EFFICIENT AVAMAR BACKUPS OVER WAN AND SIZING LINKS

Balaji Panchanathan
EMC

Jayalakshmi Suresh
EMC

Pravin Ashok Kumar
EMC

EMC²

Table of Contents

Introduction

This article will focus on four things:

1. New features in Avamar® 7.1 which help WAN backups
2. How to estimate the bandwidth required for WAN links based on the application and data size. This depends on the dedupe rate for the application and use of a WAN emulator to measure it (Linux, open source tools, i.e. netem, etc.)
3. Performance number for desktop/laptop (DTLT), Avamar Extended Retention (AER), Data Domain® (DD) with different encryption strengths
4. Broad recommendations for the customer

New in Avamar 7.1 for WAN

Starting with Avamar 7.1, WAN is a supported configuration with Data Domain as the target storage device. With this support, metadata can be stored in Avamar and the data can be moved across the WAN to the Data Domain device.

A salient feature is support of a 60-minute outage of the WAN Link and support for over-WAN backup to Data Domain as the target.

Figure 1 depicts a type of network configuration that is supported.

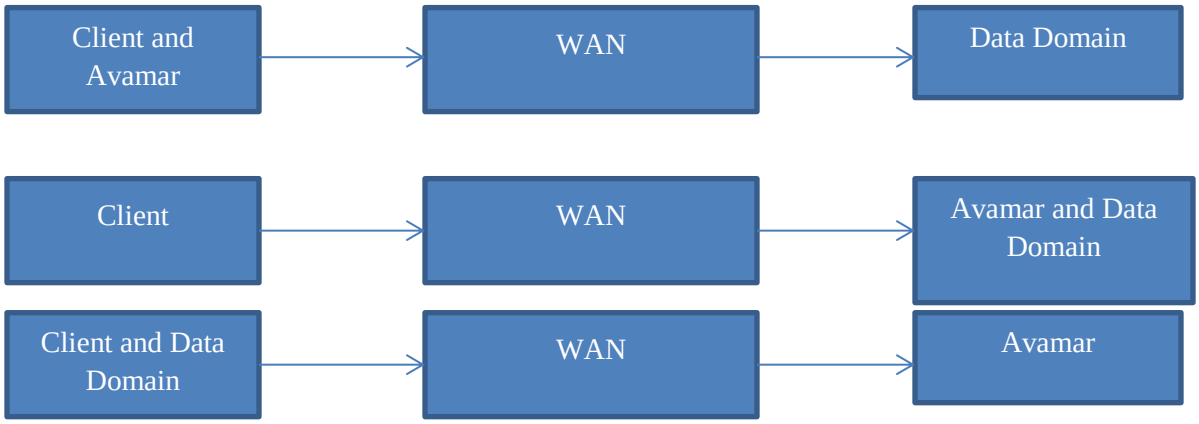


Figure 1

This support provides customers the flexibility to deploy AVE's in each remote office and have one Data Domain in a central location. Optionally, the customer can have one central Avamar server and deploy Data Domain virtual edition in each branch office.

Sizing of WAN Links

The customer has to estimate the size of the WAN links required for backing up their data. To do so, open source tools like netem, available on any Avamar server or Linux machine, can be used.

Customers can use the results of the test shown below to decide where Avamar and Data Domain needs to be deployed, i.e. in a remote location or in a central office.

The following set of test equipment will enable customers to easily perform the test decide for themselves.

- ESX Server - host
- Avamar Server – AVE virtual edition
- Data Domain – Virtual Edition
- Linux – SLES 11 SP1
- Windows client
- Linux Client

Avamar server virtual edition, Data Domain virtual edition, and Linux WAN emulator can be installed in a single ESX host.

The network diagram will look like that which is shown in Figure 2.

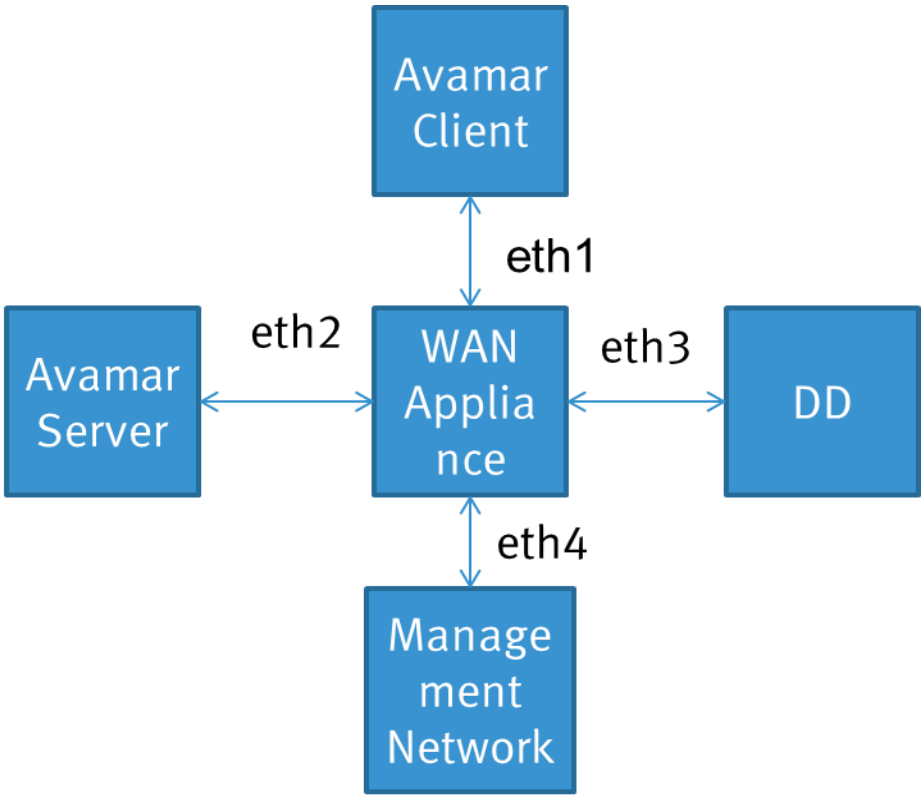


Figure 2

Configuration

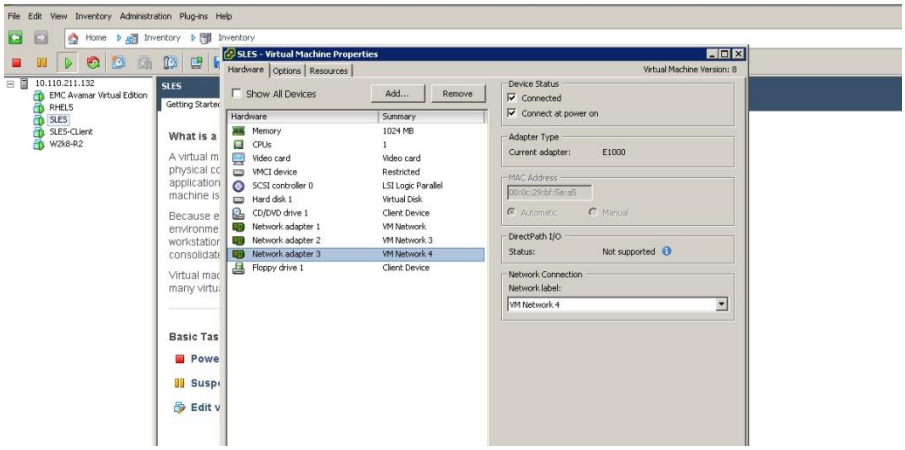
- Client should be in the same network as that of one interface of the network appliance
- Server should be in the same network as that of one interface of the network appliance
- Server and client should be on different network
- Data Domain should be on a different network on the same ESX host

Follow the steps below:

1. ESX Configuration: Step to add new network to the ESX.
    - Log in to the ESX host using vSphere Client
    - Click on Configuration
    - Click on Networking
    - Click on Add networking (which will be displayed on the right side)
    - Select Virtual machine
    - Use the network label as VM Network 1
    - Repeat the above steps again and add VM Network 2, 3, 4, and 5

2. ESX VM Appliance Configuration: We need to add four interfaces to the SLES machine. The interfaces can be added by following the steps below.
    - Log in to the ESX host using vSphere Client
    - Deploy the VM using the vmdk file
    - Add disk capacity
    - Power On
    - Right click on the VM
    - Edit Settings
    - Click on ADD
    - Select Ethernet Adapter
    - Select the VM network (for the second interface, select VM network 1 (by default first interface will be added). For the third interface, select VM Network 2)

    Shown below is the sample snapshot after the interfaces are added.



3. ESX Configuration for client and server:
    - Log in in to the ESX host using vSphere Client
    - Right click on the Client VM
    - Edit Settings
    - Click on the Network Adaptor and then change the label to VM Network 1
    - In a similar way, click on the Server VM (AVE) and then change the label to VM Network 2
4. IP Address on the network appliance:
    - Give the command ifconfig –a and get the list of interfaces (ex: eth0, eth2, eth5, etc. and then configure the IPs using the commands below (replacing the interface respectively)

i. Ifconfig eth0 10.110.209.230 netmask 255.255.252.0
ii. Ifconfig eth1 192.168.2.11 netmask 255.255.255.0
iii. Ifconfig eth2 192.168.1.3 netmask 255.255.255.0
iv. Ifconfig eth3 192.168.3.1 netmask 255.255.255.0

After configuring the IP address, the configs can be checked by using the ifconfig command.

5. Routing-related config
   - Sysctl –p net.ipv4.ip_forward=1
   - On Client side
     i. route add –net 192.168.1.0 netmask 255.255.255.0 gw 192.168.2.11
     ii. route add –net 192.168.3.0 netmask 255.255.255.0 gw 192.168.2.11
   b. On Server side
     i. route add –net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.3
     ii. route add –net 192.168.3.0 netmask 255.255.255.0 gw 192.168.1.3
   c. On Data Domain side
     i. route add –net 192.168.1.0 netmask 255.255.255.0 gw 192.168.3.1
     ii. route add –net 192.168.2.0 netmask 255.255.255.0 gw 192.168.3.1
   d. Route-related config can be checked using route –n command

Note: In the above sample ifconfig and route commands, the ipaddress/netmask should be replaced by your ip/netmask, respectively

Type of WAN simulations and their configurations

1. Drop, delay, out-of-order(TCP Level)
2. Bandwidth throttle
3. Network impairments can be done on both client/server interfaces

Commands to simulate Network impairments:

Scenario	Command to Add	Command to disable
Drop – On Server side	iptables -A FORWARD -d <Clientipaddress> -m statistic --mode random --probability <dropratio> -j DROP	iptables -D FORWARD -d <clientipaddress> -m statistic --mode random --probability <dropratio> -j DROP
Drop – On Client Side	iptables -A FORWARD -d <serveraddress> -m statistic --mode random --probability 0.95 -j DROP	iptables -A FORWARD -d <serveraddress> -m statistic --mode random --probability 0.95 -j DROP
Delay on Server Side	tc qdisc add dev eth1 root netem delay 0.7ms	tc qdisc del dev eth1 root netem delay 0.7ms
Delay – On Client Side	tc qdisc add dev eth2 root netem delay 0.7ms	tc qdisc del dev eth2 root netem delay 0.7ms
Out-of-order – Client Side	tc qdisc add dev eth1 root netem gap 5 delay 10ms	tc qdisc del dev eth1 root netem gap 5 delay 10ms
Out-of-order – Server Side	tc qdisc add dev eth2 root netem gap 5 delay 10ms reorder 10	tc qdisc del dev eth2 root netem gap 5 delay 10ms reorder 10
Bandwidth Throttle	tc qdisc add dev eth1 root handle 1:0 netem delay 100ms tc qdisc add dev eth1 parent 1:1 handle 10: tbf rate 256kbit buffer 1600 limit 3000	tc qdisc del dev eth1 root handle 1:0 netem delay 100ms

After executing the command, we can check whether those settings are in effect using the tc filter show dev <interface>command.

How to measure the traffic on the appliance

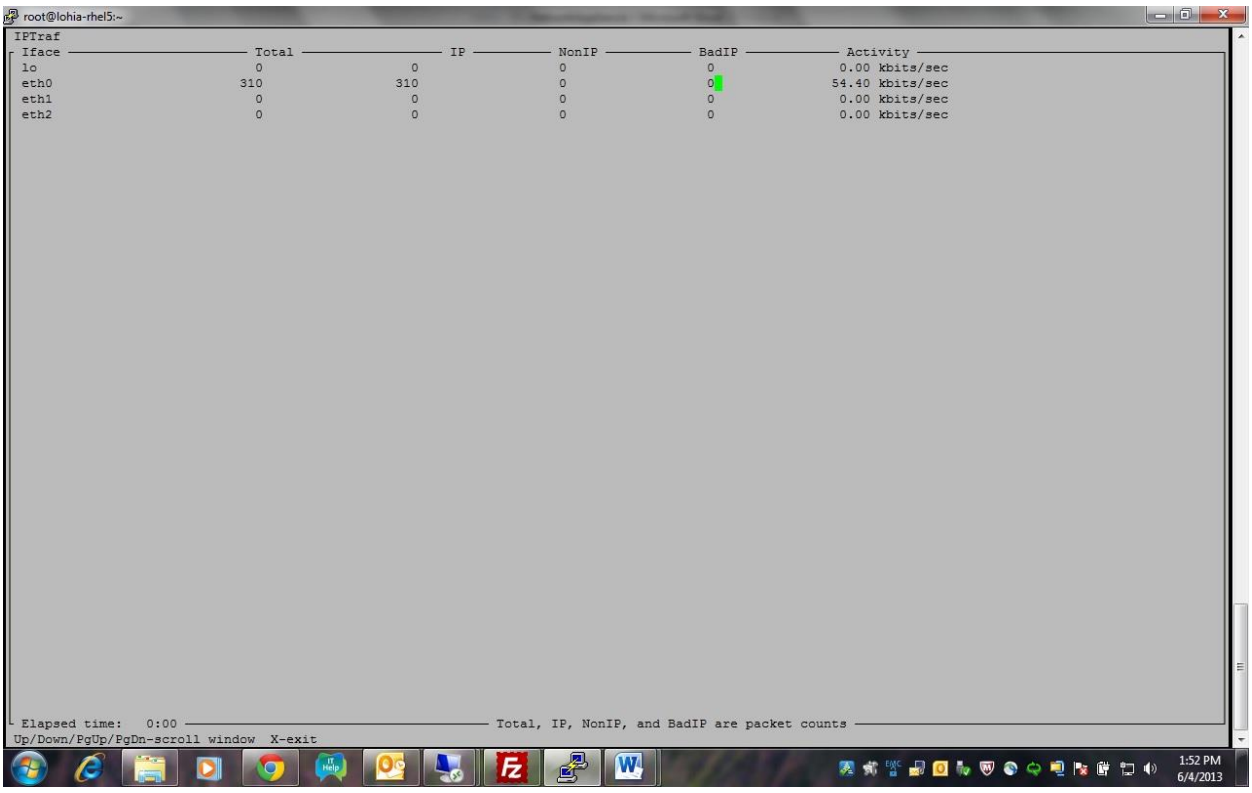
The iptraf tool, installed and running the iptraf command, will help monitor traffic on the appliance.

Follow the steps below:

- On the command line, run the command iptraf
- Enter a key to continue
- Select General Interface statistics
- Select a file to which you want to log the stats

The screen will display the traffic flowing through each of the interfaces.

Below is the snapshot of how it will look after following the steps above.



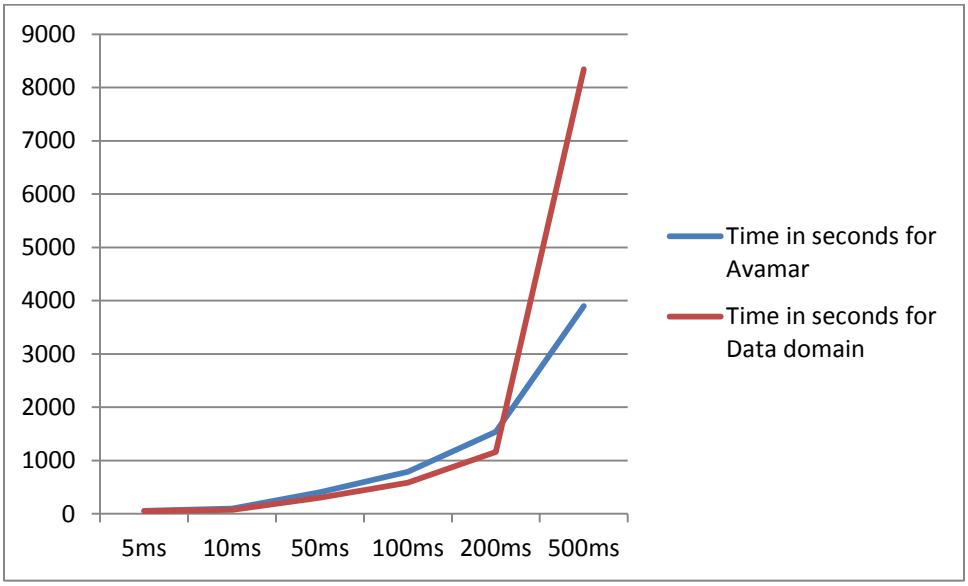
Performing the test set up above and using those commands, customers can simulate different WAN conditions, i.e. Drop rate, bandwidth throttle, etc.

Customers can also disable WAN conditions on Avamar® and have only WAN condition for Data Domain (and vice versa), enabling them to check which application offers better results and decide on the architecture .

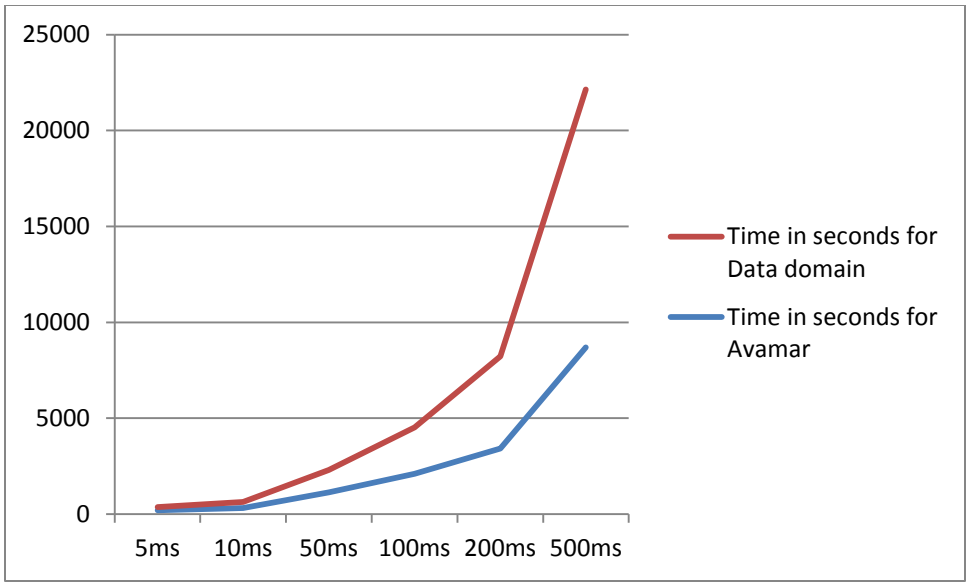
Performance test results over WAN

Our testing on filesystem backup over WAN delivered the results below.

Delay	Time in seconds for Avamar	Time in seconds for Data Domain
5ms	53	47
10ms	96	76
50ms	405	301
100ms	791	585
200ms	1538	1160
500ms	3900	8340



1% drop and different delay	Time in seconds for Avamar	Time in seconds for Data Domain
5ms	196	183
10ms	315	328
50ms	1140	1168
100ms	2100	2428
200ms	3420	4800
500ms	8700	13440



Bandwidth Throttle results:

With 1Mbps

1% drop and different delay	Time in seconds for Avamar	Time in seconds for Data Domain
50ms	147	148
500ms	177	156

With 10Mbps

1% drop and different delay	Time in seconds for Avamar	Time in seconds for Data Domain
5ms	15	14
50ms	24	15
500ms	124	48

DTLT

Results for different WAN profiles we have tested in desktop laptop environment (DTLT) are shown below.

Profile	SW	Speed UP	Average latency	Jitter	Packet Drop	streams count	Time taken for Backup to DD 5.5
C1 - Worst	1Gb	up-64Kbps-down768Kbps	100ms	25%	1in 100	1	24 Hrs5min --Time out End
C1 - Medium	1Gb	up200kbps-down3000Kbps	100ms	20%	1 in 500	1	12Hrs 23 mins
C1 - Best	1Gb	up-384Kbps-down6000kbps	100ms	10%	1 in 1000	1	6Hrs 27min
C2 - Worst	1Gb	up-512kbps-down-6Mbps	100ms	25%	1 in 200	1	4Hrs50min42secs
C2 - Medium	1Gb	up-768Kbps-down9Mbps	100ms	20%	1 in 600	1	3Hr9mins64sec
C2 - Best	1Gb	up 1024Kbps-down12Mbps	100ms	10%	1 in 1000	1	2hr25min26s
C3 - Worst	1Gb	1.5Mbps-down1.5Mbps	100ms	20%	1 in 200	1	1hr39mins20s
C3 - Medium	1Gb	20Mbps-down20Mbps	100ms	15%	1 in 1000	1	7min38secs
C3 - Best	1Gb	45Mbps-down45Mbps	100ms	10%	1 in 2000	1	3min32secs
C4 - Worst	1Gb	1.5Mbps-down1.5Mbps	100ms	20%	1 in 500	1	1Hr39mins20sec
C4 - Medium	1Gb	20Mbps-down20Mbps	100ms	15%	1 in 3000	1	7min39secs
C4- Best	1Gb	45Mbps-down45Mbps	100ms	10%	1 in 5000	1	3min31secs

AER

The Avamar Extended Retention (AER) feature is used for Avamar backup retention to tape and restore those retained backups to clients. Formerly called Direct-to-Tape Out (DTO), it is an archiving solution for Avamar.

Main tasks involved in AER are

- Exports (Identifying the backups and pushing it to tape libraries which is attached to AER Node),
- Imports (Moves the backup from Tape to AER Node (physical storage).
- Restore (Registering the client to AER and restoring the respective backups to Client).

1GB Data	Export	43 sec	normal
		2m:20sec	delay 10ms
		4m:27sec	delay 20ms
		6m:56sec	delay 30ms
		8m:40sec	delay 40ms
	Restore	32 sec	normal
		3m:46sec	delay 10ms
		7m:14sec	delay 20ms
		10m:36sec3m:46sec	delay 30msdelay 10ms
		14m:04sec7m:14sec	delay 40msdelay 20ms
		10m:36sec	delay 30ms
		14m:04sec	delay 40ms

Throttle

1GB data	Export	43sec	normal
		9m:09sec	2mbps
		7m:07sec	3mbps
		5m:02sec	4mbps
		3m:43sec	5mbps
	Restore	32 sec	normal
		55m:56sec	2mbps
		51m:49sec	3mbps
		49m:36sec	4mbps
		44m:11sec	5mbps

Observations

- Impact of delay on restore is greater compared to backup in exports. Additionally, there is 50% greater impact on restore compared to backup.
- Impact of bandwidth throttle is greater in restore, at least 10x worse. These things should be taken into account when the customer wants to restore (import) from AER node.

Recommendations

- WAN throughput different between medium and high encryption is minimal
- Backup window required for different clients/applications
    - Test with different CPU throttles and test whether CPU usage has any impact on WAN throughput. Our assumption is that the bottleneck is only the network and this assumption needs to be validated.

Broad recommendations based on the tests conducted

- Data Domain performs better if the delay is less, in the range of 5-100ms. If the delay is 500ms, Avamar performance is much better, by at least 2x. However, with bandwidth less than 1Mbps, even with 500ms delay, Data Domain is better.
- The impact of delay when the available bandwidth is 1Mbps is much less, roughly a 20% drop in performance for Avamar and 5% for Data Domain when the delay increases from 5ms to 500ms. Hence, with bandwidth throttle, it is better to use Data Domain as storage target rather than Avamar.

Conclusion

Performance numbers in WAN conditions are given in this article. The same can be used for sizing the WAN links. Customers can also easily test their numbers using open source tools like netem/tc, etc. This will help customers avoid surprises and evaluate the different products available to select the best product. This set of WAN tools cannot only be used with Avamar but also with other backup products to select the right product and right WAN size.

Appendix

Below is the bandwidth script which can be used on the Linux SLES box (WAN Emulator). Using the script, bandwidth throttle can be applied and tests can be conducted.

```
#!/bin/bash
#
#  tc uses the following units when passed as a parameter.
#  kbps: Kilobytes per second
#  mbps: Megabytes per second
#  kbit: Kilobits per second
#  mbit: Megabits per second
#  bps: Bytes per second
#       Amounts of data can be specified in:
#       kb or k: Kilobytes
#       mb or m: Megabytes
#       mbit: Megabits
#       kbit: Kilobits
#  To get the byte figure from bits, divide the number by 8 bit
#


#
# Name of the traffic control command.
TC=tc


# The network interface we're planning on limiting bandwidth.
IF=eth5          # Interface4
```

```
# Download limit (in mega bits)
DNLD=10mbit          # DOWNLOAD Limit

# Upload limit (in mega bits)
UPLD=10mbit          # UPLOAD Limitit

# IP address of the machine we are controlling
IP=192.168.4.12 # Host IP

# Filter options for limiting the intended interface.
U32="$TC filter add dev $IF protocol ip parent 1:0 prio 1 u32"

start() {

# We'll use Hierarchical Token Bucket (HTB) to shape bandwidth.
# For detailed configuration options, please consult Linux man
# page.

$TC qdisc add dev $IF root handle 1: htb default 30
$TC class add dev $IF parent 1: classid 1:1 htb rate $DNLD
$TC class add dev $IF parent 1: classid 1:2 htb rate $UPLD
$U32 match ip dst $IP/32 flowid 1:1
$U32 match ip src $IP/32 flowid 1:2

# The first line creates the root qdisc, and the next two lines
# create two child qdisc that are to be used to shape download
```

```
# and upload bandwidth.
#
# The 4th and 5th line creates the filter to match the interface.
# The 'dst' IP address is used to limit download speed, and the
# 'src' IP address is used to limit upload speed.

}

stop() {

# Stop the bandwidth shaping.
$TC qdisc del dev $IF root

}

restart() {

# Self-explanatory.
stop
sleep 1
start

}

show() {
```

```
# Display status of traffic control status.
$TC -s qdisc ls dev $IF

}

case "$1" in

start)

echo -n "Starting bandwidth shaping: "
start
echo "done"
;;

stop)

echo -n "Stopping bandwidth shaping: "
stop
echo "done"
;;

restart)

echo -n "Restarting bandwidth shaping: "
restart
echo "done"
```

```
;;

show)

echo "Bandwidth shaping status for $IF:"
show
echo ""
;;

*)

pwd=$(pwd)
echo "Usage: tc.bash {start|stop|restart|show}"
;;

esac
exit
```

References

http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

http://man7.org/linux/man-pages/man8/tc-netem.8.html

http://www.spippolazione.net/index.php/tag/traffic-shaping/

http://iptraf.seul.org/2.2/manual.html

http://www.slashroot.in/linux-iptraf-and-iftop-monitor-and-analyse-network-traffic-and-bandwidth

https://www.youtube.com/watch?v=Y5un7JTGp3o

https://gist.github.com/jterrace/1895081

http://linux.die.net/man/8/ifconfig

http://linux.about.com/od/commands/l/blcmdl8_route.htm

http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-networking-guide.pdf

http://www.iplocation.net/tools/traffic-control.php