PIPES =         11      LU-LU SESSIONS      =         10  2
. . . .
```

Note the following explanation for Example 10-30:

- 1: Display the EE status of the EE VIPA address to determine connection information.
- 2: Message IST2009I indicates there are active RTP pipes with LU-LU sessions. The EE VIPA is usable and is carrying connections.

Is each local EE VIPA’s hostname active in the EE environment

Example 10-31 shows how to determine if the EE VIPA host name is active under the XCA.

Example 10-31 Displaying a local EE VIPA host name

```
ro sc30,d net,ee,hostname=sc30m-ee1.itso.ibm.com                       1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001              2
IST2120I HOSTNAME RESOLUTION IN PROGRESS
IST314I END

IST350I DISPLAY TYPE = EE
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001              2
IST2121I HOSTNAME RESOLUTION COMPLETE
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM
IST314I END

IST350I DISPLAY TYPE = EE
IST2002I ENTERPRISE EXTENDER AGGREGATE CONNECTION INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001              2
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM
IST2009I RTP PIPES =          4      LU-LU SESSIONS      =          3  3
```

```
. . . .
ro sc30,d net,ee,hostname=sc30m-ee2.itso.ibm.com                       1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000002              2
IST2120I HOSTNAME RESOLUTION IN PROGRESS
IST314I END

IST350I DISPLAY TYPE = EE
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000002              2
IST2121I HOSTNAME RESOLUTION COMPLETE
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM
IST314I END

IST350I DISPLAY TYPE = EE
IST2002I ENTERPRISE EXTENDER AGGREGATE CONNECTION INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000002             2
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM
IST2009I RTP PIPES =         11      LU-LU SESSIONS      =        10 3
. . . .
```

Note the following explanation for Example 10-31:

- 1: Display the EE status of the EE VIPA host name to determine connection information.
- 2: All message groups are associated with the original display request by the CORRELATOR value. See Example 10-23 on page 344 for an explanation of the CORRELATOR value.
- 3: Message IST2009I indicates there are active RTP pipes with LU-LU sessions. The EE VIPA is usable and is carrying connections.

10.6.4 Diagnosing EE VIPA problems in an EE SWNET

VIPA problems in an EE SWNET can be diagnosed using the following procedures:

- “Are the remote partner’s addresses active in the EE environment?”
- “Are the remote partner’s hostnames active in the EE environment?”
- “What is the status of an EE partner’s switched PU names?”
- “Is a connectivity test between EE VIPAs successful?”
- “Is a connectivity test between EE host names successful?”
- “What does a pending EE VIPA connectivity test look like?”
- “What does a failed EE VIPA connectivity test look like?”

Are the remote partner’s addresses active in the EE environment?

Example 10-32 shows how to determine the status of remote partners using their IP addresses.

Example 10-32 Displaying partner EE VIPA addresses

```
ro sc30,d net,ee,ipaddr=(,10.10.1.131),sum                               1
. . . .
IST1680I LOCAL IP ADDRESS 10.10.1.231
```

```
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM
IST1680I REMOTE IP ADDRESS 10.10.1.131
IST1909I REMOTE HOSTNAME SC47M-EE1.ITSO.IBM.COM
IST2022I EE CONNECTION ACTIVATED ON 10/18/06 AT 06:05:58           2
IST2114I LIVTIME:    INITIAL =   10   MAXIMUM =   60   CURRENT =   60
IST2023I CONNECTED TO LINE EEM30002                                3
IST2024I CONNECTED TO SWITCHED PU EEP63047                         4
. . . .
ro sc30,d net,ee,ipaddr=(,10.10.1.132),sum                         1
. . . .
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM
IST1680I REMOTE IP ADDRESS 10.10.1.132
IST1909I REMOTE HOSTNAME SC47M-EE2.ITSO.IBM.COM
IST2022I EE CONNECTION ACTIVATED ON 10/18/06 AT 06:05:40           2
IST2114I LIVTIME:    INITIAL =   10   MAXIMUM =   60   CURRENT =   60
IST2023I CONNECTED TO LINE EEX30002                                3
IST2024I CONNECTED TO SWITCHED PU EEP73047                         4
. . . .
```

Note the following explanation for Example 10-32 (these response messages indicate that both EE VIPAs are being used on both endpoints):

- 1: Use the Display EE by IP address command to determine the connection status to a remote partner.
- 2: Message IST2022I shows the date and time when the connection was established.
- 3: Message IST2023I indicates the line name over which the connection is established.
- 4: Message IST2024I indicates the PU name assigned to the connection.

Are the remote partner's hostnames active in the EE environment?

Example 10-33 shows how to display the status of remote partners using their host names.

Example 10-33 Displaying partner EE VIPA host names

```
ro sc30,d net,ee,hostname=(,sc47m-ee1.itso.ibm.com),sum
ro sc30,d net,ee,hostname=(,sc47m-ee2.itso.ibm.com),sum
```

Note the following explanation for Example 10-33:

- The response messages that are of concern are the same messages discussed in Example 10-32.

What is the status of an EE partner's switched PU names?

Example 10-34 shows how to determine the status of remote partners using their PU names. The PU name can be obtained from message IST2024I, as shown in Example 10-32 on page 351.

Example 10-34 Displaying PU status of a connected EE partner

```
ro sc30,d net,ee,id=eep63047,sum                                1
ro sc30,d net,ee,id=eep73047,sum
. . . .
IST075I NAME = EEP73047, TYPE = PU_T2.1                         2
IST1680I LOCAL IP ADDRESS 10.10.1.232                           3
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM                  3
IST1680I REMOTE IP ADDRESS 10.10.1.132                          3
```

```
IST1909I REMOTE HOSTNAME SC47M-EE2.ITSO.IBM.COM                       3
IST2022I EE CONNECTION ACTIVATED ON 10/18/06 AT 06:05:40              4
. . . .
```

Note the following explanation for Example 10-34:

- 1: The status of a remote partner connection can be determined by displaying the PU name of the EE connection.
- 2: Message IST075I confirms the PU name from the display request.
- 3: IP addresses and host names of the local and remote partners confirms the connection.
- 4: Message IST2022I shows the date and time the connection was established.

Is a connectivity test between EE VIPAs successful?

Example 10-35 shows a *successful* connectivity test between the local VIPA and partner VIPA.

Example 10-35 Performing a connectivity test to an EE partner's VIPA

```
ro sc30,d net,eediag,test,ip=(10.10.1.231,10.10.1.131)                         1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000005                      2
IST2067I EEDIAG DISPLAY ISSUED ON 10/18/06 AT 23:06:22                         3
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1680I REMOTE IP ADDRESS 10.10.1.131
IST2023I CONNECTED TO LINE EEM3000F                                            4
IST2126I CONNECTIVITY TEST IN PROGRESS
IST314I END

IST350I DISPLAY TYPE = EEDIAG
IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000005
IST2131I EEDIAG DISPLAY COMPLETED ON 10/18/06 AT 23:06:22                      5
IST2132I LDLC PROBE VERSIONS: VTAM = V1           PARTNER = V1
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1680I REMOTE IP ADDRESS 10.10.1.131
IST924I -------------------------------------------------------------
IST2133I INTFNAME: OSA2080LNK              INTFTYPE: IPAQENET
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12000          6
```

Note the following explanation for Example 10-35:

- 1: The command requests a connectivity test between SC30 and SC47, using their EE VIPAs.
- 2: Message IST2119I shows the correlator value assigned to the original test request.
- 3: Message IST2067I indicates the date and time of the test request command.
- 4: Message IST2023I indicates the Line being used by the test.
- 5: Message IST2131I shows the data and time when the test finished.
- 6: Message IST2134I indicates the connectivity test was successful.

Example 10-36 on page 354 shows an *unsuccessful* connectivity test between the local VIPA and partner VIPA. (The reason for the failure is an inactive EE XCA major node at the remote end).

Example 10-36 Performing a connectivity test to an EE partner's VIPA

```
ro sc30,d net,eediag,test,ip=(10.10.1.232,10.10.1.121)                         1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000008                      2
IST2067I EEDIAG DISPLAY ISSUED ON 10/18/06 AT 23:10:51                         3
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.10.1.121
IST2023I CONNECTED TO LINE EEX30003
IST2126I CONNECTIVITY TEST IN PROGRESS
IST314I END

ro sc30,d net,eediag,pend                                                     4
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG
IST2145I PENDING ENTERPRISE EXTENDER DISPLAY COMMANDS
IST2067I EEDIAG DISPLAY ISSUED ON 10/18/06 AT 23:10:55                        5
IST924I -------------------------------------------------------------
IST2147I CORRELATOR: EE000008 LINE: EEX30003 STATUS: TEST-IN-PROGRESS         6
IST2148I EE CONNECTIVITY TEST REACHES MAXTIME ON 10/18/06 AT 23:11:51         7
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.10.1.121
IST924I -------------------------------------------------------------
IST2149I 1 OF 1 CORRELATORS DISPLAYED
IST314I END

IST350I DISPLAY TYPE = EEDIAG
IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000008                     8
IST2131I EEDIAG DISPLAY COMPLETED ON 10/18/06 AT 23:11:06                     9
IST2132I LDLC PROBE VERSIONS: VTAM = V1           PARTNER = UNKNOWN          10
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.10.1.121
IST924I -------------------------------------------------------------
IST2133I INTFNAME: OSA2080LNK              INTFTYPE: IPAQENET
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12000        11
IST2137I     2  10.10.2.1          D-1      RTT:     1
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12001
IST2137I     2  10.10.2.1          D-1      RTT:     0
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12002
IST2137I     2  10.10.2.1          D-1      RTT:     1
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12003
IST2137I     3  10.10.2.1          D-1      RTT:     1
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12004
IST2137I     3  10.10.2.1          D-1      RTT:     1
IST924I -------------------------------------------------------------
IST2139I CONNECTIVITY TEST RESULTS DISPLAYED FOR 1 INTERFACES
. . . .
```

In the example, we deactivated SC42's EE XCA major node so we could show this failed test. The remote IP address is a valid active VIPA on system SC42, but with the SC42 EE XCA inactive, there is no EE connection between SC30 (the local system) and SC42 (the remote system). The EE connectivity test is expected to fail.

Note the following explanation for Example 10-36:

- 1: The command requests a connectivity test between SC30 and SC42, using their EE VIPAs.
- 2: Message IST2119I shows the correlator value assigned to the initial test request.
- 3: Message IST2067I shows the date and time of the initial test request.
- 4: We entered a request to list all unfinished test requests still in progress (pending).
- 5: Message IST2067I shows the date and time of our display command that asked for a list of all other pending tests.
- 6: Message IST2147I shows the correlator value of the original unfinished test request, the line over which the test is running, and the status of the test right now.
- 7: Message IST2148I indicates the date and time that the original test will time-out if it has not finished by that time.
- 8: Another group of messages is correlated with the initial test request.
- 9: Message 2131I shows the test request has finished.
- 10: Message IST2132I indicates that the remote partner (UNKNOWN) does not have an EE connection with the local system.
- 11: Message IST2135I confirms an unsuccessful connectivity test. We expected this because we had deactivated the remote EE XCA major node in order to create this situation.

Is a connectivity test between EE host names successful?

Example 10-37 shows the commands to use in order to perform a connectivity test between the local host name and partner host name.

Example 10-37 Performing a connectivity test to an EE partner's host name

```
ro sc30,d net,eediag,test,hostname=(sc30m-ee1.itso.ibm.com,sc47m-ee1.itso.ibm.com)
ro sc30,d net,eediag,test,hostname=(sc30m-ee2.itso.ibm.com,sc42m-ee1.itso.ibm.com)
```

Note the following explanation for Example 10-37:

- The results are similar to those in Example 10-35 on page 353 and Example 10-36 on page 354.

What does a pending EE VIPA connectivity test look like?

Example 10-38 shows a pending connectivity test between the local VIPA and partner VIPA, where we used a remote IP address that did not exist. We did this on purpose so the test would last long enough for us to show the use of the *pending* display. While the test is attempting to find a way to the remote end, we issued a couple of *display pending* commands.

Example 10-38 Determining status of uncompleted EE VIPA connectivity tests

```
ro sc30,d net,eediag,test,ip=(10.10.1.232,10.10.10.10)                         1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000009                      2
IST2067I EEDIAG DISPLAY ISSUED ON 10/18/06 AT 23:15:15                         3
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.10.10.10
IST2023I CONNECTED TO LINE EEX30003
IST2126I CONNECTIVITY TEST IN PROGRESS
```

```
IST314I END

ro sc30,d net,eediag,pend                                                 4
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG
IST2145I PENDING ENTERPRISE EXTENDER DISPLAY COMMANDS
IST2067I EEDIAG DISPLAY ISSUED ON 10/18/06 AT 23:15:18                    5
IST924I -------------------------------------------------------------
IST2147I CORRELATOR: EE000009 LINE: EEX30003 STATUS: TEST-IN-PROGRESS     6
IST2148I EE CONNECTIVITY TEST REACHES MAXTIME ON 10/18/06 AT 23:16:15     7
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.10.10.10
IST924I -------------------------------------------------------------
IST2149I 1 OF 1 CORRELATORS DISPLAYED
IST314I END

ro sc30,d net,eediag,pend                                                 8
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG
IST2145I PENDING ENTERPRISE EXTENDER DISPLAY COMMANDS
IST2067I EEDIAG DISPLAY ISSUED ON 10/18/06 AT 23:15:54                    9
IST924I -------------------------------------------------------------
IST2147I CORRELATOR: EE000009 LINE: EEX30003 STATUS: TEST-IN-PROGRESS    10
IST2148I EE CONNECTIVITY TEST REACHES MAXTIME ON 10/18/06 AT 23:16:15    11
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.10.10.10
IST924I -------------------------------------------------------------
IST2149I 1 OF 1 CORRELATORS DISPLAYED
IST314I END
```

Note the following explanation for Example 10-38:

- **1**: The connectivity test specifies a remote IP address that is *not reachable*. We did this so we would have time to show the output of a couple of *display pending* displays.
- **2** and **3**: The correlator value and time stamp of the initial test request.
- **4** and **5**: This is the first of two Display Pending commands requesting status information for unfinished tests, and the time it was entered.
- **6** and **7**: The response to our first Display Pending command is a group of messages that show the status of our initial test and when it will time-out. The group is identified by the correlator value that was assigned to our initial test request (**1**).
- **8** and **9**: This is the second of two Display Pending commands requesting status information for unfinished tests, and the time it was entered.
- **10** and **11**: The response to our second Display Pending command is a group of messages that show the status of our initial test and when it will time-out. The group is identified by the correlator value that was assigned to our initial test request (**1**).

What does a failed EE VIPA connectivity test look like?

Example 10-39 on page 357 shows a *failed* connectivity test between the local system and an unreachable remote partner. We knew ahead of time that the partner was unreachable. In order to show the messages involved in a failed test, we specified an IP address that did not exist.

Example 10-39 Recognizing a failed EE connectivity test

```
ro sc30,d net,eediag,test,ip=(10.10.1.232,10.10.10.10)                        1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000009
IST2067I EEDIAG DISPLAY ISSUED ON 10/18/06 AT 23:15:15                        2

IST350I DISPLAY TYPE = EEDIAG
IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000009
IST2131I EEDIAG DISPLAY COMPLETED ON 10/18/06 AT 23:16:15                     3
IST2132I LDLC PROBE VERSIONS: VTAM = V1           PARTNER = UNKNOWN
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.10.10.10
IST924I -------------------------------------------------------------
IST2133I INTFNAME: OSA2080LNK              INTFTYPE: IPAQENET
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12000         4
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED              5
IST2137I     1  10.10.2.1                  RTT:     1
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12001
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  10.10.2.1                  RTT:     1
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12002
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  10.10.2.1                  RTT:     1
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12003
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  10.10.2.1                  RTT:     1
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12004
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  10.10.2.1                  RTT:     1
IST924I -------------------------------------------------------------
IST2139I CONNECTIVITY TEST RESULTS DISPLAYED FOR 1 INTERFACES
IST314I END
```

Note the following explanation for Example 10-39:

- 1: We entered a test request for a remote IP address that we know is not reachable. We expect to see the group of messages that indicate the test timed-out.
- 2: Message IST2067I shows when the test started.
- 3: Message IST2131I shows when the test completed. Notice that the time difference is one minute (60 seconds is the default time-out value). The time-out value for a specific test can be specified using the MAXTIME parameter on the test command itself.
- 4 and 5: Messages IST2135I and IST2136I show that the test failed due to a time-out. This is indicative of an unreachable IP address or one that does not respond.

10.7 Summary

This chapter explained the use of single and multiple VIPAs in the Enterprise Extender (EE) environment. Guidelines for implementing multiple EE VIPAs were reviewed. Examples of configuration definitions and commands for monitoring the multiple EE VIPAs environment were described.

11

Implementing an EBN connection for z/OS

This chapter discusses the recommended steps to implement an Extended Border Node (EBN) configuration in a z/OS environment.

In this chapter we provide the necessary definitions to create a highly available connection between two Advanced Peer to Peer Networking (APPN) networks using a pair of EBNs on each side of the APPN subnet boundary. We show how APPN searches across NETID boundaries can be controlled and what can be done to reduce the number of APPN locate flows on CP-CP sessions.

The following topics will be discussed:

- "Overview of Extended Border Node"
- "Design considerations in an EBN environment"
- "Configuring VTAM in the native network to be an EBN"
- "Verifying the EBN implementation"
- "Diagnosing EBN/EE problems"

11.1 Overview of Extended Border Node

A single APPN environment consists of APPN nodes and APPN transmission groups (TGs), all belonging to the same APPN subnet. It is determined by the use of a common Network Identifier NETID. In a traditional Systems Network Architecture (SNA) subarea, an SNA Network also consisted of multiple nodes all belonging to the same SNA Network Identifier. To communicate with LUs in other networks, an SNA Network Interconnection (SNI) connection had to be implemented. To interconnect two different APPN environments an APPN boundary must be built, consisting of two Border Nodes (BNs) connecting to each other.

Figure 11-1 shows a typical EBN configuration.

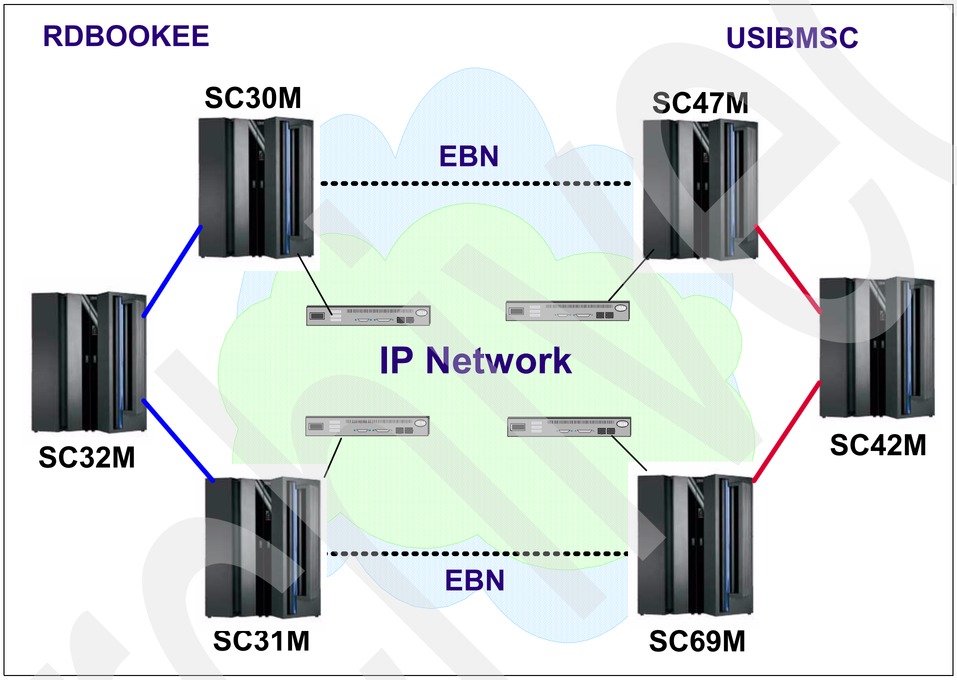


Figure 11-1 High available EBN configuration

This section describes the required steps to enable the Border Node function in VTAM.

> **Note:** For SLU initiated sessions to traverse an APPN boundary, the Border Node must support Session Services Extensions option set in APPN. Only *Extended* Border Nodes provide this functionality and the only platform that supports this function as of today is Virtual Telecommunications Access Method (VTAM). Besides Communications Server on z/OS, VTAM on z/VM and VSE operating systems also have EBN functionality and can be used to connect to a different APPN environment using traditional links. However they do not provide HPR/IP connectivity.

11.2 Design considerations in an EBN environment

This section describes the design considerations of an EBN implementation. The section covers the following topics:

- “Number of EBNs and connectivity options”
- “Global Virtual Routing Node (GVRN): Connection Network”

- “Security Concerns: SME/DSME, IPSec versus Session Level Encryption”

11.2.1 Number of EBNs and connectivity options

For high availability reasons we recommend using a pair of EBNs on each side of the APPN environment boundary, with one EBN on each side connecting to one EBN on the non-native side. We do not recommend connecting all EBNs to each other as this increases complexity in APPN search algorithms and does not provide additional redundancy.

11.2.2 Global Virtual Routing Node (GVRN): Connection Network

The use of Global Connection Network across network boundaries is possible and recommended whenever possible to avoid Automatic Network Routing (ANR) in intermediate APPN nodes. However, there are some aspects that need to be verified before deciding to implement Global Connection networks.

Connection Network across Network Address Translation (NAT) requires the use of IP host names instead of IP addresses and proper Domain Name System (DNS) setup in both, your and your business partner’s DNS systems.

Connection Network across firewall infrastructure requires all possible IP address combinations that want to communicate across the connection network to be defined in firewall rules

11.2.3 Security Concerns: SME/DSME, IPSec versus Session Level Encryption

Prior to z/OS V1R8, the Session Management Exit (SME) that controls SNA Network Interconnection (SNI) sessions is no longer used at EBNs. A Directory Services Management Exit (DSME) is used instead to control APPN locates associated with session setup.

11.3 Configuring VTAM in the native network to be an EBN

This section describes the necessary steps to enable the EBN function and to define connections to an external network.

VTAM consists of the following steps:

- “VTAM start options: BN, BNDYN, and BNORD”
- “VTAM major nodes: SWNET, ADJCP, ADJCLUST and Model CDRSC”

11.3.1 VTAM start options: BN, BNDYN, and BNORD

In order to enable the Border Node function, VTAM must be restarted to pick up the BN=YES start option. The BN start option cannot be changed dynamically. The other two start options BNDYN and BNORD can be modified dynamically using the F NET,VTAMOPTS,option=value command.

Example 11-1 lists the additional start options that enable the EBN function in VTAM.

Example 11-1 Start options to enable APPN Border Node function in VTAM

```
BN=YES,                                                                 X 1
BNDYN=NONE,                                                             X 2
```

```
BNORD=DEFINED,                                                       X 3
```

Here are explanatory notes to the start options in Example 11-1:

- 1: BN=YES enables the APPN Border Node function. It is only valid on APPN Network Nodes.

> **Attention:** Coding BNDYN=NONE requires that you code and activate ADJCLUST tables, also for your own NETID, otherwise existing LUs will not be found after VTAM restart.
>
> F NET,VTAMOPTS,BNDYN=LIMITED can be issued to temporarily change this option.

- 2: BNDYN=NONE provides full control of how searches are performed based on NETID. We recommend coding BNDYN=NONE and BNORD=DEFINED to reduce the number of APPN locates in the network. For more details on this subject, see “ADJCLUST Table: Native Network” on page 365.
- 3: BNORD=DEFINE advises VTAM to strictly follow the predefined search order.

Example 11-2 shows the start options after restart as a Border Node.

Example 11-2 D NET, VTAMOPTS after restart with BN=YES

```
D NET,VTAMOPTS
IST097I DISPLAY ACCEPTED
IST1188I VTAM CSV1R8 STARTED AT 13:36:50 ON 10/17/06 812
IST1349I COMPONENT ID IS 5695-11701-180
IST1348I VTAM STARTED AS NETWORK NODE   1
IST1189I AFFDELAY = 600                 ALSREQ   = NO
IST1189I API64R   = YES                 APPNCOS  = #CONNECT
IST1189I ASIRFMSG = OLUSSCP             ASYDE    = TERM
IST1189I AUTHLEN  = YES                 AUTORTRY = AUTOCAP
IST1189I AUTOTI   = 0                   BN       = YES
IST1189I BNDYN    = NONE                BNORD    = DEFINED
```

Example 11-2 shows the BN related start options after VTAM restart. Even though the BN is a new node type, VTAM still shows ‘NETWORK NODE’ in IST1348I.

11.3.2 VTAM major nodes: SWNET, ADJCP, ADJCLUST and Model CDRSC

This section describes all the major nodes that are created on the native VTAM SC30M:

- “MODEL CDRSC Major Node”
- “SWNET major node”
- “ATCCON: Sequence of activation of EBN related major nodes”

SWNET major node

The switched major nodes that define the links to the non-native EBNs will be activated on our own EBNs only. We decided to have parallel TGs addressing two different VIPAs to use a separate IP infrastructure. See 10.1, “Overview of Enterprise Extender VIPA requirements” on page 322 for a discussion about multiple VIPAs.

Example 11-3 shows the switched major node of SC31M.

Example 11-3 SWNET major node on EBN defining parallel TGs to external EBN

```
EESWEB31 VBUILD TYPE=SWNET
************************************************************************
* SWITCHED MAJORNODE FOR ENTERPRISE EXTENDER TO SC31M USING            *
* PARALLEL TGS TO USE SEPERATE IP INFRASTRUCTRE                        *
*   CPCP AND SENSITIVE SESSION ONLY VIA SLOWER BUT SECURE LINK         *
*              COMMON PREFIX FOR EE PU                                 *
*              |  TGN                                                  *
*              |  |SYSCLONE_LEFT                                       *
* 10.10.1.241  |  || SYSCLONE_RIGHT                  10.10.1.141       *
*  _____    |  |  || |                                 |  _____        *
* |     |-EE1--EEP63169-TG6--(PUBLIC IP)----2M--------EE1-|     |   1 *
* |SC31M|                                                 |SC69M|      *
* |_____|-EE2--EEP73169-TG7--(SECURE IP)---256K-------EE2-|_____|   2 *
*           |                                          |               *
* 10.10.1.242                                        10.10.1.142       *
************************************************************************
*******************************************************************
*  TG6 = FAST BUT UNSECURE LINK TO SC69M
*******************************************************************
EEP63169 PU 2  CPNAME=SC69M,NETID=USIBMSC,TGN=06, *
               VERALSID=YES, 1                                         *
               TGP=EEXTWAN,CAPACITY=2M,SECURITY=UNSECURE,        3    *
            7  CONNTYPE=APPN,CPCP=NO,HPR=YES,DISCNT=NO,DYNLU=YES,      *
            4  ISTATUS=ACTIVE,DWACT=YES,DWINOP=YES
EEPT6T69 PATH  HOSTNAME=SC69M-EE1.ITSO.IBM.COM,SAPADDR=4,    5 6 *
               REDIAL=FOREVER,REDDELAY=31,                             *
               GRPNM=EEGVL&SYSCLONE.1
*******************************************************************
*  TG7 = SLOW BUT SECURED LINK TO SC69M
*******************************************************************
EEP73169 PU    CPNAME=SC69M,NETID=USIBMSC,TGN=07, *
               VERALSID=YES,        * PUNAME HAS TO MATCH AT BOTH ENDS*
               TGP=EEXTWAN,CAPACITY=256K,SECURITY=ENCRYPT,             *
               CONNTYPE=APPN,CPCP=YES,HPR=YES,DISCNT=NO,DYNLU=YES,     *
               ISTATUS=ACTIVE,DWACT=YES,DWINOP=YES
EEPT7T69 PATH  HOSTNAME=SC69M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=31,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

Here, we assume that we have two possible IP infrastructures to connect to a business partner. One consisting of a high speed connection but not encrypted route, the second route through a different ISP with slower speed but secured with IPSec tunnels. With APPN route selection based upon APPN Class of Service, we define that all ‘normal’ sessions should use the high speed route. CP-CP sessions and sessions with sensitive data that require a SECURE class of service must take the slower route. So from an APPN perspective, we have two parallel TGs with different characteristics addressing different destination VIPAs.

Here are explanatory notes to Example 11-3:

- 1: We must ensure, that the PU selected by VTAM on an inbound XID is the one that matches the same TG number defined at the other end. As both TGs will have the same destination CPNAME, we cannot use NETID.CPNAME alone to identify which PU definition to select. With VERALSID=YES VTAM will inspect the incoming XID for the link station name and select the PU with the same label. Hence, both sides must specify the same PU name for this to work. Therefore, the PUNAME EEP63169 (2) has to match on both sides of the connection.

- 3: By choosing different TG characteristics we assign different weights to an APPNCOS. For instance a session requiring an APPNCOS=INTERSC will not be able to set up a session across TG6 as this is defined as UNSECURE.
- 4: In order to avoid XID race condition problems, we recommend deciding on a client-server relationship between the EBNs. The client is responsible for activation and recovery of the link in case of an outage.
- 5: The path statement contains a host name that must be resolved into the static VIPA at the remote side. If NAT devices are involved, this must be the NAT device's IP address as known in your IP network.

> **Attention:** Using host names in any VTAM major node requires VTAM to issue get_host_by_name() calls. VTAM must therefore have an on OMVS segment and the necessary user authorization in the SAF product.

- 6: VTAM uses 8 as the default destination SAP when starting an EE link. The source SAP is always 4. Changing the default destination SAP to 4 helps to avoid XID race conditions when both sides try to activate the same link at the same time.
- 7: CP-CP sessions are considered to contain sensitive data and, therefore, should always use the secured TG. CPCP=NO prohibits those sessions over the unsecured link.

Activation of the SWNET major node towards the EBN

Example 11-4 shows the activation of the two parallel TGs and CP-CP sessions between SC31M and SC69M.

Example 11-4 Activation of the two parallel TGs towards SC69M

```
V NET,ACT,ID=EESWEB31
IST097I VARY ACCEPTED
IST093I EEP63169 ACTIVE
IST093I EEP73169 ACTIVE
IST093I EESWEB31 ACTIVE
IST2180I  DYNLU = YES FOR USIBMSC.SC69M     SET FROM EEP63169
IST590I  CONNECTOUT ESTABLISHED FOR PU EEP63169 ON LINE EEM31000
IST590I  CONNECTOUT ESTABLISHED FOR PU EEP73169 ON LINE EEX31000
IST1086I APPN CONNECTION FOR USIBMSC.SC69M IS ACTIVE - TGN = 7
IST1086I APPN CONNECTION FOR USIBMSC.SC69M IS ACTIVE - TGN = 6
IST1488I  ACTIVATION   OF RTP CNR00047 AS PASSIVE TO USIBMSC.SC69M
IST1488I  ACTIVATION   OF RTP CNR00046 AS ACTIVE  TO USIBMSC.SC69M
IST1096I CP-CP SESSIONS WITH USIBMSC.SC69M ACTIVATED

D NET,ADJCP,ID=USIBMSC.SC69M,E
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CONTROL POINT 126
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1197I ADJCP MAJOR NODE = EEADJCP
IST1101I ADJCP DISPLAY SUMMARY FOR USIBMSC.SC69M
IST1102I NODENAME          NODETYPE CONNECTIONS CP CONNECTIONS NATIVE
IST1103I USIBMSC.SC69M      NN        2           1            NO
IST2157I ALIASRCH = NO
IST1104I CONNECTION SUMMARY FOR USIBMSC.SC69M
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
```

```
IST1106I EEP73169 AC/R     7 YES   98550000000000000000A09100808080
IST1106I EEP63169 AC/R     6 NO    906D0000000000000000019100808080
IST1500I STATE TRACE = OFF
IST1493I RTP SUMMARY FOR USIBMSC.SC69M COUNT = 2 RTPONLY = NO
IST1486I   RTP NAME   STATE             DESTINATION CP     MNPS  TYPE
IST1487I   CNR00047   CONNECTED         USIBMSC.SC69M      NO    CPCP
IST1487I   CNR00046   CONNECTED         USIBMSC.SC69M      NO    CPCP
IST314I END
```

ADJCLUST Table: Native Network

The Adjacent Cluster Table is required if BNDYN=NO is coded. Here you specify the Subnet Routing lists on a NETID basis. Note that we used MVS symbols. The *&SYSNAME.M* resolves to our own SSCPNAME, and *&BROTHER.* to our native neighbor EBN's SSCPNAME. Using system symbolics makes it possible for both EBNs to activate the same adjacent cluster table. If system symbolics cannot be used, then separate ADJCLUST tables must be defined for each VTAM.

Example 11-5 shows the ADJCLUST table activated in the EBNs in the native NETID REDBOOKEE.

Example 11-5 Adjacent Cluster Table in SC30M and SC31M

```
***********************************************************************
* RDBOOKEE      |       USIBMSC  BNDYN=NONE                           *
*  _____        |        _____                                        *
* |     |       |       |     |  DEFAULT AND NATIVE NETID: SNVC1      *
* |SC30M|______TG7______|SC47M|    1ST AND ONLY CHOICE LOCAL SSCPNAME *
* |_____|       |       |_____|    SNVC=1                             *
*    |          |          |     ADJACENT NETID: SNVC=2               *
*    |          |          |       1ST CHOICE  NON-NATIVE EBN         *
*  __|__        |        __|__     BACKUP NATIVE ADJ EBN              *
* |     |       |       |     |                                       *
* |SC31M|______TG7______|SC69M|                                       *
* |_____|               |_____|                                       *
*                                                                     *
EEADJCLS VBUILD TYPE=ADJCLUST
********************************************
DEFAULT  NETWORK BNDYN=NONE 1
         NEXTCP  CPNAME=&SYSNAME.M,SNVC=1
NATIVE   NETWORK NETID=RDBOOKEE,BNDYN=NONE  2
         NEXTCP  CPNAME=&SYSNAME.M,SNVC=1
SNVC2    NETWORK NETID=USIBMSC,BNDYN=NONE   3
         NEXTCP  CPNAME=USIBMSC.SC47M,SNVC=2
         NEXTCP  CPNAME=USIBMSC.SC69M,SNVC=2
         NEXTCP  CPNAME=&BROTHER.,SNVC=2
```

Note the following explanation for Example 11-5:

- 1: There is one entry without NETID coded which is used for ALIAS searches, when the real NETID has not yet been determined. This is called the DEFAULT and contains only our own CPNAME representing our own NETID. This has the effect, that we do not forward any search request to other companies that do not provide a real network identifier in their session request. Using a "MODEL CDRSC Major Node" definition we assign a real NETID and can then use any of the following Subnet Routing Lists (SRL).

- 2: The NATIVE entry is used when a search is received that contains our own NETID. Only our own CPNAME is listed here which represents the whole native APPN environment, including our brother EBN. Because we do not want our own network to be searched twice for non-existent LUs, we do not list our neighbor EBN in this list. Existing LUs must be found in the first search.
- 3: The USIBMSC network contains the two EBNs, only one is active (SC47M). Searches for specific CPs are sent to SC47M. Only if the CP-CP sessions to SC47M are lost, will the searches for existing CPs be sent to an active neighbor EBN. The neighbor can then forward the locates to its external non-native EBN.

Verification of the ADJCLUST table

The displays in Example 11-6 on SC30M list the Subnet Routing Lists after activation of the ADJCLUST table. It shows the output of the D NET, ADJCLUST commands on SC30M.

Example 11-6 Display of ADJCLUST tables on SC30M

```
D NET,ADJCLUST                  1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE 431
IST1325I DEFINED TABLE FOR DEFAULT_NETID  DYNAMICS = NONE
IST1326I CP NAME           TYPE    STATE      STATUS       SNVC
IST1327I RDBOOKEE.SC30M    DEFINED ACTIVE     *** N/A ***  N/A
IST314I END

D NET,ADJCLUST,NETID=RDBOOKEE    2
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE 434
IST1325I DEFINED TABLE FOR RDBOOKEE       DYNAMICS = NONE
IST1326I CP NAME           TYPE    STATE      STATUS       SNVC
IST1327I RDBOOKEE.SC30M    DEFINED ACTIVE     *** N/A ***  N/A
IST314I END

D NET,ADJCLUST,NETID=USIBMSC      3
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE 447
IST1325I DEFINED TABLE FOR USIBMSC        DYNAMICS = NONE
IST1326I CP NAME           TYPE    STATE      STATUS       SNVC
IST1327I USIBMSC.SC47M     DEFINED ACTIVE     FOUND        002
IST1327I USIBMSC.SC69M     DEFINED NOT ACTIVE NOT SEARCHED 002
IST1327I RDBOOKEE.SC31M    DEFINED ACTIVE     NOT SEARCHED 002
IST314I END
```

Here are explanatory notes to Example 11-6:

- 1:The DEFAULT NETID only shows CPNAME SC30M.
- 2: NETID=RDBOOKEE only lists SC30M.
- 3: For the non-native NETID we search the ACTIVE non-native EBN first

The same table is activated on SC31M resulting in the output shown Example 11-7.

Example 11-7 Display of ADJCLUST table on SC31M

```
D NET,ADJCLUST                  1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE 848
IST1325I DEFINED TABLE FOR DEFAULT_NETID  DYNAMICS = NONE
```

```
IST1326I CP NAME           TYPE    STATE      STATUS       SNVC
IST1327I RDBOOKEE.SC31M    DEFINED ACTIVE     *** N/A ***  N/A
IST314I END

D NET,ADJCLUST,NETID=RDBOOKEE    2
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE 851
IST1325I DEFINED TABLE FOR RDBOOKEE       DYNAMICS = NONE
IST1326I CP NAME           TYPE    STATE      STATUS       SNVC
IST1327I RDBOOKEE.SC31M    DEFINED ACTIVE     *** N/A ***  N/A
IST314I END

D NET,ADJCLUST,NETID=USIBMSC      3
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE 854
IST1325I DEFINED TABLE FOR USIBMSC        DYNAMICS = NONE
IST1326I CP NAME           TYPE    STATE      STATUS       SNVC
IST1327I USIBMSC.SC47M     DEFINED NOT ACTIVE NOT SEARCHED 002
IST1327I USIBMSC.SC69M     DEFINED ACTIVE     FOUND        002
IST1327I RDBOOKEE.SC30M    DEFINED ACTIVE     NOT SEARCHED 002
IST314I END
```

Here are explanatory notes to Example 11-7:

- 1: The DEFAULT NETID only shows CPNAME SC31M.
- 2: NETID=RDBOOKEE only lists SC31M.
- 3: For the non-native NETID we search the ACTIVE non-native EBN first. If this is not available we search SC30M.

The same table is activated on SC31M resulting in the output as shown in Example 11-7.

ADJCP Major Node

Our global definitions allow adjacent control point definitions to be dynamically activated. In general we do not need predefined ADJCPs. The exception is ADJCP definitions for external EBNs. In z/OS 1.8 a new parameter was introduced to enforce REAL searching on inbound APPN locate flows. ALIASRCH=NO can be coded on the ADJCP definition.

> **Attention:** We strongly encourage you to predefine the ADJCP for non-native EBNs and prohibit inbound alias searching by coding ALIASRCH=NO. ALIAS searching most often occurs for inactive or non-existing resources in the originating network and cause APPN broadcasts in your and other networks, which can be very CPU consumptive in your NNs.

Example 11-8 shows the ADJCP major node defining the EBNs.

Example 11-8 Adjacent CP major node on both EBNs

```
**********************************************************************
*  ADJCP MAJORNODE TO REJECT INCOMING ALIAS SEARCHES SG24-7359      *
*  ADJACENT EBN SHOULD RESOLVE ALIAS TO REAL LUS                     *
*  BEFORE SERCHING OTHER NETIDS                                      *
**********************************************************************
EEADJCPS VBUILD TYPE=ADJCP
*
SC47M    ADJCP NETID=USIBMSC,ALIASRCH=NO,RTPONLY=NO
```

```
SC69M    ADJCP NETID=USIBMSC,ALIASRCH=NO,RTPONLY=NO
```

In Example 11-8 we force USIBMSC’s EBNs to search for fully-qualified resources in our NETID.

Verification of ALIASRCH=NO

Example 11-9 shows the new settings of ALIASRCH.

Example 11-9 Display of the ADJCP major node showing ALIASRCH NO

```
D NET,ID=EEADJCP,E
IST097I DISPLAY ACCEPTED
IST075I NAME = EEADJCP, TYPE = ADJCP MAJOR NODE 014
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1100I ADJACENT CONTROL POINTS FROM MAJOR NODE EEADJCP
IST1102I NODENAME         NODETYPE CONNECTIONS CP CONNECTIONS NATIVE
IST1103I USIBMSC.SC47M     *NA*       0           0            *NA*
IST2157I ALIASRCH = NO  1
IST1103I USIBMSC.SC69M     *NA*       0           0            *NA*
IST2157I ALIASRCH = NO  1
IST314I END
```

Here are explanatory note to Example 11-9:

- 1: ALIASRCH=NO will cause inbound APPN locates to be rejected unless they contain the real NETID of the target LU.

MODEL CDRSC Major Node

In order to control searches across APPN Borders, we recommend resolving ALIAS names to REAL names by assigning the NETID to the target LU. This way, the APPN search algorithm can use NETID specific Subnet Routing LISTs (SRL). There are several ways to do alias translation. We chose the use of a MODEL CDRSC major node. This major node must be active on the Central Directory Servers which in our scenario are also the EBNs.

Example 11-10. shows how the use of MODEL CDRSC definitions can simplify ALIAS resolution.

Example 11-10 MODEL CDRSC major node on Central Directory Servers

```
EECDRSC  VBUILD TYPE=CDRSC
*
         NETWORK NETID=RDBOOKEE 1
SC30* 1  CDRSC
SC31*    CDRSC
SC32*    CDRSC
         NETWORK NETID=USIBMSC  2
SC*   2  CDRSC CPNAME=SC47M
WTSCSC47 CDRSC CPNAME=USIBMSC.SC47M     3
         NETWORK NETID=DEIBMIV
IV* 4 CDRSC CPNAME=USIBMSC.SC47M,                                  X
               MODETAB=ALLMODES
         NETWORK NETID=DEIBMPX
PX*   4  CDRSC CPNAME=USIBMSC.SC47M,                                  X
               MODETAB=ALLMODES
```

Here are explanatory notes to Example 11-10:

- 1: We use wildcards to specify a model CDRSC because we found that a very strict naming convention was already in place. Most of our applications started with our system name. This definition assigns the NETID RDBOOKEE to all target LUs that start with our system name. These entries are needed if inbound ALIAS searches arrive.
- 2: All other applications starting with SC are to be found in NETID USIBMSC. Those applications must be defined with a CPNAME so that they are also defined in APPN's directory database and can be found by APPN locates.
- 3: All applications that do not adhere to the naming standards must be defined with their full name.
- 4: The assigned NETID does not have to match the NETID of the defined owning CNAME; in fact, the CPNAME can be any CPNAME.

> **Note:** The CPNAME on the CDRSC definition does not have to be correct. If it is incorrect, APPN will find out and update the directory entry with learned information.

ATCCON: Sequence of activation of EBN related major nodes

It is important to activate the major nodes in the correct sequence. Example 11-11 lists the additional members that need to be activated on the native EBN VTAMs.

Example 11-11 Additional entries in ATCCONxx of EBNs

```
EEADJCP,                   ADJACENT CP TABLE       SG24-7359      1 +
EEADJCLS,                  ADJACENT CLUSTER TABLE  SG24-7359        +
EECDRSC,                   CDRSC TO RESOLVE ALIAS  SG24-7359        +
EESWEB30,                  START LINKS TO EBNS     SG24-7359      2 +
```

Here are explanatory notes to Example 11-11:

- 1: The ADJCP definition must be active before any connection to the external VTAM can activate. Otherwise a dynamic ADCJP with the default ALIASRCH=YES will be used when the link activates.
- 2: The SWNET major node should be the last in this sequence, but still be activated before the XCA major node so that we can always use the predefined PU definitions.

11.4 Configuring VTAM in the non-native network to be an EBN

This section discusses the necessary definitions in the EBNs of the external network. We show how the adjacent cluster table should look like to improve efficiency in APPN searching.

Adjacent cluster table: non-native network

On the EBNs in the non-native NETID we coded two separate unique ADJCLUST tables as we did not have systems symbolics set up.

Example 11-12 shows the adjacent cluster table as activated in SC69M. A similar, yet not same, table is activated in SC47M.

Example 11-12 Unique ADJCLUST table in USIBMSC.SC69M

```
***********************************************************************
* RDBOOKEE      |      USIBMSC  BNDYN=NONE                           *
*  _____        |       _____                                        *
* |     |       |       |     |  DEFAULT AND NATIVE NETID: SNVC1      *
* |SC30M|______TG7______|SC47M|    1ST AND ONLY CHOICE LOCAL SSCPNAME *
* |_____|               |_____|    SNVC=1                             *
*    |          |          |     ADJACENT NETID: SNVC=2               *
*    |          |          |       1ST CHOICE  NON-NATIVE EBN         *
*   __|__       |        __|__     BACKUP NATIVE ADJ EBN              *
* |     |       |       |     |                                       *
* |SC31M|______TG7______|SC69M|                                       *
* |_____|               |_____|                                       *
*                                                                     *
EEADJC69 VBUILD TYPE=ADJCLUST         * ACTIVE ON SC69M               *
***********************************************************************
DEFAULT  NETWORK BNDYN=NONE                  1
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC69M  * SNVC=1 ONLY NATIVE
NATIVE   NETWORK NETID=USIBMSC,BNDYN=NONE    2
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC69M  * SNVC=1 ONLY NATIVE
RDBOOKEE NETWORK BNDYN=NONE,NETID=RDBOOKEE   3
         NEXTCP  SNVC=2,CPNAME=RDBOOKEE.SC31M * SNVC=2 SEARCH 1 HO
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC47M
```

Here are explanatory notes to Example 11-12:

- The default (1) and native (2) NETID only specify the local CPNAME.
- The non-native (3) NETID lists the non-native EBN first. Only if this is not available the native parallel EBN in our own NETID will be searched in the hope that it still has connectivity to the remote NETID.

11.5 Verifying the EBN implementation

This section describes how you can verify your definitions and manage implementation.

Verification of APPN Subnet Routing Lists with APING

Example 11-13 shows the successful session setup across the network boundary on SC47M.

Example 11-13 APINGs on SC47M showing native and cross network session setups

```
D NET,APING,ID=USIBMSC.SC42M             1
IST097I DISPLAY ACCEPTED
IST1489I APING SESSION INFORMATION 097
IST1490I DLU=USIBMSC.SC42M SID=ECF3DCDA0C7508DB
IST933I LOGMODE=#INTER  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #INTER
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I  21  USIBMSC.SC42M      APPN         RTP
IST314I END

D NET,APING,ID=RDBOOKEE.SC32M           2
IST097I DISPLAY ACCEPTED
IST1488I ACTIVATION OF RTP CNR00034 AS ACTIVE TO RDBOOKEE.SC32M
IST1488I ACTIVATION OF RTP CNR00035 AS ACTIVE TO RDBOOKEE.SC32M
IST1489I APING SESSION INFORMATION 089
IST1490I DLU=RDBOOKEE.SC32M SID=ECF3DCDA0C7508D8
IST933I LOGMODE=#INTER  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #INTER
```

```
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   7  RDBOOKEE.SC31M     ISL          RTP
IST1461I  21  RDBOOKEE.VRNLOCAL  APPN         RTP
IST1461I  21  RDBOOKEE.SC32M     APPN         RTP
IST314I END
```

Here are explanatory notes to Example 11-13:

- 1: An APING command issued on SC47M results in a session setup to SC42M in the own netid, using TG21 XCF links as the session path.
- 2: An APING to the EN in the non-native NETID results in a successful session setup using the Inter Subnet Link (ISL) TG7 to the external EBN, from there via connection network toward the EN.

> **Note:** D NET,APING without a specified NETID does *not* perform an alias search. VTAM will assume its own NETID and the search will be a real search.

To verify the opposite direction we issued APING commands from RDBOOKEE into USIBMSC.

Example 11-14 shows a successful session from EN RDBOOKEE.SC32M to EN USIBMSC.SC42M.

Example 11-14 APING from EN RDBOOKEE.SC32M towards USIBMSC.SC42M

```
D NET,APING,ID=USIBMSC.SC42M
IST097I DISPLAY ACCEPTED
IST1489I APING SESSION INFORMATION 174
IST1490I DLU=USIBMSC.SC42M SID=DE1FD59B6FDA63FB
IST933I LOGMODE=#INTER  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #INTER
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I  21  W3IBMCOM.VRN       APPN         RTP       1
IST1461I  21  USIBMSC.SC47M      ISL          RTP
IST1461I  21  USIBMSC.SC42M      APPN         RTP
IST314I END
```

Example 11-14 shows that the session sets up successfully using the global connection network (1).

11.6 Diagnosing EBN/EE problems

This section includes the following diagnostic topics:

- “Using display commands”
- “Gathering traces”

11.6.1 Using display commands

Several display commands are available, which can help in diagnosing EBN problems.

D NET,ADJCLUST

The command in Example 11-15 displays the Adjacent Cluster Table as it is currently used by VTAM.

Example 11-15 ADJCLUST

```
D NET,ADJCLUST,NETID=USIBMSC
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CLUSTER TABLE 954
IST1325I DEFINED TABLE FOR USIBMSC        DYNAMICS = NONE
IST1326I CP NAME           TYPE    STATE      STATUS       SNVC
IST1327I USIBMSC.SC47M     DEFINED ACTIVE     NOT SEARCHED 002
IST1327I USIBMSC.SC69M     DEFINED NOT ACTIVE NOT SEARCHED 002
IST1327I RDBOOKEE.SC31M    DEFINED ACTIVE     FOUND        002
IST314I END
```

Note that in Example 11-15, we currently do not search the direct EBN but ask our neighbor first.

D NET,APING

APING can also be used to verify that the sessions can set up correctly across the APPN border. Example 11-16 shows a successful APING session setting up.

Example 11-16 APING

```
D NET,ID=USIBMSC.SC42M,APING
IST097I DISPLAY ACCEPTED
IST1576I  DYNAMIC SWITCHED MAJOR NODE ISTDSWMN CREATED
IST2180I  DYNLU = YES FOR USIBMSC.SC42M     SET FROM CNV00094
IST590I  CONNECTOUT ESTABLISHED FOR PU CNV00094 ON LINE EEX30002
IST1086I  APPN CONNECTION FOR USIBMSC.SC42M     IS ACTIVE - TGN =  21
IST1488I  ACTIVATION   OF RTP CNR00095 AS ACTIVE  TO USIBMSC.SC42M
IST1488I  ACTIVATION   OF RTP CNR00096 AS ACTIVE  TO USIBMSC.SC42M
IST2180I  DYNLU = YES FOR USIBMSC.SC69M     SET FROM CNV00097
IST590I  CONNECTOUT ESTABLISHED FOR PU CNV00097 ON LINE EEX30003
IST1086I  APPN CONNECTION FOR USIBMSC.SC69M     IS ACTIVE - TGN =  21
IST1488I  ACTIVATION   OF RTP CNR00098 AS ACTIVE  TO USIBMSC.SC69M
IST1488I  ACTIVATION   OF RTP CNR00099 AS ACTIVE  TO USIBMSC.SC42M
IST1489I APING SESSION INFORMATION 935
IST1490I DLU=USIBMSC.SC42M SID=DE1FDCBBF9BF9E2A
IST933I LOGMODE=#INTER  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #INTER
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I  21  W3IBMCOM.VRN       APPN         RTP
IST1461I  21  USIBMSC.SC69M      ISL          RTP
IST1461I  21  USIBMSC.SC42M      APPN         RTP
IST314I END
```

Note that the session sets up over a global connection network W3IBMCO.VRN.

D NET,RTPS,CPNAME=netid.*

The D NET,RTPS command can be specified with wildcards, which is a very efficient method to see which HPR pipes are active in a foreign network.

Example 11-17 on page 373 lists all pipes into NETID USIBMSC.

Example 11-17 Display all pipes into a specified netid

```
D NET,RTPS,CPNAME=USIBMSC.*
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = RTPS 957
IST1695I PU NAME       CP NAME      COSNAME SWITCH CONGEST STALL SESS
IST1960I CNR00096 USIBMSC.SC42M     SNASVCMG  NO     NO     NO      1
IST1960I CNR00095 USIBMSC.SC42M     RSETUP    NO     NO     NO      0
IST1960I CNR0000B USIBMSC.SC47M     CPSVCMG   NO     NO     NO      1
IST1960I CNR0000A USIBMSC.SC47M     CPSVCMG   NO     NO     NO      1
IST2084I 4 OF 4 MATCHING RTP PIPES DISPLAYED
IST314I END
```

Note that in Example 11-17, we have two CPSVCMG pipes to the same destination CP.

D NET,EEDIAG,ID=puname

If firewalls are involved, this command can be used to see whether all the required UDP ports are allowed in the FW filters.

Example 11-18 shows the result of a EEDIAG,TEST=YES display command.

Example 11-18 EEDIAG TEST=YES

```
D NET,EEDIAG,ID=EEP63047,TEST=YES
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG 968
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001
IST2067I EEDIAG DISPLAY ISSUED ON 10/27/06 AT 15:19:13
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1680I REMOTE IP ADDRESS 10.10.1.131
IST2023I CONNECTED TO LINE EEM3000E
IST2126I CONNECTIVITY TEST IN PROGRESS
IST314I END
IST350I DISPLAY TYPE = EEDIAG 969
IST350I DISPLAY TYPE = EEDIAG 969
IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001
IST2131I EEDIAG DISPLAY COMPLETED ON 10/27/06 AT 15:19:13
IST2132I LDLC PROBE VERSIONS: VTAM = V1           PARTNER = V1
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1680I REMOTE IP ADDRESS 10.10.1.131
IST924I -------------------------------------------------------------
IST2133I INTFNAME: OSA2080LNK             INTFTYPE: IPAQENET
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12000
IST2137I     1  10.10.1.131                 RTT:     1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12001
IST2137I     1  10.10.1.131                 RTT:     1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12002
IST2137I     1  10.10.1.131                 RTT:     1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12003
IST2137I     1  10.10.1.131                 RTT:     1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12004
IST2137I     1  10.10.1.131                 RTT:     1
IST924I -------------------------------------------------------------
IST2139I CONNECTIVITY TEST RESULTS DISPLAYED FOR 1 INTERFACES
```

```
IST314I END
```

Here is an explanatory note to Example 11-18:

- All tests were successful, we have full IP connectivity to 10.10.1.131.

11.6.2 Gathering traces

Packet Trace

In the first approach, the TCP/IP packet trace is probably the best tool to use when it comes to diagnosing EE problems. Ideally, traces should be taken at both ends of the connection as most problems are caused by IP packets not reaching the destination. Therefore you must verify that the packets which were sent are also received at the destination.

See 13.2, “Traces in CS for z/OS” on page 411 for information about how to collect traces in a z/OS environment.

12

Advanced APPN topics

This chapter describes advanced topics that are usually not considered, because there is no direct requirement to do so. However in some cases, an APPN environment could benefit by implementing them.

In this chapter, the following topics are discussed:

- "APPN route selection"
- "APPN topology and ARB algorithm"
- "RTPONLY for adjacent networks"

12.1 APPN route selection

APPN Route Calculation is done based on a *Class of Service table (COSTAB)*. For every *APPN Class of Service (APPNCOS)* all nodes and transmission groups (TGs) in the Advanced Peer-to-Peer Networking (APPN) topology are assigned a weight based on their TG or node characteristics. When a route through the APPN environment is calculated, the Network Node (NN) doing the route calculation inspects all possible paths from the origin Control Point (CP) to the destination CP, adds the weights of all TGs and nodes along the path and compares all possible paths with regards to their total weight. The route with the least weight will be selected to carry the new session. If there are multiple equally weighted routes, the NN tries to load balance evenly across all available paths. While this looks like a good idea at a first glance, there are some issues with multiple equally weighted paths.

12.1.1 Multiple equally weighted routes versus single preferred routes

Here are some aspects to be considered when using multiple possible paths:

- The number of High Performance Routing (HPR) pipes increases dramatically with every additional same-weight TG and additional hop along the path. This can cause a higher CPU and storage utilization, especially when HPR is extended into the branches with thousands of rapid transport protocol (RTP) endpoints.
- Problem diagnosis is more complex, as sessions do not take a predictable path.
- If the number of sessions over an HPR pipe is low, HPR pipes may not operate at optimal performance when there is not enough traffic to ramp up to the desired speed.

To solve all these issues, we decided to have a single preferred route only, with a single second best route and a single third choice. Still we wanted to use all the available links to carry LU-LU sessions, so we decided to prefer the available TGs based on the APPNCOS differently:

- The APPN route selection for standard LU-LU sessions between LPARs must:
  - prefer EE (TG6) for interactive sessions with MPC (TG2) as backup
  - prefer MPC (TG2) for medium and batch sessions with EE (TG6) as backup
  - use Cross Coupling Facility (XCF) links only as a last resort in case both EE and Multi Path Channel (MPC) connectivity is lost
- CP-CP sessions should prefer XCF over FICON and EE

  Doing so, we utilize all available TGs but still have a minimum of active HPR pipes. We can easily predict which path a session will take if we know the APPNCOS of the session and we can use APING to use a predictable HPR pipe by selecting an appropriate logmode and are able to test the performance of an already active pipe.

The logic that applies to connections to other external EE nodes is as follows:

- Prefer direct TGs over connection network TGs (VRN) whenever they exist.
- If no direct connection exists, use a path via the local connection network first.
- If no local network connectivity exists, use a global connection network
- Only if no connection network path is available, use a multi-hop-route via intermediate NNs.

12.1.2 Changing TG characteristics: Changing UPARM1-3 in TG Profiles

To achieve our goal described above, we need to assign different weights to the available TGs. For LU-LU sessions, we have APPNCOS values available that map to three priorities. Within the characteristics that describe a TG there are also three parameters that can be used to modify the standard weights: UPARM1, UPARM2, and UPARM3. We are using UPARM1 to determine the weight for high priority sessions, UPARM2 for the medium priority sessions and UPARM3 for the low priority sessions. At the end of this step we have to modify the APPN Class of Service table to assign weights based on those UPARM settings.

Transmission Group Profiles

A *Transmission Group Profile (TGP)* defines a set of TG characteristics that can be assigned to a link or multiple links, for example, capacity, security properties, and user defined parameters (UPARM). IBM supplies a list of commonly used characteristics in SYS1.SAMPLIB member IBMTGPS. D NET,TGPS lists all TGPs that VTAM knows about and can, therefore, be coded on the PU statements. The latest version of IBMTGPS should be copied into VTAMLST, modified to your needs, and put in ATCCON to be activated automatically when VTAM is started. We copied IBMTGPS from SYS1.SAMPLIB and saved the table as EETGPS in VTAMLST and put the new member name into ATCCONxx. to be activated automatically after a VTAM restart.

Table 12-1 lists the modifications to the existing IBMTGPS member.

Table 12-1 Modified Transmission Group Profiles from IBMTGPSS

TG Profile	Original	Changed/added	Reason

XCF	CAPACITY=25M	CAPACITY=600M
		UPARM1=60
		UPARM2=60
		UPARM3=60
			More accurate value
			3rd choice for #INTER
			3rd choice for #CONNECT
			3rd choice for #BATCH

FICON	SECURITY=UNSECURE	SECURITY=SHIELDED
		UPARM1=40
		UPARM2=40
		UPARM3=20
			2nd choice for #INTER
			2nd choice for #CONNECT
			1st choice for #BATCH

HIPERSOC	SECURITY=UNSECURE	SECURITY=SHIELDED
		UPARM1=20
		UPARM2=20
		UPARM3=40
			1st choice for #INTER
			1st choice for #CONNECT
			2nd choice for #BATCH

GIGENET		UPARM1=20
		UPARM2=20
		UPARM3=20
			1st choice for all APPNCOS on direct links versus VRN connections

FASTENET		UPARM1=40
		UPARM2=40
		UPARM3=40
			2nd choice for #INTER
			2nd choice for #CONNECT
			2nd choice for #BATCH

EEXTCAMP	PDELAY=PACKET
	CAPACITY=4M
		PDELAY=NEGLIGIB
		CAPACITY=100M
		UPARM1=80
		UPARM2=80
		UPARM3=80
			more accurate
			more accurate
			for local VRN, if available prefer predefined TG

EEXTWAN	CAPACITY=56K	CAPACITY=2M
		UPARM1=90
		UPARM2=90
		UPARM3=90
			more accurate
			for global VRN, if available prefer local VRN

Note the following explanation for Table 12-1:

- The major change to the existing TGPs were the UPARM1, UPARM2, and UPARM3 values. The numbers we assigned reflect the actual weight that those TGs will have after we modified the Class of Service table. For example, a UPARM1=40 will result in a weight of 40 for #INTER sessions, causing TGs with this profile to be second best choice as there will be a better TG with a weight of 20 if a TG with a GIGENET or HIPERSOC profile is active.

The D NET,TGPS command can be issued to verify that our definitions are active.

Example 12-1 shows a display of all available TGPs known to Virtual Telecommunications Access Method (VTAM).

Example 12-1 Display of the active Transmission Group Profiles

```
D NET,TGPS
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TGPS 701
IST1107I TGP NAME TG CHARACTERISTICS
...                  1                         2
IST1108I XCF      00AF0000000000000000604C003C3C3C
...
IST1108I EEXTCAMP 009A0000000000000000014C005A5A5A
IST1108I EEXTWAN  006D00000000006400000191005A5A5A
...
IST1108I GIGENET  00B40000000000000000014C00141414
IST1108I FICON    00AF0000000000000000C04C00281414
...
IST1108I HIPERSOC 00B40000000000000000C04C00142828
IST1454I 26 TGP(S) DISPLAYED
IST314I END
```

Note the following explanation for Example 12-1:

- **1:** In the display above, the second byte in the TG characteristics is the hexadecimal representation of the capacity.
- **2:** The last 3 bytes are the hexadecimal representation of UPARM1, UPARM2, UPARM3.

We now have a set of characteristics defined that we can later apply to our APPN link definitions by referring to a profile with the TGP= parameter on the PU statement.

> **Important:** the Transmission Group Profile member must be active before any PU with a TGP= parameter gets activated. Make sure to place it to the top of your ATCCON member.

12.1.3 Changing the APPN Class of Service table

In our scenario we use UPARM1-3 to select a preferred TG for a certain APPNCOS.

In “Transmission Group Profiles” on page 377 we assigned UPARM1, UPARM2 and UPARM3 values to TGPs. In this section we create an APPN Class of Service table to assign weights based on UPARM values. In order to achieve this, the existing Class of Service table ISTACST2 must be modified to narrow down the allowed ranges for UPARM1, UPARM2 and UPARM3 for each APPNCOS entries.

Example 12-2 shows the modifications done to the #INTER Class of Service.

Example 12-2 Extract of #INTER Class of Service in EECOSTAB

```
#INTER   APPNCOS  PRIORITY=HIGH,NUMBER=12   transmission priority
         LINEROW  WEIGHT=20,       1        line row weight            *
               NUMBER=1,                    line row number            *
               UPARM1=(20,20),     2        1st choice        *MBUR   *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(100M,MAXIMUM),     line speed                 *
               COSTTIME=(0,64),             cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=1,                 node row number            *
               WEIGHT=5,                    node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,31)              route addition resistance
         LINEROW  WEIGHT=40,                line row weight            *
               NUMBER=2,            3       line row number            *
               UPARM1=(40,40),              2nd choice        *MBUR   *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(100M,MAXIMUM),     line speed                 *
               COSTTIME=(0,64),             cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
```

The weights for the #INTER APPNCOS with the PRIORITY=HIGH should be controlled by the UPARM1 setting.

Note the following explanation for Example 12-2:

- **1**: LINEROW NUMBER=1 assigns a WEIGHT=20 only, if *all* defined criteria for *all* characteristics are fulfilled.
- **2**: The allowed range for UPARM1 in LINEROW NUMBER=1 is narrowed down to the value 20 only. All other UPARM1 values will be considered as not eligible and proceed to the next LINEROW to pass the test.
- **3**: LINEROW NUMBER=2 assigns a WEIGHT=40 only, if *all* defined criteria for *all* characteristics are fulfilled. For UPARM1 the value has to be 40, otherwise a WEIGHT=40 cannot be assigned.

This method is applied for every TG to every APPNCOS. We simplified the process by only focussing on UPARM1 values, providing a one-to-one mapping of UPARM value to TG weight. The algorithm still takes all characteristics into account..

Analogous to the UPARM1 parameter for the #INTER APPNCOS, we use the same method with UPARM2 for #CONNECT and UPARM3 for #BATCH.

Table 12-2 lists the result of our modification: UPARMX dictates the weight of APPNCOS X.

Table 12-2 TG weights based on UPARM and APPNCOS

	UPARM1 1	**UPARM2 2**	**UPARM3 3**	**WEIGHT**
APPNCOS's 1 #INTER #INTERSC SNASVCMG	**20-20**	ANY	ANY	**20**
	40-40	ANY	ANY	**40**
	60-60	ANY	ANY	**60**
APPNCOS 2 #CONNECT	ANY	**20-20**	ANY	**20**
	ANY	**4040**	ANY	**40**
	ANY	**60-60**	ANY	**60**
APPNCOS 3 #BATCH #BATCHSC	ANY	ANY	**20-20**	**20**
	ANY	ANY	**40-40**	**40**
	ANY	ANY	**60-60**	**60**

Attention: All NNs in the network should calculate routes using the same Class of Service table, otherwise HPR pipes may permanently switch to a *perceived* better path when in fact there is none. Make sure all NNs activate the same table before the first session can set up. Put EECOSTAB right after EETGPS into ATCCONxx.

12.1.4 Verification of the new COSTAB

Activating a new COSTAB using V NET,ACT,ID=EECOSTAB will override existing APPNCOS entries and add new ones. Old entries that are not in the new table will be kept unchanged.

Example 12-3 shows a D NET,COS,TYPE=APPN command that can be used to verify, which APPNCOS entries are available, which Class of Service table is in use by VTAM and when it was last activated.

Example 12-3 Display the active APPN Class of Service Table

```
D NET,COS,TYPE=APPN
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = APPN COS
IST1782I ENTRY NAME     TABLE NAME     ACTIVATION TIME
IST1783I CPSVCMG        EECOSTAB       10/05/06  10:35:26
IST1783I SNASVCMG       EECOSTAB       10/05/06  10:35:26
IST1783I #CONNECT       EECOSTAB       10/05/06  10:35:26
IST1783I #INTER         EECOSTAB       10/05/06  10:35:26
IST1783I #INTERSC       EECOSTAB       10/05/06  10:35:26
IST1783I #BATCH         EECOSTAB       10/05/06  10:35:26
IST1783I #BATCHSC       EECOSTAB       10/05/06  10:35:26
IST314I END
```

The APPNCOS values above are the standard Classes of Services available on every SNA protocol stack. We recommend using one of those and refrain from creating new ones, because this would impose that every APPN node (EN or NN, z/OS or distributed platform) must update their Class of Service table to include those new entries as well.

12.1.5 Assigning weights to Transmission Groups

This section discusses how weights can be assigned to Transmissions Groups.

Planning weight assignments in the APPN environment

The order of preference of Transmission Groups needs to be assessed in each installation. There is no general rule as to which link type is “best” to be used. We decided to NOT use XCF links for normal data traffic, as this might add load to the coupling facility which might already be busy serving other subsytems. For bulk data transmission we decided on MPC connections as they provide a robust and very efficient DLC type with large NLP sizes without involving other software products. For interactive traffic we chose to use EE as the preferred method.

Table 12-3 lists all TGPs that we use and their associated weights showing the preferred TGP with the lowest weight in bold.

Table 12-3 TGPs and their weights

TGP	XCF 1	FICON 2	HIPERSOC 3
TGN	21	2	6
CPSVCMG	**30**	60	90
SNASVCMG	60	40	**20**
#INTER	60	40	**20**
#INTERSC	60	40	**20**
#CONNECT	60	**20**	40
#BATCH	60	**20**	40
#BATCHSC	60	**20**	40

Note the following explanations to Table 12-3:

- 1: XCF links with TG21 will only be first choice for CP-CP sessions. We decided to keep HPR traffic off the coupling facility links because those may already be used for other sysplex related functions and we did not want to put additional load to the coupling facility when there are other DLCs available.
- 2: FICON MPCs will be carrying #CONNECT, #BATCH, and #BATCHSC traffic only. If MPCs are not available, those HPR pipes should switch to EE (HIPERSOC).
- 3: EE links will be used for SNASVCMG, #INTER, and #INTERSC sessions primarily. If EE fails, they will use MPC.

Model major node for XCF

Now that we have modified the Class of Service table and corresponding TGP entries in our Transmission Group Profile member, we need to assign TGPs to the PU definitions. The first links that activate between VTAMs in a sysplex are XCF links. The PUs representing those links are defined dynamically and put in a major node called ISTLSXCF. VTAM provides a model PU to be used if TG characteristics of those XCF links must be overridden to meet certain requirements. We use this model major node to specify the TGP=XCF and thus assign a weight of 60 for all classes of service with the exception of CPSVCMG, as we want to run CP-CP sessions across XCF.

Example 12-4 shows the model Major Node for the XCF links.

Example 12-4 Model Major Node for XCF links on all VTAMs

```
***********************************************************************
* MODEL Major NODE TO ASSIGN A TGP TO DYNAMIC XCF LINKS
***********************************************************************
MODELXCF VBUILD TYPE=MODEL
ISTP*    PU    TRLE=*,CPCP=YES,TGP=XCF
```

> **Note:** As with all model definitions, the XCF model PU will show a RESET status when displayed. This is normal and not to worry about.

LOCAL SNA Major Node for MPC links

The PUs that represent the MPC channels are defined in a Local SNA major node.

Example 12-5 lists the Local SNA Major Node in SC32M.

Example 12-5 Local SNA Major Node in SC32M

```
***********************************************************************
* LOCAL MAJOR NODE FOR MPC CONNECTIONS TO SC31 AND SC30
***********************************************************************
MPC4SC32 VBUILD TYPE=LOCAL
MPC2SC31 PU    PUTYPE=2,XID=YES,NETID=RDBOOKEE,CPNAME=SC31M,TGN=2,     *
               CPCP=YES,CONNTYPE=APPN,TGP=FICON,                       *
               TRLE=TRL2SC31
MPC2SC30 PU    PUTYPE=2,XID=YES,NETID=RDBOOKEE,CPNAME=SC30M,TGN=2,     *
               CPCP=YES,CONNTYPE=APPN,TGP=FICON,                       *
               TRLE=TRL2SC30
```

Note the following explanation for Example 12-5:

- The TGP definition refers to the FICON TG profile in EETGPS.

Switched major node for EE connections between VTAMs

The third type of connections that will be available between VTAMs are EE connections. Those are defined in a SWNET Major Node.

Example 12-6 lists the Switched Major Node of the EN SC32M.

Example 12-6 SWNET MAjor Node on SC32M

```
EESWEN32 VBUILD TYPE=SWNET
*******************************************************************
*  Link to primary NNS SC30M
*******************************************************************
EEPUNN30 PU    CPNAME=SC30M,NETID=RDBOOKEE,TGN=06,                     *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,DISCNT=NO,               *
               DWACT=YES,DWINOP=YES,                                   *
               TGP=HIPERSOC
EEPTNN30 PATH  HOSTNAME=SC30M-EE2.ITSO.IBM.COM,SAPADDR=4,               *
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
*******************************************************************
*  Link to backup  NNS SC31M
```

```
*******************************************************************
EEPUNN31 PU    CPNAME=SC31M,NETID=RDBOOKEE,TGN=06,                     *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,DISCNT=NO,               *
               DWACT=YES,DWINOP=YES,                                   *
               TGP=HIPERSOC
EEPTNN31 PATH  HOSTNAME=SC31M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

Note the following explanation to Example 12-6:

- The TGP=HIPERSOC refers to a Transmission Group Profile in EETGPS where the TG characteristics are defined.

12.1.6 Verification of the TG weights

Now we have completed the preparation of the EECOSTAB, EETGPS and the Major Nodes defining the PUs, we must to verity that the weights are what we think they should be.

Example 12-7 shows a D NET,TOPO,ORIG=xx,DEST=yy,APPNCOS=appncos command showing the different weights for the various APPNCOS values.

Example 12-7 Topology display showing the weights

```
D NET,TOPO,ORIG=SC30M,DEST=SC31M,TGN=21,APPNCOS=CPSVCMG
ST1299I TRANSMISSION GROUPS ORIGINATING AT CP RDBOOKEE.SC30M
IST1357I                                                 CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.SC31M    21       OPER     INTERM      YES   30
...
IST1736I                   PU NAME
IST1737I                   ISTP3031

D NET,TOPO,ORIG=SC30M,DEST=SC31M,TGN=21,APPNCOS=#INTER
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP RDBOOKEE.SC30M
IST1357I                                                 CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.SC31M    21       OPER     INTERM      YES   60

...
IST1304I                   SECURITY UPARM1   UPARM2      UPARM3
IST1305I                   SECURE   60       60          60
IST1579I                   ------------------------------------------
IST1736I                   PU NAME
IST1737I                   ISTP3031
```

Note the following explanation for Example 12-7:

- XCF has a weight of 30 for CPSVCMG Class of Service and a weight of 60 for the #INTER APPNCOS.

Example 12-8 shows how APING commands can be used to verify the predictable session path.

Example 12-8 APINGs with different logmodes using different TGs

```
D NET,APING,ID=SC31M,LOGMODE=#INTER    1
IST097I DISPLAY ACCEPTED
IST1489I APING SESSION INFORMATION 460
IST1490I DLU=RDBOOKEE.SC31M SID=DE1FD59B7147096C
IST933I LOGMODE=#INTER  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #INTER
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   6  RDBOOKEE.SC31M     APPN         RTP
IST314I END

D NET,APING,ID=SC31M,LOGMODE=#CONNECT 2
IST097I DISPLAY ACCEPTED
IST1488I  ACTIVATION   OF RTP CNR00036 AS ACTIVE
IST1488I  ACTIVATION   OF RTP CNR0243A AS PASSIVE
IST1489I APING SESSION INFORMATION 471
IST1490I DLU=RDBOOKEE.SC31M SID=DE1FD59B71470970
IST933I LOGMODE=#CONNECT, COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #CONNECT
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC31M     APPN         RTP
IST314I END

D NET,APING,ID=SC31M,LOGMODE=#BATCH 3
IST097I DISPLAY ACCEPTED
IST1488I  ACTIVATION   OF RTP CNR02438 AS PASSIVE
IST1488I  ACTIVATION   OF RTP CNR00034 AS ACTIVE
IST1488I  ACTIVATION   OF RTP CNR00035 AS ACTIVE
IST1488I  ACTIVATION   OF RTP CNR02439 AS PASSIVE
IST1489I APING SESSION INFORMATION 466
IST1490I DLU=RDBOOKEE.SC31M SID=DE1FD59B7147096D
IST933I LOGMODE=#BATCH  , COS=*BLANK*
IST875I APPNCOS TOWARDS SLU = #BATCH
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC31M     APPN         RTP
IST314I END
```

Note the following explanation for Example 12-8:

- **1**: The APING session with LOGMODE=#INTER maps to an APPNCOS= #INTER and is using TG6, the EE link.
- **2**: The APING session with LOGMODE=#CONNECT maps to APPNCOS= #CONNECT and is using TG2, the MPC link.
- **3**:The APING session with LOGMODE=#BATCH maps to APPNCOS=#BATCH and uses TG2, the MPC link.

Tip: There is no LOGMODE #CONNECT available by default. We recommend defining one in ISTINCLM and distribute it to all VTAMs (see Figure 12-9 on page 397). If non-VTAM EE nodes are in the network, the #CONNECT logmode should also be defined there, so that it can be used by APING to test connectivity across #CONNECT HPR pipes.

Example 12-9 LOGMDOE #CONNECT source code

```
************************************************************************
*                                                                      *
*        LOGMODE #CONNECT to be used with APING                        *
************************************************************************
#CONNECT MODEENT LOGMODE=#CONNECT,FMPROF=X'13',TSPROF=X'07',           *
               ENCR=B'0000',SSNDPAC=7,RUSIZES=X'F7F7',                 *
               SRCVPAC=7,PSNDPAC=7,APPNCOS=#CONNECT                    *
```

12.2 APPN topology and ARB algorithm

The Adaptive Rate Based (ARB) flow and congestion control algorithm in HPR provides a very efficient mechanism that aims at using the available bandwidth but not overloading the network thus causing a packet loss. It does this by adjusting the allowed send rate to the current capabilities of the network and the receiving RTP endpoint. An RTP endpoint is allowed to send a certain amount of data to the remote RTP endpoint during a given interval. That amount of data is called the allowed send rate. Periodically, an RTP endpoint sends an NLP containing an ARB request. The remote RTP endpoint sends back an ARB reply with one of five possible values based on current conditions at the remote RTP endpoint and network delays. The ARB reply value indicates whether the sending RTP endpoint (that sent the ARB request) can raise its send rate, keep the send rate the same, or reduce its send rate by 12.5%, 25% or 50%. The allowed send rate value for an RTP connection is adaptive and changes based on conditions in the network and remote RTP endpoint. The send rate is unidirectional, which means that on a given RTP connection the amount of data RTP endpoint A is allowed to send to RTP endpoint B is not necessarily the same as the amount of data RTP endpoint B is allowed to send to RTP endpoint A.

There are two versions of ARB. The initial ARB algorithm proved to be not efficient enough to compete for bandwidth in multi protocol networks. So when HPR/IP was first implemented in a newer version of ARB called *ARB-2* or *responsive mode ARB* was added to the architecture and is now available on all the platforms providing HPR/IP support. We recommend keeping the default start option HPRARB=RESPMODE to operate based on the new algorithm.

Example 12-10 shows the send rates of an HPR pipe.

Example 12-10 HPR pipe and send rates in VTAM

```
D NET,ID=CNR0000A
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR0000A, TYPE = PU_T2.1 683
IST486I STATUS= ACTIV---X-, DESIRED STATE= ACTIV
IST1043I CP NAME = EECPBURK - CP NETID = RDBOOKEE - DYNAMIC LU = YES
IST1589I XNETALS = YES
IST2178I RPNCB ADDRESS 11FD5800
IST1963I APPNCOS = #INTER - PRIORITY = HIGH
IST1476I TCID X'1BE4112D00010051' - REMOTE TCID X'0000000004002FB4'
IST1481I DESTINATION CP RDBOOKEE.EECPBURK - NCE X'80'
IST1587I ORIGIN NCE X'D000000000000000'
IST1966I ACTIVATED AS ACTIVE ON 10/26/06 AT 06:50:52
IST1477I ALLOWED DATA FLOW RATE = 5263 KBITS/SEC 1
```

```
IST1516I INITIAL DATA FLOW RATE = 47 MBITS/SEC 2
IST1841I ACTUAL DATA FLOW RATE = 3775 KBITS/SEC 3
IST1511I MAXIMUM NETWORK LAYER PACKET SIZE = 1458 BYTES
...
IST1959I DATA FLOW STATE = NORMAL
```

Note the following explanation for Example 12-10:

- 1: The ALLOWED DATA FLOW RATE is what the sending RTP may currently use.
- 2: The INITIAL DATA FLOW rate is what the RTP pipe started with initially or after a path switch occurred.
- 1: The ACTUAL DATA FLOW is what the sending RTP currently is using.

Remember, the send rates are unidirectional. They can and probably will be different if looked at from the other side of the RTP pipe.

Example 12-11 shows the same pipe displayed in PCOMM.

Example 12-11 HPR pipe and send rates in PCOMM

```
csdspy rtp /d3 > rtps.txt
Name                            @R000004
First Hop                       EEPUBURK
Destination                     RDBOOKEE.SC32M
Active Sessions                 0
Class Of Service Name           #INTER
Maximum BTU Size                1461
Timer                           180
Local TCID                      0000000004000000
Remote TCID                     1BE4112D00010051 1
Bytes Sent                      2174
Bytes Received                  506102
Bytes Resent                    0
Bytes Discarded                 0
Packets Sent                    144
Packets Received                750
Packets Resent                  0
Packets Discarded               0
Send Rate                       798 2
Receive Rate                    4740 3
Up Time                         0,000
Round Trip Time                 0
Burst Size                      4991
SRT Expires                     1704
Inbound Incorrect SNA Frames    0
Inbound Session Control Frames  2
Outbound Session Control Frames 2
```

Notes to Example 12-11:

- 1: The Transport Connection IDentifiers (TCIDs) can be used to correlate HPR pipes at different RTP endpoints. The local TCID uniquely identifies an HPR pipe within a local RTP, the remote TCID is unique for this HPR pipe at the remote RTP.
- 2: The allowed send rate on this node is 798 Kbps, compared to the receive rate (3), which relatively low.

The initial data flow rate is exchanged during Route Setup Flows that precede the activation of a new HPR pipe or a switch to a new path. Every TG along the path is inspected with regards to the defined capacity. The slowest speed along the path dominates the perceived end-to-end capacity. The initial send rate will then be set to 5% of that slowest speed along the path. Example 12-12 shows the relationship between initial data flow rate and defined capacity.

Example 12-12 Display of HPR pipe and APPN Topology

```
D NET,ID=CNR0000C
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR0000C, TYPE = PU_T2.1 693
IST486I STATUS= ACTIV---X-, DESIRED STATE= ACTIV
IST1043I CP NAME = SC31M - CP NETID = RDBOOKEE - DYNAMIC LU = YES
IST1589I XNETALS = YES
IST2178I RPNCB ADDRESS 11FC5800
IST1963I APPNCOS = #INTER - PRIORITY = HIGH
IST1476I TCID X'1BE4112F00010053' - REMOTE TCID X'1BED862A0001005D'
IST1481I DESTINATION CP RDBOOKEE.SC31M - NCE X'D000000000000000'
IST1587I ORIGIN NCE X'D000000000000000'
IST1967I ACTIVATED AS PASSIVE ON 10/26/06 AT 07:33:15
IST1477I ALLOWED DATA FLOW RATE = 94 MBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 47 MBITS/SEC 1
IST1841I ACTUAL DATA FLOW RATE = 0 BITS/SEC
IST1697I RTP PACING ALGORITHM = ARB RESPONSIVE MODE
IST1480I RTP END TO END ROUTE - RSCV PATH
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   6  RDBOOKEE.SC31M    APPN         RTP 2
IST875I ALSNAME TOWARDS RTP = EEPUNN31

D NET,TOPO,ORIG=SC32M,DEST=SC31M,TGN=6 3
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TOPOLOGY 696
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP RDBOOKEE.SC32M
IST1357I                                                CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.SC31M    6        OPER     ENDPT       YES   *NA*
IST1579I                   -----------------------------------------
IST1163I                   RSN               HPR         TIME LEFT
IST1164I                   12                YES         15
IST1579I                   -----------------------------------------
IST1302I                   CAPACITY PDELAY   COSTTIME    COSTBYTE
IST1303I                  4 944M     NEGLIGIB 0           0
IST1579I                   -----------------------------------------
IST1304I                   SECURITY UPARM1   UPARM2      UPARM3
IST1305I                   SHIELDED 20       40          40
IST1579I                   -----------------------------------------
```

```
IST1736I                   PU NAME
IST1737I                   EEPUNN31
IST314I END

EEPUNN31 PU    CPNAME=SC31M,NETID=RDBOOKEE,TGN=06,                         *
               TGP=HIPERSOC,DISCNT=NO, 5 *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,                             *
               DWACT=YES,DWINOP=YES,                                       *
               ISTATUS=ACTIVE
```

Note the following explanation Example 12-12:

- 1: The initial data flow rate for this pipe is 47 Mbps.
- 2: The RTP path indicates that it is a single hop path over TG 6 towards SC31M.
- 3: A D NET,TOPO command shows the TG characteristics, the capacity is 944 Mbps.
- 4: This capacity (944M) is defined in a transmission group profile HIPERSOC, which the PU definition refers to in the SWNET major node (5).

Note: For single hop HPR pipes, the initial data flow rate is 5% of the TG's defined capacity in the APPN topology. TG characteristics in the topology are unidirectional and so are the initial data flow rates of the HPR pipes.

Example 12-12 shows an HPR pipe between two VTAMs.

Example 12-13 shows the TG characteristics of an EE link from a Personal Communications towards VTAM.

Example 12-13 Topology display showing a link from PCOMM to VTAM

```
D NET,TOPO,ORIG=EECPBURK,DEST=SC31M
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TOPOLOGY 116
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP RDBOOKEE.EECPBURK
IST1357I                                                 CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.SC31M    6        OPER     ENDPT       YES   *NA*
IST1579I                   ------------------------------------------
IST1163I                   RSN               HPR         TIME LEFT
IST1164I                   0                 YES         15
IST1579I                   ------------------------------------------
IST1302I                   CAPACITY PDELAY   COSTTIME    COSTBYTE
IST1303I                   16M      NEGLIGIB 0           0
IST1579I                   ------------------------------------------
IST1304I                   SECURITY UPARM1   UPARM2      UPARM3
IST1305I                   UNSECURE 0        0           0
```

Note in Example 12-13 the capacity is 16 Mbps. This is the default setting in PCOMM.

Example 12-14 shows the same TG in the opposite direction, VTAM towards PCOMM.

Example 12-14 Topology display showing a link from VTAM to PCOMM

```
D NET,TOPO,DEST=EECPBURK,ORIG=SC31M
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = TOPOLOGY 122
```

```
IST1299I TRANSMISSION GROUPS ORIGINATING AT CP RDBOOKEE.SC31M
IST1357I                                                  CPCP
IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
IST1301I RDBOOKEE.EECPBURK 6        OPER     ENDPT       YES   *NA*
IST1579I                   ------------------------------------------
IST1163I                   RSN               HPR         TIME LEFT
IST1164I                   2                 YES         15
IST1579I                   ------------------------------------------
IST1302I                   CAPACITY PDELAY   COSTTIME    COSTBYTE
IST1303I                   944M     NEGLIGIB 0           0
IST1579I                   ------------------------------------------
IST1304I                   SECURITY UPARM1   UPARM2      UPARM3
IST1305I                   UNSECURE 20       20          20
IST1579I                   ------------------------------------------
IST1736I                   PU NAME
IST1737I                   EEPUBURK

EEPUBURK PU    CPNAME=EECPBURK,NETID=RDBOOKEE,TGN=06,                  *
               CPCP=YES,HPR=YES,DISCNT=NO,DWACT=NO,DWINOP=NO,          *
               TGP=GIGENET
```

Note in Example 12-14 that the capacity in this direction is 944 Mbps. This value is taken from the TGP=GIGENET definition in the SWNET Major Node defining the PU EEPUBURK.

> **Important:** We recommend having the same or similar capacity values for APPN TGs in both directions so that the initial data flow rate at both RTP endpoints is the same.

12.2.1 Additional TG Profiles

In order to achieve the goal of having the same capacity values in both directions we need to decide which value to chose and how to assign that value. For VTAM to VTAM connections this can be done in the SWNET major nodes for predefined PUs or on a model definition that is valid for all dynamic PUs. In distributed SNA stacks, those definitions are hidden in some advanced definition panels and are usually kept to the defaults. Some implementations can not assign specific capacities at all but only select between a limited number of TGPs. The most viable solution to this problem is to create new TGPs specifying the default values of the various platforms and assign that TGP to the PU representing a link to that platform.

Table 12-4 lists the default values of the TG capacity in the distributed platforms along with existing TGPs in VTAM’s default IBMTGPS table.

Table 12-4 Distributed SNA stacks and their default capacities

Platform	Link type	Capacity	TGP
CS/ AIX - CS/Linux	predefined	158M	
CS/ AIX - CS/Linux	dynamic	4M	TOKNRING
PComm - CS/Win	predefined/VRN	16M	TRING16M
System i		5M	
SNASw		16M	TRING16M
MS HIS	predefined	100M	

Here is an explanatory note to Table 12-4:

- There are several different default values available in the various implementations. AIX and Linux even have different capacity defaults for predefined and dynamic links.

To make it easier to match the TG characteristics in the distributed platforms with the VTAM definitions of the TGs, we created new TG Profiles in EETGPS.

Example 12-15 lists the new transmission group profiles.

Example 12-15 New TGPs for distributed servers

```
**********************************************************************
* TGP to match the default characteristics of CS/AIX and CS/Linux    *
**********************************************************************
TGPCS/AIX TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=158M,                          *
               UPARM1=128,UPARM2=128,UPARM3=128
*
**********************************************************************
* TGP to match the default characteristics of System i V5R4           *
**********************************************************************
TGPAS400 TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=TERRESTR,CAPACITY=5M,                            *
               UPARM1=128,UPARM2=128,UPARM3=128
*
**********************************************************************
* TGP to match the default characteristics of SNASw                  *
**********************************************************************
TGPSNASW TGP   COSTTIME=196,COSTBYTE=196,SECURITY=UNSECURE,            *
               PDELAY=NEGLIGIB,CAPACITY=16M,                           *
               UPARM1=128,UPARM2=128,UPARM3=128
*
**********************************************************************
* TGP to match the default characteristics of HIS Server             *
**********************************************************************
TGPMSHIS TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=100M,                          *
               UPARM1=128,UPARM2=128,UPARM3=128
*
```

Note that in Example 12-15 we also added TGPSNASW because of the COSTTIME and COSTBYTE defaults of 196, which do not match the TRING16M TGP that we considered in Table 12-4 on page 389 by looking at the capacity only.

12.2.2 Applying the new TGPs to the PU definitions

Now that we have new TG profiles we can apply them to the predefined SWNET PUs.

Example 12-16 shows the SWNET Major Node that defines the connection to a System i.

Example 12-16 Switched Major Node for System i with new TGP

```
EESWA400 VBUILD TYPE=SWNET
EEPUA400 PU    CPNAME=EECPA400,NETID=RDBOOKEE,                          *
               CPCP=YES,HPR=YES,DISCNT=NO,DWACT=NO,DWINOP=NO,           *
               TGP=TGPAS400 1
EEPTA400 PATH  IPADDR=10.10.2.242,                                      *
               SAPADDR=4,                                               *
               GRPNM=EEGVG&SYSCLONE.1
```

Note that in Example 12-16, the Switched Major Node for the System i points to a special TGP that defines a CAPACITY of 5M.

12.3 RTPONLY

RTPONLY can be coded on an ADJCP definition to ensure no HPR pipes will pass through an EBN node. That is, there will be no ANR routing through the EBN. Therefore, the EBN is aware of all sessions between the two networks. A diagram of the EBN topology is shown in Figure 12-1. The secondary LU is a Telnet APPL on RDBOOKEE.SC32M, an End Node. The PLU is TSO on USIBMSC.SC42M, also an End Node.

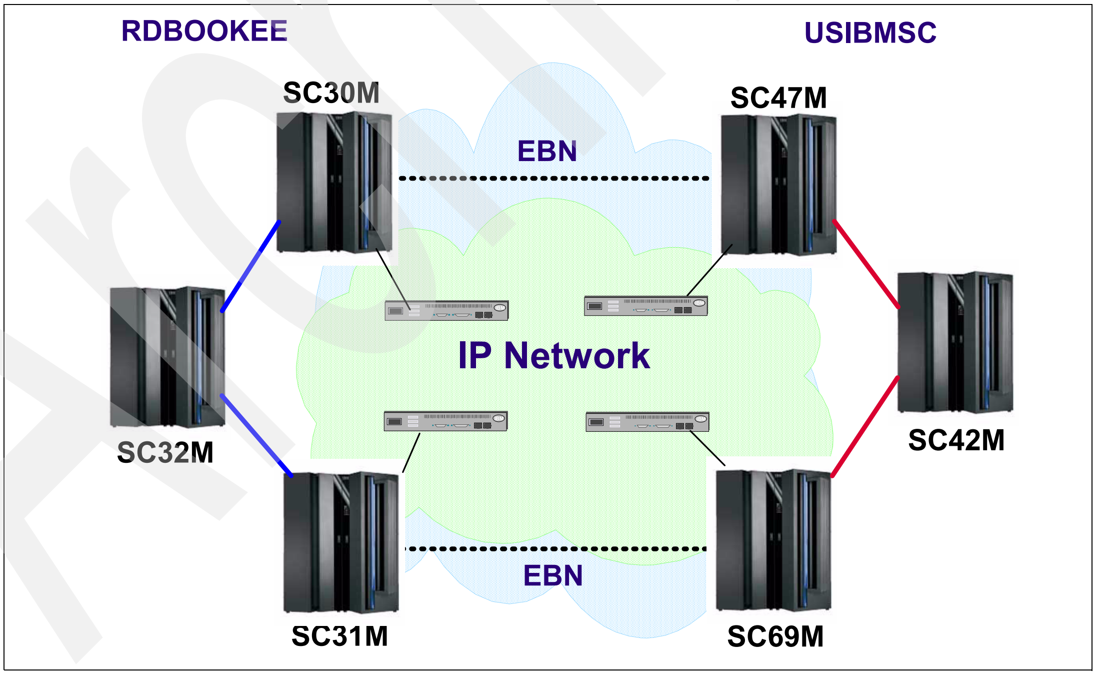


Figure 12-1 EBN configuration

RTPONLY = NO

To begin, the EBNs in RDBOOKEE were configured to allow ANR routing to network USIBMSC. The ADJCP table for SC30M and SC31M is shown in Figure 12-2.

```
**********************************************************************
*  ADJCP Major NODE TO REJECT INCOMING ALIAS SEARCHES SG24-7359      *
*  ADJACENT EBN SHOULD RESOLVE ALIAS TO REAL LUS                    *
*  BEFORE SERCHING OTHER NETIDS                                     *
**********************************************************************
EEADJCPS VBUILD TYPE=ADJCP
*
SC47M    ADJCP NETID=USIBMSC,ALIASRCH=NO,RTPONLY=NO                   1
SC69M    ADJCP NETID=USIBMSC,ALIASRCH=NO,RTPONLY=NO                   1
```

Figure 12-2   ADJCP table, RTPONLY=NO

We can confirm the value for RTPONLY by issuing the command:

D NET,ADJCP,ID=USIBMSC.SC47M,SCOPE=ALL

This is shown in Figure 12-3.

```
D NET,ADJCP,ID=USIBMSC.SC47M,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CONTROL POINT 984
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1197I ADJCP MAJOR NODE = EEADJCP
IST1101I ADJCP DISPLAY SUMMARY FOR USIBMSC.SC47M
IST1102I NODENAME          NODETYPE CONNECTIONS CP CONNECTIONS NATIVE
IST1103I USIBMSC.SC47M      NN       1          1            NO
IST2157I ALIASRCH = NO
IST1104I CONNECTION SUMMARY FOR USIBMSC.SC47M
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
IST1106I EEP73047 AC/R     7 YES   985500000000000000000A09100808080
IST1106I EEP63047 IN/R     6 NO    106D0000000000640000019100 5A5A5A
IST1500I STATE TRACE = OFF
IST1493I RTP SUMMARY FOR USIBMSC.SC47M COUNT = 3 RTPONLY = NO        1
IST1486I   RTP NAME   STATE              DESTINATION CP     MNPS  TYPE
IST1487I   CNR0007C   CONNECTED          USIBMSC.SC47M      NO    RSTP
IST1487I   CNR00073   CONNECTED          USIBMSC.SC47M      NO    CPCP
IST1487I   CNR00072   CONNECTED          USIBMSC.SC47M      NO    CPCP
IST314I END
```

Figure 12-3   SC47M is set to RTPONLY=NO

In Figure 12-3, the message IST1493I shows that RTPONLY is set to NO (1).

We then displayed the LU, TN32L211, on SC32M. This is shown in Figure 12-4.

```
D NET,E,ID=TN32L211                                                  2
IST097I DISPLAY ACCEPTED
IST075I NAME = RDBOOKEE.TN32L211, TYPE = DYNAMIC APPL 975            2 ▮
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = CDSERVR
IST1629I MODSRCH = NEVER
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST861I MODETAB=ALLMODES USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
IST1632I VPACING =  7
IST1938I APPC = NO
IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT 00000001
IST231I APPL MAJOR NODE = TN3270
IST1425I DEFINED USING MODEL TN32L*
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST271I JOBNAME = TN3270, STEPNAME = TN3270, DSPNAME = ISTE9F5B
IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
IST1633I ASRCVLM = 1000000
IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 0
IST1669I IPADDR..PORT 9.12.12.227..1547
IST171I ACTIVE SESSIONS = 0000000001, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I SC42TS04 ACTIV-P    ED1315CA6BE52981 000A 001A       USIBMSC  3
IST314I END
```

Figure 12-4 Displaying the secondary LU

The key information from the display in Figure 12-44 is as follows:

- 2:Confirm the LU Name
- 3:The application name and session ID (SID)

We use the SID (3) from the LU display to determine which RTP pipe is being used. Display the session using the SID. The display is shown in Figure 12-5.

```
D NET,SESSIONS,SID=ED1315CA6BE52981                                      3
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SESSIONS 978
IST879I PLU/OLU REAL = USIBMSC.SC42TS04  ALIAS = RDBOOKEE.SC42TS04
IST879I SLU/DLU REAL = RDBOOKEE.TN32L211 ALIAS = USIBMSC.TN32L211
IST880I SETUP STATUS = ACTIV
IST875I ADJSSCP TOWARDS PLU = ISTAPNCP
IST875I ALSNAME TOWARDS PLU = CNR00084                                   4
IST933I LOGMODE=SNX32703, COS=*BLANK*
IST875I APPNCOS TOWARDS PLU = #CONNECT
IST1635I PLU HSCB TYPE: BSB LOCATED AT ADDRESS X'13838450'
IST1635I SLU HSCB TYPE: FMCB LOCATED AT ADDRESS X'0E7F0E48'
IST2064I PLU TO SLU RU SIZE =  3840    SLU TO PLU RU SIZE =  1024
IST1636I PACING STAGE(S) AND VALUES:
IST1644I PLU--STAGE 1-----|-----STAGE 2--SLU
IST1638I STAGE1: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
IST1640I         SECONDARY RECEIVE       =     7
IST1641I STAGE1: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
IST1642I         SECONDARY SEND: CURRENT =   119     NEXT =   128
IST1638I STAGE2: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
IST1639I         PRIMARY SEND: CURRENT   =     3     NEXT =     7
IST1640I         SECONDARY RECEIVE       =     7
IST1641I STAGE2: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
IST1642I         SECONDARY SEND: CURRENT =     5     NEXT =     7
IST1643I         PRIMARY RECEIVE         =     7
IST1710I RSCV FROM PLU SAVED AT SESSION ACTIVATION
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I  21  W3IBMCOM.VRN      APPN         RTP
IST1461I  21  RDBOOKEE.SC32M    APPN         RTP
IST1713I RTP RSCV IN THE DIRECTION OF THE PLU                            5
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC30M    APPN         RTP
IST1461I  21  W3IBMCOM.VRN      APPN         RTP
IST1461I  21  USIBMSC.SC42M     APPN         RTP
IST314I END
```

Figure 12-5 Displaying the session ID

In Figure 12-5 the D NET,SESSIONS,SID=`ED1315CA6BE52981` command (**3**) provides details about the session. Message IST875I gives the name of the Adjacent Link Station (ALS) in the PLU direction (**4**), CNR00084. This is sometimes also referred to as the RTP PU. Message IST1713I (**5**) heads a message group that provides information about the HPR Pipe. The IST1461I messages show the nodes that the HPR pipe traverses. W3IBMCOM.VRN is a global connection network. Our End Node, SC32M, is not part of that connection network. SC30M is the EBN between networks. So the path is SC32M to the EBN SC30M and then via connection network to SC42M in the other network.

But that did not tell us what type of HPR function was provided in each node. To get more information we can display the HPR ALS (or RTP PU). This is shown in Figure 12-6.

```
D NET,E,ID=CNR00084                                                     4
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR00084, TYPE = PU_T2.1 981
IST1392I DISCNTIM = 00010 DEFINED AT PU FOR DISCONNECT
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = SC42M - CP NETID = USIBMSC - DYNAMIC LU = YES        6
IST1589I XNETALS = YES
IST2178I RPNCB ADDRESS 11F74018
IST1964I APPNCOS = #CONNECT - PRIORITY = MEDIUM
IST1476I TCID X'122C37E000010062' - REMOTE TCID X'122C9CA400010056'
IST1481I DESTINATION CP USIBMSC.SC42M - X'D000000000000000'             7
IST1587I ORIGIN NCE X'D000000000000000'
IST1967I ACTIVATED AS PASSIVE ON 10/24/06 AT 10:02:57
IST1477I ALLOWED DATA FLOW RATE = 400 KBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 200 KBITS/SEC
IST1841I ACTUAL DATA FLOW RATE = 0 BITS/SEC
IST1511I MAXIMUM NETWORK LAYER PACKET SIZE = 1469 BYTES
IST1478I NUMBER OF UNACKNOWLEDGED BUFFERS = 0
IST1479I RTP CONNECTION STATE = CONNECTED - MNPS = NO
IST1959I DATA FLOW STATE = NORMAL
IST1855I NUMBER OF SESSIONS USING RTP = 2
IST1697I RTP PACING ALGORITHM = ARB RESPONSIVE MODE
IST1480I RTP END TO END ROUTE - RSCV PATH                               9
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC30M     APPN         RTP
IST1461I  21  W3IBMCOM.VRN       APPN         RTP
IST1461I  21  USIBMSC.SC42M      APPN         RTP
IST875I ALSNAME TOWARDS RTP = MPC2SC30
IST1738I ANR LABEL              TP           ER NUMBER
IST1739I 8042000C00000000       *NA*         *NA*                       8
IST1739I 8027008A01000000       *NA*         *NA*                       8
IST231I RTP MAJOR NODE = ISTRTPMN
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST355I LOGICAL UNITS:
IST080I SC42TS04 ACT/S      SC42TS02 ACT/S
IST314I END
```

Figure 12-6 Display HPR ALS

Displaying the HPR ALS, or RTP PU, 4 provides information about the HPR pipe. Two messages, IST1043I (6) and IST1481I(7), show that the other end of the HPR pipe is USIBMSC.SC42M. This means the RTP endpoints are EN SC32M in our network and EN SSC42M in the partner network. All other nodes in the HPR path provide only ANR routing.

The message group after IST1480I (9) shows the HPR route end-to-end. Messages IST1460I and IST1461I again show the nodes the HPR pipe traverses. Note the entry for W3IBMCOM.VRN is the global connection network. The pipe logically goes through this node but there is no ANR label for it. There is no ANR label because it is a virtual node and will not actually take part in the path. The physical connection is from SC30M to SC42M using EE.

The two IST1739I messages(8), show the ANR routing tags used. The ANR routing labels apply to the first two nodes in the path, SC32M and SC30M. The labels show which ports will be used to exit the HPR node. The NCE label (7) is used by the node the other end of the pipe showing that SC42M is the RTP end of the pipe.

sentence is not clear

Now switch over to the EBN, RDBOOKEE.SC30M, and display the LU, TN32L211, and HPR pipe, CNR00084 to see what that system knows about them. These displays are in Figure 12-7.

```
D NET,E,ID=TN32L211
IST097I DISPLAY ACCEPTED
IST075I NAME = RDBOOKEE.TN32L211, TYPE = DIRECTORY ENTRY 069
IST1186I DIRECTORY ENTRY = REGISTERED LU
IST1184I CPNAME = RDBOOKEE.SC32M - NETSRVR = RDBOOKEE.SC30M
IST484I SUBAREA = ****NA****
IST1703I DESIRED LOCATE SIZE = 1K LAST LOCATE SIZE = 16K
IST314I END


D NET,E,ID=CNR00084
IST453I ID PARAMETER VALUE RDBOOKEE.CNR00084 NOT VALID
```

Figure 12-7 Display LU and HPR ALS on EBN SC30M

We see from these displays that:

1. SC30M only has directory information about the LU, TN32L211. It has no session awareness of the LU's session with SC42TS04.
2. SC30M has no awareness at all of the HPR pipe CNR00084. This is as it should be as SC30M performs ANR routing for this HPR pipe.

RTPONLY = YES

Now we changed the RTPONLY parameter to YES (see Figure 12-8). We now expect all HPR pipes to terminate on the RDBOOKEE EBNs.

```
*********************************************************************
*  ADJCP Major NODE TO REJECT INCOMING ALIAS SEARCHES SG24-7359      *
*  ADJACENT EBN SHOULD RESOLVE ALIAS TO REAL LUS                    *
*  BEFORE SERCHING OTHER NETIDS                                     *
*********************************************************************
EEADJCPS VBUILD TYPE=ADJCP
*
SC47M    ADJCP NETID=USIBMSC,ALIASRCH=NO,RTPONLY=YES                 1
SC69M    ADJCP NETID=USIBMSC,ALIASRCH=NO,RTPONLY=YES                 1
```

Figure 12-8 ADJCP table, RTPONLY=YES

We can see that RTPONLY is set to YES (1) in Figure 12-9 and confirm it with a D NET,ADJCP command in Figure 12-9.

```
D NET,ADJCP,ID=USIBMSC.SC47M,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = ADJACENT CONTROL POINT 134
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1197I ADJCP MAJOR NODE = EEADJCP
IST1101I ADJCP DISPLAY SUMMARY FOR USIBMSC.SC47M
IST1102I NODENAME          NODETYPE CONNECTIONS CP CONNECTIONS NATIVE
IST1103I USIBMSC.SC47M      NN        1           1            NO
IST2157I ALIASRCH = NO
IST1104I CONNECTION SUMMARY FOR USIBMSC.SC47M
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
IST1106I EEP73047 AC/R     7 YES   98550000000000000000A09100808080
IST1106I EEP63047 IN/R     6 NO    106D000000000064000001191005A5A5A
IST1500I STATE TRACE = OFF
IST1493I RTP SUMMARY FOR USIBMSC.SC47M COUNT = 3 RTPONLY = YES        1
IST1486I   RTP NAME   STATE             DESTINATION CP     MNPS  TYPE
IST1487I   CNR0007C   CONNECTED         USIBMSC.SC47M      NO    RSTP
IST1487I   CNR00073   CONNECTED         USIBMSC.SC47M      NO    CPCP
IST1487I   CNR00072   CONNECTED         USIBMSC.SC47M      NO    CPCP
IST314I END
```

Figure 12-9 Displaying the Adjacent CP for USIBMSC

We logged onto SC42TS from our TN3270 APPL, TN32L211, and were passed to SC42TS03.

A display of LU TN32L211(2) on RDBOOKEE.SC32M is shown in Figure 12-10.

```
 D NET,E,ID=TN32L211                                                 2
 IST097I DISPLAY ACCEPTED
 IST075I NAME = RDBOOKEE.TN32L211, TYPE = DYNAMIC APPL 031
 IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
 IST1447I REGISTRATION TYPE = CDSERVR
 IST1629I MODSRCH = NEVER
 IST977I MDLTAB=***NA*** ASLTAB=***NA***
 IST861I MODETAB=ALLMODES USSTAB=***NA*** LOGTAB=***NA***
 IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
 IST1632I VPACING =  7
 IST1938I APPC = NO
 IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT 00000001
 IST231I APPL MAJOR NODE = TN3270
 IST1425I DEFINED USING MODEL TN32L*
 IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
 IST1500I STATE TRACE = OFF
 IST271I JOBNAME = TN3270, STEPNAME = TN3270, DSPNAME = ISTE9F5B
 IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
 IST1633I ASRCVLM = 1000000
 IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 0
 IST1669I IPADDR..PORT 9.12.12.227..1547
 IST171I ACTIVE SESSIONS = 0000000001, SESSION REQUESTS = 0000000000
 IST206I SESSIONS:
 IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
 IST635I SC42TS03 ACTIV-P    ED1315CA6BE52996 0007 0016       USIBMSC  3
IST314I END
```

Figure 12-10 Display TN32L211 on SC32M

From the LU display we now find the SID (3) and use that to display the session details in Figure 12-11.

```
D NET,SESSIONS,SID=ED1315CA6BE52996                                   3
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SESSIONS 034
IST879I PLU/OLU REAL = USIBMSC.SC42TS03  ALIAS = RDBOOKEE.SC42TS03
IST879I SLU/DLU REAL = RDBOOKEE.TN32L211 ALIAS = USIBMSC.TN32L211
IST880I SETUP STATUS = ACTIV
IST875I ADJSSCP TOWARDS PLU = ISTAPNCP
IST875I ALSNAME TOWARDS PLU = CNR00095                                4
IST933I LOGMODE=SNX32703, COS=*BLANK*
IST875I APPNCOS TOWARDS PLU = #CONNECT
IST1635I PLU HSCB TYPE: BSB LOCATED AT ADDRESS X'13838890'
IST1635I SLU HSCB TYPE: FMCB LOCATED AT ADDRESS X'0E7F0E48'
IST2064I PLU TO SLU RU SIZE =  3840    SLU TO PLU RU SIZE =  1024
IST1636I PACING STAGE(S) AND VALUES:
IST1644I PLU--STAGE 1-----|-----STAGE 2--SLU
IST1638I STAGE1: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
IST1640I         SECONDARY RECEIVE       =     7
IST1641I STAGE1: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
IST1642I         SECONDARY SEND: CURRENT =   122     NEXT =   128
IST1638I STAGE2: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
IST1639I         PRIMARY SEND: CURRENT   =     0     NEXT =     7
IST1640I         SECONDARY RECEIVE       =     7
IST1641I STAGE2: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
IST1642I         SECONDARY SEND: CURRENT =     1     NEXT =     7
IST1643I         PRIMARY RECEIVE         =     7
IST1710I RSCV FROM PLU SAVED AT SESSION ACTIVATION
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I  21  USIBMSC.SC47M     APPN         RTP
IST1461I   7  RDBOOKEE.SC30M    ISL          RTP
IST1461I   2  RDBOOKEE.SC32M    APPN         RTP
IST1713I RTP RSCV IN THE DIRECTION OF THE PLU                         5
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC30M    APPN         RTP
IST314I END
```

Figure 12-11 Display Session ID on SC32M

Displaying the SID gives us two pieces of information:

- The adjacent link station on the HPR pipe is CNR00095 (4)
- There is only one node in the HPR pipe in the direction of the PLU (5).

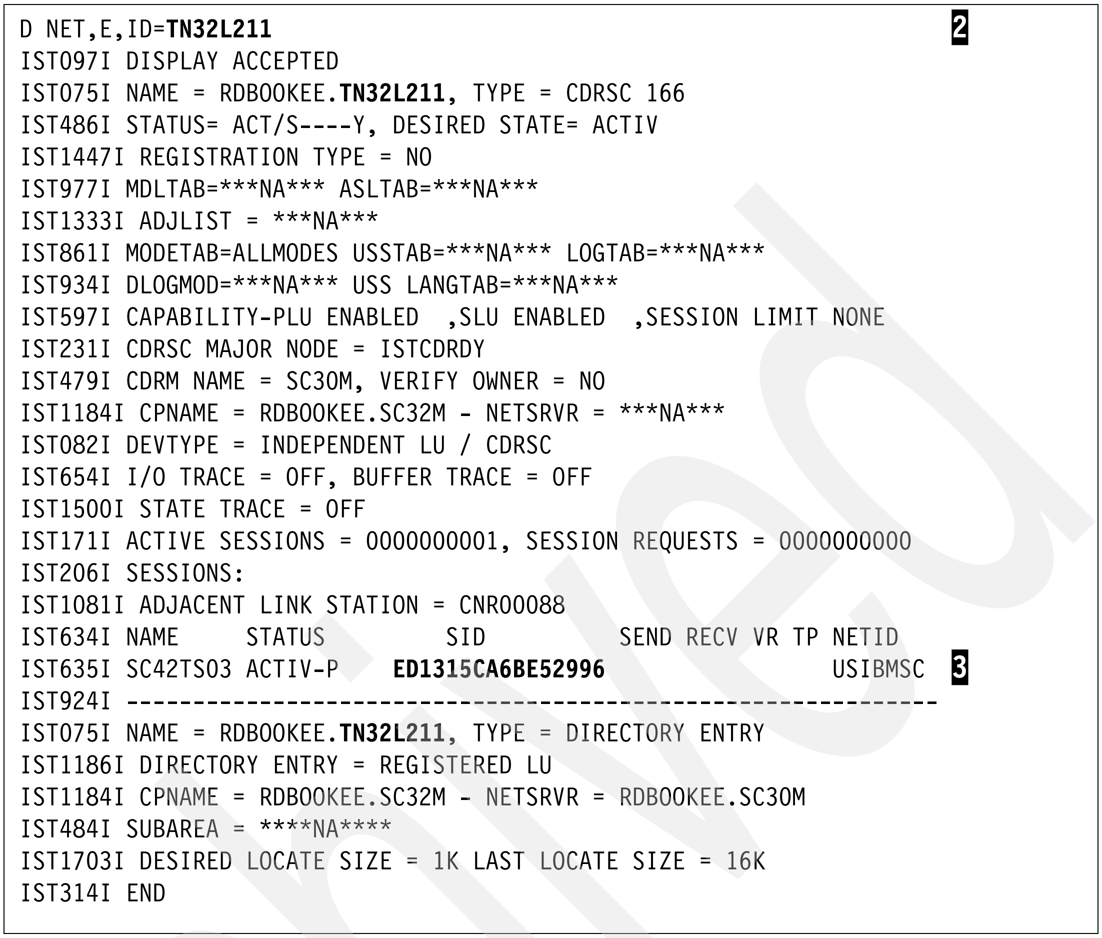
This is quite different from what we had when RTPONLY=NO was coded. A display of the HPR pipe, Figure 12-12, might show what is different.

```
D NET,E,ID=CNR00095                                                   4
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR00095, TYPE = PU_T2.1 037
IST1392I DISCNTIM = 00010 DEFINED AT PU FOR DISCONNECT
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = SC30M - CP NETID = RDBOOKEE - DYNAMIC LU = YES     6
IST1589I XNETALS = YES
IST2178I RPNCB ADDRESS 0E7A4018
IST1964I APPNCOS = #CONNECT - PRIORITY = MEDIUM
IST1476I TCID X'122C37EF00010016' - REMOTE TCID X'1587F7610001002B'
IST1481I DESTINATION CP RDBOOKEE.SC30M - NCE X'D000000000000000'      7
IST1587I ORIGIN NCE X'D000000000000000'
IST1967I ACTIVATED AS PASSIVE ON 10/24/06 AT 16:08:21
IST1477I ALLOWED DATA FLOW RATE = 59 MBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 29 MBITS/SEC
IST1841I ACTUAL DATA FLOW RATE = 1000 BITS/SEC
IST1511I MAXIMUM NETWORK LAYER PACKET SIZE = 61466 BYTES
IST1478I NUMBER OF UNACKNOWLEDGED BUFFERS = 0
IST1479I RTP CONNECTION STATE = CONNECTED - MNPS = NO
IST1959I DATA FLOW STATE = NORMAL
IST1855I NUMBER OF SESSIONS USING RTP = 1
IST1697I RTP PACING ALGORITHM = ARB RESPONSIVE MODE
IST1480I RTP END TO END ROUTE - RSCV PATH
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC30M     APPN         RTP
IST875I ALSNAME TOWARDS RTP = MPC2SC30
IST1738I ANR LABEL               TP           ER NUMBER
IST1739I 8042000C00000000        *NA*         *NA*                    8
IST231I RTP MAJOR NODE = ISTRTPMN
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST355I LOGICAL UNITS:
IST080I SC42TS03 ACT/S
IST314I END
```

Figure 12-12 Displaying HPR pipe on SC32M

We can see that the Adjacent LS (7) name is now RDBOOKEE.SC30M, which is our EBN. The path only contains one ANR label (8) that gets us out of our own node, SC32M and the NCE for SC30M terminates the HPR pipe.

To see how the session gets to SC42M we have to look on our EBN, SC30M. A display of our LU, TN32L211, on our EBN, SC30M is shown in Figure 12-13.



```
D NET,E,ID=TN32L211                                                        2
IST097I DISPLAY ACCEPTED
IST075I NAME = RDBOOKEE.TN32L211, TYPE = CDRSC 166
IST486I STATUS= ACT/S----Y, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = NO
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST1333I ADJLIST = ***NA***
IST861I MODETAB=ALLMODES USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
IST231I CDRSC MAJOR NODE = ISTCDRDY
IST479I CDRM NAME = SC30M, VERIFY OWNER = NO
IST1184I CPNAME = RDBOOKEE.SC32M - NETSRVR = ***NA***
IST082I DEVTYPE = INDEPENDENT LU / CDRSC
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000001, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST1081I ADJACENT LINK STATION = CNR00088
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I SC42TS03 ACTIV-P    ED1315CA6BE52996                 USIBMSC  3
IST924I -------------------------------------------------------------
IST075I NAME = RDBOOKEE.TN32L211, TYPE = DIRECTORY ENTRY
IST1186I DIRECTORY ENTRY = REGISTERED LU
IST1184I CPNAME = RDBOOKEE.SC32M - NETSRVR = RDBOOKEE.SC30M
IST484I SUBAREA = ****NA****
IST1703I DESIRED LOCATE SIZE = 1K LAST LOCATE SIZE = 16K
IST314I END
```

Figure 12-13 TN32L211 on EBN SC30M

With RTPONLY=YES we now have session awareness of our LU TN32L211 session from the EBN. In Figure 12-7 on page 396 we had no awareness on SC30M of any sessions to do with TN32L211. Since we now have session awareness, we also have a SID (3). This is shown in Figure 12-14.

```
D NET,SESSIONS,SID=ED1315CA6BE52996                                        3
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SESSIONS 230
IST879I PLU/OLU REAL = USIBMSC.SC42TS03  ALIAS = RDBOOKEE.SC42TS03
IST879I SLU/DLU REAL = RDBOOKEE.TN32L211 ALIAS = USIBMSC.TN32L211
IST880I SETUP STATUS = ACTIV
IST875I ADJSSCP TOWARDS PLU = ISTAPNCP
IST875I ADJSSCP TOWARDS SLU = ISTAPNCP
IST875I ALSNAME TOWARDS PLU = CNR00087                                     4
IST875I ALSNAME TOWARDS SLU = CNR00088                                     4
IST933I LOGMODE=SNX32703, COS=*BLANK*
IST875I APPNCOS TOWARDS PLU = #CONNECT
IST875I APPNCOS TOWARDS SLU = #CONNECT
IST1635I PLU HSCB TYPE: BSB LOCATED AT ADDRESS X'1386F120'
IST1635I SLU HSCB TYPE: BSB LOCATED AT ADDRESS X'1386FCD0'
IST1636I PACING STAGE(S) AND VALUES:
IST1645I PLU--STAGE 1-----|-----STAGE 2-----|-----STAGE 3--SLU
IST1638I STAGE1: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
IST1640I         SECONDARY RECEIVE       = 32767
IST1641I STAGE1: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
IST1642I         SECONDARY SEND: CURRENT =     1     NEXT =     1
IST1638I STAGE2: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
IST1639I         PRIMARY SEND: CURRENT   =     1     NEXT =     1
IST1640I         SECONDARY RECEIVE       =     7
IST1641I STAGE2: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
IST1642I         SECONDARY SEND: CURRENT =     1     NEXT =     1
IST1643I         PRIMARY RECEIVE         = 32767
IST1638I STAGE3: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
IST1639I         PRIMARY SEND: CURRENT   =     0     NEXT =     7
IST1641I STAGE3: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
IST1643I         PRIMARY RECEIVE         =   128
IST1710I RSCV FROM PLU SAVED AT SESSION ACTIVATION
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I  21  USIBMSC.SC47M      APPN         RTP
IST1461I   7  RDBOOKEE.SC30M     ISL          RTP
IST1461I   2  RDBOOKEE.SC32M     APPN         RTP
IST1713I RTP RSCV IN THE DIRECTION OF THE PLU                              5
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   7  USIBMSC.SC47M      ISL          RTP
IST1461I  21  USIBMSC.SC42M      APPN         RTP
IST1713I RTP RSCV IN THE DIRECTION OF THE SLU                              5
IST1460I TGN  CPNAME             TG TYPE      HPR
IST1461I   2  RDBOOKEE.SC32M     APPN         RTP
IST314I END
```

Figure 12-14 Session ID on SC30M

This is quite different. We now have two adjacent RTP link station names (4). One is in the PLU direction and the other in the SLU direction. Also note that the RTP path (5) towards the PLU no longer uses the connection network W3IBMCOM.VRN. Instead it goes to USIBMSC.SC47M, which is the EBN for the partner network. When RTPONLY=YES is coded, global connection networks will not be used.

A display of the RTP PU in the PLU direction is shown in Figure 12-15.

```
D NET,E,ID=CNR00087                                                    4
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR00087, TYPE = PU_T2.1 243
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = SC42M - CP NETID = USIBMSC - DYNAMIC LU = YES      6
IST1589I XNETALS = YES
IST2178I RPNCB ADDRESS 0EB25018
IST1964I APPNCOS = #CONNECT - PRIORITY = MEDIUM
IST1476I TCID X'1587F7600001002A' - REMOTE TCID X'122C9CBB00010052'
IST1481I DESTINATION CP USIBMSC.SC42M - NCE X'D000000000000000'        7
IST1587I ORIGIN NCE X'D000000000000000'
IST1967I ACTIVATED AS PASSIVE ON 10/24/06 AT 16:08:21
IST1477I ALLOWED DATA FLOW RATE = 30 KBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 12 KBITS/SEC
IST1841I ACTUAL DATA FLOW RATE = 1000 BITS/SEC
IST1511I MAXIMUM NETWORK LAYER PACKET SIZE = 1469 BYTES
IST1478I NUMBER OF UNACKNOWLEDGED BUFFERS = 0
IST1479I RTP CONNECTION STATE = CONNECTED - MNPS = NO
IST1959I DATA FLOW STATE = NORMAL
IST1855I NUMBER OF SESSIONS USING RTP = 1
IST1697I RTP PACING ALGORITHM = ARB RESPONSIVE MODE
IST1480I RTP END TO END ROUTE - RSCV PATH
IST1460I TGN  CPNAME             TG TYPE      HPR                      9
IST1461I   7  USIBMSC.SC47M      ISL          RTP
IST1461I  21  USIBMSC.SC42M      APPN         RTP
IST875I ALSNAME TOWARDS RTP = EEP73047
IST1738I ANR LABEL               TP           ER NUMBER
IST1739I 801E008C01000000        *NA*         *NA*                     8
IST1739I 8011014D00A00000        *NA*         *NA*                     8
IST231I RTP MAJOR NODE = ISTRTPMN
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST355I LOGICAL UNITS:
IST080I SC42TS03 ACT/S
IST314I END
```

Figure 12-15 HPR pipe #1 on SC30M

We can see that the destination of the HPR pipe is USIBMSC.SC42M. That is a pipe from our EBN to the PLUS's End Node. The `RTP END TO END ROUTE` (9) is from EBM SC30M in our network to the EBN in the adjacent network, SC47M, and then on to the destination End Node, SC42M. From this we can see that USIBMSC has set RTPONLY=NO, otherwise the HPR pipe would have terminated at SC47M.

A display of the RTP PU in the direction of the SLU is shown in Figure 12-16.

```
D NET,E,ID=CNR00088                                                   4
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR00088, TYPE = PU_T2.1 246
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = SC32M - CP NETID = RDBOOKEE - DYNAMIC LU = YES     6
IST1589I XNETALS = YES
IST2178I RPNCB ADDRESS 0EB25800
IST1964I APPNCOS = #CONNECT - PRIORITY = MEDIUM
IST1476I TCID X'1587F7610001002B' - REMOTE TCID X'122C37EF00010016'
IST1481I DESTINATION CP RDBOOKEE.SC32M - NCE X'D000000000000000'     7
IST1587I ORIGIN NCE X'D000000000000000'
IST1966I ACTIVATED AS ACTIVE ON 10/24/06 AT 16:08:21
IST1477I ALLOWED DATA FLOW RATE = 59 MBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 29 MBITS/SEC
IST1841I ACTUAL DATA FLOW RATE = 11 KBITS/SEC
IST1511I MAXIMUM NETWORK LAYER PACKET SIZE = 61466 BYTES
IST1478I NUMBER OF UNACKNOWLEDGED BUFFERS = 0
IST1479I RTP CONNECTION STATE = CONNECTED - MNPS = NO
IST1959I DATA FLOW STATE = NORMAL
IST1855I NUMBER OF SESSIONS USING RTP = 1
IST1697I RTP PACING ALGORITHM = ARB RESPONSIVE MODE
IST1480I RTP END TO END ROUTE - RSCV PATH
IST1460I TGN  CPNAME            TG TYPE      HPR                      9
IST1461I   2  RDBOOKEE.SC32M     APPN         RTP
IST875I ALSNAME TOWARDS RTP = MPC2SC32
IST1738I ANR LABEL              TP           ER NUMBER
IST1739I 8024002900000000       *NA*         *NA*                     8
IST231I RTP MAJOR NODE = ISTRTPMN
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST355I LOGICAL UNITS:
IST080I TN32L211 ACT/S----Y
IST314I END
```

Figure 12-16 HPR pipe #2 on SC30M

The `RTP END TO END ROUTE` (9) from SC30M to SC32M is one hop, as expected.

Summary

When RTPONLY=NO is coded for the adjacent network we see there is one HPR pipe between the SLU and the PLU. If RTPONLY=YES is coded in one network, there are two HPR pipes. One from the SLU to its EBN where RTPONLY=YES is coded and the second from that EBN to the node with the PLU.

If EBNs in both the networks had coded RTPONLY=YES there would have been three RTP pipes from SLU to PLU:

- SLU to EBN
- Cross network EBN to EBN
- EBN to PLU

13

Collecting diagnostic information

This chapter discusses the commands and tools that are available for collecting diagnostic information. The chapter focuses on the following topics:

- "VTAM display commands"
- "Traces in CS for z/OS"
- "F NET,CSDUMP command"
- "Summary"

13.1 VTAM display commands

Virtual Telecommunications Access Method (VTAM) provides a variety of display commands that can be used to diagnose High Performance Routing (HPR) problems. In addition to the general HPR displays, there are also some Enterprise Extender (EE) specific display commands that can prove very helpful in diagnosing EE problems. For example D NET,EE shows statistical information about local or remote IP addresses, PUs, Lines, and so on. The `D` NET,EEDIAG command can be used to verify connectivity through the IP infrastructure by generating User Datagram Protocol (UDP) packets using the EE ports.

This section discusses the following commands:

- “D NET,EE,IPADDR=(local-vipa)”
- “D NET,EE,IPADDR=(,remote-ip)”
- “D NET,EE,ID=puname”
- “D NET,EEDIAG,TEST=YES”

13.1.1 D NET,EE,IPADDR=(local-vipa)

This display shows EE related information based on a local EE VIPA. Example 13-1 shows EE related information for a specified local IP address.

Example 13-1 D NET,EE,IPADDR=(local-vipa)

```
D NET,EE,IPADDR=(10.10.1.231) 1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE 222
IST2002I ENTERPRISE EXTENDER AGGREGATE CONNECTION INFORMATION
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.231
IST1910I LOCAL HOSTNAME SC30M-EE1.ITSO.IBM.COM
IST2009I RTP PIPES =        3      LU-LU SESSIONS      =         1 2
IST2010I INOPS DUE TO SRQRETRY EXPIRATION              =         5 3
IST1324I VNNAME = RDBOOKEE.VRNLOCAL  VNGROUP = EEGVL301  (LOCAL) 4
IST2011I         AVAILABLE LINES FOR THIS EE VRN       =        13
IST2012I         ACTIVE CONNECTIONS USING THIS EE VRN   =         2
IST2013I AVAILABLE LINES FOR PREDEFINED EE CONNECTIONS   =         0
IST2014I ACTIVE PREDEFINED EE CONNECTIONS              =         1
IST2015I ACTIVE LOCAL  VRN EE CONNECTIONS              =         2
IST2016I ACTIVE GLOBAL VRN EE CONNECTIONS              =         0
IST2044I TOTAL ACTIVE EE CONNECTIONS FOR LOCAL IPADDR    =         3
IST924I -------------------------------------------------------------
IST2035I TOTALS FOR ALL PORT PRIORITIES
IST2036I   NLPS SENT          =              32979  ( 032K )
IST2037I   BYTES SENT         =             161955  ( 161K )
IST2038I   NLPS RETRANSMITTED  =                 0  ( 000K )
IST2039I   BYTES RETRANSMITTED =                 0  ( 000K )
IST2040I   NLPS RECEIVED      =              32650  ( 032K )
IST2041I   BYTES RECEIVED     =             116267  ( 116K )
IST314I END
```

Here are explanatory notes to Example 13-1:

- 1: This is the display with the local VIPA provided in the IPADDR parameter.
- 2: A lot of statistical data is shown, for instance how many HPR pipes and how many sessions are using this VIPA.

- 3: There were 5 INOPs because of LDLC time-outs since EE was started.
- 4: This Local IP address participates in a local connection network.

13.1.2 D NET,EE,IPADDR=(,remote-ip)

Statistical information can also be displayed for a given remote EE node. Example 13-2 shows the EE display in a detailed form.

Example 13-2 Display to show EE related information for a given remote IP address

```
D NET,EE,IPADDR=(,10.20.4.203),LIST=DETAIL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE 231
IST2001I ENTERPRISE EXTENDER CONNECTION INFORMATION
IST924I -------------------------------------------------------------
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM
IST1680I REMOTE IP ADDRESS 10.20.4.203
IST2022I EE CONNECTION ACTIVATED ON 10/20/06 AT 21:34:40
IST2114I LIVTIME:    INITIAL =   10   MAXIMUM =   60   CURRENT =   60 1
IST2023I CONNECTED TO LINE EEX3000D
IST2024I CONNECTED TO SWITCHED PU EEPUAIX
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
IST2029I MTU SIZE = 1472 2
IST924I -------------------------------------------------------------
IST2030I PORT PRIORITY = SIGNAL 3
IST2036I   NLPS SENT           =                24144  ( 024K )
IST2037I   BYTES SENT          =                73134  ( 073K )
IST2038I   NLPS RETRANSMITTED  =                    0  ( 000K )
IST2039I   BYTES RETRANSMITTED =                    0  ( 000K )
IST2040I   NLPS RECEIVED       =                24145  ( 024K )
IST2041I   BYTES RECEIVED      =                73092  ( 073K )
IST924I -------------------------------------------------------------
IST2031I PORT PRIORITY = NETWORK
IST2036I   NLPS SENT           =                 1507  ( 001K )
IST2037I   BYTES SENT          =                75723  ( 075K )
IST2038I   NLPS RETRANSMITTED  =                    0  ( 000K )
IST2039I   BYTES RETRANSMITTED =                    0  ( 000K )
IST2040I   NLPS RECEIVED       =                  255  ( 000K )
IST2041I   BYTES RECEIVED      =                46834  ( 046K )
IST924I -------------------------------------------------------------
... repeated for each priority
IST924I -------------------------------------------------------------
IST2035I TOTALS FOR ALL PORT PRIORITIES
IST2036I   NLPS SENT           =                25651  ( 025K )
IST2037I   BYTES SENT          =               148857  ( 148K )
```

```
IST2038I   NLPS RETRANSMITTED  =                    0  ( 000K )
IST2039I   BYTES RETRANSMITTED =                    0  ( 000K )
IST2040I   NLPS RECEIVED       =                24400  ( 024K )
IST2041I   BYTES RECEIVED      =               119926  ( 119K )
IST2042I 1 OF 1 EE CONNECTIONS DISPLAYED
IST314I END
```

Here are explanatory notes to Example 13-2:

- 1: IST2114I shows the defined liveness timer limits as well as the current value.
- 2: The MTU size as known by VTAM is displayed in IST2029I.
- 3: Statistics for every priority is listed.

13.1.3 D NET,EE,ID=puname

This command displays information based on a switched PU name. Example 13-3 shows summary information for a specific PU. It shows the display on the PU name showing the same information as the previous display based on a remote IP address.

Example 13-3 D NET,EE,ID=puname

```
D NET,EE,ID=EEPUPCOM
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EE 619
IST2001I ENTERPRISE EXTENDER CONNECTION INFORMATION
IST075I NAME = EEPUPCOM, TYPE = PU_T2.1
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1910I LOCAL HOSTNAME SC30M-EE2.ITSO.IBM.COM
IST1680I REMOTE IP ADDRESS 10.20.4.201
IST2022I EE CONNECTION ACTIVATED ON 10/23/06 AT 17:59:21
IST2114I LIVTIME:    INITIAL =   10   MAXIMUM =   60   CURRENT =   60
IST2023I CONNECTED TO LINE EEX3000E
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
IST2029I MTU SIZE = 1472
IST924I -------------------------------------------------------------
IST2035I TOTALS FOR ALL PORT PRIORITIES
IST2036I   NLPS SENT           =                  563  ( 000K )
IST2037I   BYTES SENT          =                 6310  ( 006K )
IST2038I   NLPS RETRANSMITTED  =                    0  ( 000K )
IST2039I   BYTES RETRANSMITTED =                    0  ( 000K )
IST2040I   NLPS RECEIVED       =                  581  ( 000K )
IST2041I   BYTES RECEIVED      =                 7520  ( 007K )
IST314I END
```

13.1.4 D NET,EEDIAG,TEST=YES

Starting with z/OS V1R8 there is a command that enables you to test connectivity to a remote EE node. Similar to an **IP traceroute** VTAM sends out UDP packets with incremental initial TTL values. The initial TTL value is decremented by every router on its way until it reaches 0. If this happens, the router cannot forward it and will send an ICMP timeout message back to the source IP address. From the source IP address of these ICMP packets, we can conclude which routers are on the way to the destination. Unlike the **`IP traceroute`** VTAM is sending real UDP packets with the real EE destination port numbers. Therefore, this command can be used to test firewall filter rules for EE traffic because the packets resemble exactly what normal EE UDP packets would look like.

> **Note:** Even though the display command is only available in z/OS V1R8, other releases and platforms provide support to *respond* to LDLC probes. To respond correctly to connectivity test, CS/AIX needs IY79677, CS Linux requires v6.2.2 or later, and previous releases of z/OS need APAR PK17858.

Example 13-4 shows the D NET,EEDIAG,TEST=YES being blocked by a firewall.

Example 13-4 Connectivity test not getting through

```
IST350I DISPLAY TYPE = EEDIAG 677
IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000004
IST2131I EEDIAG DISPLAY COMPLETED ON 10/23/06 AT 19:56:41
IST2132I LDLC PROBE VERSIONS: VTAM = V1           PARTNER = UNKNOWN
IST1680I LOCAL IP ADDRESS 9.12.4.214
IST1680I REMOTE IP ADDRESS 19.155.34.181
IST924I -----------------------------------------------------------
IST2133I INTFNAME: OSA2040LNK              INTFTYPE: IPAQENET
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12000 1 I
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  9.12.4.92 2                RTT:     1
IST2137I     2  *                  (3)      RTT: *N/A*
IST2137I     8  *                  (3)      RTT: *N/A*
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12001 1
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  9.12.4.92                   RTT:     1
IST2137I     2  *                  (3)      RTT: *N/A*
IST2137I     8  *                  (3)      RTT: *N/A*
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12002 1
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  9.12.4.92                   RTT:     1
IST2137I     2  *                  (3)      RTT: *N/A*
IST2137I     8  *                  (3)      RTT: *N/A*
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12003 1
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  9.12.4.92                   RTT:     1
IST2137I     2  *                  (3)      RTT: *N/A*
IST2137I     8  *                  (3)      RTT: *N/A*
IST2135I   CONNECTIVITY UNSUCCESSFUL    SENSE: ***NA***   PORT: 12004 1
IST2136I   CONNECTIVITY TEST ENDED - MAXIMUM TIME LIMIT EXCEEDED
IST2137I     1  9.12.4.92                   RTT:     1
IST2137I     2  *                  (3)      RTT: *N/A*
IST2137I     8  *                  (3)      RTT: *N/A*
IST924I -------------------------------------------------------------
IST2139I CONNECTIVITY TEST RESULTS DISPLAYED FOR 1 INTERFACES
IST314I END
```

Here are explanatory notes to Example 13-4:

- 1: All 5 UDP ports were tested, none of them succeeded.

- 2: IP address 9.12.4.92 is the only router that returned an ICMP TIMEOUT message.

Example 13-5 shows the same display to a different IP address resulting in a successful connectivity test over multiple interfaces. If IPCONFIG MULTIPATH is enabled, VTAM will use all the available interfaces to test connectivity.

Example 13-5 Successful connectivity test

```
D NET,EEDIAG,TEST=YES,IPADDR=(10.10.1.241,10.10.1.231),LIST=DETAIL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = EEDIAG 249
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000002
IST2067I EEDIAG DISPLAY ISSUED ON 10/20/06 AT 23:05:00
IST1680I LOCAL IP ADDRESS 10.10.1.241
IST1680I REMOTE IP ADDRESS 10.10.1.231
IST2023I CONNECTED TO LINE EEM3100E
IST2126I CONNECTIVITY TEST IN PROGRESS
IST314I END
IST350I DISPLAY TYPE = EEDIAG 942
IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000007
IST2131I EEDIAG DISPLAY COMPLETED ON 10/23/06 AT 20:48:44
IST2132I LDLC PROBE VERSIONS: VTAM = V1           PARTNER = UNKNOWN
IST1680I LOCAL IP ADDRESS 10.10.1.232
IST1680I REMOTE IP ADDRESS 10.20.4.201
IST924I -------------------------------------------------------------
IST2133I INTFNAME: OSA2080LNK              INTFTYPE: IPAQENET
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12000
IST2137I     1  *                  (3)      RTT: *N/A*
IST2137I     2  10.20.4.201                 RTT:     1 1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12001
IST2137I     1  *                  (3)      RTT: *N/A*
IST2137I     2  10.20.4.201                 RTT:     1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12002
IST2137I     1  *                  (3)      RTT: *N/A*
IST2137I     2  10.20.4.201                 RTT:     1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12003
IST2137I     1  *                  (3)      RTT: *N/A*
IST2137I     2  10.20.4.201                 RTT:     1
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12004
IST2137I     1  *                  (3)      RTT: *N/A*
IST2137I     2  10.20.4.201                 RTT:     2
IST924I -------------------------------------------------------------
IST2133I INTFNAME: OSA20C0LNK              INTFTYPE: IPAQENET
IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12000
IST2137I     1  *                  (3)      RTT: *N/A*
IST2137I     2  10.20.4.201                 RTT:     1
...
IST2139I CONNECTIVITY TEST RESULTS DISPLAYED FOR 3 INTERFACES 2
IST314I END
```

Here are explanatory notes to Example 13-5:

- 1: The destination EE node is 2 hops away.
- 2: All 3 available interfaces to this IP destination were successfully tested.

13.2 Traces in CS for z/OS

Most problems in EE environments are caused by missing IP packets. In those cases, it must be verified whether the sending TCP/IP stack did send the packet into the network and whether it arrived at the receiving TCP/IP stack. An IP trace is required to perform this verification. In this section we describe how to take IP traces on z/OS using different tools.

- “IP packet trace to external CTRACE writer”
- “OSA Network Traffic Analyzer”
- “NetView Real-time Packet Tracing”
- “VTAM internal trace to external GTF dataset”

13.2.1 IP packet trace to external CTRACE writer

The TCP/IP packet trace is probably the most commonly used tool when it comes to diagnosing IP problems in z/OS. With HPR/IP EE being *just another* UDP application, a TCP/IP packet trace is a good start to diagnose EE problems from a z/OS perspective. It might be good enough to identify problems outside of z/OS. If this is the case you need to present the evidence to another group. For that purpose, it is possible to convert the z/OS packet trace into generally used LAN trace format to be analyzed with common networking tools.

This section provides information about:

- “Taking a packet trace”
- “Formatting a packet trace using IPCS interactively”
- “Converting a packet trace in a LAN trace using IPCS in batch mode”

Taking a packet trace

Here we describe the procedure to take a TCP/IP packet trace. It consist of 8 steps that must be issued in the right sequence, otherwise you will not get any data. It will prove very useful to automate this procedure and get used to it so that it can be done quickly and in a timely manner.

The sequence of actions is as follows:

1. Start CTRACE writer procedure in SYS1.PROCLIB.
2. Start SYSTCPDA component trace for the TCP/IP stack and connect it to the external CTRACE writer.
3. Start the packet trace in the TCP/IP stack.
4. Trace the incident as it occurs.
5. Stop the packet trace in the TCP/IP stack.
6. Disconnect SYSTCPDA from the external CTRACE writer.
7. Stop the external CTRACE writer.
8. Stop SYSTCPDA Component trace.

Example 13-6 shows the sequence of events with the associated messages when a TCP/IP packet trace is taken.

Example 13-6 Sequence of events when taking an IP packet trace

```
TRACE CT,WTRSTART=CTTCP 1
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.
ITT110I INITIALIZATION OF CTRACE WRITER CTTCP COMPLETE.

TRACE CT,ON,COMP=SYSTCPDA,SUB=(TCPIPA)  2
*027 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
 R 27,WTR=CTTCP,END
 IEE600I REPLY TO 027 IS;WTR=CTTCP,END
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.

V TCPIP,TCPIPA,PKT,ON,ABBREV=1500 3
EZZ0060I PROCESSING COMMAND: VARY TCPIP,TCPIPA,PKT,ON
EZZ0053I COMMAND VARY PKTTRACE COMPLETED SUCCESSFULLY

V TCPIP,TCPIPA,PKT,OFF  4
EZZ0060I PROCESSING COMMAND: VARY TCPIP,TCPIPA,PKT,OFF
EZZ0053I COMMAND VARY PKTTRACE COMPLETED SUCCESSFULLY

TRACE CT,ON,COMP=SYSTCPDA,SUB=(TCPIPA)   5
*028 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
 R 28,WTR=DISCONNECT,END
 IEE600I REPLY TO 028 IS;WTR=DISCONNECT,END
 ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
 WERE SUCCESSFULLY EXECUTED.

TRACE CT,WTRSTOP=CTTCP,FLUSH   6
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.
IEF196I AHL904I THE FOLLOWING TRACE DATASETS CONTAIN TRACE DATA :
IEF196I          SYS1.SC30.TCPIPA.CTRACE
AHL904I THE FOLLOWING TRACE DATASETS CONTAIN TRACE DATA : 070
          SYS1.SC30.TCPIPA.CTRACE

TRACE CT,OFF,COMP=SYSTCPDA,SUB=(TCPIPA)   7
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.
IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,024K) 088
        ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
        ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
```

Here are explanatory notes to Example 13-6:

- 1: Start an external CTRACE writer. The procedure is in SYS1.PROCLIB. You should trace into a single dataset with LRECL=27994 RECFM=VB and enough *primary* space to hold packet trace data for at least a couple of minutes. We recommend using a unique CTRACE writer procedure for every CTRACE component, otherwise there will be problems in stopping the writer while other components are still using it. ITT110I indicates that this step was successful and the writer is now waiting for data from any z/OS component. For a TCP/IP packet trace data the component capturing the data from the TCP/IP stack and passing it to the external CTRACE writer is called SYSTCPDA.

- 2: Start the CTRACE component called SYSTCPDA and connect it to the external CTRACE writer. ITT038I acknowledges this step as being successful. At this point the component SYSTCPDA is ready to accept packet trace data from the selected TCP/IP stack (TCPIPA).
- 3: Here the packet trace in our TCP/IP stack TCPIPA is started. In this example we trace all packets and cut them after 1500 bytes. As we only have a MTU size of 1500 this is not really meaningful, but it is there to show how you can limit the size of the packets being traced. In production systems, we recommend capturing only 200 bytes of each packet to save space in the trace dataset and avoid a wrap around too soon that would cause older data to be overridden. This amount of data, 200 bytes, is still enough to capture all the headers involved in EE.

Attention: We advice against applying filters during the capture of the trace. Other IP packets not matching a specific filter might still be worthwhile and might provide essential information to explain a misbehavior in EE. For instance ICMP messages from network devices.

- 4: Now packet tracing is enabled. Re-create the problem or wait for the problem to occur again. The size of the trace dataset determines how long it takes until the dataset wraps. Allow a long enough time to be able to react before the dataset wraps. Do not allocate too large datasets either, as this slows down the analysis of the trace dramatically. As a rule of thumb, if ABBREV=200 is used, 200 Cylinders primary space should give some ten-minutes time frame even in heavy loaded systems. You should determine what your needs are and adjust the size of the dataset accordingly *before* you need to document a problem. Ideally, automation should be used to capture intermittent problems and stop the running trace in a timely manner.
- 5: Stop the packet trace in the TCP/IP stack TCPIPA first.
- 6: Disconnect the SYSTCPDA component from the external CTRACE writer. Note that we did not turn the component trace **OFF** yet, as this would empty the trace buffer and the data would be lost.
- 7: Stop the external CTRACE writer and flush the data into the dataset.
- 8: Stop the SYSTCPDA component trace.

You should now be able to browse the dataset.

Example 13-7 shows a raw dataset containing a z/OS packet trace.

Example 13-7 Browse the raw CTRACE dataset

```
BROWSE    SYS1.SC30.TCPIPA.CTRACE                        CHARS X'2EE02EE0' found
Command ===> F x'2EE02EE0'
******************************************************* Top of Data **********
¶......×k¸ßü;.ö..........-$............Ø...EZBPTFM4SC30    SYSTCPDA........TCPIPA
¶......×k¸ßü;.ö..........-$............Ø...EZBPTFM4SC30    SYSTCPDA........TCPIPA
```

Here are explanatory notes to Example 13-7:

- You should recognize the system name SC30, the component `SYSTCPDA`, along with the jobname `TCPIPA`. If you traced EE traffic a **`F x'2EE02EE0'`** should find several occurrences.

‘2EE0’ is the hexadecimal notation of decimal 12000 which is the UDP port number used for LDLC test frames.

Formatting a packet trace using IPCS interactively

Enterprise Extender problems typically need several approaches to format the right packets in the right output format. We, therefore, recommend to use the interactive mode of IPCS initially to look at an EE packet trace.

We show some commonly used IPCS commands that proved helpful in diagnosis. The list is not complete and you should refer to *IP Diagnosis Guide,* GC31-8782. for a complete list of available options.

> **Note:** All the commands can be issued out of IPCS Option 6: Commands or through the Trace Analysis Panel.

These commands are as follows:

- **DROPD**
- **SETD DSN(‘SYS1.SC30.TCPIPA.CTRACE’)**
- **CTRACE COMP(SYSTCPDA) SUB((TCPIPA)) OPTION((SESSION PORT(12001)))**
- **CTRACE COMP(SYSTCPDA) SUB((TCPIPA)) OPT((PORT(12000) IPA(10.20.4.203) SUM))**
- **CTRACE COMP(SYSTCPDA) SUB((TCPIPA)) OPT((PORT(12001) IPA(10.20.4.202))) FU**
- **CTRACE COMP(SYSTCPDA) SUB((TCPIPA)) OPT((DUMP IPA(10.20.4.202)INT(OSA*)EBC))**

You must issue the following commands to format a packet trace:

- **DROPD** drops the dataset name out of the dump directory. This is required when you reused the same trace dataset to capture a new packet trace.
- **SETD** is used to define the input dataset.
- This **CTRACE** command prints a session report and gives statistic information about he connections found in the packet trace.
- This command shows how filters can be specified. We filter out a remote **IPAddress** and **PORT(12000)** and print a two-line **SUM**mary output for each packet.
- Same filter as in 4 but we get a **FU**ll report now, which means the formatter dissects all the headers it knows, such as IP, UDP, LDLC, NHDR, THDR, an so on.
- This output only contains packets that come from or go to all **INT**erfaces that start with the name **OSA** and the data portion is printed in EBCDIC.

Converting a packet trace in a LAN trace using IPCS in batch mode

It is often quite useful to feed IP traces into networking tools that provide a larger variety of analysis and filtering features. Also, if you need to discuss your findings with non-z/OS networking people, they would prefer looking at a LAN trace versus a formatted z/OS packet trace. Here is how to export a z/OS packet trace into a LAN trace format so it can be downloaded in binary as *filename.cap* and then be processed by commonly used networking tools.

Example 13-8 shows how a packet trace can be converted using IPCS in a batch JCL.

Example 13-8 JCL to convert z/OS Packet Trace into LAN trace for binary download

```
//PKT2SNIF JOB (999,POK),'BURKHARD',NOTIFY=&SYSUID,
//    CLASS=A,MSGCLASS=T,TIME=1439,
//    REGION=0M,MSGLEVEL=(1,1)
//    SET INDUMP='SYS1.SC30.TCPIPA.CTRACE'
//     SET SNIFF='SYS1.SC30.TCPIPA.D061018.CAP'
//IPCSBTCH EXEC PGM=IKJEFT01,DYNAMNBR=30
//IPCSDDIR DD DISP=SHR,DSN=&SYSUID..DDIR
//IPCSDUMP DD *
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//INDMP    DD DISP=SHR,DSN=&INDUMP.
//SNIFFER  DD DSN=&SNIFF.,UNIT=SYSDA,
//  DISP=(NEW,CATLG),LRECL=1560,SPACE=(CYL,(10,1)),RECFM=FB,DSORG=PS
//* DISP=SHR
//IPCSPRNT DD DSN=&SNIFF..SUMM,UNIT=SYSDA,
//  DISP=(NEW,CATLG),LRECL=133,SPACE=(CYL,(10,1)),RECFM=VBS,DSORG=PS
//IPCSTOC  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD *
 PROFILE MSGID
  IPCS NOPARM
  SETD PRINT NOTERM LENGTH(160000) NOCONFIRM FILE(INDMP)
  DROPD
 CTRACE COMP(SYSTCPDA) SUB((TCPIPA)) OPTIONS((SNIFFER(1500)))
 END
```

Note that in Example 13-8, you can apply all the options on the CTRACE command as you can in interactive mode.

13.2.2 OSA Network Traffic Analyzer

Starting with z/OS V1R8 it is possible to trace IP packets as they enter or leave the System z servers via an OSA-Express2 port. This trace shows the packets as they are seen in the LAN. For example, if the OSA needs to fragment IP packets because of an invalid MTU size, this trace will show the fragmented packets, whereas the IP packet trace would show them still complete. This section describes the steps required to prepare TCP/IP for the OSA Network Traffic Analyzer trace.

The following topics are covered:

- “Create a CTINAT00 member on SYS1.PARMLIB”
- “Restarting SYSTCPOT with a new buffer size”
- “Start the OSAENTA trace in TCP/IP”

> **Note:** This is just the preparation for the OSA Network Traffic Analyzer function in z/OS. The OSA function itself will be made available later. Check the PSP buckets (2094DEVICE and 2096DEVICE) for availability.

Create a CTINAT00 member on SYS1.PARMLIB

When you start a TCP/IP stack in z/OS V1R8 you will probably get the message shown in Example 13-9.

Example 13-9   CTRACE DEFINE FAILED FOR CTINTA00

```
EZZ4210I CTRACE DEFINE FAILED FOR CTINTA00,
         RETURN CODE: 0000000C REASON CODE: 00000401
```

The member CTINTA00 does not exist in SYS1.PARMLIB and so the message is issued during start of TCP/IP. You need to create a member with this name to avoid this message. Example 13-10 shows a CTINTA00 parmlib member that can be copied into SYS1.PARMLIB.

Example 13-10   CTINTA00 PARMLIB member to configure the OSA Network Traffic Analyzer

```
SYS1.PARMLIB(CTINTA00) - 01.03                           Columns 000
 ===>                                                           Scroll =
/*  DESCRIPTION = This parmlib member causes component trace for    */
/*              OSA Network Traffic Analyzer to start with a      */
/*              trace buffer size of 16 megabytes.               */
/*              To start the OSAENTA Trace either code           */
/*              OSAENTA ON PORTNAME=OSA4 IPADDR=9.1.2.2 in PROFILE*/
/*              or use                                           */
/*              V TCPIP,tcpip,OSAENTA,ON,PORTNAME=OSA4,IPADDR=    */
/*******************************************************************/
 TRACEOPTS
 /* --------------------------------------------------------------- */
 /*   ON OR OFF: PICK 1                                             */
 /* --------------------------------------------------------------- */
          ON
 /*       OFF                                                       */
 /* --------------------------------------------------------------- */
 /*   BUFSIZE: A VALUE IN RANGE 1M TO 256M                          */
 /* --------------------------------------------------------------- */
          BUFSIZE(16M)
```

Here are explanatory notes to Example 13-10:

- With the BUFSIZE parameter you specify the amount of storage that will be used in TCP/IP’s data space to hold the trace data received from the OSA card. We defined a buffer of 16 MB. The status of this CTRACE can be displayed.

Example 13-11 shows the D TRACE,COMP=SYSTCPOT,SUB=(TCPIPA) command

Example 13-11   SYSTCPOT CTRACE running

```
D TRACE,COMP=SYSTCPOT,SUB=(TCPIPA)
IEE843I 11.38.22  TRACE DISPLAY 822
        SYSTEM STATUS INFORMATION
 ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,024K)
  TRACENAME
  =========
  SYSTCPOT
                      MODE BUFFER HEAD SUBS
                      =====================
                      OFF        HEAD    2
    NO HEAD OPTIONS
  SUBTRACE            MODE BUFFER HEAD SUBS
 ---------------------------------------------------------------
  TCPIPA              ON   0016M
     ASIDS      *NONE*
     JOBNAMES   *NONE*
     OPTIONS    MINIMUM
     WRITER     *NONE*
```

Here is an explanatory note to Example 13-11:

- SYSTCPOT, like any other component trace, can be displayed using the D TRACE command. It shows the buffer size set to 16 M.

Restarting SYSTCPOT with a new buffer size

The size can be modified without a TCP/IP restart by stopping the CTRACE SYSTCPOT and restarting it with the new size.

Example 13-12 shows the commands to restart the component trace with a new buffer size.

Example 13-12 Stopping SYSTCPOT and restarting with a new buffer size

```
TRACE CT,OFF,COMP=SYSTCPOT,SUB=(TCPIPA) 1
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.
IEE839I ST=(ON,0256K,00512K) AS=ON  BR=OFF EX=ON  MT=(ON,024K) 879
        ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
        ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
TRACE CT,32M,COMP=SYSTCPOT,SUB=(TCPIPA) 2
*037 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
 R 37,END
```

Here are explanatory notes to Example 13-12:

- 1: With this command the running SYSTCPOT trace is stopped.
- 2 The TRACE CT,32M, instead of the ‘ON’ defines a larger size of the buffer.

Start the OSAENTA trace in TCP/IP

Now that we have the CTRACE running and waiting for packets to arrive, we need to activate the tracing in TCP/IP. Example 13-13 shows the VARY command in TCP/IP.

Example 13-13 Starting OSA Network Traffic Analyzer trace in TCPIPA

```
V TCPIP,TCPIPA,OSAENTA,ON,PORTNAME=OSA20A0
EZZ0060I PROCESSING COMMAND: VARY TCPIP,TCPIPA,OSAENTA,ON,PORTNAME=OSA20A0
EZZ0053I COMMAND VARY OSAENTA COMPLETED SUCCESSFULLY
EZD0015I OSA20A0 DOES NOT SUPPORT OSAENTA TRACE
EZD0018I OSAENTA TRACE DISABLED FOR OSA20A0
```

Here is an explanatory note to Example 13-13:

- As the installed microcode level of the OSA cards did not support this new function, the trace command fails with EZD0015I.

13.2.3 NetView Real-time Packet Tracing

Starting with z/OS V1R6, TCP/IP provides a new Network Management Interface (NMI) that can be used to get packet trace data directly from the TCP/IP stack, without the need for the SYSTCPDA component. Here is what needs to be done to enable this function.

- “Update TCP/IP profile to include the NETMONITOR statement”
- “Issue NetView PKTS commands to start and stop data collection”
- “Format the IP packets using the NetView FMTPACKT command”

Update TCP/IP profile to include the NETMONITOR statement

Before any network management product can use the new NMI, it must be enabled in the TCP/IP profile. Example 13-14 on page 418 shows the NETMONITOR statement in TCP/IP profile.

Example 13-14 TCPIP profile statement

```
NETMONITOR PKTTRCSERVICE
```

Now TCP/IP is enabled to provide the new NMI interface. There are additional parameters in the NETMONITOR statement, which are not shown here as they are not relevant for EE.

Issue NetView PKTS commands to start and stop data collection

Now that TCP/IP enabled the NMI interface, it is time to start the trace in NetView. Example 13-15 shows the required steps to activate this trace.

Example 13-15 Tracing procedure via the NetView PKTS commands

```
PKTS PURGE 1
PKTS DEFINE TCPNAME=TCPIPA OPID=AUTO1 2
DSI633I DEFINE COMMAND SUCCESSFULLY COMPLETED
MVS V TCPIP,TCPIPA,PKT,ON 3
PKTS START TCPNAME=TCPIPA 4
DSI500I 'COLLECTION for PKTS-TCPIPA' restarted

PKTS STOPCOLL 5
BNH605I SOCKET INTERFACE HAS TERMINATED
```

Here are explanatory notes to Example 13-15:

- 1: PKTS PURGE clears all previously taken packets.
- 2: PKTS DEFINE associates a TCP/IP stack name with the name of an autotask that will collect packet data for the stack.
- 3: V TCPIP,TCPIPA,PKT,ON starts tracing in the stack.
- 4: PKTS START starts the collection of packet data.
- 5: PKTS STOPCOLL terminates the collection of packet data.

Format the IP packets using the NetView FMTPACKT command

Example 13-16 shows how to format the packet trace data.

Example 13-16 Formatting the captured packets using FMTPACKT command

```
WINDOW FMTPACKT RADDR=10.20.4.203 RPORT=12000 FULL MAXRECS=100
1225768 SC30     PACKET   00000004 13:27:04.247303 Packet Trace
 From Interface    : OSA20COLNK       Device: QDIO Ethernet    Full=31
  Tod Clock        : 2006/10/23 13:27:04.247302                Intfx: 15
  Sequence #       : 0                Flags: Pkt Ver2 Adj
  Source           : 10.20.4.203
  Destination      : 10.10.1.232
  Source Port      : 12000            Dest Port: 12000 Asid: 0046 TCB: 00000000
 IpHeader: Version : 4                Header Length: 20
  Tos              : C0               QOS: Internetwork Normal Service
  Packet Length    : 31               ID Number: 23D7
  Fragment         :                  Offset: 0
 TTL     : 29      Protocol: UDP     CheckSum: 5E67 FFFF
  Source          : 10.20.4.203
  Destination     : 10.10.1.232

 UDP
  Source Port     : 12000 (EE-XID)   Destination Port: 12000 (EE-XID)
  Datagram Length : 11               CheckSum: 9042 FFFF
EE: 3
LDLC:
 Remote Sap       : 04               Source Sap: 04 Request  Control: F3 (TEST
 Control)
IP Header         : 20
000000 45C0001F 23D70000 1D115E67 0A1404CB  0A0A01E8
Protocol Header   : 8
000000 2EE02EE0 000B9042
Data              : 3      Data Length: 3
000000 0404F3                            |..3             |
```

Note that in Example 13-16, there are several options to format the packet trace data. The format is identical to the IPCS formatter. For more information about tracing in NetView, see the readme text of OA04304

13.2.4 VTAM internal trace to external GTF dataset

This section describes how to take a VTAM Internal Trace (VIT) to document EE problems. While this trace is most probably not the one that you will typically look at, it is the trace that IBM support will ask for in most cases. This is because the VIT shows all the layers involved as data flows from an application into the network and back. It can be run internally into ECSA/data space or externally into a GTF dataset. The following steps illustrate the procedure of tracing into an external GTF dataset:

- “Modify GTF procedure and parameters”
- “Start the GTF procedure”
- “Start the VTAM internal trace”
- “Stop the GTF procedure”
- “Stop the VTAM internal trace”

Modify GTF procedure and parameters

In this section we show some parameters, that help to gather the right documentation information. Example 13-17 shows the GTF procedure and parameters to capture VTAM Internal Trace data.

Example 13-17 shows the GTF procedure.

Example 13-17 GTF procedure

```
SYS1.PROCLIB(GTFVTAM)
//GTFVTAM PROC MEMBER=GTFVTAM
//IEFPROC EXEC PGM=AHLGTF,PARM='MODE=EXT,DEBUG=NO,TIME=YES,BLOK=4M',   * 1
//  TIME=1440,REGION=0M
//IEFRDER DD   DSNAME=SYS1.&SYSNAME..GTFVTAM,DISP=SHR  2
//*EFRDER DD   DSNAME=SYS1.TRACE,UNIT=SYSDA,SPACE=(CYL,(200,10)),       *
//* DISP=(NEW,CATLG)
//SYSLIB  DD   DSNAME=SYS1.PARMLIB(&MEMBER),DISP=SHR
```

Here are explanatory notes to Example 13-17:

- 1: In GTF’s procedure, we recommend coding BLOK=4M, a parameter that increases an internal buffer in case there is some congestion in writing the trace data to disk.
- 2: The Trace dataset should be allocated with a LRECL=27994, RECFM=VB with enough primary allocation space to cover at least a couple of minutes worth of trace data. How large the dataset must be depends on the load of the system. Usually 200 Cylinders should be enough to capture a time frame of several minutes. Make sure that you verify how long it takes to wrap the dataset in your installation by running a trace during peak load.

> **Attention:** GTF does not use secondary space allocation for the trace dataset, so allocate the required size in *primary* space.

Example 13-18 shows the SYS.PARMLIB member that is selected by the procedure.

Example 13-18 TGTFVTAM parameter file on SYS1.PARMLIB

```
SYS1.PARMLIB(GTFVTAM)
TRACE=USRP                              1
USR=(FEF,FF1,FE1,FE2,FE4)               2
END
```

Here are explanatory notes to Example 13-18:

- 1: The PARMLIB member should prompt for USR keywords.
- 2: The USR keyword should only contain VTAM Event IDs (EIDs). The ID FE1 is required for VTAM Internal Trace records. The remaining EIDs are used for other VTAM traces.

> **Attention:** Do not start GTF with TRACE=USR, that is without prompting for USR keywords. The trace dataset will be flooded with other non-relevant trace records and may cause important data to be overridden.

Start the GTF procedure

Example 13-19 shows the required commands to successfully start GTF.

Example 13-19 Commands and messages when GTF is started

```
S GTFVTAM.GTF 1
 IRR812I PROFILE ** (G) IN THE STARTED CLASS WAS USED 218
         TO START GTFVTAM WITH JOBNAME GTFVTAM.
 $HASP100 GTFVTAM  ON STCINRDR
 IEF695I START GTFVTAM  WITH JOBNAME GTFVTAM  IS ASSIGNED TO USER
 IBMUSER , GROUP SYS1
 $HASP373 GTFVTAM  STARTED
 AHL121I  TRACE OPTION INPUT INDICATED FROM MEMBER GTFVTAM  OF PDS
 SYS1.PARMLIB
 TRACE=USRP
 USR=(FEF,FF1,FE1,FE2,FE4)
 END
 AHL103I  TRACE OPTIONS SELECTED --USR=(FEF,FE1,FE2,FE4,FF1) 2
*026 AHL125A  RESPECIFY TRACE OPTIONS OR REPLY U
R 26,U
IEE600I REPLY TO 026 IS;U
U
```

```
AHL080I GTF STORAGE USED FOR GTF DATA: 243
        GTFBLOCK STORAGE     4101K BYTES (BLOK=      4096K)
        PRIVATE STORAGE      1038K BYTES (SIZE=      1024K)
        SADMP HISTORY          54K BYTES (SADMP=       40K)
        SDUMP HISTORY          54K BYTES (SDUMP=       40K)
        ABEND DUMP DATA         0K BYTES (ABDUMP=       0K)
AHL031I GTF INITIALIZATION COMPLETE
```

Here are explanatory notes to Example 13-19:

- 1: If you start the procedure with a stop qualifier .GTF it can later be stopped with that stop qualifier.
- 2: AHL103I lists the currently selected USR keywords and allows you to override them here. By replying *u* the presented values are accepted and AHL031I reports the successful initialization of GTF.

Now GTF is ready to capture trace records from VTAM.

Start the VTAM internal trace

In VTAM we must activate the VIT with Options PIU,HPR,CIA and MODE=EXT to direct the trace records to GTF.

Example 13-20 shows the VTAM command to start external tracing.

Example 13-20 Starting VTAM internal Trace

```
F NET,TRACE,TYPE=VTAM,MODE=EXT,BFRNUM=2,OPT=(PIU,HPR,CIA) 1
IST097I MODIFY ACCEPTED
IST315I VTAM INTERNAL TRACE ACTIVE - MODE = INT, SIZE = 999 PAGES 250 2
IST1659I DATA SPACE ISTITDS1 DSPSIZE = 5 (50M)
IST199I OPTIONS = CIA HPR PSS SMS
IST315I VTAM INTERNAL TRACE ACTIVE - MODE = EXT, SIZE = 002 BUFFERS
IST199I OPTIONS = CIA HPR PIU
IST314I END
```

Note the following explanations for Example 13-20:

- 1: The MODE=EXT redirects the trace records to GTF.

  BFRNUM is the amount of 8 K buffers that VTAM collects before sending them to GTF. All VTAM Internal Trace records in these buffers will have the same timestamps, so if you need to have very accurate timestamps, we recommend coding BFRNUM=0.

  The trace options showing the packets through all the layers are:

  – PIU, tracing Path Information Units
  – HPR tracing Network Layer Packets
  – CIA tracing IP packets at the DLC layer

- 2: Notice that 2 VITs are active, one running internally with the options we activated in ATCSTR00, and the external VIT that we just started. The active trace options can be different.

To terminate the trace, we recommend stopping GTF first, to avoid flooding of the existing dataset while you are issuing other commands.

Stop the GTF procedure

Example 13-21 shows the MVS Stop command used to terminate GTF procedure with the stop qualifier.

Example 13-21 Stopping GTF

```
P GTF
AHL006I GTF ACKNOWLEDGES STOP COMMAND
AHL904I THE FOLLOWING TRACE DATASETS CONTAIN TRACE DATA :
          SYS1.SC30.GTFVTAM 1
$HASP395 GTFVTAM  ENDED

BROWSE    SYS1.SC30.GTFVTAM 2
 Command ===>  f x'EFE1'
*******************************************************
....-$×kÌÃ..B.........GTS .. .SP7.0.8 HBB7730 SC30    .
.Ù×k`$¥.ÑiÕ÷.Ûö.TCPIPA  BUFFå...........×k`ïËÆÉó......ø
.Ù×k`).Äµ+Õ÷. .ØNET     BUFFä...........×k`$¥.Ú.......ø
.Ù×k`¬.òÔ.Õ÷. .ØNET     BUFF............×k`).À.+......&
OÙ×k`¬%L~öÕ÷.......Ð.ÛI.OMPA    BUFF............×k`¬.óf
....-$×k`¬%LB<........GTS .. .SP7.0.8 HBB7730 SC30    .
3Ù×k`¬%L~öÕ÷.......Ðb.....b.3RDBOOKEE.EECPWIPIU2N...E..
.Ù×k`/Ûý'+Õ÷.Ûö.TCPIPA  BUFFä...........×k`¬%M+ö......ø
```

Note the following explanations for Example 13-21:

- **1**: AHL904I lists the trace dataset that contains the VTAM internal trace records.
- **2**: Browsing the trace dataset you should see the system name SC30, and BUFF entries. To verify if this is really VIT data, issue a find on x'EFE1', which is the Event ID for VTAM internal Trace records.

Stop the VTAM internal trace

Finally you can stop the external VIT using the command in Example 13-22.

Example 13-22 Stop external VIT

```
F NET,NOTRACE,TYPE=VTAM,MODE=EXT,OPT=(END)
IST097I MODIFY ACCEPTED
IST315I VTAM INTERNAL TRACE ACTIVE - MODE = INT, SIZE = 999 PAGES
IST1659I DATA SPACE ISTITDS1 DSPSIZE = 5 (50M)
IST199I OPTIONS = CIA HPR PSS SMS
IST314I END
```

Here are explanatory note to Example 13-22:

- After stopping the *external* VIT, the *internal* VIT is still running with the options activated in ATCSTR00.

Finally, as taking an external trace is not a trivial process, we recommend using automation to do this and to get used to the process. In the next section we describe how this documentation can be gathered more conveniently using VTAM's F NET,CSDUMP command.

13.3 F NET,CSDUMP command

This section describes how the collection of EE relevant diagnostic information can be simplified. In “TRACE” on page 106 we activated the VTAM Internal Trace to trace into a 50 MB storage area in VTAM’s data space. In case of a problem, you only have to dump the data space and the problem is documented with a matching VIT. The VTAM command F NET,CSDUMP can be used to dump that data space along with VTAM’s address space. In TCP/IP you can also trace into a data space instead of tracing to external CTRACE writer. Starting with z/OS V1R8, the CSDUMP command has been enhanced to specify a TCP/IP stack name that can be dumped in addition to the VTAM address, and data space. CSDUMP can be used to trigger events like VTAM messages or sense codes, enabling automatic documentation collection of intermittent problems.

We discuss the following commands:

- “Default CSDUMP dumping VTAM and VTAM’s data space ISTITDS1”
- “CSDUMP triggered by a VTAM message, dumping VTAM and TCP/IP”

Default CSDUMP dumping VTAM and VTAM’s data space ISTITDS1

This command takes a dump immediately. It dumps VTAM’s ASID and the ISTITDS1 data space.

Example 13-23 shows how a CSDUMP is taken.

Example 13-23 F NET,CSDUMP

```
F NET,CSDUMP
IEA794I SVC DUMP HAS CAPTURED: 256
DUMPID=003 REQUESTED BY JOB (NET     )
DUMP TITLE=ISTRACSW - DEFAULT CSDUMP WITH ISTITDS1 1
IST097I MODIFY ACCEPTED
IST1881I VTAM DUMPING FOR CSDUMP - IMMEDIATE DUMP
IST223I MODIFY CSDUMP COMMAND COMPLETED

IEA611I COMPLETE DUMP ON DUMP.D1021.H03.SC31.NET.S00003 2
DUMPID=003 REQUESTED BY JOB (NET     )
FOR ASID (003B)
```

Here are explanatory notes to Example 13-23:

- 1: The title indicates that a default CSDUMP with the data space is taken.
- 2: IEA611I informs you, whether the dump was complete or partial and which dataset is holding the dump. Make sure your dump datasets are large enough to capture complete dumps.

CSDUMP triggered by a VTAM message, dumping VTAM and TCP/IP

In EE environments, it is often necessary to look at both, the SNA and the IP side, so it may be necessary to provide traces of both the components. The V1R8 enhancement of CSDUMP enables to dump TCP/IP’s data space and address space along with the VTAM information. If a packet trace was running into data space, the dump will include the packet trace data also and can be processed like an external CTRACE writer dataset.

Example 13-24 on page 424 shows how to activate packet tracing into data space and then automatically take a dump on a VTAM message.

Example 13-24 CSDUMP triggered on VTAM message including TCP/IP stack

```
TRACE CT,ON,COMP=SYSTCPDA,SUB=(TCPIPA) 1
*036 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
R 36,END
IEE600I REPLY TO 036 IS;END
ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE TRACE CT COMMAND
WERE SUCCESSFULLY EXECUTED.
V TCPIP,TCPIPA,PKT,ON 2
EZZ0060I PROCESSING COMMAND: VARY TCPIP,TCPIPA,PKT,ON
EZZ0053I COMMAND VARY PKTTRACE COMPLETED SUCCESSFULLY

F NET,CSDUMP,MESSAGE=IST1430I,TCPNAME=TCPIPA 3
IST097I MODIFY ACCEPTED
IST223I MODIFY CSDUMP COMMAND COMPLETED
...
IST1430I  REASON FOR INOP IS XID OR LDLC COMMAND TIMEOUT 4
IST1879I VTAM DUMPING FOR CSDUMP TRIGGER MESSAGE IST1430I
IEA794I SVC DUMP HAS CAPTURED: 508
DUMPID=002 REQUESTED BY JOB (NET     )
DUMP TITLE=ISTRACSW - MSG CSDUMP WITH ISTITDS1 AND TCPIP
...
IEA611I COMPLETE DUMP ON DUMP.D1023.H21.SC30.NET.S00002 5
DUMPID=002 REQUESTED BY JOB (NET     )
FOR ASIDS(0056,0046)
```

Note the following explanations for Example 13-24:

- 1: SYSTCPDA component is activated to trace into data space (No external CTRACE writer is specified).
- 2: Packet trace is activated in the EE stack.
- 3: CSDUMP command is issued, waiting for IST1480I.
- 4: IST1480I occurs, causing the CSDUMP probe to match and VTAM taking a dump.
- 5: IEA611 indicating a complete dump was written.

The dump contains the TCP/IP packet trace which can be processed as though it was an external CTRACE writer. Example 13-25 shows an example to export the data space packet trace into a LAN compatible format.

Example 13-25 JCL to export a packet trace from a dump

```
//PKT2SNIF JOB (999,POK),'BURKHARD',NOTIFY=&SYSUID,
//    CLASS=A,MSGCLASS=T,TIME=1439,
//    REGION=0M,MSGLEVEL=(1,1)
//    SET INDUMP='DUMP.D1023.H21.SC30.NET.S00002'
//     SET SNIFF='SYS1.SC30.TCPIPA.D061023.CAP'
//IPCSBTCH EXEC PGM=IKJEFT01,DYNAMNBR=30
//IPCSDDIR DD DISP=SHR,DSN=&SYSUID..DDIR
//IPCSDUMP DD *
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//INDMP    DD DISP=SHR,DSN=&INDUMP.
//SNIFFER  DD DSN=&SNIFF.,UNIT=SYSDA,
//  DISP=(NEW,CATLG),LRECL=1560,SPACE=(CYL,(20,1)),RECFM=FB,DSORG=PS
//* DISP=SHR
```

```
//IPCSPRNT DD DSN=&SNIFF..SUMM,UNIT=SYSDA,
//  DISP=(NEW,CATLG),LRECL=133,SPACE=(CYL,(10,1)),RECFM=VBS,DSORG=PS
//IPCSTOC  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD *
 PROFILE MSGID
  IPCS NOPARM
  SETD PRINT NOTERM LENGTH(160000) NOCONFIRM FILE(INDMP)
  DROPD
 CTRACE COMP(SYSTCPDA) SUB((TCPIPA)) OPTIONS((SNIFFER(1500)))
 END
```

13.4 Summary

Most EE problems turn out to be caused by packets not reaching their destination, arriving with a delay, fragmented, or out of sequence. To prove where the problem lies, it usually takes two traces to run concurrently, one at each end of the EE connection. An IP trace on each side is a good start, but may not be enough if the problem turns out to be in the SNA/HPR side. We recommend using a standard format of the traces to exchange the documentation across platforms or send them to other groups for analysis. The Open Source Software WireShark, formerly known as Ethereal, can be used to analyze IP traces taken on various platforms, including z/OS if they were converted using IPCS. It can be downloaded from:

http://www.wireshark.org

A

z/OS configuration files for NETID=RDBOOKEE

The lab scenarios presented in this book use systems from two APPN NETIDs: RDBOOKEE and USIBMSC. The z/OS systems are SC30, SC31, and SC32. The source of the z/OS configuration files used in support of NETID=RDBOOKEE are included in this appendix.

The z/OS configuration files are organized as follows:

- "Configuration files common to all three systems" on page 428
- "Configuration files for SC30 Network Node" on page 434
- "Configuration files for SC31 Network Node" on page 443
- "Configuration files for SC32 End Node" on page 451

A.1 Configuration files common to all three systems

This section includes the following configuration files common to all systems:

- “Common VTAM files” on page 428
- “Common TCP/IP files” on page 432

A.1.1 Common VTAM files

This section includes common VTAM files:

- “Common VTAM procedure”
- “Common VTAM ATCSTR00 base for all systems”
- “Common VTAM Transmission Group Profiles”
- “Common VTAM APPN Class of Service table”
- “Common VTAM XCF Model major node”
- “Common VTAM EE Model major node”
- “Common VTAM XCA major node (with symbolics)”

Common VTAM procedure

The NET PROC is a common procedure for all three systems in the sysplex, SC30, SC31, and SC32.

Example: A-1 Common procedure for VTAM

```
//NET PROC
//NET  EXEC   PGM=ISTINM01,REGION=2048K,DPRTY=(15,12)
//STEPLIB DD DSN=NCP.SSPLIB,DISP=SHR
//VTAMLST DD DSN=SYS1.VTAMLST,DISP=SHR
//VTAMLIB DD DSN=SYS1.LOCAL.VTAMLIB,DISP=SHR
//        DD DSN=SYS1.VTAMLIB,DISP=SHR
//NCPLOAD DD DSN=NCPUSER.LOADLIB,DISP=SHR
//NCPDUMP DD DSN=NCPUSER.NCPDUMP,DISP=SHR
//CSPDUMP DD DSN=NCPUSER.CDUMP,DISP=SHR
//MOSSDUMP DD DSN=NCPUSER.MDUMP,DISP=SHR
//SISTCLIB DD DSN=SYS1.SISTCLIB,DISP=SHR
//TRSDB    DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..TRSDB
//DSDBCTRL DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..DSDBCTRL
//DSDB1    DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..DSDB1
//DSDB2    DD DISP=SHR,DSN=SYS1.VTAM.&SYSNAME..DSDB2
```

Common VTAM ATCSTR00 base for all systems

Example: A-2 Common ATCSTR00 base

```
CONNTYPE=APPN,                                                         X
CPCP=YES,                                                              X
APPNCOS=#CONNECT,                                                      X
DIALRTRY=NO,                                                           X
DUPDEFS=NONE,                                                          X
DYNADJCP=YES,                                                          X
DYNLU=YES,                                                             X
DYNMODTB=ALLMODES,                                                     X
ENCRYPTN=NO,                                                           X
HOSTSA=&SYSCLONE.,SACONNS=NO,                                          X
ISTCOSDF=ALL,                                                          X
```

```
IOPURGE=120,                                                            X
NETID=RDBOOKEE,SSCPNAME=&SYSNAME.M,                                     X
SAVERSCV=YES,                                                           X
TRACE,TYPE=VTAM,MODE=INT,SIZE=999,DSPSIZE=5,OPT=(CIA,HPR),              X
XCFINIT=YES,                                                            X
APBUF=(56,,2,,1,3),                                                     X
BSBUF=(1200,,,,48,48),                                                  X
CRA4BUF=(4000,,0,,10,20),                                               X
CRA8BUF=(396,,0,,6,2),  ),                                              X
CRPLBUF=(1300,,0,,60,29),                                               X
IOBUF=(2000,447,19,,8,48),                                              X
LFBUF=(8160,,0,,1,1),                                                   X
LPBUF=(54,,0,,6,2),                                                     X
SFBUF=(192,,0,,1,1),                                                    X
SPBUF=(84,,0,,1,1),                                                     X
TIBUF=(2000,,0,,60,120),                                                X
T1BUF=(2000,,0,,60,120),                                                X
T2BUF=(2000,,0,,60,120),                                                X
XDBUF=(60,,0,,1,4),                                                     X
SSCPID=&SYSCLONE.,                                                      X
CONFIG=&SYSCLONE.,                                                      X
HOSTPU=SC&SYSCLONE.PU,                                                  X
NOPROMPT,SUPP=NOSUP,                                                    X
IQDCHPID=F7,                                                            X
PPOLOG=YES
*DYNADJCP=YES,
*DYNLU=NO,
```

Common VTAM Transmission Group Profiles

The IBM supplied IBMTGPS table was modified and saved as EETGPS. Only those entries that were modified are included here.

Example: A-3   Common EETGPS (modified IBMTGPS)

```
************************************************************************
* XCF                                                                  *
*                                                                      *
* The CAPACITY value of 600M is chosen for XCF instead of the 32M *
* for other AHHC connections                                           *
************************************************************************
XCF      TGP   COSTTIME=0,COSTBYTE=0,SECURITY=SECURE,                   *
               PDELAY=NEGLIGIB,CAPACITY=600M,                           *
               UPARM1=60,UPARM2=60,UPARM3=60
.
************************************************************************
* Enterprise Extender, Campus                                          *
************************************************************************
EEXTCAMP TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                 *
               PDELAY=NEGLIGIB,CAPACITY=158M,                           *
               UPARM1=90,UPARM2=90,UPARM3=90
.
************************************************************************
* TG Profile to match AIX defaults for connection network links        *
************************************************************************
TGPVRN   TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                 *
```

```
               PDELAY=NEGLIGIB,CAPACITY=4M,                            *
               UPARM1=0,UPARM2=0,UPARM3=0
.
**********************************************************************
* Enterprise Extender, Wide Area Network                             *
**********************************************************************
EEXTWAN  TGP   COSTTIME=100,COSTBYTE=0,SECURITY=UNSECURE,              *
               PDELAY=PACKET,CAPACITY=2M,                              *
               UPARM1=90,UPARM2=90,UPARM3=90
.
**********************************************************************
* Fast Ethernet                                                      *
**********************************************************************
FASTENET TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=100M,                          *
               UPARM1=40,UPARM2=40,UPARM3=40
.
**********************************************************************
* Gigabit Ethernet                                                   *
**********************************************************************
GIGENET  TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=1000M,                         *
               UPARM1=20,UPARM2=20,UPARM3=20
.
**********************************************************************
* FICON                                                              *
**********************************************************************
FICON    TGP   COSTTIME=0,COSTBYTE=0,SECURITY=SHIELDED,                *
               PDELAY=NEGLIGIB,CAPACITY=600M,                          *
               UPARM1=40,UPARM2=20,UPARM3=20
.
**********************************************************************
* TGP to match the default characteristics of CS/AIX and CS/Linux    *
**********************************************************************
TGPCS/AIX TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=158M,                          *
               UPARM1=128,UPARM2=128,UPARM3=128
.
**********************************************************************
* TGP to match the default characteristics of System i V5R4          *
**********************************************************************
TGPAS400 TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=TERRESTR,CAPACITY=5M,                            *
               UPARM1=128,UPARM2=128,UPARM3=128
.
**********************************************************************
* TGP to match the default characteristics of SNASw                  *
**********************************************************************
TGPSNASW TGP   COSTTIME=196,COSTBYTE=196,SECURITY=UNSECURE,            *
               PDELAY=NEGLIGIB,CAPACITY=16M,                           *
               UPARM1=128,UPARM2=128,UPARM3=128
.
**********************************************************************
* TGP to match the default characteristics of HIS Server             *
**********************************************************************
```

```
TGPMSHIS TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=100M,                          *
               UPARM1=128,UPARM2=128,UPARM3=128
*
```

Common VTAM APPN Class of Service table

Used IBM supplied default, until preparing for EE. The entire table is not shown here. Only the explanation of the modifications is included.

Example: A-4   Common EECOSTAB (modified COSAPPN)

```
***********************************************************************
*                                                                     *
* MACRO NAME(S):    EECOSTAB                                          *
*                                                                     *
* DESCRIPTIVE NAME: Modified APPNCOS table derived from ISTACST2      *
*                   Specify the weight for a given APPNCOS via UPARMx *
*                   to be able to prefer a TG over another one.       *
*                   UPARM1 controls the high priority COS             *
*                   UPARM2 controls the medium priority COS           *
*                   UPARM3 controls the low priority COS              *
*                                                                     *
*                   __________________________________________        *
*                  | APPNCOS  |UPARM1  UPARM2  UPARM3   WEIGHT|       *
*                  |__________|_______|_______|_______|_______|       *
*  1st choice      | SNASVCMG | 20-20 | ANY   | ANY   |  20   |       *
*  definitions     | #INTER   | 20-20 | ANY   | ANY   |  20   |       *
*  get a weight    | #INTERSC | 20-20 | ANY   | ANY   |  20   |       *
*  of 20           | #CONNECT | ANY   | 20-20 | ANY   |  20   |       *
*                  | #BATCH   | ANY   | ANY   | 20-20 |  20   |       *
*                  | #BATCHSC | ANY   | ANY   | 20-20 |  20   |       *
*                  |__________|_______|_______|_______|_______|       *
*  2nd choice      | SNASVCMG | 40-40 | ANY   | ANY   |  40   |       *
*  definitions     | #INTER   | 40-40 | ANY   | ANY   |  40   |       *
*  get a weight    | #INTERSC | 40-40 | ANY   | ANY   |  40   |       *
*  of 40           | #CONNECT | ANY   | 40-40 | ANY   |  40   |       *
*                  | #BATCH   | ANY   | ANY   | 40-40 |  40   |       *
*                  | #BATCHSC | ANY   | ANY   | 40-40 |  40   |       *
*                  |__________|_______|_______|_______|_______|       *
*  3rd choice      | SNASVCMG | 60-60 | ANY   | ANY   |  60   |       *
*  definitions     | #INTER   | 60-60 | ANY   | ANY   |  60   |       *
*  get a weight    | #INTERSC | 60-60 | ANY   | ANY   |  60   |       *
*  of 60           | #CONNECT | ANY   | 60-60 | ANY   |  60   |       *
*                  | #BATCH   | ANY   | ANY   | 60-60 |  60   |       *
*                  | #BATCHSC | ANY   | ANY   | 60-60 |  60   |       *
*                  |__________|_______|_______|_______|_______|       *
*  3rd choice      | SNASVCMG | 41-60 | ANY   | ANY   |  60   |       *
*  definitions     | #INTER   | 41-60 | ANY   | ANY   |  60   |       *
*  get a weight    | #INTERSC | 41-60 | ANY   | ANY   |  60   |       *
*  of 60           | #CONNECT | ANY   | 41-60 | ANY   |  60   |       *
*                  | #BATCH   | ANY   | ANY   | 41-60 |  60   |       *
*                  | #BATCHSC | ANY   | ANY   | 41-60 |  60   |       *
*                  |__________|_______|_______|_______|_______|       *
```

Common VTAM XCF Model major node

Example: A-5   Common XCF model, MODELXCF

```
***********************************************************************
* MODEL MAJORNODE TO ASSIGN A TGP TO DYNAMIC XCF LINKS
***********************************************************************
MODELXCF VBUILD TYPE=MODEL
ISTP*    PU    TRLE=*,CPCP=YES,TGP=XCF
```

Common VTAM EE Model major node

Example: A-6   Common EE Model major node, EEMODEL

```
EEMODEL  VBUILD TYPE=MODEL
EEPUMODL PU    DYNTYPE=EE,DISCNT=NO,CPCP=YES,TGP=EEXTCAMP
VRNMODEL PU    DYNTYPE=VN,DISCNT=(YES,,120)
RTPMODEL PU    DYNTYPE=RTP,DISCNT=NO
```

Common VTAM XCA major node (with symbolics)

Two different XCA major nodes were used and tested. One with system symbolics and host names, the other with no symbolics and IP addresses. Using symbolics enabled us to use a common XCA major node on all three systems. If symbolics are not used, then a unique EE XCA major node is necessary for each system.

Example: A-7   Common EE XCA major node, EEXCA, with symbolics

```
EEXCA&SYSCLONE. VBUILD TYPE=XCA
EEPORT&SYSCLONE. PORT MEDIUM=HPRIP,                                    *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                 *
               IPRESOLV=2,                                             *
               SRQTIME=15,SRQRETRY=3
*
EEGVL&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               HOSTNAME=SC&SYSCLONE.M-EE1.ITSO.IBM.COM,                *
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,TGP=TGPVRN,       *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              *
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM&SYSCLONE.,EEN&SYSCLONE.)
*
EEGVG&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               HOSTNAME=SC&SYSCLONE.M-EE2.ITSO.IBM.COM,                *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,TGP=EEXTWAN,          *
               UNRCHTIM=180,                                           *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               *
               AUTOGEN=(16,EEX&SYSCLONE.,EEY&SYSCLONE.)
```

A.1.2 Common TCP/IP files

This section includes common TCP/IP files:

- “Common TCP/IP procedure” on page 433
- “Common RESOLVER procedure” on page 433
- “Common Global Resolver TCPDATA file, GLOBAL” on page 433

- “Common OMPROUTE procedure” on page 433

Common TCP/IP procedure

Example: A-8 Common TCP/IP procedure (TCPIPA)

```
//TCPIPA    PROC PARMS='CTRACE(CTIEZB00),IDS=00',
//             PROFILE=PROFA&SYSCLONE.,TCPDATA=DATAA&SYSCLONE
//TCPIPA    EXEC PGM=EZBTCPIP,REGION=0M,TIME=1440,
//             PARM=('&PARMS',
//         'ENVAR("RESOLVER_CONFIG=//''TCPIPA.TCPPARMS(&TCPDATA)''")')
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//ALGPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CFGPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSOUT   DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSERROR DD SYSOUT=*
//*TNDBCSCN DD DSN=TCPIP.SEZAINST(TNDBCSCN),DISP=SHR
//*TNDBCSXL DD DSN=TCPIP.SEZAXLD2,DISP=SHR
//*TNDBCSER DD SYSOUT=*
//PROFILE  DD DISP=SHR,DSN=TCPIPA.TCPPARMS(&PROFILE.)
//*SYSTCPD  DD DSN=TCPIPA.TCPPARMS(&TCPDATA.),DISP=SHR
```

Common RESOLVER procedure

Example: A-9 Common Resolver procedure (RESOLVER)

```
//RESOLVER PROC PARMS='CTRACE(CTIRES00)'
//EZBREINI EXEC PGM=EZBREINI,REGION=0M,TIME=1440,PARM=&PARMS
//SETUP  DD  DSN=TCPIPA.TCPPARMS(RESOLV&SYSCLONE.),DISP=SHR,FREE=CLOSE
```

Common Global Resolver TCPDATA file, GLOBAL

Example: A-10 Common file: TCPIPA.TCPPARMS(GLOBAL)

```
SEARCH  ITSO.IBM.COM IBM.COM
DATASETPREFIX TCPIP
MESSAGECASE MIXED
NSINTERADDR  10.20.4.203
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
LOOKUP DNS
```

Common OMPROUTE procedure

Example: A-11 Common OMPROUTE procedure (OMPA)

```
//OMPA   PROC STDENV=STDENV&SYSCLONE
//OMPA   EXEC PGM=OMPROUTE,REGION=4096K,TIME=NOLIMIT,
//        PARM=('POSIX(ON) ALL31(ON)',
//            'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA"',
//            '"_CEE_ENVFILE=DD:STDENV")/')
//*           '"_CEE_ENVFILE=DD:STDENV")/-t2 -d1')
//STDENV   DD DISP=SHR,DSN=TCPIPA.OMPROUTE.&STDENV
//SYSOUT   DD SYSOUT=*
```

```
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

A.2 Configuration files for SC30 Network Node

This section includes the following configuration files for SC30:

- “SC30 VTAM files that did not change with EE”
- “SC30 VTAM files new or changed with EE”
- “SC30 TCP/IP files that did not change with EE”
- “SC30 TCP/IP files new or changed with EE”

A.2.1 SC30 VTAM files that did not change with EE

This section includes the following VTAM configuration files that did not change with implementation of EE:

- “SC30 VTAM MPC TRL major node”
- “SC30 VTAM MPC Local SNA major node”

SC30 VTAM MPC TRL major node

Example: A-12 SC30 MPC TRL major node, TRL4SC30

```
***********************************************************************
* TRANSPORT RESOURCE LIST MAJOR NODE FOR APPN CONNECTION ACROSS FCTC
***********************************************************************
TRL4SC30 VBUILD TYPE=TRL
TRL2SC31 TRLE  LNCTL=MPC,       MPC FICON TO SC31                      *
               MAXBFRU=16,                                             *
               READ=(5842,5843),                                       *
               WRITE=(4842,4843)
TRL2SC32 TRLE  LNCTL=MPC,       MPC FICON TO SC32                      *
               MAXBFRU=16,                                             *
               READ=(5852,5853),                                       *
               WRITE=(4852,4853)
```

SC30 VTAM MPC Local SNA major node

Example: A-13 SC30 MPC Local SNA PU major node, MPC4SC30

```
MPC4SC30 VBUILD TYPE=LOCAL
MPC2SC31 PU    PUTYPE=2,XID=YES,                                       *
               NETID=RDBOOKEE,CPNAME=SC31M,TGN=2,     * MPC = TG2      *
               TGP=FICON,                                              *
               TRLE=TRL2SC31
MPC2SC32 PU    PUTYPE=2,XID=YES,                                       *
               NETID=RDBOOKEE,CPNAME=SC32M,TGN=2,      * MPC = TG2     *
               TGP=FICON,                                              *
               TRLE=TRL2SC32
```

A.2.2 SC30 VTAM files new or changed with EE

This section includes the following VTAM configuration files for SC30 new or changed with implementation of EE:

- “SC30 VTAM ATCSTR30 changed for EE”
- “SC30 VTAM ATCCON30 changed for EE”
- “SC30 VTAM XCA major node (new with EE)”
- “SC30 VTAM EE SWNET major nodes, new with EE”

SC30 VTAM ATCSTR30 changed for EE

Example: A-14 SC30 ATCSTR30 changed for EE

```
NODETYPE=NN,                                                            X
BN=YES,                                                                 X
BNDYN=NONE,                                                             X
BNORD=DEFINED,                                                          X
TCPNAME=TCPIPA,                                                         X
CDSERVR=YES,                                                            X
DIALRTRY=YES,                                                           X
INITDB=DIR,                                                             X
CSALIMIT=0
```

SC30 VTAM ATCCON30 changed for EE

Example: A-15 SC30 ATCCON30 changed for EE

```
EETGPS,                     NEW TGPS FROM CS 1.8     SG24-7359          +
EECOSTAB,                   NEW APPNCOS TABLE FOR EE SG24-7359          +
MODELXCF,                   NEW XCF MODEL            SG24-7359          +
EEMODEL,                    EE MODEL                 SG24-7359          +
TRL4SC30,                   NEW TRLS TO LPARS        SG24-7359          +
MPC4SC30,                   NEW MPCS TO LPARS        SG24-7359          +
EESWNN30,                   START LINK TO OTHER LPARSSG24-7359          +
DLSWAIX,                    START DLURPU  CS/AIX     SG24-7359          +
DLSWLNX,                    START DLURPU  LINUX      SG24-7359          +
DLSWPCOM,                   START DLURPU  PCOMM      SG24-7359          +
DLSWWIN ,                   START DLURPU  CS/WIN     SG24-7359          +
EESWAIX0,                   START LINK TO CS/AIX     SG24-7359          +
EESWLNX0,                   START LINK TO CS/LINUX   SG24-7359          +
EESWWIN0,                   START LINK TO CS/WIN     SG24-7359          +
EESWPCOM,                   START LINK TO PCOMM      SG24-7359          +
EEADJCP,                    ADJACENT CP TABLE        SG24-7359          +
EEADJCLS,                   ADJACENT CLUSTER TABLE   SG24-7359          +
EECDRSC,                    CDRSC TO RESOLVE ALIAS   SG24-7359          +
EESWEB30,                   START LINKS TO EBNS      SG24-7359          +
EEXCA,                      START XCA MAJORNODE      SG24-7359          +
APECHO,                     TPNS ECHO APPLICATION                       +
TSO30,                      TSO APPLICATIONS                            +
@TCPLUS,                    TCP/IP TN3270 LUS FOR TCPIPB                +
OSA2080,                                                                +
OSA20A0,                                                                +
OSA20C0,                                                                +
OSA20E0,                                                                +
APEMSXX,                    NVAS                                        +
APNETVXX,                   NETVIEW                                     +
C1L08E0                     NON SNA LOCAL 8E0-8FF
```

SC30 VTAM XCA major node (new with EE)

Two different XCA major nodes were used and tested. One with system symbolics and host names, the other with no symbolics and IP addresses. Using symbolics enabled us to use a common XCA major node on all three systems. If symbolics are not used, then a unique XCA major node is necessary for each system.

Example: A-16 SC30 EE XCA major node, EEXCA, without symbolics

```
         VBUILD TYPE=XCA
EEPORT30 PORT MEDIUM=HPRIP,                                    *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                 *
               IPRESOLV=2,                                             *
               SRQTIME=15,SRQRETRY=3
*
EEGVL301 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.231,                                     *
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,TGP=TGPVRN,       *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              *
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM30,EEN30)
*
EEGVG301 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.232,                                     *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,TGP=EEXTWAN,          *
               UNRCHTIM=180,                                           *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               *
               AUTOGEN=(16,EEX30,EEY30)
```

SC30 VTAM EE SWNET major nodes, new with EE

This section includes the following EE SWNET major nodes for SC30:

- "SC30 VTAM SWNET pointing to SC31 and SC32"
- "SC30 VTAM SWNET pointing to CS/AIX"
- "SC30 VTAM SWNET pointing to CSWIN"
- "SC30 VTAM SWNET pointing to CSLINUX"
- "SC30 VTAM SWNET pointing to PCOMM"
- "SC30 VTAM SWNET pointing to SC47"

SC30 VTAM SWNET pointing to SC31 and SC32

Example: A-17 SC30 SWNET pointing to SC31 and SC32, EESWNN30

```
EESWNN30 VBUILD TYPE=SWNET
EEPUNN31 PU    CPNAME=SC31M,NETID=RDBOOKEE,                            *
               TGN=06,                                                 *
               TGP=HIPERSOC,                                           *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=YES,                                               *
               DWACT=YES,                                              *
               DWINOP=YES,                                             *
               ISTATUS=ACTIVE
EEPTNN31 PATH  HOSTNAME=SC31M-EE1.ITSO.IBM.COM,                        *
               SAPADDR=4,                                              *
               REDIAL=FOREVER,                                         *
               REDDELAY=30,                                            *
               GRPNM=EEGVL&SYSCLONE.1
EEPUEN32 PU    CPNAME=SC32M,NETID=RDBOOKEE,                            *
               TGN=06,                                                 *
               TGP=HIPERSOC,                                           *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=YES,                                               *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPTEN32 PATH  HOSTNAME=SC32M-EE1.ITSO.IBM.COM,                        *
               SAPADDR=4,                                              *
               GRPNM=EEGVL&SYSCLONE.1
```

SC30 VTAM SWNET pointing to CS/AIX

Example: A-18 SC30 SWNET for CS/AIX, EESWAIX

```
EESWAIX  VBUILD TYPE=SWNET
EEPUAIX PU     CPNAME=EECPAIX,NETID=RDBOOKEE,TGN=06,                   *
               CPCP=YES,HPR=YES,DISCNT=NO,DWACT=NO,DWINOP=NO,          *
               TGP=TGPCS/AIX
EEPTAIX PATH   IPADDR=10.20.4.203,                                     *
               SAPADDR=4,                                              *
               GRPNM=EEGVG&SYSCLONE.1
```

Example: A-19 SC30 SWNET for CS/AIX, DLSWAIX

```
DLSWAIX  VBUILD TYPE=SWNET
DLAIXPU1 PU    PUTYPE=2,USSTAB=USSSNAEE,                               *
               IDBLK=010,IDNUM=00001,ANS=CONT,                         *
               MODETAB=ALLMODES,DLOGMOD=DYNHIGH
DLLAIX02 LU    LOCADDR=2
DLLAIX03 LU    LOCADDR=3
DLLAIX04 LU    LOCADDR=4
DLLAIX05 LU    LOCADDR=5
DLLAIX06 LU    LOCADDR=6
```

SC30 VTAM SWNET pointing to CSWIN

Example: A-20 SC30 SWNET for CSWIN, EESWWIN

```
EESWWIN  VBUILD TYPE=SWNET
EEPUWIN PU     CPNAME=EECPWIN,NETID=RDBOOKEE,TGN=06,                   *
               CPCP=YES,HPR=YES,DISCNT=NO,DWACT=NO,DWINOP=NO,          *
               TGP=TRING16M
EEPTWIN PATH   IPADDR=10.20.4.204,                                     *
               SAPADDR=4,                                              *
               GRPNM=EEGVG&SYSCLONE.1
```

Example: A-21 SC30 SWNET for CSWIN, DLSWWIN

```
DLSWWIN  VBUILD TYPE=SWNET
DLPUWIN1 PU    PUTYPE=2,USSTAB=USSSNAEE,                               *
```

```
               IDBLK=012,IDNUM=00001,ANS=CONT,                         *
               MODETAB=ALLMODES,DLOGMOD=DYNEEHIG
DLLWIN02 LU    LOCADDR=2
DLLWIN03 LU    LOCADDR=3
DLLWIN04 LU    LOCADDR=4
DLLWIN05 LU    LOCADDR=5
DLLWIN06 LU    LOCADDR=6
```

SC30 VTAM SWNET pointing to CSLINUX

Example: A-22 SC30 SWNET for CSLINUX, EESWLNX

```
EESWLNX  VBUILD TYPE=SWNET
EEPULNX PU     CPNAME=EECPLNX,NETID=RDBOOKEE,TGN=06,                   *
               CPCP=YES,HPR=YES,DISCNT=NO,DWACT=NO,DWINOP=NO,          *
               TGP=TGPCS/AIX
EEPTLNX PATH   IPADDR=10.20.4.202,                                     *
               SAPADDR=4,                                              *
               GRPNM=EEGVG&SYSCLONE.1
```

Example: A-23 SC30 SWNET for CSLINUX, DLSWLNX

```
DLSWLNX  VBUILD TYPE=SWNET
DLLNXPU1 PU    PUTYPE=2,USSTAB=USSSNAEE,                               *
               IDBLK=011,IDNUM=00001,                                  *
               MODETAB=ALLMODES,DLOGMOD=DYNHIGH
DLLNX002 LU    LOCADDR=2
DLLNX003 LU    LOCADDR=3
DLLNX004 LU    LOCADDR=4
DLLNX005 LU    LOCADDR=5
DLLNX006 LU    LOCADDR=6
```

SC30 VTAM SWNET pointing to PCOMM

Example: A-24 SC30 SWNET for PCOMM, EESWPCOM

```
EESWPCOM VBUILD TYPE=SWNET
EEPUPCOM PU    CPNAME=EECPPCOM,NETID=RDBOOKEE,TGN=06,                  *
               CPCP=YES,                                               *
               HPR=YES,                                                *
               TGP=GIGENET,                                            *
               DISCNT=NO,                                              *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPTPCOM PATH  IPADDR=10.20.4.201,                                     *
               SAPADDR=4,                                              *
               GRPNM=EEGVG&SYSCLONE.1
```

Example: A-25 SC30 SWNET for PCOMM, DLSWPCOM

```
DLSWPCOM VBUILD TYPE=SWNET
DLPUPCOM PU    PUTYPE=2,IDBLK=013,IDNUM=00010,USSTAB=USSSNAEE,         *
               MODETAB=ALLMODES,DLOGMOD=DYNHIGH,ANS=CONT
DLPCOM02 LU    LOCADDR=2
DLPCOM03 LU    LOCADDR=3
DLPCOM04 LU    LOCADDR=4
```

```
DLPCOM05 LU    LOCADDR=5
DLPCOM06 LU    LOCADDR=6
```

SC30 VTAM SWNET pointing to SC47

Example: A-26   SC30 SWNET pointing to SC47, EESWEB30

```
EESWEB30 VBUILD TYPE=SWNET
************************************************************************
* SWITCHED MAJORNODE FOR ENTERPRISE EXTENDER TO SC47M USING            *
* PARALLEL TGS TO USE SEPERATE IP INFRASTRUCTRE                        *
*   CPCP AND SENSITIVE SESSION ONLY VIA SLOWER BUT SECURE LINK         *
*              COMMON PREFIX FOR EE PU                                 *
*              |  TGN                                                  *
*              |  |SYSCLONE_LEFT                                       *
* 10.10.1.231  |  || SYSCLONE_RIGHT                   10.1.1.131       *
*  _____   |   |  || |                                 |   _____       *
* |     |-EE1--EEP63047-TG6--(PUBLIC IP)----2M--------EE1-|     |      *
* |SC30M|                                                 |SC47M|      *
* |_____|-EE2--EEP73047-TG7--(SECURE IP)---256K-------EE2-|_____|      *
*          |                                           |               *
* 10.10.1.232                                         10.1.1.132       *
*******************************************************************
*******************************************************************
*  TG6 = FAST BUT UNSECURE LINK TO SC47M
*******************************************************************
EEP63047 PU    CPNAME=SC47M,NETID=USIBMSC,TGN=06,                      *
               VERALSID=YES,                                           *
               TGP=EEXTWAN,CAPACITY=2M,SECURITY=UNSECURE,              *
               CONNTYPE=APPN,CPCP=NO,HPR=YES,DISCNT=NO,DYNLU=YES,      *
               ISTATUS=ACTIVE,DWACT=YES,DWINOP=YES
EEPT6T47 PATH  HOSTNAME=SC47M-EE1.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=31,                             *
               GRPNM=EEGVL&SYSCLONE.1
*******************************************************************
*  TG7 = SLOW BUT SECURED LINK TO SC47M
*******************************************************************
EEP73047 PU    CPNAME=SC47M,NETID=USIBMSC,                             *
               VERALSID=YES,         * PUNAME HAS TO MATCH AT BOTH ENDS*
               TGP=EEXTWAN,CAPACITY=256K,SECURITY=ENCRYPT,             *
               CONNTYPE=APPN,CPCP=YES,HPR=YES,DISCNT=NO,DYNLU=YES,     *
               ISTATUS=ACTIVE,DWACT=YES,DWINOP=YES
EEPT7T47 PATH  HOSTNAME=SC47M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=31,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

A.2.3 SC30 TCP/IP files that did not change with EE

This section includes the following TCP/IP configuration files for SC30 that did not change with implementation of EE:

- “SC30 Resolver Setup file, no change”
- “SC30 TCP/IP TCPDATA, no change” on page 440
- “SC30 OMPROUTE Environment Variable file, no change”

SC30 Resolver Setup file, no change

Example: A-27   SC30 Resolver Setup file: TCPIPA.TCPPARMS(RESOLV30)

```
GLOBALTCPIPDATA('TCPIPA.TCPPARMS(GLOBAL)')
DEFAULTTCPIPDATA('TCPIPA.TCPPARMS(DATAA30)')
;
; GLOBALIPNODES('TCPIPA.TCPPARMS(IPNODES)')
DEFAULTIPNODES('TCPIPA.TCPPARMS(IPNODES)')
;
COMMONSEARCH
```

SC30 TCP/IP TCPDATA, no change

Example: A-28   SC30 TCP/IP TCPDATA, no change

```
TCPIPJOBNAME TCPIPA
HOSTNAME SC30M
SEARCH ITSO.IBM.COM IBM.COM
DATASETPREFIX TCPIPA
MESSAGECASE MIXED
NSINTERADDR  10.20.4.203
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
LOOKUP DNS
```

SC30 OMPROUTE Environment Variable file, no change

TCPIPA.OMPROUTE.STDENV30 is a flat sequential file with RECFM=V, not RECFM=F. RECFM=F causes padding the HFS Debug file name with blanks, resulting in a filename not found condition. Thus, use RECFM=V for the environment variable file.

Example: A-29   SC30 OMPROUTE Environment Variable control file, no change

```
RESOLVER_CONFIG=//'TCPIPA.TCPPARMS(DATAA30)'
OMPROUTE_FILE=//'TCPIPA.TCPPARMS(OMPA30)’
OMPROUTE_DEBUG_FILE=/tmp/syslog/debuga30
OMPROUTE_DEBUG_FILE_CONTROL=10000,5
```

A.2.4 SC30 TCP/IP files new or changed with EE

This section includes the following TCP/IP configuration files for SC30 new or changed with implementation of EE:

- “SC30 TCP/IP Profile, changed with EE”
- “SC30 OMPROUTE configuration file”

SC30 TCP/IP Profile, changed with EE

Example: A-30   SC30 TCP/IP Profile, changed with EE

```
;***********************************************************************
GLOBALCONFIG NOTCPIPSTATISTICS
NETMONITOR PKTTRCSERVICE
;
IPCONFIG SOURCEVIPA IGNOREREDIRECT
IPCONFIG DYNAMICXCF 10.10.20.100 255.255.255.0 1
;
UDPCONFIG UDPCHKSUM
;
TCPCONFIG TCPSENDBFRSIZE 256K TCPRCVBUFRSIZE 256K TCPMAXRCVBUFRSIZE 512K
TCPCONFIG RESTRICTLOWPORTS  SENDGARBAGE FALSE
;
AUTOLOG 5
OMPA
ENDAUTOLOG
;
PORTRANGE 12000 5 UDP NET
;
DEVICE VIPADEV  VIRTUAL 0
LINK VIPALINK VIRTUAL 0 VIPADEV
;
DEVICE EEVIPA1  VIRTUAL 0
LINK EELINK1 VIRTUAL 0 EEVIPA1
;
DEVICE EEVIPA2  VIRTUAL 0
LINK EELINK2 VIRTUAL 0 EEVIPA2
;
DEVICE OSA2080 MPCIPA NONROUTER AUTORESTART
LINK OSA2080LNK IPAQENET OSA2080
;
DEVICE OSA20A0 MPCIPA NONROUTER AUTORESTART
LINK OSA20A0LNK IPAQENET OSA20A0
;
DEVICE OSA20C0 MPCIPA NONROUTER AUTORESTART
LINK OSA20C0LNK IPAQENET OSA20C0
;
DEVICE OSA20E0 MPCIPA NONROUTER AUTORESTART
LINK OSA20E0LNK IPAQENET OSA20E0
;
HOME
  10.10.1.230    VIPALINK
  10.10.2.232    OSA2080LNK
  10.10.3.233    OSA20A0LNK
  10.10.2.234    OSA20C0LNK
  10.10.3.235    OSA20E0LNK
  10.10.1.231    EELINK1
  10.10.1.232    EELINK2
  PRIMARYINTERFACE VIPALINK
;
START OSA2080
START OSA20A0
START OSA20C0
START OSA20E0
```

SC30 OMPROUTE configuration file

Example: A-31 SC30 OMPROUTE configuration file (changed with EE)

```
Area Area_Number=0.0.0.2
Stub_Area=YES
Authentication_type=None;
```

```
OSPF RouterID=10.10.1.230;
;
; Static vipa #1
ospf_interface ip_address=10.10.1.230
           subnet_mask=255.255.255.0
           name=VIPALINK
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA2080LNK
ospf_interface ip_address=10.10.2.232
           subnet_mask=255.255.255.0
           name=OSA2080LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20A0LNK
ospf_interface ip_address=10.10.3.233
           subnet_mask=255.255.255.0
           name=OSA20A0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20C0LNK
ospf_interface ip_address=10.10.2.234
           subnet_mask=255.255.255.0
           name=OSA20C0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20E0LNK
ospf_interface ip_address=10.10.3.235
           subnet_mask=255.255.255.0
           name=OSA20E0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;
; Static vipa #2
ospf_interface ip_address=10.10.1.231
           subnet_mask=255.255.255.0
           name=EELINK1
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; Static vipa #3
ospf_interface ip_address=10.10.1.232
           subnet_mask=255.255.255.0
           name=EELINK2
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; XCF interfaces  (non-OSPF, no advertisements)
INTERFACE
     IP_Address=10.10.20.*
     Subnet_Mask=255.255.255.0
     MTU=1500;
;
AS_Boundary_routing
  Import_Direct_Routes=yes;
```

A.3 Configuration files for SC31 Network Node

This section includes the following configuration files for SC31:

- “SC31 VTAM files that did not change with EE”
- “SC31 TCP/IP files that did not change with EE”
- “SC31 VTAM files new or changed with EE”
- “SC31 TCP/IP files new or changed with EE”

A.3.1 SC31 VTAM files that did not change with EE

This section includes the following VTAM configuration files that did not change with implementation of EE:

- “SC31 VTAM MPC TRL major node”
- “SC31 VTAM MPC Local SNA major node”

SC31 VTAM MPC TRL major node

Example: A-32 SC31 MPC TRL major node, TRL4SC31

```
***********************************************************************
* TRANSPORT RESOURCE LIST MAJOR NODE FOR APPN CONNECTION ACROSS FCTC
***********************************************************************
TRL4SC31 VBUILD TYPE=TRL
TRL2SC30 TRLE  LNCTL=MPC,       MPC FICON TO SC30                      *
               MAXBFRU=16,                                             *
               LASTRW=ALLOW,                                           *
               READ=(5832,5833),                                       *
               WRITE=(4832,4833)
TRL2SC32 TRLE  LNCTL=MPC,       MPC FICON TO SC32                      *
               MAXBFRU=16,                                             *
               LASTRW=ALLOW,                                           *
               READ=(5852,5853),                                       *
               WRITE=(4852,4853)
```

SC31 VTAM MPC Local SNA major node

Example: A-33   SC31 MPC Local SNA PU major node, MPC4SC31

```
***********************************************************************
* LOCAL MAJOR NODE FOR MPC CONNETIONS TO SC30 AND SC32
***********************************************************************
MPC4SC31 VBUILD TYPE=LOCAL
MPC2SC30 PU    PUTYPE=2,XID=YES,                                       *
               NETID=RDBOOKEE,CPNAME=SC30M,TGN=2,      * MPC = TG2     *
               TGP=FICONEXP,          * NEW IBMTGPS ENTRY              *
               TRLE=TRL2SC30
MPC2SC32 PU    PUTYPE=2,XID=YES,                                       *
               NETID=RDBOOKEE,CPNAME=SC32M,TGN=2,      * MPC = TG2     *
               TGP=FICONEXP,          * NEW IBMTGPS ENTRY              *
               TRLE=TRL2SC32
```

A.3.2 SC31 VTAM files new or changed with EE

This section includes the following VTAM configuration files for SC31 new or changed with implementation of EE:

- "SC31 VTAM ATCSTR31 changed for EE"
- "SC31 VTAM ATCCON31 changed for EE"
- "SC31 VTAM EE XCA major node, new with EE"
- "SC31 VTAM EE SWNET major nodes"

SC31 VTAM ATCSTR31 changed for EE

Example: A-34   SC31 ATCSTR31 changed for EE

```
NODETYPE=NN,                                                           X
BN=YES,                                                                X
BNDYN=NONE,                                                            X
BNORD=DEFINED,                                                         X
TCPNAME=TCPIPA,                                                        X
CDSERVR=YES,                                                           X
DIALRTRY=YES,                                                          X
INITDB=DIR,                                                            X
CSALIMIT=0
```

SC31 VTAM ATCCON31 changed for EE

Example: A-35   SC31 ATCCON31 changed for EE

```
EECOSTAB,                   NEW APPNCOS TABLE FOR EE SG24-7359         +
EETGPS,                     NEW TGPS FROM CS 1.8     SG24-7359         +
MODELXCF,                   NEW XCF MODEL            SG24-7359         +
EEMODEL,                    EE MODEL                 SG24-7359         +
TRL4SC31,                   NEW TRLS FOR ADD'L MPCS TO SC30 & SC32     +
MPC4SC31,                                                              +
EESWNN31,                                                              +
DLSWAIX,                    START DLURPU  CS/AIX     SG24-7359         +
DLSWLNX,                    START DLURPU  LINUX      SG24-7359         +
DLSWPCOM,                   START DLURPU  PCOMM      SG24-7359          +
DLSWWIN ,                   START DLURPU  CS/WIN     SG24-7359          +
EESWAIX0,                   START LINK TO CS/AIX     SG24-7359          +
EESWLNX0,                   START LINK TO CS/LINUX   SG24-7359          +
EESWWIN0,                   START LINK TO CS/WIN     SG24-7359          +
EESWPCOM,                   START LINK TO PCOMM      SG24-7359          +
EEADJCP,                    ADJACENT CP TABLE        SG24-7359          +
EEADJCLS,                   ADJACENT CLUSTER TABLE   SG24-7359          +
EECDRSC,                    CDRSC TO RESOLVE ALIAS   SG24-7359          +
EESWEB31,                   START LINKS TO EBNS      SG24-7359          +
EEXCA,                                                                  +
APECHO,                     TPNS                                        +
@TCPLUS,                    TCP/IP TN3270 LUS FOR TCPIPB                +
OSA2080,                                                                +
OSA20A0,                                                                +
OSA20C0,                                                                +
OSA20E0,                                                                +
APEMSXX,                    NVAS                                        +
APNETVXX,                   NETVIEW                                     +
C1L08E0                     NON SNA LOCAL 8E0-8FF
```

SC31 VTAM EE XCA major node, new with EE

If system symbolics are used with the HOSTNAME in this major node, it can be shared by all three systems. For the contents of the EE XCA major node that uses symbolics in the HOSTNAME, see “SC30 VTAM XCA major node (new with EE)” on page 436.

Example: A-36 SC31 EE XCA major node, EEXCA with no system symbolics in HOSTNAME

```
EEXCA&SYSCLONE. VBUILD TYPE=XCA
EEPORT&SYSCLONE. PORT MEDIUM=HPRIP,                                     *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                  *
               SRQTIME=15,SRQRETRY=3
*
EEGVL&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                             *
               IPADDR=10.10.1.241,                                      *
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                   *
               TGP=EEXTCAMP,                                            *
               KEEPACT=YES,                                             *
               DYNPU=YES,                                               *
               UNRCHTIM=30,                                             *
               AUTOGEN=(16,EEM&SYSCLONE.,EEN&SYSCLONE.)
*
EEGVG&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                             *
               IPADDR=10.10.1.242,                                      *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                       *
               TGP=EEXTWAN,                                             *
               KEEPACT=YES,                                             *
               DYNPU=NO,                                                *
               UNRCHTIM=180,                                            *
               AUTOGEN=(16,EEX&SYSCLONE.,EEY&SYSCLONE.)
```

SC31 VTAM EE SWNET major nodes

This section includes the following EE SWNET major nodes for SC31:

- “SC31 VTAM SWNET pointing to SC30 and SC32”

- “SC31 VTAM SWNET pointing to SC69”

SC31 VTAM SWNET pointing to SC30 and SC32

Example: A-37 SC31 SWNET for SC30, EESWNN31

```
EESWNN31 VBUILD TYPE=SWNET
EEPUNN30 PU    CPNAME=SC30M,NETID=RDBOOKEE,                            *
               TGN=06,                                                 *
               TGP=HIPERSOC,                                           *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=YES,                                               *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPTNN30 PATH  HOSTNAME=SC30M-EE2.ITSO.IBM.COM,                        *
               SAPADDR=4,                                              *
               GRPNM=EEGVG&SYSCLONE.1
EEPUEN32 PU    CPNAME=SC32M,NETID=RDBOOKEE,                            *
               TGN=06,                                                 *
               TGP=HIPERSOC,                                           *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=YES,                                               *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPTEN32 PATH  HOSTNAME=SC32M-EE2.ITSO.IBM.COM,                        *
               SAPADDR=4,                                              *
               GRPNM=EEGVG&SYSCLONE.1
```

SC31 VTAM SWNET pointing to SC69

Example: A-38 SC31 SWNET for SC69, EESWEB31

```
EESWEB31 VBUILD TYPE=SWNET
************************************************************************
* SWITCHED MAJORNODE FOR ENTERPRISE EXTENDER TO SC31M USING            *
* PARALLEL TGS TO USE SEPERATE IP INFRASTRUCTRE                        *
*   CPCP AND SENSITIVE SESSION ONLY VIA SLOWER BUT SECURE LINK         *
*              COMMON PREFIX FOR EE PU                                 *
*              |  TGN                                                  *
*              |  |SYSCLONE_LEFT                                       *
* 10.10.1.241  |  || SYSCLONE_RIGHT                   10.10.1.141      *
*  _____   |   |  || |                                 |   _____       *
* |     |-EE1--EEP63169-TG6--(PUBLIC IP)----2M--------EE1-|     |      *
* |SC31M|                                                 |SC69M|      *
* |_____|-EE2--EEP73169-TG7--(SECURE IP)---256K-------EE2-|_____|      *
*          |                                           |               *
* 10.10.1.242                                         10.10.1.142      *
************************************************************************
*******************************************************************
*  TG6 = FAST BUT UNSECURE LINK TO SC69M
*******************************************************************
```

```
EEP63169 PU    CPNAME=SC69M,NETID=USIBMSC,TGN=06,                      *
               VERALSID=YES,                                           *
               TGP=EEXTWAN,CAPACITY=2M,SECURITY=UNSECURE,              *
               CONNTYPE=APPN,CPCP=NO,HPR=YES,DISCNT=NO,DYNLU=YES,      *
               ISTATUS=ACTIVE,DWACT=YES,DWINOP=YES
EEPT6T69 PATH  HOSTNAME=SC69M-EE1.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=31,                             *
               GRPNM=EEGVL&SYSCLONE.1
*******************************************************************
*  TG7 = SLOW BUT SECURED LINK TO SC69M
*******************************************************************
EEP73169 PU    CPNAME=SC69M,NETID=USIBMSC,TGN=07,                      *
               VERALSID=YES,         * PUNAME HAS TO MATCH AT BOTH ENDS*
               TGP=EEXTWAN,CAPACITY=256K,SECURITY=ENCRYPT,             *
               CONNTYPE=APPN,CPCP=YES,HPR=YES,DISCNT=NO,DYNLU=YES,     *
               ISTATUS=ACTIVE,DWACT=YES,DWINOP=YES
EEPT7T69 PATH  HOSTNAME=SC69M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=31,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

A.3.3 SC31 TCP/IP files that did not change with EE

This section includes the following TCP/IP configuration files for SC31 that did not change with implementation of EE:

- “SC31 Resolver Setup file, no change”
- “SC31 TCP/IP TCPDATA, no change”
- “SC31 OMPROUTE environment variable file (no change)”

SC31 Resolver Setup file, no change

Example: A-39 SC31 Resolver Setup file: TCPIPA.TCPPARMS(RESOLV31)

```
GLOBALTCPIPDATA('TCPIPA.TCPPARMS(GLOBAL)')
DEFAULTTCPIPDATA('TCPIPA.TCPPARMS(DATAA31)')
;
; GLOBALIPNODES('TCPIPA.TCPPARMS(IPNODES)')
DEFAULTIPNODES('TCPIPA.TCPPARMS(IPNODES)')
;
COMMONSEARCH
```

SC31 TCP/IP TCPDATA, no change

Example: A-40 SC31 TCP/IP TCPDATA, no change

```
TCPIPJOBNAME TCPIPA
HOSTNAME SC31M
SEARCH  ITSO.IBM.COM IBM.COM
DATASETPREFIX TCPIPA
MESSAGECASE MIXED
NSINTERADDR  10.20.4.203
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
LOOKUP DNS
```

SC31 OMPROUTE environment variable file (no change)

TCPIPA.OMPROUTE.STDENV31 is a flat sequential file with RECFM=V, not RECFM=F. RECFM=F causes padding the HFS Debuf file name with blanks, resulting in a filename not found condition. Therefore, use RECFM=V for the environment variable file.

Example: A-41 SC31 OMPROUTE Environment Variable control file

```
RESOLVER_CONFIG=//'TCPIPA.TCPPARMS(DATAA31)'
OMPROUTE_FILE=//'TCPIPA.TCPPARMS(OMPA31)'
OMPROUTE_DEBUG_FILE=/tmp/syslog/debuga31
OMPROUTE_DEBUG_FILE_CONTROL=10000,5
```

A.3.4 SC31 TCP/IP files new or changed with EE

This section includes the following TCP/IP configuration files for SC31 new or changed with EE:

- “SC31 TCP/IP Profile, changed with EE”
- “SC31 OMPROUTE Configuration file, changed with EE”

SC31 TCP/IP Profile, changed with EE

Example: A-42 SC31 TCP/IP Profile, changed with EE

```
;***********************************************************************
GLOBALCONFIG NOTCPIPSTATISTICS
NETMONITOR PKTTRCSERVICE
;
IPCONFIG SOURCEVIPA IGNOREREDIRECT
IPCONFIG DYNAMICXCF 10.10.20.101 255.255.255.0 1
;
UDPCONFIG UDPCHKSUM
;
TCPCONFIG TCPSENDBFRSIZE 256K TCPRCVBUFRSIZE 256K TCPMAXRCVBUFRSIZE 512K
TCPCONFIG RESTRICTLOWPORTS  SENDGARBAGE FALSE
;
AUTOLOG 5
OMPA
ENDAUTOLOG
;
PORTRANGE 12000 5 UDP NET
;
DEVICE VIPADEV  VIRTUAL 0
LINK VIPALINK VIRTUAL 0 VIPADEV
;
DEVICE EEVIPA1  VIRTUAL 0
LINK EELINK1 VIRTUAL 0 EEVIPA1
;
DEVICE EEVIPA2  VIRTUAL 0
LINK EELINK2 VIRTUAL 0 EEVIPA2
;
DEVICE OSA2080 MPCIPA NONROUTER AUTORESTART
LINK OSA2080LNK IPAQENET OSA2080
;
DEVICE OSA20A0 MPCIPA NONROUTER AUTORESTART
LINK OSA20A0LNK IPAQENET OSA20A0
```

```
;
DEVICE OSA20C0 MPCIPA NONROUTER AUTORESTART
LINK OSA20C0LNK IPAQENET OSA20C0
;
DEVICE OSA20E0 MPCIPA NONROUTER AUTORESTART
LINK OSA20E0LNK IPAQENET OSA20E0
;
HOME
  10.10.1.240    VIPALINK
  10.10.2.242    OSA2080LNK
  10.10.3.243    OSA20A0LNK
  10.10.2.244    OSA20C0LNK
  10.10.3.245    OSA20E0LNK
  10.10.1.241    EELINK1
  10.10.1.242    EELINK2
  PRIMARYINTERFACE VIPALINK
;
START OSA2080
START OSA20A0
START OSA20C0
START OSA20E0
```

SC31 OMPROUTE Configuration file, changed with EE

Example: A-43   SC31 OMPROUTE Configuration file, changed with EE

```
Area Area_Number=0.0.0.2
Stub_Area=YES
Authentication_type=None;
OSPF RouterID=10.10.1.240;
;
; Static vipa #1
ospf_interface ip_address=10.10.1.240
           subnet_mask=255.255.255.0
           name=VIPALINK
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA2080LNK
ospf_interface ip_address=10.10.2.242
           subnet_mask=255.255.255.0
           name=OSA2080LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20A0LNK
ospf_interface ip_address=10.10.3.243
           subnet_mask=255.255.255.0
           name=OSA20A0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20COLNK
ospf_interface ip_address=10.10.2.244
           subnet_mask=255.255.255.0
           name=OSA20COLNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20EOLNK
ospf_interface ip_address=10.10.3.245
           subnet_mask=255.255.255.0
           name=OSA20EOLNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;
; Static vipa #2
ospf_interface ip_address=10.10.1.241
           subnet_mask=255.255.255.0
           name=EELINK1
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; Static vipa #3
ospf_interface ip_address=10.10.1.242
           subnet_mask=255.255.255.0
           name=EELINK2
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; XCF interfaces  (non-OSPF, no advertisements)
INTERFACE
     IP_Address=10.10.20.*
     Subnet_Mask=255.255.255.0
     MTU=1500;
;
AS_Boundary_routing
  Import_Direct_Routes=yes;
```

A.4 Configuration files for SC32 End Node

This section includes the following configuration files for SC32:

- “SC32 VTAM files that did not change with EE”
- “SC32 TCP/IP files that did not change with EE”
- “SC32 VTAM files new or changed with EE”
- “SC32 TCP/IP files new or changed with EE”

A.4.1 SC32 VTAM files that did not change with EE

This section includes the following VTAM configuration files that did not change with implementation of EE:

- “SC32 VTAM MPC TRL major node”
- “SC32 VTAM MPC Local SNA major node”

SC32 VTAM MPC TRL major node

Example: A-44 SC32 MPC TRL major node, TRL4SC32

```
***********************************************************************
* TRANSPORT RESOURCE LIST MAJOR NODE FOR APPN CONNECTION ACROSS FCTC
***********************************************************************
TRL4SC32 VBUILD TYPE=TRL
TRL2SC30 TRLE  LNCTL=MPC,       MPC FICON TO SC30                      *
               MAXBFRU=16,                                             *
               READ=(5832,5833),                                       *
               WRITE=(4832,4833)
TRL2SC31 TRLE  LNCTL=MPC,       MPC FICON TO SC32                      *
               MAXBFRU=16,                                             *
               READ=(5842,5843),                                       *
               WRITE=(4842,4843)
```

SC32 VTAM MPC Local SNA major node

Example: A-45 SC32 MPC Local SNA PU major node, MPC4SC32

```
***********************************************************************
* LOCAL MAJOR NODE FOR MPC CONNETIONS TO SC31 AND SC32
***********************************************************************
MPC4SC32 VBUILD TYPE=LOCAL
MPC2SC31 PU    PUTYPE=2,XID=YES,                                       *
               CPCP=YES,                                               *
               CONNTYPE=APPN,                                          *
               NETID=RDBOOKEE,CPNAME=SC31M,TGN=2,     * MPC = TG2      *
               TGP=FICONEXP,          * NEW IBMTGPS ENTRY              *
               TRLE=TRL2SC31
MPC2SC30 PU    PUTYPE=2,XID=YES,                                       *
               CPCP=YES,                                               *
               CONNTYPE=APPN,                                          *
               NETID=RDBOOKEE,CPNAME=SC30M,TGN=2,     * MPC = TG2      *
               TGP=FICONEXP,          * NEW IBMTGPS ENTRY              *
               TRLE=TRL2SC30
```

A.4.2 SC32 VTAM files new or changed with EE

This section includes the following VTAM configuration files for SC32 new or changed with implementation of EE:

- “SC32 VTAM ATCSTR32 changed for EE”
- “SC32 VTAM ATCCON32 changed for EE”
- “SC32 VTAM EE XCA major node, new with EE”
- “SC32 VTAM EE SWNET major nodes, new with EE”

SC32 VTAM ATCSTR32 changed for EE

Example: A-46 SC32 ATCSTR32 changed for EE

```
NODETYPE=EN,                                                           X
NNSPREF=SC30M,                                                         X
TCPNAME=TCPIPA,                                                        X
CSALIMIT=0
```

SC32 VTAM ATCCON32 changed for EE

Example: A-47 SC32 ATCCON32 changed for EE

```
EETGPS,                     NEW TGPS FROM CS 1.8     SG24-7359         +
MODELXCF,                   NEW XCF MODEL            SG24-7359         +
EECOSTAB,                   NEW APPNCOS TABLE FOR EE SG24-7359         +
EEMODEL,                    NEW MODEL MAJORNODE      SG24-7359         +
EESWEN32,                   NEW SWITCHED MAJPRNODE   SG24-7359         +
EEXCA,                      NEW XCA MAJORNODE        SG24-7359         +
TRL4SC32,                   NEW TRLS TO LPARS        SG24-7359         +
MPC4SC32,                   NEW MPCS TO LPARS        SG24-7359         +
APECHO,                     TPNS ECHO APPLICATION                      +
@TCPLUS,                    TCP/IP TN3270 LUS FOR TCPIPB               +
OSA2080,                                                               +
OSA20A0,                                                               +
OSA20C0,                                                               +
OSA20E0,                                                               +
APEMSXX,                    NVAS                                       +
APNETVXX,                   NETVIEW                                    +
C1L08E0                     NON SNA LOCAL 8E0-8FF
```

SC32 VTAM EE XCA major node, new with EE

Example: A-48 SC32 EE XCA major node, EEXCA

```
EEXCA&SYSCLONE. VBUILD TYPE=XCA
EEPORT&SYSCLONE. PORT MEDIUM=HPRIP,                                    *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                 *
               SRQTIME=15,SRQRETRY=3
*
EEGVL&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               HOSTNAME=SC&SYSCLONE.M-EE1.ITSO.IBM.COM,                *
               VNNAME=RDBOOKEE.VRNLOCAL,VNTYPE=LOCAL,                  *
               TGP=EEXTCAMP,                                           *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              *
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM&SYSCLONE.,EEN&SYSCLONE.)
```

```
*
EEGVG&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               HOSTNAME=SC&SYSCLONE.M-EE2.ITSO.IBM.COM,                *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      *
               TGP=EEXTWAN,                                            *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               *
               UNRCHTIM=180,                                           *
               AUTOGEN=(16,EEX&SYSCLONE.,EEY&SYSCLONE.)
```

SC32 VTAM EE SWNET major nodes, new with EE

This section includes the following EE SWNET major nodes for SC32:

- “SC32 VTAM SWNET pointing to SC30 and SC31”

SC32 VTAM SWNET pointing to SC30 and SC31

Example: A-49   SC32 SWNET for SC30, EESWEN32

```
EESWEN32 VBUILD TYPE=SWNET
*******************************************************************
*  Link to primary NNS SC30M
*******************************************************************
EEPUNN30 PU    CPNAME=SC30M,NETID=RDBOOKEE,TGN=06,                     *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,DISCNT=NO,               *
               DWACT=YES,DWINOP=YES,                                   *
               TGP=HIPERSOC
EEPTNN30 PATH  HOSTNAME=SC30M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
*******************************************************************
*  Link to backup  NNS SC31M
*******************************************************************
EEPUNN31 PU    CPNAME=SC31M,NETID=RDBOOKEE,TGN=06,                     *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,DISCNT=NO,               *
               DWACT=YES,DWINOP=YES,                                   *
               TGP=HIPERSOC
EEPTNN31 PATH  HOSTNAME=SC31M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

A.4.3 SC32 TCP/IP files that did not change with EE

This section includes the following TCP/IP configuration files that did not change with implementation of EE:

- “SC32 Resolver Setup file, no change”
- “SC32 TCP/IP TCPDATA, no change” on page 454
- “SC32 OMPROUTE Environment Variable file, no change”

SC32 Resolver Setup file, no change

Example: A-50   SC32 Resolver Setup file: TCPIPA.TCPPARMS(RESOLV32)

```
GLOBALTCPIPDATA('TCPIPA.TCPPARMS(GLOBAL)')
DEFAULTTCPIPDATA('TCPIPA.TCPPARMS(DATAA32)')
;
```

```
; GLOBALIPNODES('TCPIPA.TCPPARMS(IPNODES)')
DEFAULTIPNODES('TCPIPA.TCPPARMS(IPNODES)')
;
COMMONSEARCH
```

SC32 TCP/IP TCPDATA, no change

Example: A-51   SC32 TCP/IP TCPDATA, no change

```
TCPIPJOBNAME TCPIPA
HOSTNAME SC32M
SEARCH ITSO.IBM.COM IBM.COM
DATASETPREFIX TCPIPA
MESSAGECASE MIXED
NSINTERADDR  10.20.4.203
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
LOOKUP DNS
```

SC32 OMPROUTE Environment Variable file, no change

Example: A-52   SC32 OMPROUTE Environment Variable control file, no change

```
RESOLVER_CONFIG=//'TCPIPA.TCPPARMS(DATAA32)'
OMPROUTE_FILE=//'TCPIPA.TCPPARMS(OMPA32)'
OMPROUTE_DEBUG_FILE=/tmp/syslog/debuga32
OMPROUTE_DEBUG_FILE_CONTROL=10000,5
```

A.4.4 SC32 TCP/IP files new or changed with EE

This section includes the following TCP/IP configuration files for SC32 new or changed with implementation of EE:

- “SC32 TCP/IP Profile, changed with EE” on page 454
- “SC32 OMPROUTE Configuration file, changed with EE”

SC32 TCP/IP Profile, changed with EE

Example: A-53   SC32 TCP/IP Profile, changed with EE

```
;***********************************************************************
GLOBALCONFIG NOTCPIPSTATISTICS
NETMONITOR PKTTRCSERVICE
;
IPCONFIG SOURCEVIPA IGNOREREDIRECT
IPCONFIG DYNAMICXCF 10.10.20.102 255.255.255.0 1
;
UDPCONFIG UDPCHKSUM
;
TCPCONFIG TCPSENDBFRSIZE 256K TCPRCVBUFRSIZE 256K TCPMAXRCVBUFRSIZE 512K
TCPCONFIG RESTRICTLOWPORTS  SENDGARBAGE FALSE
;
AUTOLOG 5
OMPA
```

```
ENDAUTOLOG
;
PORTRANGE 12000 5 UDP NET
;
DEVICE VIPADEV  VIRTUAL 0
LINK VIPALINK VIRTUAL 0 VIPADEV
;
DEVICE EEVIPA1  VIRTUAL 0
LINK EELINK1 VIRTUAL 0 EEVIPA1
;
DEVICE EEVIPA2  VIRTUAL 0
LINK EELINK2 VIRTUAL 0 EEVIPA2
;
DEVICE OSA2080 MPCIPA NONROUTER AUTORESTART
LINK OSA2080LNK IPAQENET OSA2080
;
DEVICE OSA20A0 MPCIPA NONROUTER AUTORESTART
LINK OSA20A0LNK IPAQENET OSA20A0
;
DEVICE OSA20C0 MPCIPA NONROUTER AUTORESTART
LINK OSA20C0LNK IPAQENET OSA20C0
;
DEVICE OSA20E0 MPCIPA NONROUTER AUTORESTART
LINK OSA20E0LNK IPAQENET OSA20E0
;
HOME
  10.10.1.220    VIPALINK
  10.10.2.222    OSA2080LNK
  10.10.3.223    OSA20A0LNK
  10.10.2.224    OSA20C0LNK
  10.10.3.225    OSA20E0LNK
  10.10.1.221    EELINK1
  10.10.1.222    EELINK2
  PRIMARYINTERFACE VIPALINK
;
START OSA2080
START OSA20A0
START OSA20C0
START OSA20E0
```

SC32 OMPROUTE Configuration file, changed with EE

Example: A-54   SC32 OMPROUTE Configuration file, changed with EE

```
Area Area_Number=0.0.0.2
Stub_Area=YES
Authentication_type=None;
OSPF RouterID=10.10.1.220;
;
; Static vipa #1
ospf_interface ip_address=10.10.1.220
           subnet_mask=255.255.255.0
           name=VIPALINK
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA2080LNK
ospf_interface ip_address=10.10.2.222
           subnet_mask=255.255.255.0
           name=OSA2080LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20A0LNK
ospf_interface ip_address=10.10.3.223
           subnet_mask=255.255.255.0
           name=OSA20A0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20C0LNK
ospf_interface ip_address=10.10.2.224
           subnet_mask=255.255.255.0
           name=OSA20C0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;          subnet=yes
;
; OSA Qdio OSA20E0LNK
ospf_interface ip_address=10.10.3.225
           subnet_mask=255.255.255.0
           name=OSA20E0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500
;
; Static vipa #2
ospf_interface ip_address=10.10.1.221
           subnet_mask=255.255.255.0
           name=EELINK1
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; Static vipa #3
ospf_interface ip_address=10.10.1.222
           subnet_mask=255.255.255.0
           name=EELINK2
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; XCF interfaces  (non-OSPF, no advertisements)
INTERFACE
     IP_Address=10.10.20.*
     Subnet_Mask=255.255.255.0
     MTU=1500;
;
AS_Boundary_routing
  Import_Direct_Routes=yes;
```

B

z/OS configuration files for NETID=USIBMSC

The lab scenarios presented in this book use two APPN NETIDs: RDBOOKEE and USIBMSC. The z/OS systems are SC47, SC69, and SC42. The source of the z/OS configuration files used in support of NETID=USIBMSC are included in this appendix.

The z/OS configuration files are organized as follows:

- “Configuration files common to all three systems”
- “Configuration files for SC47 Network Node”
- “Configuration files for SC69 Network Node”
- “Configuration files for SC42 End Node”

B.1 Configuration files common to all three systems

This section includes the following configuration files common to all three systems:

- "Common VTAM files"
- "Common TCP/IP files"

B.1.1 Common VTAM files

This section includes common VTAM files:

- "Common VTAM procedure"
- "Common EE model major node"

Common VTAM procedure

The VTAM44 PROC is a common procedure for all three systems in the sysplex, SC47, SC69, and SC42.

Example: B-1 Common VTAM procedure (VTAM44)

```
//NET    PROC
//NET     EXEC PGM=ISTINM01,REGION=6M,
//             DPRTY=(15,15),TIME=1440
//VTAMLST   DD DSN=SYS1.LOCAL.VTAMLST,DISP=SHR
//VTAMLIB   DD DSN=SYS1.LOCAL.VTAMLIB,DISP=SHR
//          DD DSN=SYS1.VTAMLIB,DISP=SHR
//NCPLOAD   DD DSN=NCPUSER.LOADLIB,DISP=SHR
//SISTCLIB  DD DSN=SYS1.SISTCLIB,DISP=SHR
//ISTCMIP   DD DSN=SYS1.SISTCMIP,DISP=SHR
//ISTASN1   DD DSN=SYS1.SISTASN1,DISP=SHR
//ACYGDMO   DD DSN=SYS1.SISTGDMO(ACYGDMO),DISP=SHR
//LDRIOTAB  DD DISP=SHR,DSN=WTSCPLX1.&SYSNAME..LDRIOTAB
```

Common EE model major node

Example: B-2 Common model major node (EEMODEL)

```
EEMODEL  VBUILD TYPE=MODEL
EEPUMODL PU    DYNTYPE=EE,DISCNT=NO,CPCP=YES,TGP=EEXTCAMP
VRNMODEL PU    DYNTYPE=VN,DISCNT=(YES,,120)
RTPMODEL PU    DYNTYPE=RTP,DISCNT=NO
```

B.1.2 Common TCP/IP files

This section includes the following common TCP/IP files for SC47, SC69, and SC42:

- "Common TCP/IP procedure"
- "Common OMPROUTE procedure"

Common TCP/IP procedure

Example: B-3 Common TCP/IP procedure (TCPIPB)

```
//TCPIPB   PROC P1='CTRACE(CTIEZB00)',
//             TCPPROF=PROFB&SYSCLONE,
//             TCPDATA=DATAB&SYSCLONE
//*
//TCPIP EXEC PGM=EZBTCPIP,REGION=0M,TIME=1440,
//   PARM=&P1
//STEPLIB   DD DSN=TCPIP.SEZATCP,DISP=SHR
//SYSPRINT  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//SYSERR    DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//SYSERROR  DD SYSOUT=*
//CEEDUMP   DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//PROFILE   DD DSN=TCPIPB.&SYSNAME..TCPPARMS(&TCPPROF.),
//             DISP=SHR,FREE=CLOSE
//SYSTCPD   DD DSN=TCPIPB.&SYSNAME..TCPPARMS(&TCPDATA.),DISP=SHR
//SYSABEND DD SYSOUT=*
```

Common OMPROUTE procedure

Example: B-4   Common OMPROUTE procedure (OMPB)

```
//OMPB   PROC OMPENV=OMPENV&SYSCLONE
//OMPB   EXEC PGM=OMPROUTE,REGION=4096K,TIME=NOLIMIT,
//        PARM=('POSIX(ON) ALL31(ON)',
//            'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPB"',
//            '"_CEE_ENVFILE=DD:STDENV")/')
//*           '"_CEE_ENVFILE=DD:STDENV")/-T2 -D1')
//*
//STDENV   DD DISP=SHR,DSN=TCPIPB.&SYSNAME..TCPPARMS(&OMPENV)
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

B.2 Configuration files for SC47 Network Node

This section includes the following configuration files for SC47:

- “SC47 VTAM files with EE”
- “SC47 TCP/IP files with EE”

B.2.1 SC47 VTAM files with EE

This section includes the following VTAM configuration files for SC47 with EE:

- “SC47 VTAM ATCSTR47 for base APPN, with EE”
- “SC47 VTAM ATCCON47 for base APPN, with EE”
- “SC47 VTAM Adjacent Cluster Table”
- “SC47 VTAM EE XCA major node”
- “SC47 VTAM EE SWNET major nodes”

SC47 VTAM ATCSTR47 for base APPN, with EE

Example: B-5   SC47 ATCSTR47 base APPN, with EE

```
CONFIG=47,                                                             X
SSCPID=47,                                                             X
NOPROMPT,                                                              X
SSCPNAME=SC47M,                                                        X
NETID=USIBMSC,                                                         X
IQDCHPID=F0,                    CHPID FOR HIPERSOCKETS                 X
NODETYPE=NN,                    NETWORK NODE                           X
APPNCOS=#INTER,                 DEFAULT APPN COS                       X
```

```
CONNTYPE=APPN,                                                         X
CPCP=YES,                                                              X
CDSERVR=YES,                    CENTRAL DIRECTORY SERVER               X
TCPNAME=TCPIPB,                 ADDED FOR EE 18 OCT 2006               X
BN=YES,                         ADDED FOR EE 04 OCT 2006               X
BNDYN=NONE,                     ADDED FOR EE 04 OCT 2006               X
BNORD=DEFINED,                  ADDED FOR EE 04 OCT 2006               X
HPR=RTP,                        ADDED FOR EE 04 OCT 2006               X
HPRARB=RESPMODE,                ADDED FOR EE 04 OCT 2006               X
XNETALS=YES,                    ADDED FOR EE 04 OCT 2006               X
IOPURGE=180,                                                           X
SUPP=NOSUP,                                                            X
HOSTPU=SC47MPU,                                                        X
PPOLOG=YES,                                                            X
DYNLU=YES,                                                             X
ENCRYPTN=NO,                    NO ENCRYPTION                          X
VERIFYCP=OPTIONAL,                                                     X
VFYREDTI=0,                                                            X
CRPLBUF=(208,,15,,1,16),                                               X
IOBUF=(182,440,19,,8,48),                                              X
LPBUF=(9,,0,,6,1)
*HOSTSA=47,        FORCE ICN
*MAXSUBA=255,      FORCE ICN
```

SC47 VTAM ATCCON47 for base APPN, with EE

Example: B-6 SC47 ATCCON47 base APPN, with EE

```
TRLTONET,                         TRL DEFINITIONS                      X
MPCTONET,                         MPC LOCAL MAJOR NODE                 X
EETGPS,                           TRANSMISSION GROUP PROFILES FOR EE   X
EEADJ47 ,                         ADJACENT CLUSTER TABLE               X
EESWEB47,                         SWNET MAJORNODE TO SC30M             X
EEXCA47,                          HPR/IP XCA MAJORNODE FOR EE          X
COSAPPN,                          DEFAULT APPN COS TABLE               X
APPNTGP,                          TRANSMISSION GROUP PROFILE FOR APPN  X
OSA2020,                          OSD GIGABIT ETHERNET                 X
OSA2080,                          ADDED FOR EE REDBOOK - LGT           X
OSA20A0,                          ADDED FOR EE REDBOOK - LGT           X
OSA20C0,                          ADDED FOR EE REDBOOK - LGT           X
OSA20E0                           ADDED FOR EE REDBOOK - LGT
```

SC47 VTAM Adjacent Cluster Table

Example: B-7 SC47 EEADJ47

```
EEADJC47 VBUILD TYPE=ADJCLUST
*******************************************************************
DEFAULT  NETWORK BNDYN=NONE
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC47M  * SNVC=1 ONLY NATIVE
NATIVE   NETWORK NETID=USIBMSC,BNDYN=NONE
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC47M  * SNVC=1 ONLY NATIVE
RDBOOKEE NETWORK BNDYN=NONE,NETID=RDBOOKEE
         NEXTCP  SNVC=2,CPNAME=RDBOOKEE.SC30M * SNVC=2 SEARCH 1 HOP
         NEXTCP  SNVC=2,CPNAME=USIBMSC.SC69M  * SNVC=2 SEARCH 1 HOP
```

SC47 VTAM EE XCA major node

Example: B-8   SC47 EE XCA major node, EEXCA47

```
EEXCA&SYSCLONE. VBUILD TYPE=XCA
EEPORT&SYSCLONE. PORT MEDIUM=HPRIP,                                    *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                 *
               SRQTIME=15,SRQRETRY=3
*
EEGVL&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.131,                                     *
               TGP=EEXTCAMP,                                           *
               VNNAME=USIBMSC.VRNLOCAL,VNTYPE=LOCAL,                   *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              *
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM&SYSCLONE.,EEN&SYSCLONE.)
*
EEGVG&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.132,                                     *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      *
               TGP=EEXTWAN,                                            *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               *
               UNRCHTIM=180,                                           *
               AUTOGEN=(16,EEX&SYSCLONE.,EEY&SYSCLONE.)
```

SC47 VTAM EE SWNET major nodes

This section includes the following EE SWNET major node for SC47:

- "SC47 VTAM SWNET pointing to SC30"

SC47 VTAM SWNET pointing to SC30

Example: B-9   SC47 SWNET for SC30, EESWEB47

```
EESWEB47 VBUILD TYPE=SWNET
************************************************************************
* SWITCHED MAJORNODE FOR ENTERPRISE EXTENDER TO SC30M USING            *
* PARALLEL TGS TO USE SEPERATE IP INFRASTRUCTRE                        *
*   CPCP AND SENSITIVE SESSION ONLY VIA SLOWER BUT SECURE LINK         *
*              COMMON PREFIX FOR EE PU                                 *
*              |  TGN                                                  *
*              |  |SYSCLONE_LEFT                                       *
* 10.10.1.231  |  || SYSCLONE_RIGHT                   10.1.1.131       *
*  _____   |   |  || |                                 |   _____       *
* |     |-EE1--EEP63047-TG6--(PUBLIC IP)----2M--------EE1-|     |      *
* |SC30M|                                                 |SC47M|      *
* |_____|-EE2--EEP73047-TG7--(SECURE IP)---256K-------EE2-|_____|      *
*          |                                           |               *
* 10.10.1.232                                         10.1.1.132       *
*******************************************************************
*  TG6 = FAST BUT UNSECURE LINK TO SC30M
*******************************************************************
EEP63047 PU    CPNAME=SC30M,NETID=RDBOOKEE,                            *
               TGN=06,VERALSID=YES,                                    *
               TGP=EEXTWAN,CAPACITY=2M,SECURITY=UNSECURE,              *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=NO,                                                *
               DYNLU=YES,                                              *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPT6T30 PATH  IPADDR=10.10.1.231,                                     *
               SAPADDR=4,                                              *
               REDIAL=0,                                               *
               GRPNM=EEGVL&SYSCLONE.1
*******************************************************************
*  TG7 = SLOW BUT SECURED LINK TO SC30M
*******************************************************************
EEP73047 PU    CPNAME=SC30M,NETID=RDBOOKEE,                            *
               TGN=07,VERALSID=YES,                                    *
               TGP=EEXTWAN,CAPACITY=256K,SECURITY=ENCRYPT,             *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=YES,                                               *
               DYNLU=YES,                                              *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPT7T30 PATH  IPADDR=10.10.1.231,                                     *
               SAPADDR=4,                                              *
               REDIAL=0,                                               *
               GRPNM=EEGVG&SYSCLONE.1
```

B.2.2 SC47 TCP/IP files with EE

This section includes the following TCP/IP configuration files for SC47 with EE:

- “SC47 TCP/IP TCPDATA with EE”
- “SC47 TCP/IP Profile with EE”
- “SC47 OMPROUTE Environment Variable control file”
- “SC47 OMPROUTE Configuration file with EE”

SC47 TCP/IP TCPDATA with EE

Example: B-10 SC47 TCP/IP TCPDATA, with EE

```
TCPIPJOBNAME TCPIPB
HOSTNAME SC47M
DOMAINORIGIN ITSO.IBM.COM
DATASETPREFIX TCPIPB
LOOKUP LOCAL DNS
MESSAGECASE MIXED
;NSINTERADDR  10.1.6.7
;NSPORTADDR 53
;RESOLVEVIA UDP
;RESOLVERTIMEOUT 10
;RESOLVERUDPRETRIES 1
```

SC47 TCP/IP Profile with EE

Example: B-11   SC47 TCP/IP Profile, with EE

```
GLOBALCONFIG NOTCPIPSTATISTICS
;
NETMONITOR PKTTRCSERVICE
;
IPCONFIG IGNOREREDIRECT
IPCONFIG SOURCEVIPA
IPCONFIG DYNAMICXCF 10.10.20.147 255.255.255.0 1
;
UDPCONFIG UDPCHKSUM
;
TCPCONFIG TCPSENDBFRSIZE 64K TCPRCVBUFRSIZE 64K SENDGARBAGE FALSE
TCPCONFIG RESTRICTLOWPORTS
TCPCONFIG RESTRICTLOWPORTS
;
AUTOLOG 5
OMPB
ENDAUTOLOG
;
;PORTRANGE
PORTRANGE 12000 5 UDP VTAM44
;
DEVICE VIPADEV  VIRTUAL 0
LINK VIPALINK VIRTUAL 0 VIPADEV
;
DEVICE EEVIPA1  VIRTUAL 0
LINK EELINK1 VIRTUAL 0 EEVIPA1
;
DEVICE EEVIPA2  VIRTUAL 0
LINK EELINK2 VIRTUAL 0 EEVIPA2
;
DEVICE OSA2080 MPCIPA NONROUTER AUTORESTART
LINK OSA2080LNK IPAQENET OSA2080
;
DEVICE OSA20A0 MPCIPA NONROUTER AUTORESTART
LINK OSA20A0LNK IPAQENET OSA20A0
;
DEVICE OSA20C0 MPCIPA NONROUTER AUTORESTART
LINK OSA20C0LNK IPAQENET OSA20C0
;
DEVICE OSA20E0 MPCIPA NONROUTER AUTORESTART
LINK OSA20E0LNK IPAQENET OSA20E0
;
HOME
  10.10.1.130    VIPALINK
  10.10.2.132    OSA2080LNK
  10.10.3.133    OSA20A0LNK
  10.10.2.134    OSA20C0LNK
  10.10.3.135    OSA20E0LNK
  10.10.1.131    EELINK1
  10.10.1.132    EELINK2
  PRIMARYINTERFACE VIPALINK
;
START OSA2080
START OSA20A0
START OSA20C0
START OSA20E0
```

SC47 OMPROUTE Environment Variable control file

Example: B-12   SC47 OMPROUTE Environment Variable control file

```
RESOLVER_CONFIG=//'TCPIPB.SC47.TCPPARMS(DATAB47)'
OMPROUTE_FILE=//'TCPIPB.SC47.TCPPARMS(OMPB47)'
OMPROUTE_OPTIONS=hello_hi
OMPROUTE_DEBUG_FILE=/tmp/syslog/debugb47
OMPROUTE_DEBUG_FILE_CONTROL=10000,5
```

SC47 OMPROUTE Configuration file with EE

Example: B-13   SC47 OMPROUTE Configuration file, with EE

```
Area Area_Number=0.0.0.2
Stub_Area=YES
Authentication_type=None;
OSPF RouterID=10.10.1.130;
;
; Static vipa #1
ospf_interface ip_address=10.10.1.130
           subnet_mask=255.255.255.0
           name=VIPALINK
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA2080LNK
ospf_interface ip_address=10.10.2.132
           subnet_mask=255.255.255.0
           name=OSA2080LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20A0LNK
ospf_interface ip_address=10.10.3.133
           subnet_mask=255.255.255.0
           name=OSA20A0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20C0LNK
ospf_interface ip_address=10.10.2.134
           subnet_mask=255.255.255.0
           name=OSA20C0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20E0LNK
ospf_interface ip_address=10.10.3.135
           subnet_mask=255.255.255.0
           name=OSA20E0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; EE vipa #1
ospf_interface ip_address=10.10.1.131
           subnet_mask=255.255.255.0
           name=EELINK1
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; EE vipa #2
ospf_interface ip_address=10.10.1.132
           subnet_mask=255.255.255.0
           name=EELINK2
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
; XCF interfaces (non-OSPF, not advertised)
INTERFACE
     IP_Address=10.10.20.*
     Subnet_Mask=255.255.255.0
     MTU=1500;
;
AS_Boundary_routing
  Import_Direct_Routes=yes;
```

B.3 Configuration files for SC69 Network Node

This section includes the following configuration files for SC69:

- “SC69 VTAM files with EE”
- “SC69 TCP/IP files with EE”

B.3.1 SC69 VTAM files with EE

This section includes the following VTAM configuration files for SC69 with EE:

- “SC69 VTAM ATCSTR69 for base APPN, with EE”
- “SC69 VTAM ATCCON69 for base APPN, with EE”
- “SC69 VTAM Adjacent Cluster Table”
- “SC69 VTAM EE XCA major node”
- “SC69 VTAM EE SWNET major nodes”

SC69 VTAM ATCSTR69 for base APPN, with EE

Example: B-14 SC69 ATCSTR69 base APPN, with EE

```
CONFIG=69,                                                             X
SSCPID=69,                                                             X
NOPROMPT,                                                              X
SSCPNAME=SC69M,                                                        X
NETID=USIBMSC,                                                         X
IQDCHPID=FC,       NEXT FO      CHPID FOR HIPERSOCKETS                 X
NODETYPE=NN,                                                           X
APPNCOS=#INTER,                 DEFAULT APPN COS                       X
CONNTYPE=APPN,                                                         X
CPCP=YES,                                                              X
CDSERVR=YES,                                                           X
TCPNAME=TCPIPB,                 ADDED FOR EE 18 OCT 2006               X
BN=YES,                         ADDED FOR EE 04 OCT 2006               X
BNDYN=NONE,                     ADDED FOR EE 04 OCT 2006               X
BNORD=DEFINED,                  ADDED FOR EE 04 OCT 2006               X
HPR=RTP,                        ADDED FOR EE 04 OCT 2006               X
HPRARB=RESPMODE,                ADDED FOR EE 04 OCT 2006               X
XNETALS=YES,                    ADDED FOR EE 04 OCT 2006               X
IOPURGE=180,                                                           X
SUPP=NOSUP,                                                            X
HOSTPU=SC69MPU,                                                        X
PPOLOG=YES,                                                            X
DYNLU=YES,                                                             X
ENCRYPTN=NO,                    NO ENCRYPTION                          X
VERIFYCP=OPTIONAL,                                                     X
VFYREDTI=0,                                                            X
CRPLBUF=(208,,15,,1,16),                                               X
IOBUF=(182,440,19,,8,48),                                              X
LPBUF=(9,,0,,6,1)
```

SC69 VTAM ATCCON69 for base APPN, with EE

Example: B-15 SC69 ATCCON69 base APPN, with EE

```
TRLTONET,                         TRL DEFINITIONS                      X
MPCTONET,                         MPC LOCAL MAJOR NODE                 X
EETGPS,                           TRANSMISSION GROUP PROFILES FOR EE   X
EEADJ69 ,                         ADJACENT CLUSTER TABLE               X
EESWEB69,                         SWNET MAJORNODE TO SC31M             X
EEXCA69,                          HPR/IP XCA MAJORNODE FOR EE          X
COSAPPN,                          DEFAULT APPN COS TABLE               X
APPNTGP,                          TRANSMISSION GROUP PROFILE FOR APPN  X
OSA2080,                          ADDED FOR EE REDBOOK - LGT           X
OSA20A0,                          ADDED FOR EE REDBOOK - LGT           X
OSA20C0,                          ADDED FOR EE REDBOOK - LGT           X
OSA20E0                           ADDED FOR EE REDBOOK - LGT
```

SC69 VTAM Adjacent Cluster Table

Example: B-16   SC69 EEADJ69

```
***********************************************************************
* RDBOOKEE      |       USIBMSC  BNDYN=NONE                           *
*  _____        |        _____                                        *
* |     |       |       |     |  DEFAULT AND NATIVE NETID: SNVC1      *
* |SC30M|______TG7______|SC47M|    1ST AND ONLY CHOICE LOCAL SSCPNAME *
* |_____|               |_____|    SNVC=1                             *
*    |          |          |     ADJACENT NETID: SNVC=2               *
*    |          |          |       1ST CHOICE  NON-NATIVE EBN         *
*  __|__        |        __|__     BACKUP NATIVE ADJ EBN              *
* |     |       |       |     |                                       *
* |SC31M|______TG7______|SC69M|                                       *
* |_____|               |_____|                                       *
*                                                                     *
EEADJC69 VBUILD TYPE=ADJCLUST         * ACTIVE ON SC69M               *
***********************************************************************
DEFAULT  NETWORK BNDYN=NONE
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC69M  * SNVC=1 ONLY NATIVE
NATIVE   NETWORK NETID=USIBMSC,BNDYN=NONE
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC69M  * SNVC=1 ONLY NATIVE
RDBOOKEE NETWORK BNDYN=NONE,NETID=RDBOOKEE
         NEXTCP  SNVC=2,CPNAME=RDBOOKEE.SC31M * SNVC=2 SEARCH 1 HOP
         NEXTCP  SNVC=1,CPNAME=USIBMSC.SC47M
```

SC69 VTAM EE XCA major node

Example: B-17   SC69 EE XCA major node, EEXCA69

```
EEXCA&SYSCLONE. VBUILD TYPE=XCA
EEPORT&SYSCLONE. PORT MEDIUM=HPRIP,                                    *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(12,24),                 *
               SRQTIME=10,SRQRETRY=3
*
EEGVL&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.141,                                     *
               TGP=EEXTCAMP,CAPACITY=100M,                             *
               VNNAME=USIBMSC.VRNLOCAL,VNTYPE=LOCAL,                   *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              *
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM&SYSCLONE.,EEN&SYSCLONE.)
*
EEGVG&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.142,                                     *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      *
               TGP=EEXTWAN,CAPACITY=2M,COSTBYTE=1,                     *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               *
               UNRCHTIM=180,                                           *
               AUTOGEN=(16,EEX&SYSCLONE.,EEY&SYSCLONE.)
```

SC69 VTAM EE SWNET major nodes

This section includes the following EE SWNET major nodes for SC69:

SC69 VTAM SWNET pointing to SC31

Example: B-18   SC69 SWNET to SC31, EESWEB69

```
EESWEB69 VBUILD TYPE=SWNET
************************************************************************
* SWITCHED MAJORNODE FOR ENTERPRISE EXTENDER TO SC31M USING            *
* PARALLEL TGS TO USE SEPERATE IP INFRASTRUCTRE                        *
*   CPCP AND SENSITIVE SESSION ONLY VIA SLOWER BUT SECURE LINK         *
*              COMMON PREFIX FOR EE PU                                 *
*              |  TGN                                                  *
*              |  |SYSCLONE_LEFT                                       *
* 10.10.1.241  |  || SYSCLONE_RIGHT                   10.10.1.141      *
*  _____   |   |  || |                                 |  _____       *
* |     |-EE1--EEP63169-TG6--(PUBLIC IP)----2M--------EE1-|     |      *
* |SC31M|                                                 |SC69M|      *
* |_____|-EE2--EEP73169-TG7--(SECURE IP)---256K-------EE2-|_____|      *
*          |                                           |               *
* 10.10.1.242                                         10.10.1.142      *
*******************************************************************
*  TG6 = FAST BUT UNSECURE LINK TO SC31M
EEP63169 PU    CPNAME=SC31M,NETID=RDBOOKEE,                            *
               TGN=06,VERALSID=YES,                                    *
               TGP=EEXTWAN,CAPACITY=2M,SECURITY=UNSECURE,              *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=NO,                                                *
               DYNLU=YES,                                              *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPT6T31 PATH  IPADDR=10.10.1.241,                                     *
               SAPADDR=4,                                              *
               REDIAL=0,                                               *
               GRPNM=EEGVL&SYSCLONE.1
*******************************************************************
*  TG7 = SLOW BUT SECURED LINK TO SC31M
*******************************************************************
EEP73169 PU    CPNAME=SC31M,NETID=RDBOOKEE,                            *
               TGN=07,VERALSID=YES,                                    *
               TGP=EEXTWAN,CAPACITY=256K,SECURITY=ENCRYPT,             *
               DISCNT=NO,                                              *
               CONNTYPE=APPN,                                          *
               HPR=YES,                                                *
               CPCP=YES,                                               *
               DYNLU=YES,                                              *
               DWACT=NO,                                               *
               DWINOP=NO,                                              *
               ISTATUS=ACTIVE
EEPT7T31 PATH  IPADDR=10.10.1.242,                                     *
               SAPADDR=4,                                              *
               REDIAL=0,                                               *
               GRPNM=EEGVG&SYSCLONE.1
```

B.3.2 SC69 TCP/IP files with EE

This section includes the following TCP/IP configuration files for SC69 with EE:

- "SC69 TCP/IP TCPDATA with EE"
- "SC69 TCP/IP Profile with EE"
- "SC69 OMPROUTE Environment Variable control file"
- "SC69 OMPROUTE Configuration file with EE"

SC69 TCP/IP TCPDATA with EE

Example: B-19 SC69 TCP/IP TCPDATA, with EE

```
TCPIPJOBNAME TCPIPB
HOSTNAME SC69M
DOMAINORIGIN ITSO.IBM.COM
DATASETPREFIX TCPIPB
LOOKUP LOCAL DNS
MESSAGECASE MIXED
;NSINTERADDR  10.1.6.7
;NSPORTADDR 53
;RESOLVEVIA UDP
;RESOLVERTIMEOUT 10
;RESOLVERUDPRETRIES 1
```

SC69 TCP/IP Profile with EE

Example: B-20 SC69 TCP/IP Profile, with EE

```
GLOBALCONFIG NOTCPIPSTATISTICS
;
NETMONITOR PKTTRCSERVICE
;
IPCONFIG IGNOREREDIRECT
IPCONFIG SOURCEVIPA
IPCONFIG DYNAMICXCF 10.10.20.169 255.255.255.0 1
;
UDPCONFIG UDPCHKSUM

TCPCONFIG TCPSENDBFRSIZE 64K TCPRCVBUFRSIZE 64K SENDGARBAGE FALSE
TCPCONFIG RESTRICTLOWPORTS
TCPCONFIG RESTRICTLOWPORTS
;
AUTOLOG 5
OMPB
ENDAUTOLOG
;
;PORTRANGE
PORTRANGE 12000 5 UDP VTAM44
;
DEVICE VIPADEV  VIRTUAL 0
LINK VIPALINK VIRTUAL 0 VIPADEV
;
DEVICE EEVIPA1  VIRTUAL 0
LINK EELINK1 VIRTUAL 0 EEVIPA1
```

```
;
DEVICE EEVIPA2  VIRTUAL 0
LINK EELINK2 VIRTUAL 0 EEVIPA2
;
DEVICE OSA2080 MPCIPA NONROUTER AUTORESTART
LINK OSA2080LNK IPAQENET OSA2080
;
DEVICE OSA20A0 MPCIPA NONROUTER AUTORESTART
LINK OSA20A0LNK IPAQENET OSA20A0
;
DEVICE OSA20C0 MPCIPA NONROUTER AUTORESTART
LINK OSA20C0LNK IPAQENET OSA20C0
;
DEVICE OSA20E0 MPCIPA NONROUTER AUTORESTART
LINK OSA20E0LNK IPAQENET OSA20E0
;
HOME
  10.10.1.140    VIPALINK
  10.10.2.142    OSA2080LNK
  10.10.3.143    OSA20A0LNK
  10.10.2.144    OSA20C0LNK
  10.10.3.145    OSA20E0LNK
  10.10.1.141    EELINK1
  10.10.1.142    EELINK2
  PRIMARYINTERFACE VIPALINK
;
START OSA2080
START OSA20A0
START OSA20C0
START OSA20E0
```

SC69 OMPROUTE Environment Variable control file

Example: B-21   SC69 OMPROUTE Environment Variable control file

```
RESOLVER_CONFIG=//'TCPIPB.SC69.TCPPARMS(DATAB69)'
OMPROUTE_FILE=//'TCPIPB.SC69.TCPPARMS(OMPB69)'
OMPROUTE_OPTIONS=hello_hi
OMPROUTE_DEBUG_FILE=/tmp/syslog/debugb69
OMPROUTE_DEBUG_FILE_CONTROL=10000,5
```

SC69 OMPROUTE Configuration file with EE

Example: B-22   SC69 OMPROUTE Configuration file, with EE

```
Area Area_Number=0.0.0.2
Stub_Area=YES
Authentication_type=None;
OSPF RouterID=10.10.1.140;
;
; Static vipa #1
ospf_interface ip_address=10.10.1.140
           subnet_mask=255.255.255.0
           name=VIPALINK
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA2080LNK
ospf_interface ip_address=10.10.2.142
           subnet_mask=255.255.255.0
           name=OSA2080LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20A0LNK
ospf_interface ip_address=10.10.3.143
           subnet_mask=255.255.255.0
           name=OSA20A0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20C0LNK
ospf_interface ip_address=10.10.2.144
           subnet_mask=255.255.255.0
           name=OSA20C0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20E0LNK
ospf_interface ip_address=10.10.3.145
           subnet_mask=255.255.255.0
           name=OSA20E0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; EE vipa #1
ospf_interface ip_address=10.10.1.141
           subnet_mask=255.255.255.0
           name=EELINK1
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; EE vipa #2
ospf_interface ip_address=10.10.1.142
           subnet_mask=255.255.255.0
           name=EELINK2
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
INTERFACE
     IP_Address=10.10.20.*
     Subnet_Mask=255.255.255.0
     MTU=1500;
;
AS_Boundary_routing
  Import_Direct_Routes=yes;
```

B.4 Configuration files for SC42 End Node

This section includes the following configuration files for SC42:

- “SC42 VTAM files with EE”
- “SC42 TCP/IP files with EE”

B.4.1 SC42 VTAM files with EE

This section includes the following VTAM configuration files for SC42 with EE:

- “SC42 VTAM ATCSTR42 for base APPN, with EE”
- “SC42 VTAM ATCCON42 for base APPN, with EE”
- “SC42 VTAM EE XCA major node”
- “SC42 VTAM EE SWNET major nodes”

SC42 VTAM ATCSTR42 for base APPN, with EE

Example: B-23 SC42 ATCSTR42 base APPN, with EE

```
CONFIG=42,                                                             X
SSCPID=42,                                                             X
NOPROMPT,                                                              X
SSCPNAME=SC42M,                                                        X
NETID=USIBMSC,                                                         X
IQDCHPID=FO,                    CHPID FOR HIPERSOCKETS                 X
NODETYPE=EN,                                                           X
TCPNAME=TCPIPB,                                                        X
CONNTYPE=APPN,                                                         X
APPNCOS=#INTER,                 DEFAULT APPN COS                       X
CPCP=YES,                                                              X
SUPP=NOSUP,                                                            X
IOPURGE=180,                                                           X
HOSTPU=SC42MPU,                                                        X
PPOLOG=YES,                                                            X
DYNLU=YES,                                                             X
ENCRYPTN=NO,                    NO ENCRYPTION                          X
CRPLBUF=(208,,15,,1,16),                                               X
IOBUF=(182,440,19,,8,48),                                              X
LPBUF=(9,,0,,6,1)
```

SC42 VTAM ATCCON42 for base APPN, with EE

Example: B-24 SC42 ATCCON42 base APPN, with EE

```
COSAPPN,                          DEFAULT APPN COS TABLE               X
```

```
EETGPS,                           TRANSMISSION GROUP PROFILES FOR EE   X
EESWEN42,                         SWNET MAJORNODE TO SC31M             X
EEXCA42,                          HPR/IP XCA MAJORNODE FOR EE          X
OSA2080,                          ADDED FOR EE REDBOOK - LGT           X
OSA20A0,                          ADDED FOR EE REDBOOK - LGT           X
OSA20C0,                          ADDED FOR EE REDBOOK - LGT           X
OSA20E0,                          ADDED FOR EE REDBOOK - LGT           X
. . . .
```

SC42 VTAM EE XCA major node

Example: B-25   SC42 EE XCA major node, EEXCA42

```
EEXCA&SYSCLONE. VBUILD TYPE=XCA
EEPORT&SYSCLONE. PORT MEDIUM=HPRIP,                                    *
               IPRESOLV=3,                                             *
               IPTOS=(20,40,80,C0,C0),LIVTIME=(10,60),                 *
               SRQTIME=15,SRQRETRY=3
*
EEGVL&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.121,                                     *
               TGP=EEXTCAMP,                                           *
               VNNAME=USIBMSC.VRNLOCAL,VNTYPE=LOCAL,                   *
               KEEPACT=YES,                                            *
               DYNPU=YES,                                              *
               UNRCHTIM=30,                                            *
               AUTOGEN=(16,EEM&SYSCLONE.,EEN&SYSCLONE.)
*
EEGVG&SYSCLONE.1 GROUP DIAL=YES,CALL=INOUT,                            *
               IPADDR=10.10.1.122,                                     *
               VNNAME=W3IBMCOM.VRN,VNTYPE=GLOBAL,                      *
               TGP=EEXTWAN,                                            *
               KEEPACT=YES,                                            *
               DYNPU=NO,                                               *
               UNRCHTIM=180,                                           *
               AUTOGEN=(16,EEX&SYSCLONE.,EEY&SYSCLONE.)
```

SC42 VTAM EE SWNET major nodes

This section includes the folowing EE SWNET major nodes for SC42:

- “SC42 VTAM SWNET pointing to SC47 and SC69”

SC42 VTAM SWNET pointing to SC47 and SC69

Example: B-26   SC42 SWNET to SC47 and SC69, EESWEN42

```
EESWEN42 VBUILD TYPE=SWNET
*******************************************************************
*  Link to primary NNS SC47M
*******************************************************************
EEPUNN47 PU    CPNAME=SC47M,NETID=RDBOOKEE,TGN=06,                     *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,DISCNT=NO,               *
               DWACT=YES,DWINOP=YES,                                   *
               TGP=HIPERSOC
EEPTNN47 PATH  HOSTNAME=SC47M-EE1.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

```
*******************************************************************
*  Link to backup  NNS SC69M
*******************************************************************
EEPUNN69 PU    CPNAME=SC69M,NETID=RDBOOKEE,TGN=06,                     *
               HPR=YES,CPCP=YES,CONNTYPE=APPN,DISCNT=NO,               *
               DWACT=YES,DWINOP=YES,                                   *
               TGP=HIPERSOC
EEPTNN69 PATH  HOSTNAME=SC69M-EE2.ITSO.IBM.COM,SAPADDR=4,              *
               REDIAL=FOREVER,REDDELAY=30,                             *
               GRPNM=EEGVG&SYSCLONE.1
```

B.4.2 SC42 TCP/IP files with EE

This section includes the following TCP/IP configuration files for SC42 with EE:

- “SC42 TCP/IP TCPDATA with EE”
- “SC42 TCP/IP Profile with EE”
- “SC42 OMPROUTE Environment Variable control file”
- “SC42 OMPROUTE Configuration file with EE”

SC42 TCP/IP TCPDATA with EE

Example: B-27 SC42 TCP/IP TCPDATA, with EE

```
TCPIPJOBNAME TCPIPB
HOSTNAME SC42M
DOMAINORIGIN ITSO.IBM.COM
DATASETPREFIX TCPIPB
LOOKUP LOCAL DNS
MESSAGECASE MIXED
;NSINTERADDR  10.1.6.7
;NSPORTADDR 53
;RESOLVEVIA UDP
;RESOLVERTIMEOUT 10
;RESOLVERUDPRETRIES 1
```

SC42 TCP/IP Profile with EE

Example: B-28 SC42 TCP/IP Profile, with EE

```
GLOBALCONFIG NOTCPIPSTATISTICS
;
NETMONITOR PKTTRCSERVICE
;
IPCONFIG IGNOREREDIRECT
IPCONFIG SOURCEVIPA
IPCONFIG DYNAMICXCF 10.10.20.142 255.255.255.0 1
;
UDPCONFIG UDPCHKSUM
;
TCPCONFIG TCPSENDBFRSIZE 64K TCPRCVBUFRSIZE 64K SENDGARBAGE FALSE
TCPCONFIG RESTRICTLOWPORTS
TCPCONFIG RESTRICTLOWPORTS
;
AUTOLOG 5
OMPB
ENDAUTOLOG
```

```
;
;PORTRANGE
PORTRANGE 12000 5 UDP VTAM44
;
DEVICE VIPADEV  VIRTUAL 0
LINK VIPALINK VIRTUAL 0 VIPADEV
;
DEVICE EEVIPA1  VIRTUAL 0
LINK EELINK1 VIRTUAL 0 EEVIPA1
;
DEVICE EEVIPA2  VIRTUAL 0
LINK EELINK2 VIRTUAL 0 EEVIPA2
;
DEVICE OSA2080 MPCIPA NONROUTER AUTORESTART
LINK OSA2080LNK IPAQENET OSA2080
;
DEVICE OSA20A0 MPCIPA NONROUTER AUTORESTART
LINK OSA20A0LNK IPAQENET OSA20A0
;
DEVICE OSA20C0 MPCIPA NONROUTER AUTORESTART
LINK OSA20C0LNK IPAQENET OSA20C0
;
DEVICE OSA20E0 MPCIPA NONROUTER AUTORESTART
LINK OSA20E0LNK IPAQENET OSA20E0
;
HOME
  10.10.1.120    VIPALINK
  10.10.2.122    OSA2080LNK
  10.10.3.123    OSA20A0LNK
  10.10.2.124    OSA20C0LNK
  10.10.3.125    OSA20E0LNK
  10.10.1.121    EELINK1
  10.10.1.122    EELINK2
  PRIMARYINTERFACE VIPALINK
;
START OSA2080
START OSA20A0
START OSA20C0
START OSA20E0
```

SC42 OMPROUTE Environment Variable control file

Example: B-29   SC42 OMPROUTE Environment Variable control file

```
RESOLVER_CONFIG=//'TCPIPB.SC42.TCPPARMS(DATAB42)'
OMPROUTE_FILE=//'TCPIPB.SC42.TCPPARMS(OMPB42)'
OMPROUTE_OPTIONS=hello_hi
OMPROUTE_DEBUG_FILE=/tmp/syslog/debugb42
OMPROUTE_DEBUG_FILE_CONTROL=10000,5
```

SC42 OMPROUTE Configuration file with EE

Example: B-30   SC42 OMPROUTE Configuration file, with EE

```
Area Area_Number=0.0.0.2
Stub_Area=YES
```

```
Authentication_type=None;
OSPF RouterID=10.10.1.120;
;
; Static vipa #1
ospf_interface ip_address=10.10.1.120
           subnet_mask=255.255.255.0
           name=VIPALINK
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA2080LNK
ospf_interface ip_address=10.10.2.122
           subnet_mask=255.255.255.0
           name=OSA2080LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20A0LNK
ospf_interface ip_address=10.10.3.123
           subnet_mask=255.255.255.0
           name=OSA20A0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20C0LNK
ospf_interface ip_address=10.10.2.124
           subnet_mask=255.255.255.0
           name=OSA20C0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; OSA Qdio OSA20E0LNK
ospf_interface ip_address=10.10.3.125
           subnet_mask=255.255.255.0
           name=OSA20E0LNK
           ROUTER_PRIORITY=0
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; EE vipa #1
ospf_interface ip_address=10.10.1.121
           subnet_mask=255.255.255.0
           name=EELINK1
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
; EE vipa #2
ospf_interface ip_address=10.10.1.122
           subnet_mask=255.255.255.0
           name=EELINK2
           Advertise_VIPA_Routes=HOST_ONLY
           attaches_to_area=0.0.0.2
           cost0=10
           mtu=1500;
;
INTERFACE
     IP_Address=10.10.20.*
     Subnet_Mask=255.255.255.0
     MTU=1500;
;
AS_Boundary_routing
  Import_Direct_Routes=yes;
```

C

Additional configuration files for NETID=RDBOOKEE

The lab scenarios presented in this book use two APPN NETIDs: RDBOOKEE and USIBMSC. The source of the configuration files used on the mid-range servers in support of NETID=RDBOOKEE are included in this appendix.

The mid-range configuration files are organized as follows:

- “Configuration files for Communications server for AIX (CS/AIX)” on page 482
- “Configuration files for CSWIN” on page 487
- “Configuration files for CSLINUX” on page 496
- “Configuration files for Personal Communications (PCOMM)” on page 502

C.1 Configuration files for Communications server for AIX (CS/AIX)

The configuration file of CS/AIX is in `/etc/sna/sna_node.cfg`.

Example: C-1 sna_node.cfg file in CS/AIX

```
[define_node_config_file]
major_version = 5
minor_version = 1
update_release = 1
revision_level = 116

[define_node]
cp_alias = EECPAIX
description = ""
fqcp_name = RDBOOKEE.EECPAIX
node_type = END_NODE
mode_to_cos_map_supp = YES
mds_supported = YES
node_id = <07100000>
max_locates = 1500
dir_cache_size = 255
max_dir_entries = 0
locate_timeout = 60
reg_with_nn = YES
reg_with_cds = YES
mds_send_alert_q_size = 100
cos_cache_size = 24
tree_cache_size = 40
tree_cache_use_limit = 40
max_tdm_nodes = 0
max_tdm_tgs = 0
max_isr_sessions = 1000
isr_sessions_upper_threshold = 900
isr_sessions_lower_threshold = 800
isr_max_ru_size = 16384
isr_rcv_pac_window = 8
store_endpt_rscvs = NO
store_isr_rscvs = NO
store_dlur_rscvs = NO
cos_table_version = VERSION_1_COS_TABLES
send_term_self = NO
disable_branch_awareness = NO
cplu_syncpt_support = NO
cplu_attributes = NONE
dlur_support = LIMITED_MULTI_SUBNET
pu_conc_support = YES
nn_rar = 128
max_ls_exception_events = 0
max_compress_level = LZ10
ms_support = NORMAL
queue_nmvts = NO
clear_initial_topology = NO
ptf_flags = NONE
```

```
[define_ip_dlc]
dlc_name = EEDLC
description = ""
initially_active = YES
udp_port_llc = 12000
udp_port_network = 12001
udp_port_high = 12002
udp_port_medium = 12003
udp_port_low = 12004
ip_precedence_llc = 6
ip_precedence_network = 6
ip_precedence_high = 4
ip_precedence_medium = 2
ip_precedence_low = 1
no_dns_lookup = NO

[define_ip_port]
port_name = EEPORT
description = ""
lsap_address = 0x04
dlc_name = EEDLC
initially_active = YES
max_rcv_btu_size = 1500
tot_link_act_lim = 4096
inb_link_act_lim = 0
out_link_act_lim = 0
implicit_ls_limit = 0
act_xid_exchange_limit = 9
nonact_xid_exchange_limit = 5
max_ifrm_rcvd = 7
target_pacing_count = 7
max_send_btu_size = 1500
implicit_cp_cp_sess_support = YES
implicit_limited_resource = NO
implicit_deact_timer = 30
implicit_uplink_to_en = NO
effect_cap = 157286400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
local_ip_interface = en2
react_timer = 30
react_timer_retry = 65535
ack_timeout = 2000
max_retry = 10
liveness_timeout = 2000
short_hold_mode = NO

[define_ip_ls]
ls_name = EELINK01
```

```
description = ""
port_name = EEPORT
adj_cp_name = RDBOOKEE.SC30M
adj_cp_type = NETWORK_NODE
max_send_btu_size = 1500
ls_attributes = SNA
cp_cp_sess_support = YES
default_nn_server = YES
lsap_address = 0x04
auto_act_supp = NO
tg_number = 0
limited_resource = NO
disable_remote_act = NO
link_deact_timer = 30
use_default_tg_chars = YES
effect_cap = 157286400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
target_pacing_count = 7
max_ifrm_rcvd = 0
conventional_lu_compression = NO
branch_link_type = NONE
adj_brnn_cp_support = ALLOWED
initially_active = YES
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
remote_ip_host = SC30M-EE2.itso.ibm.com
ack_timeout = 2000
max_retry = 10
liveness_timeout = 10000
short_hold_mode = NO

[define_ip_ls]
ls_name = EELINK02
description = ""
port_name = EEPORT
adj_cp_name = RDBOOKEE.SC31M
adj_cp_type = NETWORK_NODE
max_send_btu_size = 1500
ls_attributes = SNA
cp_cp_sess_support = YES
default_nn_server = NO
lsap_address = 0x04
auto_act_supp = NO
tg_number = 0
limited_resource = NO
disable_remote_act = NO
link_deact_timer = 30
use_default_tg_chars = YES
```

```
effect_cap = 157286400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
target_pacing_count = 7
max_ifrm_rcvd = 0
conventional_lu_compression = NO
branch_link_type = NONE
adj_brnn_cp_support = ALLOWED
initially_active = YES
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
remote_ip_host = SC31M-EE2.itso.ibm.com
ack_timeout = 2000
max_retry = 10
liveness_timeout = 10000
short_hold_mode = NO

[define_internal_pu]
pu_name = DLAIXPU1
description = ""
dlus_name = RDBOOKEE.SC30M
bkup_dlus_name = RDBOOKEE.SC31M
pu_id = <01000001>
initially_active = YES
dlus_retry_timeout = 0
dlus_retry_limit = 65535
conventional_lu_compression = NO
dddlu_offline_supported = NO

[define_cn]
fqcn_name = RDBOOKEE.VRNLOCAL
description = ""
effect_cap = 3686400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 0
user_def_parm_2 = 0
user_def_parm_3 = 0
port_name = EEPORT

[define_local_lu]
lu_alias = ILAIX01
list_name = ""
description = ""
lu_name = ILAIXL01
lu_session_limit = 0
pu_name = <0000000000000000>
```

```
nau_address = 0
default_pool = NO
syncpt_support = NO
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO

[define_mode]
mode_name = #CONNECT
description = ""
max_neg_sess_lim = 32767
plu_mode_session_limit = 2
min_conwin_src = 1
min_conloser_src = 0
auto_act = 0
receive_pacing_win = 4
max_receive_pacing_win = 0
default_ru_size = YES
max_ru_size_upp = 1024
max_ru_size_low = 0
cos_name = #CONNECT
compression = PROHIBITED
max_compress_level = NONE
max_decompress_level = NONE

[define_lu_pool]
pool_name = LUPOOL
description = ""

[define_lu_pool]
pool_name = POOL01
description = ""

[define_lu_0_to_3_range]
base_name = DLAIX
description = ""
pu_name = DLAIXPU1
min_nau = 2
max_nau = 6
lu_model = UNKNOWN
pool_name = POOL01
priority = MEDIUM
timeout = 0
sscp_id = 0
name_attributes = NONE
base_number = 2
term_method = USE_NODE_DEFAULT

[define_tn3270_access]
description = ""
default_record = YES
client_address = <DEFAULT>
```

```
{tn3270_session_data}
port_number = 2000
listen_local_address = 9.12.4.150
description = ""
lu_name = TNPOOL
printer_lu_name = ""
tn3270_support = TN3270E
allow_specific_lu = YES
ssl_enabled = NO
security_level = SSL_AUTHENTICATE_MIN
cert_key_label = ""
{tn3270_session_data}
port_number = 2323
listen_local_address = ""
description = ""
lu_name = POOL01
printer_lu_name = ""
tn3270_support = TN3270E
allow_specific_lu = YES
ssl_enabled = NO
security_level = SSL_AUTHENTICATE_MIN
cert_key_label = ""
```

C.2 Configuration files for CSWIN

In CS Windows, configuration file is saved to *.acg file.

Example: C-2 redbook.acg configuration file

```
*TSWed Oct 18 10:55:24 2006
NODE=(
     ANYNET_SUPPORT=NONE
     CP_ALIAS=EECPWIN
     DEFAULT_PREFERENCE=NATIVE
     DISCOVERY_SUPPORT=DISCOVERY_CLIENT
     DLUR_SUPPORT=MULTI_SUBNET
     FQ_CP_NAME=RDBOOKEE.EECPWIN
     GVRN_SUPPORT=0
     MAX_LOCATES=150
     MAX_LS_EXCEPTION_EVENTS=200
     NODE_ID=05D00000
     NODE_TYPE=END_NODE
     REGISTER_WITH_CDS=1
     REGISTER_WITH_NN=ALL
     SEND_TERM_SELF=0
     TP_SECURITY_BEHAVIOR=VERIFY_EVEN_IF_NOT_DEFINED
)

PORT=(
     PORT_NAME=IBMEEDLC
     ACTIVATION_DELAY_TIMER=30
     ALLOW_ABM_XID_MISMATCH=0
     DELAY_APPLICATION_RETRIES=1
     DLC_NAME=IBMEEDLC
     IMPLICIT_BRANCH_EXTENDER_LINK=0
     IMPLICIT_CP_CP_SESS_SUPPORT=1
     IMPLICIT_DEACT_TIMER=600
     IMPLICIT_DSPU_SERVICES=NONE
     IMPLICIT_HPR_SUPPORT=1
     IMPLICIT_LIMITED_RESOURCE=NO
     IMPLICIT_LINK_LVL_ERROR=0
     LINK_STATION_ROLE=NEGOTIABLE
     MAX_ACTIVATION_ATTEMPTS=0
     MAX_IFRM_RCVD=8
     MAX_RCV_BTU_SIZE=1500
     PORT_TYPE=SATF
     RETRY_LINK_ON_DISCONNECT=1
     RETRY_LINK_ON_FAILED_START=1
     RETRY_LINK_ON_FAILURE=1
     DEFAULT_TG_CHARS=(
          COST_PER_BYTE=0
          COST_PER_CONNECT_TIME=0
          EFFECTIVE_CAPACITY=133
          PROPAGATION_DELAY=LAN
          SECURITY=NONSECURE
          USER_DEFINED_1=0
          USER_DEFINED_2=0
          USER_DEFINED_3=0
     )
     PORT_OEM_SPECIFIC_DATA=(
          OEM_LINK_DATA=(
               OEM_DATA=010000000000000000000000030000000F00000000000000
               OEM_DATA=0A0000000000000000
          )
          OEM_PORT_DEFAULTS=(
               COST_PER_CONNECT_TIME=0
               EFFECTIVE_CAPACITY=133
               INB_LINK_ACT_LIM=128
               OUT_LINK_ACT_LIM=127
               PROPAGATION_DELAY=LAN
               SECURITY=NONSECURE
               TOT_LINK_ACT_LIM=255
          )
     )
)

LINK_STATION=(
     LS_NAME=EELINK01
     ACTIVATE_AT_STARTUP=1
     ACTIVATION_DELAY_TIMER=-1
     ADJACENT_BRANCH_EXTENDER_NODE=PROHIBITED
     ADJACENT_NODE_TYPE=NETWORK_NODE
     AUTO_ACTIVATE_SUPPORT=0
     BRANCH_EXTENDER_LINK=1
     CP_CP_SESS_SUPPORT=1
     DEFAULT_NN_SERVER=1
     DELAY_APPLICATION_RETRIES=0
     DEPENDENT_LU_COMPRESSION=0
     DEPENDENT_LU_ENCRYPTION=OPTIONAL
```

```
    DEST_ADDRESS=044000
    DISABLE_REMOTE_ACT=0
    DSPU_SERVICES=NONE
    FQ_ADJACENT_CP_NAME=RDBOOKEE.SC30M
    HPR_LINK_LVL_ERROR=0
    HPR_SUPPORT=1
    INHERIT_PORT_RETRY_PARMS=1
    LIMITED_RESOURCE=NO
    LINK_DEACT_TIMER=600
    LINK_STATION_ROLE=NEGOTIABLE
    MAX_ACTIVATION_ATTEMPTS=-1
    MAX_IFRM_RCVD=7
    MAX_SEND_BTU_SIZE=1500
    NODE_ID=05D00000
    NULL_ADDRESS_MEANING=USE_WILDCARD
    PORT_NAME=IBMEEDLC
    PU_NAME=EELINK01
    RETRY_LINK_ON_DISCONNECT=0
    RETRY_LINK_ON_FAILED_START=0
    RETRY_LINK_ON_FAILURE=0
    REVERSE_ADDRESS_BYTES=0
    SOLICIT_SSCP_SESSION=0
    TG_NUMBER=0
    USE_DEFAULT_TG_CHARS=1
    USE_PU_NAME_IN_XID=0
    LINK_STATION_OEM_SPECIFIC_DATA=(
        OEM_LINK_DATA=(
            OEM_DATA=010000000400000004000000030000000F00000000000000
            OEM_DATA=0A00000016000000534333304D2D4545322E6974736F2E69
            OEM_DATA=626D2E636F6D00
        )
    )
    TG_CHARS=(
        COST_PER_BYTE=0
        COST_PER_CONNECT_TIME=0
        EFFECTIVE_CAPACITY=0
        PROPAGATION_DELAY=MINIMUM
        SECURITY=
        USER_DEFINED_1=0
        USER_DEFINED_2=0
        USER_DEFINED_3=0
    )
)

LINK_STATION=(
    LS_NAME=EELINK02
    ACTIVATE_AT_STARTUP=1
    ACTIVATION_DELAY_TIMER=-1
    ADJACENT_BRANCH_EXTENDER_NODE=PROHIBITED
    ADJACENT_NODE_TYPE=NETWORK_NODE
    AUTO_ACTIVATE_SUPPORT=0
    BRANCH_EXTENDER_LINK=1
    CP_CP_SESS_SUPPORT=1
    DEFAULT_NN_SERVER=0
    DELAY_APPLICATION_RETRIES=0
```

```
        DEPENDENT_LU_COMPRESSION=0
        DEPENDENT_LU_ENCRYPTION=OPTIONAL
        DEST_ADDRESS=044001
        DISABLE_REMOTE_ACT=0
        DSPU_SERVICES=NONE
        FQ_ADJACENT_CP_NAME=RDBOOKEE.SC31M
        HPR_LINK_LVL_ERROR=0
        HPR_SUPPORT=1
        INHERIT_PORT_RETRY_PARMS=1
        LIMITED_RESOURCE=NO
        LINK_DEACT_TIMER=600
        LINK_STATION_ROLE=NEGOTIABLE
        MAX_ACTIVATION_ATTEMPTS=-1
        MAX_IFRM_RCVD=7
        MAX_SEND_BTU_SIZE=1500
        NODE_ID=05D00000
        NULL_ADDRESS_MEANING=USE_WILDCARD
        PORT_NAME=IBMEEDLC
        PU_NAME=EELINK02
        RETRY_LINK_ON_DISCONNECT=0
        RETRY_LINK_ON_FAILED_START=0
        RETRY_LINK_ON_FAILURE=0
        REVERSE_ADDRESS_BYTES=0
        SOLICIT_SSCP_SESSION=0
        TG_NUMBER=0
        USE_DEFAULT_TG_CHARS=1
        USE_PU_NAME_IN_XID=0
        LINK_STATION_OEM_SPECIFIC_DATA=(
             OEM_LINK_DATA=(
                  OEM_DATA=01000000040000000400000003000000F0000000000000000
                  OEM_DATA=0A0000001600000053433331 4D2D4545322E6974736F2E69
                  OEM_DATA=626D2E636F6D00
             )
        )
        TG_CHARS=(
             COST_PER_BYTE=0
             COST_PER_CONNECT_TIME=0
             EFFECTIVE_CAPACITY=0
             PROPAGATION_DELAY=MINIMUM
             SECURITY=
             USER_DEFINED_1=0
             USER_DEFINED_2=0
             USER_DEFINED_3=0
        )
   )

   INTERNAL_PU=(
        PU_NAME=DLPUWIN1
        BKUP_DLUS_NAME=RDBOOKEE.SC31M
        DEPENDENT_LU_COMPRESSION=0
        DEPENDENT_LU_ENCRYPTION=OPTIONAL
        FQ_DLUS_NAME=RDBOOKEE.SC30M
        NODE_ID=01200001
        STARTUP=1
   )
```

```
DLUR_DEFAULTS=(
     DEFAULT_PU_NAME=EECPWIN
     DLUS_RETRY_LIMIT=3
     DLUS_RETRY_TIMEOUT=5
)

LU_0_TO_3=(
     LU_NAME=DLWIN002
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=2
     POOL_NAME=PUBLIC
     PRIORITY=HIGH
     PU_NAME=DLPUWIN1
)

LU_0_TO_3=(
     LU_NAME=DLWIN003
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=3
     POOL_NAME=PUBLIC
     PRIORITY=HIGH
     PU_NAME=DLPUWIN1
)

LU_0_TO_3=(
     LU_NAME=DLWIN004
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=4
     POOL_NAME=PUBLIC
     PRIORITY=HIGH
     PU_NAME=DLPUWIN1
)

LU_0_TO_3=(
     LU_NAME=DLWIN005
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=5
     POOL_NAME=PUBLIC
     PRIORITY=HIGH
     PU_NAME=DLPUWIN1
)

LU_0_TO_3=(
     LU_NAME=DLWIN006
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=6
     POOL_NAME=PUBLIC
     PRIORITY=HIGH
     PU_NAME=DLPUWIN1
)

LOCAL_LU=(
     LU_NAME=ILWIN01
     DEFAULT_POOL=0
     LU_ALIAS=ILWIN01
     LU_SESSION_LIMIT=0
     NAU_ADDRESS=0
     ROUTE_TO_CLIENT=0
     SYNCPT_SUPPORT=0
)

MODE=(
     MODE_NAME=BLANK
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=8192
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=4096
     PLU_MODE_SESSION_LIMIT=8192
     RECEIVE_PACING_WINDOW=3
)

MODE=(
     MODE_NAME=#BATCH
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#BATCH
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=0
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=8192
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=4096
     PLU_MODE_SESSION_LIMIT=8192
     RECEIVE_PACING_WINDOW=20
)

MODE=(
     MODE_NAME=#BATCHC
     AUTO_ACT=0
     COMPRESSION=REQUESTED
     COS_NAME=#BATCH
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=0
     MAX_INCOMING_COMPRESSION_LEVEL=LZ9
     MAX_NEGOTIABLE_SESSION_LIMIT=8192
     MAX_OUTGOING_COMPRESSION_LEVEL=LZ9
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=4096
     PLU_MODE_SESSION_LIMIT=8192
     RECEIVE_PACING_WINDOW=20
)

MODE=(
     MODE_NAME=#BATCHCS
     AUTO_ACT=0
     COMPRESSION=REQUESTED
     COS_NAME=#BATCHSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=LZ9
     MAX_NEGOTIABLE_SESSION_LIMIT=8
     MAX_OUTGOING_COMPRESSION_LEVEL=LZ9
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=4
     PLU_MODE_SESSION_LIMIT=8
     RECEIVE_PACING_WINDOW=3
)

MODE=(
     MODE_NAME=#BATCHSC
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#BATCHSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=8
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=4
     PLU_MODE_SESSION_LIMIT=8
     RECEIVE_PACING_WINDOW=3
)

MODE=(
     MODE_NAME=#CONNECT
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=128
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=4096
     MIN_CONWINNERS_SOURCE=16
     PLU_MODE_SESSION_LIMIT=32
     RECEIVE_PACING_WINDOW=1
)

MODE=(
     MODE_NAME=#INTER
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#INTER
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=8192
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=4096
     MIN_CONWINNERS_SOURCE=4096
     PLU_MODE_SESSION_LIMIT=8192
     RECEIVE_PACING_WINDOW=20
)

MODE=(
     MODE_NAME=#INTERC
     AUTO_ACT=0
     COMPRESSION=REQUESTED
     COS_NAME=#INTER
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=LZ9
     MAX_NEGOTIABLE_SESSION_LIMIT=8192
     MAX_OUTGOING_COMPRESSION_LEVEL=LZ9
     MAX_RU_SIZE_UPPER_BOUND=4096
     MIN_CONWINNERS_SOURCE=4096
     PLU_MODE_SESSION_LIMIT=8192
     RECEIVE_PACING_WINDOW=20
)

MODE=(
     MODE_NAME=#INTERCS
     AUTO_ACT=0
     COMPRESSION=REQUESTED
     COS_NAME=#INTERSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=LZ9
     MAX_NEGOTIABLE_SESSION_LIMIT=8
     MAX_OUTGOING_COMPRESSION_LEVEL=LZ9
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=4
     PLU_MODE_SESSION_LIMIT=8
     RECEIVE_PACING_WINDOW=7
)

MODE=(
     MODE_NAME=#INTERSC
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#INTERSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=8
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=4
     PLU_MODE_SESSION_LIMIT=8
     RECEIVE_PACING_WINDOW=7
)
```

```
MODE=(
     MODE_NAME=QPCSUPP
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=1024
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=512
     PLU_MODE_SESSION_LIMIT=1024
     RECEIVE_PACING_WINDOW=2
)

MODE=(
     MODE_NAME=QSERVER
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=64
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=0
     PLU_MODE_SESSION_LIMIT=64
     RECEIVE_PACING_WINDOW=7
)

MODE=(
     MODE_NAME=SNASVCMG
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=SNASVCMG
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=0
     MAX_INCOMING_COMPRESSION_LEVEL=NONE
     MAX_NEGOTIABLE_SESSION_LIMIT=2
     MAX_OUTGOING_COMPRESSION_LEVEL=NONE
     MAX_RU_SIZE_UPPER_BOUND=512
     MIN_CONWINNERS_SOURCE=1
     PLU_MODE_SESSION_LIMIT=2
     RECEIVE_PACING_WINDOW=1
)

TP=(
     TP_NAME=APINGD
     API_CLIENT_USE=0
     CONVERSATION_TYPE=EITHER
     DUPLEX_SUPPORT=EITHER_DUPLEX
     DYNAMIC_LOAD=1
     INCOMING_ALLOCATE_TIMEOUT=30
     LOAD_TYPE=0
     PATHNAME=C:\IBMCS\apingd.exe
     PIP_ALLOWED=1
     QUEUED=0
     RECEIVE_ALLOCATE_TIMEOUT=3600
     SECURITY_RQD=0
     SYNC_LEVEL=EITHER
     TP_INSTANCE_LIMIT=0
     TP_NAME_FORMAT=0
)

CONNECTION_NETWORK=(
     FQCN_NAME=RDBOOKEE.VRNLOCAL
     PORT_NAME=IBMEEDLC
     INHERIT_PORT_LIMITED_RESOURCE=NO
)

SPLIT_STACK=(
     POOL_NAME=<None>
     STARTUP=1
)

SHARED_FOLDERS=(
     EXTENSION_LIST=(
     )
     CACHE_SIZE=256
)

LOAD_BALANCING=(
     ADVERTISE_FREQUENCY=1
     APPC_LU_LOAD_FACTOR=0
     DEFAULT_MAX_LU62_SESSIONS=512
     ENABLE_LOAD_BALANCING=0
     HOST_LU_LOAD_FACTOR=0
     LOAD_VARIANCE=3
)

VERIFY=(
     CFG_MODIFICATION_LEVEL=12
     CFG_VERSION_LEVEL=1
     CFG_LAST_SCENARIO=14
)
```

C.3 Configuration files for CSLINUX

The Configuration file of CS Linux is in /etc/opt/ibm/sna/sna_node.cfg.

Example: C-3 sna_node.cfg file in CS Linux

```
[define_node_config_file]
major_version = 5
minor_version = 1
update_release = 1
```

```
revision_level = 35

[define_node]
cp_alias = EECPLNX
description = ""
fqcp_name = RDBOOKEE.EECPLNX
node_type = BRANCH_NETWORK_NODE
mode_to_cos_map_supp = YES
mds_supported = YES
node_id = <07100000>
max_locates = 1500
dir_cache_size = 255
max_dir_entries = 0
locate_timeout = 60
reg_with_nn = YES
reg_with_cds = YES
mds_send_alert_q_size = 100
cos_cache_size = 24
tree_cache_size = 40
tree_cache_use_limit = 40
max_tdm_nodes = 0
max_tdm_tgs = 0
max_isr_sessions = 1000
isr_sessions_upper_threshold = 900
isr_sessions_lower_threshold = 800
isr_max_ru_size = 16384
isr_rcv_pac_window = 8
store_endpt_rscvs = NO
store_isr_rscvs = NO
store_dlur_rscvs = NO
cos_table_version = VERSION_0_COS_TABLES
send_term_self = NO
disable_branch_awareness = NO
cplu_syncpt_support = NO
cplu_attributes = NONE
dlur_support = LIMITED_MULTI_SUBNET
pu_conc_support = YES
nn_rar = 128
max_ls_exception_events = 0
max_compress_level = LZ10
clear_initial_topology = NO
ptf_flags = NONE

[define_ip_dlc]
dlc_name = IP0
description = ""
initially_active = NO
udp_port_llc = 12000
udp_port_network = 12001
udp_port_high = 12002
udp_port_medium = 12003
udp_port_low = 12004
ip_precedence_llc = 6
ip_precedence_network = 6
ip_precedence_high = 4
```

```
ip_precedence_medium = 2
ip_precedence_low = 1
no_dns_lookup = NO

[define_ip_port]
port_name = EEPORT01
description = ""
lsap_address = 0x04
dlc_name = IP0
initially_active = YES
max_rcv_btu_size = 1500
tot_link_act_lim = 4096
inb_link_act_lim = 0
out_link_act_lim = 0
implicit_ls_limit = 0
act_xid_exchange_limit = 9
nonact_xid_exchange_limit = 5
max_ifrm_rcvd = 7
target_pacing_count = 7
max_send_btu_size = 1500
implicit_cp_cp_sess_support = YES
implicit_limited_resource = NO
implicit_deact_timer = 30
implicit_uplink_to_en = NO
effect_cap = 157286400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
local_ip_interface = eth0
react_timer = 30
react_timer_retry = 65535
ack_timeout = 10000
max_retry = 10
liveness_timeout = 10000
short_hold_mode = NO

[define_ip_ls]
ls_name = EELINK01
description = ""
port_name = EEPORT01
adj_cp_name = RDBOOKEE.SC30M
adj_cp_type = NETWORK_NODE
max_send_btu_size = 1500
ls_attributes = SNA
cp_cp_sess_support = YES
default_nn_server = YES
lsap_address = 0x04
auto_act_supp = NO
tg_number = 0
limited_resource = NO
disable_remote_act = NO
```

```
link_deact_timer = 30
use_default_tg_chars = YES
effect_cap = 157286400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
target_pacing_count = 7
max_ifrm_rcvd = 0
conventional_lu_compression = NO
branch_link_type = UPLINK
adj_brnn_cp_support = ALLOWED
initially_active = YES
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
remote_ip_host = SC30M-EE2.itso.ibm.com
ack_timeout = 10000
max_retry = 10
liveness_timeout = 10000
short_hold_mode = NO

[define_ip_ls]
ls_name = EELINK02
description = ""
port_name = EEPORT01
adj_cp_name = RDBOOKEE.SC31M
adj_cp_type = NETWORK_NODE
max_send_btu_size = 1500
ls_attributes = SNA
cp_cp_sess_support = YES
default_nn_server = NO
lsap_address = 0x04
auto_act_supp = NO
tg_number = 0
limited_resource = NO
disable_remote_act = NO
link_deact_timer = 30
use_default_tg_chars = YES
effect_cap = 157286400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 128
user_def_parm_2 = 128
user_def_parm_3 = 128
target_pacing_count = 7
max_ifrm_rcvd = 0
conventional_lu_compression = NO
branch_link_type = UPLINK
adj_brnn_cp_support = ALLOWED
```

```
initially_active = YES
restart_on_normal_deact = NO
react_timer = 30
react_timer_retry = 65535
remote_ip_host = SC31M-EE2.itso.ibm.com
ack_timeout = 10000
max_retry = 10
liveness_timeout = 10000
short_hold_mode = NO

[define_internal_pu]
pu_name = DLLNXPU1
description = ""
dlus_name = RDBOOKEE.SC30M
bkup_dlus_name = RDBOOKEE.SC31M
pu_id = <01100001>
initially_active = YES
dlus_retry_timeout = 0
dlus_retry_limit = 65535
conventional_lu_compression = NO
dddlu_offline_supported = NO

[define_cn]
fqcn_name = RDBOOKEE.VRNLOCAL
description = ""
effect_cap = 3686400
connect_cost = 0
byte_cost = 0
security = SEC_NONSECURE
prop_delay = PROP_DELAY_LAN
user_def_parm_1 = 0
user_def_parm_2 = 0
user_def_parm_3 = 0
port_name = EEPORT01

[define_local_lu]
lu_alias = ILLNX01
list_name = ""
description = ""
lu_name = ILLNX01
lu_session_limit = 0
pu_name = <0000000000000000>
nau_address = 0
default_pool = NO
syncpt_support = NO
lu_attributes = NONE
sscp_id = 0
disable = NO
sys_name = ""
timeout = 60
back_level = NO

[define_mode]
mode_name = #CONNECT
description = ""
```

```
max_neg_sess_lim = 32767
plu_mode_session_limit = 2
min_conwin_src = 1
min_conloser_src = 0
auto_act = 0
receive_pacing_win = 4
max_receive_pacing_win = 0
default_ru_size = YES
max_ru_size_upp = 1024
max_ru_size_low = 0
cos_name = #CONNECT
compression = PROHIBITED
max_compress_level = NONE
max_decompress_level = NONE

[define_lu_pool]
pool_name = POOL01
description = ""

[define_lu_0_to_3_range]
base_name = DLLNX
description = ""
pu_name = DLLNXPU1
min_nau = 2
max_nau = 6
lu_model = UNKNOWN
pool_name = POOL01
priority = MEDIUM
timeout = 0
sscp_id = 0
name_attributes = USE_BASE_NUMBER
base_number = 2
term_method = USE_NODE_DEFAULT

[define_tn3270_access]
description = ""
default_record = NO
client_address = <DEFAULT>
{tn3270_session_data}
port_number = 2323
listen_local_address = ""
description = ""
lu_name = POOL01
printer_lu_name = ""
tn3270_support = TN3270E
allow_specific_lu = YES
ssl_enabled = NO
security_level = SSL_AUTHENTICATE_MIN
cert_key_label = ""

[define_tn3270_defaults]
force_responses = NO
keepalive_method = NONE
keepalive_interval = 600
```

C.4 Configuration files for Personal Communications (PCOMM)

In PCOMM, configuration file is saved to *.acg file. We have defined two configurations, one for quick 3270 connection and one for full definition. Example C-4 shows the configuration file of a quick 3270 connection.

Example: C-4 Configuration file for quick 3270 connection

```
*TSMon Oct 23 10:21:04 2006
NODE=(
     ANYNET_SUPPORT=NONE
     CP_ALIAS=EECPPCOM
     DEFAULT_PREFERENCE=NATIVE
     DISCOVERY_SUPPORT=NO
     DLUR_SUPPORT=MULTI_SUBNET
     FQ_CP_NAME=RDBOOKEE.EECPPCOM
     GVRN_SUPPORT=1
     NODE_ID=05D00000
     NODE_TYPE=END_NODE
     REGISTER_WITH_CDS=1
     REGISTER_WITH_NN=1
     SEND_TERM_SELF=0
)

PORT=(
     PORT_NAME=IBMEEDLC
     ACTIVATION_DELAY_TIMER=30
     DELAY_APPLICATION_RETRIES=1
     DLC_NAME=IBMEEDLC
     IMPLICIT_CP_CP_SESS_SUPPORT=1
     IMPLICIT_DEACT_TIMER=600
     IMPLICIT_DSPU_SERVICES=NONE
     IMPLICIT_HPR_SUPPORT=1
     IMPLICIT_LIMITED_RESOURCE=NO
     IMPLICIT_LINK_LVL_ERROR=0
     LINK_STATION_ROLE=NEGOTIABLE
     MAX_ACTIVATION_ATTEMPTS=0
     MAX_IFRM_RCVD=8
     MAX_RCV_BTU_SIZE=1461
     PORT_TYPE=SATF
     RETRY_LINK_ON_DISCONNECT=1
     RETRY_LINK_ON_FAILED_START=1
     RETRY_LINK_ON_FAILURE=1
     DEFAULT_TG_CHARS=(
          COST_PER_BYTE=0
          COST_PER_CONNECT_TIME=0
          EFFECTIVE_CAPACITY=160
          PROPAGATION_DELAY=LAN
          SECURITY=NONSECURE
          USER_DEFINED_1=0
          USER_DEFINED_2=0
          USER_DEFINED_3=0
     )
     PORT_OEM_SPECIFIC_DATA=(
          OEM_LINK_DATA=(
               OEM_DATA=0100000000000000000000000300000000F00000000000000
               OEM_DATA=0A00000000000000
          )
          OEM_PORT_DEFAULTS=(
               COST_PER_CONNECT_TIME=0
               EFFECTIVE_CAPACITY=160
               INB_LINK_ACT_LIM=128
               OUT_LINK_ACT_LIM=127
               PROPAGATION_DELAY=LAN
               SECURITY=NONSECURE
               TOT_LINK_ACT_LIM=255
          )
     )
)

LINK_STATION=(
     LS_NAME=EELINK01
     ACTIVATE_AT_STARTUP=0
     ACTIVATION_DELAY_TIMER=-1
     ADJACENT_NODE_TYPE=LEARN
     AUTO_ACTIVATE_SUPPORT=0
     CP_CP_SESS_SUPPORT=1
     DEFAULT_NN_SERVER=0
     DELAY_APPLICATION_RETRIES=1
     DEPENDENT_LU_ENCRYPTION=NONE
     DEST_ADDRESS=044000
     DISABLE_REMOTE_ACT=1
     DSPU_SERVICES=NONE
     ETHERNET_FORMAT=0
     HPR_LINK_LVL_ERROR=0
     HPR_SUPPORT=1
     INHERIT_PORT_RETRY_PARMS=0
     LIMITED_RESOURCE=NO
     LINK_DEACT_TIMER=600
     LINK_STATION_ROLE=NEGOTIABLE
     MAX_ACTIVATION_ATTEMPTS=-1
     MAX_IFRM_RCVD=7
     MAX_SEND_BTU_SIZE=1461
     NODE_ID=05D00000
     PORT_NAME=IBMEEDLC
     RETRY_LINK_ON_DISCONNECT=1
     RETRY_LINK_ON_FAILED_START=1
     RETRY_LINK_ON_FAILURE=1
     SOLICIT_SSCP_SESSION=0
     TG_NUMBER=0
     USE_DEFAULT_TG_CHARS=1
     LINK_STATION_OEM_SPECIFIC_DATA=(
          OEM_LINK_DATA=(
               OEM_DATA=010000000400000004000000030000000F0000000000000000
               OEM_DATA=0A0000001600000053433304D2D4545312E6974736F2E69
               OEM_DATA=626D2E636F6D00
          )
     )
     TG_CHARS=(
          COST_PER_BYTE=0
          COST_PER_CONNECT_TIME=0
          EFFECTIVE_CAPACITY=0
          PROPAGATION_DELAY=MINIMUM
          SECURITY=
          USER_DEFINED_1=0
          USER_DEFINED_2=0
          USER_DEFINED_3=0
     )
)

INTERNAL_PU=(
     PU_NAME=DLPUPCOM
     BKUP_DLUS_NAME=RDBOOKEE.SC31M
     DEPENDENT_LU_ENCRYPTION=NONE
     FQ_DLUS_NAME=RDBOOKEE.SC30M
     NODE_ID=01300010
     STARTUP=0
)

MODE=(
     MODE_NAME=BLANK
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=3
)

MODE=(
     MODE_NAME=#BATCH
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#BATCH
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=0
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=20
)

MODE=(
     MODE_NAME=#BATCHSC
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#BATCHSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=3
)

MODE=(
     MODE_NAME=#INTER
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#INTER
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=4096
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=20
)

MODE=(
     MODE_NAME=#INTERSC
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#INTERSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=7
)

MODE=(
     MODE_NAME=QPCSUPP
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=52
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=26
     PLU_MODE_SESSION_LIMIT=52
     RECEIVE_PACING_WINDOW=2
)

MODE=(
     MODE_NAME=QSERVER
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=64
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=0
     PLU_MODE_SESSION_LIMIT=64
     RECEIVE_PACING_WINDOW=7
)

MODE=(
     MODE_NAME=SNASVCMG
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=SNASVCMG
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=0
     MAX_NEGOTIABLE_SESSION_LIMIT=2
     MAX_RU_SIZE_UPPER_BOUND=512
     MIN_CONWINNERS_SOURCE=1
     PLU_MODE_SESSION_LIMIT=2
     RECEIVE_PACING_WINDOW=1
)

SHARED_FOLDERS=(
     CACHE_SIZE=256
)

VERIFY=(
     CFG_MODIFICATION_LEVEL=13
     CFG_VERSION_LEVEL=1
)
```

Example C-5 shows the full configuration file.

Example: C-5   configuration file for PCOMM

```
*TSTue Nov  7 20:33:08 2006
NODE=(
     ANYNET_SUPPORT=NONE
     CP_ALIAS=EECPPCOM
     DEFAULT_PREFERENCE=NATIVE
     DISCOVERY_SUPPORT=NO
     DLUR_SUPPORT=MULTI_SUBNET
     FQ_CP_NAME=RDBOOKEE.EECPPCOM
     GVRN_SUPPORT=1
     NODE_ID=05D00000
     NODE_TYPE=END_NODE
     REGISTER_WITH_CDS=1
     REGISTER_WITH_NN=1
     SEND_TERM_SELF=0
)

PORT=(
     PORT_NAME=IBMEEDLC
     ACTIVATION_DELAY_TIMER=30
     DELAY_APPLICATION_RETRIES=1
```

```
          DLC_NAME=IBMEEDLC
          IMPLICIT_CP_CP_SESS_SUPPORT=1
          IMPLICIT_DEACT_TIMER=600
          IMPLICIT_DSPU_SERVICES=NONE
          IMPLICIT_HPR_SUPPORT=1
          IMPLICIT_LIMITED_RESOURCE=NO
          IMPLICIT_LINK_LVL_ERROR=0
          LINK_STATION_ROLE=NEGOTIABLE
          MAX_ACTIVATION_ATTEMPTS=0
          MAX_IFRM_RCVD=8
          MAX_RCV_BTU_SIZE=1461
          PORT_TYPE=SATF
          RETRY_LINK_ON_DISCONNECT=1
          RETRY_LINK_ON_FAILED_START=1
          RETRY_LINK_ON_FAILURE=1
          DEFAULT_TG_CHARS=(
               COST_PER_BYTE=0
               COST_PER_CONNECT_TIME=0
               EFFECTIVE_CAPACITY=160
               PROPAGATION_DELAY=LAN
               SECURITY=NONSECURE
               USER_DEFINED_1=0
               USER_DEFINED_2=0
               USER_DEFINED_3=0
          )
          PORT_OEM_SPECIFIC_DATA=(
               OEM_LINK_DATA=(
                    OEM_DATA=010000000000000000000000030000000F00000000000000
                    OEM_DATA=0A0000000000000000
               )
               OEM_PORT_DEFAULTS=(
                    COST_PER_CONNECT_TIME=0
                    EFFECTIVE_CAPACITY=160
                    INB_LINK_ACT_LIM=128
                    OUT_LINK_ACT_LIM=127
                    PROPAGATION_DELAY=LAN
                    SECURITY=NONSECURE
                    TOT_LINK_ACT_LIM=255
               )
          )
     )

     LINK_STATION=(
          LS_NAME=EELINK01
          ACTIVATE_AT_STARTUP=1
          ACTIVATION_DELAY_TIMER=-1
          ADJACENT_NODE_TYPE=NETWORK_NODE
          AUTO_ACTIVATE_SUPPORT=0
          CP_CP_SESS_SUPPORT=1
          DEFAULT_NN_SERVER=1
          DELAY_APPLICATION_RETRIES=1
          DEPENDENT_LU_ENCRYPTION=NONE
          DEST_ADDRESS=044000
          DISABLE_REMOTE_ACT=0
          DSPU_SERVICES=NONE
          ETHERNET_FORMAT=0
          FQ_ADJACENT_CP_NAME=RDBOOKEE.SC30M
          HPR_LINK_LVL_ERROR=0
          HPR_SUPPORT=1
          INHERIT_PORT_RETRY_PARMS=0
          LIMITED_RESOURCE=NO
          LINK_DEACT_TIMER=600
          LINK_STATION_ROLE=NEGOTIABLE
          MAX_ACTIVATION_ATTEMPTS=-1
          MAX_IFRM_RCVD=7
          MAX_SEND_BTU_SIZE=1461
          NODE_ID=05D00000
          PORT_NAME=IBMEEDLC
          RETRY_LINK_ON_DISCONNECT=1
          RETRY_LINK_ON_FAILED_START=1
          RETRY_LINK_ON_FAILURE=1
          SOLICIT_SSCP_SESSION=0
          TG_NUMBER=0
          USE_DEFAULT_TG_CHARS=1
          LINK_STATION_OEM_SPECIFIC_DATA=(
               OEM_LINK_DATA=(
                    OEM_DATA=010000000400000004000000030000000F00000000000000
                    OEM_DATA=0A00000016000000534333304D2D4545322E6974736F2E69
                    OEM_DATA=626D2E636F6D00
               )
          )
          TG_CHARS=(
               COST_PER_BYTE=0
               COST_PER_CONNECT_TIME=0
               EFFECTIVE_CAPACITY=0
               PROPAGATION_DELAY=MINIMUM
               USER_DEFINED_1=0
               USER_DEFINED_2=0
               USER_DEFINED_3=0
          )
     )

     LINK_STATION=(
          LS_NAME=EELINK02
          ACTIVATE_AT_STARTUP=1
          ACTIVATION_DELAY_TIMER=-1
          ADJACENT_NODE_TYPE=NETWORK_NODE
          AUTO_ACTIVATE_SUPPORT=0
          CP_CP_SESS_SUPPORT=1
          DEFAULT_NN_SERVER=0
          DELAY_APPLICATION_RETRIES=1
          DEPENDENT_LU_ENCRYPTION=NONE
          DEST_ADDRESS=044000
          DISABLE_REMOTE_ACT=0
          DSPU_SERVICES=NONE
          ETHERNET_FORMAT=0
          FQ_ADJACENT_CP_NAME=RDBOOKEE.SC31M
          HPR_LINK_LVL_ERROR=0
          HPR_SUPPORT=1
          INHERIT_PORT_RETRY_PARMS=0
          LIMITED_RESOURCE=NO
          LINK_DEACT_TIMER=600
          LINK_STATION_ROLE=NEGOTIABLE
          MAX_ACTIVATION_ATTEMPTS=-1
          MAX_IFRM_RCVD=7
          MAX_SEND_BTU_SIZE=1461
          NODE_ID=05D00000
          PORT_NAME=IBMEEDLC
          RETRY_LINK_ON_DISCONNECT=1
          RETRY_LINK_ON_FAILED_START=1
          RETRY_LINK_ON_FAILURE=1
          SOLICIT_SSCP_SESSION=0
          TG_NUMBER=0
          USE_DEFAULT_TG_CHARS=1
          LINK_STATION_OEM_SPECIFIC_DATA=(
               OEM_LINK_DATA=(
                    OEM_DATA=010000000400000004000000030000000F00000000000000
                    OEM_DATA=0A00000016000000534333314D2D4545322E6974736F2E69
                    OEM_DATA=626D2E636F6D00
               )
          )
          TG_CHARS=(
               COST_PER_BYTE=0
               COST_PER_CONNECT_TIME=0
               EFFECTIVE_CAPACITY=0
               PROPAGATION_DELAY=MINIMUM
               USER_DEFINED_1=0
               USER_DEFINED_2=0
               USER_DEFINED_3=0
          )
     )

     INTERNAL_PU=(
          PU_NAME=DLPUPCOM
          BKUP_DLUS_NAME=RDBOOKEE.SC31M
          DEPENDENT_LU_ENCRYPTION=NONE
          FQ_DLUS_NAME=RDBOOKEE.SC30M
          NODE_ID=01300010
          STARTUP=1
     )

     DLUR_DEFAULTS=(
          DEFAULT_PU_NAME=EECPPCOM
          DLUS_RETRY_LIMIT=65535
          DLUS_RETRY_TIMEOUT=5
     )

     LU_0_TO_3=(
          LU_NAME=DLPCOM02
          LU_MODEL=UNKNOWN
          NAU_ADDRESS=2
          PRIORITY=MEDIUM
          PU_NAME=DLPUPCOM
     )
```

```
LU_0_TO_3=(
     LU_NAME=DLPCOM03
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=3
     PRIORITY=MEDIUM
     PU_NAME=DLPUPCOM
)

LU_0_TO_3=(
     LU_NAME=DLPCOM04
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=4
     PRIORITY=MEDIUM
     PU_NAME=DLPUPCOM
)

LU_0_TO_3=(
     LU_NAME=DLPCOM05
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=5
     PRIORITY=MEDIUM
     PU_NAME=DLPUPCOM
)

LU_0_TO_3=(
     LU_NAME=DLPCOM06
     LU_MODEL=UNKNOWN
     NAU_ADDRESS=6
     PRIORITY=MEDIUM
     PU_NAME=DLPUPCOM
)

LOCAL_LU=(
     LU_NAME=ILPCOM01
     DEFAULT_POOL=0
     LU_ALIAS=ILPCOM01
     LU_SESSION_LIMIT=0
     NAU_ADDRESS=0
)

MODE=(
     MODE_NAME=BLANK
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=3
)

MODE=(
     MODE_NAME=#BATCH
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#BATCH
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=0
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=20
)

MODE=(
     MODE_NAME=#BATCHSC
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#BATCHSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=3
)

MODE=(
     MODE_NAME=#CONNECT
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=128
     MAX_RU_SIZE_UPPER_BOUND=4096
     MIN_CONWINNERS_SOURCE=16
     PLU_MODE_SESSION_LIMIT=32
     RECEIVE_PACING_WINDOW=1
)

MODE=(
     MODE_NAME=#INTER
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#INTER
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=4096
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=20
)
```

```
MODE=(
     MODE_NAME=#INTERSC
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#INTERSC
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=256
     MAX_RU_SIZE_UPPER_BOUND=2048
     MIN_CONWINNERS_SOURCE=128
     PLU_MODE_SESSION_LIMIT=256
     RECEIVE_PACING_WINDOW=7
)

MODE=(
     MODE_NAME=QPCSUPP
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=52
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=26
     PLU_MODE_SESSION_LIMIT=52
     RECEIVE_PACING_WINDOW=2
)

MODE=(
     MODE_NAME=QSERVER
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=#CONNECT
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=1
     MAX_NEGOTIABLE_SESSION_LIMIT=64
     MAX_RU_SIZE_UPPER_BOUND=1024
     MIN_CONWINNERS_SOURCE=0
     PLU_MODE_SESSION_LIMIT=64
     RECEIVE_PACING_WINDOW=7
)

MODE=(
     MODE_NAME=SNASVCMG
     AUTO_ACT=0
     COMPRESSION=PROHIBITED
     COS_NAME=SNASVCMG
     ENCRYPTION_SUPPORT=NONE
     DEFAULT_RU_SIZE=0
     MAX_NEGOTIABLE_SESSION_LIMIT=2
     MAX_RU_SIZE_UPPER_BOUND=512
     MIN_CONWINNERS_SOURCE=1
     PLU_MODE_SESSION_LIMIT=2
     RECEIVE_PACING_WINDOW=1
)
```

```
CONNECTION_NETWORK=(
     FQCN_NAME=RDBOOKEE.VRNLOCAL
     PORT_NAME=IBMEEDLC
     INHERIT_PORT_LIMITED_RESOURCE=NO
)

SHARED_FOLDERS=(
     CACHE_SIZE=256
)

VERIFY=(
     CFG_MODIFICATION_LEVEL=13
     CFG_VERSION_LEVEL=1
)
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 517. Note that some of the documents referenced here may be available in softcopy only.

- *Migrating Subarea Networks to an IP Infrastructure Using Enterprise Extender*, SG24-5957
- *Subarea to APPN Migration: VTAM and APPN Implementation*, SG24-4656
- *Subarea to APPN Migration: HPR and DLUR Implementation*, SG24-5204
- *Inside APPN and HPR - The Essential Guide to New SNA*, SG24-3669
- *VTAM V4.3: High Performance Routing (HPR) Early User Experiences*, SG24-4507
- *Dynamic Subarea and APPN Management Using NetView V3R1*, SG24-4520
- *Managing Your APPN Environments Using NetView*, GG24-2559
- *Communications Server for z/OS V1R8 TCP/IP Implementation Volume 1: Base Functions, Connectivity, and Routing*, SG24-7339
- *Communications Server for z/OS V1R8 TCP/IP Implementation Volume 2: Standard Applications*, SG24-7340
- *Communications Server for z/OS V1R8 TCP/IP Implementation Volume 3: High Availability, Scalability, and Performance*, SG24-7341
- *Communications Server for z/OS V1R8 TCP/IP Implementation Volume 4: Policy-Based Network Security*, SG24-7342
- *A Structured Approach to Modernizing the SNA Environment*, SG24-7334
- *OSA-Express Implementation Guide*, SG24-5948
- *IBM Communication Controller for Linux on System z V1.2.1 Implementation Guide*, SG24-7223

Other publications

These publications are also relevant as further information sources:

- *z/OS V1R8.0 Communications Server: SNA Network Implementation,* SC31-8777
- *z/OS V1R8.0 Communications Server: New Function Summary,* GC31-8771
- *z/OS Communications Server: IP System Administrator’s Commands*, SC31-8781
- *z/OS MVS IPCS Commands*, SA22-7594
- *z/OS MVS System Commands*, SA22-7627
- *z/OS UNIX System Services Command Reference*, SA22-7802
- *z/OS Communications Server: CSM Guide*, SC31-8808

- *z/OS Communications Server: Quick Reference*, SX75-0124
- *z/OS Communications Server: IP and SNA Codes*, SC31-8791
- *z/OS MVS IPCS Commands*, SA22-7594
- *z/OS Communications Server: IP Diagnosis Guide*, GC31-8782
- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Communications Server: IP Messages Volume 1 (EZA)*, SC31-8783
- *z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)*, SC31-8784
- *z/OS Communications Server: IP Messages Volume 3 (EZY)*, SC31-8785
- *z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)*, SC31-8786
- *z/OS Communications Server: IP Programmer's Guide and Reference*, SC31-8787
- *z/OS Communications Server: IP Configuration Reference*, SC31-8776
- *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*, SC31-8788
- *z/OS Communications Server: IP User's Guide and Commands*, SC31-8780
- *z/OS Communications Server: IP User's Guide and Commands*, SC31-8780
- *z/OS Communications Server: IPv6 Network and Application Design Guide*, SC31-8885
- *Linux on zSeries Device Drivers, Features and Commands*, SC33-8281
- *z/OS Migration*, GA22-7499
- Communications Server migration actions chapter in *z/OS Migration*, GA22-7499
- *z/OS MVS System Commands*, SA22-7627
- *OSA-Express Customer's Guide and Reference*, SA22-7935
- *z/OS Communications Server SNA Data Areas, Volume 1*, GC31-6852
- *z/OS Communications Server SNA Data Areas, Volume 2*, GC31-6853
- *z/OS Communications Server: IP and SNA Codes*, SC31-8791
- *z/OS Communications Server: SNA Operation*, SC31-8779
- *z/OS Communications Server SNA Customization*, SC31-6854
- *z/OS Communications Server SNA Diagnosis, Volume 1: Techniques and Procedures*, GC31-6850
- *z/OS Communications Server SNA Diagnosis, Volume 2: FFST Dumps and the VIT*, GC31-6851
- *z/OS Communications Server: SNA Messages*, SC31-8790
- *z/OS Communications Server SNA Resource Definition Samples*, SC31-8836
- *z/OS TSO/E Command Reference*, SA22-7782
- *z/OS UNIX System Services User's Guide*, SA22-7801

Online resources

These Web sites are also relevant as further information sources:

- Wireshark - Protocol analyzer based on open source projects

  `http://www.wireshark.org`

- Understanding Enterprise Extender, Part 1 - Concepts and Considerations

  http://www-1.ibm.com/support/docview.wss?rs=852&uid=swg27006667
- z/OS Communications Server - Product support

  http://www-306.ibm.com/software/network/commserver/zos/support/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

Enterprise Extender Implementation Guide

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

IBM®

Redbooks®

Enterprise Extender Implementation Guide

Enterprise Extender concepts, terminology, and supported functions

Planning, implementation, and migration guidance

Realistic examples and scenarios

This IBM Redbooks publication will help you to tailor and configure Communications Server for z/OS (CS z/OS) to make full use of Enterprise Extender (EE) capabilities. It focuses on the migration of your Advanced Peer-to-Peer Networking (APPN) environment to Enterprise Extender, while offering easy-to-understand, step-by-step guidance. Sample scenarios are provided that discuss Enterprise Extender connections between multiple z/OS systems as well as between z/OS and non-mainframe systems. The non-mainframe platforms in our examples include IBM Communications Server for AIX (CS/AIX), IBM Communications Server for Linux (CS Linux), IBM Communications Server for Windows (CS Windows), IBM Personal Communications for Windows (PCOMM), and i5/OS Enterprise Extender support.

This publication provides information to assist you with the planning, implementation, and setup of Enterprise Extender. In addition, it describes helpful utilities and commands that you can use to monitor and operate the Enterprise Extender environment. It discusses the motivation for migrating to Enterprise Extender and explains the planning decisions that must be considered before attempting each phase of a migration. Further, it illustrates working scenarios, highlighting the important aspects of configuration and connectivity, and discusses techniques for avoiding undesirable situations. It explains the changes that are necessary in the VTAM and TCP/IP definitions to support Enterprise Extender in each scenario.

You should have a solid background in SNA, APPN, and TCP/IP networking, as well as experience in the setup and operation of Communications Server for z/OS (VTAM and TCP/IP). Whether you are a systems engineer, network administrator, or systems programmer that will plan for and configure Enterprise Extender this book will be useful to you. Enterprise Extender requires an APPN/HPR environment. Adding HPR support to the base APPN environment is explained and described in detail, with examples.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7359-00 ISBN 0738486272