

Technical Tips and Tricks | for Routers and Managed Layer 3 Switches

Introduction

This document contains useful technical tips and tricks for Allied Telesis routers and managed Layer 3 switches.

Contents

This revision of Tips and Tricks contains the following new tips and tricks:

1. How to capture command output in a text file 3
2. How to automatically capture output when particular events occur 5
3. How to upgrade the GUI when upgrading to Software Version 2.8.1 9
4. A simpler way to save the current configuration 10
5. How to store stack dump files 11
6. How to securely manage remote devices from an Asyn (console) port on a router or Rapier 13
7. How to reduce the impact of storms by using QoS policy storm protection 16
8. How to reduce the impact of storms by controlling rapid MAC movement 18
9. How to use SNMP to monitor STP and RSTP links 20
10. How to use SNMP to monitor master and slave SwitchBlade controller cards 25
11. Why you need to use an idle timer on a PPPoE link 32
12. How to handle RIP route tags 34
13. Route compatibility when RIP is set to receive both RIPv1 and RIPv2 routes 37
14. How to select the right ISAKMP policy during incoming Phase 1 ISAKMP proposals 40
15. Why the remote peer VPN router may set up multiple ISAKMP SAs when responding to my router 43
16. About the firewall's aggressive mode 47
17. How to combine firewall standard and enhanced NAT 49

These Tips and Tricks apply to:

Routers

AR415S
AR440S, AR441S
AR442S
AR450S
AR750S, AR750S-DP
AR770S
AR725, AR745
AR720, AR740
AR410, AR410S

Switches

AT-8624T/2M
AT-8624PoE
AT-8648T/2SP
AT-8724XL
AT-8748XL
Rapier 24i, Rapier 24
Rapier 48i, Rapier 48
Rapier 16fi
Rapier 16f
Rapier G6
AT-8824
AT-8848
AT-9812T
AT-9816GB
SwitchBlade
AT-8948, x900-48FE
x900-48FE-N
AT-9924T, AT-9924SP
AT-9924T/4SP
AT-9924Ts
x900-24XT
x900-24XT-N

This document also contains the following Tips and Tricks from earlier revisions:

18.	Why the switch has many “interface is UP” and “interface is DOWN” log messages	51
19.	How to gather useful debugging information for a suspected memory leak	52
20.	Why unexpected SNMP link-traps are sent from an AR410 router	60
21.	How to deal with spoofed packets	61
22.	How to deal with invalid DA packets	61
23.	How to interrupt text flow that is continuously streaming to the CLI	62
24.	How to display a text message at login	62
25.	How to set an inactivity timeout on console and TTY connections	63
26.	How to set Summer Time and time zones	65
27.	How to ensure that system traffic is given priority when your switch is very busy	67
28.	How to enable and install a release on the SwitchBlade with two controllers	70
29.	How to fix switch port speed but still negotiate duplex	72
30.	How to make private and public VLANs share the same uplink	73
31.	RSTP BPDU detection features	74
32.	How to allocate a WAN IP address to a PPP peer, and create a separate route to the subnet on the LAN side of the peer	75
33.	How to reflect TOS onto L2TP tunnelled packets	76
34.	How to use Ping or Trace using Domain Name Service (DNS)	80
35.	How OSPF metrics are calculated	81
36.	Filtering OSPF static routes with a whitelist or blacklist route map	83
37.	How to identify and combat worm attacks	87
38.	Whether encryption is performed in hardware or software	89
39.	How and when to use VRRP IP address adoption	91
40.	Support for RADIUS accounting for 802.1x dynamic VLAN assignment	92
41.	How to configure the firewall to allow outward-going pings but to block inward-coming pings	93
42.	How to use firewall NAT to translate subnets	94
43.	Correct use of firewall NAT when FTP does not use port 21	96
44.	How to enable the firewall enhanced fragment handling mode	97
45.	How to use the HTTP proxy (application gateway)	99
46.	How to use the trustprivate parameter on the firewall to block users on the private side from accessing the device	100
47.	How to use the firewall to control Internet access on the basis of private hosts’ MAC addresses	102
48.	How to configure a timeout on particular UDP ports in a firewall policy	105
49.	Firewall messages relating to SYN attacks	106

How to capture command output in a text file

Instead of displaying command output on screen, you can capture it in a text file in Flash memory.

This Tip describes two things you can use the feature for. For full command details, see the Release Note for Software Version 2.8.1 or 3.1.1, or the File System chapter of the Software Reference.

Products this Tip applies to

All routers and switches listed on page I that run the versions below

Software Versions

3.1.1 or later for AT-9924Ts and x900-24 Series switches

2.8.1 or later for other products

Capturing individual commands

You can use this feature to capture the output of one or more individual commands into a text file.

For example, if you want to capture the output of the command **show ip route** into a text file called **ip-route.txt**, use the command:

```
create file=ip-route.txt command="show ip route"
```

Then you can add other commands—for example, **show ip route count**—to the end of the same file by using the command:

```
add file=ip-route.txt command="show ip route count"
```

Capturing ongoing command output

You can also use this feature to capture ongoing output. For example, you can enable IP route debugging and save the debug output in a file, instead of displaying the output on screen. To do this, create the file and use the **permanentredirect** parameter, as shown in the following example:

```
create file=ip-route.txt command="enable ip route debug" perm
```

When you have captured enough debug, or if you want to view the file contents or upload the file, stop the capture by using the command:

```
reset file=ip-route.txt perm
```

Note that resetting the file does not disable the debugging, although the debug output is no longer displayed or saved.

To start capturing output to the same file again, use the command:

```
add file=ip-route.txt command="enable ip route debug" perm
```

Automatically uploading command output daily

You can use the **create file** command, together with a trigger, to run commands daily and upload the output to a TFTP server in a date-stamped file.

Note that this Tip assumes that the TFTP server accepts files with arbitrary names, which the Allied Telesis AT-TFTP Server 1.9 for Windows does. If your server only allows you to upload files when the filename already exists, you could use a script on the server to create an empty file before the upload is due.

To create and upload the date-stamped file, use the following steps.

1. Create a script

In this example, the script is called **route.scp** and contains the following commands to capture and upload the output of the **show ip route** command:

```
create file=route-%D.txt command="show ip route"
upload file=route-%D.txt
wait 10
del file=route-%D.txt
```

This creates a file called (for example) **route-30-Sep-2006.txt**. As well as the %D variable, which gives the file date, you can also use %N (the router or switch name) and %S (the serial number). However, you cannot use %T (the time), because it includes colons and these are not valid in filenames.

2. Create a trigger to run the script daily

Enable the trigger module, by using the command:

```
enable trigger
```

Create a trigger to run the command at the desired time (2pm in this example), by using the command:

```
create trigger=1 time=14:00 script=route.scp
```

How to automatically capture output when particular events occur

Often when we are trying to track down the cause of an infrequently occurring problem, we would like to capture some particular output when the problem occurs.

One way that this can be done is to have a terminal emulator attached to the console port of the device in question, and a script running on the device that is regularly generating output that is being captured by the terminal emulator. Given that the output is being generated frequently, then there is a very good chance of a capture at the moment of the problem occurring.

However, this is a bit of a hit-and-miss approach. It can generate a **very** large log file, and still not capture anything at the vital moment. It can be inconvenient to the customer to have a terminal emulator attached to the console port of their equipment. Worst of all, this method does not survive across a power cycle or a reboot (the script does not start up automatically after the reboot).

A better method is to use triggers and save the output. The following sections describe two alternatives for how to save the output:

- save it to a file in Flash memory (Software Versions 2.8.1 and 3.1.1, or later)
- log it and send it to a syslog server (all software versions)

Save the output to a file in Flash memory

Since Software Versions 2.8.1 and 3.1.1, you can save script output to a file in Flash memory.

This process is similar to the The Tip "[How to capture command output in a text file](#)" on page 3, which describes this process for saving command output.

I. Create a script that runs the relevant "show" commands. Call it debug.scp

The contents of the script could be something like:

```
-----  
show debug  
wait 30  
show swi port=xyz count  
show ip count=arp  
show swi fdb  
...  
-----
```

In other words, include whatever **show** commands have been identified as capturing the evidence required to pinpoint what is going on when the problem occurs.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All

2. Create another script to run the debug.scp script and save its output to a file

Create a script called (for example) **capture.scp**, containing the following command:

```
add file=debug.txt script=debug.scp
```

3. Create a trigger to run the capture.scp script at the relevant event

These commands need to go into the boot script on the device.

Enable the trigger module, by using the command:

```
enable trigger
```

Create the relevant trigger. If you want to run the script immediately after a reboot, the trigger would be

```
create trigger=1 reboot=all script=capture.scp
```

If you want to run the script when the CPU gets very busy, the trigger would be

```
create trigger=1 cpu=85 direction=up script=capture.scp
```

If you want to run the script when a port goes down, the trigger would be

```
create trigger=1 port=<number> event=linkdown script=capture.scp
```

If you want to run the script when memory gets low, the trigger would be

```
create trigger=1 memory=10 direction=down script=capture.scp
```

If you want to run the script when a session with a BGP peer goes down, the trigger would be

```
create trigger=1 module=bgp event=peerstate peer=<peer-address>  
state=established direction=leave script=capture.scp
```

If you want to run the script when the load balancer fails to find a required resource, the trigger would be

```
create trigger=1 module=lb event=lastfail respool=<pool-name>  
script=capture.scp
```

4. Examine the file debug.txt

The file **debug.txt** will contain the output of the commands from your **debug.scp** script. Each time the script runs, it writes the new output to the end of the file.

You can view the file on the router, or upload it to a PC.

Run the script and send the output to a syslog server

Alternatively (or on older releases), you can send output to a syslog server for storage and analysis.

1. Create a script that runs the relevant “show” commands. Call it debug.scp

The contents of the script could be something like:

```
-----  
show debug  
wait 30  
show swi port=xyz count  
show ip count=arp  
show swi fdb  
...  
-----
```

In other words, include whatever **show** commands have been identified as capturing the evidence required to pinpoint what is going on when the problem occurs.

2. Create a trigger that runs the debug script at the relevant event

The following commands need to go into the boot script on the device.

Enable the trigger module, by using the command:

```
enable trigger
```

Create the relevant trigger. If you want to run the script immediately after a reboot, the trigger would be

```
create trigger=1 reboot=all script=capture.scp
```

If you want to run the script when the CPU gets very busy, the trigger would be

```
create trigger=1 cpu=85 direction=up script=capture.scp
```

If you want to run the script when a port goes down, the trigger would be

```
create trigger=1 port=<number> event=linkdown script=capture.scp
```

If you want to run the script when memory gets low, the trigger would be

```
create trigger=1 memory=10 direction=down script=capture.scp
```

If you want to run the script when a session with a BGP peer goes down, the trigger would be

```
create trigger=1 module=bgp event=peerstate peer=<peer-address>  
state=established direction=leave script=capture.scp
```

If you want to run the script when the load balancer fails to find a required resource, the trigger would be

```
create trigger=1 module=lb event=lastfail respool=<pool-name>  
script=capture.scp
```

3. Take all log output of type 'batch' and send it to a syslog server

The output of a script that is called from a trigger can be sent to the log. The particular log type is 'batch'. So, we want to take all log output of type 'batch' and send it to a syslog server.

Add the following lines to the boot script:

```
create log out=2 destination=syslog secure=yes
    server=<syslog-server-IP-address>
add log out=2 type=batch
```

4. View the syslog server

The output sent to the syslog server will be of the form:

```
-----
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT, SCR, show debug
Dec 29 14:59:20 193.6.241.190 TRG:BATCH/ACT, Trigger 1 activated (Automatic)
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT, SHOW SYSTEM
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT,
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT, Switch System Status
    Time 14:59:16 Date 29-Dec-2004.
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT, Board    ID Bay Board Name
    Rev      Serial number
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT,
-----
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT, Base    186    AT-9812T
    M4-3    61553784
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT
-----
Dec 29 14:59:20 193.6.241.190 SCR:BATCH/OUT, Memory-DRAM :131072 kB FLASH : 16384
    kB
...
-----
```

So, it has all this “Dec 29 14:59:20 193.6.241.190 SCR:BATC**H**/OUT;” stuff at the start of each line.

If the syslog file is on a Linux PC, then you can remove the unnecessary stuff at the start of each line with the following bash command:

```
sed s%^.*BATCH/..T,%% logfile > newlogfile
```

In this command, “logfile” is the name of the syslog file that contains the original output, and “newlogfile” is the name that you want to give to the file containing the output with the “Dec 29 14:59:20 193.6.241.190 SCR:BATC**H**/OUT;” stuff taken out.

How to upgrade the GUI when upgrading to Software Version 2.8.1

The naming convention for GUI resource files changed from Software Version 2.8.1 onwards. Names are now longer, and they include more information about the router or switch model and software version to which the file applies. For example, the GUI resource file for AT-9924 Series switches for Software Version 281-01 is 9924_281-01_en_d.rsc.

Software versions before 2.8.1 do not recognise the new GUI name format. This changes the upgrade process slightly.

Products this Tip applies to

All routers and switches listed on page 1 that have GUI support and run the versions below.

Software Versions

2.8.1 or later

Previous approach With earlier software versions, you could upgrade the release file and the GUI file at the same time, by using the following steps:

1. Load each file onto the router or switch.
2. Set both files as the preferred install files, by using the command:

```
set install=pref rel=new-rez-file gui=new-gui-file
```
3. Reboot the router or switch.

New approach The first time you upgrade to a 2.8.1 version, you need to install the release before the GUI, by using the following steps:

1. Load each file onto the router or switch.
2. Set the new release file as the preferred install file and uninstall the previous GUI file, by using the command:

```
set install=pref rel=281-rez-file gui=
```
3. Reboot the router or switch.
4. Set the new GUI file as the preferred GUI file, by using the command:

```
set install=pref gui=281-gui-file
```

If you upgrade from a 2.8.1 version to a later 2.8.1 maintenance version, you can install the release and GUI in the same step.

Also, some TFTP servers do not support filenames longer than 8 characters and therefore will not allow you to load the file from the server. With such servers, you can simply rename the GUI file to a short name on the TFTP server, then rename it correctly on the router or switch.

A simpler way to save the current configuration

With Software Versions 2.8.1 and 3.1.1, it became simpler to save the current configuration and set the router or switch to use it on reboot. You can now do this in a single step, by using the command:

```
create config=filename set
```

This is an alternative to the previous 2-step approach of:

```
create config=filename  
set config=filename
```

However, the command **set config** is still valid, and the **set** parameter is optional in the command **create config**.

Products this Tip applies to

All routers and switches listed on page 1 that run the versions below

Software Versions

3.1.1 or later for AT-9924Ts and x900-24 Series switches

2.8.1 or later for other products

How to store stack dump files

Since Software Version 2.7.6, the router or switch stores any stack dumps that occur. Before this feature, you could only view the most recent stack dump (by using the command **show debug stack**). The new storage feature has the following major advantages:

- It prevents the stack dumps from becoming lost after a router or switch is rebooted.
- It lets support staff access more than just the most recent stack dump.
- It make it easier to send stack dumps to support engineers and ATL NZ.

Products this Tip applies to

All routers and switches listed on page I that run the versions below

Software Versions

2.7.6 or later

The storage method

Stack dumps are now saved in Flash memory as files named:

```
dmex<number>.txt
```

<number> increments, and the router or switch stores up to 8 dmex files. The following figure shows a file list for a switch with 8 stack dumps stored.

Filename	Device	Size	Created	Locks
boot.cfg	flash	4785	15-Sep-2005 01:36:46	0
config.ins	flash	32	16-Nov-2005 01:24:31	0
default.ins	flash	64	16-Nov-2005 01:22:41	0
dmex0001.txt	flash	5262	14-Dec-2005 16:20:09	0
dmex0002.txt	flash	4215	13-Feb-2006 14:48:35	0
dmex0003.txt	flash	5262	15-Feb-2006 15:48:15	0
dmex0004.txt	flash	4215	16-Feb-2006 15:55:09	0
dmex0005.txt	flash	4388	17-Feb-2006 16:23:22	0
dmex0006.txt	flash	4215	18-Feb-2006 17:38:17	0
dmex0007.txt	flash	3688	19-Feb-2006 20:25:48	0
dmex0008.txt	flash	4215	20-Feb-2006 21:16:59	0
.				
.				
.				

The file name would continue to increment with each new stack dump until dmexffff.txt is reached. After this it would roll back to dmex0001.txt.

As mentioned previously, only a maximum of 8 files can be stored in flash at one time. As subsequent stack dumps occur, the oldest one is replaced. For example, dmex0001.txt will be replaced by dmex0009.txt, as the following figure shows.

Filename	Device	Size	Created	Locks
boot.cfg	flash	4785	15-Sep-2005 01:36:46	0
config.ins	flash	32	16-Nov-2005 01:24:31	0
default.ins	flash	64	16-Nov-2005 01:22:41	0
dmex0002.txt	flash	4215	13-Feb-2006 14:48:35	0
dmex0003.txt	flash	5262	15-Feb-2006 15:48:15	0
dmex0004.txt	flash	4215	16-Feb-2006 15:55:09	0
dmex0005.txt	flash	4388	17-Feb-2006 16:23:22	0
dmex0006.txt	flash	4215	18-Feb-2006 17:38:17	0
dmex0007.txt	flash	3688	19-Feb-2006 20:25:48	0
dmex0008.txt	flash	4215	20-Feb-2006 21:16:59	0
dmex0009.txt	flash	4215	22-Feb-2006 09:30:02	0
.				
.				
.				

What to do with dmex files

A dmex file is just a text file, so you can view it on screen. However, most of the file is in hexadecimal code rather than in human-readable format.

Therefore, we recommend you upload the dmex files from the router or switch to your PC. International support engineers can then send them to ATL NZ for decoding.

At ATL NZ, we would now prefer to receive stack dumps in the form of the uploaded dmex file. This guarantees that we have the entire stack dump information, and reduces the risk of corrupt stack dumps.

How to securely manage remote devices from an Asyn (console) port on a router or Rapier

This Tip describes how to manage remote devices (such as other Allied Telesis routers or switches, or modems) that are connected to a router or Rapier switch back-to-back via Asyn ports.

The following steps summarise this process. For more information about services and the associated commands, see the Terminal Server chapter of your router or switch's Software Reference.

Products this Tip applies to

AR400 and AR700 Series routers
Rapier 24 and Rapier 24i switches

Software Versions

All that support these products

1. Connect the devices you want to manage

First, ensure that you have enough Asyn ports, and connect everything together. For more information about cables and connectors, see your router or Rapier's Hardware Reference.

Some Allied Telesis routers come with two Asyn ports. If you want to manage more devices remotely, or if you have a router or switch with only one Asyn port, you will need one or more AT-AR024-00 PICs. Each PIC provides four Asyn ports.

Note that the Asyn ports on the PIC use a different pinout from some other industry equipment. See the PIC Hardware Reference for details of the pinout. This Hardware Reference is on your router or Rapier's Documentation and Tools CD-ROM, or you can download it from www.alliedtelesyn.co.nz/documentation/manuals.html

2. Create a service

To manage a remote device, we will create a service and then connect to the service. You can instead connect directly via an Asyn port, but creating a service lets you give it a meaningful name instead of having to remember the port number. In this example, we call our service by the generic name `asyn1`. Replace this with a name that describes the remote device.

To create a service:

1. Access your router or Rapier via a console port, or use SSH or Telnet to connect to it. We recommend SSH instead of Telnet because it is encrypted.
2. Set up a service (in this example, to the device attached to `asyn1`) by using the following commands on your router or Rapier:

```
create service=asyn1 type=interactive descr="Asyn 1 Connection"  
set asyn=1 speed=value data=value service=asyn1
```

3. Connect to the remote device

Connect to the remote device (in this example, the device attached to `asyn1`) by using the command:

```
c asyn1
```

Note that the letter **c** is the shortest valid string for the command **connect**.

The router displays the following message:

```
Info (1036266): Local port ( Asyn 0 ) assigned to service ( asyn1 ).
```

You are now connected to `asyn1`. Note that you did not need to use reverse telnet. As long as you have the right cables and Asyn port settings, simply connecting like this will work.

4. Clear the command line on the remote session

When you connect, an info message displays, as described above. If you press the Enter key, that message will be sent to the remote session, which will produce an error. Therefore, you need to clear the line before you type any commands.

To clear the line if the remote device is an Allied Telesis router or switch, press Ctrl+u.

5. Enter commands on the remote device as required

Configure or monitor the remote device by entering commands into the remote session.

6. Pause the remote session

After you have finished using the remote device, you need to close the Asyn connection. To do this, first send a Break command. The Break command depends on the application that you have used to connect to the device. For example, for minicom use Ctrl+p or Ctrl+a f. For Putty (a Windows SSH client) use Ctrl+p.

The router or Rapier responds by displaying the following message:

```
Session 1 to asyn1 paused
```

You can customise the Break command on a router or Rapier by using (for example) the command

```
set asyn attention=^P
```

This example sets the Break command to Ctrl+p. However, note that changing the Break command can have unpredictable results. Your console, SSH or Telnet client may interpret these characters as other commands.

7. Disconnect from the remote session

Sending a Break command does not actually disconnect from the Asyn port. Therefore, the router or Rapier still thinks it has a connection to `asyn1`. You must disconnect after sending the Break, by using the command:

```
d 1
```

Note that the letter **d** is the shortest valid string for the command **disconnect**. You need to disconnect the specific session—in this example, session number 1. You can tell the session number from the message you get when you pause the session.

The router displays the following message:

```
Host port ( asyn1 ) free
Info (1036275): Disconnected from session 1 ( asyn1 ).
```

8. Connect to the remote session again

If you disconnected as shown in step 7, you can reconnect by using the command:

```
c asyn1
```

If you did not disconnect, but only paused the session as shown in step 6, you can reconnect to the existing session by using the command:

```
r asyn1
```

Note that the letter **r** is the shortest valid string for the command **reconnect**.

How to reduce the impact of storms by using QoS policy storm protection

Software Versions 2.8.1 and 3.1.1 let you use QoS mechanisms to classify on traffic likely to cause a packet storm (broadcast and multicast) and limit the storm's impact.

Before this feature, you could only control broadcast or multicast storms by setting a limit on a port. Once the limit was reached, any more broadcast or multicast traffic was discarded. The new feature gives you more flexibility in determining whether a storm has occurred, and different options for what action the switch takes once the limit is reached.

Products this Tip applies to

AT-8948 switches

AT-9900 Series switches

AT-9924Ts switches

x900 Series switches

Software Versions

3.1.1 or later for AT-9924Ts and

x900-24 Series switches

2.8.1 or later for other switches

The different storm actions possible are:

- **portDisable**
The port is logically disabled for a period of time, which prevents traffic flow, but the link remains up. The device at the other end does not notice that the port has changed status, and the link LEDs at both ends stay on. This is the default.
- **linkDown**
The port is physically disabled and the link goes down for a period of time.
- **vlanDisable**
The port is disabled for a period of time, but only for traffic on the VLAN on which the storm has occurred. The port can still receive and transmit traffic for any other VLANs that it is a member of.

When the switch detects a storm on a port, it automatically records a message in the log. You can also configure an SNMP trap to signal that a port has been disabled.

When the switch detects on a trunk or port group, it takes the configured action on the entire trunk or port group.

Configuring storm protection

Note that enhanced mode must be enabled before you can configure storm protection. To enable it, use the command:

```
set switch enhancedmode=qoscounters
```

Storm protection is an aspect of traffic classes. You can configure it on user-defined traffic classes and on the default traffic class. If you configure it on traffic classes that you create, this lets you create classifiers for some or all types of broadcast or multicast traffic and set limits for that traffic. If you configure it on the default traffic class, that applies the limits to all traffic that isn't classified into another traffic class.

The following table explains the basic concepts involved with storm protection.

Concept	Description
Window	How often matching traffic is measured to determine whether storm protection should be activated.
Rate	The amount of matching traffic per second that must be exceeded before the switch takes the configured action.
Action	What the switch does when it detects a storm on a port, as described above.
Timeout	The length of time the port remains disabled after a port has been disabled due to a packet storm.

To create a policy and enable storm protection on its default traffic class, use the command:

```
create qos policy=id-list [dtestormstatus={enable|disable}]
[dtestormwindow={window-size|none}] [dtestormrate={rate|none}]
[dtestormaction={linkdown|portdisable}]
[dtestormtimeout={timeout-length|none}]
[other-parameters]
```

You can use the same parameters in the **set qos policy** command to modify an existing policy's default traffic class.

To enable storm protection on a traffic class when you create one, use the command:

```
create qos trafficclass=trafficclass-list
[stormstatus={enable|disable}] [stormwindow={window-size|none}]
[stormrate={rate|none}]
[stormaction={linkdown|portdisable|vlandisable}]
[stormtimeout={timeout-length|none}]
[other-parameters]
```

You can use the same parameters in the **set qos trafficclass** command to modify an existing traffic class.

Re-enabling ports

Unless the timeout is set to **none**, the switch enables the port again when the timeout expires. If the timeout is **none** or you want to enable the port sooner, you have the following choices for manually re-enabling it:

- use the command **enable switch port={*port-list*|all}**
- use the command **enable switch port={*port-list*|all} vlan**, if the **stormaction** is **vlandisable**
- use SNMP
- restart the switch

How to reduce the impact of storms by controlling rapid MAC movement

Software Versions 2.8.1 and 3.1.1 let you limit the impact of rapid MAC movement (MAC address thrashing). MAC address thrashing occurs when MAC addresses move rapidly between one or more ports or trunks, for example, because of a network loop.

Before this feature, you could only control MAC address thrashing by setting a broadcast limit on a port. Once the limit was reached, any more broadcast traffic was discarded. The new feature gives you other options for what action the switch takes once the MAC address limit is reached.

The different actions possible when thrashing occurs are called thrash actions, and are:

- **learnDisable**
Address learning is disabled on the port for a period of time. This is the default, and is approach taken before the new feature became available. Note that this option is the only one that lets the switch continue to forward unicast and multicast traffic over the port.
- **portDisable**
The port is logically disabled for a period of time, which prevents traffic flow, but the link remains up. The device at the other end does not notice that the port has changed status, and the link LEDs at both ends stay on.
- **linkDown**
The port is physically disabled and the link goes down for a period of time.
- **vlanDisable**
The port is disabled for a period of time, but only for traffic on the VLAN on which thrashing has occurred. The port can still receive and transmit traffic for any other VLANs that it is a member of.

When a MAC address is thrashing between two ports, only one of those ports is disabled. When multiple ports are involved, enough ports are disabled to prevent the storm.

Configuring rapid MAC movement protection

To set a thrash action for one or more ports, use the command:

```
set switch port={port-list|all} thrashaction={learndisable|linkdown|none|portdisable|vlandisable} [thrashtimeout={none|1..86400}]
```

To set a thrash action for a trunk, use the command:

```
create switch trunk=name [port=port-list] thrashaction={learndisable|linkdown|none|portdisable|vlandisable} [thrashtimeout={none|1..86400}]
```

Products this Tip applies to

AT-8948 switches
AT-9900 Series switches
AT-9924Ts switches
x900 Series switches

Software Versions

3.1.1 or later for AT-9924Ts and x900-24 Series switches
2.8.1 or later for other switches

To set a thrash action for a trunk, use the command:

```
set lacp priority=priority thrashaction={learndisable|linkdown|
none|portdisable|vlandisable} [thrashtimeout={none|1..86400}]
```

The **thrashtimeout** parameter in the above commands sets how long the switch applies the action for. After that length of time, the port is re-enabled. The default is 1 second. Note that you have to set the timeout on the individual ports in a trunk as well as the trunk.

As well as the above commands, you can also globally set the threshold at which the switch considers that MAC addresses are thrashing. To do this, use the command:

```
set switch thrashlimit=5..255
```

This command sets the maximum number of times a MAC address can move between ports in one second before the switch takes action. The default is 10 times.

Re-enabling ports

Unless the timeout is set to **none**, the switch enables the port again when the timeout expires. If the timeout is **none** or you want to enable the port sooner, you have the following choices for manually re-enabling it:

- use the command **enable switch port={port-list|all}**
- use the command **enable switch port={port-list|all} vlan**, if the **thrashaction** is **vlandisable**
- use SNMP
- restart the switch

Note that you cannot manually re-start learning if the **thrashaction** is **learndisable**, except by restarting the switch.

How to use SNMP to monitor STP and RSTP links

Spanning Tree Protocol is used to prevent network loops and provide resilience so that if a link or unit fails the network automatically recovers connectivity. One important point about this is that you need to be alerted if there is a failure in the network that causes the backup link to cut in. Otherwise you may only learn of the first failure when the second link in the loop fails and the network is broken.

Products this Tip applies to

All switches listed on page 1

Software Versions

All that support STP or RSTP

One way to find out about broken links is to set up monitoring and alarms based on the STP state change traps. However, if the network becomes disconnected from the SNMPc server during the period these traps are sent, you can miss the event and fail to raise the relevant alarms.

Alternatively, you can poll the link status of the port at one end of each link of the STP or RSTP loop, and configure an alarm to warn you if any of the links go down. This Tip shows you how to use the program SNMPc (www.snmpc.com) to do this.

Note that this solution does not tell you the STP port state of each port (such as blocking or forwarding)—instead it notifies you of system failures that cause a backup link to cut in.

1. Record the network topology

List the units that take part in the STP or RSTP loop. Note the specific ports on each unit. Make a network drawing and include all the port information on this.

2. Configure SNMP on each switch in the STP or RSTP loop

This solution polls ports on switches, so the switches have to be running SNMP—you cannot just use Ping to check their status. We recommend using SNMPv3 for security.

3. In SNMPc, position the devices in the network to match their physical location

In SNMPc, you can set up your network by running SNMPc's auto-discovery feature. This locates every device in the network.

However, it places all the devices in a subnet into a “bus network” layout. This does not look much like your physical network, and does not give you much flexibility to move objects to where you want them. Therefore, replace this bus layout with the physical layout of your network by:

1. Go to the map view (View menu > Root Submap) and expand the root to see the whole network
2. Click on the root network's line in the map to select it, then delete it
3. Place the now-unconnected device icons where they are logically in your network, to make a map that is clear to you

4. Link devices together

Now that you have placed the devices in the network, you need to link the devices together.

If a device is not in an STP loop, to link it to another device:

1. select the pair of devices (Ctrl + Click)
2. from the Insert menu, select Map Object > PtoPlink. This draws in a simple line connection.

For devices that are connected together and taking part in STP or RSTP, you need to connect them together with a “network” line that you can poll, instead of a simple link. To do this:

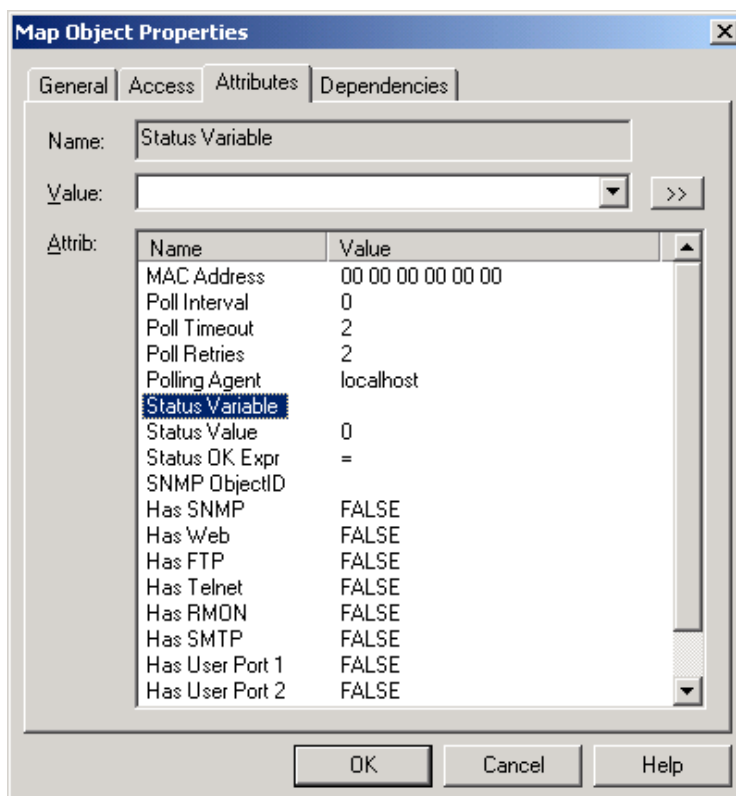
1. from the Insert menu, select Map Object > Network
2. enter a label for the line (such as “SwitchAToSwitchB”) and the IP address of the switch that you want to poll for the link status. Click OK to draw the line on your map
3. move the line to the desired position between the devices by dragging and dropping
4. select the line and the two devices (Ctrl + Click)
5. from the Insert menu, select Map Object > PtoPlink. This joins the line to the devices

5. Set the poll variable for each new segment

Right-click on the new network line and select Properties. Click the Attributes tab.

Select the entry named Poll interval and set it to the desired interval depending on how quickly you need to know of a failure (1 minute is a good interval for many networks).

Select the entry named Status Variable, as shown in the following figure.

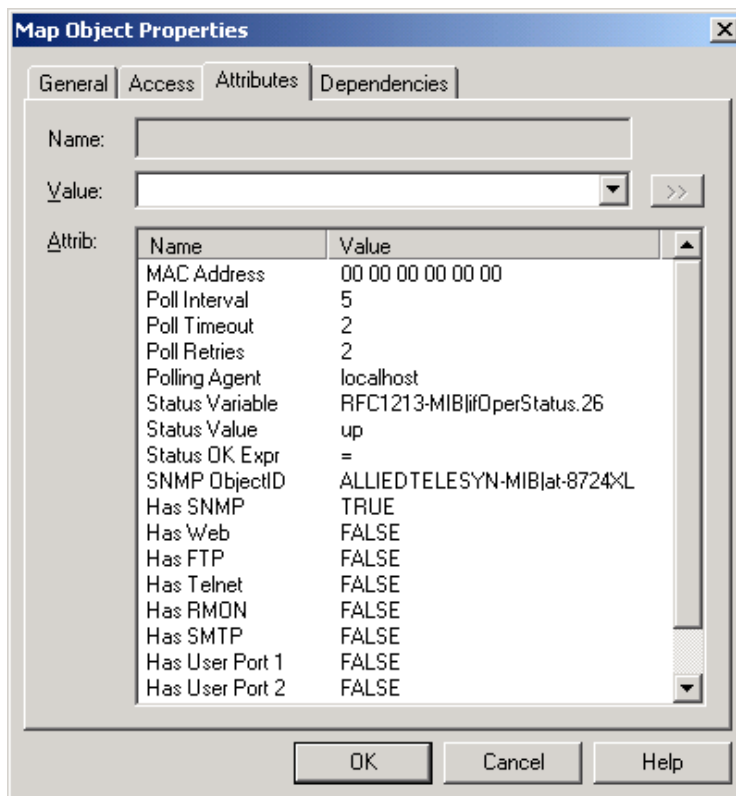


Set the Status Variable as follows:

1. click on the >> button (next to the Value field) and browse to mgmt > Interfaces > IfTable > IfOperStatus
2. click OK to return to the Attributes dialog box
3. at the end of the Value, enter a dot and the index number for the port you want to poll. Note that this may not be the same as the port number—you may need to use the command **show interface** and check the ifIndex value to see how the index matches to the port number

Select the entry named Status Value and set it to “up”.

The final Attributes list should be similar to the following figure.

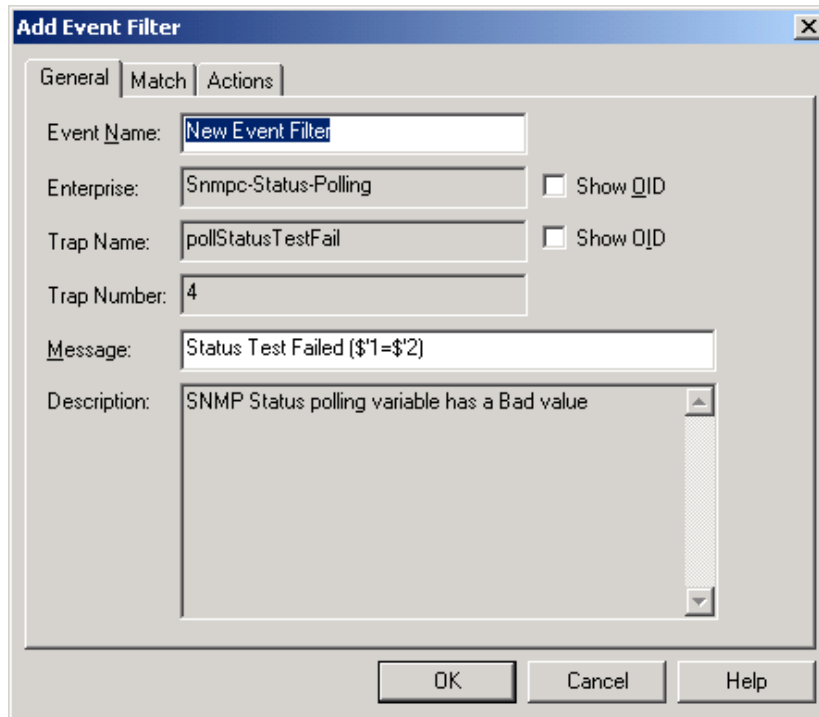


Click the OK button to accept the changes.

Repeat this step for each new line in your STP or RSTP loop.

6. Configure the alarm for each new segment

Now configure the alarms to alert you if a link fails in the STP loop. On the left of the screen, select the Event tab. Then navigate to Event Actions > SNMPc-Status-Polling > pollStatusTestFail. Right-click on this and select Insert Event Filter. The Add Event Filter dialog opens, as shown in the following figure.



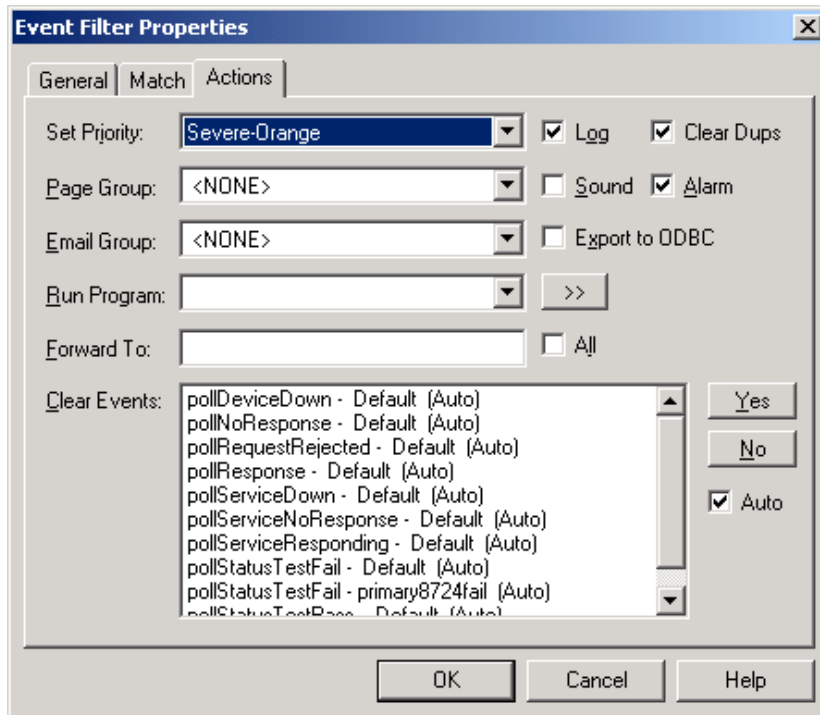
The screenshot shows the 'Add Event Filter' dialog box with the following configuration:

- Event Name: New Event Filter
- Enterprise: Snmpc-Status-Polling (Show QID checkbox is unchecked)
- Trap Name: pollStatusTestFail (Show QID checkbox is unchecked)
- Trap Number: 4
- Message: Status Test Failed (\$'1=\$'2)
- Description: SNMP Status polling variable has a Bad value

Enter a name for the filter (such as “SwitchAToSwitchBDown”) and enter a suitable message (such as “Link between A and B is down (see map)”).

Next select the Match tab. Click on the Add button and select the Show Ports checkbox. This lists all network ports and links, including the new network links that you added and named above. Select your chosen link (“SwitchAToSwitchB” in this example) and click OK to return to the Match dialog box.

Finally select the Actions tab. Here you select a colour for the STP link to change to if it fails (we chose orange, as shown in the following figure). You can also select other options such as logging, alarms, and sound. Click OK to save the changes.



The Help in this screen is very useful and gives a lot more detail on customising messages and event filtering. The main SNMPc manual describes how to set up mail etc so you can be alerted by this method too.

Repeat this step for each new line in your STP or RSTP loop.

When you have finished, your network map should show each STP link as a fat green line when the link is up and a fat orange line when the link is down.

How to use SNMP to monitor master and slave SwitchBlade controller cards

If you have a SwitchBlade with two switch controller cards, it is important that you are notified if one card fails. This Tip shows you how to use the program SNMPc (www.snmpc.com) to poll for the presence of the two switch controllers and receive an alert when one of them goes down. The Tip also shows you how to be alerted if a PSU or fan fails.

Products this Tip applies to

SwitchBlade switches

Software Versions

All that support SwitchBlade

The polled variables

You need to poll the following variables from the AlliedTelesyn MIB:

```
arSlotHeldBoardIndex.2.9
```

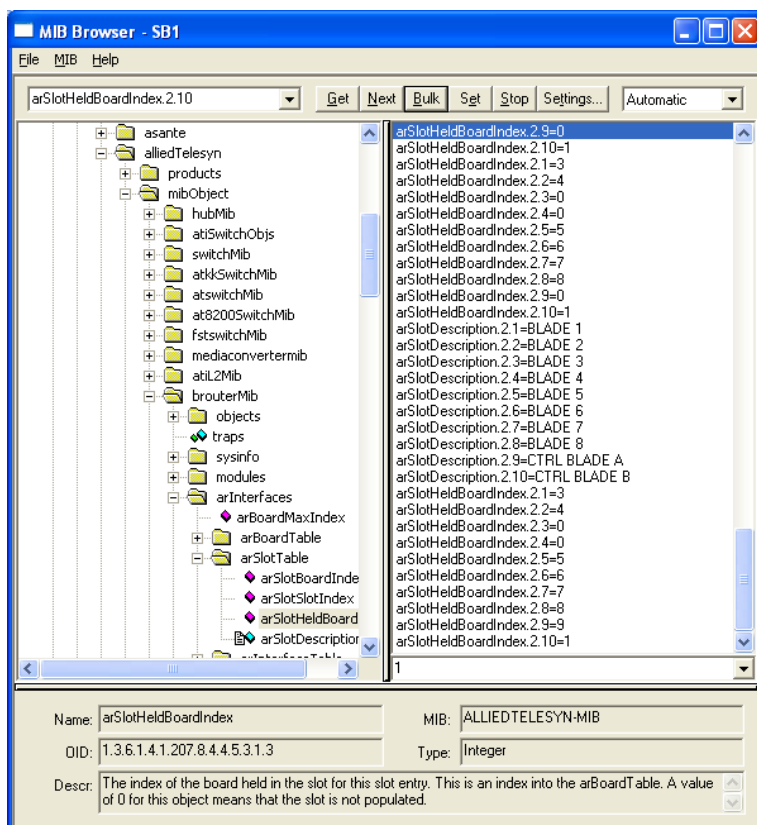
```
arSlotHeldBoardIndex.2.10
```

These variables return one of the following values:

Value	Meaning
0	The card is not present, or has failed
1	The card is the Master
9	The card is the Slave

The variables are found, as the following figure shows, at:

```
/Private/AlliedTelesyn/MIBObject/routerMib/arInterfaces/arSlotTable
```



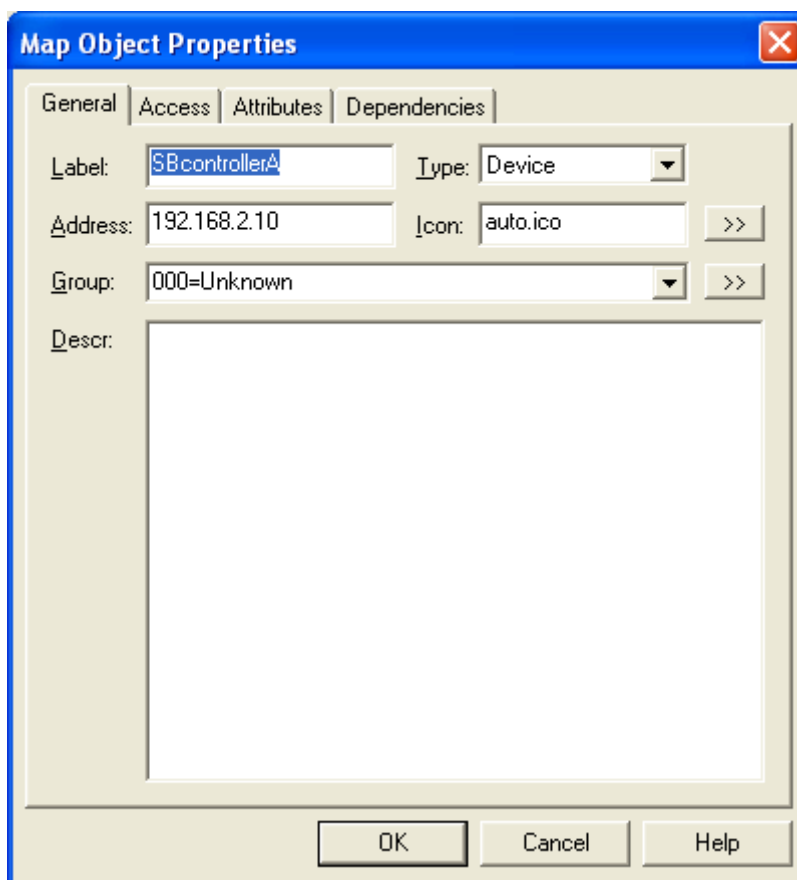
Creating an alert when a controller fails

Perhaps the easiest way to create an alert system is to create two *additional* icons on the SNMP map, one for each controller card. If you set these up with the SwitchBlade's IP address, then you will see two identical SwitchBlade icons on the SNMPC map view. The following instructions describe how to do this.

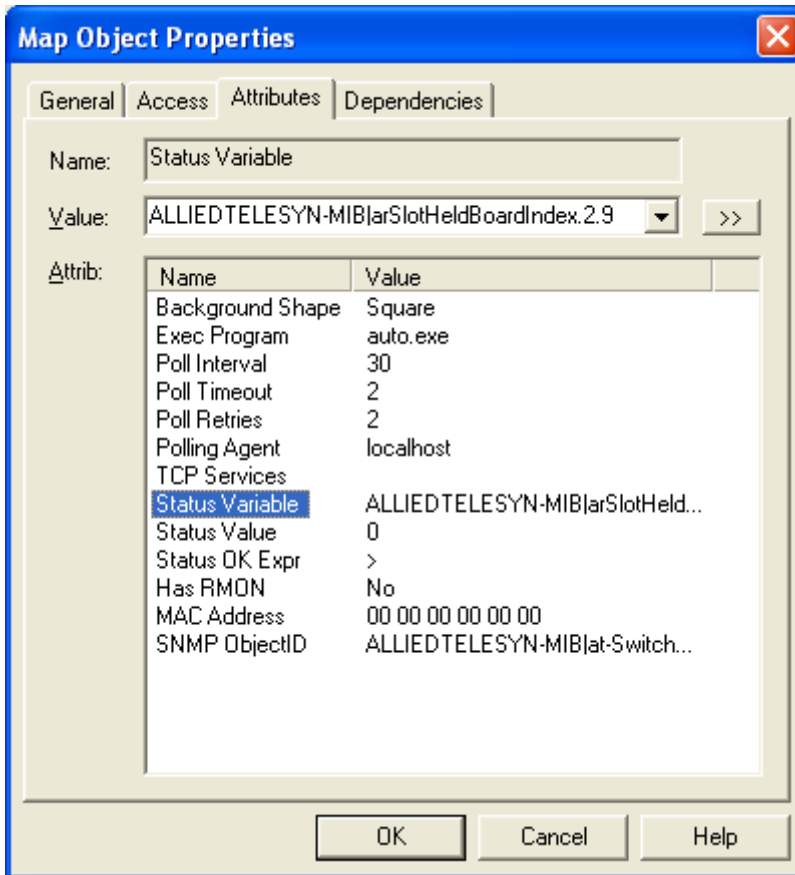
However, make sure your main SwitchBlade icon is already there and remains, because trap information should be reported from it.

I. Create an icon for switch controller A

From the menu at the top of SNMPC, select Insert > Map Object > Device and enter the SwitchBlade details, as shown in the following figure.



Then select the Attributes tab. On the Attributes tab, select the entry called Status Variable and click on the >> button. Browse to the variable arSlotHeldBoardIndex and click OK. At the end of the Value, add .2.9 as shown in the following figure.



Next, select the entry called Status OK Expr and enter > (instead of its default value of =).

Click OK. Your icon for the presence of switch controller A will be on the map.

2. Create an icon for switch controller B

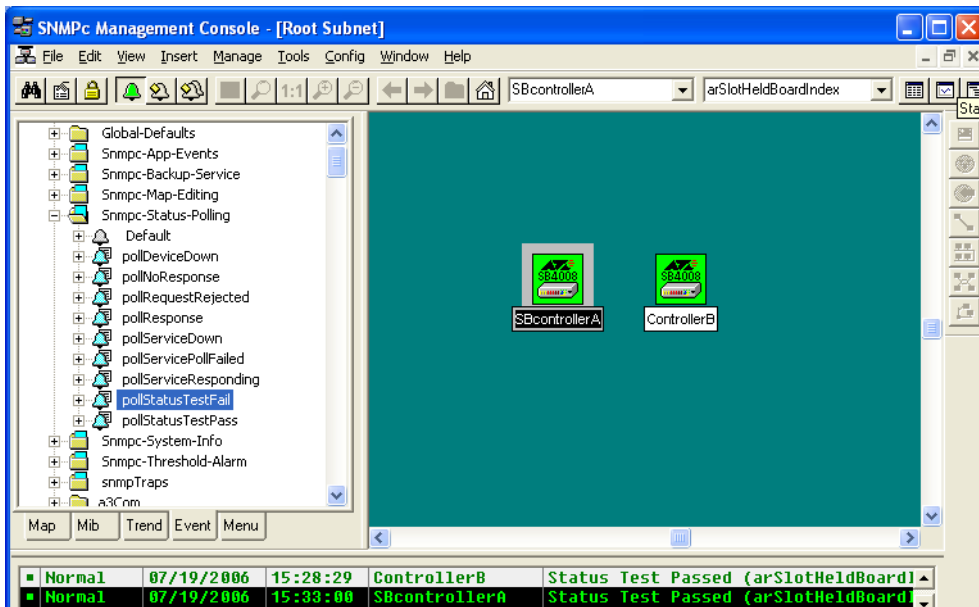
Repeat the step above, but use ControllerB for the name and insert arSlotHeldBoardIndex .2.10 as the Status Variable.

This creates an icon for displaying the status of controller B.

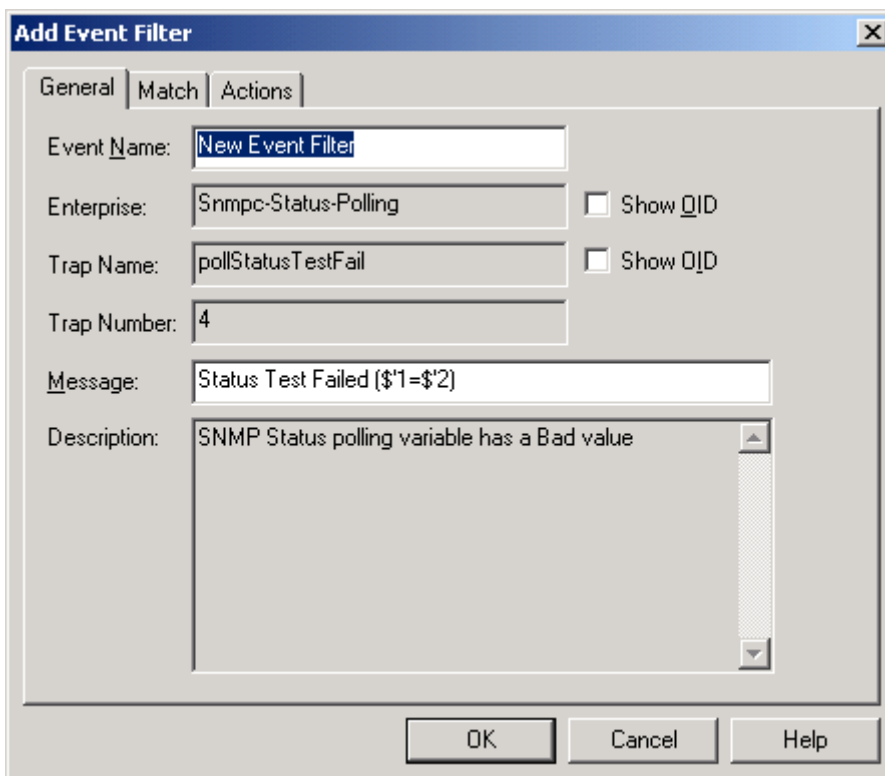
3. Create an alert for failure of a switch controller

The final stage is to add alarms to alert you on screen if a SwitchBlade controller fails.

On the left of the screen, select the Event tab. Then navigate to Event Actions > SNMPc-Status-Polling > pollStatusTestFail, as shown in the following figure.

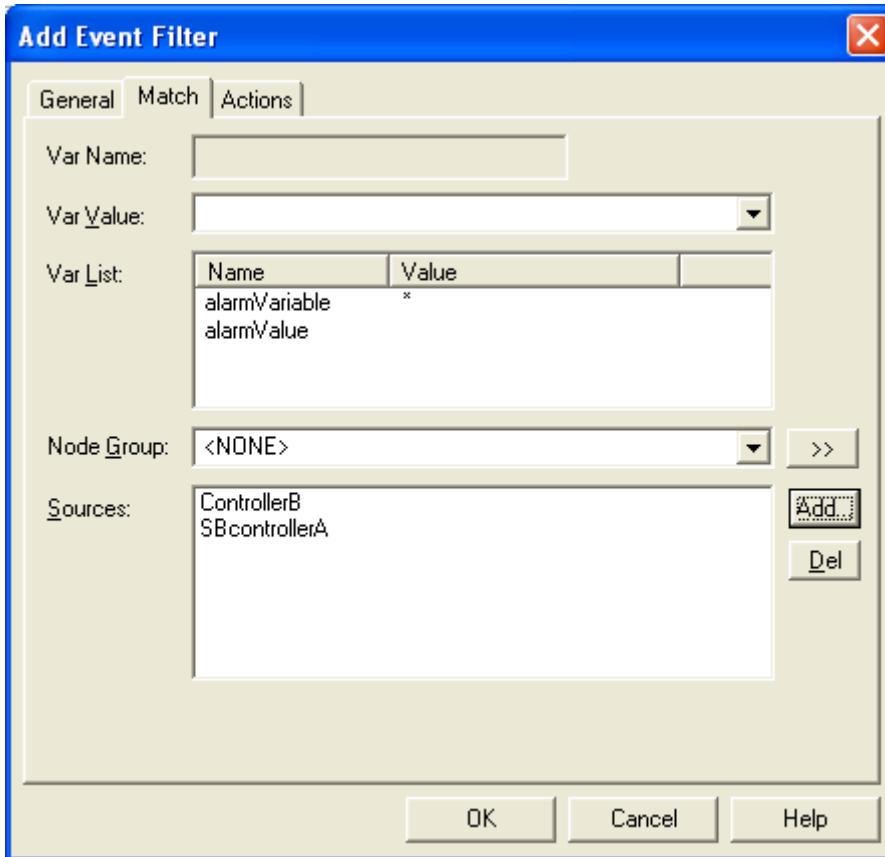


Right-click on pollStatusTestFail and select Insert Event Filter. The Add Event Filter dialog opens, as shown in the following figure.

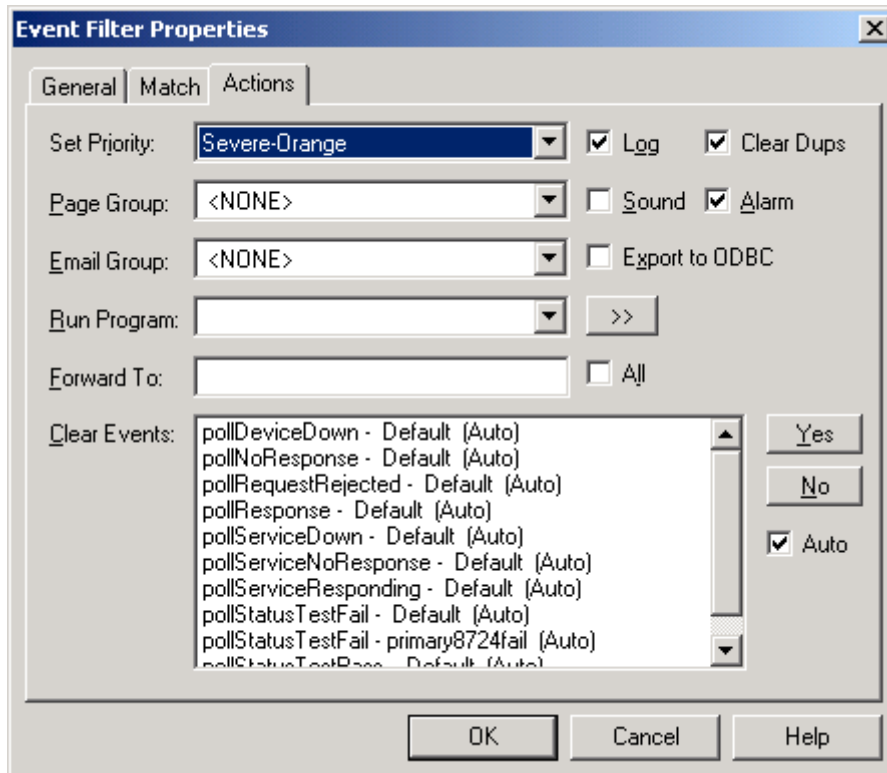


Enter a name for the filter event, such as “SB controller failure” and enter a suitable message, such as “SwitchBlade Controller card failure”.

Next select the Match tab. Click on the Add button and select the Show Ports checkbox. This will now display the new icons that you added and named above. Select them and click the OK button to return to the Match dialog box as shown in the following figure.



Finally select the Actions tab. Here you select a colour to indicate controller failure (we chose orange, as shown in the following figure). You can also select other options such as logging, alarms, and sound. Click OK to save the changes.



The Help in this screen is very useful and gives a lot more detail on customising messages and event filtering. The main SNMPc manual describes how to set up mail etc so you can be alerted by this method too.

Setting up traps to notify when a fan or PSU fails

Notification of FAN and PSU status changes can be done effectively via traps.

I. Configure the SwitchBlade

First, make sure that your SNMP configuration on the SwitchBlade has traps enabled and a trap host set to the SNMPc management station address. As a minimum this should include the following commands:

```
enable snmp
create snmp community=public open=on
enable snmp community=public trap
add snmp community=public trap host=host-ip-add
```

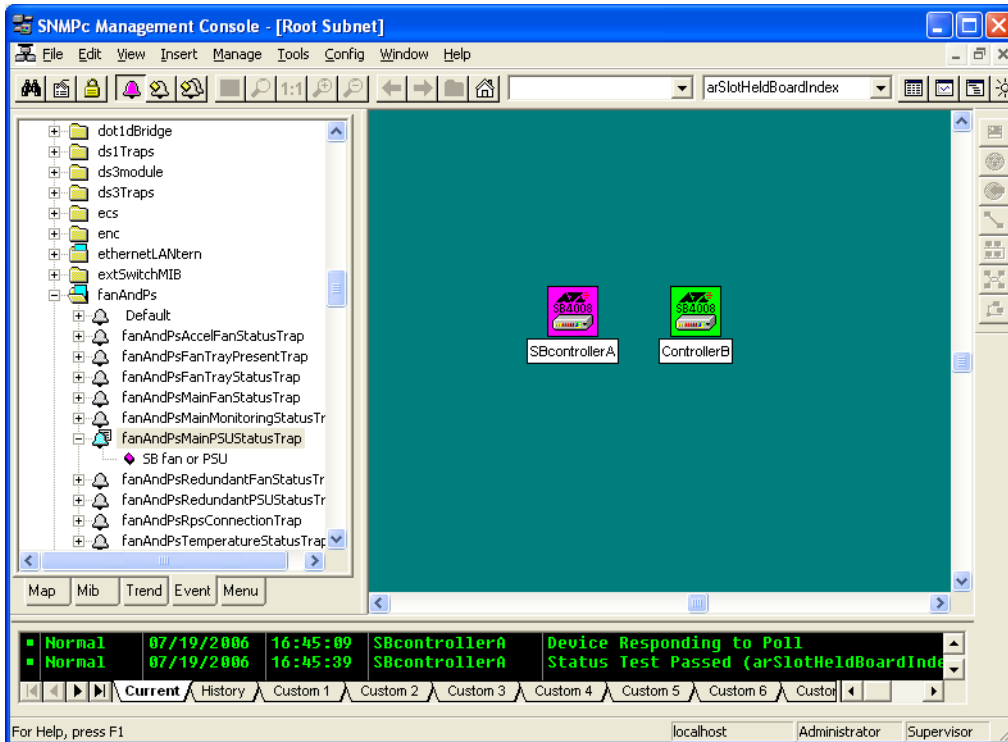
Once this is done, traps for any changes in PSU and fan status will be sent to SNMPc and will pop up in the event log window at the bottom of the screen.

Note that this configuration makes the SwitchBlade send all traps, not only PSU and fan status traps.

2. Create event filters for events for which you want to see popup box alerts

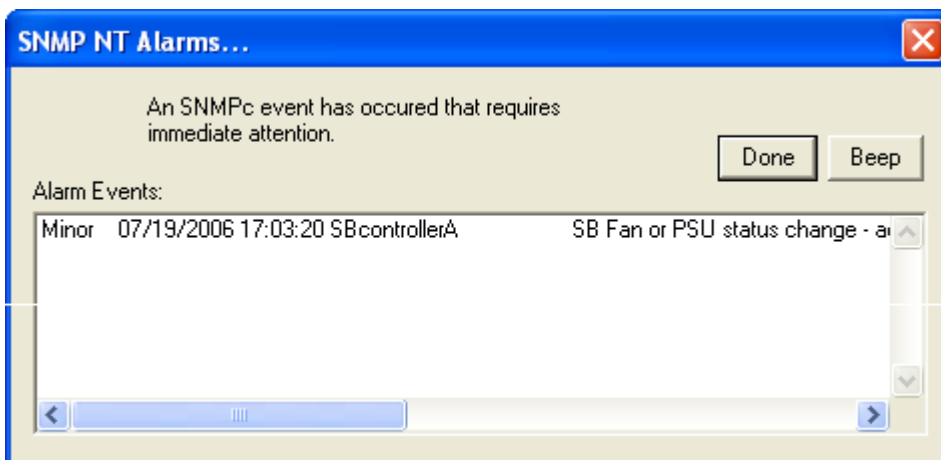
If there are some events for which you want to see alerts on the screen, then you can select the trap variable and add an event filter to flag this. For example, it is worth flagging a PSU failure. This will be sent as a fanAndPsMainPSUStatusTrap.

To create the event filter, go to the event window and browse down to fanAndPs. Expand this to find the fanAndPsMainPSUStatusTrap, as shown in the following figure.



You can then right-click this and add an Event filter in the same way as we did for the SwitchBlade controller failures in the previous section.

The result will be that if a PSU fails, you get a message on screen like the following figure.



Why you need to use an idle timer on a PPPoE link

On PPPoE interfaces, we recommend that you configure an idle timer with a very large value, such as 100000. This Tip explains why.

Products this Tip applies to

All routers listed on page 1

Rapier, AT-8800, AT-9800,

SwitchBlade, AT-8900, x900-48 and

AT-9900 Series switches

Software Versions

All that support PPPoE

Introduction: the effect of setting a PPP idle timer

Setting the idle timer on a PPP interface puts it into an “on-demand” mode. This causes the router to do the following:

- close the PPP link after a certain period of no traffic
- attempt to re-establish the link when there is traffic to send over the link.

The idea of an on-demand link was originally introduced to support links that were provided over dial-up services: ISDN, PSTN, or synchronous dial-up. If the user is paying for this service based on the length of uptime, then they wish to minimise the uptime, and so have the link closed when there is no traffic to send. However, they also need the link to re-establish automatically when there is traffic to send.

By contrast, when PPP is configured over an always-up Layer 1 service, like a synchronous leased line, the PPP link should be open whenever the Layer 1 link is up. Hence, PPP interfaces configured over always-up Layer 1 services do not come up based on traffic demand. Instead, when the Layer 1 link comes open, it indicates to the PPP layer to come open immediately.

The special case of PPPoE

PPPoE cannot quite be categorised as either a dial-up service or a service configured over an always-up Layer 1 link.

Instead, a PPPoE link requires the router to establish communication with a PPPoE access concentrator. The router can lose this communication with the access concentrator for a number of reasons, including the following:

- a break somewhere in the Ethernet link between the router and the access concentrator. This link could traverse multiple switches and even an ADSL link. It is quite possible for the link to break without any physical interface on the router itself going down—for example, a break can occur between other switches in the path between the router and the access concentrator
- A failure of the access concentrator itself. In this case, there is no break in the path to the access concentrator, but the access concentrator does not respond correctly to the packets sent from the router

Therefore, the router cannot rely on the state of its physical Ethernet interface to indicate whether it can still communicate with the access concentrator. Instead, the router needs to use a keep-alive mechanism such as LQR or ECHO to monitor the state of communication with the access concentrator. If the LQR or ECHO fails, then the router needs to bring down the link.

This leads to another question: if the link comes back up, how does the router bring the PPP interface back up again?

If PPPoE behaved like an always-up Layer 1 link, as described above, re-establishing the physical layer would trigger PPP to come open again. But PPPoE does not work like that, because the state of the locally connected Ethernet layer is not a reliable mechanism for deciding whether or not the communication to the AC is re-established. In fact, the only way a PPPoE interface can decide if communication to the access concentrator has been re-established is by keeping on trying to communicate with the access concentrator.

This means that a down PPPoE interface needs to behave like a dial-up interface—if it is down, it needs to try to re-establish communication to the access concentrator every time it has traffic to send. An idle timer forces this behaviour,

Therefore, we recommend you configure an idle timer on PPPoE interfaces. Set the idle timer to a very large value, such as 100000. That way, the link will effectively never idle out (because it is very unlikely to have 100000 seconds of no traffic), but will try to re-establish communication if the link goes down for any reason.

How to handle RIP route tags

RIP supports route tags from RIPv2 (RFC 1724). They are generally used as a way of separating internal RIP routes (routes for networks that are within the RIP routing domain) from external RIP routes (routes that have been learnt from an EGP or possibly from another IGP).

RFC 1724 states that “Routers that support routing protocols other than RIP should allow the route tag to be configured for routes imported from different sources”.

In other words, it should be possible to set an imported route’s tag to an arbitrary value, or at least to the AS number of the Autonomous System from which the routes were learnt.

It is valid to use RIP route tags in other ways so long as the usage is consistent on all routers throughout the RIP domain. The route tag itself is a 16 bit attribute of the RIPv2 entry table. In a RIPv1 packet this field is set to “0”.

The router or switch cannot be configured to add a RIP route tag to a RIP packet. However, if it receives any RIP packets that have a RIP route tag value set, it maintains the tag integrity and RIP forwards the routes with the tag intact.

Below is an ethereal capture of the RIP packet, showing where the RIP route tag is found in the packet.

The screenshot shows a Wireshark capture of a RIPv2 packet. The packet list pane shows two packets, both identified as RIPv2 Responses. The packet details pane for the selected packet (packet 2) shows the following structure:

- Frame 1 (86 bytes on wire, 86 bytes captured)
- Ethernet II, Src: 00:00:cd:1c:04:0c, Dst: 01:00:5e:00:00:09
 - Destination: 01:00:5e:00:00:09 (01:00:5e:00:00:09)
 - Source: 00:00:cd:1c:04:0c (Arlindie_1c:04:0c)
 - Type: IP (0x0800)
- Internet Protocol, Src Addr: 192.168.52.1 (192.168.52.1), Dst Addr: 224.0.0.9 (224.0.0.9)
- User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
- Routing Information Protocol
 - Command: Response (2)
 - Version: RIPv2 (2)
 - Routing Domain: 0
 - IP Address: 192.168.50.0, Metric: 3
 - Address Family: IP (2)
 - Route Tag: 1280
 - IP Address: 192.168.50.0 (192.168.50.0)

The packet bytes pane shows the raw data of the packet, with the route tag value 1280 (0x0500) highlighted in blue. The hex dump shows the route tag field at offset 0030: 05 00 c0 a8 32 00 ff ff ff 00 c0 a8 34 01 00 c0.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All, except for route map filtering which applies to 2.7.6 and later

The following figure shows output of the **show ip route** command from the router which first receives the RIP update that contains the RIP route tag. The tag is shown in bold.

```

IP Routes
-----
Destination      Mask      NextHop      Interface      Age
DLCI/Circ.      Type      Policy      Protocol      Tag      Metrics      Preference
-----
192.168.50.0     255.255.255.0  192.168.51.2  vlan1          29
-                remote    0            rip            1280    3            100
192.168.51.0     255.255.255.0  0.0.0.0       interface      -        1            0
-                direct    0            interface      -        1            0
192.168.52.0     255.255.255.0  0.0.0.0       interface      -        1            0
-                direct    0            interface      -        1            0
192.168.53.0     255.255.255.0  192.168.52.2  vlan2          2843
-                remote    0            rip            -        2            100
-----

```

The following figure shows output of the **show ip route** command from a downstream router. This shows that the route has been learnt and the RIP route tag has been preserved.

```

IP Routes
-----
Destination      Mask      NextHop      Interface      Age
DLCI/Circ.      Type      Policy      Protocol      Tag      Metrics      Preference
-----
192.168.50.0     255.255.255.0  192.168.52.1  vlan2          41
-                remote    0            rip            1280    4            100
192.168.51.0     255.255.255.0  192.168.52.1  vlan2          1299
-                remote    0            rip            -        2            100
192.168.52.0     255.255.255.0  0.0.0.0       vlan2          2868
-                direct    0            interface      -        1            0
192.168.53.0     255.255.255.0  0.0.0.0       vlan1          2868
-                direct    0            interface      -        1            0
-----

```

Using route maps to filter using RIP route tags

From Software Version 2.7.6, it is also possible to filter routes based on the RIP route tag by utilizing route maps. For example, if in the example above we want to prevent the 192.168.50.0 network from being advertised to a BGP peer, we can do so by adding the configuration shown below:

```

add ip routemap=ripmap entry=1 match tag=1280 action=exclude
add bgp import=rip routemap=ripmap

```

The following figure shows the BGP route table by using the **show bgp route** command. The route map filter has meant that the 192.168.50.0/24 network has not been imported into BGP, while the 192.168.53.0/24 route (also learnt by RIP) has been imported into BGP.

```
BGP route table
-----
  Prefix                Next hop          Origin    MED      Local pref
  Originator           Path
  Cluster List
-----
> 192.168.53.0/24      192.168.52.2     IGP       -         100
-
-
-----
Flags: >=Best route for the given prefix, *=Unreachable next hop
       A=Aggregate route, S=Aggregate Suppressed, D=Damped, W=Withdrawn
```

Route compatibility when RIP is set to receive both RIPv1 and RIPv2 routes

The router or switch has three possible modes for receiving RIP updates:

- **rip1**—only accepts RIPv1 updates
- **rip2**—only accepts RIPv2 updates
- **both**—accepts both RIPv1 and RIPv2 updates.

The default mode is **both**. Therefore, if you add a RIP interface, by using the command:

```
add ip rip interface=int send=rip2
```

without specifying the **receive** parameter, then your router goes into the receive mode of **both**. That means that RIP accepts both RIPv1 and RIPv2 messages.

However, when the mode is **both** and a RIP neighbour sends a RIPv2 message with a route that does not conform to classful addressing, then the recipient router or switch ignores that route. The router or switch processes routes in an update if they conform to classful addressing and ignores just the routes that do not.

This Tip first demonstrates the effect of **receive=both**. Then it shows how setting the mode to **receive=rip2** allows the router or switch to accept a route that it would otherwise ignore.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All

Example of route with incompatible address

Initially, Switch_A is configured with:

```
enable ip
add ip int=vlan100 ip=172.16.0.1
add ip int=vlan200 ip=10.0.0.1
add ip rip int=vlan200
```

Switch_B is configured with:

```
enable ip
add ip int=vlan100 ip=172.17.0.1
add ip int=vlan200 ip=10.0.0.2
add ip rou=192.168.0.0 mask=255.255.0.0 int=vlan100
    next=192.168.0.254
add ip rip int=vlan200 send=rip2
```

The address 192.168.0.0 is actually a class C address (as are all other addresses whose first three bits are 110—the range 192.0.0.0 to 223.0.0.0). When adhering to the rules of classful addressing, a class C address must have a 24 bit mask. Hence, the route 192.168.0.0/16 does not conform to classful routing.

Switch_B sends two routes (172.17.0.0/16 and 192.168.0.0/16) to its neighbours, as shown in bold in the following output:

```

Manager Switch_B> show ip route

IP Routes
-----
Destination      Mask           NextHop          Interface        Age
Type             Policy         Protocol         Tag              Metrics         Preference
-----
10.0.0.0          255.0.0.0     0.0.0.0         vlan200          19049
                  direct        0               interface        1              0
172.16.0.0       255.255.0.0   10.0.0.1        vlan200          16164
                  remote       0               rip              16             100
172.17.0.0      255.255.0.0   0.0.0.0        vlan100         19049
                  direct      0               interface      1              0
192.168.0.0    255.255.0.0   192.168.0.254 vlan100         15551
                  direct      0               static         -              60
-----

Manager Switch_B> enable ip route debug=rip

Info (1005287): Route debugging has been enabled.

Manager Switch_B> out(v.2):172.17.0.0/255.255.0.0/10.0.0.2/1/0 vlan100
out(v.2):192.168.0.0/255.255.0.0/10.0.0.2/1/0 vlan100
RIP >> Sending update to UDP
RIP Tx:Local->224.0.0.9, 520->520 ~ reg. update

```

Switch_A gets the packet. However, note in the debugging below that it only registers the receipt of 172.16.0.0/16. This is because Switch_A is in receive mode **both**, so it accepted the RIPv2 packet, but ignored the route that broke the class subnetting rules.

```

Manager Switch_A> enable ip route debug=rip

Info (1005287): Route debugging has been enabled.

Manager Switch_A> RIP Rx:224.0.0.9<-10.0.0.2, 520<-520 (v.2) ~ response. OK
in:172.17.0.0/255.255.0.0(255.255.0.0)/10.0.0.2(10.0.0.2)/1
RipRxRoutes: route found

Manager > show ip route

IP Routes
-----
Destination      Mask           NextHop          Interface        Age
DLCI/Circ.       Type             Policy         Protocol         Tag              Metrics         Preference
-----
10.0.0.0          255.0.0.0     0.0.0.0         vlan200          231
-                direct        0               interface        1              0
172.16.0.0       255.255.0.0   0.0.0.0         vlan100          231
-                direct        0               interface        1              0
172.17.0.0      255.255.0.0   10.0.0.2        vlan200         227
-                remote      0               rip            -              100
-----

```

Solution

Setting Switch_A to accept RIPv2 (and not be compatible with RIPv1) causes the switch to accept the classless route, as shown in the following figure.

```
Manager > set ip rip int=vlan200 receive=rip2

Info (1005282): RIP neighbour successfully updated.

Manager > RIP Rx:224.0.0.9<-10.0.0.2, 520<-520 (v.2) ~ response. OK
in:172.17.0.0/255.255.0.0(255.255.0.0)/10.0.0.2(10.0.0.2)/1
RipRxRoutes: route found
in:192.168.0.0/255.255.0.0(255.255.0.0)/10.0.0.2(10.0.0.2)/1

Manager > sh ip route

IP Routes
-----
Destination      Mask      NextHop      Interface      Age
DLCI/Circ.      Type      Policy      Protocol      Tag      Metrics      Preference
-----
10.0.0.0          255.0.0.0      0.0.0.0      vlan200          998
-                direct      0            interface      -          1              0
172.16.0.0        255.255.0.0    0.0.0.0      vlan100          998
-                direct      0            interface      -          1              0
172.17.0.0        255.255.0.0    10.0.0.2     vlan200          994
-                remote     0            rip              -          2              100
192.168.0.0      255.255.0.0    10.0.0.2     vlan200          3
-                remote      0            rip              -          2              100
-----
```

How to select the right ISAKMP policy during incoming Phase I ISAKMP proposals

VPN users sometimes experience issues when using more complicated IPsec configurations. Typically these occur for configurations using multiple ISAKMP policies referring to same peer address or for policies configured for “any” peer address.

When you have multiple ISAKMP policies defined, sometimes it may seem that an incoming ISAKMP Phase I proposal is matched to an incorrect ISAKMP policy. This Tip explains why this happens, and how to control policy selection.

A companion Tip, ["Why the remote peer VPN router may set up multiple ISAKMP SAs when responding to my router" on page 43](#), describes how to troubleshoot configurations that result in many ISAKMP SAs on the peer.

Here is a configuration example where the wrong policy could be selected:

```
create isakmp policy=isakmp peer=any key=1 mode=main natt=true
  encalg=3desouter

set isakmp policy=isakmp localid=example1 remoteid=ARRouter
set isakmp policy=isakmp sendn=true sendd=true
set isakmp policy=isakmp heart=both

create isakmp policy=isakmp2 peer=any key=2 mode=main natt=true
  encalg=3desouter

set isakmp policy=isakmp2 localid=example2 remoteid=WindowsVPN
set isakmp policy=isakmp2 sendn=true sendd=true
set isakmp policy=isakmp2 heart=none

ena isakmp
```

Background comments

There may be situations where you need to have more than one ISAKMP policy that are either to the same peer or are both configured with **peer=any**. For example, you could want two different dynamic peers to use unique pre-shared key values.

For another example, you can define two ISAKMP policies with **peer=any** in order to meet different feature requirements for two different types of peers. Two such peers could be:

- an AR router peer that is dynamically assigned an IP address by its ISP, and
- a Windows VPN client peer which also receives a dynamically assigned IP address

In this situation, you might want to use ISAKMP heartbeats for the AR router peer so that you can have good recovery of SAs after a power cycle. However, you would need a separate policy not using heartbeats for the Windows VPN Client peers, because these do not support ISAKMP heartbeats. Instead, they use another mechanism for recovery.

Products this Tip applies to

All routers listed on page 1

Rapier, Rapier i, and AT-8800 Series switches

Software Versions

All that support these products

Understanding selection

In these situations, it is important to appreciate which ISAKMP policy will be selected during an incoming ISAKMP phase 1 proposal.

The router works out which policy to select by checking the following items, in order:

1. peer address
2. remote ID (in aggressive mode only)
3. SA proposal values (encryption algorithm, hashing algorithm, etc).

It selects the policy on the basis of the first item that gives a unique match. For example, in aggressive mode, if two policies have the same peer address (or both have **peer=any**), then the router next inspects their remote IDs.

The selection steps for ISAKMP main mode are the same as for aggressive mode, except that the remote ID field cannot be used as part of ISAKMP policy selection. This means that if you have multiple ISAKMP policies configured to the same peer address (or with **peer=any**) and the only difference between the policies is the remote ID, then ISAKMP main mode cannot select a policy on this difference alone. It selects the first matching ISAKMP policy by policy name order instead.

Example problem

Let's look again at the configuration example that suffered a selection problem:

```
create isakmp pol=isakmp peer=any key=1 mode=main natt=true
  encalg=3desouter

set isakmp pol=isakmp localid=example1 remoteid=ARRouter
set isakmp pol=isakmp sendn=true sendd=true
set isakmp pol=isakmp heart=both

create isakmp pol=isakmp2 peer=any key=2 mode=main natt=true
  encalg=3desouter

set isakmp pol=isakmp2 localid=example2 remoteid=WindowsVPN
set isakmp pol=isakmp2 sendn=true sendd=true
set isakmp pol=isakmp2 heart=none

ena isakmp
```

There are a number of points of difference between these two policies. However, the following fields cannot not be used as selection points, because this information is not received in the phase 1 exchange:

- These policies have different keys, but the pre-shared key is obviously not sent during exchanges, so this cannot be used as a selection point.
- These policies have different heartbeat modes, but the incoming proposal does not indicate preference for heartbeat.

- These policies have different ID fields, but when using ISAKMP main mode, the incoming phase 1 proposal does not define IDs.
- These policies at present use the same hash algorithm and encryption algorithm: **hashalg=sha** and **encalg=3desouter**. These details are proposed in the incoming phase 1 proposal and could be used for selection, but in this example they are the same.

The end result of this problem is that all incoming proposals are being matched against the first policy only. This policy uses heartbeat mode. This might be good for some peers, but those that don't support heartbeat mode (Windows VPN) will find themselves being disconnected (VPN dropped) after a short period.

Solution

The solution is to configure the policies so that they have a point of difference for phase 1 selection. There are two approaches for this.

Differentiation by encryption algorithm

One approach would be to make the encryption algorithms unique. For example, you could make the AR router peer propose **encalg=3des2key** and adjust the local configuration to suit. 3des2key is a unique encryption algorithm that Windows will not propose. Windows does support the more standard **3desouter**, so this becomes a convenient point of distinction, as shown in the following commands:

```
set isakmp pol=isakmp encalg=3des2key
set isakmp pol=isakmp2 encalg=3desouter
```

Differentiation by ISAKMP mode

The other approach would be to set ISAKMP aggressive mode. This way the ID fields will be listed as part of the incoming phase 1 proposal and we can select on a unique Remote ID field.

However, note that Windows VPN clients do not support ISAKMP aggressive mode, but only main mode. This does not have to matter—making one ISAKMP policy use aggressive mode and the other use main mode serves as a point of distinction in itself.

Because Windows clients use main mode, the ID fields are proposed during phase 2 exchange. Phase 2 exchange is too late to serve for policy selection, but because the ID fields are quoted in Phase 2 exchange, the remote ID in your selected policy must match the local ID of the Windows client (or your policy can use **remoteid=any**).

To use the mode to differentiate, use the following command for the policy for the AR router peer:

```
set isakmp pol=isakmp mode=aggressive
```

and the following command for the policy for the Windows peer, configured second:

```
set isakmp pol=isakmp2 mode=main
```

Why the remote peer VPN router may set up multiple ISAKMP SAs when responding to my router

VPN users sometimes experience issues when using more complicated IPsec configurations. This Tip describes how to troubleshoot configurations if they result in many ISAKMP SAs on the peer. A companion Tip, "[How to select the right ISAKMP policy during incoming Phase I ISAKMP proposals](#)" on page 40, explains ISAKMP policy selection in complex configurations.

Products this Tip applies to

All routers listed on page I
Rapier, Rapier i, and AT-8800 Series switches

Software Versions

All that support these products

If the peer router has a large number of ISAKMP SAs, it is probably because of one of the following causes:

- power failure, which stops the router from sending ISAKMP Delete messages
- misconfiguration or non-support of heartbeat messages

This Tip first describes these causes, then the solutions. Then it lists other less-common possible causes.

Cause 1: Power failure

If the power fails on your router, or it restarts, the router and its peer negotiate for new VPN SAs when they next need to exchange traffic. If your router initiates the negotiation and is using ISAKMP aggressive mode, the previous SAs will not be deleted on the peer. This is because:

- aggressive mode does not reset the peer's SAs by sending Initial Contact messages, and
- a power failure prevents your router from sending Delete messages to the peer

If a succession of power cycles occurs, the peer may end up with many ISAKMP SAs.

See "[Preventing multiple SAs on the peer](#)" on page 44 for solutions in this situation.

Cause 2: Mismatched heartbeat configuration

A large number of SAs may be created when one peer sends and expects to receive ISAKMP heartbeat messages while the other peer does not send ISAKMP heartbeat messages (either through misconfiguration or because it does not support ISAKMP heartbeats). In this situation, the router that expects to receive heartbeats deletes the associated SAs, but the peer does not.

Example configuration with misconfigured heartbeats

Let's look in more detail at two routers with misconfigured ISAKMP heartbeats. The configuration of Router 1 is:

```
create isakmp pol=isakmp peer=ipadd key=1 mode=aggressive natt=true
set isakmp pol=isakmp localid=ARRouter1 remoteid=ARRouter2
set isakmp pol=isakmp sendnotify=true senddeletes=false
set isakmp pol=isakmp heart=both
```

The configuration of Router 2 is:

```
create isakmp pol=isakmp peer=any key=1 mode=aggressive natt=true
set isakmp pol=isakmp localid=ARRouter2 remoteid=ARRouter1
set isakmp pol=isakmp sendnotify=true senddeletes=false
set isakmp pol=isakmp heart=none
```

In this configuration one peer believes ISAKMP heartbeats are to be sent and received, and the other peer does not. Neither peer is configured to send Delete messages, and note that aggressive mode is being used.

The ISAKMP heartbeat feature is a good feature to use to detect the health of your peer—as long as both peers support heartbeats. If only one peer supports heartbeats, the following sequence could happen:

1. Router 1 initiates a VPN towards Router 2. Router 1 has ISAKMP heartbeats enabled, so it sends and expects to receive these heartbeat messages. When heartbeats are not received, Router 1 disconnects its peer (Router 2).
2. Router 1 deletes both the associated IPsec and ISAKMP SAs, but does not send a Delete message to the peer because **senddeletes=false**. Therefore, Router 2 still has both the IPsec and ISAKMP SAs up, and these will remain until expiry.
3. So now if Router 1 still wants to send VPN payload traffic, it quickly re-starts ISAKMP negotiations. This configuration uses aggressive mode, therefore the Initial Contact message that main mode uses is not sent. Without this Initial Contact message, the new ISAKMP SA establishes but any existing SAs on the peer remain. This means that the peer now has redundant ISAKMP SAs for this VPN connection.
4. Because of the mismatch with the heartbeat configuration, this process continues. The newly established ISAKMP SA and IPsec SA soon get deleted on Router 1, but again they remain on Router 2 because no Delete message is sent to it. The cycle continues for as long as Router 1 continues to need to send VPN payload traffic, and very soon Router 2 has many ISAKMP SAs up.

Preventing multiple SAs on the peer

There are several items that can be corrected to give a robust solution. These items apply to both problem causes:

- Use main mode instead of aggressive mode, if possible. Using main mode means that an Initial Contact message is sent for new negotiations, which triggers the peer to remove old SAs. You can use main mode in conjunction with the send Delete messages feature, for increased robustness. Note that ISAKMP main mode depends on the **sendnotify** parameter being enabled, to ensure that the Initial Contact message is sent.
- Set **senddeletes=true**, which ensures that the router sends Delete messages for previous SAs.
- Ensure that both peers use the same heartbeats configuration—**heart=both** if both peers support heartbeats.

In some situations you need to use aggressive mode (for an example see "[How to select the right ISAKMP policy during incoming Phase 1 ISAKMP proposals](#)" on page 40). In this case, we recommend you set **senddeletes=true**. This is a sufficient solution in most situations. However, the router cannot send a Delete message if the power fails or the router restarts. In

that case, the ISAKMP SAs time out after 24 hours by default. If you have frequent power cycles, you can reduce the ISAKMP SA expiry time (**set isakmp policy=name expiryseconds=seconds**).

Example configuration that avoids excess SAs

The following configuration is a robust example that avoids excess SAs from both causes.

Router 1 (with changes in bold):

```
create isakmp pol=isakmp peer=ipadd key=1 mode=main natt=true
set isakmp pol=isakmp localid=ARRouter1 remoteid=ARRouter2
set isakmp pol=isakmp sendnotify=true senddeletes=true
set isakmp pol=isakmp heart=both
```

Router 2:

```
create isakmp pol=isakmp peer=any key=1 mode=main natt=true
set isakmp pol=isakmp localid=ARRouter2 remoteid=ARRouter1
set isakmp pol=isakmp sendnotify=true senddeletes=true
set isakmp pol=isakmp heart=both
```

Other reasons for accumulation of SAs

ISAKMP SAs might also accumulate in the following situations (although probably only if sending of Delete messages is also disabled):

- if the **expirybytes** parameter is set for the ISAKMP policy, and the **expirybytes** soft-expiry threshold is reached much more quickly than the **expiryseconds** limit. When the **expirybytes** limit is reached, the router negotiates a new ISAKMP SA, but the old SA remains until the **expiryseconds** limit is reached.
- if the **phase2xchglimit** parameter is configured and phase 2 re-negotiations occur frequently. However, this is not a commonly used configuration option.
- if the peer uses a different heartbeat mechanism. Other vendors' equipment may be configured to use proprietary heartbeats, or Dead Peer Detection (RFC 3706), which is not available on Allied Telesis Labs routers. This may cause the peer to detect the VPN connection as failing, in which case the peer would delete the ISAKMP SAs.
- if the lower-layer link is dynamic and the link is reset. For example, Allied Telesis Labs routers will remove the IPsec SAs if the link has a dynamic IP address that is reset. Similar behaviour might happen for Windows L2TP/VPN if the L2TP or PPP connection is reset.

Also, in the following situations, the peer may not receive ISAKMP Delete messages:

- if the Delete message was lost or dropped. Informational Delete messages are not reliably transported, so if the message is dropped enroute, then a retransmission will not be sent.
- if you disable the ISAKMP module (by using the command **disable isakmp**). In this case, the peer may be left with one ISAKMP SA remaining. This is because the Delete message cannot be sent if no suitable SA exists.

When you have multiple policies to the same peer, you also need to consider the following points:

- As described above, ISAKMP main mode uses Initial Contact messages when a connection is first established with a peer. The Initial Contact message means that any existing SAs on the peer are removed when the first SA for the peer is established. So, if only one policy existed for the peer and heartbeats failed, the redundant ISAKMP SAs would be removed on the peer by the Initial Contact message. With multiple policies, this might not happen because another SA might exist for the peer for another policy.
- The **respondbadspi** IPsec policy parameter may not work in all cases with multiple policies to the same peer. It only works if no other ISAKMP SAs exist for the peer. So it works correctly if the router restarts (which is what it is primarily designed for). However, it does not work if multiple SAs to the same peer exist, then one policy's ISAKMP/IPsec SAs are removed, and then IPsec messages with bad SPIs are received.

About the firewall's aggressive mode

Aggressive mode is a state that the firewall enters when the TCP traffic is heavy—in particular, when a resource called the SYN queue is becoming full. The SYN queue records TCP sessions that have not completed the 3-way handshake. Aggressive mode is a key tool in the firewall's detection of and protection against SYN flood attacks.

The firewall enters aggressive mode in response to the following events:

- when the SYN queue reaches its limit (256 sessions)
- when the maximum number of suspicious hosts (4) has been reached and another suspicious host is detected.

A suspicious host is one that has 32 or more sessions in the SYN queue (32 TCP sessions that have been initiated but have not yet completed the 3-way handshake). For a host to no longer be suspicious it must have 16 or fewer sessions in the SYN queue.

There is also a mode called semi-aggressive. The mode changes to semi-aggressive when there is one or more suspicious hosts, but the total SYN queue limit has not been reached. When there are no suspicious hosts remaining the mode changes back to normal.

Aggressive mode and semi-aggressive mode support the firewall SYN flood protection. The SYN flood protection limits the number of sessions that have not completed the TCP 3-way handshake. There can be no more than 64 such sessions from one host and no more than 256 in total. This stops SYN floods from passing through the firewall.

Sometimes, legitimate traffic can look like a SYN flood. For example, a web proxy might need to open lots of sessions. One situation in which this could occur is when Adobe Acrobat periodically sends web traffic to the Adobe website to check for updates. If the Adobe website were to become unavailable, then sessions through the web proxy to the Adobe website would not complete the 3-way handshake. If there were 64 hosts checking for updates at the same time, it would become impossible for new web sessions to be started from the proxy. The purpose of the semi-aggressive and aggressive mode is to age out such sessions quickly so that new sessions can be established.

How the firewall behaves in aggressive mode

In aggressive mode, the SYN queue entries are aged out aggressively. In aggressive mode, if a TCP session fails to become established, the session is deleted from the firewall SYN queue after 18 seconds. In normal mode, it is deleted after 2 minutes.

Also, when the TCP setup proxy is in use, the number of packet retransmissions generated by the firewall depends on the mode. In normal mode up to 6 packets are sent (1 initial and 5 retransmissions). In aggressive mode the maximum is 3. If the maximum number of retransmissions is reached for any phase of the process, the session will be deleted.

In semi-aggressive mode, the firewall applies aggressive-mode behaviour to sessions that belong to suspicious hosts.

Products this Tip applies to

All routers listed on page 1 that run the versions below

Rapier, Rapier i, AT-8800, and AT-9800 Series switches

Software Versions

2.7.3 or later

The maximum number of retransmissions may be reached before the session timeout expires. For example, if the proxy receives a SYN packet from a public device, the proxy responds with a SYN/ACK, and expects to receive an ACK in response to the SYN/ACK. The proxy retransmits the SYN/ACK according to TCP methods (in which the interval doubles each time) until it receives an ACK from the public device. Therefore, in normal mode, the proxy sends up to 6 packets as follows:

```
Pkt1 (wait 1 second) Pkt2 (wait 2) Pkt 3 (wait 4) Pkt 4 (wait 8)
Pkt 5 (wait 16) Pkt 6 (wait 32) delete session
```

Transmitting these packets and waiting for the reply only takes 63 seconds.

When the firewall enters and leaves aggressive or semi-aggressive mode, it displays notification messages similar to the following:

```
Manager >
Warning (2077257): 16 Jan 2006 02:55:14
  Policy example's SYN queue state changed to: Semi Aggressive.
Manager >
Warning (2077257): 16 Jan 2006 02:57:14
  Policy example's SYN queue state changed to: Normal.
```


How to combine firewall standard and enhanced NAT

It is possible to use standard NAT for some firewall sessions and enhanced NAT for others. This Tip gives an example of how to do so.

In this example:

- The firewall router has two public interfaces, eth0-0 and eth0-1. The interface eth0-0 has an IP address of 40.30.20.10 and the default route goes out this interface. The interface eth0-1 has an IP address of 200.200.200.10.
- The firewall router has one private interface, eth1-0, with an IP address of 10.10.0.1.
- A subnet (10.30.0.0) is connected to the 10.10.0.0 private subnet via a next hop of 10.10.0.2
- Traffic to and from 10.30.0.10 is to use a standard NAT translation and be translated to and from 200.200.200.10
- Other traffic to and from the 10.30.0.0 subnet is to use an enhanced NAT translation and be translated to and from 40.30.20.10

Products this Tip applies to

All routers listed on page 1
Rapier, Rapier i, AT-8800, and
AT-9800 Series switches

Software Versions

All that support these products

Incorrect config To achieve this, you cannot simply create a standard NAT translation between 10.30.0.10 and 200.200.200.10, and an enhanced NAT translation between eth0-1 and eth0-0, by using the commands:

```
add firewall poli=example nat=standard int=eth1-0 ip=10.30.0.10
    gblint=eth0-1 gblip=200.200.200.10

add firewall poli=example nat=enhanced int=eth1-0 gblin=eth0-0
```

In this configuration, standard NAT does not work for outbound connections to, for example, 40.30.20.1. Outbound connections use the enhanced NAT definition.

This is because of the way the firewall works out which NAT to use. First, the firewall works out what the source and destination interfaces are. The firewall determines the destination interface by doing a route lookup for the destination address, to work out which interface the packet will eventually be routed out. In this example, the destination IP is 40.30.20.1, which means the session is routed over eth0-0, using the default route.

Next, the firewall looks for a NAT definition that is configured with those interfaces as the **int** and **gblint** parameters. In this example, the enhanced NAT definition has **gblint=eth0-0**. Therefore, the firewall uses the enhanced NAT definition for this session.

Correct config To force the firewall to use the standard NAT, you need to make the standard NAT definition use **gblint=eth0-0**. Then you can create a firewall rule that matches the **ip/gblip** pair and sets the interface to the value you actually want it to be. To do this, use the following four commands:

```
add firewall poli=example nat=standard int=eth1-0 ip=10.30.0.10
    gblint=eth0-0 gblip=200.200.200.10

add firewall poli=example nat=enhanced int=eth1-0 gblin=eth0-0
```

```
add firewall poli=example rule=25 act=nat int=eth0-1 protocol=udp
port=1-65000 ip=10.30.0.10 gblip=200.200.200.10 gblport=1-65000
add firewall poli=example rule=26 act=nat int=eth0-1 protocol=tcp
port=1-65000 ip=10.30.0.10 gblip=200.200.200.10 gblport=1-65000
```

The standard NAT is now the first NAT definition that matches the source and destination interfaces.

Note that these NAT definitions and rules do not determine which interface the packet is routed out, only which public interface and address is written into the packet header.

The router's IP and firewall configuration is:

```
enable ip
add ip int=eth0-0 ip=40.30.20.10 mask=255.255.255.0
add ip int=eth0-1 ip=200.200.200.10 mask=255.255.255.255
add ip int=eth1-0 ip=10.10.0.1 mask=255.255.255.252

add ip rou=0.0.0.0 mask=0.0.0.0 int=eth0-0 next=40.30.20.1
add ip rou=10.30.0.0 mask=255.255.0.0 int=eth1-0 next=10.10.0.2

enable firewall
create firewall policy=example
add firewall policy=example int=eth1-0 type=private
add firewall policy=example int=eth0-0 type=public
add firewall policy=example int=eth0-1 type=public

add firewall poli=example nat=standard int=eth1-0 ip=10.30.0.10
gblint=eth0-0 gblip=200.200.200.10
add firewall poli=example nat=enhanced int=eth1-0 gblin=eth0-0

add firewall poli=example rule=25 act=nat int=eth0-1 protocol=udp port=1-
65000 ip=10.30.0.10 gblip=200.200.200.10 gblport=1-65000
add firewall poli=example rule=26 act=nat int=eth0-1 protocol=tcp port=1-
65000 ip=10.30.0.10 gblip=200.200.200.10 gblport=1-65000
```

Why the switch has many “interface is UP” and “interface is DOWN” log messages

When a computer starts up it takes the Network Interface Card (NIC) through several stages. If your computer is PXE compliant (Boot from Network) the NIC may change state even more times on startup. If your computer supports WOL (Wake On Lane) it may keep the network link up when the PC is off, often at Half Duplex. Below is an example of the log messages from a switch (output of the **show log** command) and what the PC is doing to cause the messages. The PC under test is connected to Port1 of the switch and all messages are for this port.

Products this Tip applies to

All switches listed on page 1

Software Versions

All that support these switches

Date/Time	S	Mod	Type	SType	Message	
26 05:01:58	6	SWIT	PINT	DOWN	Port1: interface is DOWN	← Turned PC off
26 05:02:02	6	SWIT	PINT	UP	Port1: interface is UP	← PC shut down and set NIC at 10Mh WOL standby PC is powered up
26 05:02:19	6	SWIT	PINT	DOWN	Port1: interface is DOWN	← PC took down NIC and went through POST
26 05:02:22	6	SWIT	PINT	UP	Port1: interface is UP	← PC brought up NIC and looked for PXE DHCP server
26 05:02:32	6	SWIT	PINT	DOWN	Port1: interface is DOWN	← PC failed PXE boot and proceeded to HDD boot
26 05:02:35	6	SWIT	PINT	UP	Port1: interface is UP	← Windows is loading and NIC driver is not loaded yet
26 05:02:37	6	SWIT	PINT	DOWN	Port1: interface is DOWN	← Windows takes NIC down and loads drivers
26 05:02:40	6	SWIT	PINT	UP	Port1: interface is UP	← Windows brings up and initialises NIC
26 05:03:04	6	SWIT	PINT	DOWN	Port1: interface is DOWN	← Windows binds TCP/IP and IP address to NIC
26 05:03:09	6	SWIT	PINT	UP	Port1: interface is UP	← Windows has booted with IP bound to NIC

If you would like to stop these messages from being written to the log use the following commands:

```
add log output=temporary type=pint subtype=down action=ignore
add log output=temporary type=pint subtype=up action=ignore
```

How to gather useful debugging information for a suspected memory leak

Situation: Suspected buffer loss

Command: show buffer scan

Allied Telesis routers and switches allocate buffers (portions of memory) to various routines and operations. Buffer usage levels fluctuate as each routine requests and gains access to additional buffers (for example, when routing updates are received and processed additional buffers may be temporarily required).

The routers and switches utilise a hierarchical system of buffer management. They go through three increasingly severe phases of degradation as buffers are depleted:

- the router or switch stops responding to command input
- the router or switch discards input frames
- the router or switch reboots.

The **show buffer scan** command outputs a list of the number of buffers in use at each starting address. When investigating suspected buffer loss, you should use the **show time** and **show buffer scan** command to assess how fast the buffer leak is. For example, if you suspect the 'crash' or lock up is every hour, use the **show time** and **show buffer scan** commands at 10 second intervals, if you suspect it is every 24 hours, use the commands every 4 hours. Entering the **show time** command before **show buffer scan** allows the Help desk to see the time periods between buffer scans.

Note: Note that it is normal for some routines to utilise large numbers of buffers and it should not automatically be assumed that these routines are at fault. Note also that there is often a block of exactly 500 buffers.

Based on the output of this command, Allied Telesis engineers are often able to determine the area of code responsible for the consumption of the buffers and this assists with root cause identification.

The example below explains how to detect if your system has a buffer leak.

I. Display the time and the results of a buffer scan twice

Figure 1 on page 53 and Figure on page 53 show the first **show time** and **show buffer scan** outputs respectively. Use the first outputs as a base reference.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All

The output of **show buffer scan** is also in the output of **show debug**, but it is important that the **show buffer scan** output is captured several times. This will enable Allied Telesis engineers to verify the area in which buffers are being depleted.

```
System time is 15:03:22 on Monday 14-Jul-2003
```

Figure 1: Example output from the **show time** command.

```
004402a0 451 0008a1c8 27 0008a238 1 0008a268 9565 00000000 2880
00000200 11 00000400 11 00000600 11 00000800 11 00000a00 11
00000c00 11 00000e00 11 00001000 11 00001200 11 00001400 11
00001600 11 00001800 11 00001a00 11 00001c00 11 00001e00 11
00001010 11 0644e5b8 1 06812db8 1 06bd75b8 1 06f9bdb8 1
073605b8 1 07724db8 1 07ae95b8 1 004dc6c0 37 004dc724 9
004dc80c 66 004dadcc 503 00072764 1 0007cc20 122 0007cce4 104
00072b30 1 000727ec 5 003b7c2c 19 00280bc4 1 00280edc 1
0003a488 1 000ca4dc 2 000d9924 1 000d9a80 2 000d8ee4 3
00284928 25 0037ed90 1 0037eda4 1 004a2d84 2 001a198c 500
00213398 1 0021354c 1 004dba50 10 004bfa44 1 00087dd8 7
00087b88 1 00087d24 1 001ffa00 1 001ffbbc 1 004d556c 5
00396fac 137 001389d8 3 00138b50 23 00138dfc 36 00138d34 97
00138ee0 4 0013889c 7 0013878c 1 00139028 3 000d9080 1
0012cb34 1 001339ac 1 001339c4 1 001339d0 1 001339dc 1
001339e8 1 00490e4c 1 00097a2c 1 004b91d0 1 000c0b24 1
000ba82c 1 000ba834 1 000c3228 1 000ba084 40 008be0b0 11
008be0b8 11 008be0c0 11 008be0c8 11 00413ec0 4 008c5254 11
008c527c 11 008c5300 11 008c5350 23 008c53a0 22 008c53f0 212
004332d0 2 00433320 66 00433370 16 004333c0 29 0043fb24 9
004444cc 23 0044507c 35 0040fe2c 4 008ae9c0 9 00440748 128
0044f40c 1 0044ebf4 1 00396c94 61 00396d74 121 00396e58 6
00497154 2 00122020 1 008c4ed0 30 008c501c 6 008c5078 5
008c4f28 1 004fd4a8 1 001bd184 1 0011d018 120 001c03b4 1
001c0294 1 001bcf0c 1 004e6b90 1 001da108 1 001d8660 1
001d8fa8 1 00894384 1 008b661c 1 001a260c 7 008946c4 10
0048da9c 1 0041254c 1 004a2c44 1 004a2d04 120 001c9abc 59
001ba120 2 004c1804 1 004b6bd8 19 004b7034 4 004970f0 2
003b3568 1 0048b28c 1 004970f8 2 004e1c6c 1 0008bf58 1
0008bfc8 1 0049b524 1

Total buffers in use - 66168

Scan of fast buffers in use

Total fast buffers in use - 0

Memory ( DRAM ) ..... 262144 kB
Free Memory ..... 44 %
Free fast buffers ..... 0
Total fast buffers ..... 0
Free buffers ..... 58987
Total buffers ..... 87040
Buffer level 3 ..... 125 (don't process input frames)
Buffer level 2 ..... 250 (don't do monitor or command output)
Buffer level 1 ..... 500 (don't buffer up log messages)
```

Figure 2: Example output from the **show buffer scan** command

Figure 3 on page 54 and Figure 4 on page 54 show the second **show time** and **show buffer scan** outputs respectively. The highlighted address **0007cc20** shows that the memory has increased by approximately 1000 buffers since the last scan. The Free Memory is decreasing, and the number of Free buffers is decreasing.

```
System time is 15:05:32 on Monday 14-Jul-2003
```

Figure 3: Example output from the **show time** command

```
004402a0 448 0008a1c8 27 0008a238 1 0008a268 9565 00000000 2880
00000200 11 00000400 11 00000600 11 00000800 11 00000a00 11
00000c00 11 00000e00 11 00001000 11 00001200 11 00001400 11
00001600 11 00001800 11 00001a00 11 00001c00 11 00001e00 11
00001010 11 0644e5b8 1 06812db8 1 06bd75b8 1 06f9bdb8 1
073605b8 1 07724db8 1 07ae95b8 1 004dc6c0 37 004dc724 9
004dc80c 66 004dadcc 503 00072764 1 0007cc20 1132 0007cce4 104
00072b30 1 000727ec 5 003b7c2c 19 00280bc4 1 00280edc 1
0003a488 1 000ca4dc 2 000d9924 1 000d9a80 2 000d8ee4 3
00284928 25 0037ed90 1 0037eda4 1 004a2d84 2 001a198c 500
00213398 1 0021354c 1 004dba50 10 004bfa44 1 00087dd8 7
00087b88 1 00087d24 1 001ffa00 1 001ffbd8 1 004d556c 5
00396fac 137 001389d8 3 00138b50 23 00138dfc 36 00138d34 97
00138ee0 4 0013889c 7 0013878c 1 00139028 3 000d9080 1
0012cb34 1 001339ac 1 001339c4 1 001339d0 1 001339dc 1
001339e8 1 00490e4c 1 00097a2c 1 004b91d0 1 000c0b24 1
000ba82c 1 000ba834 1 000c3228 1 000ba084 40 008be0b0 11
008be0b8 11 008be0c0 11 008be0c8 11 00413ec0 4 008c5254 11
008c527c 11 008c5300 11 008c5350 23 008c53a0 22 008c53f0 212
004332d0 2 00433320 66 00433370 16 004333c0 29 0043fb24 9
004444cc 23 0044507c 35 0040fe2c 4 008ae9c0 9 00440748 128
0044f40c 1 0044ebf4 1 00396c94 61 00396d74 121 00396e58 6
00497154 2 00122020 1 008c4ed0 30 008c501c 6 008c5078 5
008c4f28 1 004fd4a8 1 001bd184 1 0011d018 120 001c03b4 1
001c0294 1 001bcf0c 1 004e6b90 1 001da108 1 001d8660 1
001d8fa8 1 00894384 1 008b661c 1 001a260c 7 008946c4 10
0048da9c 1 0041254c 1 004a2c44 1 004a2d04 120 001c9abc 59
001ba120 2 004c1804 1 004b6bd8 19 004b7034 4 004970f0 2
003b3568 1 0048b28c 1 004970f8 2 004e1c6c 1 0049b524 1
0008bf58 1 0008bfc8 1

Total buffers in use - 67165

Scan of fast buffers in use

Total fast buffers in use - 0

Memory ( DRAM ) ..... 262144 kB
Free Memory ..... 40 %
Free fast buffers ..... 0
Total fast buffers ..... 0
Free buffers ..... 57991
Total buffers ..... 87040
Buffer level 3 ..... 125 (don't process input frames)
Buffer level 2 ..... 250 (don't do monitor or command output)
Buffer level 1 ..... 500 (don't buffer up log messages)
```

Figure 4: Example output from the **show buffer scan** command

2. Display the time and the results of a buffer scan a third time

After the third **show time** and **show buffer scan** commands, the memory address has increased by approximately 1500 buffers and the Free Memory buffers have dropped 4%. This is shown in [Figure 6 on page 55](#) and is a good indication that you have a buffer leak.

```
System time is 15:07:38 on Monday 14-Jul-2003
```

Figure 5: Example output from the **show time** command.

```
004402a0 448 0008a1c8 27 0008a238 1 0008a268 9565 00000000 2880
00000200 11 00000400 11 00000600 11 00000800 11 00000a00 11
00000c00 11 00000e00 11 00001000 11 00001200 11 00001400 11
00001600 11 00001800 11 00001a00 11 00001c00 11 00001e00 11
00001010 11 0644e5b8 1 06812db8 1 06bd75b8 1 06f9bdb8 1
073605b8 1 07724db8 1 07ae95b8 1 004dc6c0 37 004dc724 9
004dc80c 66 004dadcc 503 00072764 1 0007cc20 2622 0007cce4 104
00072b30 1 000727ec 5 003b7c2c 19 00280bc4 1 00280edc 1
0003a488 1 000ca4dc 2 000d9924 1 000d9a80 2 000d8ee4 3
00284928 25 0037ed90 1 0037eda4 1 004a2d84 2 001a198c 500
00213398 1 0021354c 1 004dba50 10 004bfa44 1 00087dd8 7
00087b88 1 00087d24 1 001ffa00 1 001ffbd8 1 004d556c 5
00396fac 137 001389d8 3 00138b50 23 00138dfc 36 00138d34 97
00138ee0 4 0013889c 7 0013878c 1 00139028 3 000d9080 1
0012cb34 1 001339ac 1 001339c4 1 001339d0 1 001339dc 1
001339e8 1 00490e4c 1 00097a2c 1 004b91d0 1 000c0b24 1
000ba82c 1 000ba834 1 000c3228 1 000ba084 40 008be0b0 11
008be0b8 11 008be0c0 11 008be0c8 11 00413ec0 4 008c5254 11
008c527c 11 008c5300 11 008c5350 23 008c53a0 22 008c53f0 212
004332d0 2 00433320 66 00433370 16 004333c0 29 0043fb24 9
004444cc 23 0044507c 35 0040fe2c 4 008ae9c0 9 00440748 128
0044f40c 1 0044ebf4 1 00396c94 61 00396d74 121 00396e58 6
00497154 2 00122020 1 008c4ed0 30 008c501c 6 008c5078 5
008c4f28 1 004fd4a8 1 001bd184 1 0011d018 120 001c03b4 1
001c0294 1 001bcf0c 1 004e6b90 1 001da108 1 001d8660 1
001d8fa8 1 00894384 1 008b661c 1 001a260c 7 008946c4 10
0048da9c 1 0041254c 1 004a2c44 1 004a2d04 120 001c9abc 59
001ba120 2 004c1804 1 004b6bd8 19 004b7034 4 004970f0 2
0048b28c 1 004970f8 2 0008bfc8 1 0008bf58 1 004e1c6c 1
0049b524 1 003b3568 1
```

```
Total buffers in use - 69465
```

```
Scan of fast buffers in use
```

```
Total fast buffers in use - 0
```

```
Memory ( DRAM ) ..... 262144 kB
Free Memory ..... 36 %
Free fast buffers ..... 0
Total fast buffers ..... 0
Free buffers ..... 56490
Total buffers ..... 87040
Buffer level 3 ..... 125 (don't process input frames)
Buffer level 2 ..... 250 (don't do monitor or command output)
Buffer level 1 ..... 500 (don't buffer up log messages)
```

Figure 6: Example output from the **show buffer scan** command.

Once an address has been identified from the repeated **show buffer scan** command as being an address at which memory is accumulating, you can move onto the next stage of the process. In this example we have identified the address **0007cc20**.

The address identified from the repeated **show buffer scan** outputs is an address in the operating system code at which there is an instruction which is causing memory to be allocated, but this memory is not being subsequently freed.

3. Scan the identified address

The next step is to enter the command:

```
show buffer scan=<identified address from the show buff scan output>
```

This will output a long list of addresses as shown in [Figure 7 on page 56](#)

0096530c	0896db0c	0897cb0c	089ecb0c	08a0630c	08a7630c
08b3830c	08b6a30c	08bb430c	08d22b0c	08ec030c	08f0930c
08f1930c	0908c30c	0909130c	090b230c	090c730c	0916d30c
0917530c	0921ab0c	092ab30c	092edb0c	092f930c	0930eb0c
093b5b0c	093fc30c	09451b0c	094aeb0c	095acb0c	095c030c
0960030c	0964630c	0968bb0c	096cab0c	0971230c	09760b0c
0979330c	09796b0c	0981d30c	0984cb0c	0988b30c	098a130c
0990630c	09976b0c	0998bb0c	099b8b0c	099ce30c	099cf30c
099d6b0c	099de30c	099fbb0c	09a0a30c	09abbb0c	09adab0c
09b5630c	09bbeb0c	09be230c	09cb4b0c	09cbd30c	09cbfb0c
09d6430c	09d8230c	09d8630c	09db7b0c	09dc030c	09dcb30c
09e1cb0c	09e40b0c	09ee0b0c	09f2db0c	09f8d30c	09fe230c
09fe930c	09fff30c	0a01430c	0a04030c	0a06a30c	0a06c30c
0a0ab30c	0a0ae30c	0a0bd30c	0a0c930c	0a0d4b0c	0a0ee30c
0a110b0c	0a14bb0c	0a1bb30c	0a1ccb0c	0a1dab0c	0a1e730c
0a1f830c	0a20a30c	0a22bb0c	0a23f30c	0a23fb0c	0a241b0c
0a249b0c	0a264b0c	0a26cb0c	0a27d30c	0a281b0c	0a28b30c
0a28c30c	0a2a4b0c	0a2bbb0c	0a2d230c	0a2d4b0c	0a2e630c
0a30330c	0a33ab0c	0a33bb0c	0a352b0c	0a35930c	0a35a30c
0a37bb0c	0a38830c	0a38bb0c	0a390b0c	0a39d30c	0a3b730c
0a3bbb0c	0a3e130c	0a3e9b0c	0a41830c	0a41930c	0a437b0c
0a457b0c	0a46a30c	0a470b0c	0a49230c	0a49630c	0a49730c
0a49830c	0a4a2b0c	0a4bbb0c	0a4bdb0c	0a4c6b0c	0a4ce30c
0a4d5b0c	0a4d6b0c	0a50230c	0a511b0c	0a514b0c	0a51eb0c
0a525b0c	0a53230c	0a53930c	0a542b0c	0a54830c	0a557b0c
0a55ab0c	0a57fb0c	0a580b0c	0a58630c	0a59130c	0a592b0c
0a5a0b0c	0a5b0b0c	0a5c1b0c	0a5c3b0c	0a5c9b0c	0a5d5b0c
0a5db30c	0a5e1b0c	0a5ecb0c	0a5f330c	0a610b0c	0a63330c
0a633b0c	0a63b30c	0a64530c	0a64a30c	0a65430c	0a65730c
0a66330c	0a66e30c	0a68430c	0a68c30c	0a69730c	0a6aab0c
0a6c630c	0a6c730c	0a6cb30c	0a6d6b0c	0a6d7b0c	0a6d930c
0a6da30c	0a6e0b0c	0a6e6b0c	0a6eb30c	0a6edb0c	0a6f1b0c
0a6f230c	0a6fe30c	0a70cb0c	0a70eb0c	0a71eb0c	0a72930c
0a73530c	0a738b0c	0a73cb0c	0a746b0c	0a74830c	0a749b0c
0a74b30c	0a75930c	0a760b0c	0a772b0c	0a77430c	0a77fb0c
0a7a530c	0a7b1b0c	0a7d730c	0a7e830c	0a7f030c	0a7f4b0c
0a7f630c	0a7f8b0c	0a82730c	0a82c30c	0a832b0c	0a834b0c
0a83530c	0a83730c	0a85030c	0a850b0c	0a85830c	0a85ab0c
0a86430c	0a888b0c	0a88ab0c	0a88db0c	0a88e30c	0a89430c
0a8abb0c	0a8b7b0c	0a8c5b0c	0a8e330c	0a8f4b0c	0a915b0c
0a91630c	0a917b0c	0a928b0c	0a92e30c	0a92f30c	0a94130c

Figure 7: Example output from the **show buffer scan=0007cc20** command

Example output from the **show buffer scan=0007cc20** command (Continued)

```
0a95130c 0a95530c 0a95730c 0a95f30c 0a97db0c 0a97e30c
0a97fb0c 0a994b0c 0a9a1b0c 0a9ac30c 0a9b8b0c 0a9c530c
0a9c8b0c 0a9cc30c 0aa0130c 0aa0a30c 0aa14b0c 0aa1830c
0aa25b0c 0aa2630c 0aa4330c 0aa43b0c 0aa4630c 0aa5930c
0aa69b0c 0aa75b0c 0aa7930c 0aa86b0c 0aa89b0c 0aa9bb0c
0aaa030c 0aaa1b0c 0aaaab0c 0aab5b0c 0aaba30c 0aac0b0c
0aad130c 0aad8b0c 0aaecb0c 0ab10b0c 0ab15b0c 0ab27b0c
0ab3230c 0ab3a30c 0ab3b30c 0ab3e30c 0ab5030c 0ab54b0c
0ab55b0c 0ab57b0c 0ab66b0c 0ab6a30c 0ab7e30c 0ab8bb0c
0ab8eb0c 0ab9230c 0aba3b0c 0abadb0c 0abc30c 0abce30c
0abd9b0c 0abe1b0c 0abea30c 0ac04b0c 0ac1330c 0ac2130c
0ac22b0c 0ac2fb0c 0ac55b0c 0ac58b0c 0ac5eb0c 0ac66b0c
0ac7db0c 0ac9c30c 0ac9e30c 0ac9eb0c 0acb6b0c 0acc330c
0ace1b0c 0acecb0c 0acfa30c 0acff30c 0ad1cb0c 0ad2ab0c
0ad4430c 0ad6fb0c 0ad7cb0c 0ad8f30c 0ada530c 0adb30c
0adbe30c 0adde30c 0adf3b0c 0ae2230c 0ae3030c 0ae33b0c
0ae4930c 0ae5930c 0ae59b0c 0ae64b0c 0ae66b0c 0ae69b0c
0ae8030c 0ae8830c 0ae91b0c 0af8130c 0b188b0c 0b227b0c
0b504b0c 0b5b430c 0bfb1b0c 0c13230c 0c1bcb0c 0c478b0c
0c79c30c 0c7e7b0c 0c947b0c 0c977b0c 0caba30c 0cae830c
0cc3630c 0ccb330c 0ccb7b0c 0cd09b0c 0cdae30c 0ce3830c
0d0b2b0c 0d105b0c 0d1a530c 0d1e1b0c 0d2f230c 0d34030c
0d403b0c 0d41b30c 0d42f30c 0d44c30c 0d527b0c 0d567b0c
0d60b30c 0d6f330c 0d7b3b0c 0d9b830c 0da7830c 0dad2b0c
0dbeeb0c 0dcfd30c 0dd00b0c 0df8eb0c 0e00230c 0e5f830c
0e67fb0c 0e68a30c 0e6a2b0c 0e70ab0c 0e898b0c 0e951b0c
0e989b0c 0ea50b0c 0eb2730c 0eb4fb0c 0ec0930c 0ec6030c
0eccfb0c 0ed5cb0c 0edfdb0c 0ee60b0c 0eeb2b0c 0f09330c
0f102b0c 0f29130c 0f29a30c 0f2c6b0c 0f32930c 0f40d30c
0f484b0c 0f49db0c 0f4c7b0c 0f557b0c 0f59e30c 0f5f330c
0f61bb0c 0f64030c 0f67130c 0f6b730c 0f745b0c 0f879b0c
0f91230c 0f91530c 0f9a930c 0fa4230c 0fa5530c 0faef30c
0fb12b0c 0fb8e30c 0fb9db0c 0fbfcb0c 0fc1f30c 0fc3830c
0fc5d30c 0fd8530c 0fdf4b0c 0fe2fb0c

Memory ( DRAM ) ..... 262144 kB
Free Memory ..... 44 %
Free fast buffers ..... 0
Total fast buffers ..... 0
Free buffers ..... 58989
Total buffers ..... 87040
Buffer level 3 ..... 125 (don't process input frames)
Buffer level 2 ..... 250 (don't do monitor or command output)
Buffer level 1 ..... 500 (don't buffer up log messages)
```

Figure 7 on page 56 shows the addresses of the memory locations that have been allocated, but not yet freed (because of the logic error that is failing to free them). We need to see the contents of these memory locations, to see what data is being stored there. So, choose 2 or 3 addresses at random from the list. In this example we have chosen the addresses **0096530c** and **0aab5b0c**.

For each of these addresses, enter the commands:

```
dump address=<address> size=1 length=100
dump
dump
```

Screen outputs showing details of these commands are shown below in [Figure 8 on page 58](#) to [Figure 13 on page 60](#). They show the area of memory which has the buffer leak.

```

0ae8030c 0ae8830c 0ae91b0c 0af8130c 0b188b0c 0b227b0c
0b504b0c 0b5b430c 0bfb1b0c 0c13230c 0c1bcb0c 0c478b0c
0c79c30c 0c7e7b0c 0c947b0c 0c977b0c 0caba30c 0cae830c
0cc3630c 0ccb330c 0ccb7b0c 0cd09b0c 0cdae30c 0ce3830c
0d0b2b0c 0d105b0c 0d1a530c 0d1e1b0c 0d2f230c 0d34030c
0d403b0c 0d41b30c 0d42f30c 0d44c30c 0d527b0c 0d567b0c
0d60b30c 0d6f330c 0d7b3b0c 0d9b830c 0da7830c 0dad2b0c
0dbeeb0c 0dcfd30c 0dd00b0c 0df8eb0c 0e00230c 0e5f830c
0e67fb0c 0e68a30c 0e6a2b0c 0e70ab0c 0e898b0c 0e951b0c
0e989b0c 0ea50b0c 0eb2730c 0eb4fb0c 0ec0930c 0ec6030c
0eccfb0c 0ed5cb0c 0edfdb0c 0ee60b0c 0eeb2b0c 0f09330c
0f102b0c 0f29130c 0f29a30c 0f2c6b0c 0f32930c 0f40d30c
0f484b0c 0f49db0c 0f4c7b0c 0f557b0c 0f59e30c 0f5f330c
0f61bb0c 0f64030c 0f67130c 0f6b730c 0f745b0c 0f879b0c
0f91230c 0f91530c 0f9a930c 0fa4230c 0fa5530c 0faef30c
0fb12b0c 0fb8e30c 0fb9db0c 0fbfcb0c 0fc1f30c 0fc3830c
0fc5d30c 0fd8530c 0fdf4b0c 0fe2fb0c

Memory ( DRAM ) ..... 262144 kB
Free Memory ..... 44 %
Free fast buffers ..... 0
Total fast buffers ..... 0
Free buffers ..... 58989
Total buffers ..... 87040
Buffer level 3 ..... 125 (don't process input frames)
Buffer level 2 ..... 250 (don't do monitor or command output)
Buffer level 1 ..... 500 (don't buffer up log messages)

```

Figure 8: Example output from the `dump a=0096530c size=1 length=100` command

```

0096530c ffffffff 0903553c 00000000 00965450 .....U<.....TP
0096531c 00500002 00000050 00000001 00000000 .P.....P.....
0096532c 00000001 00000008 00000001 00000000 .....
0096533c 00000505 00000000 ffff00d0 b7b90b00 .....
0096534c 00000001 0000000c 00000003 00000003 .....
0096535c 81000001 00000001 00000000 72742033 .....rt 3
0096536c 2e393820 088b530c 088b5340 61727465 .98 ..S...S@arte
0096537c 64206175 746f6e65 676f7469 6174696f d autonegotiatio
0096538c 6e0d0a00 20202020 30202020 2020506b n... 0 Pk
0096539c 74734469 73636172 64656442 79426373 tsDiscardedByBcs
009653ac 63202020 20202020 20202020 20202030 c 0
009653bc 0d0a2020 4950506b 74735265 63656976 .. IPPktsReceiv
009653cc 65644174 56727020 20202020 20202020 edAtVrp
009653dc 20203135 34352020 20202049 5058506b 1545 IPXPk
009653ec 74735265 63656976 65644174 56727020 tsReceivedAtVrp
009653fc 20202020 20202020 20202020 300d0a20 0..

```

Figure 9: Example output from the `dump` command

```

0096540c 20495050 6b747346 6f727761 72646564          IPPktsForwarded
0096541c 20202020 20202020 20202020 20202031          1
0096542c 31323820 20202020 49505850 6b747346          128      IPXPktsF
0096543c 6f727761 72646564 20202020 20202020          orwarded
0096544c 20202020 52785478 4d616769 ffffffff          RxTxMagi....
0096545c ffff00d0 b7b90b00 81000001 08004500          .....E.
0096546c 0050fd69 000080e0 080f0a20 10160a20          .P.i.....
0096547c 10000000 27dd68c8 21010000 118e1000          ....'h.!.....
0096548c 00000000 00008e11 00000a20 10160a20          .....
0096549c 10160a20 10160300 00000400 00002c12          ... ..,
009654ac 00000000 00002a12 00000000 00000000          .....*.....
009654bc 2032340d 0a0a5061 636b6574 20444d41          24...Packet DMA
009654cc 20636f75 6e746572 733a0d0a 0a205265          counters:... Re
009654dc 63656976 653a2020 20202020 2031322d          ceive:      12-
009654ec 4a554c2d 32303033 00202020 20202020          JUL-2003.
009654fc 20202020 20547261 6e736d30 313a3533          Transm01:53

```

Figure 10: .Example output from the **dump** command

```

0096550c 3a353500 636b6574 73202020 20202020          :55.ckets
0096551c 20202020 20202020 20202020 20202020
0096552c 20203937 20202020 20506163 6b657473          97      Packets
0096553c 20202020 20202020 20202020 20202020
0096554c 20202020 20202020 2020300d 0a202044          0.. D
0096555c 69736361 72647320 20202020 20202020          iscards
0096556c 20202020 20202020 20202020 20202020
0096557c 30202020 20204469 73636172 64732020          0      Discards
0096558c 20202020 20202020 20202020 20202020
0096559c 20202020 20202030 0d0a2020 546f6f46          0.. TooF
009655ac 65774275 66666572 73202020 20202020          ewBuffers
009655bc 20202020 20202020 20202020 20302020          0
009655cc 20202041 626f7274 73202020 20202020          Aborts
009655dc 20202020 20202020 20202020 20202020
009655ec 20202020 300d0a20 20446573 63726970          0.. Descrip
009655fc 746f7273 45786861 75737465 64732020          torsExhausteds

```

Figure 11: .Example output from the **dump a=0096530c size=1 length=100** command

```

0096560c 20202020 20202020 20203020 20202020          0
0096561c 44657363 72697074 6f724172 65614669          DescriptorAreaFi
0096562c 6c6c6564 73202020 20202020 20202020          lleds
0096563c 20300d0a 20205175 6575654c 656e6774          0.. QueueLengt
0096564c 68202020 20202020 20202020 20202020          h
0096565c 20202020 20202030 20202020 2020517565          0      Que
0096566c 75654c65 6e677468 20202020 20202020          ueLength
0096567c 20202020 20202020 20202020 2020300d          0.
0096568c 0a0a5043 49206275 7320636f 756e7465          ..PCI bus counte
0096569c 72733a0d 0a202050 61726974 79457272          rs:... ParityErr
009656ac 6f727320 20202020 20202020 20202020          ors
009656bc 20202020 20202020 30202020 20204572          0      Er
009656cc 726f7243 68616e6e 656c2020 20202020          rrorChannel
009656dc 20202020 20202020 20202020 20202030          0
009656ec 0d0a2020 46617461 6c457272 6f727320          .. FatalErrors
009656fc 20202020 20202020 20202020 20202020

```

Figure 12: Example output from the **dump a=0aab5b0c** command

```

0096530c  ffffffff 0903553c 00000000 00965450          .....U<.....TP
0096531c  00500002 00000050 00000001 00000000          .P.....P.....
0096532c  00000001 00000008 00000001 00000000          .....
0096533c  00000505 00000000 ffff00d0 b7b90b00          .....
0096534c  00000001 0000000c 00000003 00000003          .....
0096535c  81000001 00000001 00000000 72742033          .....rt 3
0096536c  2e393820 088b530c 088b5340 61727465          .98 ..S...S@arte
0096537c  64206175 746f6e65 676f7469 6174696f          d autonegotiatio
0096538c  6e0d0a00 20202020 30202020 2020506b          n... 0 Pk
0096539c  74734469 73636172 64656442 79426373          tsDiscardedByBcs
009653ac  63202020 20202020 20202020 20202030          c 0
009653bc  0d0a2020 4950506b 74735265 63656976          .. IPPktsReceiv
009653cc  65644174 56727020 20202020 20202020          edAtVrp
009653dc  20203135 34352020 20202049 5058506b          1545 IPXPk
009653ec  74735265 63656976 65644174 56727020          tsReceivedAtVrp
009653fc  20202020 20202020 20202020 300d0a20          0..

```

Figure 13: Example output from the `dump a=0aab5b0c` command.

Why unexpected SNMP link-traps are sent from an AR410 router

When speed and/or duplex is reconfigured on an active port of the AR410 and AR410S routers, link up/down traps are sent for *all* the ports.

This is because the switching chip in these routers has to be reset every time its configuration is changed. Setting the speed or mode of an active port will cause the switch chip to be reset. This causes the link state of all the switch ports, during the next link state poll, to be detected by the software as having changed, resulting in traps being sent for all ports, not just the ports which were previously active / connected.

Products this Tip applies to
AR410 and AR410S routers

Software Versions
All that support these products

How to deal with spoofed packets

Spoofed packets are packets that have arrived into the device, but the source IP address on the packets is one of the IP addresses of the device itself.

You might see messages like the following in the log of your routers or switches.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All

```
10 18:20:06 3 IPG IPFIL FRAG Spoofed packet Fail 172.22.128.199>192.168.3.4
Prot=1 Int=vlan128
```

If so, what does this indicate is happening, and what should you do about it?

This indicates either that there is another device on the network that has been given the same IP address as this switch, **or** there is a routing loop, so that packets sent out by the switch are somehow going around a loop, and coming back to the switch.

To stop the spoofed packets, trace back around the network path that the packets have followed, and either find the device with the same IP as the switch, or find the routing loop.

How to deal with invalid DA packets

Invalid DA packet log messages indicate that the device has received packets whose destination IP address is an impossible address.

You might see messages like the following in the log of your routers and/or switches:

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All

```
10 18:49:53 3 IPG IPFIL DUMP Received invalid DA 172.22.128.18>0.0.4.56
Prot=6 Int=vlan128
```

This would indicate that there is a device on the network that is sending out invalid IP packets (In this case it is not valid to have an IP address like 0.0.4.56).

To stop the 'invalid DA' packets, trace back along the network path that the packets have followed, and find the device that is sending out the malformed packets.

How to interrupt text flow that is continuously streaming to the CLI

A keyboard short key allows you to interrupt text output to the CLI once a command has been entered. The keyboard short key is **Ctrl-Q**.

When you enter Ctrl-Q, the output pauses and the paging prompt appears. This gives you the choice of displaying the next line of text output, or the next page, or returning to printing text continuously with no further paging prompts, or simply aborting the text output.

Note that this function only works on output of a fixed length. This means text output from **enable debug** commands cannot be interrupted with this function.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

2.6.4 and later

How to display a text message at login

You can create a login banner message that will be displayed to users when they login to the switch or router. This is done by displaying the contents of the file “login.txt” if it exists in the Flash memory. The login banner message will be the last information displayed before the login prompt.

The login.txt file can be created or edited using the command:

```
edit login.txt
```

Or you can load an existing text file into Flash and rename it to login.txt. For more information on how to create a login.txt file or load a pre-existing text file refer to the **edit** and **load** commands in your router or switch Software Reference. Only users with Manager or Security Officer privilege level may create or load login.txt files.

See below for an example text message at login:

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

2.6.4 and later

```
Manager > restart router
INFO: Initialising Flash File System.

INFO: IGMP packet trapping is active for IGMP snooping, L3FILT is activated
INFO: MLD Snooping is active, L3FILT is activated
INFO: Switch startup complete

Warning - This is a live network device! Do not make changes to the
configuration unless you are authorised to do so.

login:
```

Figure 14: Example login message

How to set an inactivity timeout on console and TTY connections

It is possible to set an idle timeout on both telnet and TTY connections, and also on console connections over the ASYN ports. This is accomplished with an **idle** parameter in the **set tty** and **set asyn** commands. The idle timeout value is in seconds, and the valid options are a range between 10 and 4294967294, 0 or **off**. If the parameter is set to **off** or **0**, then the connections will never time out. The default timeout value is **off**.

Products this Tip applies to

All routers and switches listed on page 1 that run the following versions

Software Versions

2.7.4 and later

The captures displayed below show the syntax for setting the idle timeout on a TTY and ASYN connection, and the relevant **show** command for each.

```
Manager > set tty=16 idle=300

Info (1036003): Operation successful.

Manager > show tty=16

TTY information
Instance ..... 16
Login Name ..... manager
Description ..... Asyn 0
Secure ..... yes
Connections to .....
Current connection ..... none
In flow state ..... on
Out flow state ..... on
Type ..... VT100
Service ..... none
Prompt ..... default
Echo ..... yes
Attention ..... break
Manager ..... yes
Edit mode ..... insert
History length ..... 20
Page size ..... 22
Idle timeout (seconds) .... 300
```

Figure 21: Example output from the **show tty** command

```
Manager > set asyn=0 idletimeout=50

Info (1043281): Configuration updated.

Manager > show asyn=0

ASYN 0 : 0000001896 seconds   Last change at: 0000001892 seconds
ASYN information
Name ..... Asyn 0
Status ..... enabled
Mode ..... Ten
Data rate ..... 9600
Parity ..... none
Data bits ..... 8
Stop bits ..... 1
Test mode ..... no
In flow state (mode) ..... on (Hardware)
Out flow state (mode) ..... on (Hardware)
Autobaud mode ..... disabled
Max tx queue length ..... 16
TX queue length ..... 0
Transmit frame ..... none
RX queue length ..... 0
Enable Mode ..... break
Enabled Status Time Left .. 0

Control signals
  DTR (out) ..... on on      1
  RTS (out) ..... on -       1
  CD (in)  ..... n/a ignore  0
  CTS (in) ..... on -       0
  RNG (in) ..... n/a -      -

TTY information
Instance ..... 16
Login Name ..... manager
Description ..... Asyn 0
Secure ..... yes
Connections to .....
Current connection ..... none
In flow state ..... on
Out flow state ..... on
Type ..... VT100
Service ..... none
Prompt ..... default
Echo ..... yes
Attention ..... break
Manager ..... yes
Edit mode ..... insert
History length ..... 20
Page size ..... 22
Idle timeout (seconds) .... 50
```

Figure 22: Example output from the **show asyn** command

How to set Summer Time and time zones

You can configure the switch or router to automatically adjust for daylight saving. You will still need to set the local time using the **set time** command. If you set the time before configuring summer time, set the time to standard time. Once you enable summer time the device will automatically adjust the time in line with your summer time settings.

Products this Tip applies to

All routers and switches listed on page I that run the versions below

Software Versions

2.7.4 and later

To configure summer time use the command:

```
enable summertime
```

The default summer time parameters are set for North America, so if you are outside of the default area you will need to set the summer time parameters so they are correct for your region.

You can then set the correct start and end dates and times with the command:

```
set summertime [option]
```

Valid options are:

```
STARTDate, STARTMonth, STARTWeek, STARTDay, STARTTime, ENDDate  
ENDMonth, ENDWeek, ENDDay, ENDTime, OFFset
```

Once you have enabled summer time you can check what it has been set to with the **show summertime** command. You will notice that once summer time has been enabled there is a default summer time name of “DST” to indicate “Daylight Savings Time”. If desired, you can change this name with the command:

```
set summertime=text-string
```

The text string must be between 3 and 7 characters. The capture below show an example of how you could configure summer time.

Note: The command **set summertime=text-string** ONLY changes the name attached to the summer time settings, and makes no change to the actual settings (startmonth, startweek, etc).

```
Manager > enable summertime

Manager > set summertime=NZTIME startmonth=oct startweek=1 startday=su
      endmonth=mar endweek=3 endday=sun

Info (1034003): Operation successful.

Manager > show summertime

Summertime configuration
-----
Enabled ..... Yes
Summertime name ... NZTIME
Start ..... Sunday 01-Oct-2006 02:00am
End ..... Sunday 19-Mar-2006 02:00am
Offset ..... 60 minutes
Start rule ..... Recurring, First Sunday in October at 02:00am
End rule ..... Recurring, Third Sunday in March at 02:00am
-----
```

Figure 23: Example output from the **show summertime** command

It is also possible to configure the router or switch with an internationally recognised time zone. This is done with the command:

```
set timezone=text-string utc=offset-from-UTC
```

UTC is the current term for what is commonly known as Greenwich Mean Time (GMT).

Note: The text string the user defines with the **set timezone=text-string** command is just a label, and has no affect on the timezone settings. The timezone settings are controlled solely with the **utc=offset-from-UTC** parameter.

Below is an example capture of how you could configure a timezone.

```
Manager > set timezone=NZZONE utc=+12:00:00

Info (1034003): Operation successful.

Info (1034057): Timezone has been enabled.

Manager > show timezone

Timezone is set to 'NZZONE', offset from UTC is +12:00
```

How to ensure that system traffic is given priority when your switch is very busy

This Tip describes how to improve performance of AT-8948, AT-9900 and x900-48 Series switches in a congested network.

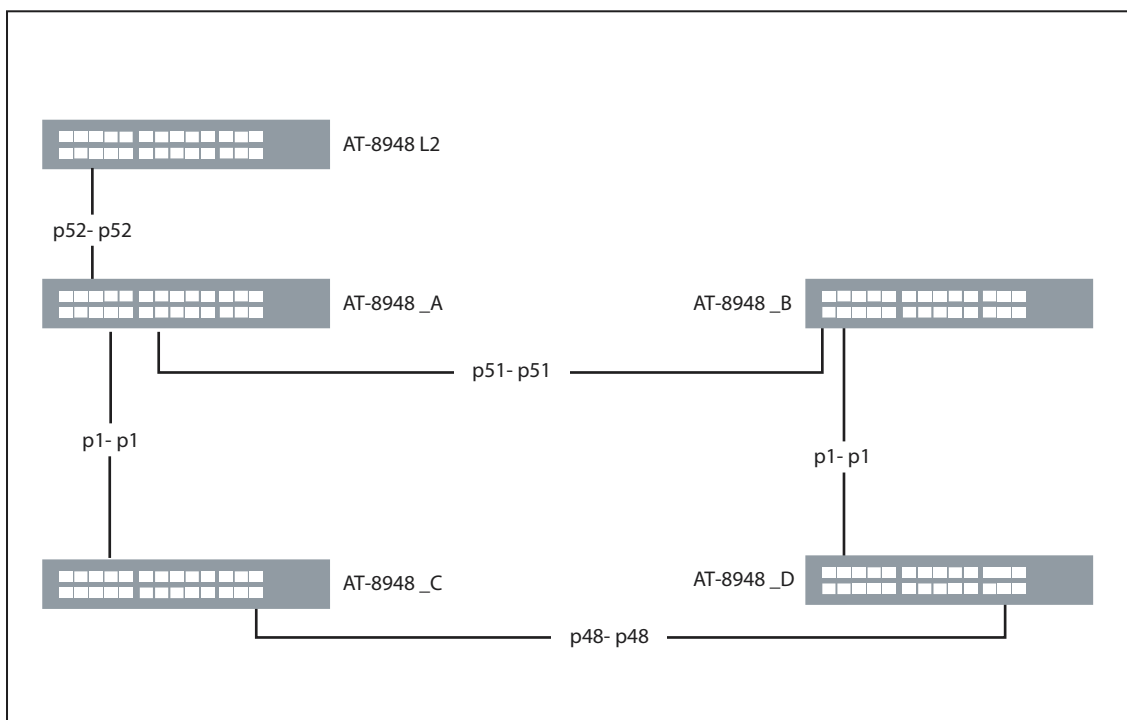
The specific configuration of this example only an illustration—it is not important. The focus is on the generic commands that we can add to improve performance in a congested network.

Products this Tip applies to

AT-8948, AT-9900 and x900-48 Series switches

Software Versions

2.7.3 or later



Description

Traffic is arriving into 8948_A on port 52 at 1000Mbps (full rate). The traffic is IP and the destination is not able to be resolved with ARP. Furthermore, no switch in the network has learned the MAC address of the destination in the Ethernet header so the switches are forced to flood the traffic to all ports in the VLAN. So, all the interfaces in one VLAN are 100% utilised and the CPU is extremely busy.

The switches run a routing protocol (for example OSPF). Routing updates are being affected, the route table does change, routing updates are arriving but are being processed intermittently.

On 8948_A we can see that the CPU is busy and the default queue (2) is overloaded:

```
Manager 8948_A> show cpu

CPU Utilisation ( as a percentage )
-----
Maximum since router restarted ..... 100
Maximum over last 5 minutes ..... 100
Average since router restarted ..... 40
Average over last 5 minutes ..... 33
Average over last minute ..... 41
Average over last 10 seconds ..... 64
Average over last second ..... 100
-----

Manager 8948_A> show qos port=1 count egress

Port 1 Egress Queue Counters:
Port queue length ..... 128 (maximum 128)
Egress queue length:
Queue 0 ..... 0 (maximum 128)
Queue 1 ..... 0 (maximum 128)
Queue 2 ..... 128 (maximum 128)
Queue 3 ..... 0 (maximum 128)
Queue 4 ..... 0 (maximum 128)
Queue 5 ..... 0 (maximum 128)
Queue 6 ..... 0 (maximum 128)
Queue 7 ..... 0 (maximum 128)
```

```
Manager 8948_B> show qos port=1 count egress

Port 1 Egress Queue Counters:
Port queue length ..... 128 (maximum 128)
Egress queue length:
Queue 0 ..... 0 (maximum 128)
Queue 1 ..... 0 (maximum 128)
Queue 2 ..... 128 (maximum 128)
Queue 3 ..... 0 (maximum 128)
Queue 4 ..... 0 (maximum 128)
Queue 5 ..... 0 (maximum 128)
Queue 6 ..... 0 (maximum 128)
Queue 7 ..... 0 (maximum 128)
```

There are two problems here:

1. The CPUs on 8948_A and 8948_B are busy servicing address lookups for a device they will never find. This means that important messages, like routing updates, never get processed by the CPU.
2. The ports 8948_A and 8948_B are overloaded with traffic.

We can fix this. However, we need to change the configuration of every device that we want to protect in the network. So the following has to be applied to 8948_A, 8948_B, 8948_C and 8948_D.

First, give priority to routing protocol traffic sent to the CPU, by using the following commands:

```
create classifier=1 IPProtocol=89 # OSPF
create class=2 UDPDport=520      # RIP
create class=3 TCPDport=179      # BGP4
add swi hwfilt class=1 action=setl2qos L2QOqueue=7
add swi hwfilt class=2 action=setl2qos L2QOqueue=7
add swi hwfilt class=3 action=setl2qos L2QOqueue=7
```

Second, prioritise the CPU generated traffic so that it is sent to queue 7 and hence given a higher priority, by using the following commands:

```
set switch cputxpriority=7
set switch cputxqueue=7
```

Soon, the IP route table is back to normal and all the OSPF routes have reappeared.

Summary

- The effect of broadcast storms on routing protocols can be controlled on the AT-8948, AT-9900 and x900-48 Series switches
- OSPF is very sensitive to lost packets and will be the first routing protocol to delete routing entries if links are very congested.

How to enable and install a release on the SwitchBlade with two controllers

This Tip describes how to enable and install a release on a SwitchBlade that has both a master and a slave controller installed.

Products this Tip applies to

SwitchBlade switches

Software Versions

All that support SwitchBlade

If the switch is running version 2.7.5A or later

First, load the release file onto the switch. After the release file is loaded onto the switch, the following 3 steps are required to enable and install it.

1. Enable a release license

```
enable release=release-name password=password number=release-number
```

If two switch controllers are present, this command only enables the release on the master controller. If the slave becomes the master, the release will become unavailable.

2. Enable the future use of a software release license on a slave controller

```
enable system sysr slvrelease=releasename slvnumber=releasename  
slvpassword=releasepassword
```

Note that this command is issued on the master controller card.

You can display the current status of release licenses in the switch on the master switch controller by using the command:

```
show release
```

You can display the current status of release licenses in the switch on the slave switch controller by using the command:

```
show system sysr slave
```

3. Set the preferred release on the switch

```
set install=preferred release=release-name
```

This will set the release file to be the preferred release on both the master and slave controllers.

If the switch is running a version prior to 2.7.5A

The above procedure was not always reliable for upgrading from earlier releases. Instead, we recommend you install the release onto the master and slave cards separately, by following the steps below. During this upgrade, the switch reboots twice, so you should do it during a scheduled network outage. The procedure takes approximately 15 minutes.

1. Remove the slave controller
2. Load the 275A-0x release onto the master controller
3. Enable the release licence and set the preferred install on the master, by using the commands:

```
enable release=release-name password=password number=release-number  
set install=preferred release=release-name
```
4. Remove the master controller and insert the slave
5. Load the 275A-0x release onto the slave controller
6. Enable the release licence and set the preferred install on the slave, by using the commands:

```
enable release=release-name password=password number=release-number  
set install=preferred release=release-name
```
7. Remove the slave controller
8. Re-insert the master controller and then the slave controller
9. Confirm that the master is running the correct release by entering the command **show install**
10. Confirm that the slave is running the correct release by entering the command **show sys sysr**. Towards the end of the output, check the entry "Comms Method". If this entry says "Master", then both controllers are running a 275-0x release. If it says "Master-Slave", then the slave is still running an earlier release or the bootrom release. In this case, check and repeat steps 4 onwards of the procedure.

How to fix switch port speed but still negotiate duplex

It is possible to fix the speed of switch ports to either 10 or 100 Mbps, but still allow them to autonegotiate the duplex mode. This was originally implemented for a Service Provider who wanted to ensure that their customers could not negotiate to speeds above 10 Mbps, but needed to allow them to negotiate to either full or half duplex in order to support a range of NIC cards that the customers were using.

Products this Tip applies to

All switches listed on page 1 that run the versions below

Software Versions

2.7.5 or later

Note that this feature only covers 10 Mbps and 100 Mbps speeds; it is not possible to fix a switch port to 1000 Mbps while still autonegotiating duplex mode.

The following tables show the complete list of speed and duplex options covering all switch platforms. The options that allow fixed speeds and autonegotiation of duplex mode are shown in bold.

autonegotiate	Autonegotiate both speed and duplex
10mauto	Fix speed to 10Mbps, autonegotiate duplex
10mhauto	Autonegotiate to ONLY 10Mbps, half duplex
10mhalf	Fix speed to 10Mbps, fix to half duplex
10mfauto	Autonegotiate to ONLY 10Mbps, full duplex
10mfull	Fix speed to 10Mbps, fix to full duplex

autonegotiate	Autonegotiate both speed and duplex
100mauto	Fix speed to 100Mbps, autonegotiate duplex
100mhauto	Autonegotiate to ONLY 100Mbps, half duplex
100mhalf	Fix speed to 100Mbps, fix to half duplex
100mfauto	Autonegotiate to ONLY 100Mbps, full duplex
100mfull	Fix speed to 100Mbps, fix to full duplex

autonegotiate	Autonegotiate both speed and duplex
1000mfauto	Autonegotiate to ONLY 1000Mbps, full duplex
1000mhalf	Fix speed to 1000Mbps, fix to half duplex
1000mfull	Fix speed to 1000Mbps, fix to full duplex

autonegotiate	Autonegotiate both speed and duplex
10G	Fix speed to 10Gbps, fix to full duplex

The options that are actually available to you from the list above depend on the model of switch or router you are using, and the type of port you are configuring. For details of the settings for each model and port type, see the Speed and Duplex Mode section of the Switching chapter in your switch's Software Reference.

How to make private and public VLANs share the same uplink

On AT-8600, AT-8700XL, Rapier, Rapier i and AT-8800 Series switches, there is a restriction that if a port has been added as an uplink port in a private VLAN, it cannot also be a member of a non-private VLAN. However, if you have an edge switch with only one uplink back to the network core, and you want to have both private and public VLANs on that switch, then they need to share that one uplink.

Products this Tip applies to

Rapier, Rapier i, AT-8600, AT-8700XL and AT-8800 Series switches

Software Versions

2.5.1 or later

Fortunately, however, a private VLAN can actually be configured in such a way that it acts like a non-private VLAN. The key to achieving this is the **group** parameter on the **add vlan port** command. If the **group** parameter is present on the command that adds a set of ports to a private VLAN, then those ports are allowed to exchange packets with each other, but may not exchange packets with other members of the private VLAN. (With the exception, of course, that they *can* exchange packets with the uplink port).

So, if *all* the edge ports are added to the VLAN together, and the **group** parameter is invoked, then *all* the edge ports can exchange packets with each other, so the VLAN effectively acts like a non-private VLAN.

The steps for configuring the private and non-private VLANs on the switch would be as follows:

1. Simply create the private VLANs in the standard way:

```
create vlan=vlanx vid=x private
add vlan=vlanx port=a,b,c,d,e
add vlan=vlanx port=u uplink
```

where *a, b, c, d, e* are port numbers, *u* is the uplink port number, and *x* is the VLAN ID

2. Create the “non-private” VLANs as private VLANs, but put all the ports into a group, so it actually acts as a non-private VLAN:

```
create vlan=vlany vid=y private
add vlan=vlany port=g,h,i,j group
add vlan=vlany port=u uplink
```

where *g, h, i, j* are port numbers, *u* is the uplink port number, and *y* is the VLAN ID

Note that if you need to add further ports to the “non-private” VLAN, then you must first delete *all* edge ports from the VLAN, then add the full new set of edge port members in a single command:

```
delete vlan=vlany port=g,h,i,j
add vlan=vlany port=g,h,i,j,k group
```

This is because if you simply added port *k* with the command:

```
add vlan=vlany port=k group
```

then port *k* would be added into a group of its own, which would be a separate group from the group containing ports *g, h, i, j*. This means that port *k* would not be able to exchange packets with ports *g, h, i, j*.

RSTP BPDU detection features

With RSTP it is never a good thing to have RSTP BPDUs coming into the same port that they were sent out of. If this is happening it indicates a serious problem with the network. The best way for the switch to handle this to disable the port and alert the user. For this reason Allied Telesis has implemented RSTP BPDU loopback protection, which puts the port into the “loopback disabled” state if a BPDU is received on the same RSTP enabled port from which it was sent. The “loopback disabled” state is essentially the same as the “backup” state, in that it will ignore all packets except BPDUs. It will also record that the port has been put into the “loopback disabled” state in the output of the **show stp port** command as shown below:

Products this Tip applies to

All switches listed on page 1 that run the versions below

Software Versions

2.7.3 and later

```
STP Port Information
-----
STP ..... default
  STP Status ..... ON

Port ..... 1
  RSTP Port Role ..... BackUp (Loopback Disabled)
  State ..... Discarding
  Point To Point ..... Yes (Auto)
  Port Priority ..... 128
  Port Identifier ..... 8001
  Pathcost ..... 200000 (auto configured)
  Designated Root ..... 32768 : 00-00-cd-03-00-44
  Designated Cost ..... 200000
  Designated Bridge ... 32768 : 00-00-cd-08-76-60
  Designated Port ..... 8001
  EdgePort ..... No
  VLAN membership ..... 1
  Counters:
    Loopback Disabled ..... 2
-----
```

Figure 24: STP port information output

The example above shows that the port has gone into the “loopback disabled” state, and it is the second time this has occurred since the device was rebooted.

As well as RSTP BPDU loopback protection as described above, if a port that has been configured as an “edge” port receives a BPDU it will change state to being an “active” RSTP port, meaning it will accept and transmit RSTP BPDUs.

How to allocate a WAN IP address to a PPP peer, and create a separate route to the subnet on the LAN side of the peer

When a device makes a PPP connection to the router or switch, it can request an IP address. There are several methods that the router or switch can use to decide which IP address to allocate to the peer. It can use an IP address configured on an entry in the user database, or an address from an IP pool, or it can ask a RADIUS server for an address, or obtain an address by reverse DNS lookup.

When the address is allocated to the peer, the router or switch will automatically create a route to that address, via the PPP interface on which the peer connected.

But, what if there is a LAN on the far side of the PPP peer, and it is necessary to **also** create a route to the subnet being used on that LAN?

The way to instruct the router or switch to create such a route is to authenticate the peer by RADIUS, and have one or more “framed route” attributes defined on the peer’s user entry on the RADIUS server.

For Freeradius, for example, this is achieved by creating an entry in the users file with a syntax like:

```
username1 Password = "123456",
User Service Type = Framed User,
Framed Protocol = PPP,
Framed Address = 1.2.3.4,
Framed Netmask = 255.255.255.255,
Framed Route = "131.100.123.0/28"
Framed MTU = 1500
```

This will cause the router or switch to create a route to 131.100.123.0/255.255.255.240, with a nexthop of 1.2.3.4, via the PPP interface on which the peer connected.

The syntax of the Framed route attribute is defined in RFC 2865. Note that you can have more than one framed route defined for a single user, in which case the router or switch will create a route for each framed route.

Products this Tip applies to

All routers listed on page 1

Rapier, Rapier i, AT-8800, AT-8948, x900-48, AT-9900, AT-9800, and SwitchBlade Series switches

Software Versions

All that support these products

How to reflect TOS onto L2TP tunnelled packets

It is possible to configure the router to reflect the TOS/DSCP field of the IP packet's header onto the IP header of the tunnelled L2TP packet. In essence reflecting the TOS/DSCP field onto the IP header of the L2TP packet means that the tunnelled packet will then contain the QoS information of the original IP packet, which can be used to prioritise the traffic based on the values set in the TOS/DSCP field, giving you the ability to use QoS across remote networks that are connected only over L2TP tunnels.

Products this Tip applies to

All routers listed on page 1 that run the versions below

Rapier i, AT-8800, AT-8948, AT-9900, x900-48, AT-9800, and SwitchBlade Series switches

Software Versions

2.7.5 or later

This feature can be enabled in three ways, for specific:

- L2TP calls
- IP addresses of L2TP peers
- L2TP users

So the commands **add l2tp call**, **add lt2p ip** and **add l2tp user** now all have a parameter called **tosreflection**.

The example below illustrates how you could configure a pair of routers for TOS reflection. Router1 is configured for TOS reflection for a specific L2TP call, and for Router2 the TOS reflection is configured on the L2TP peer IP address definition.

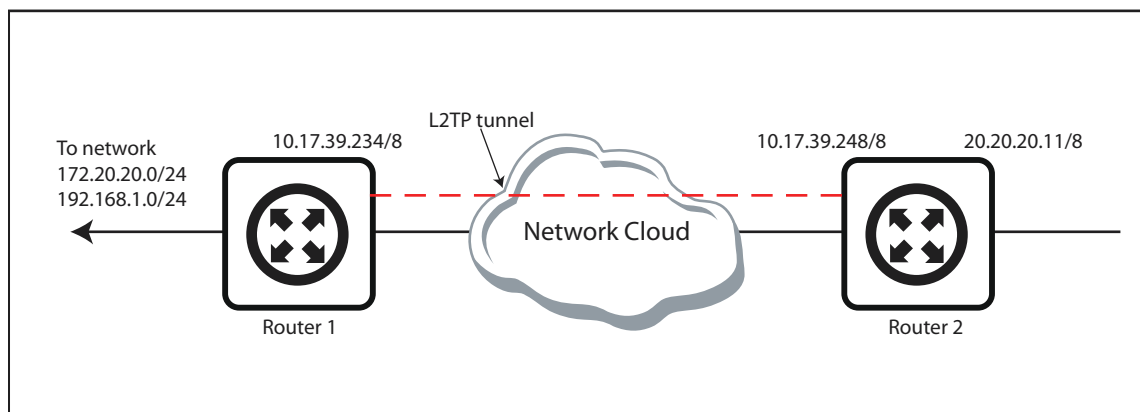


Figure 26: Routers configured for TOS reflection

Router I configuration

```
#L2TP configuration
enable l2tp
enable l2tp server=both

# Configure the TOS reflection on the L2TP call:
add l2tp call="test" ip=10.17.39.248 type=virtual prec=out tosreflection=on

#VLAN configuration
create vlan="vlan2" vid=2
create vlan="vlan3" vid=3
add vlan="2" port=1-5

#CLASSIFIER configuration
create class=1 prot="ip" ipsa=192.168.1.1/32
create class=2 prot="ip" ipsa=172.20.20.1/32

#PPP configuration
create ppp=0 idle=60 over=tnl-test
set ppp=0 bap=off iprequest=on username="user" password="password"
set ppp=0 over=tnl-test lqr=off echo=10

#IP configuration
enable ip
add ip int=vlan2 ip=10.17.39.234
add ip int=ppp0 ip=0.0.0.0
enable ip rou cou
add ip rou=0.0.0.0 mask=0.0.0.0 int=ppp0 next=0.0.0.0
add ip rou=172.20.20.0 mask=255.255.255.0 int=vlan2 next=10.17.39.41
add ip rou=192.168.1.0 mask=255.255.255.0 int=vlan2 next=10.17.39.40

#SQOS general configuration
ena sqos
cre sqos tr=1 premarkd=46
cre sqos tr=2 premarkd=32
cre sqos poli=1
add sqos poli=1 tr=1,2
add sqos tr=1 class=1
add sqos tr=2 class=2
set sqos int=ppp0 ou=1
```

Router 2 configuration

In this example we are using a PPP template on Router 2, therefore the dynamic “PPP0” interface does not exist unless the L2TP call is active. To ensure the router has a route back via the PPP we use a trigger to add the default route each time the IP control protocol (IPCP) of the PPP comes up. This route will then be deleted when the dynamic PPP goes down.

Router commands

```
#User Configuration
add user=user password=password login=no priv=user

#PPP templates configuration
create ppp template=1
set ppp template=1 bap=off login=user ippool="test" authentication=chap

#L2TP configuration
enable l2tp
enable l2tp server=both

# Configure the TOS reflection on the L2TP peer IP address definition:
add l2tp ip=10.17.39.234 ppptemplate=1 tosreflection=on

#IP configuration
enable ip
add ip int=eth0 ip=10.17.39.248
add ip int=eth1 ip=20.20.20.11
create ip pool="test" ip=10.17.39.150-10.17.39.160

#TRIGGER Configuration
enable trigger
create trigger=1 interface=ppp0 event=up cp=ipcp script=ppproute.scp
```

Script The script **ppproute.scp** contains the following command:

```
add ip rou=0.0.0.0 int=ppp0 mask=0.0.0.0 next=0.0.0.0
```

The screenshot below shows an ethereal capture of a packet from IP address 172.20.20.1 that has been encapsulated into L2TP by Router1. You can see that the DSCP field in the IP header has been set to 0x20 (32 in decimal), and that the same value has been set in the DSCP field of IP header of the L2TP encapsulated packet.

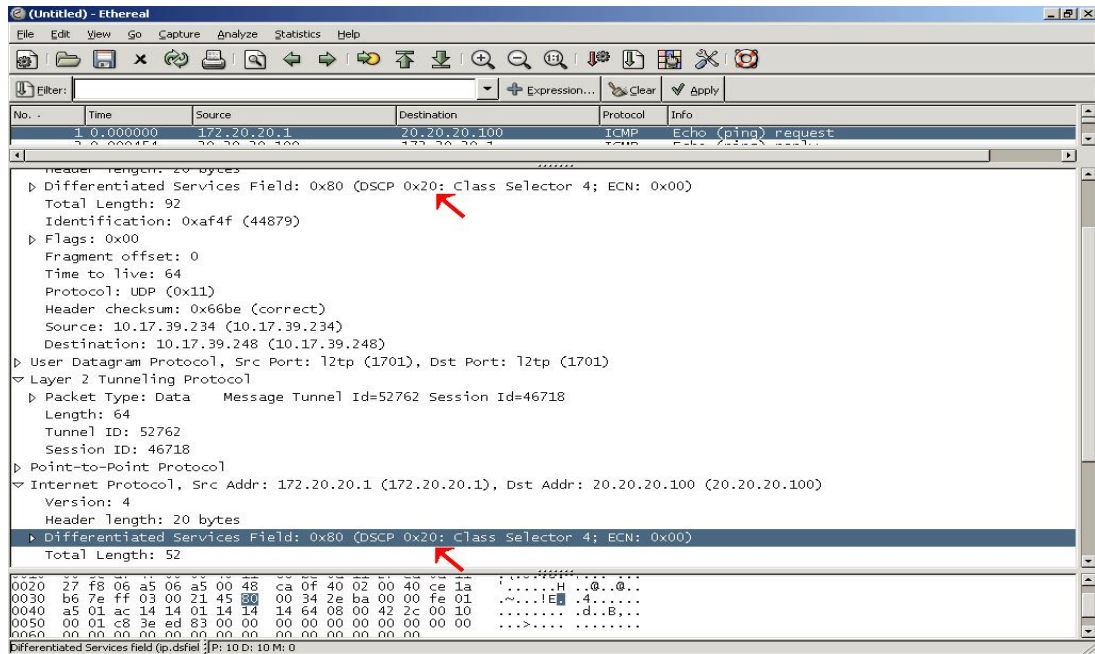


Figure 27: DSCP field in the IP header set to 0x20

The screenshot below shows another example ethereal capture of a packet from IP address 192.168.1.1 (which matches traffic class 2 in the software QoS configuration). This time you can see that the DSCP fields in both the IP packet header, and the IP header of the L2TP packet have been set to 0x2e (46 in decimal).

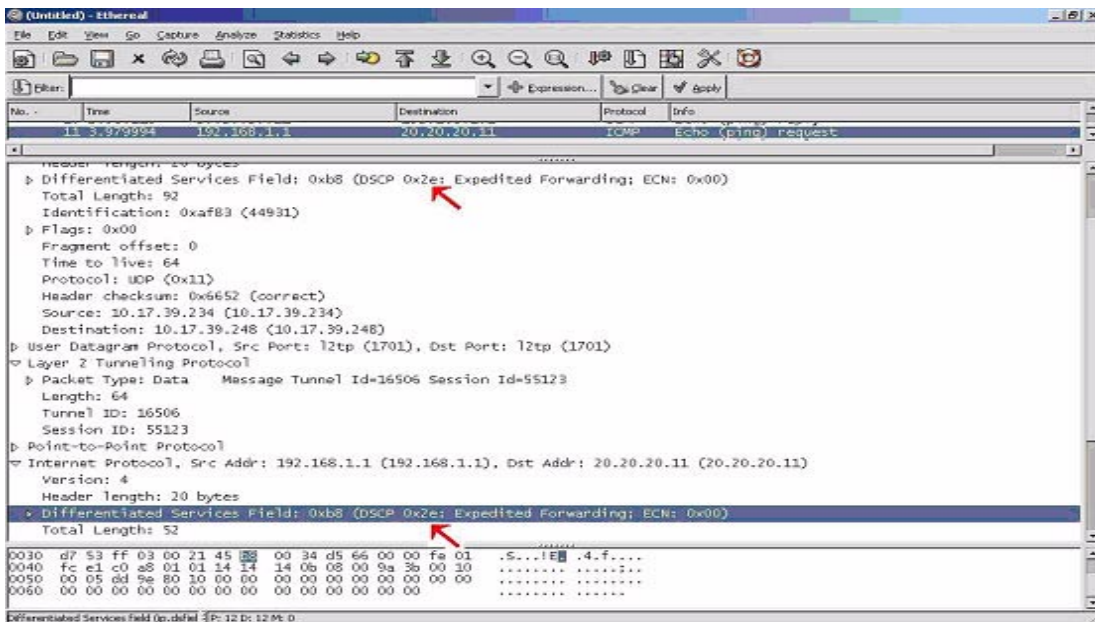


Figure 28: DSCP field in the IP header set to 0x2e

How to use Ping or Trace using Domain Name Service (DNS)

It is possible to ping a domain name or trace a route to a domain name. The router or switch will perform a DNS lookup to resolve the domain name. The commands are:

```
ping domain-name [other options]
trace domain-name [other options]
```

To add a DNS address for the router or switch to use in order to resolve domain names the command is:

```
add ip dns primary=ipaddress [secondary=ipaddress] [other options]
```

It is also possible to specify a domain name as the default destination. The switch or router will store the domain name as the default ping or trace destination, and will perform a DNS lookup to resolve the name to an address. This could be most useful in a test situation where you need to repeatedly ping or trace to the same address.

To do this use the commands:

```
set ping domain-name [other options]
set trace domain-name [other options]
```

You can see what the default ping and trace destination is set to using the **show ping** or **show trace** command. The output from these commands will also show you if the router or switch has resolved the domain name, and the IP address to which it has been resolved.

The following figure shows an example output from the **show ping** command.

```
Manager > show ping

Ping Information
-----
Defaults:
Type ..... IP
Source ..... 10.1.1.1
Destination ..... test-domain
Number of packets ..... 5
Size of packets (bytes) ..... 24
Timeout (seconds) ..... 1
Delay (seconds) ..... 1
Data pattern ..... Not set
Type of service ..... 0
Direct output to screen ..... Yes
-----

Info (1058264): No PING info for this device.

Manager >
```

Figure 29: Example output from **show ping**

Products this Tip applies to

All routers and switches listed on page I that run the versions below

Software Versions

2.7.3 and later

How OSPF metrics are calculated

If OSPF autocost has been enabled by using the command:

```
set ospf autocost=on
```

then every OSPF interface is assigned a cost in the range 1 to 65535. It represents the cost for traffic to EXIT the router from the interface. The default value of the cost is calculated from the following formula:

$$\text{OSPF Cost} = \frac{\text{reference bandwidth}}{\text{Interface Speed in bps}}$$

By default the reference bandwidth is 1000Mbps

Based on that formula, the following OSPF costs will be assigned to the following interface types, once OSPF is enabled on them.

Interface Type (Speed)	Default OSPF Cost
Gigabit Ethernet	1
10/100 Mbps Ethernet	10
T3	22
T1	667

Note: The reference bandwidth can be altered using the command: **set ospf refbandwidth**

Now, the cost to reach any given remote route will generally be an accumulation of the individual costs along the path.

In fact, OSPF calculates the metric differently for different types of route:

1. Intra area route (RFC 2328 chapter 16.1)

Each router within the area will advertise the networks that they are attached to in the router Link State Advertisements (LSAs). The cost to any given network is the smallest distance between the route source and the advertising router plus the cost advertised by that router.

2. Inter area route (RFC 2328 chapter 16.2)

Inter area routes are advertised by the area border router in the type 3 summary LSA, the cost is the distance to border router plus the cost specified in the type 3 summary LSA.

3. AS external route (RFC 2328 chapter 16.4)

AS external routes are advertised by autonomous system border routers (ASBR) in AS external LSAs. If the forwarding address specified in the LSA is 0.0.0.0, it means forward packets through the ASBR. If the forwarding address is non zero, that informs the router to look up the forwarding address in the routing table and forward the packet through a specified forwarding address.

Products this Tip applies to

All routers and switches listed on page 1 that run the versions below

Software Versions

2.7.1 and later

AS External route types

There are two types of AS external routes—Type-1 and Type-2—and the type is specified in the LSA.

Let X be the cost specified by the preferred routing table entry for the ASBR/forwarding address, and Y the cost specified in the LSA. If the external metric type is 1, then the path type is set to Type-1 external and the cost is equal to $X+Y$.

If the external metric type is 2, the path type is set to Type -2 external. The link state component of the route's cost is X , and the Type-2 cost is Y .

Comparing external route types

A Type-1 route is always better than a Type-2 route. If we only have Type-2 routes, then we only compare the cost Y (cost X is ignored when comparing Type 2 routes). So, the route with the lower external component wins.

Let us look at an OSPF Type-2 external route entry:

Destination DLCI/Circ.	Mask Type	Policy	NextHop Protocol	Tag	Interface Metrics	Age Preference
195.135.233.96	255.255.255.240		83.243.73.4		vlan4	240356
-	remote	0	ospf-EXT2	-	1(20)	151

You will see that the metric is presented in the form $Y(X)$. The X in this situation is the internal metric of OSPF (from Rapiere to the ASBR) and Y is the external metric (from ASBR to the final destination). When looking for the best route to a destination, the router will ignore the internal metric (which is X) and just take the external metric into consideration (Y).

If multiple routes to a particular destination all have the same external cost, then the internal cost can be used as a tie-breaker.

Filtering OSPF static routes with a whitelist or blacklist route map

OSPF can be configured to import static routes based on prefix. This can be done in either a blacklist or whitelist style. Whitelists and blacklists are not commands, they are concepts. In this Tip, the concept of a blacklist is where static routes are specifically blocked from being advertised by OSPF. The concept of a whitelist is where static routes are specifically permitted to be advertised by OSPF.

Products this Tip applies to

All routers and switches listed on page 1 that run the versions below.

Software Versions

2.7.5 or later

The **add ospf redistribute protocol=static** command has been specifically designed for handling static routes.

In this example, RIP and OSPF are running in an internal network of switches. Both routing protocols carry the same network information. On the AS Border Router (named ASBR), which here is actually a switch, we add some static networks via the next hop 192.168.0.1.

The first example is a blacklist, where we specify routes we **don't** want. The second example is a whitelist, where we specify routes we **do** want.

Blacklist

Initially, before any route maps are applied, a neighbour router has the following IP route table. **Show ip route** shows that it has learnt several AS EXT2 routes from ASBR (192.168.0.1):

```
IP Routes
```

Destination	Mask	Policy	NextHop	Interface	Age
	Type		Protocol	Metrics	Preference
<snip>					
200.255.250.0	255.255.255.0	remote 0	192.168.0.1 rip	vlan168 2	133 100
200.255.250.0	255.255.255.0	remote 0	192.168.0.1 ospf EXT2	vlan168 1(1)	119 151
200.255.251.0	255.255.255.0	remote 0	192.168.0.1 rip	vlan168 2	133 100
200.255.251.0	255.255.255.0	remote 0	192.168.0.1 ospf EXT2	vlan168 1(1)	119 151
200.255.252.0	255.255.255.0	remote 0	192.168.0.1 rip	vlan168 2	133 100
200.255.252.0	255.255.255.0	remote 0	192.168.0.1 ospf EXT2	vlan168 1(1)	119 151
200.255.253.0	255.255.255.0	remote 0	192.168.0.1 rip	vlan168 2	133 100
200.255.253.0	255.255.255.0	remote 0	192.168.0.1 ospf EXT2	vlan168 1(1)	119 151
200.255.254.0	255.255.255.0	remote 0	192.168.0.1 rip	vlan168 2	133 100
200.255.254.0	255.255.255.0	remote 0	192.168.0.1 ospf EXT2	vlan168 1(1)	119 151
200.255.255.0	255.255.255.0	remote 0	192.168.0.1 rip	vlan168 2	133 100
200.255.255.0	255.255.255.0	remote 0	192.168.0.1 ospf EXT2	vlan168 1(1)	119 151

Now we add the commands for the filtering. In the following configuration outputs, the commands that are most relevant to the filtering examples are shown in bold.

```
# IP configuration

enable ip

set ip autonomous=65000

add ip int=vlan168 ip=192.168.0.3

add ip prefixlist=asbr_only entry=1 action=match
prefix=200.255.252.0 masklength=24

add ip prefixlist=asbr_only entry=2 action=match
prefix=200.255.253.0 masklength=24

add ip rou=200.255.250.0 mask=255.255.255.0 int=vlan168
next=192.168.0.1

add ip rou=200.255.251.0 mask=255.255.255.0 int=vlan168
next=192.168.0.1

add ip rou=200.255.252.0 mask=255.255.255.0 int=vlan168
next=192.168.0.1

add ip rou=200.255.253.0 mask=255.255.255.0 int=vlan168
next=192.168.0.1

add ip rou=200.255.254.0 mask=255.255.255.0 int=vlan168
next=192.168.0.1

add ip rou=200.255.255.0 mask=255.255.255.0 int=vlan168
next=192.168.0.1

add ip routemap=asbr ent=1 act=exclude match prefixlist=asbr_only

add ip rip int=vlan168 send=rip2 receive=rip2

# OSPF configuration

set ospf routerid=192.168.0.3 asexternal=on

add ospf redistribute protocol=static routemap=asbr

add ospf area=0.0.0.1 authentication=password stubarea=off
summary=no

add ospf range=192.168.0.0 area=0.0.0.1 mask=255.255.255.0

add ospf interface=vlan168 area=0.0.0.1

enable ospf
```

Once this “blacklist” route map has been applied on the ASBR, **show ip route** shows that the route table on the neighbour router no longer contains some of the EXT2 routes it previously received from the ASBR. You can see that RIP is still passing the static routes, but OSPF has not advertised the networks 200.255.252.0 and 200.255.253.0.

```

IP Routes

Destination      Mask      NextHop      Interface      Age
                  Type      Policy      Protocol      Tag      Metrics      Preference

<snip>

200.255.250.0    255.255.255.0    192.168.0.1    vlan168      19
                  remote 0          rip           2          100
200.255.250.0    255.255.255.0    192.168.0.1    vlan168      9
                  remote 0          ospf EXT2     1(20)       151
200.255.251.0    255.255.255.0    192.168.0.1    vlan168      19
                  remote 0          rip           2          100
200.255.251.0    255.255.255.0    192.168.0.1    vlan168      9
                  remote 0          ospf EXT2     1(20)       151
200.255.252.0    255.255.255.0    192.168.0.1    vlan168      19
                  remote 0          rip           2          100
200.255.253.0    255.255.255.0    192.168.0.1    vlan168      19
                  remote 0          rip           2          100
200.255.254.0    255.255.255.0    192.168.0.1    vlan168      19
                  remote 0          rip           2          100
200.255.254.0    255.255.255.0    192.168.0.1    vlan168      9
                  remote 0          ospf EXT2     1(20)       151
200.255.255.0    255.255.255.0    192.168.0.1    vlan168      19
                  remote 0          rip           2          100
200.255.255.0    255.255.255.0    192.168.0.1    vlan168      9
                  remote 0          ospf EXT2     1(20)       151

```

Whitelist

Now change a little bit of the IP config—the action taken on matching and non-matching prefixes—on ASBR:

```
set ip routemap=asbr entry=1 match prefixl=asbr_only action=include
add ip routemap=asbr entry=2 action=exclude
```

... and after a minute or two use **show ip route** to observe the effect on the route table of the neighbour router:

IP Routes						
Destination	Mask	Policy	NextHop	Tag	Interface	Age
	Type		Protocol		Metrics	Preference
<snip>						
200.255.250.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		rip		2	100
200.255.251.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		rip		2	100
200.255.252.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		rip		2	100
200.255.252.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		ospf EXT2		1(20)	151
200.255.253.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		rip		2	100
200.255.253.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		ospf EXT2		1(20)	151
200.255.254.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		rip		2	100
200.255.255.0	255.255.255.0		192.168.0.1		vlan168	489
	remote 0		rip		2	100

OSPF on ASBR has advertised the two routes 200.255.252.0 and 200.255.253.0 and excluded the others. RIP, of course, is not affected.

How to identify and combat worm attacks

This Tip describes a method for dealing with worm attacks, using the Sasser Worm as an example. For other worms, apply the same filtering principles on the ports that the worm attacks.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All

What is the Sasser Worm?

The Sasser Worm is a piece of code that can install itself into Microsoft® Windows® system files. Once a PC has been infected, a remote attacker can get 'back door' access to the PC, and perform actions on the PC's files. Additionally, the worm will automatically transfer itself to other PCs.

How does the worm transfer itself?

It makes use of the fact that a vulnerability in the Windows Local Security Authority Service Server will allow it to connect to a PC on TCP port 445. When it has found a PC that responds on TCP 445, it can create a remote shell on the PC, on TCP port 9996.

It then starts up an FTP service on the original machine, listening on TCP port 5554. Via the remote shell session, it forces the new machine to make an FTP connection to port 5554, and retrieve a copy of the worm.

How do I combat the worm?

The telltale sign of the presence of the worm on a network is that there will be a lot of TCP SYN packets being sent to port 445 on a large number of different IP addresses, as it searches for vulnerable systems. So, the number one way to block the spread of the worm is to block TCP packets to port 445. It is important to also block traffic to ports 5554 and 9996.

To do this, use the following IP filters on **routers**, applied to the proper interfaces:

```
add ip filter=1 source=0.0.0.0 action=exclude protocol=tcp dport=445
add ip filter=1 source=0.0.0.0 action=exclude protocol=tcp
  dport=5554
add ip filter=1 source=0.0.0.0 action=exclude protocol=tcp
  dport=9996
add ip filter=1 source=0.0.0.0 action=include
```

Use the following hardware filters on **Layer 3 switches**:

Use the following commands on SwitchBlade and AT-8900 series switches:

```
create classifier=1 tcpdport=445
create classifier=2 tcpdport=5554
create classifier=3 tcpdport=9996
add switch hwf classifier=1 action=dis dport=all
add switch hwf classifier=2 action=dis dport=all
add switch hwf classifier=3 action=dis dport=all
```

Use the following commands on AT-8600, AT-8700XL, AT-8800, AT-8900, AT-9900, x900-48, Rapier and Rapier i series switches:

```
create classifier=1 tcpdport=445
create classifier=2 tcpdport=5554
create classifier=3 tcpdport=9996
add switch hwf classifier=1 action=dis
add switch hwf classifier=2 action=dis
add switch hwf classifier=3 action=dis
```


Whether encryption is performed in hardware or software

When no hardware encryption accelerator is installed

For a router or switch **without a hardware encryption accelerator** installed, the router or switch performs all tasks using the CPU.

The router or switch provides the following encryption algorithms performed in **software**:

- RSA—RSA Encryption
- DH—Diffie Hellman
- DES—available using software for encryption of SSL and SSH management traffic only
- 3DES—available using software for encryption of SSL and SSH management traffic only; requires a 3DES feature licence

IPsec cannot be used.

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

All

Note: *You need to have feature licences for SSL and SSH.*

Without a hardware encryption accelerator in the device, the router or switch provides the following hashing algorithms (used for authentication) performed in **software**:

- HMAC MD5
- HMAC SHA

When an AT-AR061 ECPAC, Encryption/Compression PAC is installed—AR725 and AR745 routers and Rapier 24 and Rapier 24i switches

For a router or switch using the AT-AR061 ECPAC, Encryption/Compression PAC hardware encryption accelerator, the router or switch provides the following encryption algorithms performed in **hardware**:

- DES—DES Encryption
- 3DES—Triple DES Encryption, if a 3DES feature licence is installed
- RSA—RSA Encryption
- DH—Diffie Hellman

IPsec can be used.

The router or switch provides the following hashing algorithms (used for authentication) performed in **hardware**:

- HMAC MD5
- HMAC SHA
- DES-MAC

When an AT-AR011 v2 ECMAC is installed—AR300 Series routers, AR410, AR410S, AR720, and AR740 routers

For a router using the AT-AR011 V2 ECMAC, Encryption/Compression MAC hardware encryption accelerator installed, the router provides the following encryption algorithms performed in **hardware**:

- DES—DES Encryption
- 3DES—Triple DES Encryption, if a 3DES feature licence is installed.

The router provides the following encryption algorithms performed in **software**:

- RSA—RSA Encryption
- DH—Diffie Hellman

IPsec can be used.

The router provides the following hashing algorithms (used for authentication) performed in **hardware**:

- HMAC MD5
- HMAC SHA
- DES-MAC

When models have an on-board hardware encryption processor—AR415S, AR440S, AR441S, AR442S, AR450S, AR750S, AR750S-DP and AR770S routers

These routers have an on-board hardware encryption processor, so there is no need to install any extra hardware encryption accelerator.

All encryption is done in hardware by the on-board hardware encryption processor.

The router provides the following encryption algorithms performed in **hardware**:

- DES—DES Encryption
- 3DES—Triple DES Encryption if a 3DES feature licence is installed.
- AES—AES Encryption if an AES feature licence is installed.
- RSA—RSA Encryption
- DH—Diffie Hellman

IPsec can be used.

The router provides the following hashing algorithms (used for authentication) performed in **hardware**:

- HMAC MD5
- HMAC SHA
- DES-MAC

How and when to use VRRP IP address adoption

VRRP IP address adoption is when the VRRP master router is configured to respond to some packet types destined for the Virtual Router address, even if it does not own this IP address on any of its interfaces. Below is a list of the packet types that the VRRP master will respond to if configured for VRRP IP address adoption.

- ICMP echo requests (pings)
- Telnet and SSH connection requests
- HTTP and SSL GUI management requests
- SNMP requests
- DNS relay requests

Products this Tip applies to

All routers and switches listed on page 1

Software Versions

2.6.4 and later

The benefits of address adoption

There are a number of benefits in configuring VRRP IP address adoption, including:

- The continuous accessibility of the Virtual Router IP address, even if the VRRP master is in the process of changing states.
- Being able to test that the Virtual Router is functioning by pinging the single VR IP address.
- It is easy to monitor the performance of the Virtual Router regardless of which participating router is acting as the VRRP master.
- DNS relay can continue to function using the same IP address at all times.

The potential area of concern with address adoption

There is one potential area of concern when VRRP IP address adoption is used that needs to be considered. When VRRP IP address adoption is enabled, the master VRRP router accepts packets that are destined for the Virtual Router address, even though it does not necessarily own the address on any of its interfaces. This is a deviation from RFC 2338, therefore care must be taken when VRRP IP address adoption is configured, and it is a good idea to make the Virtual Router address unique on the network so there is no potential for confusion with IP addresses configured on any physical interface on a devices in the network.

Configuring address adoption

You can configure VRRP IP Address Adoption using the new parameter, **adoptvrip**, that has been added to the **create vrrp** and **set vrrp** commands:

```
create vrrp=vr-identifier over=physical-interface
ipaddress=ipadd [adoptvrip={on|off}] [other parameters]
set vrrp=vr-identifier [adoptvrip={on|off}] [other parameters]
```

It is important to configure all the routers or switches that belong to a virtual router with the same values for the VRRP virtual router identifier, IP address, adopt VR IP address mode, advertisement interval, pre-empt mode, authentication type and password. If there are any differences in the VRRP configuration across network devices, this could cause advertisement packets to be rejected and the virtual router may not perform as desired.

Support for RADIUS accounting for 802.1x dynamic VLAN assignment

Support for RADIUS accounting is included for:

1. MAC based port authentication
2. 802.1x port authentication in single supplicant mode

Once a supplicant has been authenticated for a port, a START Accounting-Request message is sent to the RADIUS server, which tells it to start logging counters for that port. When the supplicant becomes unauthenticated (i.e. the session ends) a STOP Accounting-Request is sent to the radius server, which contains the port counter information for the RADIUS server to log. This implementation does not send update packets containing counter information during a session, only at the end of a session.

If an Accounting-Response is not received from the RADIUS server after either a START Accounting-Request or a STOP Accounting-Request is sent, and once the RADIUS module has reached its timeout and retry limit, the authorisation status of the supplicant remains unchanged, but a entry is added into the device log indicating that an Accounting Response was not received.

When a RADIUS server is defined in a device using the **add radius server** command, authentication and accounting are enabled as default. The default port number for RADIUS authentication is 1645, and 1646 for RADIUS accounting.

You can disable one or other of RADIUS authentication or accounting by setting the appropriate port number to zero. This can be done using the **port** and **accport** parameters of the **add radius server** command. An example is shown below:

```
add radius server=192.168.42.9 port=1645 accport=0
```

The above command will enable authentication on port 1645, but disable accounting.

Products this Tip applies to

All routers and switches listed on page 1 that run the versions below

Software Versions

2.7.4 and later

How to configure the firewall to allow outward-going pings but to block inward-coming pings

By default, a firewall policy will block ALL ICMP packets, in either direction. To allow any ping packets to pass through a firewall policy, you must enter the command:

```
enable firewall policy=name
icmp_forward=ping
```

This will have the effect of allowing all ping packets in both directions. So, how do we block the incoming pings? The answer depends on whether or not NAT has been enabled on the policy.

If the policy has been configured to perform NAT, then ping forwarding only really works for internally initiated pings to external addresses – the ping will be NATed appropriately. Externally initiated pings trying to reach a private address will be dropped. Any packet coming in to the public interface with a destination address on the private LAN is flagged by the firewall as an IP spoof attack because they are attempting to bypass the NATing requirement.

So, the only address that an external host can ping to is the address on the firewall's public interface. Whether or not the firewall will reply to such pings is controlled by the commands

```
enable firewall policy=name ping
disable firewall policy=name ping
```

If the **disable** command is used, then any ping to the public IP address of the firewall will be dropped.

If the policy has not been configured to perform NAT, then if you only want to allow internally initiated pings, you must use IP filters to achieve this. You cannot set up firewall rules to handle ICMP because the firewall has to handle ICMP in special conservative ways because ICMP is very commonly used for attacks.

So, you would need to create an IP filter on the public IP interface that would drop ICMP echo requests, but not drop ICMP echo replies:

```
add ip filter=1 source=0.0.0.0 protocol=icmp icmptype=8 act=exclude
add ip filter=1 source=0.0.0.0 act=include
set ip int=<public int> filter=1
```

Products this Tip applies to

All routers listed on page 1
Rapier, Rapier i, AT-8800, and
AT-9800 Series switches

Software Versions

All that support the above products

How to use firewall NAT to translate subnets

This Tip gives an example of how you can use firewall NAT to translate subnets from both Public to Private and from Private to Public. Additionally in this example we are translating a global IP address that does not belong to any interface in the network.

The following figure shows the network settings for this example.

Products this Tip applies to

All routers listed on page 1
Rapier, Rapier i, AT-8800, and
AT-9800 Series switches

Software Versions

All that support the above products

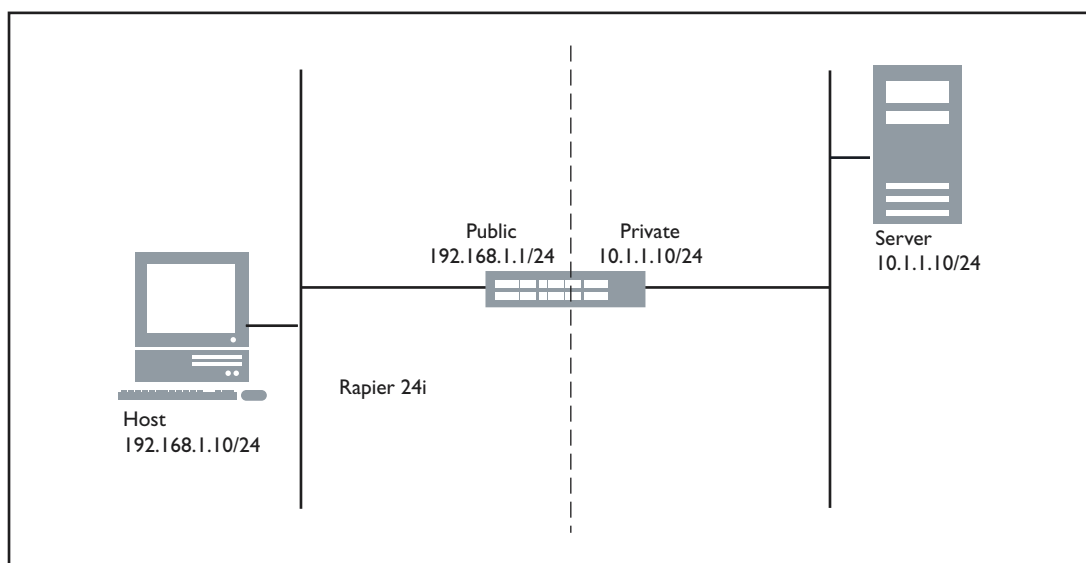


Figure 30: Using firewall NAT to translate public and private subnets

What we want is that from the public side, it appears that the LAN on the private side is using the address range 192.168.2.0/24 (even though it is actually using the address range 10.1.1.0/24).

So, specifically what we need to achieve is:

- When connecting to a destination of 192.168.2.x from a source of 192.168.1.x (i.e. from the public side), the destination address is translated to 10.1.1.x.
- When connecting to a destination of 192.168.1.x from a source of 10.1.1.x (i.e. from the private side), the source address is translated to 192.168.2.x.

A configuration example is shown below:

```
create vlan="vlan10" vid=10
add vlan="10" port=10
enable ip
add ip int=vlan1 ip=192.168.1.1
add ip int=vlan10 ip=10.1.1.1 mask=255.255.255.0
add ip route=192.168.2.0 mask=255.255.255.0 int=vlan10 next=0.0.0.0
enable firewall
create firewall policy="a"
enable firewall policy="a" icmp_f=ping
add firewall policy="a" int=vlan10 type=private
add firewall policy="a" int=vlan1 type=public
add firewall policy="a" rule=111 action=nat int=vlan1 protocol=ALL
    ip=10.1.1.0 gblip=192.168.2.0 natmask=255.255.255.0
add firewall poli="a" ru=100 action=nat int=vlan10 protocol=ALL ip=10.1.1.0
    gblip=192.168.2.0 natmask=255.255.255.0
```

Correct use of firewall NAT when FTP does not use port 21

Active and passive FTP modes can traverse a basic firewall NAT implementation. However, if the public FTP server is listening on a port other than 21, data transfer will not work under active mode. This is because the router or switch needs to monitor an FTP session. The router or switch will alter any FTP PORT command sent by the private side FTP client. It looks for FTP port commands in packets with destination TCP port 21.

Products this Tip applies to

All routers listed on page 1
Rapier, Rapier i, AT-8800, and
AT-9800 Series switches

Software Versions

All that support the above products

For example, to tell the NAT router or switch that FTP sessions also occur with a destination port of 6000 (instead of 21), use the following command:

```
add fire poli=test apprule=1 act=allo appli=ftp port=6000 int=vlan2
```

where vlan2 is the private interface.

Now the public FTP server transfers data to the client because the firewall NAT knows that port 6000 is an FTP session and so now looks for FTP PORT commands in packets with destination TCP port 6000

How to enable the firewall enhanced fragment handling mode

When using the firewall to introduce security between different sections of a network, there is a subtle matter to be aware of.

Some LAN based applications pass data in large chunks that are transmitted on Ethernet as a series of IP fragments.

In an unsecured routed or switched network, the fragments are simply forwarded as individual IP packets. However, a firewall cannot simply forward packets, but needs to examine their contents.

By default, the firewall will completely re-assemble fragmented packets before examining them. But there is a limit to how big a re-assembled packet, or how fragmented a packet, can be handled in this manner.

The firewall can handle re-assembling a packet that has been cut into up to no more than 8 fragments, and which has a combined size of up to 1730 bytes.

Note - the firewall re-assembles and examines fragmented packets even if they are being passed between two different IP interfaces that are both private members of the firewall policy.

If the applications in your network are sending packets that are cut into more than 8 fragments, or have a re-assembled size of more than 1730 bytes, then the default fragment handling of the firewall will drop these packets.

This sort of situation can easily be seen by sniffing the LAN. For example, consider the extract of an ethereal LAN trace shown below in [Figure 25](#).

It can be seen that packets 17, 18, 19 of the packet trace are a fragmented TCP packet with a combined size of at least 2960 bytes (the offset of the start of the third fragment is 2960 bytes from the start of the packet, so the combined size must be at least 2960 bytes).

Products this Tip applies to

All routers listed on page 1
Rapier, Rapier i, AT-8800, and
AT-9800 Series switches

Software Versions

2.5.1 or later

Figure 25: Ethereal LAN trace

13	1.619778	141.73.145.25	192.168.10.55	CLEARCAS v3 proc-16 Reply (Call In 12)
14	1.625093	192.168.10.55	141.73.145.25	CLEARCAS v3 proc-16 Call
15	1.625518	141.73.145.25	192.168.10.55	CLEARCAS v3 proc-16 Reply (Call in 14)
16	1.626673	192.168.10.55	141.73.145.25	CLEARCAS v3 proc-16 Call
17	1.628243	141.73.145.25	192.168.10.55	CLEARCAS v3 proc-16 Reply (Call In 16)
18	1.629398	141.73.145.25	192.168.10.55	IP Fragmented IP protocol (proto=UDP 0x11, off=1480)
19	1.630448	141.73.145.25	192.168.10.55	IP Fragmented IP protocol (proto=UDP 0x11, off=2960)
20	2.343720	141.73.145.28	141.73.145.31	NBNS Name query NB STAND-ZS<87>
21	3.093701	141.73.145.28	141.73.145.31	NBNS Name query NB STAND-ZS<87>
22	3.953138	141.73.145.28	141.73.145.31	NBNS Name query NB STAND-ZS<87>
23	4.618749	192.168.10.55	141.73.145.25	CLEARCAS [RPC retransmission of #16]v3 proc-16 Call (Reply
24	4.620285	141.73.145.25	192.168.10.55	CLEARCAS [RPC retransmission of #17]v3 proc-16 Reply (Call
25	4.621439	141.73.145.25	192.168.10.55	IP Fragmented IP protocol (proto=UDP 0x11, off=1480)
26	4.622489	141.73.145.25	192.168.10.55	IP Fragmented IP protocol (proto=UDP 0x11, off=2960)
27	4.703109	141.73.145.28	141.73.145.31	NBNS Name query NB STAND-ZS<87>
28	5.453077	141.73.145.28	141.73.145.31	NBNS Name query NB STAND-ZS<87>
29	6.296839	141.73.145.28	141.73.145.31	NBNS Name query NB STAND-ZS<87>
30	6.483306	141.73.145.4	141.73.145.31	NBNS Name query NB STUTTGART<1c>
31	7.046809	141.73.145.28	141.73.145.31	NBNS Name query NB STAND-ZS<87>
32	7.233371	141.73.145.4	141.73.145.31	NBNS Name query NB STUTTGART<1c>
33	7.618632	192.168.10.55	141.73.145.25	CLEARCAS [RPC retransmission of #16]v3 proc-16 Call (Reply
34	7.620159	141.73.145.25	192.168.10.55	CLEARCAS [RPC retransmission of #17]v3 proc-16 Reply (Call
35	7.621415	141.73.145.25	192.168.10.55	IP Fragmented IP protocol (proto=UDP 0x11, off=1480)
36	7.622465	141.73.145.25	192.168.10.55	IP Fragmented IP protocol (proto=UDP 0x11, off=2960)

That packet gets dropped when it reaches the firewall. The results in the client device re-requesting the data packet, see the re transmission of the request at packet 23, and the retransmission of the reply that is packets 24,25,26. The re-request/re-reply happens again at packets 33-36.

Eventually the client will just give up re-requesting, and the data transfer will fail.

The solution to this type of problem is to enable firewall enhanced fragment handling. To enable enhanced fragment handling, use the command:

```
enable firewall policy=policy-name
  [fragments={icmp|udp|other}[,...]]
```

If enhanced fragment handling is enabled, the default maximum number of fragments that an IP packet may consist of is 20.

To alter the maximum number of fragments that a fragmented IP packet may consist of when packet fragment handling is enabled, use the command:

```
set firewall maxfragments=8..50
```

When enhanced fragment handling is enabled, there is no upper limit on the re-assembled size of the packets. This is because in the enhanced case the firewall does not actually reassemble the packet. It groups up all the fragments and then passes them through the firewall at the same time.

How to use the HTTP proxy (application gateway)

The firewall's HTTP proxy is provided to allow for configuration of filtering on HTTP sessions based on the URLs requested. The proxy can also be configured to filter the setting of all cookies, or cookies requested from servers in a specific domain.

The proxy may require additional software licences, depending on the model and software version.

Products this Tip applies to

All routers listed on page 1
Rapier, Rapier i, AT-8800, and
AT-9800 Series switches

Software Versions

All that support these products

Note: *You should only use the HTTP proxy if you intend to use one of these filtering facilities. There is no security advantage in having proxy configured when filtering is not needed—it just creates unnecessary work for the CPU. The firewall is already doing stateful inspection of all the HTTP sessions passing through it, which provides protection from intruder attacks.*

Also be aware that HTTP proxy configuration can potentially conflict with the router's Graphical User Interface (GUI) configuration facility. You can avoid this conflict simply by configuring the GUI to operate on a different port—such as 8080—before you configure the HTTP proxy.

Configuration example

The example below shows the commands for a firewall configuration with the HTTP proxy service. For this example, the firewall public interface is assumed to be a PPP interface (such as a PPP over ATM interface).

Change the GUI service to use port 8080 to avoid conflict with the HTTP proxy service.

```
set http server po=8080
```

Configure the firewall with HTTP Proxy:

```
enable firewall
create firewall policy=test
add firewall policy=test int=vlan1 type=private
add firewall policy=test int=ppp0 type=public
add firewall poli=test prox=http int=vlan1 gblin=ppp0
add firewall poli=test nat=enhanced int=vlan1 gblin=ppp0
add firewall policy=test httpfilter=filter.txt direction=out
```

Create filter.txt separately, containing a list of URLs to be filtered. For the format to use, see the Firewall chapter of your router or switch Software Reference.

Configure the HTTP proxy to discard all HTTP cookie sets from all responses:

```
disable firewall policy=test httpcookies
```

How to use the `trustprivate` parameter on the firewall to block users on the private side from accessing the device

At first glance, it might seem that simply adding a private interface to the firewall policy, and specifying `trustprivate=off` on the interface should block all access from the private VLAN to the management address(es) of the router or switch, but actually, it does not.

To understand the true meaning of the `trustprivate` parameter, we have to look at how the firewall used to work before the `trustprivate` parameter was added.

It used to be that if you configured a rule to block traffic arriving on a private interface, like:

```
add firewall policy=1 rule=1 int=vlan12 prot=tcp port=80
  remoteip=<ip address of the router> act=deny
```

then the firewall would only apply the rule to packets that were destined to go out through the public interface of the firewall. So, if the destination address of the packet was the router itself, or was reached via another private interface, then the firewall did not examine the packet at all. So, effectively, the firewall completely trusted the hosts on the private LAN, and was only interested in monitoring their access to devices beyond the public interface.

So, if you configure:

```
add firewall policy=1 int=vlan12 type=private trustprivate=yes
```

then the firewall will continue to operate in the 'pre-trustprivate' fashion.

In other words, if you configure:

```
add firewall policy=1 int=vlan12 type=private trustprivate=yes
add firewall policy=1 rule=1 int=vlan12 prot=tcp port=80
  remoteip=<ip address of the router> act=deny
```

then the firewall will NOT block hosts on VLAN12 from web-browsing to the router.

But, if you configure:

```
add firewall policy=1 int=vlan12 type=private trustprivate=no
```

then the firewall will examine all packets that arrive from hosts on VLAN12, even if the packets are not aimed at destinations out beyond the public interface. So, if there is a rule configured on the firewall that blocks certain packets not destined to a public destination, such as a rule like:

```
add firewall policy=1 rule=1 int=vlan12 prot=tcp port=80
  remoteip=<ip address of the router> act=deny
```

then, this rule will be applied, and the packets which match it will be dropped.

So, for example, if you had a firewall with 3 private interfaces, and you wanted to block users on all those interfaces from accessing the router, then you would have configure all the interfaces

Products this Tip applies to

All routers listed on page 1
Rapier, Rapier i, AT-8800, and
AT-9800 Series switches

Software Versions

2.6.4 and later

with **trustprivate=off**, and explicitly configure rules to block access from any of the private VLANs to any of the private IP addresses of the router.

If the configuration of the router was something like:

```
add ip int=vlan10 ip=192.168.10.254
add ip int=vlan11 ip=192.168.11.254
add ip int=vlan12 ip=192.168.12.254

enable firewall
create firewall policy=1
enable firewall policy=1 icmp_f=all

add firewall policy=1 int=vlan12 type=private trustprivate=no
add firewall policy=1 int=vlan11 type=private trustprivate=no
add firewall policy=1 int=vlan10 type=private trustprivate=no
add firewall policy=1 int=vlan1 type=public
add firewall policy=1 int=eth0 type=public
add firewall poli=1 nat=enhanced int=vlan11 gblin=eth0
  gblip=193.179.159.123
add firewall poli=1 ru=1 ac=allo int=eth0 prot=tcp po=22
  ip=193.179.159.123 gblip=193.179.159.123
```

then, to ensure that no hosts on the private LANs could access the router, it would be necessary to add the rules:

```
add firewall poli=1 rule=2 int=vlan10 prot=all
  remoteip=192.168.10.254 act=deny
add firewall poli=1 rule=3 int=vlan10 prot=all
  remoteip=192.168.11.254 act=deny
add firewall poli=1 rule=4 int=vlan10 prot=all
  remoteip=192.168.12.254 act=deny

add firewall poli=1 rule=5 int=vlan11 prot=all
  remoteip=192.168.10.254 act=deny
add firewall poli=1 rule=6 int=vlan11 prot=all
  remoteip=192.168.11.254 act=deny
add firewall poli=1 rule=7 int=vlan11 prot=all
  remoteip=192.168.12.254 act=deny
add firewall poli=1 rule=8 int=vlan12 prot=all
  remoteip=192.168.10.254 act=deny
add firewall poli=1 rule=9 int=vlan12 prot=all
  remoteip=192.168.11.254 act=deny
add firewall poli=1 rule=10 int=vlan12 prot=all
  remoteip=192.168.12.25 act=deny
```

How to use the firewall to control Internet access on the basis of private hosts' MAC addresses

A network administrator may wish to limit Internet access to just a certain set of specific hosts on their LAN. To avoid circumvention of the policy by IP spoofing, the Network Administrator wishes to identify the hosts by MAC address, not by IP address.

The AlliedWare firewall is able to limit Internet access on the basis of host's MAC address, and can store the list of allowed MAC addresses on a single RADIUS server.

Products this Tip applies to

All routers listed on page 1 that run the versions below

Rapier i, AT-8800, and AT-9800 Series switches

Software Versions

2.7.5 and later

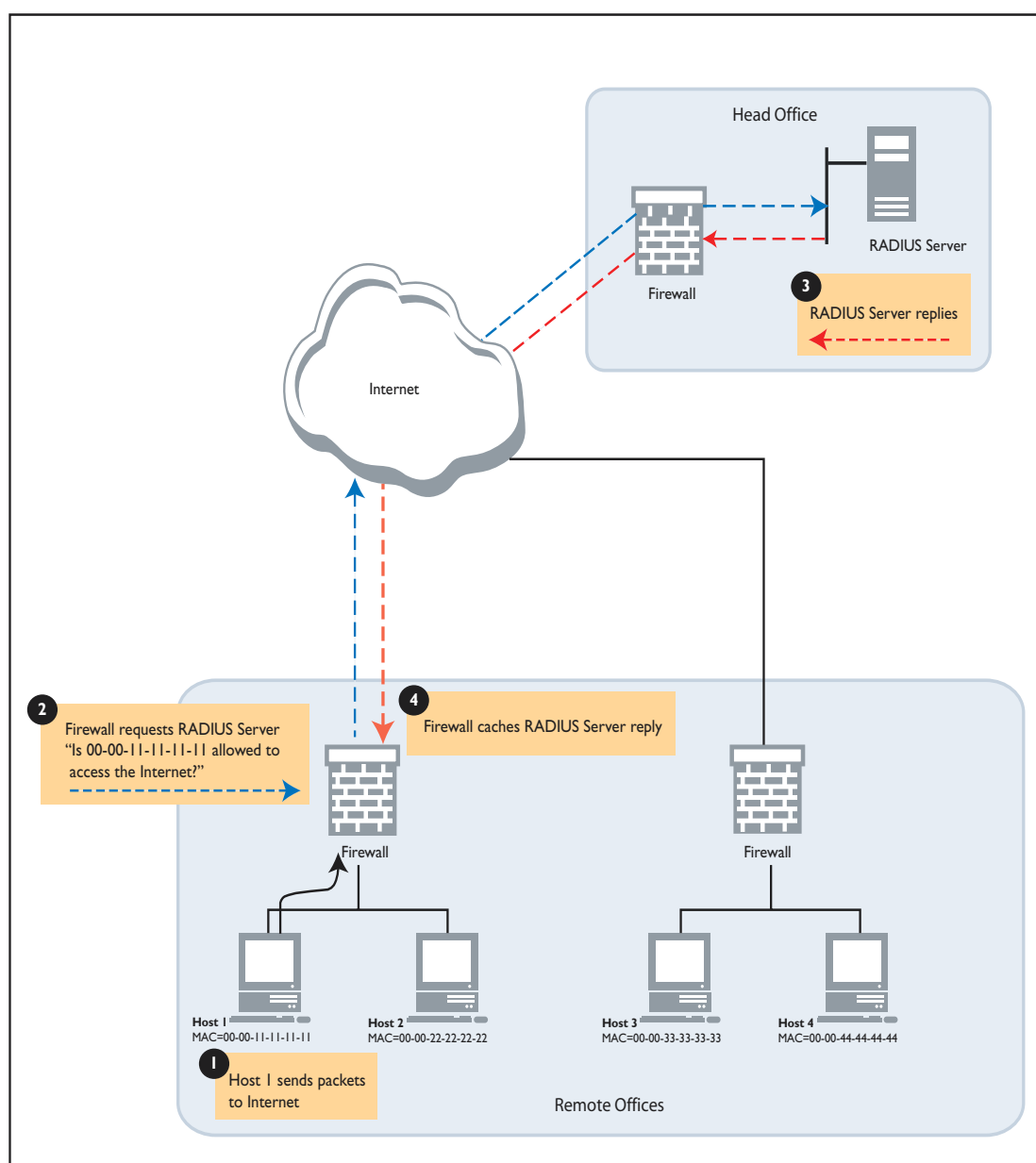


Figure 31: Limiting Internet access based on private host's MAC address

Configuration on the firewall

The configuration required to enable this process is as follows:

1. Define a RADIUS server to which to make the queries:

```
add radius server=<server-address> secret=<secret-key> port=1812
```

2. Enable the firewall and create a policy (in this case, the policy is performing NAT, but that is not always required):

```
enable firewall
create firewall policy=nat
enable firewall policy=nat icmp_f=all
add firewall policy=nat int=ppp0 type=public
add firewall policy=nat int=vlan1 type=private
add firewall policy=nat nat=enhanced int=vlan1 gblint=ppp0
```

3. Ensure that the firewall will allow RADIUS requests out:

```
add fire poli=nat rule=1 int=vlan1 prot=udp port=1812 act=allow
```

4. Create a firewall rule that allows ALL data out PROVIDED that the source MAC address is approved by the RADIUS server. Note that this rule is applied to the PRIVATE interface of the firewall.

```
add fire poli=nat rule=2 int=vlan1 prot=all act=allow
add fire poli=nat rule=2 list=macradius
```

Configuration on the RADIUS server

On the RADIUS server, each allowed MAC address requires an entry in the “users” file.

The format of the entry is:

```
-----
User name = [00-00-ab-cd-ed-gh]
Auth-Type = local
Password = "allowdeny"
Framed-IP-Address = "1.1.1.1"
-----
```

Also, in the RADIUS “clients” file, there needs to be an entry for the router's WAN IP address, with a secret key the same as the one configured in the **add radius server** command above.

Now, one thing that might seem surprising about the format of the entry in the “users” file is the fact that it contains a framed-IP-address. Typically, the framed-IP-address parameter is used for allocating addresses to dial-in users. But, in this case, there is no IP address allocation going on, so what is the purpose of the framed-IP-address parameter?

Well, the fact is that for this MAC authentication, we are using RADIUS in a slightly non-standard way. Normally, for user authentication, you create an entry in the Radius USER file with a particular username and password. When the Router sends a username and password to the RADIUS server, the server looks in its USER file for a match on that username/password. If it finds a match then it sends an access-accept.

BUT for the MAC authentication, we need to be able to say that certain MAC addresses are explicitly allowed, and certain MAC addresses are explicitly denied. (You may only need to say that certain MAC addresses are explicitly allowed, but in general, we also need to be able to support the ability say that certain MAC addresses are explicitly denied).

So, we need to have entries in the USER file for both allowed and denied MAC addresses. But, if an entry for a particular MAC address is present in the USER file, RADIUS will always send back an access-accept when the router sends a RADIUS query for that MAC address. So, how does the router know if the RADIUS server is explicitly allowing or explicitly denying that MAC when all the access-accept message tells us is that the USER file has an entry for that MAC address?

We need the RADIUS server to send another parameter in the access-accept message. We have decided to use the Framed-IP-address parameter.

- Framed-IP-address = 0.0.0.0 means “that MAC is denied”.
- Framed -IP-address = 1.1.1.1 means “that MAC is allowed”.

So, if the Radius server sends back an access-accept that contains Framed-IP-address 0.0.0.0, the message is saying “I have an entry for the MAC address in my USERS file, and that entry states that the MAC is DENIED”.

If the Radius server sends back an access-accept that contains Framed-IP-address 1.1.1.1, the message is saying “I have an entry for the MAC address in my USERS file, and that entry states that the MAC is ALLOWED”.

Now, if the Radius server sends back an access-accept that contains NO Framed-IP-address. We also deem that message to be saying “I have an entry for the MAC address in my USERS file, and that entry states that the MAC is DENIED”. Also, if the Radius server has NO entry for a particular MAC address, it will send back an access-reject, which is also interpreted as stating “that MAC is denied”.

Caching results

It would, of course, be highly inefficient to send a RADIUS request for EVERY single packet that tries to pass through the firewall. So, the firewall actually caches the results of RADIUS requests, and looks in the cache for a given packet's MAC address. It only queries the RADIUS server if the MAC address is not found in the result cache.

This MAC address cache stores results for a time length specified by using the command:

```
set firewall policy maccachetimeout=max-age
```

The default timeout is 1440 minutes (24 hours).

The cache can be cleared at any time using the command

```
RESET FIREwall POLIcy=policy-name MACCACHE
```

The current contents of the cache can be seen using the command:

```
SHoW FIREwall POLIcy=policy-name MACCACHE
```


How to configure a timeout on particular UDP ports in a firewall policy

You can set a timeout period for inactive UDP sessions on a given policy to apply to either:

- all sessions irrespective of the UDP port number
- a particular UDP port, or group of ports

If you configure a timeout on a particular UDP port, the timeout that you set overrides the timeout for the firewall policy for that port only. For all other UDP ports the firewall policy timeout applies.

Products this Tip applies to

All routers listed on page 1 that run the versions below

Rapier, Rapier i, AT-8800, and AT-9800 Series switches

Software Versions

2.7.5 or later

To apply a timeout to a particular UDP port or group of ports, use the command:

```
add firewall policy=<name> udpporttimeout=<port-number-or-list>
    timeout=<timeout>
```

Or you can modify the timeout for UDP ports with the command:

```
set firewall policy=<name> udpporttimeout=<port-number-or-list>
    timeout=<timeout>
```

The *<port-number-or-list>* value in the **udpporttimeout** parameter value can be a single port number, or a list of port numbers separated by commas. The **timeout** parameter is a number from 0 to 43200 in minutes.

To view the UDP port timeout settings that are configured for a particular firewall policy, use the command:

```
show firewall policy=<name> udpporttimeout
```

An example output from the **show firewall policy udpporttimeout** command is displayed below:

```
Policy : 1
Default UDP Timeout (s) : 1200
Number of Configured UDP Port Timeouts : 1

  UDP Port          Timeout (s)
  -----
           80          3600
```

Firewall messages relating to SYN attacks

When the firewall detects a SYN attack, it might emit one or more alert messages.

For more information about its response to SYN attacks, see "[About the firewall's aggressive mode](#)" on page 47.

Three messages and their meanings are listed below:

1. SYN attack from 192.168.2.17 is underway

A SYN attack is a specific attack in which the attacker sends a series of TCP SYN packets in an attempt to exhaust all the resource on the victim as the victim tries to make TCP sessions to respond to all those SYN packets. This message indicates that the firewall has detected that such an attack is underway.

2. Host 192.168.2.14 has exceeded its per host limit

This means that the specified host has 64 TCP sessions that haven't yet reached the established state and has tried to start another one. 64 is the per host limit for non established sessions, so the new session is refused.

It is important to note that the limit is not on the total number of sessions that can be alive from a single host, but the limit is on the number of sessions from a single host that can be in an as-yet unestablished state (i.e. sessions that are still in the process of the SYN; SYN-ACK; ACK exchange).

Note: *When a host has exceeded its per host limit, it can't access the public network any more.*

3. Policy <name>'s SYN queue state changed to: Semi Aggressive

This means that there are one or more hosts that have over 32 entries in the TCP SYN queue (suspicious hosts). Firewall will try to age these entries out more quickly. This involves sending the standard number of SYN retries more rapidly.

Products this Tip applies to

All routers listed on page 1 that run the versions below

Rapier, Rapier i, AT-8800, and AT-9800 Series switches

Software Versions

2.7.3 or later