Late-Breaking Features

Software Release 2.6.6
For AT-9900 Series Switches

Introduction .......................................................................................................3
Link Aggregation Control Protocol (LACP) .........................................................3
Loopback Interface ............................................................................................3
BGP Route Flap Damping ..................................................................................4
  New Commands ............................................................................................4
  Modified Commands ....................................................................................11
New Community Format for BGP ....................................................................14
Tag Distinction of Static Routes for BGP Import ...............................................15
Soft Resetting of BGP Peers ..........................................................................16
BGP Peer Policy Templates .............................................................................18
  New Commands ............................................................................................18
  Modified Commands ....................................................................................20
IP Prefix Lists and Extensions to Route Map Attributes ....................................23
  New Commands ...........................................................................................23
  Modified Commands ...................................................................................27
BGP Route Reflection .......................................................................................29
  Modified Commands ...................................................................................29
BGP Fast Fallover .............................................................................................31
  Modified Commands ...................................................................................31
BGP Private AS Numbers ...............................................................................33
  Modified Commands ...................................................................................33
MD5 Authentication for BGP ..........................................................................35
  Modified Commands ...................................................................................35
Importing BGP routes into OSPF ......................................................................37
  Enabling BGP route import .........................................................................37
  Limiting the number of routes ....................................................................37
  Advertising desired routes ..........................................................................37
  Configuration example ..................................................................................38
BGP Backoff .....................................................................................................39
  Configuration ...............................................................................................39
  Consecutive Backoffs ...................................................................................40
  New Commands ............................................................................................40
BGP memory accounting ..................................................................................42
  New Commands ............................................................................................42
Actions for QoS Traffic Classes and Flow Groups .........................................44
L2 QoS Actions in Hardware Filters ..................................................................46
Classification By L4 Mask and Inner VLAN Settings ......................................47
Extensions to Discarding of Source-Routed IP Packets ....................................48
Late-Breaking Features

QoS Counters .................................................................49
Enable/Disable Port Egress Queue and Flow Control .....................53
802.1x and RADIUS Servers Enhancement for EAP-TLS ..................55
Steps in the Authentication Process ........................................55
Logging and SNMP Traps for PIM-SM ......................................57
PIM4 Hash Mask Length Support ...........................................59
Modified Commands .......................................................59
OSPF Not-So-Stubby Area Option (NSSA) .................................60
Modified Commands .......................................................60
OSPF Auto Cost Calculation ...............................................62
IP DSCP Overide ..........................................................63
CPU Transmit User Priority Override ....................................63
CPU Transmit Queue Priority Override .................................64
Syslog Facility Override ..................................................65
Modified Commands .......................................................65
Temperature and Fan Monitoring ..........................................66
Use of Default Route by Proxy ARP .......................................67
Modified Commands .......................................................67
Increase in VLAN Name Length .........................................69
RSTP BPDU Loopback Detection .........................................69
Telnet Session Limit and Timeout ......................................70
Extensions to Stacking .....................................................71
  Synchronising Time Across Stacks ..................................71
  Copy command ........................................................71
ICMP Router Discovery Advertisements ................................73
  Configuration Procedure ..............................................75
Adopting the VRRP IP Address ...........................................76
  Benefits of VRRP IP Address Adoption ............................76
  Risks of VRRP IP Address Adoption ...............................76
  Recommendations ....................................................76
  Configuration of VR IP Address Adoption .......................77
Static IGMP .................................................................78
IPsec NAT-Traversal .......................................................79
  Basic NAT-T Operations ..............................................79
  NAT-T on the Switch ................................................80
Commands Modified for NAT-T ........................................81
  ISAKMP policy commands ..........................................82
Introduction

This Late-Breaking Features note describes a number of features for Software Release 2.6.6 for AT-9900 Series Switches. These features are not described in the Software Reference included on this CD-ROM. An updated Software Reference and the latest release note will be available from: www.alliedtelesyn.co.nz/support/at9900

WARNING: Information in this document is subject to change without notice and does not represent a commitment on the part of Allied Telesyn International. While every effort has been made to ensure that the information contained within this document and the features and changes described are accurate, Allied Telesyn International can not accept any type of liability for errors in, or omissions arising from the use of this information.

Corrections to the Software Reference

As well as the extra features described here, please note:

- References to the IP Security (IPsec) chapter of the Software Reference were excluded from the Table of Contents and other navigational aids. A link to the IPsec chapter is provided at the end of the Contents bookmarks.
- The firewall is not supported on AT-9900 Series switches.
- Appletalk is not supported on AT-9900 Series switches.
Link Aggregation Control Protocol (LACP)

The implementation of the link access control protocol (LACP) follows the IEEE 802.3-2002 Standard, “CSMA/CD access method and physical layer specifications.”

The LACP protocol operates where systems are connected over multiple communications links. In this configuration, links that are controlled by LACP are constantly monitored and are automatically added to, or removed from, trunk groups (or aggregated links).

Information about LACP and its commands is included at the end of this Late-Breaking Features Note.

Local Interfaces

A Loopback interface is one that is always available for higher layer protocols to use and advertise out into a network. A local interface is assigned an IP address, but does not have the usual requirement of needing to be connected to a lower layer physical entity. It is the lack of an attachment to a physical entity that allows the perception of it always being accessible via the network.

Loopback interfaces can be utilised by a number of protocols for various purposes. They can be used to improve access to a device, as well as increasing the reliability, security, scalability and protection offered by the device. In addition they can add flexibility and simplify management, information gathering and filtering.

Information about local interfaces and their commands is included at the end of this Late-Breaking Features Note.
BGP Route Flap Damping

Route flap damping is a mechanism for suppressing the generation of update messages for unstable BGP routes. It is accomplished via the maintenance of a Figure of Merit (FoM) value for each known BGP route, which is used to quantify the route’s history of stability (or lack thereof). Each time a route becomes unreachable, the FoM is incremented by 1000. If the FoM goes above a user-configurable suppression threshold, the route is suppressed (the route is deemed unusable). The FoM decays exponentially (at a rate defined by a user-configurable half life interval) and upon falling below the user-configurable reuse threshold, it becomes unsuppressed and is available for selection.

To prevent routing loops, route flap damping is not applied to any routes that are learned internally to an AS (via IBGP). Therefore, it will only be applied to routes learned via EBGP.

To use parameter sets, a routemap must be created that will attach parameter set IDs to routes. Once this routemap is attached to a BGP peer, any routes that arrive from that peer are treated by the routemap to attach the desired parameter set ID. From here there are two possible actions:

- If BGP damping is enabled as a module
  - If a match is found and the defined parameter set is present and enabled, it is attached to the route.
  - If a match is found and the parameter set defined is not present or it is present but disabled, the default set is attached.
  - If no match is found, the default parameter set is attached.

- If BGP damping is enabled for an individual parameter set(s) but not as a module
  - If a match is found and the defined parameter set is present and it is enabled, it is attached to the route.
  - If a match is found and the parameter set defined is not present or it is present but disabled, no damping occurs.
  - If no match is found, no damping occurs.

New Commands

Use the following new commands to configure BGP damping.

create bgp damping parameterset

Syntax

```plaintext
CREATE BGP DAMPING PARAMETERSET=1..100
[DESCRIPTION=description] [SUPPRESSION={DEFAULT | 1..20000}] [REUSE={DEFAULT | 1..20000}]
[HALFLIFE={DEFAULT | 1..45}] [MAXHOLD={DEFAULT | 1..8}]
```

where:

- **description** is a character string, 1 to 63 characters in length. Valid characters are any printable characters except '?' and '"'. If description contains spaces, it must be enclosed in double quotes.
Description

This command creates a parameter set for BGP Route Flap Damping. If an allowed parameter is not given in the command, the default value for that parameter is used. The status of a newly created parameter set will be that of the BGP Route Flap Damping feature (see the `enable bgp damping` command for further details).

The `parameterset` parameter specifies the number of the parameter set. Two parameter sets cannot be defined to use the same ID number. The default parameter set will use value 0.

The `description` parameter specifies a free format description of this parameter set. This parameter is purely for administrative convenience, and has no effect on the operation of BGP route flap damping.

The `suppression` parameter specifies the threshold above which a route with a higher FoM will be suppressed. The suppression value must be greater than or equal to the reuse value. The default value is 2000.

The `reuse` parameter specifies the threshold below which a route with a lower FoM will not be suppressed. The reuse value must be less than or equal to the suppression value. The default value is 750.

The `halflife` parameter specifies the time interval within which the route's FoM will be decreased by half, if the route remains stable. The default value is 15 minutes.

The `maxhold` parameter specifies the ratio (in the form of 1:maxhold) of the half-life value for a route to the maximum time a route will be held in the suppressed state, regardless of the stability history. The range of values (1 to 8) give ratios of 1:1 to 1:8 and the default value is 4, giving a ratio of 1:4.

Examples

To create a BGP Route Flap Damping parameter set 3 with a half-life of 5 minutes and a suppression threshold of 3000, use the command:

```
create bgp damping parameterset=3 halflife=5 suppression=3
```
disable bgp damping

Syntax
DISABLE BGP DAMPING [PARAMETERSET={ALL|0..100}]

Description
This command disables monitoring and suppression of flapping BGP routes. This command will clear any route stability history information and may be used to turn off this feature temporarily for configuration changes.

The parameterset parameter specifies the user-created parameter set to be disabled. If the parameterset parameter is not supplied, all bgp damping will be disabled. The default parameter set (0) cannot be disabled unless all bgp damping is being disabled. Use the set bgp damping parameterset or purge bgp damping commands to alter the default parameter set. If the parameterset parameter is used for the last enabled parameter set, then the feature will be disabled (as if no parameter set were specified).

Example
To disable BGP Route Flap Damping for all enabled parameter sets and the feature, use the command:

    disable bgp damping

To disable BGP Route Flap Damping for parameter set 3 only, use the command:

    disable bgp damping parameterset=3

Note: If parameter set 3 is the last enabled parameter set, the feature will be disabled.

enable bgp damping

Syntax
ENABLE BGP DAMPING [PARAMETERSET={ALL|0..100}]

Description
This command enables the monitoring and suppression of flapping BGP routes. If the parameterset parameter is specified and the BGP Route Flap Damping feature is disabled, the feature will become enabled. The default parameter set will be used for all routes that have not been bound to an user created parameter set via a route map unless only individual parameter sets are enabled.

Once the BGP Route Flap Damping feature is enabled, the feature cannot be re-enabled with this command - the command may still be used with the parameterset parameter specified. Use the command show bgp damping to see the state of the BGP Route Flap Damping feature.

The parameterset parameter specifies an existing user-created parameter set to be enabled. If the parameterset parameter is not supplied, all bgp damping will be enabled. The default parameter set (0) will not be enabled unless all bgp damping is being enabled. Use the set bgp damping parameterset or purge bgp damping commands to alter the default parameter set. If the parameterset parameter is used to enable the first parameter set (no other parameter sets enabled) and the feature is disabled, the feature will also be enabled.
Example
To enable BGP Route Flap Damping for all existing parameter sets and for the feature, use the command:

```bash
enable bgp damping
```

To enable BGP Route Flap Damping for parameter set 3 and for the feature, use the command:

```bash
enable bgp damping parameterset=3
```

If only some of the parameter sets are enabled, to enable BGP Route Flap Damping for all other parameter sets, use the command:

```bash
enable bgp damping parameterset=all
```

**purge bgp damping**

**Syntax**
PURGE BGP DAMPING

**Description**
This command purges all configuration information relating to the BGP route flap damping. All user defined parameter sets are destroyed. Any accumulated route stability history is cleared, and the feature is disabled.

All current configuration and stability history information will be lost. Use with extreme caution!

**Example**
To purge BGP Route Flap Damping, use the command:

```bash
purge bgp damping
```

**reset bgp damping**

**Syntax**
RESET BGP DAMPING [PARAMETERSET={ALL|0..100}]

**Description**
This command clears the BGP route flap damping stability history for all BGP routes. If the `parameterset` parameter is specified, only the routes attached to this parameter set will be affected.

The `parameterset` parameter specifies the user-created parameter set to have all attached routes reset. If the `parameterset` parameter is not supplied or if the option `all` is specified, all bgp damping routes will be reset.

**Example**
To clear BGP Route Flap Damping history for all routes attached to parameter set 3, use the command:

```bash
reset bgp damping parameterset=3
```
set bgp damping parameterset

Syntax

```plaintext
SET BGP DAMPING PARAMETERSET={DEFAULT|0|1..100} [DESCRIPTION=description] [SUPPRESSION={DEFAULT|1..20000}] [REUSE={DEFAULT|1..20000}] [HALFLIFE={DEFAULT|1..45}] [MAXHOLD={DEFAULT|1..8}]
```

where:

- `description` is a character string, 1 to 63 characters in length. Valid characters are any printable characters except '?' and '"'. If description contains spaces, it must be enclosed in double quotes.

Description

This command changes the settings of a predefined parameter set for BGP Route Flap Damping. If a parameter is not used in the command, its associated value does not change.

This command may only be used when BGP damping has been disabled for the parameter set specified or if the whole feature is disabled.

The `parameterset` parameter specifies the number of the parameter set. The value of '0' refers to the default parameter set.

The `description` parameter specifies a free format description of this parameter set. This parameter is purely for administrative convenience, and has no effect on the operation of BGP route flap damping.

The `suppression` parameter specifies the threshold above which a route with a higher FoM will be suppressed. The suppression value must be greater than or equal to the reuse value. The default value is 2000.

The `reuse` parameter specifies the threshold below which a route with a lower FoM will not be suppressed. The reuse value must be less than or equal to the suppression value. The default value is 750.

The `halflife` parameter specifies the time interval within which the route's FoM will be decreased by half, if the route remains stable. The default value is 15 minutes.

The `maxhold` parameter specifies the ratio of the half-life value for a route to the maximum time a route will be held in the suppressed state, regardless of the stability history. The default value is a ratio of 1:4.

Example

To change the half-life to 20 minutes for parameter set “3”, use the command:

```plaintext
set bgp damping parameterset=3 halflife=20
```
show bgp damping

Syntax
SHOW BGP DAMPING [ROUTES]

Description
This command displays information about the BGP Route Flap Damping configuration and operation.

If the routes parameter is specified, information about the routes in the suppression engine is displayed.

Figure 1: Example output from the show bgp damping command

---

<table>
<thead>
<tr>
<th>BGP Route Flap Damping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status: ENABLED</td>
</tr>
<tr>
<td>Routes in Engine: 40</td>
</tr>
<tr>
<td>Monitored Routes: 2</td>
</tr>
<tr>
<td>Suppressed Routes: 9</td>
</tr>
<tr>
<td>Forgotten Routes: 46</td>
</tr>
</tbody>
</table>

Parameter set 0

- Default configuration
  - Current State: DISABLED
  - Suppression: 2000
  - Half life: 15 min
  - Maximum Hold: 1:4

Parameter set 1

- Severely penalise unreachable to test network
  - Current State: DISABLED
  - Suppression: 1200
  - Half life: 15 min
  - Maximum Hold: 1:4

Parameter set 6

- Current State: ENABLED
  - Suppression: 1500
  - Half life: 10 min
  - Maximum Hold: 1:5

---

Table 1: Parameters displayed in the output of the show bgp damping command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>One of “Yes” or “No”. “Yes” means that BGP Route Flap Damping is enabled on the switch. “No” means that BGP Route Flap Damping is disabled on the switch.</td>
</tr>
<tr>
<td>Monitored Routes</td>
<td>The counter for routes that are not suppressed, but are being monitored by the suppression engine</td>
</tr>
<tr>
<td>Routes in Engine</td>
<td>The number of routes for which the suppression engine is currently maintaining a FoM.</td>
</tr>
<tr>
<td>Suppressed Routes</td>
<td>The counter for routes that are currently being suppressed by the suppression engine</td>
</tr>
<tr>
<td>Forgotten Routes</td>
<td>The number of routes that incurred a damping penalty in the past, but have had that penalty forgotten due to those routes experiencing a sufficient period of stability. Note that if the same route is forgotten more than once, it will be counted more than once. The counter will not decrement if a previously forgotten route incurs a new damping penalty.</td>
</tr>
</tbody>
</table>
Table 1: Parameters displayed in the output of the **show bgp damping** command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameterset n</td>
<td>The number of the parameter set, numbering occurs at time of creation.</td>
</tr>
<tr>
<td>Description line</td>
<td>The description given to the parameter set when created (or modified when using the SET BGP DAMP command). If no description has been given, the display is “&lt;Parametersetn&gt;” where n is the number of the parameter set.</td>
</tr>
<tr>
<td>Current state</td>
<td>The current state of the parameter set; one of “enabled” or “disabled”</td>
</tr>
<tr>
<td>Configuration Values</td>
<td>The configuration settings of the parameter set</td>
</tr>
<tr>
<td>Suppression</td>
<td>The value above which a route advertisement will be suppressed.</td>
</tr>
<tr>
<td>Reuse</td>
<td>The value below which a suppressed route will be selectable.</td>
</tr>
<tr>
<td>Half Life</td>
<td>The rate of decay to be given as the half-life for a route that is stable.</td>
</tr>
<tr>
<td>Maximum Hold</td>
<td>The ratio of the half life to the maximum time a route may be suppressed, regardless of stability history.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Prefix/Mask</th>
<th>Next Hop</th>
<th>Current State</th>
<th>Pen (FoM)</th>
<th>Num Flaps</th>
<th>Last St Change</th>
<th>Next St Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>192.168.5.0/24</td>
<td>1.1.1.1</td>
<td>&gt;eM</td>
<td>992</td>
<td>1</td>
<td>00:00:10</td>
<td>01:23:40</td>
</tr>
<tr>
<td>0</td>
<td>192.168.10.0/24</td>
<td>1.1.1.1</td>
<td>&gt;eM</td>
<td>992</td>
<td>1</td>
<td>00:00:10</td>
<td>01:23:40</td>
</tr>
<tr>
<td>0</td>
<td>192.168.7.0/24</td>
<td>1.1.1.1</td>
<td>&gt;eS</td>
<td>2961</td>
<td>3</td>
<td>00:00:20</td>
<td>00:29:45</td>
</tr>
<tr>
<td>0</td>
<td>192.168.3.0/24</td>
<td>1.1.1.1</td>
<td>&gt;eS</td>
<td>4938</td>
<td>5</td>
<td>00:00:20</td>
<td>00:40:50</td>
</tr>
<tr>
<td>0</td>
<td>192.168.9.0/24</td>
<td>1.1.1.1</td>
<td>&gt;eM</td>
<td>992</td>
<td>1</td>
<td>00:00:10</td>
<td>01:23:40</td>
</tr>
<tr>
<td>0</td>
<td>192.168.6.0/24</td>
<td>1.1.1.1</td>
<td>&gt;eM</td>
<td>1976</td>
<td>2</td>
<td>00:00:20</td>
<td>01:38:40</td>
</tr>
<tr>
<td>0</td>
<td>192.168.4.0/24</td>
<td>1.1.1.1</td>
<td>&gt;eS</td>
<td>1984</td>
<td>2</td>
<td>00:00:10</td>
<td>00:21:05</td>
</tr>
</tbody>
</table>

Figure 2: Example output from the **show bgp damping routes** command
Modified Commands

Use the highlighted new parameters in existing commands to configure BGP damping.

**add ip routemap**

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD IP ROUTEMAP=route-map ENTRY=1..4294967295 [ACTION={INCLUDE</td>
</tr>
</tbody>
</table>

The bgpdampid parameter specifies, for a set clause, that matched routes should have their BGP damping ID set to the given value. This value corresponds to the parameter set that will be used to maintain that route’s FoM upon it exhibiting instability. The corresponding parameter set need not be defined; if a route enters the BGP route flap damping suppression engine with a BGP damping ID that does not correspond to a valid parameter set, the default parameter set is applied.
disable bgp debug

Syntax
DISABLE BGP DEBUG[={MSG|DAMPING|STATE|UPDATE|ALL}[, ...]]
[PEER=ipadd]

enable bgp debug

Syntax
ENABLE BGP DEBUG[={MSG|DAMPING|STATE|UPDATE|ALL}[, ...]]
[PEER=ipadd]

Damping debugging shows messages to reflect BGP route flap damping state changes and events. Events include routes getting penalised for flapping, route state transition to suppressed and to reuse, routes becoming reachable and routes becoming unreachable.

set ip routemap

Syntax
SET IP ROUTEMAP=routemap ENTRY=1..4294967295
[ACTION={INCLUDE|EXCLUDE}] SET BGPDAMPID=0..100
[other-options]

The bgpdampid parameter specifies, for a set clause, that matched routes should have their BGP damping ID set to the given value. This value corresponds to the parameter set that will be used to maintain that route's FoM upon it exhibiting instability. The corresponding parameter set need not be defined; if a route enters the BGP route flap damping suppression engine with a BGP damping ID that does not correspond to a valid parameter set, the default parameter set is applied.

show bgp

Syntax
SHOW BGP

Description
This command displays information about BGP global configuration and operation, and has been modified to include whether BGP damping is enabled.
Figure 3: Example output from the `show bgp` command

```
BGP router ID .................. 192.168.1.1
Local autonomous system ........ 123
Confederation ID ............... 1234
Local preference .............. 100 (default)
Multi Exit Discriminator ...... -
Route table route map .......... -

Number of peers
  Defined ..................... 4
  Established ................. 2

BGP route table
  Iteration ................... 231
  Number of routes ............ 12654
  Route table memory ........  431872

BGP route flap damping ........ Enabled
```

**show bgp route**

**Syntax**

```
SHOW BGP ROUTE[*prefix*] [REGEXP=*aspathegexp*]
  [COMMUNITY={INTERNET|NOEXPORT|NOEXPORTSUBCONFED|
  NOADVERTISE|1..4294967295}]
```

Output of this command has been modified to display a “D” in front of any route that is currently suppressed by route flap damping.

Figure 4: Example output from the `show bgp route` command

```
BGP route table

<table>
<thead>
<tr>
<th>Prefix Path</th>
<th>Next hop</th>
<th>Origin</th>
<th>MED</th>
<th>Local pref</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 10.0.0.0/8</td>
<td>0.0.0.0</td>
<td>INCOMPLETE</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>EMPTY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D12.0.16.0/16</td>
<td>0.0.0.0</td>
<td>INCOMPLETE</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>12.12.0.0/16</td>
<td>11.0.0.1</td>
<td>INCOMPLETE</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>SEQ 1023 1024 1025;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt; 192.168.1.0/24</td>
<td>10.89.0.1</td>
<td>IGP</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>EMPTY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
New Community Format for BGP

This enhancement enables the administration of BGP community numbers in the correct format:

AA:XX

where

- AA is the AS number, in the range 0 to 65534
- XX is a value specified by the ASN administrator, in the range 0 to 65534

The old format is also still valid, and is of the form YYYYY, calculated using the formula:

\[ \text{AS number} \times 65536 + \text{community value} \]

You can use the new format in the:

- **include** and **exclude** parameters of the command **add ip communitylist**
- **community** parameter of the commands **add ip routemap** and **set ip routemap**
- **community** parameter of the command **show bgp route**

By default, community numbers are displayed using the new format. To display the original format instead, specify the optional parameter **oldcommunityformat** in the commands:

- `show ip routemap oldcommunityformat`
- `show ip communitylist oldcommunityformat`
Soft Resetting of BGP Peers

Prior to this enhancement, to make local policy changes that affected BGP peers, you had to disable the affected BGP peers, make the local policy change and re-enable the BGP peers. This enhancement removes the need for disabling and re-enabling the peers. The enhancement supports:

- RFC2918 - Route Refresh Capability for BGP-4.
- RFC2842 - Capabilities Advertisement with BGP-4.

You can configure the switch to make updates automatically, using the command:

```
enable bgp autosofupdate
```

This is disabled by default.

Alternatively, you can trigger soft reset of BGP peers, using the command:

```
reset bgp peer soft=[in|out|all]
```

There are two types of updates, inbound updates and outbound updates. Inbound updates are triggered by sending a Route Refresh message to the affected BGP peer. The Route Refresh message triggers the BGP peer to resend a BGP Update message. Outbound updates are carried out by sending a BGP Update message to the affected BGP peers. At any time, a local BGP policy change can trigger an inbound or outbound update, or both. The soft parameter determines where an inbound update or outbound update or both should be performed on the specified BGP peers.

The `show bgp` and `show bgp peer` commands have been modified to display these new parameters. Changes have been highlighted in the output below.

Figure 5: Example output from the command `show bgp`

```
BGP router ID ................. 10.42.0.237
Local autonomous system ....... 2
Confederation ID .............. 0
Local preference .............. 100 (default)
Multi exit descriminator ...... -
Route table route map ........ -
Auto soft reconfiguration ..... Disabled

Number of peers
  Defined ..................... 2
  Established ................. 1

BGP route table
  Iteration ................... 2
  Number of routes ............ 2
  Route table memory .......... 430
```
Figure 6: Example output from the command `show bgp peer=10.42.0.166`

```
Peer ................. 10.42.0.166
Description .......... -
State ................. Active
Remote AS ............. 1
BGP Identifier ...... 10.42.0.237
Connect retry ...... 120s
Hold time .......... 90s
Keep alive .......... 30s
Min AS originated ... 15
Min route advert .... 30

Filtering
  In filter .......... -
  In path filter ... -
  In route map .... -
  Out filter ........ -
  Out path filter ... -
  Out route map ..... -

Max prefix ............ OFF
External hops ....... 1 (EBGP multihop disabled)
Next hop self ....... No
Send community ...... No
Messages In/Out ..... 0/0
Debugging .......... -
  Device .......... -

Capabilities .......... Route Refresh

Established transitions ........ 0

Message counters:
  inOpen ...................... 0  outOpen ...................... 0
  inKeepAlive .................. 0  outKeepAlive .................. 0
  inUpdate ..................... 0  outUpdate ..................... 0
  inNotification .............. 0  outNotification .............. 0
  inRouteRefresh .............. 0  outRouteRefresh .............. 0
```
BGP Peer Policy Templates

This enhancement makes it easier to create BGP peers when many peers have identical inbound and outbound filtering policies. It enables you to define a template set of inbound and outbound filtering policy settings, which you can subsequently apply to many different peers.

A BGP peer can be assigned with a policy template upon or after creation of the peer. Once a peer has been assigned a template, the inbound and outbound policy parameters of that individual peer cannot be modified; if a parameter is to be changed, the route's template must be changed, thereby affecting every peer currently using that template. If it becomes necessary to change the policy of an individual peer that has been assigned a template, that template must be removed from the peer, at which point the peer is given its own individual copy of the filtering policy parameters maintained by the template.

New Commands

Use the following new commands to create or delete BGP peer policy templates.

add bgp peertemplate

Syntax

ADD BGP PEERTEMPLATE=id [CONNECTRETRY=(DEFAULT | 0..4294967295)] [DESCRIPTION=(description)] [HOLDTIME=(DEFAULT | 0|3..65535)] [INFILTER=(NONE | 300..399)] [INPATHFILTER=(NONE | 1..99)] [INROUTEMAP=(routemap)] [KEEPALIVE=(DEFAULT | 1..21845)] [MAXPREFIX=(OFF | 1..4294967295)] [MAXPREFIXACTION=(WARNING | TERMINATE)] [MINASORIGINATED=(DEFAULT | 0..3600)] [MINROUTEADEVERT=(DEFAULT | 0..3600)] [NEXTHOPSELF=(NO | YES)] [OUTFILTER=(NONE | 300..399)] [OUTPATHFILTER=(NONE | 1..99)] [OUTROUTEMAP=(routemap)] [PRIVATEASFILTER=(YES | NO)] [SENDCOMMUNITY=(NO | YES)]

where:

- **id** is a previously unused identification number for the peer policy template.
- **description** is a character string, 1 to 63 characters in length. Valid characters are any printable characters except '?' and '"'. If description contains spaces, it must be enclosed in double quotes.
- **routemap** is a character string, 0 to 15 characters in length. Valid characters are uppercase letters (A-Z), lowercase letters (a-z), digits (0-9) and the underscore character ('_').

Description

This command adds a BGP peer policy template to the switch. A template contains inbound and outbound policy filtering settings. These can be applied to one or more peers using the `add bgp peer` and `set bgp peer` commands.

For parameter descriptions, see the `add bgp peer` command in the Software Reference.
Examples
To create a new peer policy template with a hold-time of 30 seconds, and assign it to a peer, use the commands:

```
add bgp peertemplate=1 holdtime=30
add bgp peer=192.168.1.0/24 policytemplate=1
```

delete bgp peertemplate

Syntax
```
DELETE BGP PEERTEMPLATE=id
```

where:
- **id** is a previously unused identification number for the peer policy template.

Description
This command deletes an existing BGP peer policy template from the switch. All peers that have been assigned the specified peer template receive their own copies of the current peer template settings, and these peers can subsequently be modified individually.

set bgp peertemplate

Syntax
```
SET BGP PEERTEMPLATE=id [CONNECTRETRY={DEFAULT|0..4294967295}] [DESCRIPTION={description}] [HOLDTIME={DEFAULT|0|3..65535}] [INFILTER={NONE|300..399}] [INPATHFILTER={NONE|1..99}] [INROUTEMAP={routemap}] [KEEPALIVE={DEFAULT|1..21845}] [MAXPREFIX={OFF|1..4294967295}] [MAXPREFIXACTION={WARNING|TERMINATE}] [MINASORIGINATED={DEFAULT|0..3600}] [MINROUTEADVERT={DEFAULT|0..3600}] [NEXTHOPSELF={NO|YES}] [OUTFILTER={NONE|300..399}] [OUTPATHFILTER={NONE|1..99}] [OUTROUTEMAP={routemap}] [PRIVATEASFILTER={YES|NO}] [SENDCOMMUNITY={NO|YES}]
```

where:
- **id** is a previously unused identification number for the peer policy template.
- **description** is a character string, 1 to 63 characters in length. Valid characters are any printable characters except '?' and '"'. If description contains spaces, it must be enclosed in double quotes.
- **routemap** is a character string, 0 to 15 characters in length. Valid characters are uppercase letters (A-Z), lowercase letters (a-z), digits (0-9) and the underscore character ("_").

Description
This command modifies an existing BGP peer policy template.

For parameter descriptions, see the `add bgp peer` command in the Software Reference.
**show bgp peertemplate**

**Syntax**

SHOW BGP PEERTEMPLATE[=id]

**Description**

This command displays summary information about all BGP peer policy templates, or about the specified BGP peer policy template.

The *peertemplate* parameter specifies the identification number of the BGP peer policy template about which information is to be displayed. If a value is not specified, information is displayed for all BGP peers.

<table>
<thead>
<tr>
<th>BGP Peer Template Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template............. 1</td>
</tr>
<tr>
<td>Description .......... -</td>
</tr>
<tr>
<td>Connect retry ........ 120s</td>
</tr>
<tr>
<td>Hold time ........... 90s</td>
</tr>
<tr>
<td>Keep alive ........... 30s</td>
</tr>
<tr>
<td>Min AS originated .... 15</td>
</tr>
<tr>
<td>Min route advert ...... 30</td>
</tr>
</tbody>
</table>

**Filtering**

| In filter .......... - |
| In path filter ..... - |
| In route map ....... - |
| Out filter .......... - |
| Out path filter ..... - |
| Out route map ....... - |

| Max prefix ........... OFF |
| Next hop self ........ No |
| Send community ........ No |

Private AS Filter ... No
Modified Commands

Use the highlighted new parameters in existing commands to add or remove a template from BGP peer.

add bgp peer

**Syntax**

`ADD BGP PEER=ipadd POLICYTEMPLATE=n [other-options]

where:

■ n is the identification number of the policy template to apply to this peer.

**Description**
The `policytemplate` parameter identifies the peer policy template to be applied to this peer. The specified policy template must have been previously defined using the `add bgp peertemplate` command. If the `policytemplate` parameter is specified, only the parameters `description`, `ehops` and `remoteas` may accompany it; the specified policy template provides all other configuration values.

set bgp peer

**Syntax**

`SET BGP PEER=ipadd POLICYTEMPLATE[=n] [other-options]

where:

■ n is the identification number of the policy template to apply to this peer.

**Description**
The `policytemplate` parameter specifies the pre-defined policy template to be applied to this peer. If no value is given, any previously assigned template is removed, although the peer will still retain the settings from that template until they are explicitly changed. The specified policy template must have been previously defined using the `add bgp peertemplate` command. If the `policytemplate` parameter is specified, only the parameters `description`, `ehops` and `remoteas` may accompany it; the specified policy template provides all other configuration values. The policy configuration values from the given template replace any previously held configuration values.

Assigning a policy template to a peer overrides all existing inbound and outbound policy settings.
show bgp peer

Syntax  SHOW BGP PEER=ipadd

Figure 7: Example summary output from the show bgp peer command

<table>
<thead>
<tr>
<th>Peer</th>
<th>State</th>
<th>AS</th>
<th>InMsg</th>
<th>OutMsg</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.254</td>
<td>Estab</td>
<td>12345</td>
<td>23456</td>
<td>3245</td>
<td>-</td>
</tr>
<tr>
<td>192.168.3.16</td>
<td>Idle(D)</td>
<td>123</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 8: Example detailed output from the show bgp peer command

Peer ............... 192.168.2.254
Description ........ Sprint's AS 12345
State .............. Established
Policy Template ..... 4
   Description ........ Test Template 1
Remote AS ........... 12345
Connect Retry ...... 200s
Hold time .......... 90s (actual 0s)
Keep alive .......... 30s (actual 0s - no KEEPALIVES)
Min AS originated ... 20s
Min route advert .... 40s

Filtering
   In filter ........ -
   In path filter .... -
   In route map ...... -
   Out filter ........ 334
   Out path filter ... -
   Out route map ..... -

Max prefix .......... 2000 (action is WARNING)
External hops ...... 5 (EBGP multihop enabled)
Next hop self ...... No
Send community ...... No
Messages In/Out ..... 23456/3245
Debugging .......... -
   Device ............ -
IP Prefix Lists and Extensions to Route Map Attributes

This enhancement provides new match and set clauses for IP route maps, including prefix lists.

New Commands

Use the following new commands to create a prefix list.

**add ip prefixlist**

**Syntax**

ADD IP PREFIXLIST=name ENTRY=1..65535 [ACTION=MATCH|NOMATCH] PREFIX=prefix MASKLENGTH=range

where:

- **name** is a character string, 1 to 15 characters in length. Valid characters are uppercase letters (A-Z), lowercase letters (a-z), digits (0-9) and the underscore character (“_”).
- **prefix** is an IP address in dotted decimal notation.
- **range** is a pair of CIDR masks (in the form of decimal numbers from 0 to 32), separated by a “-” character. It may also be specified as a single CIDR mask.

**Description**

This command adds a new entry to the given prefix list on the switch. If a prefix list with the given name has previously been created, the new entry is added to that prefix list. Otherwise, a new prefix list is created and the given entry added to it.

The maximum number of prefix lists that can be added is 400. The maximum number of entries that can be added to each prefix list is 1000.

The **entry** parameter specifies the position of the new entry in the prefix list. Entry numbers can take on any value, and are used to sort the entries in ascending order. By leaving gaps between entry numbers, it is possible to add future entries in between existing entries. Note that if an error is made in the entry number, (e.g. entering entry=11 instead of entry=1), a new entry is added to the prefix list.

Each prefix list has an implicit final entry that matches all addresses, with an action of **nomatch**.

The **action** parameter specifies, for a given prefix list entry, the action to take when an IP route matches against that prefix list entry. If match is specified, the route will be included in the activity that is using the prefix list. If nomatch is specified, the route will not be included in the activity that is using the prefix list. In either case, if a match is not made, the next prefix list entry will be examined. The default action is match.
The **prefix** parameter specifies an IP address and CIDR mask length range for the given list entry. If the **prefix** parameter is not specified, the entry will match all routes with a suitable prefix mask length.

The **masklength** parameter specifies a range of acceptable prefix mask lengths. The mask length range can be specified in three different ways:

- As a mask length range (**masklength**=a-b).
  
  For any route to be considered as a potential match against this entry, its prefix mask length must be between \(a\) and \(b\) inclusive. \(a\) must be less than \(b\).

- As a single mask length (**masklength**=a).
  
  For any route to be considered as a potential match against this entry, its prefix mask length must be exactly \(a\).

- As an implicit mask length (**prefix**=192.168.0.0).
  
  For any route to be considered as a potential match against this entry, its prefix mask length must be exactly that of the mask length that corresponds to the class of the given address; in this case, 24.

**Examples**

To match only routes from the 192.168.0.0/16 network:

```
add ip prefixlist=sample entry=1 action=match prefix=192.168.0.0 masklength=16
```

To exclude the default route:

```
add ip prefixlist=sample entry=1 action=nomatch masklength=0
```

To include all routes:

```
add ip prefixlist=sample entry=1 action=match masklength=0-32
```

---

## set ip prefixlist

**Syntax**

```
SET IP PREFIXLIST=\[name\] ENTRY=1..65535 [ACTION=MATCH|NOMATCH] PREFIX=prefix MASKLENGTH=range
```

**Description**

This command modifies an existing entry in the given prefix list on the switch.

The **entry** parameter specifies the position of the new entry in the prefix list. Entry numbers can take on any value, and are used to sort the entries in ascending order. By leaving gaps between entry numbers, it is possible to add future entries in between existing entries. Note that if an error is made in the **entry** number, (e.g. entering **entry=11** instead of **entry=1**), a new entry is added to the prefix list.
Each prefix list has an implicit final entry that matches all addresses, with an action of nomatch.

The action parameter specifies, for a given prefix list entry, the action to take when an IP route matches against that prefix list entry. If match is specified, the route will be included in the activity that is using the prefix list. If nomatch is specified, the route will not be included in the activity that is using the prefix list. In either case, if a match is not made, the next prefix list entry will be examined. The default action is match.

The prefix parameter specifies an IP address and CIDR mask length range for the given list entry. If the PREFIX parameter is not specified, the entry will match all routes with a suitable prefix mask length.

The masklength parameter specifies a range of acceptable prefix mask lengths. The mask length range can be specified in three different ways:

- As a mask length range (masklength=a-b).
  For any route to be considered as a potential match against this entry, its prefix mask length must be between $a$ and $b$ inclusive. $a$ must be less than $b$.

- As a single mask length (masklength=a).
  For any route to be considered as a potential match against this entry, its prefix mask length must be exactly $a$.

- As an implicit mask length (prefix=192.168.0.0).
  For any route to be considered as a potential match against this entry, its prefix mask length must be exactly that of the mask length that corresponds to the class of the given address; in this case, 24.

Examples
To match only routes from the 192.168.0.0/16 network:

```
set ip prefixlist=sample entry=1 action=match
prefix=192.168.0.0 masklength=16
```

To exclude the default route:

```
set ip prefixlist=sample entry=1 action=nomatch masklength=0
```

To include all routes:

```
set ip prefixlist=sample entry=1 action=match masklength=0-32
```
show ip prefixlist

**Syntax**

SHOW IP PREFIXLIST[=name]

where:

- **name** is a character string, 1 to 15 characters in length. Valid characters are uppercase letters (A-Z), lowercase letters (a-z), digits (0-9) and the underscore character (“_”).

**Description**

This command displays information about any prefix lists that exist on the switch. If a prefix list name is specified as a value on the `prefixlist` parameter, detailed information about that prefix list and its entries is displayed. Otherwise, summary information about all existing prefix lists is displayed.

Figure 9: Example summary output from the *show ip prefixlist* command

<table>
<thead>
<tr>
<th>IP Prefix Lists</th>
<th>Name</th>
<th>Entries</th>
<th>In Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>11</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td>3</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10: Example detailed output from the *show ip prefixlist* command

<table>
<thead>
<tr>
<th>IP Prefix List</th>
<th>Name ................. Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Use ........</td>
<td>Yes</td>
</tr>
<tr>
<td>Entries:</td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>Action</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>Match</td>
</tr>
<tr>
<td>3</td>
<td>No Match</td>
</tr>
<tr>
<td>10</td>
<td>No Match</td>
</tr>
</tbody>
</table>
Modified Commands

Use the highlighted new parameters in existing commands to apply the new match and set options.

**add ip routemap**

**Syntax**

```
ADD IP ROUTEMAP=routemap ENTRY=1..4294967295
    [ACTION={INCLUDE|EXCLUDE}] MATCH MED=0..4294967295

ADD IP ROUTEMAP=routemap ENTRY=1..4294967295
    [ACTION={INCLUDE|EXCLUDE}] MATCH NEXTHOP=ipaddr

ADD IP ROUTEMAP=routemap ENTRY=1..4294967295
    [ACTION={INCLUDE|EXCLUDE}] MATCH ORIGIN={EGP|IGP|INCOMPLETE}

ADD IP ROUTEMAP=routemap ENTRY=1..4294967295
    [ACTION={INCLUDE|EXCLUDE}] MATCH PREFIXLIST=name

ADD IP ROUTEMAP=routemap ENTRY=1..4294967295
    [ACTION={INCLUDE|EXCLUDE}] SET MED={0..4294967295|REMOVE}
```

**Description**

The `med` parameter specifies, for a `match` clause, a Multi-Exit Discriminator value to match against. This value will only be present on routes that are being received from or sent to an external BGP peer.

The `nexthop` parameter specifies, for a `match` clause, a next hop IP address to match against.

The `origin` parameter specifies, for a `match` clause, a BGP origin type to match against.

The `prefixlist` parameter specifies, for a `match` clause, a list of one or more IP network prefixes to match against.

The `med` parameter specifies, for a set clause, that the set clause is of type `med`, and also the value to place in the Multi-Exit Discriminator attribute for the matched route. If a value of `remove` is specified, the MED attribute is removed from the matched update.
set ip routemap

**Syntax**

```
SET IP ROUTEMAP=routemap ENTRY=1..4294967295 [ACTION={INCLUDE|EXCLUDE}] MATCH MED=0..4294967295
```

```
SET IP ROUTEMAP=routemap ENTRY=1..4294967295 [ACTION={INCLUDE|EXCLUDE}] MATCH NEXTOP=ipaddr
```

```
SET IP ROUTEMAP=routemap ENTRY=1..4294967295 [ACTION={INCLUDE|EXCLUDE}] MATCH ORIGIN={EGP|IGP|INCOMPLETE}
```

```
SET IP ROUTEMAP=routemap ENTRY=1..4294967295 [ACTION={INCLUDE|EXCLUDE}] MATCH PREFIXLIST=name
```

```
SET IP ROUTEMAP=routemap ENTRY=1..4294967295 [ACTION={INCLUDE|EXCLUDE}] SET MED={0..4294967295|REMOVE}
```

**Description**

The `med` parameter specifies, for a `match` clause, a Multi-Exit Discriminator value to match against. This value will only be present on routes that are being received from or sent to an external BGP peer.

The `nexthop` parameter specifies, for a `match` clause, a next hop IP address to match against.

The `origin` parameter specifies, for a `match` clause, a BGP origin type to match against.

The `prefixlist` parameter specifies, for a `match` clause, a list of one or more IP network prefixes to match against.

The `med` parameter specifies, for a set clause, that the set clause is of type `med`, and also the value to place in the Multi-Exit Discriminator attribute for the matched route. If a value of `remove` is specified, the MED attribute is removed from the matched update.
BGP Route Reflection

This enhancement adds support for BGP route reflection, defined in RFC2439.

If a BGP peer learns a route from an eBGP peer, and selects it as the best available route to the given destination network, an advertisement is flooded to all iBGP and eBGP peers. If a peer learns a route from an iBGP peer, no advertisements are sent to its other iBGP peers. This policy requires that all BGP speakers within an AS be fully meshed. As a result, the scalability of a BGP AS is order $n^2$ ($n$ speakers require $n(n-1)/2$ peer sessions).

BGP Route Reflection defines a mechanism for improving the scalability of the BGP AS, whereby specific iBGP peers, known as Route Reflectors (RR) will be given the authority to advertise iBGP-learned routes to a predefined subset of its iBGP peers, known as Client Peers (CP), thus negating the need for full-meshing. This problem can also be solved using BGP Confederations. However, Route Reflection has the advantage that only the BGP hosts that perform the reflection need to understand Route Reflection for it to be deployed, whereas all host in a Confederation are required to be Confederation-aware.

Although suffering from poor scalability, a full-mesh AS has the advantage of being extremely robust. In contrast, an AS that employs a single RR to serve a large number of clients is relatively vulnerable to congestion or loss of connectivity due to the critical role that the route reflector plays in the operation of the AS. As such, it is desirable to support multiple RRs within an AS; if a RR has a peering relationship with another RR in the same AS, they should be configured as NCPs of each other. In order to support multiple RRs and to detect and prevent routing information loops, the following optional BGP attributes are defined, and are supported for BGP Route Reflection.

- **ORIGINATOR_ID**: This 4-byte attribute has a type code of 9. When a RR receives a new route from an IBGP peer, the originator ID attribute is added if one was not already present. The value used is the router ID of the local AS peer who received the given route from an EBGP peer.

- **CLUSTER_LIST**: This variable-length attribute has a type code of 10. It is a list of 4-byte CLUSTER_ID values that together represent the reflection path of the given route through the local AS.

The cluster ID value is user-configurable, and will be used only on peers that are configured to perform route reflection. By default, the cluster ID value will be the peer’s router ID. If multiple route reflectors exist within a single cluster, each must be manually configured to use the same cluster ID.

**Modified Commands**

To configure route reflection, use the highlighted new parameters in the following existing commands. Several `show` commands have also been modified to display route reflection information:

- `show bgp`  
- `show bgp count`  
- `show bgp peer`  
- `show bgp route`
**add bgp peer**

**Syntax**

`ADD BGP PEER=ipadd [CLIENT=YES|NO] [other-options]`

The `client` parameter specifies whether the given peer is a client of the local device. The peer can be configured as a client only if it is an IBGP peer. If the `client` parameter is not specified, the peer will become a non-client.

**set bgp peer**

**Syntax**

`SET BGP PEER=ipadd [CLIENT=YES|NO] [other-options]`

The `client` parameter specifies whether the given peer is a client of the local device. The peer can be configured as a client only if it is an IBGP peer. If the `client` parameter is not specified, the peer will become a non-client.

**set bgp**

**Syntax**

`SET BGP [CLUSTERID=ipaddr] [other-options]`

The `clusterid` parameter specifies the identifying value of the cluster for which the local peer is a route reflector, and as such is used only if BGP route reflection is being performed by the local device. By default, the cluster ID is the first found IP address on the local device, which is sufficient when the local device is the only route reflector in its cluster. However, if the cluster contains multiple route reflectors, all of the route reflectors must be configured to employ the same cluster ID, which should be the IP address of one of the route reflectors. Failure to do so may result in routing information loops.
BGP Fast Fallover

By default, when the interface that supports an eBGP peer session goes down, the corresponding peer session is not reset until that session’s keep-alive or hold timer expires. With fast fallover enabled for that peer, the session is reset immediately upon the interface going down, so that there is fast fallover in case of link failures; the switch can withdraw paths as soon as the link goes down, rather than waiting for up to three minutes to propagate the change. As a result, the rate of convergence to the new network topology is greatly improved.

It should be noted that certain types of links may be particularly susceptible to unreliability. As such, enabling fast fallover will cause the BGP sessions that are supported by such links to flap, resulting in instability and excessive update messaging within the network. As such, it is preferable to allow specific sessions to survive short line-flaps, while other sessions will be immediately reset in order to allow the rapid switch-over to an alternative path.

BGP peer fast fall-over will consider the link to have gone down when it receives notification from the relevant layer 2 interface that its link status has changed from up to down. As such, it is important that when applied to peers supported by a VLAN, that VLAN should contain only ports that are used to connect to that peer. Failure to observe this restriction may result in the occurrence of false negatives, whereby the layer 1 links that are used to reach the given peer have failed but unrelated ports in the VLAN are still up, and so the status of the VLAN does not change.

Modified Commands

To configure fast fallover, use the highlighted new parameters in the following existing commands. The `show bgp` command has also been modified to display whether fast fallover is enabled.

**add bgp peer**

**Syntax**

`ADD BGP PEER=ipadd [FASTFALLOVER={YES|NO}] [other-options]`

The `fastfallover` parameter specifies how the switch should behave when the link state of the interface that supports the given peer session goes down. If fast fallover is to be employed, the peer session will be reset immediately upon the link state of its interface going down. Otherwise, the session will be reset only once its keep-alive timer has expired. If the `fastfallover` parameter is not specified, fast fallover will not be employed.

**set bgp peer**

**Syntax**

`SET BGP PEER=ipadd [FASTFALLOVER={YES|NO}] [other-options]`

The `fastfallover` parameter specifies how the switch should behave when the link state of the interface that supports the given peer session goes down. If fast fallover is to be employed, the peer session will be reset immediately upon the
link state of its interface going down. Otherwise, the session will be reset only once its keep-alive timer has expired. If the fastfallover parameter is not specified, fast fallover will not be employed.

**Tag Distinction of Static Routes for BGP Import**

This enhancement enables you to identify static routes by tagging them with a number. You can then match against this number in a route map and only import the appropriately-tagged routes into BGP.

To do this, first use the new tag parameter to tag the route in the command:

```
ADD IP Route=ipadd INTERFACE=interface NEXTHop=ipadd
[CIRCuit=miox-circuit] [DLCi=dlci] [MASK=ipadd]
[METric=1..16] [METRIC1=1..16] [METRIC2=1..65535]
[POLIcy=0..7] [PREFerence=0..65535] [TAG=1..65535]
```

Then create or modify the route map and specify the tag value, using the new tag parameter in one of the commands:

```
ADD IP ROUTEMap=routemap ENTRY=1..4294967295
[ACtion={INCLude|EXCLude}] MAtch [TAG=1..65535]
```

```
SET IP ROUTEMap=routemap ENTRY=1..4294967295
[ACtion={INCLude|EXCLude}] MAtch [TAG=1..65535]
```

Route maps which match on the tag parameter will only take effect when importing static routes from IP to BGP, using the **add bgp network** and **add bgp import** commands. The new match will not filter routes to be sent to BGP peers and will not affect received update messages from BGP peers.

A routemap entry can only match on one of the parameters *tag, aspath, prefixlist, med, nexthop, origin* or *community*, not a combination of these parameters.

The tag parameter is now displayed in the output of the command **show ip route**:

```
IP Routes

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask Type</th>
<th>Policy</th>
<th>NextHop Protocol</th>
<th>Tag</th>
<th>Interface Metrics</th>
<th>Age Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0</td>
<td>255.0.0.0</td>
<td>0.0.0.0</td>
<td>vlan1</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>11.0.1.0</td>
<td>255.255.255.0</td>
<td>10.42.0.22</td>
<td>vlan1</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>11.0.2.0</td>
<td>255.255.255.0</td>
<td>10.42.0.22</td>
<td>vlan1</td>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>
```

**Software Release 2.6.6**

C613-10418-00 REV A
BGP Private AS Numbers

AS numbers are two bytes in length, and as such have a range from 1-65535. Of this value range, the AS numbers 1-64511 are globally unique and are assigned by InterNIC. The remaining value range from 64512-65535 is reserved for AS numbers that are private; these numbers are unique only within the scope of a given administrative domain.

Because private AS numbers are not globally unique, they should not be leaked to global BGP routing tables. To prevent private AS number from crossing administrative boundaries, the stripping of private AS numbers from the AS Path attributes of outgoing update messages is supported. This facility is be configurable on a per-peer or per-template basis; it is disabled by default for peer templates, and enabled by default on a peer if that peer's AS number is public.

If in incoming AS_PATH attribute contains both public and private AS numbers, a configuration error is present. To prevent a routing loop or other problems occurring as a result, the routes are not injected into the BGP route table.

Modified Commands

To configure stripping of private AS numbers, use the highlighted new parameters in the following existing commands. The `show bgp peer` and `show bgp peertemplate` commands have also been modified to display whether private AS numbers are filtered.

add bgp peer

**Syntax**

```plaintext
ADD BGP PEER=ipadd [PRIVATEASFILTER={YES|NO}] [other-options]
```

The `privateasfilter` parameter specifies whether private AS numbers (in the range 64512-65535) should be stripped from the AS PATH attribute on any update messages sent to external peers. If the `privateasfilter` parameter is not specified, private AS number filtering will be enabled if the specified remoteas is a public AS number.

set bgp peer

**Syntax**

```plaintext
SET BGP PEER=ipadd [PRIVATEASFILTER={YES|NO}] [other-options]
```

The `privateasfilter` parameter specifies whether private AS numbers (in the range 64512-65535) should be stripped from the AS PATH attribute on any update messages sent to external peers. If the `privateasfilter` parameter is not specified, private AS number filtering will be enabled if the specified remoteas is a public AS number.
add bgp peertemplate

Syntax
ADD BGP PEERTEMPLATE=id [PRIVATEASFILTER={YES|NO}] [other-options]

The privateasfilter parameter specifies whether private AS numbers (in the range 64512-65535) should be stripped from the AS PATH attribute on any update messages sent to external peers. By default, this feature is disabled.

set bgp peertemplate

Syntax
SET BGP PEERTEMPLATE=id [PRIVATEASFILTER={YES|NO}] [other-options]

The privateasfilter parameter specifies whether private AS numbers (in the range 64512-65535) should be stripped from the AS PATH attribute on any update messages sent to external peers. By default, this feature is disabled.
Importing BGP routes into OSPF

With this enhancement you can import routes from BGP into OSPF. OSPF will then redistribute these routes. This enhancement adds three parameters to the set ospf command, and modifies the output of the show ospf command. The new parameters are bgpimport, bgpfilter and bgplimit.

BGP can learn thousands of routes, so it is important to consider the network impact of importing these routes. Routing devices in the OSPF domain may become overloaded if they store too many routes. You can prevent this by limiting the number of routes that will be imported.

---

Do not enable the importing of BGP routes into OSPF unless you are sure about the consequences for the OSPF domain.

---

Enabling BGP route import

To enable importing BGP routes into OSPF, use the command:

```
set ospf bgpimport=on
```

Limiting the number of routes

There are two ways to limit the number of BGP routes imported into OSPF. One way is to specify a maximum number of routes with the command:

```
set ospf bgplimit=1...300
```

When the limit is reached, the importing of routes will stop until existing routes are removed. Because they are BGP routes, actions of BGP control when the routes disappear.

The other way to limit the imported routes is to configure a routing filter. This filter is used in conjunction with the bgpfilter parameter in the set ospf command to control the passing of routing information in and out of the device. To configure a filter, use the add ip filter command:

```
add ip filter=filter-number {action=include|exclude} source=ipadd [smask=ipadd] [entry=entry-number]
```

Use this filter to limit imported BGP routes with the command:

```
set ospf bgpfilter=300...399
```

where the filter number is the previously configured filter.

Take care when configuring the IP filter. If the number of imported routes reaches the bgplimit parameter, you may not have imported all the routes specified with the bgpfilter parameter.

Advertising desired routes

The order in which routes are added is arbitrary. This means that to have desired BGP routes advertised by OSPF, you must take care setting the entry number for the route filter with the add ip route command. Assign a low entry number to a filter used to import preferred BGP routes. Alternatively, set the
bgplimit parameter above the total number of routes that BGP will ever add to the routing table.

**Configuration example**

This example supposes that you want to import the route 192.168.72.0 into the OSPF routing domain, but no other routes. This route is received on the gateway router as a BGP route. The following steps show the sequence of commands to use in this scenario.

1. Set up the IP filter:
   
   ```
   add ip filter=300 source=192.168.72.0 smask=255.255.255.255 action=include
   ```

2. Set up OSPF BGP import parameters:
   
   ```
   set ospf bgpimport=on bgpfilter=300 bgplimit=1
   ```

3. Check that BGP has added the route to the IP route table:
   
   ```
   show ip route=192.168.72.0
   ```
   
   The route should be visible in the output of the command.

4. Check that OSPF has imported the route:
   
   ```
   show ospf lsa=192.168.72.0
   ```
   
   The output should show that there is an AS external LSA with this ID.
BGP Backoff

BGP backoff enables BGP to elegantly handle low system memory situations. Other software modules can cause a spike, or surge, of system memory utilisation for brief periods of time. During these times, it is better for BGP to back off and delay its processing until a time where system memory is more abundant.

The objective of the backoff utility is to be considerate to other processes that are running and consuming memory resources while not shutting BGP down until it is determined that BGP has backed off for a prolonged period of time.

Configuration

BGP system memory backoff is highly configurable. Configuration options are:

- percentage limit of total system memory utilisation that causes BGP to back off.
- time that BGP backs off for.
- total number of backoffs before all BGP peers are disabled.
- total consecutive number of backoffs before all BGP peers are disabled.

The backoff time is recalculated after a given number of backoffs. This is termed a step. The first backoff time is calculated as:

\[ \text{base time} \times \frac{\text{multiplier}}{100} \]

After each step the backoff time is recalculated, based on the current backoff time:

\[ \text{current backoff time} \times \frac{\text{multiplier}}{100} \]

The value is rounded down to the nearest second (unless it is less than 1 second, in which case it is set to 1 second).

For example, a base time of 60 seconds with a multiplier of 110 will increase the timeout by 10 percent every time the backoff time is recalculated. Thus, a step value of 2 and multiplier of 110 will result in the following figures.

<table>
<thead>
<tr>
<th>Backoff Iteration</th>
<th>Time to Backoff (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>60</td>
</tr>
<tr>
<td>1</td>
<td>66</td>
</tr>
<tr>
<td>2</td>
<td>72</td>
</tr>
<tr>
<td>3</td>
<td>79</td>
</tr>
<tr>
<td>4</td>
<td>79</td>
</tr>
</tbody>
</table>

A multiplier of less than 100 percent gives the effect of a decay mechanism, and a multiplier of greater than 100 percent gives the effect of an accumulative mechanism.
Consecutive Backoffs

A consecutive backoff is a backoff where BGP has backed off for a period of time already and is set to immediately backoff once again without performing any processing.

Where consecutive backoffs occur so that BGP processing is disabled for a prolonged period of time, and BGP may be deemed unrecoverable, a consecutive backoff limit can be set to disable all BGP peers.

The consecutive backoff count is reset to zero whenever BGP was previously backed off and performed BGP processing when the backoff time expired.

New Commands

To configure BGP backoff, use the following new commands.

**set bgp backoff**

| Syntax | SET BGP BACkoff[=0..100] [CONSecutive=1..20] [MULtiplier=0..1000] [BASEtime=0..100] [TOTallimit=0..1000] [STep=0..1000] |
| Description | This command configures BGP backoff. |

The *backoff* parameter specifies the percentage utilisation of system memory that causes BGP to back off. The default is 95.

The *consecutive* parameter specifies the number of consecutive backoffs that causes BGP to disable all peers. The default is 20.

The *multiplier* parameter specifies a multiplier for increasing or decreasing the backoff time at each backoff iteration. The default is 100. The change in backoff time at each step is calculated as:

\[
\text{current backoff time} \times \frac{\text{multiplier}}{100}
\]

The *basetime* parameter specifies the time value (in seconds) used to calculate the total backoff time for the first backoff iteration. The default is 10. The first backoff time is calculated as:

\[
\text{basetime} \times \frac{\text{multiplier}}{100}
\]

The *totallimit* parameter specifies the total number of backoffs that may occur until all peers are disabled. The default is 0.

The *step* parameter specifies the number of backoff iterations after which the backoff time is recalculated.
show bgp backoff

**Syntax**  
`show bgp backoff`

**Description**  
This command displays BGP backoff details.

**Figure 11:** Example output of the `show bgp backoff` command

<table>
<thead>
<tr>
<th>BGP Backoff Stats:</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>NORMAL</td>
</tr>
<tr>
<td>total hist backOffs</td>
<td>0</td>
</tr>
<tr>
<td>total backOffs</td>
<td>0</td>
</tr>
<tr>
<td>total backOff Limit</td>
<td>0</td>
</tr>
<tr>
<td>was backedOff</td>
<td>FALSE</td>
</tr>
<tr>
<td>consecutive backOffs</td>
<td>0</td>
</tr>
<tr>
<td>consecutive backOffs limit</td>
<td>5</td>
</tr>
<tr>
<td>base Timeout</td>
<td>60</td>
</tr>
<tr>
<td>Timeout multiplier</td>
<td>100%</td>
</tr>
<tr>
<td>Timeout step</td>
<td>1</td>
</tr>
<tr>
<td>Timeout length (sec)</td>
<td>60</td>
</tr>
<tr>
<td>Trigger on Mem use</td>
<td>95%</td>
</tr>
<tr>
<td>Current Mem use</td>
<td>7%</td>
</tr>
</tbody>
</table>

**Table 3:** Parameters in the output of the `show bgp backoff` command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>One of NORMAL, BACKING OFF, or DISABLING PEERS.</td>
</tr>
<tr>
<td>total hist backOffs</td>
<td>The cumulative count of BGP backoffs.</td>
</tr>
<tr>
<td>total backoff limit</td>
<td>The total number of backoffs that cause BGP to disconnect its peers.</td>
</tr>
<tr>
<td>was backedOff</td>
<td>TRUE if BGP was previously backed off during the last operation. This is used to detect consecutive backoffs.</td>
</tr>
<tr>
<td>consecutive backOffs</td>
<td>See Consecutive Backoffs on page 38.</td>
</tr>
<tr>
<td>consecutive backOffs limit</td>
<td>See Consecutive Backoffs on page 38.</td>
</tr>
<tr>
<td>base Timeout</td>
<td>See Configuration on page 37.</td>
</tr>
<tr>
<td>Timeout multiplier</td>
<td>See Configuration on page 37.</td>
</tr>
<tr>
<td>Timeout step</td>
<td>See Configuration on page 37.</td>
</tr>
<tr>
<td>Timeout length</td>
<td>See Configuration on page 37.</td>
</tr>
<tr>
<td>Trigger on Mem use</td>
<td>The trigger percentage value of system memory use that will cause BGP to back off.</td>
</tr>
<tr>
<td>Current Mem use</td>
<td>The amount of memory used by the system at the given moment in time when the command was executed.</td>
</tr>
</tbody>
</table>
BGP memory accounting

BGP memory accounting allows a limit to be set on BGP’s use of total system memory. Use of total system memory is expressed as a percentage value. Utilisation of memory beyond this limit activates the disabling of BGP. BGP peers are shut down, and therefore all routes learnt from those peers are dropped.

New Commands

To configure BGP memory accounting, use the following new commands.

set bgp memlimit

Syntax

SET BGP MEMlimit[=0..100]

Description

This command limits the percentage of system memory available to BGP.

The memlimit parameter specifies the memory percentage utilisation trigger for the BGP module that will shut down BGP peers. The default is 65.

show bgp memlimit

Syntax

SHOW BGP MEMlimit

Description

This command displays the current memory limit setting and the current memory use by BGP.

Figure 12: Example output of the show bgp memlimit command

| BGP Memory Limit: 65%, Actual Use: 0% |

show bgp memlimit scan

Syntax

SHOW BGP MEMlimit SCAN

Description

This command displays the freelists (as a hexadecimal address) that are registered to a given module, the size of their respective structure, the number of freelist buffers that are currently used in memory, and the number of memory buffers that these freelists buffers are utilising. This output is useful to highlight, at a given moment in time, a detailed state of BGP memory use.
Figure 13: Example output of the `show bgp memlimit` command

```
BGP Memory Limit: 65%, Actual Use: 0%
Module Freelist Stats: moduleId = 5
  module buffer use: 3
  module percent use: 0%
  list  unitSize  freeUsed  buffersUsed
  -------------------------------------------
  008418d0     80       0      0
  0084fae0     12       0      0
  0084f104     88       0      0
  00841784     68       0      0
  00841900     48       2      0
  0084fb70     12       0      0
  00855398     32       3      0
  0085b8c4    228      2      0

Module Freelist Stats: moduleId = 103
  module buffer use: 4
  module percent use: 0%
  list  unitSize  freeUsed  buffersUsed
  -------------------------------------------
  0081306c     24       0      0
  0081300c     12       0      0
  008131c0     32       0      0
  0081361c     44       0      0
  00813220     8       0      0
  00812fdc   512      0      0
  00813124     8       0      0
  0081353c     8       0      0
  00813190     8       0      0
  00812654     40      0      0
  00813154  520      0      0
  008135c4  112      0      0
  00813350     32      0      0
  00812684  128      0      0
  00811e6c     24      0      0
  008120dc     56      0      0
  008130bc  268      0      0
  008126b4   20      0      0
  00813380     16      0      0
  0081356c     52      0      0
  008131f0     36      0      0
  008123a4     16      0      0
  0081364c     20      0      0
  00811f9c     40      0      0
  00812374 1060      1      0
  008136a4   40      0      0
  0081303c     16      0      0
  008418d0     80      0      0
  0084fae0     12      0      0
  0084f104     88      0      0
  00841784     68      0      0
  00841900     48      2      0
  0084fb70     12      0      0
  00855398     32      3      0
  0085b8c4    228      2      0
```
MD5 Authentication for BGP

This enhancement enables authentication of the source of BGP updates and avoids the disruptive effect of spoofed TCP packets. There are two steps to configuring it for a given peer:

1. Configuring a password.
2. Enabling MD5 authentication.

Once these two steps have been done a digest will be added to every packet sent over the TCP connection and will be authenticated at the other end. If any part of the digest cannot be verified, the packet is dropped with no response sent.

Modified Commands

To configure MD5 authentication, use the highlighted new parameters in the following existing commands. The show bgp peer command has also been modified to display whether MD5 authentication is used, and if so, what the password is.

add bgp peer

Syntax

```
ADD BGP PEER=ipadd [AUTHENTICATION={NONE|MD5}] [PASSWORD=password] [other-options]
```

where:

- **password** is a character string between 1 and 80 characters in length. Valid characters are any printable characters except '?' and '"'. If password contains spaces, it must be enclosed in double quotes.

The **authentication** parameter specifies whether or not to use MD5 authentication for the BGP peer. If you specify **md5**, you must also specify a password. The default is **none**.

The **password** parameter specifies the key to be used as part of the authentication algorithm. This key must be the same between two BGP peers or they will not be able to communicate with each other.

set bgp peer

Syntax

```
SET BGP PEER=ipadd [AUTHENTICATION={NONE|MD5}] [PASSWORD=password] [other-options]
```

where:

- **password** is a character string between 1 and 80 characters in length. Valid characters are any printable characters except '?' and '"'. If password contains spaces, it must be enclosed in double quotes.
The **authentication** parameter specifies whether or not to use MD5 authentication for the BGP peer. If you specify **md5**, you must also specify a password. The default is **none**.

The **password** parameter specifies the key to be used as part of the authentication algorithm. This key must be the same between two BGP peers or they will not be able to communicate with each other.
## Actions for QoS Traffic Classes and Flow Groups

This enhancement enables traffic that is classified into a flow group or traffic class to be:

- **discarded**
  
  This option means unwanted traffic can be dropped without being processed.

- **sent directly to a port on a VLAN**
  
  This option allows you to mirror packets classified on any port. It is much more flexible than standard mirroring, which only monitors traffic on a single port. You can choose to also forward the mirrored traffic normally, discard it, or send it to a particular port and VLAN.

- **forwarded normally.**

To act on classified traffic at the flowgroup level, use one of the commands:

```bash
create qos flowgroup-id-list
  action={none|forward|forward,sendmirror|discard|
  discard,sendmirror|sendmirror|sendmirror,sendvlanport|
  sendvlanport} [vlan=vlan-id] [port=port] [other-options]
```

```bash
set qos flowgroup-id-list
  action={none|forward|forward,sendmirror|discard|
  discard,sendmirror|sendmirror|sendmirror,sendvlanport|
  sendvlanport} [vlan=vlan-id] [port=port] [other-options]
```

where:

- **vlan-id** is an integer in the range 1 to 4094.
- **port** is any valid port number.

The `action` parameter specifies the action that is to be performed on any traffic belonging to the flow group. The default action is `forward`.

If you specify an `action` of `none`, the action will be overridden with the setting of the flow group’s traffic class.

If you specify an `action` of `forward`, the traffic will be forwarded normally.

If you specify an `action` of `discard`, the traffic will be dropped.

If you specify an `action` of `sendvlanport`, both `vlan` and `port` must also be specified. Traffic will be sent to the VLAN specified by the `vlan` parameter and the port specified by the `port` parameter. The VLAN must exist and the specified port must be a member of that VLAN. The switch determines whether the port is tagged or untagged for that VLAN, and sends the traffic with the correct tag if the port is tagged. If the port is untagged for the specified VLAN the frame is sent untagged.

If you specify an `action` of `sendmirror`, the traffic will be sent to the pre-configured mirror port.
To act on classified traffic at the trafficclass level, use one of the commands:

```
create qos trafficclass=id-list
    action={forward|forward,sendmirror|discard|
discard,sendmirror|sendmirror|sendmirror,sendvlanport|
sendvlanport} [vlan=vlan-id] [port=port] [other-options]

set qos trafficclass=id-list
    action={forward|forward,sendmirror|discard|
discard,sendmirror|sendmirror|sendmirror,sendvlanport|
sendvlanport} [vlan=vlan-id] [port=port] [other-options]
```

where:
- `vlan-id` is an integer in the range 1 to 4094.
- `port` is any valid port number.

The actions have the same effect as the flow group actions, except that `action=none` is not valid.

To act on unclassified traffic, which will be processed by the default traffic class, use one of the commands:

```
create qos policy=id-list
    dtcaction={forward|forward,sendmirror|discard|
discard,sendmirror|sendmirror|sendmirror,sendvlanport|
sendvlanport} [vlan=vlan] [port=port] [other-options]

set qos policy=id-list
    dtcaction={forward|forward,sendmirror|discard|
discard,sendmirror|sendmirror|sendmirror,sendvlanport|
sendvlanport} [vlan=vlan] [port=port] [other-options]
```

where:
- `vlan-id` is an integer in the range 1 to 4094.
- `port` is any valid port number.

The DTC actions have the same effect as the flow group actions, except that `dtcaction=none` is not valid.
L2 QoS Actions in Hardware Filters

This enhancement enables you to classify traffic and use a hardware filter to set its queue, 802.1p user priority or bandwidth class. This is a mechanism for elevating the probability of CPU reception for packets that you determine to be “important”.

In heavily congested networks, data streams can sometimes use up all the available bandwidth of the CPU receive process. This increases the probability that infrequently-sent packets are lost, for example routing protocol packets (BGP, OSPF, PIM, DVMRP) or STP packets. By creating an appropriate classifier and hardware filter, such packets can be given preferential treatment, in terms of CPU reception.

To configure the hardware filter, use the command:

```
ADD SWITCH HWFILTER=filter-id CLASSIFIER=rule-id
   ACTION=SETL2QOS [L2QOSQUEUE=0..7] [PRIORITY=0..7]
   [BANDWIDTHCLASS=1..3]
```

The `action` parameter specifies what action the classifier entry will take with packets that match the classifier. If you specify `setl2qos`, then packets matching the classifier will have their bandwidth class (or drop precedence), queue, and 802.1p user priority modified to the values specified by the `bandwidthclass`, `l2qosqueue`, and `priority` parameters. The action `setl2qos` cannot be specified at the same time as `drop`.

The `l2qosqueue` parameter specifies the queue to send packets to that match the classifier. The default is 0.

The `priority` parameter specifies the 802.1p user priority to remark packets with that match the classifier. The default is 0.

The `bandwidthclass` parameter specifies the bandwidth class (drop precedence) to assign packets to that match the classifier. The default is 1.
Classification By L4 Mask and Inner VLAN Settings

The following new optional parameters have been added to the create classifier and set classifier commands:

[L4SMASK=mask|ANY]
[L4DMASK=mask|ANY]
[TPID=tpid|ANY]
[INNERVLANPRIORITY=0..7|ANY]
[INNERTPID=tpid|ANY]
[INNERVLANID=VLAN=1..4094|ANY]
[VLANPRIORITY=0..7|ANY]

where

- `mask` is a 2-byte hexadecimal number.
- `tpid` is a 2-byte hexadecimal number.

The L4mask parameter specifies the mask or range of TCP/UDP source ports in the packet. The default is any.

The L4dmask parameter specifies the mask or range of TCP/UDP source ports in the packet. The default is any.

The vlanpriority parameter specifies the 802.1P priority in the VLAN tag. The default is any.

The tpid parameter specifies the Tag Protocol Identifier field in the packet. The default is any.

The innervlanpriority, inntpid and innervlanid parameters are for use on double tagged traffic. `inner` refers the second 802.1Q tag in the packet. This is the one being tunnelled. Table 4 shows where in the packet the inner and outer tags will be matched. Matching outer parameters on customer ports is not possible.

<table>
<thead>
<tr>
<th>Outer VLAN parameters (normal)</th>
<th>Inner VLAN parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer port</td>
<td>Only VLAN ID.</td>
</tr>
<tr>
<td>Core port</td>
<td>1st tag</td>
</tr>
<tr>
<td>Nested VLANs disabled</td>
<td>1st tag</td>
</tr>
</tbody>
</table>

When attaching the classifier to a hardware filter and nested VLANs are not being used, it is assumed that all incoming packets are double tagged. When nested VLANs are being used outer parameters also cannot be used to match on customer ports. If the classifier is being attached to a number of ports, they will all be treated like core ports if at least one port is a core port.

The innervlanpriority parameter specifies the second 802.1P field in the packet. The default is any.
The `innertpid` parameter specifies the TPID in the second 802.1Q tag in the packet. The default is `any`.

The `innervlanid` parameter specifies the tunnelled VLAN ID in the second 802.1Q tag in the packet. The default is `any`.

Extensions to Discarding of Source-Routed IP Packets

By default, all source-routed packets are discarded. New options have been added to the commands `enable ip srcroute` and `disable ip srcroute`, to allow finer control of which packets are forwarded or discarded. The modified syntax is:

**enable ip srcroute**

**Syntax**

`ENABLE IP SRCROUTE[={LOOSE|STRICT|ALL}]`

**Description**

This command enables the forwarding of source-routed IP packets. If a specific type of source-routed IP packet is specified, forwarding of that type will be enabled. Otherwise, forwarding of all source-routed IP packets will be enabled.

**disable ip srcroute**

**Syntax**

`DISABLE IP SRCROUTE[={LOOSE|STRICT|ALL}]`

**Description**

This command disables the forwarding of source-routed IP packets. If a specific type of source-routed IP packet is specified, forwarding of that type will be disabled. Otherwise, forwarding of all source-routed IP packets will be disabled.
QoS Counters

A new command, `set switch enhancedmode`, has been added to enable you to monitor QoS counters. You can then reset the counters using new `reset` commands, and display the counters using new and expanded `show` commands, as listed below.

### set switch enhancedmode

**Syntax**

```
SET SWITCH ENHANCEDMODE={QOSCOUNTERS|NONE}
```

**Description**

This command sets the enhanced mode of the switch to include monitoring of QoS counters for egress queues and traffic classes. The default is `none`.

**Examples**

To turn monitoring of QoS counters on use the command:

```
set switch enhancedmode=qoscounters
```

To turn monitoring of QoS counters off use the command:

```
set switch enhancedmode=none
```

### reset qos port

**Syntax**

```
RESET QOS PORT={port-list} COUNTERS TRAFFICCLASS[={trafficclass-list|DEFAULT|ALL}]
```

where:

- `port-list` is a single port of group of ports; a range of integers (specified as 0-4) or a comma separated list of policy numbers and/or ranges (0, 3, 4-9).

- `trafficclass-list` is an integer in the range of 0-1023; a range of integers (specified as 0-3) or a comma separated list of integers and/or ranges without spaces.

**Description**

This command resets the counters for the specified traffic classes attached to the specified port(s).

**Examples**

To clear the traffic class counters on the traffic classes attached to port 4, use the command:

```
reset qos port=4 counters trafficclass=all
```
**reset qos accelerator**

**Syntax**

```
RESET QOS ACCELERATOR COUNTERS
TRAFFICCLASS [= {trafficclass-list | DEFAULT | ALL}]
```

where:

- `trafficclass-list` is an integer in the range of 0-1023; a range of integers (specified as 0-3) or a comma separated list of integers and/or ranges without spaces.

**Description**

This command resets the counters for the traffic classes attached to the accelerator card.

**Examples**

To clear the traffic class counters on the traffic classes attached to port 4, use the command:

```
reset qos accelerator counters trafficclass=1-5
```
Figure 14: Example output from the `show qos port counters egressqueue` command

```
Port 1 Egress Queue Counters:
  Port queue length .......... 48 (maximum 128)
  Egress queue length:
    Queue 0 ................... 0 (maximum 16)
    Queue 1 ................... 16 (maximum 16)
    Queue 2 ................... 16 (maximum 16)
    Queue 3 ................... 16 (maximum 16)
    Queue 4 ................... 0 (maximum 16)
    Queue 5 ................... 0 (maximum 16)
    Queue 6 ................... 0 (maximum 16)
    Queue 7 ................... 0 (maximum 16)
```

Figure 15: Example output from the `show qos port counters trafficclass` command

```
QOS Counter Information
Port 1:
  Policy: 0
  Traffic Class 0:
    Aggregate Bytes ............... 15220
    BwConformanceClass1 bytes .... 15220
    BwConformanceClass2 bytes .... 0
    BwConformanceClass3 bytes .... 0
    Dropped bytes ................ 15220
  Default Traffic Class:
    Aggregate Bytes ............... 680
    BwConformanceClass1 bytes .... 680
    BwConformanceClass2 bytes .... 0
    BwConformanceClass3 bytes .... 0
    Dropped bytes ................ 0
```

**show qos accelerator**

**Syntax**

```
SHOW QOS ACCELERATOR COUNTERS
  TRAFFICCLASS={trafficclass-list|DEFAULT|ALL}
  EGRESSQUEUE={queue-list}
```

where:

- `queue-list` is either an integer in the range 0 to 7; a range of integers (specified as 0-3) or a comma separated list of integers and/or ranges, without spaces.
- `trafficclass-list` is a integer in the range of 0-1023; a range of integers (specified as 0-3) or a comma separated list of integers and/or ranges without spaces.

**Description**

This command displays QoS configuration information for the accelerator card counters.

If `counters trafficclass` is specified the counters for the traffic classes belonging to the policy attached to the accelerator card will be displayed. If no value or `all` is specified, then information about all traffic classes is displayed. If `default` is specified then only information about the default traffic class will be displayed.
If a list of traffic classes is specified then only information about those traffic classes is displayed.

If counters egressqueue is specified the counters for the specified queues will be displayed. If no value is specified, then information about all egress queues is displayed. If a list of queues are specified, then only information about those egress queues is displayed.

Figure 16: Example output from the **show qos accelerator counters egressqueue** command

<table>
<thead>
<tr>
<th>Accel Card Egress Queue Counters:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total queue length ........ 0 (maximum 896)</td>
</tr>
<tr>
<td>Egress queue length:</td>
</tr>
<tr>
<td>Queue 0 .................... 0 (maximum 896)</td>
</tr>
<tr>
<td>Queue 1 .................... 0 (maximum 896)</td>
</tr>
<tr>
<td>Queue 2 .................... 0 (maximum 896)</td>
</tr>
<tr>
<td>Queue 3 .................... 0 (maximum 896)</td>
</tr>
<tr>
<td>Queue 4 .................... 0 (maximum 896)</td>
</tr>
<tr>
<td>Queue 5 .................... 0 (maximum 896)</td>
</tr>
<tr>
<td>Queue 6 .................... 0 (maximum 896)</td>
</tr>
<tr>
<td>Queue 7 .................... 0 (maximum 896)</td>
</tr>
</tbody>
</table>

Figure 17: Example output from the **show qos accelerator counters trafficclass** command

<table>
<thead>
<tr>
<th>QOS Counter Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerator Interface:</td>
</tr>
<tr>
<td>Policy: 1</td>
</tr>
<tr>
<td>There are no traffic classes to display.</td>
</tr>
<tr>
<td>Default Traffic Class:</td>
</tr>
<tr>
<td>Aggregate Bytes ............ 0</td>
</tr>
<tr>
<td>BwConformanceClass1 bytes .... 0</td>
</tr>
<tr>
<td>BwConformanceClass2 bytes .... 0</td>
</tr>
<tr>
<td>BwConformanceClass3 bytes .... 0</td>
</tr>
<tr>
<td>Dropped bytes ............. 0</td>
</tr>
</tbody>
</table>
Enable/Disable Port Egress Queue and Flow Control

This enhancement enables you to enable and disable egress queues and flow control on switch ports, using the following commands.

**disable switch port**

**Syntax**

DISABLE SWITCH PORT=(port-list|ALL) [AUTOMDI]
[EGRESSQUEUE=queue-list] [FLOW=PAUSE|JAMMING]

where:

- **port-list** is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered switch Ethernet port, including uplink ports.

- **queue-list** is an egress queue number, a range of queue numbers (specified as n-m), or a comma separated list of queue numbers and/or ranges. Egress queue numbers start at 0 and end at 7.

**Description**

This command disables a port or group of ports on the switch or disables auto MDI/MDI-X mode on the ports or disables flow control or disables egress queues. If the port is disabled, it will no longer send or receive packets. If egress queues are disabled packets scheduled for those queues will no longer be transmitted. Disabling a switch does not disable STP operation on the port. Ports should be disabled if there faulty wiring or equipment attached to the ports, or as a security measure to stop access from intruders. Switch ports are enabled by default.

The **port** parameter specifies the port or ports to be disabled or to have auto MDI/MDI-X mode disabled.

The **flow** parameter specifies the type of flow control to be disabled for the port. If **pause** is specified, flow control for full duplex ports by sending PAUSE frames will be disabled. If **jamming** is disabled, flow control for half duplex by sending a jamming signal will be disabled. No flow control is enabled by default.

The **automdi** parameter disables auto MDI/MDI-X mode, and sets the polarity to the default of MDI-X. Auto MDI/MDI-X mode is enabled by default.

The **egressqueue** parameter specifies the egress queue(s) that are to be disabled. If you specify this parameter, you must supply a value for it. At switch startup all egress queues are enabled by default.

**Examples**

To disable auto MDI/MDI-X mode on ports 2 and 4 use this command:

```
disable switch port=2,4 automdi
```

To disable egress queues 0 and 3 to 5, on port 1, use the command:

```
disable switch port=1 egressqueue=0,3-5
```

To disable pause frames on port 1 use the command:

```
disable switch port=1 flow=pause
```
enable switch port

Syntax
ENABLE SWITCH PORT={port-list|ALL} [AUTOMDI]
[EGRESSQUEUE=queue-list] [FLOW=PAUSE|JAMMING]

where:
- *port-list* is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered switch Ethernet port, including uplink ports.
- *queue-list* is an egress queue number, a range of queue numbers (specified as n-m), or a comma separated list of queue numbers and/or ranges. Egress queue numbers start at 0 and end at 7.

Description
This command enables a port or group of ports on the switch or enables auto MDI/MDI-X mode on the ports or enables flow control or enables egress queues. If the port is enabled, it will start to send or receive packets. Enabling a switch does not enable STP operation on the port. Switch ports are enabled by default.

The *port* parameter specifies the port or ports to be enabled or to have auto MDI/MDI-X mode disabled.

The *flow* parameter specifies the type of flow control to be enabled for the port. If *pause* is specified, flow control for full duplex ports by sending PAUSE frames will be enabled. If *jamming* is enabled, flow control for half duplex by sending a jamming signal will be enabled. No flow control is enabled by default.

The *automdi* parameter enables auto MDI/MDI-X mode, and sets the polarity to the default of MDI-X. Auto MDI/MDI-X mode is enabled by default.

The *egressqueue* parameter specifies the egress queue(s) that are to be enabled. If you specify this parameter, you must supply a value for it. At switch startup all egress queues are enabled by default.

Examples
To enable auto MDI/MDI-X mode on ports 2 and 4 use this command:

```
enable switch port=2,4 automdi
```

To enable egress queues 0 and 3 to 5, on port 1, use the command:

```
enable switch port=1 egressqueue=0,3-5
```

To enable pause frames on port 1 use the command:

```
enable switch port=1 flow=pause
```
802.1x and RADIUS Servers Enhancement for EAP-TLS

The authentication server verifies the supplicant’s details, passed to it by the authenticator. This implementation of 802.1x control requires that a port acting as an authenticator must communicate with a RADIUS authentication server. The RADIUS server must be capable of receiving and deciphering EAP in RADIUS packets.

Prior to this enhancement, the supported supplicant encryption mechanisms for communication with the RADIUS server were EAP-MD5 and EAP-OTP. With this enhancement the encryption methods supported by authenticators are EAP-MD5, EAP-OTP, EAP-TLS, EAP-TTLS, and EAP-PEAP.

Steps in the Authentication Process

Until authentication is successful, the supplicant can only access the authenticator to perform authentication message exchanges, or access services not controlled by the authenticator’s controlled port.

Initial 802.1x control begins with an unauthenticated supplicant and an authenticator. A port under 802.1x control acting as an authenticator is in an unauthorised state until authentication is successful.

1. Either the authenticator or the supplicant can initiate an authentication message exchange. The authenticator initiates the authentication message exchange by sending an EAPOL packet containing an encapsulated EAP-Request/Identity packet. The supplicant initiates an authentication message exchange by sending an EAPOL-Start packet, to which the authenticator responds by sending an EAPOL packet containing an encapsulated EAP-Request/Identity packet.

2. The supplicant sends an EAPOL packet containing an encapsulated EAP-Response/Identity packet to the authentication server via the authenticator, confirming its identity.

3. The authentication server selects an EAP authentication algorithm to verify the supplicant’s identity, and sends an EAP-Request packet to the supplicant via the authenticator.

4. The supplicant provides its authentication credentials to the authenticator server via an EAP-Response packet.

5. The authentication server either sends an EAP-Success packet or EAP-Reject packet to the supplicant via the authenticator.

6. Upon successful authorisation of the supplicant by the authenticator server, a port under 802.1x control is in an authorised state, unless the MAC associated with the port is either physically or administratively inoperable. Also upon successful authorisation of the supplicant by the authenticator server, the supplicant is allowed full access to services offered via the controlled port. If piggybacking is enabled on the authorised authenticator port, any other device connected will also be give full access.
7. When the supplicant sends an EAPOL-Logoff message to the authenticator the port under 802.1x control is set to unauthorised.

A successful authentication message exchange, initiated and ended by a supplicant using OTP authentication, is shown in below.

To minimise the risk of denial-of-service attacks by issuing EAPOL-Logoff messages to an Authenticator Port Access Entity (PAE) from a third party device, we recommend that 802.1x not be used in a shared media LAN.

Figure 18: Authentication Messaging Exchange Initiated by the Supplicant.
Logging and SNMP Traps for PIM-SM

PIM-SM can now be configured to produce log messages in response to status changes and errors, and SNMP traps.

Note This enhancement does not apply to PIM-DM.

Status log messages
Events that will trigger a status-change log message are:
- PIM interface is disabled
- PIM interface is enabled
- PIM neighbour adjacency has timed out
- PIM neighbour generation ID has changed
- PIM neighbour has changed port
- PIM RP has changed
- PIM DR has changed
- PIM BSR has changed

Error log messages
Errors that will trigger a log message are:
- Invalid PIM packet
- Invalid destination address
- Fragmentation reassembly
- Packet too short
- Bad group address encoding
- Bad source address encoding
- Missing option
- Internal error
- Receive packet - a range of errors that mean the packet was received but cannot be forwarded.

SNMP traps
The following traps are sent:
- PimInterfaceUpTrap - generated when a PIM interfaces comes up and is active
- PimInterfaceDownTrap - generated when a PIM interfaces goes down and is inactive
- PimNeighbourLossTrap - generated when a known PIM neighbour has lost adjacency or has timed-out. This trap is part of the experimental PIM MIBs group
- PimNeighbourAddedTrap - generated when a PIM neighbour is added
- PimNeighbourDeletedTrap - generated when a PIM neighbour is deleted
- PimErrorTrap - generated when any one of the PIM error counters is incremented or when a log message of subtype LOG_STY_PIM_ERROR is generated (see list of errors above)
**Command syntax**  To specify the type of log messages and SNMP traps that the switch generates, use the new command:

```
set pim log={none|status|error|all} [trap={none|status|error|all}]
```

To display the specified options, use the command:

```
show pim debug
```

```
PIM4 Debug Options
-----------------------------------------------
Debug Options Enabled   : None
Logging Options Enabled : status
Trapping Options Enabled: none
```
PIM4 Hash Mask Length Support

This enhancement enables you to specify the number of bits of the group number that are significant when selecting a rendezvous point (RP) candidate, if this switch becomes the BSR. Prior to this enhancement, a single RP was selected for all groups that match its prefix. With this enhancement, the RP selection will be spread across candidate RPs based on the hash mask length chosen.

Note that previous release versions did not correctly support the PIM hash mask length option. As a result, the RP selection calculation differs between old and new release versions. If a network contains switches running a mixture of versions, this leads to incorrect forwarding behaviour. To avoid this issue, either ensure that all devices on the network correctly support the hash mask length option (recommended), or ensure that the following both hold:

- The hash mask length option on all BSR candidates is configured to 4 bits. This implies that all BSR candidates must be running a new release.
- All RP candidates use a common prefix of 224.0.0.0/240.0.0.0

This will have the side effect of collapsing all groups to use a single PIM RP.

Modified Commands

To configure the hash mask length, use the highlighted new parameters in the following existing commands.

add pim bsrcandidate

Syntax

ADD PIM BSRCANDIDATE [HASHmasklength=0..32] [other-options]

The hashmasklength parameter specifies the number of bits of the group number to use when selecting a rendezvous point (RP) candidate if this switch becomes the BSR. A higher number increases the spread of groups across RPs. The default is 30.

Examples

To add the switch as a Bootstrap Router Candidate to a PIM domain, with a preference of 10 to become the bootstrap router in the domain and a hash mask length of 0, use the command:

add pim bsrcandidate preference=10 hashmasklength=0

set pim bsrcandidate

Syntax

SET PIM BSRCANDIDATE [HASHmasklength=0..32] [other-options]

The hashmasklength parameter specifies the number of bits of the group number to use when selecting a rendezvous point (RP) candidate if this switch becomes the BSR. A higher number increases the spread of groups across RPs. The default is 30.
OSPF Not-So-Stubby Area Option (NSSA)

An NSSA is an optional type of Open Shortest Path First (OSPF) area. NSSAs are similar to the existing OSPF stub area configuration option but have the additional capability of importing AS external routes in a limited fashion. NSSAs are described in RFC1587, “The OSPF Not-So-Stubby Area (NSSA) Option”.

Modified Commands

Use the highlighted new parameter options in the following existing commands to configure an NSSA. Several show commands have also been modified to display NSSA information:

- show ospf area
- show ospf
  - ‘AS boundary router status” displays NSSA
- show ospf lsa
  - You can now specify the parameter type=asnssa

add ospf area

Syntax

ADD OSPF AREA={BACKBONE|area-number} 
[AUTHENTICATION={NONE|PASSWORD}] [STUBAREA={ON|OFF|YES|NO|NSSA|TRUE|FALSE}] [STUBMETRIC=0..16777215] [SUMMARY={SEND|NONE|OFF|NO|FALSE}]

The stubarea parameter specifies whether or not the switch treats the area as a stub area. The value nssa specifies that the area is a Not-so-stubby-area (NSSA). External routes can be imported as type-7 advertisements in a NSSA.

The summary parameter controls the generation of summary LSAs into stub areas. If stubarea is set to nssa then the default is send, otherwise the default is none.

set ospf area

Syntax

SET OSPF AREA={BACKBONE|area-number} 
[AUTHENTICATION={NONE|PASSWORD}] [STUBAREA={ON|OFF|YES|NO|NSSA|TRUE|FALSE}] [STUBMETRIC=0..16777215] [SUMMARY={SEND|NONE|OFF|NO|FALSE}]

The stubarea parameter specifies whether or not the switch treats the area as a stub area. The value nssa specifies that the area is a Not-so-stubby-area (NSSA). External routes can be imported as type-7 advertisements in a NSSA.

The summary parameter controls the generation of summary LSAs into stub areas. If stubarea is set to nssa then the default is send, otherwise the default is none.
set ospf

Syntax

```
SET OSPF [ASEXTERNAL={ON|OFF|NSSA}] [DEFROUTE={ON|OFF|TRUE|FALSE|YES|NO}] [TYPE={1|2}] [METRIC=0..16777215] [DYNINTERFACE={STUB|ASEXTERNAL|NONE|NO|OFF|FALSE}] [RIP={OFF|EXPORT|IMPORT|BOTH}] [ROUTERID=ipaddr] [PTPSTUB={ON|OFF|YES|NO|TRUE|FALSE}] [STATICEXPORT=(YES|NO)]
```

The `asexternal` parameter specifies whether or not the switch will act as an Autonomous System boundary router and how routes are imported. A switch is said to be an Autonomous System (AS) boundary router if it has some interfaces in the OSPF AS and some interfaces that are not in the AS. Typically the switch will have some “external” routes in its routing table associated with the interfaces that are not in the AS. If `asexternal` is set to `on` these external routes will be advertised into the AS as type-5 LSAs for non-NSSAs and as type 7 LSAs for NSSAs. If `asexternal` is set to `nssa`, external routes will only be added to NSSAs as type 7 LSAs, which will be translated if appropriate to a type 5 LSA at an NSSA ABR. `asexternal` should be set to `nssa` if this switch only uses NSSAs. If `asexternal` is set to `off` these external routes will not be advertised into the AS. The default is `off`. 
OSPF Auto Cost Calculation

This enhancement allows OSPF interfaces to automatically set the OSPF metric of an IP interface on the basis of the bandwidth of the interface, instead of the system administrator manually setting the OSPF metric. Automatic setting takes into account that the speed of an interface can change over time, when ports change link state or change speed via auto negotiation or manual setting. If metrics are manually set, some interfaces are preferred when they should not be because the network configuration dynamically changes.

Note that interface speed used in the cost calculation is the average interface speed. For example, if the interface is a VLAN with two ports up, and one port has a speed of 10 and the other a speed of 100, then the metric will be 18.

To configure auto cost calculation:

1. Do not set the OSPF metric manually in the `add ip interface` command. If you have, remove the manual setting, using the command:
   ```
   set ip interface=int ospfmetric=default
   ```
   The `ospfmetric` parameter specifies the cost of crossing the logical interface, for OSPF. If `default` is specified the interface is restored to the default metric value. The setting of the OSPF metric to a value other than `default` provides a mechanism to provide a metric for an interface that is preferred over the OSPF automatic metric setting (if enabled via `set ospf autocost=on`).

2. Set `autocost` to `on` and change the reference bandwidth if necessary, using the command:
   ```
   set ospf autocost=on [refbandwidth=10..10000]
   ```
   The `autocost` parameter specifies whether or not the switch will assign OSPF interface metrics based on the available interface bandwidth. If an OSPF metric has been manually assigned using the `add ip interface ospfmetric=x`, the manual metric setting will take priority over an automatic metric setting. The default is `off`.
   The `refbandwidth` parameter specifies the reference bandwidth in megabits per second used for calculating the OSPF metric. The cost is calculated as `refbandwidth / Interface Bandwidth`. Using the default settings, the automatic cost calculation will result in an OSPF metric of 10 for a fast Ethernet (100M) interface. The `autocost` parameter must be set to `on` for the parameter `refbandwidth` to take effect. The default is 1000.

3. To check the settings, use the command:
   ```
   show ospf
   ```
IP DSCP Override

This enhancement enables you to give a particular DSCP to packets that come from the switch’s CPU, using the following command.

**Syntax**

```
SET IP DSCPOVERRIDE={NONE|0..63}
```

**Description**

This command sets the value that will be written into the DSCP field in the IP header of packets that are sent from the CPU to the switching chip for forwarding.

The `dscpoverride` parameter specifies the DSCP that is written into the DSCP field. The default value of the parameter is `none`, whereby the DSCP field in the packets will not be altered from whatever value the originating software module might have written into it.

**Examples**

To set the DSCP in the IP header to 56, use the command:

```
set ip dscpoverride=56
```

To stop overriding IP DSCP, use the command:

```
set ip dscpoverride=none
```

CPU Transmit User Priority Override

This enhancement enables you to give a particular 802.1p priority to packets that come from the switch’s CPU, using the following command.

**Syntax**

```
SET SWITCH CPUTXPRIORITY={NONE|0..7}
```

**Description**

This command specifies the 802.1p value that will be inserted into packets that are sent from the CPU to the switching chip to be transmitted.

The `cputxpriority` parameter specifies the value that will be put into the 802.1p field of tagged packets being sent from the CPU. The default value of the parameter is `none`, where the CPU does not set the 802.1p field in packet headers. `none` can be specified to stop the overriding.

**Examples**

To set the switch to place the value 7 in the 802.1p field of packets sent from the CPU, use the command:

```
set switch cputxpriority=7
```

To disable overriding CPU transmit priority, use the command:

```
set switch cputxpriority=none
```
CPU Transmit Queue Priority Override

This enhancement enables you to send packets that come from the switch’s CPU to a specific queue, using the following command.

**Syntax**

```
SET SWITCH CPUTXQUEUE={NONE | 0..7}
```

**Description**

This command specifies the transmit queue that will be used for packets that are sent from the CPU to the switching chip for transmission.

The `cputxqueue` parameter specifies the egress queue into which packets from the CPU will be placed by the switching chip. The default value of the parameter is `none`, which means that packets from the CPU will be put into egress queue 0. `none` can be specified to stop the overriding.

**Examples**

To set the switch to put CPU-initiated packets into egress queue 7, use the command:

```
set switch cputxqueue=7
```

To stop specifying an egress queue for CPU-initiated packets, use the command:

```
set switch cputxqueue=none
```
Syslog Facility Override

This enhancement enables you to override the facility parameter in syslog messages.

Modified Commands

To set the syslog facility, use the highlighted new parameters in the following existing commands. The show log output command has also been modified to display the facility setting.

create log output

Syntax

CREATE LOG OUTPUT={TEMPORARY | PERMANENT | output-id} [DESTINATION=SYSLOG] [FACILITY={DEFAULT | LOCAL1..LOCAL7}] [other-options]

The facility parameter specifies whether to override the mapping between logging facility type and syslog facility identifiers. The default option keeps the mapping between type and facility. If you specify a local value (local1, local2, ..., local7) then the syslog facility identifier will always be sent with the value specified. The facility parameter is valid only if destination is set to syslog. The default for facility is default.

set log output

Syntax

SET LOG OUTPUT={TEMPORARY | PERMANENT | output-id} [DESTINATION=SYSLOG] [FACILITY={DEFAULT | LOCAL1..LOCAL7}] [other-options]

The facility parameter specifies whether to override the mapping between logging facility type and syslog facility identifiers. The default option keeps the mapping between type and facility. If you specify a local value (local1, local2, ..., local7) then the syslog facility identifier will always be sent with the value specified. The facility parameter is valid only if destination is set to syslog. The default for facility is default.
Temperature and Fan Monitoring

New information has been added to the show system command output to indicate temperature, fan and PSU state. In particular, the show system output now shows if temperature and fan monitoring fails.

Figure 19: New section of output of show system command

| PSU1: (AC) | Fan: Normal | Temp: Normal | Power: Normal |
| PSU2: (AC) | Fan: Normal | Temp: Normal | Power: Failed |

Current temperature : Normal

FAN
-----------------
Main fan Normal
-----------------

Table 5: New parameters displayed in the show system command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan</td>
<td>Fan voltage of each PSU/Fan Module; one of “Normal” or “Failed”.</td>
</tr>
<tr>
<td>Temp</td>
<td>Temperature of each PSU/Fan Module; one of “Normal” or “Failed”</td>
</tr>
<tr>
<td>Power</td>
<td>Power output of each PSU; one of “Normal” or “Failed”</td>
</tr>
<tr>
<td>Current Temperature</td>
<td>Internal temperature status of the board; one of “Normal”, “Failed” or “Warning”. “Warning” indicates that temperature and fan monitoring has failed.</td>
</tr>
<tr>
<td>FAN</td>
<td>Status of each internal fan; one of “Normal”, “Failed” or “Warning”. “Warning” indicates that temperature and fan monitoring has failed.</td>
</tr>
</tbody>
</table>

The “Warning” state for current board temperature and internal fans is also indicated by the switch’s system LED flashing 3 times (flash, flash, flash, pause, flash, flash, flash, pause...).
Use of Default Route by Proxy ARP

The switch uses proxy ARP in situations where there are hosts on the LAN that have not been configured with correct information about which IP subnets are being used on the local LAN and which are being used on remote LANs. Such hosts are likely to send ARP requests for IP addresses outside the range of addresses being used on the local LAN. If the switch knows a route to the address being erroneously ARPed for, then the switch intercepts the ARP broadcast packets and sends replies in which it substitutes its own physical address for that of the remote host. By responding to the ARP request, the switch ensures that all subsequent packets from the local host to that remote host are directed to the switch’s physical address and it can then forward these to the remote host.

Prior to this enhancement, the switch only acted as a proxy when it had a specific route to the remote host. This enhancement allows the switch to also send proxy ARP responses for hosts that it would reach via the default route.

Modified Commands

To enable proxy ARP to use the default route, use the highlighted new options in the following existing commands.

The PArp entry in the output of the `show ip interface` command has also been modified, and now indicates whether this interface supports proxy ARP and if ARP responses will be generated even for hosts that can only be reached via the default route. It is one of “On” (respond to ARP requests only if a specific route exists), “Def” (respond to ARP requests if a specific route or a default route exists), or “Off”.

**add ip interface**

```plaintext
Syntax
ADD IP INTERFACE=interface [PROXYARP={FALSE|NO|OFF|ON|TRUE|YES|STRICT|DEFROUTE}] [other-options]
```

The `proxyarp` parameter enables or disables proxy ARP responses to ARP requests. If you specify `strict`, `on`, `true` or `yes`, the switch will respond to proxy ARP requests using specific routes if they exist. If you specify `off`, `false` or `no`, the switch will not respond to ARP requests. If you specify `defroute`, the switch will respond to proxy ARP Requests using specific routes if they exist or a default route (0.0.0.0) if it exists. The `proxyarp` parameter is valid for VLAN interfaces. The default is `off`. 
set ip interface

Syntax

```
set ip interface interface [proxyarp={false|no|off|on|true|yes|strict|defroute}] [other-options]
```

The `proxyarp` parameter enables or disables proxy ARP responses to ARP requests. If you specify `strict`, `on`, `true` or `yes`, the switch will respond to proxy ARP requests using specific routes if they exist. If you specify `off`, `false` or `no`, the switch will not respond to ARP requests. If you specify `defroute`, the switch will respond to proxy ARP Requests using specific routes if they exist or a default route (0.0.0.0) if it exists. The `proxyarp` parameter is valid for VLAN interfaces. The default is `off`.
Increase in VLAN Name Length

The VLAN parameter now enables you to have a VLAN name of up to 32 characters long. For example, in the command:

```
CREATE VLAN=\vlan-name\ VID=2..4094
```

`\vlan-name\` is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore character ("_"), and the hyphen ("-"). The `\vlan-name\` cannot be a number or ALL or DEFAULT.

RSTP BPDU Loopback Detection

The output of the `show stp port` command has been changed to display when RSTP detects a downstream loop and puts the port into the “Backup (Loopback Disabled)” state. It also shows the number of times it transitions to that state.

An example of the new output is:

```
<table>
<thead>
<tr>
<th>Port ................. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSTP Port Role ...... Disabled</td>
</tr>
<tr>
<td>State ............... Discarding</td>
</tr>
<tr>
<td>Point To Point ...... No (Auto)</td>
</tr>
<tr>
<td>Port Priority ....... 128</td>
</tr>
<tr>
<td>Port Identifier ..... 8004</td>
</tr>
<tr>
<td>Pathcost ............. 200000</td>
</tr>
<tr>
<td>Designated Root ..... 32768 : 00-00-cd-05-19-28</td>
</tr>
<tr>
<td>Designated Cost ..... 0</td>
</tr>
<tr>
<td>Designated Bridge ... 32768 : 00-00-cd-05-19-28</td>
</tr>
<tr>
<td>Designated Port ..... 8004</td>
</tr>
<tr>
<td>EdgePort ............. No</td>
</tr>
<tr>
<td>VLAN membership ..... 1</td>
</tr>
<tr>
<td>Counters:</td>
</tr>
<tr>
<td>Loopback Disabled 0</td>
</tr>
</tbody>
</table>
```

The “RSTP Port Role” entry can now have a status of “Backup (Loopback Disabled)”, which indicates the port has transmitted and received the same RSTP BPDU. This state has the same behaviour as the “Backup” state; it drops all packets except BPDUs.

The “Loopback Disabled” counter indicates the number of transitions to the “Backup (Loopback Disabled)” state for the port.
Telnet Session Limit and Timeout

This enhancement enables you to limit the number of concurrent telnet sessions, and to specify an idle timeout for telnet sessions. To configure this, use the command:

```
SET TELNET [IDLETIMEOUT=0..4294967295] [MAXSESSIONS=0..30]
[other-options]
```

The `idletimeout` parameter specifies a period of time, in seconds, for the telnet server's idle timer. If the specified time period lapses since the last time a telnet session received data from the remote client, the session is terminated; this applies from the moment that the telnet session becomes established, regardless of whether the user has logged in or not. If you specify 0 (zero), the idle timer remains off, and the session must be explicitly terminated. The default is 0.

If you modify the telnet server idle timeout period while there are established telnet sessions, the idle timers for those sessions are reset so that they use the new timeout value. The sessions’ timer is reset, so they lose any idle time accumulated prior to the idle time modification.

The `maxsessions` parameter specifies the number of concurrent telnet sessions that are supported by the local switch. Once this limit is reached, any subsequent session requests are rejected. The session limit cannot be set below the number of currently established telnet sessions. By default, 30 concurrent telnet sessions are supported.

The `show telnet` command now displays the number of currently-established sessions, the maximum number of sessions allowed, and the idle timeout in seconds.

```
TELNET Module Configuration
------------------------------------------
Telnet Server .................. Enabled
Telnet Server Listen Port ...... 23
Telnet Terminal Type ........... UNKNOWN
Telnet Insert Null's .......... Off
Telnet Current Sessions ........ 1
Telnet Session Limit .......... 12
Telnet Idle Timeout .......... 180
------------------------------------------
```
Extensions to Stacking

Synchronising Time Across Stacks

When you set the time through the CLI, SNMP, or NTP, the stack synchronises to that time. The time is then periodically checked and resynchronised across the stack. Special considerations are:

- When you create a stack, switches automatically synchronise to the time on one switch; however, this may not be the correct time. You may want to verify the time and set it by using the `show time` and `set time` commands.

- When you add a switch to a stack, the new member synchronises its time within 10-20 seconds of the `enable stack` command or plugging in an enabled switch. You may want to set the time on the new switch before adding it to the stack by using the `show time` and `set time` commands.

Copy command

The `copy` command has been modified so that switches in a stack can copy files to stack members.

Syntax

```
COPY [device:]filename1.ext [device:]filename2.ext
```

When Stacking is enabled:

```
COPY [source-hostid:][device:]filename1.ext
       [dest-hostid:][device:]filename2.ext]
```

where:

- `device` indicates the physical location of the file and is either flash or NVS. The default is flash.

- `filename1` is the name of an existing file.

- `filename2` is a valid filename between 1 and 28 characters long that does not already exist. Invalid characters are * + = “| \ ] ; : ? / , < >. Valid characters are:
  - uppercase and lowercase letters
  - digits 0-9
  - characters ~ ` ! @ # $ % ^ & ( ) _ - { } 

This variable is optional when Stacking is enabled but when not present, the file to be copied keeps the same name.

- `ext` is a 3-letter file extension and can be any text file (for example, txt, cfg, scp, hlp, htm, spa, or mds) except ukf or stk. The original file and the copy must have the same extension.

- `source-hostid` indicates the host ID of the stack member that holds the file to be copied. This variable is optional but if present, the `dest-hostid` is required; if not present, the `copy` command functions in the standard way.

- `dest-hostid` indicates stack members to receive the copied file:
  - a unique number from 1 to 32
  - a range of unique numbers from 1 to 32
  - a comma-separated list
This variable is optional but if present, the `source-hostid` is required; if not present, the `copy` command functions in the standard way.

**Description**  
This command copies a text file from one type of memory to another, either flash or NVS, in a single switch or stack.

**Example**  
To copy `admin.cfg` from NVS and rename it to “admin2.cfg” in flash, use the command:

```
cop nvs:admin.cfg admin2.cfg
```

To copy `file1.txt` from a stack member with host ID 1 to stack members with host IDs 2 through 4, and also 6, and rename it “file2.txt”, use the command:

```
cop 1:file1.txt 2-4,6:file2.txt
```

To copy `file1.txt` from the stack member with host ID 1 to the stack member with host ID 2 without renaming it, use the command:

```
cop 1:file.txt 2:file.txt
```

While connected to the stack member with host ID 1, use a host-directed command to direct host ID 2 to copy `file.txt` from its flash memory to its NVS without renaming it:

```
2:cop file.txt nvs:file.txt
```

Note that you **cannot** use a host-directed command to direct the `copy` command to more than one stack member at a time. For example, “1-3: cop file.txt nvs: file.txt” returns an error.

**Related Commands**
- delete file
- show file
ICMP Router Discovery Advertisements

Router Discovery on the Switch

The switch supports all of RFC 1256, ICMP Router Discovery Messages, as it applies to routers. If this feature is configured, the switch sends router advertisements periodically and in response to router solicitations. It does not support the Host Specification section of this RFC.

Benefits

Before an IP host can send an IP packet, it has to know the IP address of a neighbouring router that can forward it to its destination. ICMP Router Discovery messages allow routers to automatically advertise themselves to hosts. Other methods either require someone to manually keep these addresses up to date, or require DHCP to send the router address, or require the hosts to be able to eavesdrop on whatever routing protocol messages are being used on the LAN.

Router Discovery Process

The following table summarises what happens when Router Discovery advertisements are enabled for interfaces on the switch.

Table 6: Router discovery process

<table>
<thead>
<tr>
<th>When...</th>
<th>Then...</th>
</tr>
</thead>
</table>
| Router Discovery advertising starts on a switch interface because:  
- the switch starts up, or  
- advertisements are enabled on the switch or on an interface | the switch multicasts a router advertisement and continues to multicast them periodically until router advertising is disabled. |
| a host starts up | the host may send a router solicitation message. |
| the switch receives a router solicitation | the switch multicasts an early router advertisement on the multicast interface on which it received the router solicitation. |
| a host receives a router advertisement | the host stores the IP address and preference level for the advertisement lifetime. |
| the lifetime of all existing router advertisements on a host expires | the host sends a router solicitation. |
| a host does not receive a router advertisement after sending a small number of router solicitations | the host waits for the next unsolicited router advertisement |
| a host needs a default router address | the host uses the IP address of the router or L3 switch with the highest preference level. |
| Router Discovery advertising is deleted from the physical interface (DELETE IP ADVERTISE command), or the logical interface has ADVERTISE set to NO (SET IP INTERFACE command) | the switch multicasts a router advertisement with the IP address(es) that stopped advertising, and a lifetime of zero (0). It continues to periodically multicast router advertisements for other interfaces. |
| the switch receives a router advertisement from another router | the switch does nothing but silently discards the message. |
A *router advertisement* is an ICMP (type 10) message containing:

- In the destination address field of the IP header, the interface’s configured advertisement address, either 224.0.0.1 (ALL) or 255.255.255.255 (LIMITED).
- In the lifetime field, the interface’s configured advertisement lifetime.
- In the Router Address and Preference Level fields, the addresses and preference levels of all the logical interfaces that are set to advertise.

A *router solicitation* is an ICMP (type 10) message containing:

- Source Address: an IP address belonging to the interface from which the message is sent
- Destination Address: the configured Solicitation Address, and
- Time-to-Live: 1 if the Destination Address is an IP multicast address; at least 1 otherwise.

The router advertisement *interval* is the time between router advertisements. For the first few advertisements sent from an interface (up to 3), the switch sends the router advertisements at intervals of at most 16 seconds. After these initial transmissions, it sends router advertisements at random intervals between the minimum and maximum intervals that the user configures, to reduce the probability of synchronization with the advertisements from other routers on the same link. By default the minimum is 450 seconds (7.5 minutes), and the maximum is 600 seconds (10 minutes).

The *preference level* is the preference of the advertised address as a default router address relative to other router addresses on the same subnet. By default, all routers and layer 3 switches have the same preference level, zero (0). While it is entered as a decimal from -2147483648 to 2147483647, it is encoded in router advertisements as a twos-complement hex integer from 0x8000000 to 0x7fffffff. A higher preference level is preferred over a lower value.

The *lifetime* of a router advertisement is how long the information in the advertisement is valid. By default, the lifetime of all advertisements is 1800 seconds (30 minutes).
Configuration Procedure

By default, the switch does not send router advertisements.

To configure the router to send router advertisements

1. **Set the physical interface to advertise.**
   
   For each physical interface that is to send advertisements, add the interface. In most cases the default advertising parameters will work well, but you can change them if required. By default, the switch sends router advertisements every 7.5 to 10 minutes, with a lifetime of 30 minutes. These settings are likely to work well in most situations, and will not cause a large amount of extra traffic, even if there are several routers on the LAN. If you change these settings, keep these proportions:
   
   \[
   \text{lifetime} = 3 \times \text{maxadvertisementinterval}
   \]
   
   \[
   \text{minadvertisementinterval} = 0.75 \times \text{maxadvertisementinterval}
   \]

   To change these settings, use one of the commands:
   
   ```
   add ip advertise interface
   ```
   
   ```
   set ip advertise interface
   ```

2. **Stop advertising on other logical interfaces.**
   
   By default, logical interfaces are set to advertise if their physical interface is set to advertise. If the physical interface has more than one logical interface (IP multihoming), and you only want some of them to advertise, set the other logical interfaces not to advertise with one of the commands:
   
   ```
   add ip interface=interface ipaddress={ipadd|dhcp}
       advertise=no [other-ip-parameters]
   ```
   
   ```
   set ip interface=interface advertise=no
       [other-ip-parameters]
   ```

3. **Set preference levels.**
   
   By default, every logical interface has the same preference for becoming a default router (mid range, 0). To give a logical interface a higher preference, increase `preferencelevel`. To give it a lower preference, decrease this value. If it should never be used as a default router, set it to `notdefault`.
   
   ```
   add ip interface=interface ipaddress={ipadd|dhcp}
       preferencelevel={-2147483648..2147483647|notdefault}
       [other-ip-parameters]
   ```
   
   ```
   set ip interface=interface
       [preferencelevel={-2147483648..2147483647|notdefault}]
       [other-ip-parameters]
   ```

4. **Enable advertising.**
   
   To enable router advertisements on all configured advertising interfaces, use the command:
   
   ```
   enable ip advertise
   ```

5. **Check advertise settings.**
   
   To check the router advertisement settings, use the command:
   
   ```
   show ip advertise
   ```
Adopting the VRRP IP Address

Benefits of VRRP IP Address Adoption

The VRRP master router can adopt the IP address of the virtual router (VR), and respond to the following packets destined for the VR IP address, even if it does not own this IP address on any of its interfaces:

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

VRRP IP Address Adoption allows continuous accessibility of the VR IP address even as the VR master changes. Using this feature:

- You can easily tell whether the VR is functioning, by pinging the single VR IP address.
- You can easily monitor the performance of the VR, regardless of which participating router is acting as master.
- DNS relay can continue functioning via the same IP address at all times.

Risks of VRRP IP Address Adoption

When VRRP IP Address Adoption is used, the master router accepts packets destined for the virtual router, even though it may not own this IP address. This does not conform to RFC 2338. Because the same IP address refers to different devices at different times, there is a risk of confusion arising. This risk can be reduced by a suitable network management policy.

Recommendations

Before using VR IP address adoption, consider the following guidelines to avoid confusion:

- Ensure that the VR has an IP address that is different from the interface IP addresses of any of the individual routers in the VR.
- Ensure that all routers in the virtual router use VRRP IP Address Adoption (or that none do).
- Use the VRRP IP address to monitor the VR master. Be aware that this does not give information about one particular participating router, but about the current VR master, whichever participating router is acting as the master at the time.
- When changing the configuration of the participating routers using Telnet, GUI or SNMP, configure each device individually by pointing to their individual IP addresses.
- When changing the configuration of the participating routers, do not use the VR IP address. Only one device, the VR master, is responding to this IP address, and you may not know which device it is.
Configuration of VR IP Address Adoption

To configure VRRP IP Address Adoption, use the new parameter, adoptvrip, that has been added to the create vrpp and set vrpp commands:

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

set vrpp=vr-identifier [adoptvrip={on|off}]
   [other-parameters]
```

The adoptvrip parameter specifies that when the switch is acting as the VRRP master it should respond to requests directed at any IP address that it is backing up, even if it does not own that address. If it does not own the address the access requests that the switch will permit are limited to: ICMP echo requests (pings), Telnet, SSH, HTTP and SSL GUI, SNMP and DNS relay. All other types of access to the address will be ignored. The default is OFF.

If you set adoptvrip to on, give the VR an IP address that is different from the interface IP addresses of any of the individual routers in the VR, and only use the VR IP address to monitor the VR, not to configure any of its participating routers. Otherwise you risk confusion when you monitor or configure individual routers.

Configure all the switches in a virtual router with the same values for the VRRP virtual router identifier, IP address, adopt VR IP address mode, advertisement interval, preempt mode, authentication type and password. Inconsistent configuration will cause advertisement packets to be rejected and the virtual router will not perform properly.

To display the value of the new parameter, use the show vrpp command.

Table 7: New parameter displayed in the output of the show vrpp command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adopt VR IP Address(es)</td>
<td>Whether or not the switch should respond to ICMP echo, Telnet, GUI, SNMP and DNS relay service requests targeted at the VR IP address(es) associated with the virtual router, even if it does not own those address(es).</td>
</tr>
</tbody>
</table>
Static IGMP

Static IGMP configures the switch to forward multicast data over specified interfaces and ports. It is an alternative to dynamic IGMP, and is useful for network segments that either have no multicast group members or have hosts that are unable to report group membership with IGMP. A dynamic IGMP configuration will not send multicast traffic to such network segments.

To configure a static IGMP association:

1. Enable IGMP on the switch, using the command:
   ```
   ENABLE IP IGMP
   ```

2. Enable IGMP on the required interface, using the command:
   ```
   ENABLE IP IGMP INTERFACE=interface
   ```

3. Create the static IGMP association, using the command:
   ```
   CREATE IP IGMP DESTINATION=ipaddress INTERFACE=interface
   ```

   The multicast data for the group specified by the `DESTINATION` parameter will be forwarded over the ports specified by the `PORT` parameter. Any of the four octets of the IP address may be replaced by an asterisk (*) to enable wildcard matches. If the `PORT` parameter is not entered, the association will default to all ports belonging to the interface.

To display information about the static IGMP association, use the command:

```
SHOW IP IGMP [COUNTER] [INTERFACE=interface]
   [DESTINATION=ipaddress]
```

To add more ports to an association, use the command:

```
ADD IP IGMP DESTINATION=ipaddress INTERFACE=interface
   PORT={ALL|port-list}
```

Unlike dynamic IGMP group membership information, static IGMP associations never time out. If the network configuration changes, they must be manually modified. To delete ports from an association, use the command:

```
DELETE IP IGMP DESTINATION=ipaddress INTERFACE=interface
   PORT={ALL|port-list}
```

To remove an association from the switch, use the command:

```
DESTROY IP IGMP DESTINATION=ipaddress INTERFACE=interface
```

To enable or disable IGMP debugging of destination and source IP addresses, use the commands:

```
ENABLE IP IGMP DEBUG [DESTINATION={ALL|ipaddress}]
   [SOURCEIPADDRESS={ALL|ipaddress2}]
```

```
DISABLE IP IGMP DEBUG
```

where:

- `ipaddress` is an IGMP group destination address.
- `ipaddress2` is the IP address of a host that responds to IGMP queries.

Debugging is disabled by default. To display which debugging options are set, use the command:

```
SHOW IP IGMP DEBUG
```
IPsec NAT-Traversal

IPsec NAT-T is an enhancement to IPsec and ISAKMP protocols that lets Virtual Private Network (VPN) clients communicate through NAT gateways over the Internet. For example, business travellers commonly use IPsec on their laptops to gain remote VPN access to the central office. When working off-site, these users sometimes need to connect to the Internet through a NAT gateway such as from a hotel. Network Address Translation (NAT) gateways are often part of a company’s firewall and let its Local Area Network (LAN) appear as one IP address to the world. For more information about NAT gateways, refer to RFC 1631 and to the Network Address Translation section in the Internet Protocol chapter of the Reference Manual.

Problems arise with NAT gateways for a number of reasons. A key one is that when they handle IPsec packets, they cannot access encrypted UDP or TCP headers. Therefore, NAT gateways cannot identify traffic for different private devices and cannot properly track individual sessions.

NAT-T is not on the NAT gateway and is not an “IPsec pass-through”. NAT-T lets IPsec/ISAKMP peers send traffic through NAT gateways by putting packets inside UDP packets. This solution enables remote VPN users to communicate successfully when NAT gateways are part of the connection.

Basic NAT-T Operations

Using NAT-D (discovery) messages, NAT-T negotiates with a peer to determine if NAT gateways are present and at which end of the network. Each peer sends at least two NAT-D messages as part of the ISAKMP phase 1 negotiation. The first message contains a hash of a destination IP address; subsequent messages contain source addresses. A NAT gateway is detected when address messages from the peer have incorrect hash values, which indicates that a NAT gateway changed IP addresses.

Also during phase 1, NAT-T determines whether a peer has NAT-T capabilities by detecting a vendor ID. Vendor IDs tell what version of NAT-T the peer supports. When a NAT gateway is not detected or a peer does not support NAT-T, normal IPsec negotiations and protection occur.

When an ISAKMP initiator detects a NAT gateway during an exchange, communication changes from UDP port 500 to port 4500. Log messages inform users that the UDP port has changed. Main or Aggressive mode packets received on the old port are discarded and a separate log is created.

Because IPsec traffic can also be received on port 4500, ISAKMP adds and removes the non-ESP marker at the start of the ISAKMP message so that messages can be detected and passed to the ISAKMP module. ISAKMP drops packets when it receives them on port 4500 without a non-ESP marker.

NAT-T inserts a UDP header between the outer IP and ESP headers thereby encapsulating the ESP data (Figure 1). NAT-T encapsulates IPsec traffic only when a NAT gateway is detected.

Figure 1: UDP Encapsulation for NAT-T

<table>
<thead>
<tr>
<th>IP Header</th>
<th>New UDP Header</th>
<th>ESP Header</th>
<th>Encrypted Data</th>
</tr>
</thead>
</table>
IPsec intercepts UDP-encapsulated ESP packets before they are passed to UDP.

A peer behind a NAT gateway sends keepalive messages to ensure that port mappings in the device remain active between peers. Keepalive intervals are not configurable. The purpose of keepalive messages is different from heartbeat messages controlled by the heartbeatmode parameter in ISAKMP Policy commands, which detect an IKE peer. IKE heartbeat messages and NAT-T keepalive messages do not affect each other.

**NAT-T on the Switch**

This NAT-T implementation supports interoperability with the following VPN clients:

- SafeNet SoftRemote®
- Microsoft Windows 2000®
- Microsoft Windows XP®

NAT-T can also be implemented router-to-router for offices with their own IPsec router behind a NAT gateway. For router-to-VPN examples, see Configuration Notes for interoperability with SafeNet SoftRemote clients, and Microsoft Windows 2000 and XP VPN clients at [www.alliedtelesyn.co.nz/solutions/solutions.html](http://www.alliedtelesyn.co.nz/solutions/solutions.html).

Versions 02 and 08 of the following NAT-T IETF drafts have been implemented:

- **Negotiation of NAT-Traversal in the IKE**, draft-ietf-ipsec-nat-t-ike-02, which describes the modifications to IKE to support NAT detection and UDP tunnel negotiation
- **UDP Encapsulation of IPsec Packets**, draft-ietf-ipsec-udp-encaps-02, which defines the method of UDP encapsulation of IPsec packets
- **Negotiation of NAT-Traversal in the IKE**, draft-ietf-ipsec-nat-t-ike-08, which describes the modifications to IKE to support NAT detection and UDP tunnel negotiation
- **UDP Encapsulation of IPsec Packets**, draft-ietf-ipsec-udp-encaps-08, which defines the method of UDP encapsulation of IPsec packets

Figure 2: IPsec NAT-T peers negotiate traffic through a NAT gateway device
NAT-T is enabled by default, and is enabled or disabled in the ISAKMP policy. We recommend that users carefully read security considerations in the IETF drafts to fully understand the implications of using NAT-T. When users create an ISAKMP policy with the `create isakmp policy` command, they define how peers respond during an ISAKMP exchange, and can use the `nattraversal` parameter to disable NAT-T if they prefer. They can later use the same parameter with the `set isakmp policy` command to enable NAT-T again.

Users should configure an IPsec policy to allow ISAKMP traffic on UDP port 4500 so that it flows through the IPsec layer to ISAKMP. Refer to the ISAKMP policy commands in this chapter to change or add a policy.

Peers send their original, untranslated addresses to each other, which they store in the ISAKMP SA. Recipients use original addresses (OAs) to correct the checksums in the UDP or TCP headers in the IPsec payload.

NAT-T is implemented for IPv4 for both transport and tunnel modes. We recommend transport mode for MS clients. For SafeNet and router-to-router connections, we recommend tunnel mode and specifying unique IP addresses for remote peers. Refer to the `set ipsec saspecification` command to set modes.

**Commands Modified for NAT-T**

The following commands have been modified to include parameters for NAT-Traversal:

- `create isakmp policy`
- `set isakmp policy`
- `show ipsec sa`
- `show ipsec counter=main`
- `show isakmp policy`
- `show isakmp sa`
- `show isakmp counter=general`
- `show isakmp counter=aggressive`
- `show isakmp counter=main`
- `show isakmp counter=network`
- `show isakmp counter=quick`
ISAKMP policy commands

The existing `create isakmp policy` and `set isakmp policy` commands have been modified to add a `nattraversal` parameter. This parameter enables or disables NAT-T to let peers negotiate a UDP-encapsulated mode so that IPsec traffic can flow through a NAT gateway. The default is enabled.

**Syntax**

```
CREATE ISAKMP POLICY= name PEer={ipv4add|ipv6add|ANY} 
    [NATTraversal={ON|OFF|TRUE|FALSE}] 
    [other-isakmp-parameters]

SET ISAKMP POLICY= name [NATTraversal={ON|OFF|TRUE|FALSE}] 
    [other-isakmp-parameters]
```

**show ipsec sa**

Output for the existing `show ipsec sa` command has been modified to include the following parameters that are specific to NAT-T.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA id</td>
<td>The identification number for the SA.</td>
</tr>
<tr>
<td>Role</td>
<td>Whether this peer acted as the initiator or responder in order to create this SA.</td>
</tr>
<tr>
<td>Mode</td>
<td>The IPsec operational mode for this SA: TUNNEL, TRANSPORT, UDP_ENCAPSULATED_TUNNEL, UDP_ENCAPSULATED_TRANSPORT</td>
</tr>
<tr>
<td>NAT-Traversal NAT-OA</td>
<td>Information about original IP addresses.</td>
</tr>
<tr>
<td>Peer original source IP address</td>
<td>Source IP address that the remote peer uses when sending packets to this peer. UDP-encapsulated transport mode only.</td>
</tr>
<tr>
<td>Peer original destination IP address</td>
<td>Destination IP address that the remote peer uses when sending packets to this peer. UDP-encapsulated transport mode, and IETF draft v08, Negotiation of NAT-T in the IKE.</td>
</tr>
<tr>
<td>Filters</td>
<td>Information about the packet selections for this SA.</td>
</tr>
<tr>
<td>NAPT remote port number</td>
<td>Network Address Port Translation number. Multiple clients in UDP-encapsulated transport mode appear to come from the same source. Therefore, NAT-T changes the source port to this value to maintain a distinction.</td>
</tr>
</tbody>
</table>

**show ipsec counter=main**

Output for the existing `show ipsec counter=main` command has been modified to include the following parameter that is specific to NAT-T.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPsec main packet processing counters</td>
<td>The number of NAT keepalive packets received.</td>
</tr>
</tbody>
</table>
**show isakmp policy**

Output for the existing `show isakmp policy` command has been modified to include the following parameter that is specific to NAT-T.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT Traversal</td>
<td>Whether NAT-T is enabled or disabled.</td>
</tr>
</tbody>
</table>

**show isakmp sa**

Output for the existing `show isakmp sa` command has been modified to include the following parameters that are specific to NAT-T.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT-Traversal Information</td>
<td>Information about NAT-T capability.</td>
</tr>
<tr>
<td>NAT-T enabled</td>
<td>Whether NAT-T is enabled on the switch.</td>
</tr>
<tr>
<td>Peer NAT-T capable</td>
<td>Whether the remote peer sent a valid NAT-T vendor ID.</td>
</tr>
<tr>
<td>NAT discovered</td>
<td>Whether a NAT gateway has been detected between peers:</td>
</tr>
<tr>
<td></td>
<td>- Not detected</td>
</tr>
<tr>
<td></td>
<td>- Detected at the remote site</td>
</tr>
<tr>
<td></td>
<td>- Detected at this site</td>
</tr>
<tr>
<td></td>
<td>- Detected at local and remote sites</td>
</tr>
<tr>
<td></td>
<td>- Peer is not NAT-T capable or the NAT discovery process was incomplete</td>
</tr>
</tbody>
</table>

**show isakmp counter**

Output for the existing `show ipsec counter` command has been modified to include NAT-T output for the following parameters:

- general
- aggressive
- main
- network
- quick

Table 1: NAT-T parameters in the output of the `show isakmp counter=general` command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>msgRxBadPortIpChange</td>
<td>The number of ISAKMP packets received with an unexpected source IP address or source port and discarded.</td>
</tr>
<tr>
<td>msgRxFailOldPort</td>
<td>The number of ISAKMP packets discarded because they were received on port 500 after NAT-T had moved the ISAKMP traffic to port 4500.</td>
</tr>
</tbody>
</table>
Table 2: NAT-T parameters in the output of the `show isakmp counter=aggressive` command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>initSendNatD</td>
<td>The number of NAT-D messages sent by the initiator.</td>
</tr>
<tr>
<td>respSendNatD</td>
<td>The number of NAT-D messages sent by the responder.</td>
</tr>
<tr>
<td>initRecvNatD</td>
<td>The number of NAT-D messages received by the initiator.</td>
</tr>
<tr>
<td>respRecvNatD</td>
<td>The number of NAT-D messages received by the responder.</td>
</tr>
</tbody>
</table>

Table 3: NAT-T parameters in the output of the `show isakmp counter=main` command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>initSendNatD</td>
<td>The number of NAT-D messages sent by the initiator.</td>
</tr>
<tr>
<td>initRecvNatD</td>
<td>The number of NAT-D messages received by the initiator.</td>
</tr>
<tr>
<td>respSendNatD</td>
<td>The number of NAT-D messages sent by the responder.</td>
</tr>
<tr>
<td>respRecvNatD</td>
<td>The number of NAT-D messages received by the responder.</td>
</tr>
</tbody>
</table>

Table 4: NAT-T parameter in the output of the `show isakmp counter=network` command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>rxFailNoNonEspMarker</td>
<td>The number of packets ISAKMP dropped because they were received on NAT-T port 4500 without a non-ESP marker.</td>
</tr>
</tbody>
</table>

Table 5: NAT-T parameters in the output of the `show isakmp counter=quick` command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>natOaSent</td>
<td>The number of NAT originating messages sent.</td>
</tr>
<tr>
<td>natOaReceived</td>
<td>The number of NAT originating messages received.</td>
</tr>
<tr>
<td>badNatOa</td>
<td>The number of non-conforming NAT-OA (originating address) messages received such as unknown type.</td>
</tr>
</tbody>
</table>
Link Aggregation Control Protocol (LACP)

Link Aggregation Control Protocol (LACP) .......................................................... 2
Aggregation Criteria .......................................................................................... 2
Aggregated Group Identification ..................................................................... 2
New Commands ............................................................................................... 3
  ADD LACP PORT ...................................................................................... 3
  DELETE LACP PORT .............................................................................. 5
  DISABLE LACP ..................................................................................... 6
  DISABLE LACP DEBUG ........................................................................... 6
  ENABLE LACP ....................................................................................... 6
  ENABLE LACP DEBUG .......................................................................... 7
  PURGE LACP ....................................................................................... 7
  SET LACP PRIORITY ............................................................................. 8
  SET LACP PORT ................................................................................... 9
  RESET LACP PORT COUNTER .......................................................... 10
  SHOW LACP ....................................................................................... 11
  SHOW LACP PORT ............................................................................. 12
  SHOW LACP PORT COUNTER .......................................................... 15
  SHOW LACP TRUNK ............................................................................ 16
Modified Commands ...................................................................................... 17
  ACTIVATE SWITCH PORT ............................................................... 17
Link Aggregation Control Protocol (LACP)

The implementation of the link access control protocol (LACP) follows the IEEE 802.3-2002 Standard, “CSMA/CD access method and physical layer specifications.”

The LACP protocol operates where systems are connected over multiple communications links. In this configuration, links that are controlled by LACP are constantly monitored and are automatically added to, or removed from, trunk groups (or aggregated links).

Once LACP has been initially configured and enabled, it will automatically create aggregated link groups (also termed trunk groups) and assign appropriate links to their membership. LACP will continue to monitor these groups and dynamically add, or remove links to them as network changes occur.

LACP achieves this by performing the following tasks:

■ determines which ports are under LACP control
■ determines whether each port is in LACP Active or LACP Passive mode
■ determines which system has the highest LACP priority
■ determines the LACP priority of ports
■ determines whether the periodic timeout is fast or slow

Aggregation Criteria

For individual links to be formed into an aggregated group they must meet the following criteria:

■ originate on the same device
■ terminate on the same device
■ be members of the same VLANs
■ have the same data rate
■ share the same admin port key (assigned by using the ADD LACP PORT ADMINKEY command)

The hardware must also be capable of handling the number of links to be aggregated.

Aggregated Group Identification

In order to identify particular aggregated groups, each group is assigned a link aggregation identifier called a LAG ID. The LAG ID comprises the following components for both the local system (called the Actor) followed by their equivalent components for the remote system (called the Partner):

■ System Priority - set by the SET LACP PRIORITY command
■ System Identifier - the MAC address of the system
■ Port Key - An identifier - created by the LACP software
■ Port Priority - set by the ADD LACP PORT PRIORITY command
- Port Number - determined by the device connection

The LAG ID can be displayed for each aggregated link by entering the SHOW LACP TRUNK command.

**New Commands**

This section describes the new commands available to configure and manage LACP on the switch.

**ADD LACP PORT**

**Syntax**

```
ADD LACP PORT=[{port-list|ALL}] [ADMINKEY=key] [PRIORITY=priority] [MODE={ACTIVE|PASSIVE}] [PERIODIC={FAST|SLOW}]
```

where:
- **port-list** is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered port, including uplink ports.
- **key** is an integer from 0 to 65535
- **priority** is an integer from 0 to 65535

**Description**

This command adds a port to LACP’s control thus enabling LACP to put it into an aggregated link. By default, ports are added in the active mode. If a port is added in the active mode, and its link’s requirements for trunking are met, then the port and it associated link will be automatically aggregated without further configuration. The same situation will apply for a port configured in the “passive” mode, but whose link connects to a remote port that is configured in the “active” mode.

The PORT parameter specifies the ports whose parameters are to be modified. Where none of the ports specified are presently managed by LACP, the command will only take effect if it can be applied to all the specified ports. Where some of the ports specified are already managed by LACP, and additional ports are added, by using the ALL parameter for example, then the LACP managed ports will have their Key and other parameters changed and the command will succeed on all the specified ports.

In the following descriptions, references to an individual port should be taken as a reference to all ports selected by the PORT parameter.

The ADMINKEY parameter specifies the Admin LACP port key. This affects the LACP Port Key generated (but does not determine its value) and can be used to prevent otherwise aggregatable ports from forming a trunk. By default all ports will be given the same LACP ADMINKEY, which in turn means that all ports that can be aggregated will generate the same LACP port key.
The PRIORITY parameter specifies the LACP port priority. This priority assigned is used where the number of physical links connecting two devices is greater than the number that can be aggregated. The priority entered will then be used to determine which ports are selected for aggregation. The default value of 32,768 (0 being the highest priority) will be applied to all ports.

Where the port priority is the same, the port number governs which ports are selected (low port number being high priority). Excess ports will be put into a ‘standby’ mode, which is effectively disabled, but they will take the place of a link in a trunk that goes down.

The MODE parameter specifies whether the port will run in LACP passive or active mode. A port in passive mode will only begin sending out LACPDUs in response to a received LACPDU, whereas a port in active mode will always send LACPDUs at regular intervals (specified by the PERIODIC parameter).

The PERIODIC parameter specifies the requested rate that the LACP port will receive LACPDU update messages from its partner port. A port in fast mode will receive one LACPDU every second; a port in slow mode will receive one LACPDU every thirty seconds.

**Examples**

To add ports 3 and 5 to LACP, use the command:

```
ADD LACP PORT=3,5
```

**See Also**

DELETE LACP PORT
SET LACP PORT
SHOW LACP PORT
DELETE LACP PORT

**Syntax**

DELETE LACP PORT={port-list}

where:

- *port-list* is a port number, a range of port numbers (specified as n-m), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered switch port, including uplink ports.

**Description**

This command removes ports from LACP’s control and LACP frames will no longer be transmitted across the link. It is good practice to delete LACP from ports that are linked to a non LACP capable devices.

The PORT parameter specifies the switch ports to be deleted from LACP’s control. Ports specified must be under the control of LACP. ALL is not a configurable option. To stop LACP working on all ports refer to the DISABLE LACP command.

**Examples**

To delete ports 3 and 5 from LACP, use the command:

DELETE LACP PORT=3,5

**See Also**

ADD LACP PORT
DISABLE LACP
SET LACP PORT
SHOW LACP PORT
DISABLE LACP

Syntax  
DISABLE LACP

Description  
This command disables the LACP processes on the switch. A warning message, a notification message, and a log message are generated when this command is executed. LACP is disabled by default. Port settings that are changed whilst LACP is disabled, will take effect when LACP is re-enabled.

See Also  
SET SWITCH PORT
SHOW LACP

DISABLE LACP DEBUG

Syntax  
DISABLE LACP DEBUG=[MSG|PACKET|STATE|TRACE|ALL]

Description  
This command disables the LACP debugging process. The MSG option displays the decoded form of all incoming and outgoing LACP packets. The PACKET option displays all incoming and outgoing LACP packets in hex. The STATE option displays any internal state machine changes. The TRACE option displays the function call tree. The LACP debugging facility is disabled by default.

See Also  
ENABLE LACP DEBUG
SHOW LACP

ENABLE LACP

Syntax  
ENABLE LACP

Description  
This command enables the LACP process on the switch. A notification message and a log message file are generated when this command is executed. LACP is disabled by default.

See Also  
DISABLE LACP
SHOW LACP
RESET LACP PORT COUNTER
ENABLE LACP DEBUG

Syntax
ENABLE LACP DEBUG= [PACKET | STATE | ALL]

Description
This command enables the LACP debugging facility. The PACKET option displays all incoming and outgoing LACP packets. The STATE option displays any internal state machine changes. The LACP debugging facility is disabled by default.

See Also
DISABLE LACP DEBUG
SHOW LACP

PURGE LACP

Syntax
PURGE LACP

Description
This command destroys all LACP configuration and restores the default values to all the configurable parameters. The debug parameters for all ports are reset to their default values. This command returns the LACP module to the status that exists when first powered on.

See Also
SET SWITCH PORT
SHOW LACP
SET LACP PRIORITY

**Syntax**

```
SET LACP PRIORITY=priority
```

where:

- priority is an integer from 0-65535

**Description**

This command modifies the relative priority of LACP enabled partners.

The SYSTEM PRIORITY parameter specifies a numeric value that is used as part of the system priority calculation. When systems having multiple links connect and use LACP to control link aggregation, each system will compare its system priority data identifiers in order to determine which system will control the links. A system identifier comprises a system priority component (configured by this parameter) followed by the system’s MAC address. Link control will be assigned to the system with the numerically lower system priority data identifier. The default is 32768.

**Examples**

System A is to connect to system B using LACP and System B is to control their aggregated links.

System A has a MAC address of 00-00-cd-00-0d-42 and has been assigned an LACP SYSTEM PRIORITY value of 500. System B has a MAC address of 00-00-cd-00-0d-52.

In order to ensure that System B controls the links, its LACP SYSTEM PRIORITY must be set to a lower value than 500. The LACP SYSTEM PRIORITY on System B will therefore be set to 300. Note that the system control will be determined by the values set by the LACP System Priority values, because these have a greater numeric significance than the MAC Addresses.

```
SET LACP SYSTEM PRIORITY=300
```

**See Also**

SHOW LACP
SET LACP PORT

Syntax

```
SET LACP PORT=[{port-list|ALL}] [ADMINKEY=key number]
[PRIORITY=priority] [MODE={ACTIVE|PASSIVE}]
[PERIODIC={FAST|SLOW}]
```

where:

- `port-list` is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered Ethernet switch port, including uplink ports.
- key number is an integer from 0-65535
- priority is an integer from 0-65535

Description

This command modifies the value of parameters for LACP ports.

The PORT parameter specifies the ports for which parameters are modified. If the command would succeed on a subset of the ports specified, but cause an error on the others, then the command as a whole will fail and have no effect. Reference in the descriptions below to an individual port should be taken as a reference to all ports selected by the PORT parameter.

The ADMINKEY parameter specifies the Admin LACP port key. This affects (but does not set the value of) the internal system port key allocated. The ADMINKEY can be used to prevent otherwise aggregatable ports from aggregating together. By default all ports that can be aggregated will be given the same LACP port key.

The PRIORITY parameter specifies the LACP port priority. This value is used to decide which ports should be selected when being added to a trunk group (where there are more links existing between the two devices than the switch is able to aggregate). The default value is one. This means that port number governs which ports are selected (low port number equals high priority). Excess ports will be put into a ‘standby’ mode. In this mode they remain untrunked, but still able replace a link that goes down.

The MODE parameter specifies whether the port will run in LACP passive or active mode. A port in passive mode will only send out an LACPDU in response to a received LACPDU, whereas a port in active mode will send a LACPDU at a regular interval (specified by the PERIODIC parameter).

The PERIODIC parameter specifies the rate at which the LACP port transmits updates. A port in fast mode transmit one LACPDU every second, a port in slow mode transmits one LACPDU every thirty seconds.

See Also

DELETE LACP PORT
ADD LACP PORT
SHOW LACP PORT
RESET LACP PORT COUNTER

Syntax
RESET LACP PORT[={port-list|ALL}] COUNTER

where:
- port-list is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered Ethernet switch port, including uplink ports.

Description
This command resets all LACP counters for the specified switch ports.

Examples
To reset the LACP counters for all ports, use the command:

```
RESET LACP PORT COUNTER
```

See Also
PURGE LACP
SHOW LACP
SHOW LACP PORT COUNTER
SHOW LACP

Syntax
SHOW LACP

Description
This command displays the state of LACP on the Switch.

Figure 2-1: Example output from the ENABLE SWITCH BIST command.

<table>
<thead>
<tr>
<th>LACP Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status .................. Enabled</td>
</tr>
<tr>
<td>Actor System Priority ....... 80-00</td>
</tr>
<tr>
<td>Actor System ................. 00-3e-0a-12-00-01</td>
</tr>
<tr>
<td>LACP Ports .................. 1-3,5,7,9-12</td>
</tr>
<tr>
<td>Active .................. 1-3,5</td>
</tr>
<tr>
<td>Passive ................ 7,9-12</td>
</tr>
</tbody>
</table>

Table 2-1: Parameters displayed in the output of the SHOW LACP command.

<table>
<thead>
<tr>
<th>Debug Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Shows the current state of LACP on the device. Can be Enabled or Disabled.</td>
</tr>
<tr>
<td>Priority</td>
<td>The user configurable priority of the system. This parameter is concatenated with the Actor System parameter to generate the Actor System ID.</td>
</tr>
<tr>
<td>Actor System</td>
<td>The MAC address of the local system.</td>
</tr>
<tr>
<td>LACP Ports</td>
<td>A list of the ports currently under LACP control</td>
</tr>
<tr>
<td>Active</td>
<td>A list of the ports currently in LACP Active mode</td>
</tr>
<tr>
<td>Passive</td>
<td>A list of the ports currently in LACP Passive mode</td>
</tr>
</tbody>
</table>
SHOW LACP PORT

Syntax

SHOW LACP PORT

where:
- port-list is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered Ethernet switch port, including uplink ports.

Description

This command displays LACP information about the specified switch ports or all switch ports. For example output and parameter descriptions see the example below.

Figure 2-2: Example output from the SHOW LACP PORT command.

<table>
<thead>
<tr>
<th>LACP Port Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor Port ............ 1</td>
</tr>
<tr>
<td>Trunk Group ........... lacp1</td>
</tr>
<tr>
<td>Selected .............. Selected</td>
</tr>
<tr>
<td>Port Priority .......... 8000</td>
</tr>
<tr>
<td>LACP Port Number ....... 0001</td>
</tr>
<tr>
<td>Port Key ............... 6</td>
</tr>
<tr>
<td>Admin Key .............. 12</td>
</tr>
<tr>
<td>Mode .................. Active</td>
</tr>
<tr>
<td>Periodic ............... Fast</td>
</tr>
<tr>
<td>Individual ............ No</td>
</tr>
<tr>
<td>Synchronised .......... Yes</td>
</tr>
<tr>
<td>Collecting ............ Yes</td>
</tr>
<tr>
<td>Distributing ........... Yes</td>
</tr>
<tr>
<td>Defaulted ............. No</td>
</tr>
<tr>
<td>Expired ............... No</td>
</tr>
<tr>
<td>Actor Churn .......... No</td>
</tr>
<tr>
<td>Partner Churn .......... No</td>
</tr>
</tbody>
</table>

Table 2-2: Parameters displayed in the output of the SHOW LACP PORT command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>The number of the port.</td>
</tr>
<tr>
<td>Trunk Group</td>
<td>The name of trunk group to which the port belongs. It is a name that LACP has automatically assigned to an aggregated link. You cannot manually create a trunk starting with the letters LACP. If LACP created, then the name will have the prefix LACP followed by a numeric i.e. LACP72. Note that this number is the same as the new interface index shown by the SHOW INTERFACE command.</td>
</tr>
<tr>
<td>Priority</td>
<td>The user configurable priority assigned to the port.</td>
</tr>
<tr>
<td>LACP Port Number</td>
<td>The LACP encoded port number.</td>
</tr>
<tr>
<td>Port Key</td>
<td>The key that LACP has assigned to the port.</td>
</tr>
<tr>
<td>Admin Key</td>
<td>The user configurable key assigned to the port.</td>
</tr>
</tbody>
</table>
Table 2-2: Parameters displayed in the output of the SHOW LACP PORT command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>The participation mode. If &quot;Active&quot;, the port sends LACPDU packets regardless of the Partner Port's participation. If &quot;Passive&quot;, the port will only send LACPDU packets after receiving one from its Partner Port.</td>
</tr>
<tr>
<td>Timeout</td>
<td>The user configurable time period between transmission of periodic LACPDU packets; one of &quot;Fast&quot; (1 second) or &quot;Slow&quot; (30 seconds).</td>
</tr>
<tr>
<td>Individual</td>
<td>The user configurable setting determining whether the port is an individual. If &quot;No&quot;, the port is not an individual and may be aggregated. If &quot;Yes&quot;, it is an individual and will not be aggregated.</td>
</tr>
<tr>
<td>Synchronised</td>
<td>If &quot;Yes&quot;, the port is considered to be in a Synchronised State, that is, the port has been correctly associated with an Aggregator; otherwise &quot;No&quot;.</td>
</tr>
<tr>
<td>Collecting</td>
<td>If &quot;Yes&quot;, this Port has been enabled for receiving packets; otherwise &quot;No&quot;.</td>
</tr>
<tr>
<td>Distributing</td>
<td>If &quot;Yes&quot;, this Port has been enabled for transmitting packets; otherwise &quot;No&quot;.</td>
</tr>
<tr>
<td>Defaulted</td>
<td>Whether or not this system is using the default values for the Partner Information. If &quot;No&quot;, the values have been received from the Partner via a LACPDU. If &quot;Yes&quot;, the default values are still in use.</td>
</tr>
<tr>
<td>Expired</td>
<td>The port has not received a frame from its partner for 3 times the periodic time (3 or 90 seconds).</td>
</tr>
<tr>
<td>Partner Information</td>
<td>The information that has been received about the Partner Port. The partner port is the port on the connected device.</td>
</tr>
<tr>
<td>Partner System Priority</td>
<td>The Partner's System Priority.</td>
</tr>
<tr>
<td>Partner System</td>
<td>The Partner’s System identifier.</td>
</tr>
<tr>
<td>Port Key</td>
<td>The Partner Port’s key.</td>
</tr>
<tr>
<td>Port Priority</td>
<td>The Partner Port’s key priority.</td>
</tr>
<tr>
<td>Port Number</td>
<td>The Partner Port’s port number.</td>
</tr>
<tr>
<td>Mode</td>
<td>One of &quot;Active&quot; or &quot;Passive&quot;. If &quot;Active&quot;, the Partner Port sends LACPDU packets regardless of this port’s participation. If &quot;Passive&quot;, the Partner Port will only send LACPDU packets after receiving one from this port.</td>
</tr>
<tr>
<td>Periodic</td>
<td>The setting of the Partner Port for the time period between transmission of periodic LACPDU packets; one of &quot;Fast&quot; (1 second) or &quot;Slow&quot; (30 seconds).</td>
</tr>
<tr>
<td>Individual</td>
<td>The setting of the Partner Port determining whether the port is an individual. If &quot;No&quot;, the partner port is not an individual and may be aggregated. If &quot;Yes&quot;, it is an individual and can not be aggregated.</td>
</tr>
<tr>
<td>Synchronised</td>
<td>If &quot;Yes&quot;, the Partner System considers the Partner Port to be in a Synchronised State, that is, the port has been correctly associated with an Aggregator; otherwise &quot;No&quot;.</td>
</tr>
<tr>
<td>Collecting</td>
<td>If &quot;Yes&quot;, the Partner Port has been enabled for receiving packets; otherwise &quot;No&quot;.</td>
</tr>
<tr>
<td>Distributing</td>
<td>If &quot;Yes&quot;, the Partner Port has been enabled for transmitting packets; otherwise &quot;No&quot;.</td>
</tr>
</tbody>
</table>
Table 2-2: Parameters displayed in the output of the SHOW LACP PORT command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defaulted</td>
<td>Whether or not the Partner system is using the default values for this port's information. If &quot;No&quot;, the values have been received from this system via a LACPDU. If &quot;Yes&quot;, the default values are still in use.</td>
</tr>
<tr>
<td>Expired</td>
<td>When the partner port has not received a frame for 3 times the periodic time (3 or 90 seconds).</td>
</tr>
</tbody>
</table>

**Examples**

To show the LACP port information for all ports, use the command:

```
SHOW LACP PORT
```

**See Also**

ADD LACP PORT
SET LACP PORT
SHOW LACP
SHOW LACP PORT COUNTER

**Syntax**

SHOW LACP PORT[={port-list|ALL}] COUNTER

where:

- **port-list** is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered Ethernet switch port, including uplink ports.

**Description**

This command displays LACP counters for the specified switch ports or all switch ports. For example output and parameter descriptions see the example below.

Figure 2-3: Example output from the SHOW LACP PORT command.

<table>
<thead>
<tr>
<th>LACP Port Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port 1:</td>
</tr>
<tr>
<td>Received:</td>
</tr>
<tr>
<td>LACP Pkts</td>
</tr>
<tr>
<td>........................ 0</td>
</tr>
<tr>
<td>Invalid LACP Pkts</td>
</tr>
</tbody>
</table>

Table 2-3:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Received</td>
<td>Counters for LACP frames received.</td>
</tr>
<tr>
<td>LACP Pkts</td>
<td>The number of valid LACPDU frames received.</td>
</tr>
<tr>
<td>Invalid LACP Pkts</td>
<td>The number of invalid LACP frames received. This includes frames with an invalid type/length field, subtype field, actor information length field, partner information length field, collector information length field, terminator information length field, or invalid frame length</td>
</tr>
<tr>
<td>Transmitted</td>
<td>Counters for LACP frames transmitted.</td>
</tr>
<tr>
<td>LACP Pkts</td>
<td>The number of LACPDU frames transmitted.</td>
</tr>
</tbody>
</table>

**Examples**

To show the LACP counters for all ports, use the command:

```
SHOW LACP PORT COUNTER
```

**See Also**

RESET LACP PORT COUNTER  
SHOW LACP  
SHOW LACP PORT
SHOW LACP TRUNK

Syntax

SHOW LACP TRUNK

where:

- port-list is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered Ethernet switch port, including uplink ports.

Description

This command displays the currently dynamically configured trunks for the LACP module.

Figure 2-4: Example output from the SHOW LACP TRUNK command.

```
LACP Dynamic Trunk Group Information
---------------------------------------------------------------------------
Trunk group name .......... lacp53:
   Speed ................... 100 Mbps
   Ports in Trunk .......... 10,15
   LAG ID:
     [(8000,00-00-cd-03-00-79,0005,00,0000),(8000,00-00-cd-08-76-60,0002,00,0000)]
---------------------------------------------------------------------------
```

See Also

SHOW SWITCH TRUNK
SHOW LACP
Modified Commands

This section describes modifications to existing commands to configure and manage LACP on the switch.

ACTIVATE SWITCH PORT

Syntax

ACTIVATE SWITCH PORT=(port-list|ALL) [AUTONEGOTIATE] [LOCK]

where:

- port-list is a port number, a range of port numbers (specified as n-m), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at m, where m is the highest numbered Ethernet switch port, including uplink ports.

Description

This command activates autonegotiation of port speed and duplex mode for a port or a group of ports.

The PORT parameter specifies the port for which autonegotiation is to be activated. Only ports in the list that are set to autonegotiate are affected by this command. Ports are not modified that have a fixed speed setting or that belong to a trunk group. A port that has been added to LACP will autonegotiate until it actively becomes part of an aggregated link (i.e. trunked) at which point it will operate at the speed of the link.

A port that has been added to LACP will autonegotiate until it actively becomes part of an aggregated link (i.e. trunked) at which point it will operate at the speed of the aggregated link.

The AUTONEGOTIATE parameter specifies that the port is to activate the autonegotiation process. The port begins to autonegotiate link speed and duplex mode.

The LOCK parameter is used to manually lock the switch port before it reaches its learning limit so that no new addresses are automatically learned. The LEARN parameter for the port is set to the current number of learned MAC addresses.

Examples

To activate autonegotiation on ports 1-8 and port 10, use the command:

ACTIVATE SWITCH PORT=1-8,10 AUTONEGOTIATE

Related Commands

SET SWITCH PORT
SHOW SWITCH PORT
Local Interfaces

Introduction ...................................................................................................... 2
New Commands ............................................................................................... 3
  ADD IP LOCAL ............................................................................................ 3
  DELETE IP LOCAL ...................................................................................... 4
  SET SNMP LOCAL .................................................................................... 5
Modified Commands ........................................................................................ 6
  SET IP LOCAL ............................................................................................. 6
  SHOW IP INTERFACE ................................................................................ 7
  ADD BGP PEER .......................................................................................... 8
  SET BGP PEER ........................................................................................ 9
  SHOW BGP PEER ....................................................................................... 9
  SET BGP .................................................................................................. 12
  CREATE LOG OUTPUT ............................................................................. 12
  SET LOG OUTPUT .................................................................................... 13
  SHOW LOG OUTPUT ................................................................................. 13
  ADD RADIUS SERVER ............................................................................ 15
  SHOW RADIUS ....................................................................................... 16
  SHOW SNMP .............................................................................................. 17
  ADD PIM BSRCANDIDATE ...................................................................... 18
  ADD PIM RPCANDIDATE .......................................................................... 18
  SET PIM RPCANDIDATE ........................................................................... 18
  SHOW PIM RPCANDIDATE ....................................................................... 18
  ADD TACPLUS SERVER ............................................................................ 19
  SET TACPLUS SERVER ............................................................................ 19
  SHOW TACPLUS SERVER .......................................................................... 20
  SHOW TACPLUS SERVER .......................................................................... 20
Introduction

A Loopback interface is one that is always available for higher layer protocols to use and advertise out into a network. A local interface is assigned an IP address, but does not have the usual requirement of needing to be connected to a lower layer physical entity. It is the lack of an attachment to a physical entity that allows the perception of it always being accessible via the network.

Loopback interfaces can be utilised by a number of protocols for various purposes. They can be used to improve access to a device, as well as increasing the reliability, security, scalability and protection offered by the device. In addition they can add flexibility and simplify management, information gathering and filtering.

Loopback interfaces can help improve reliability. For example, having OSPF advertise a Loopback Interface as a interface-route, out into the network irrespective of what other physical links are up or down at the time - ensuring a higher probability of routing traffic being received and subsequently forwarded. Additionally, reliability and performance can be improved if BGP configures parallel paths to a Loopback Interface on a peer device, resulting in the improvement of load sharing.

Access and security can be improved through filtering. Incoming traffic can be filtered by rules that specify Loopback Interfaces as the only acceptable destination addresses.

Information gathering and filtering as well as management can potentially be simplified if protocols such as SNMP use Loopback interfaces for receiving and sending trap and log type information.
New Commands

Use the following new commands to configure Local Interfaces.

ADD IP LOCAL

Syntax

```
SET IP LOCAL=[1..15][FILTER={filter-number}|NONE]
GRE=[0..100 NONE][IPADDRESS=ipadd]
POLICYFILTER={filter-number|NONE}
PRIORITYFILTER={filter-number|NONE}
```

where:

- `filter-number` is a number from 0 to 299.
- `ipadd` is an IP address in dotted decimal notation.

Description

This command adds a local interface to the switch. A switch can only have fifteen local interfaces added. The switch has an additional default local interface, which is added on start up. The default local interface can be configured through the SET IP LOCAL command. A local interface is a virtual interface with no physical interface attachment. The interface can be assigned an IP address that can then be used as the source address of IP packets generated internally by IP protocols such as RIP, OSPF, PING and NTP. Higher layer protocols such as RIP, OSPF, PING and NTP must assign a source IP address to packets passed to IP for forwarding.

The following rules are used to determine which IP address to use as the source address:

1. If the higher layer protocol's configuration specifies either a source IP address or a local interface to use, then the configured address is used as the packet's source IP address. For example, the SIPADDRESS parameter of the PING command specifies the source IP address to use in ping packets. While the LOCAL parameter of the ADD BGP PEER command specifies a local interface to use to obtain a source IP address

2. If the default local interface has been assigned an IP address, then the IP address of the default local interface is used as the packet's source IP address. Otherwise, the IP routing module determines the interface over which the packet is to be transmitted, and assigns the IP address of the interface as the packet's source IP address.

The LOCAL parameter specified is a unique identifying number that is used to identify the interface. The naming convention, or alias, for this interface is the concatenation of the word “local” along this identifying number.

The FILTER parameter specifies the filter to apply to IP packets transmitted or received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface. The default is not to apply a filter.
The GRE parameter specifies the GRE (Generic Routing Encapsulation) entity associated with the interface. The specified GRE entity must have been created previously using the ADD GRE command. The default is NONE. The IPADDRESS parameter specifies the IP address of the interface. The IP address must be the IP address of one of the switch's active IP interfaces. Specifying an IP address of 0.0.0.0 effectively 'unsets' the IP address of the local interface.

The POLICYFILTER parameter specifies the policy filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Policy filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PRIORITYFILTER parameter specifies the priority filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Priority filters are applied to packets as they are transmitted. The default is not to apply a filter.

**Examples**

To add the local interface 3 with an IP address of 192.168.33.1, use:

```
ADD IP LOCAL=3 IP=192.168.33.1
```

**Related Commands**

DELETE IP LOCAL
SET IP LOCAL
SHOW IP INTERFACE

---

**DELETE IP LOCAL**

**Syntax**

`DELETE IP LOCAL=1..15`

**Description**

This command deletes a local interface from the IP module. The local interface will no longer be used by the IP routing module.

*When an IP interface is deleted, any static routes and ARP entries specific to the interface are also deleted*

**Examples**

To delete local interface 5, use:

```
DELETE IP LOCAL=5
```

**Related Commands**

ADD IP LOCAL
SET IP LOCAL
SHOW IP INTERFACE

---
**SET SNMP LOCAL**

**Syntax**  
```plaintext
SET SNMP VERSION={V1|V2|V3|ALL} LOCAL={NONE|1..15}
```

Where  
- **V1, V2 and V3** represent the version of the SNMP packets.

**Description**  
This command sets a local interface to be used with a particular version of SNMP. Once set, the local interface will have its IP address used as the source IP address for all SNMP packets the switch generates and subsequently sends of that SNMP version.

The **VERSION** parameter specifies version of SNMP packets that the local interface will apply to. The default is **ALL**.

The **LOCAL** parameter specifies a local interface to be used as the source IP address for all SNMP packets the switch generates and sends of the specified SNMP version. The local interface IP address will be also be used as the SNMP agent IP address in these outgoing packets. The local interface must already be configured and be in the range 1 to 15. If a local interface is not set for SNMP, the switch will select a source from the current available interfaces.

**Examples**  
To set the local interface local5 for SNMPv3 packets, use the command:
```
SET SNMP VERSION=V3 LOCAL=5
```

**Related Commands**  
SHOW SNMP
Modified Commands

The following commands to are changed to enable Local Interfaces to be configured. Changes have been highlighted.

SET IP LOCAL

This command has been enhanced to enable up to 15 local ports to be modified.

Syntax

```
SET IP LOCAL[=DEFAULT|1..15][FILTER={filter-number|NONE}]
  [GRE={0..100|NONE}] [IPADDRESS=ipadd]
  [POLICYFILTER={filter-number|NONE}]
  [PRIORITYFILTER={filter-number|NONE}]
```

where:
- filter-number is a number from 0 to 299.
- ipadd is an IP address in dotted decimal notation.

Description

This command modifies the parameters of one the switch's local interfaces. If the LOCAL argument is either not specified or is set to DEFAULT, the switch's default local interface is modified.

The FILTER parameter specifies the filter to apply to IP packets transmitted or received over the interface. The filter must already have been defined. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface. The default is not to apply a filter.

The GRE parameter specifies the GRE (Generic Routing Encapsulation) entity associated with the interface. The specified GRE entity must have been created previously using the ADD GRE command. The default is NONE.

The IPADDRESS parameter specifies the IP address of the interface. The IP address must be the IP address of one of the switch's active IP interfaces. Specifying an IP address of 0.0.0.0 effectively 'unsets' the IP address of the local interface.

The POLICYFILTER parameter specifies the policy filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Policy filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PRIORITYFILTER parameter specifies the priority filter to apply to IP packets transmitted over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface.
Priority filters are applied to packets as they are transmitted. The default is not to apply a filter.

**Examples**

To set the IP address of the default local IP interface to 192.168.33.11, use:

```plaintext
SET IP LOCAL IP=192.168.33.11
```

To set the local interface 3 to 192.168.33.11, use:

```plaintext
SET IP LOCAL=3 IP=192.168.33.1
```

To remove the IP address of the default local IP interface, use:

```plaintext
SET IP LOCAL IP=0.0.0.0
```

### SHOW IP INTERFACE

**Syntax**

```
SHOW IP INTERFACE[=interface] [COUNTER[=MULTICAST]]
```

where:

- `interface` is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

**Description**

This command displays interface configuration information for the interfaces that assigned to the IP module using the ADD IP INTERFACE. If an interface is specified, then information for the specified interface is displayed, otherwise information for all IP interfaces is displayed.

A hash symbol (#) after the interface name indicates that the interface has an operational status of “down”. Note that interface routes are propagated by RIP when their status at a physical level is “up”. For VLAN interfaces, this means that the VLAN’s interface route is only propagated if at least one port in e VLAN is active.

The COUNTER parameter displays counters for the specified interface or all interfaces.

---

**Figure 1:** Example output from the SHOW IP INTERFACE command.
ADD BGP PEER

Syntax

ADD BGP PEER=ipadd REMOTEAS=1..65534 [LOCAL={NONE|1..15}] [other-options]

where:

- ipadd is an IP address in dotted decimal notation.

Description

This command adds a BGP peer to the switch. This command adds the peer in the disabled state; the switch will not attempt to begin communicating with the peer until the ENABLE BGP PEER command is entered. This allows full configuration of the peer entry before starting to communicate with it.

The LOCAL parameter specifies a local interface to be used as the source for all BGP packets the switch generates and sends to this BGP peer. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

Examples

To add a BGP peer whose IP address is 192.168.1.1, whose AS number is 54321 and whose local interface is local3, use the command:

```
ADD BGP PEER=192.168.1.1 REMOTEAS=54321 DESCRIPTION="test remote BGP peer" LOCAL=3
```
SET BGP PEER

Syntax

```
SET BGP PEER=ipadd [LOCAL=(NONE|1..15)] [other-options]
```

where:

- `ipadd` is an IP address in dotted decimal notation.

Description

This command modifies parameters for an existing BGP peer in the switch. If the BGP peer is enabled, the BGP peer must be disabled, this command executed, and the BGP peer enabled before the change takes effect. If the BGP peer is disabled, the peer must be enabled before the change takes effect. The PEER parameter specifies the IP address of the peer to be modified. The peer must be an existing BGP peer on this switch.

The LOCAL parameter specifies a local interface to be used as the source for all BGP packets the switch generates and sends to this BGP peer. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

SHOW BGP PEER

Syntax

```
SHOW BGP PEER[=ipadd]
```

where:

- `ipadd` is an IP address in dotted decimal notation.

Description

This command displays summary information about all BGP peers, or detailed information about the specified BGP peer.

The PEER parameter specifies the IP address of the BGP peer about which detailed information is to be displayed. If a value is not specified, summary information is displayed for all BGP peers.

Figure 2: Example summary output from the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>BGP peer entries</th>
<th>Peer</th>
<th>State</th>
<th>AS</th>
<th>InMsg</th>
<th>OutMsg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>192.168.2.254</td>
<td>Estab</td>
<td>12345</td>
<td>23456</td>
<td>8245</td>
</tr>
<tr>
<td></td>
<td>192.168.3.16</td>
<td>Idle (D)</td>
<td>123</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 2: Example summary output from the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer</td>
<td>The IP address of the BGP peer.</td>
</tr>
<tr>
<td>State</td>
<td>The BGP peer state; one of “Idle”, “Idle(D)”, “Connect”, “Active” “OpenSent”, “OpenConf” and “Estab”. “Idle(D)” indicates that the peer is idle and also disabled.</td>
</tr>
<tr>
<td>As</td>
<td>The number of the autonomous system to which this peer belongs.</td>
</tr>
<tr>
<td>InMsg</td>
<td>The number of messages received from this peer since the TCP connection opened.</td>
</tr>
<tr>
<td>OutMsg</td>
<td>The number of messages sent to this peer since the TCP connection opened.</td>
</tr>
</tbody>
</table>

Figure 3: Example detailed output from the SHOW BGP PEER command.

```
Peer ................ 192.168.2.254
Peer ................ 192.168.2.254
Description ......... Sprint's AS 12345
State ............... Established
Remote AS ........... 12345
Connect Retry ...... 200s
Hold time .......... 90s (actual 0s)
Keep alive .......... 30s (actual 0s - no KEEPALIVES)
Min AS originated ... 20s
Min route advert .... 40s
Filtering
In filter .......... -
In path filter ..... -
In route map ...... -
Out filter .......... 334
Out path filter ... -
Out route map ..... -
Max prefix .......... 2000 (action is WARNING)
External hops ...... 5 (EBGP multihop enabled)
Next hop self ...... No
Send community ...... No
Messages In/Out ..... 23456/3245
Debugging .......... -
Device ............. -
Local Interface ..... local4
```
### Table 3: Parameters displayed in the detailed output of the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer</td>
<td>The IP address of the BGP peer.</td>
</tr>
<tr>
<td>Description</td>
<td>The description set for this BGP peer.</td>
</tr>
<tr>
<td>State</td>
<td>The BGP peer state; one of “Idle”, “Idle(D)”, “Connect”, “Active”,</td>
</tr>
<tr>
<td></td>
<td>“OpenSent”, “OpenConfirm” and “Established”. “Idle(D)” indicates that</td>
</tr>
<tr>
<td></td>
<td>the peer is idle and also disabled.</td>
</tr>
<tr>
<td>Remote AS</td>
<td>The number of the autonomous system to which this peer belongs.</td>
</tr>
<tr>
<td>Connect Retry</td>
<td>The time interval for retrying the initial TCP connection to this peer</td>
</tr>
<tr>
<td></td>
<td>in the event of a connection failure.</td>
</tr>
<tr>
<td>Hold Time</td>
<td>The configured and actual hold times for this peer. The actual hold</td>
</tr>
<tr>
<td></td>
<td>time is the lower of the configured hold times of the peer and this</td>
</tr>
<tr>
<td></td>
<td>switch.</td>
</tr>
<tr>
<td>Keep Alive</td>
<td>The configured and actual keep alive times for this peer. The actual</td>
</tr>
<tr>
<td></td>
<td>keep alive time is set by the actual hold time in such a way that the</td>
</tr>
<tr>
<td></td>
<td>ratio of actual keep alive to hold time is the same as the ratio of</td>
</tr>
<tr>
<td></td>
<td>configured keep alive to hold time.</td>
</tr>
<tr>
<td>Min AS originated</td>
<td>The minimum time between advertisements of routes which originate in</td>
</tr>
<tr>
<td></td>
<td>this autonomous system.</td>
</tr>
<tr>
<td>Min route advert</td>
<td>The minimum time between advertisements of routes which originate</td>
</tr>
<tr>
<td></td>
<td>outside this autonomous system.</td>
</tr>
<tr>
<td>Filtering</td>
<td>Settings for inward and outward filtering of routing information</td>
</tr>
<tr>
<td></td>
<td>via BGP.</td>
</tr>
<tr>
<td>In filter</td>
<td>The traffic filter used for filtering incoming routes from this peer.</td>
</tr>
<tr>
<td>In path filter</td>
<td>The AS path filter used for filtering incoming routes from this peer.</td>
</tr>
<tr>
<td>In route map</td>
<td>The route map used for filtering incoming routes from this peer.</td>
</tr>
<tr>
<td>Max prefix</td>
<td>The maximum number of route prefixes that may be received from this</td>
</tr>
<tr>
<td></td>
<td>peer, and the action taken when this number is exceeded. The action is</td>
</tr>
<tr>
<td></td>
<td>one of “WARNING” or “TERMINATE”.</td>
</tr>
<tr>
<td>External hops</td>
<td>The number of hops that can be used to reach this peer, if it is an</td>
</tr>
<tr>
<td></td>
<td>EBGP peer. Having this number exceed 1 allows multihop EBGP.</td>
</tr>
<tr>
<td>Next hop self</td>
<td>Whether or not this switch will advertise to this peer that the next hop</td>
</tr>
<tr>
<td></td>
<td>for all routes is itself; one of “No” or “Yes”.</td>
</tr>
<tr>
<td>Send community</td>
<td>Whether or not this switch will send the community attribute in the</td>
</tr>
<tr>
<td></td>
<td>path attributes of UPDATE messages; one of “Yes” or “No”.</td>
</tr>
<tr>
<td>Messages In/Out</td>
<td>The number of incoming/outgoing BGP messages from/to this peer.</td>
</tr>
<tr>
<td>Debugging</td>
<td>The debugging types enabled for this peer; one or more of</td>
</tr>
<tr>
<td></td>
<td>“MSG”, “STATE”, “UPDATE” or “ALL”.</td>
</tr>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing BGP messages to this peer.</td>
</tr>
</tbody>
</table>
**SET BGP**

**Syntax**

```plaintext
SET BGP [ROUTERID=ipadd] [other-options]
```

where:
- `ipadd` is an IP address in dotted decimal notation.

**Description**

This command sets global BGP parameters in the switch.

The `ROUTERID` parameter specifies a four-byte number that uniquely identifies the switch in a network system. If `ROUTERID` is not set, the default local interface's IP address is used in the first instance, if it is configured, otherwise the highest interface IP address on the switch is used.

---

**CREATE LOG OUTPUT**

**Syntax**

```plaintext
CREATE LOG OUTPUT=(TEMPORARY|PERMANENT|output-id)
DESTINATION={EMAIL|MEMORY|NVS|ASYN|ROUTER|SYSLOG}
[LOCAL={NONE|1..15}] [other-options]
```

where:
- `ipadd` is an IP address

**Description**

The `LOCAL` parameter specifies a local interface to be used as the source for all LOG messages sent to a SYSLOG server. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option `NONE` is specified the switch will select a source from the current available interfaces instead.

**Example**

To create an output definition to forward log messages to a local UNIX host with IP address 192.168.32.77 using the local interface, local3 as the source of the log message, use the command:

```plaintext
CREATE LOG OUTPUT=3 DESTINATION=SYSLOG SERVER=192.168.32.77 LOCAL=3
```
**SET LOG OUTPUT**

**Syntax**

```
SET LOG OUTPUT={TEMPORARY|PERMANENT|output-id} [ASYN=port-number] [DESTINATION={EMAIL|MEMORY|NVS|ASYN|ROUTER|SYSLOG}] [FORMAT={FULL|MSGONLY|SUMMARY}] [MAXQUEUESEVERITY=severity] [MESSAGES=message-count] [QUEUEONLY={FALSE|NO|OFF|ON|TRUE|YES}] [SERVER=ipaddr] [TO=email-address] [ZONE={time-zone-name|utc-offset}] [LOCAL={NONE|1..15}]
```

**Description**
The LOCAL parameter specifies a local interface to be used as the source for all LOG messages sent to a SYSLOG server. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

**SHOW LOG OUTPUT**

**Syntax**

```
SHOW LOG OUTPUT={TEMPORARY|PERMANENT|output-id} [(FILTER=filter-id|FULL)]
```

where:

- **output-id** is the index number of an output definition, in the range 1 to 20.

**Description**
This command displays the specified or all output definitions.
Figure 4: Example detailed output from the SHOW LOG OUTPUT command.

```
Output Definition ............ 1
Enabled ........................ Yes
Type .......................... Syslog
IP Address (Server) .......... 202.36.163.20
Local Interface .............. local3
Time Zone ..................... -
Secure ......................... No
Queue Only ..................... No
Output Definition ............ 2
Enabled ........................ Yes
Type .......................... Router
IP Address (Server) .......... 202.36.163.40
Time Zone ..................... -
Secure ......................... No
Queue Only ..................... No
Filter 1:
  MODULE != IPX
  SEVERITY < 7
  --> Process
Filter 2:
  ALL

Output Definition ............ Permanent
Enabled ........................ Yes
Type .......................... NVS
Max Messages .................. 20
Time Zone ..................... -
Secure ......................... Yes
Filter 1:
  ALL
Output Definition ............ Temporary
Enabled ........................ Yes
Type .......................... Memory
Max Messages .................. 200
Time Zone ..................... -
Secure ......................... Yes
Filter 1:
  ALL
```

Table 4: New parameters in the output of the SHOW LOG OUTPUT command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in log messages destined for the UNIX SYSLOG server.</td>
</tr>
</tbody>
</table>
ADD RADIUS SERVER

Syntax

ADD RADIUS SERVER=ipadd SECRET=secret PORT=port-number ACCPORT=port-number [LOCAL={NONE 1 .. 15}]

where:

- **secret** is a character string, 1 to 63 characters long. It may contain uppercase letters (A-Z), lowercase letters (a-z), digits (0-9) and the underscore character ("_"). If the string contains spaces it must be enclosed in double quotes. It is case-sensitive.
- **ipadd** is an IP address in dotted decimal notation.
- **port-number** is a port number in the range 0 to 65535.

This command adds a RADIUS server to the list of known RADIUS servers. RADIUS servers are used for user authentication.

The **SERVER** parameter specifies the IP address of the RADIUS server, in dotted decimal notation. The server must not already be in the list of known RADIUS servers. If **SERVER** is specified, but **PORT** and **ACCPORT** are not, then the RADIUS server is used for both authentication and accounting, and requests are sent to the default ports (1645 and 1646). Use the **PORT** and **ACCPORT** parameters to prevent the RADIUS server being used for authentication or accounting, or to specify a different port number to use.

The **SECRET** parameter specifies a shared secret used in communications between the switch and the RADIUS server. The secret is used by the switch to encrypt the password field in authentication requests sent to the RADIUS server, and by the RADIUS server to authenticate the switch's request. The secret is case-sensitive.

The **PORT** parameter specifies a non-standard port number for communication with the RADIUS server. Setting the port number to zero means that the server will not be used for RADIUS authentication (it may only be required for RADIUS accounting).

The **ACCPORT** parameter specifies a port number for communication with the RADIUS server running RADIUS accounting (RFC 2139). Setting the port number to zero means that the server will not be used for RADIUS accounting (it may only be required for RADIUS authentication).

The **LOCAL** parameter specifies a local interface to be used as the source for all RADIUS packets the switch generates and subsequently sends to this RADIUS server. The local interface IP address will also be used as the NAS IP address in these outgoing packets. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option **NONE** is specified the switch will select a source from the current available interfaces instead.

By default the RADIUS server uses port number 1645 to connect to RADIUS servers for authentication, and port number of 1646 for RADIUS accounting. The RADIUS accounting port is not the official port number (1813) but is the port number used by a number of commonly available packages.
Examples

To add a RADIUS server with an IP address of 192.168.17.11 and “Valid8Me“ as the shared secret, use the command:

```
ADD RADIUS SERVER=192.16817.11 SECRET=Valid8Me
```

To add a RADIUS server for accounting only, with an IP address of 192.168.17.12 and “Valid8Me“ as the shared secret, use the command:

```
ADD RADIUS SERVER=192.16817.11 SECRET=Valid8Me PORT=0 ACCPORT=1813
```

To add a RADIUS server with an IP address of 192.168.17.11, “Valid8Me“ as the shared secret and local5 as the local interface use the command:

```
ADD RADIUS SERVER=192.16817.11 SECRET=Valid8Me LOCAL=5
```

SHOW RADIUS

Syntax

SHOW RADIUS

Description

This command displays the list of known RADIUS servers. RADIUS servers are used for user authentication.

Figure 5: Example output from the SHOW RADIUS command.

<table>
<thead>
<tr>
<th>Server</th>
<th>Port</th>
<th>AccPort</th>
<th>Secret</th>
<th>Local Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.17.11</td>
<td>1645</td>
<td>1646</td>
<td>****</td>
<td>local4</td>
</tr>
<tr>
<td>172.31.253.9</td>
<td>1645</td>
<td>0</td>
<td>****</td>
<td>Not set</td>
</tr>
</tbody>
</table>

Table 5: Example summary output from the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>The IP address of this RADIUS server.</td>
</tr>
<tr>
<td>Port</td>
<td>The port number used to communicate with the RADIUS authentication server.</td>
</tr>
<tr>
<td>AccPort</td>
<td>The port number used to communicate with the RADIUS accounting server.</td>
</tr>
<tr>
<td>Secret</td>
<td>The shared secret used in communications between the switch and the RADIUS server. Asterisks are displayed to prevent accidental discovery by unauthorised users.</td>
</tr>
<tr>
<td>Passcode</td>
<td>The status of the passcode prompt generation; one of “On“ or “Off“</td>
</tr>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing messages to the RADIUS server</td>
</tr>
</tbody>
</table>
SHOW SNMP

Syntax
SHOW SNMP

Description
This command displays information about the switch’s SNMP agent.

Syntax
SHOW SNMP

Description
This command displays information about the switch’s SNMP agent.

Figure 6: Example detailed output from the SHOW BGP PEER command.

SNMP configuration:
Status .................... Enabled
Authentication failure traps .... Enabled
Local Interface SNMPv1 ....... Not Set
Local Interface SNMPv2 ....... local5
Local Interface SNMPv3 ....... local2

Community .................... public
Access ........................ read-only
Status ........................ Enabled
Traps ........................ Enabled
Open access ........................ Yes
Community ........................ Administration
Access ........................ read-write
Status ........................ Disabled
Traps ........................ Disabled
Open access ........................ No

SNMP counters:
inPkts .......................... 0 outPkts .......................... 0
inBadVersions ................... 0 outTooBigs ........................ 0
inBadCommunityNames .......... 0 outNoSuchNames .......... 0
inBadCommunityUses ............ 0 outBadValues ............ 0
inASNParseErrs .................. 0 outGenErrs .................. 0
inTooBigs ........................ 0 outGetRequests ........................ 0
inNoSuchNames .................. 0 outGetNexts .................. 0
inBadValues ........................ 0 outSetRequests ........................ 0
inReadOnlys ........................ 0 outGetResponses ........................ 0
inGenErrs ........................ 0 outTraps ........................ 0
inTotalReqVars .................. 0
inTotalSetVars .................. 0
inGetRequests .................. 0
inGetNexts .................... 0
inSetRequests .................. 0
inGetResponses .................. 0
inTraps ........................ 0

Table 6: New parameters displayed in the output of the SHOW SNMP command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing SNMP messages.</td>
</tr>
</tbody>
</table>
ADD PIM BSRCANDIDATE

Syntax
ADD PIM BSRCANDIDATE [PREFERENCE=0..255] [BSMINTERVAL={10..15000|DEFAULT}] [INTERFACE={local-interface|vlan-interface}]

Description
The INTERFACE parameter specifies an interface for the router to use when advertising itself as a candidate bootstrap router. The IP address of this interface will be advertised by the router. The INTERFACE supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified the router will advertise its first active IP interface instead.

ADD PIM RPCANDIDATE

Syntax
ADD PIM RPCANDIDATE [=rp-address] GROUP=group-address [MASK=ipaddress] [PRIORITY=0..255] [ADVINTERVAL={10..15000|DEFAULT}] [INTERFACE={local-interface|vlan-interface}]

Description
The INTERFACE parameter specifies an interface for the router to use when advertising itself as the candidate rendezvous point for a multicast group. The IP address of this interface will be advertised by the router. The INTERFACE supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified the router will advertise its first active IP interface instead.

SET PIM RPCANDIDATE

Syntax
SET PIM RPCANDIDATE [=rp-address] GROUP=group-address [MASK=ipaddress] [PRIORITY=0..255] [ADVINTERVAL={10..15000|DEFAULT}] [INTERFACE={local-interface|vlan-interface}]

Description
The INTERFACE parameter specifies an interface for the router to use when advertising itself as the candidate rendezvous point for a multicast group. The IP address of this interface will be advertised by the router. The INTERFACE supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified the router will advertise its first active IP interface instead.
**SHOW PIM RPCANDIDATE**

**Syntax**

SHOW PIM RPCANDIDATE

**Description**

This command displays information about multicast groups for which the switch is a PIM-SM Rendezvous Point candidate.

Figure 7: Example output from the SHOW PIM RPCANDIDATE command...

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>The priority for the switch to become the Rendezvous Point for any multicast groups.</td>
<td></td>
</tr>
<tr>
<td>Group Address</td>
<td>The multicast groups associated with the specified Rendezvous Point.</td>
<td></td>
</tr>
<tr>
<td>Mask</td>
<td>The mask for the address.</td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td>The interface the switch will advertise itself as when advertising as a Rendezvous Point for multicast groups.</td>
<td></td>
</tr>
</tbody>
</table>

**ADD TACPLUS SERVER**

**Syntax**

ADD TACPLUS SERVER=

ipaddress [KEY=key] [PORT=port]

[SINGLECONNECTION=(YES|NO)] [TIMEOUT=1..10]

[LOCAL={NONE | 1..15}]

where:

- **ipaddress** is an IP address in dotted decimal notation.
- **key** is a string of up to 64 characters.
- **port** is an integer value.

**Description**

This command adds a TACACS+ server.

The SERVER parameter specifies the IP address of the TACACS+ server to identify. A network can have different TACACS+ servers for the purposes of authentication, authorization and accounting.

The PORT parameter specifies the TCP port number to be used when making connections to the TACACS+ server. The default port number is 49.

The LOCAL parameter specifies a local interface to be used as the source for all TACACS+ packets the device sends to this TACACS+ server. The local
interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

**SET TACPLUS SERVER**

**Syntax**
ADD TACPLUS SERVER=ipaddress [KEY=key] [PORT=port] [SINGLECONNECTION={YES|NO}] [TIMEOUT=1..10] [LOCAL={NONE|1..15}]

where:
- *ipaddress* is an IP address in dotted decimal notation.
- *key* is a string of up to 64 characters.
- *port* is an integer value.

**Description**
This command modifies parameters already set for a TACACS+ server. The SERVER parameter specifies the IP address of the TACACS+ server to be modified.

The LOCAL parameter specifies a local interface to be used as the source for all TACACS+ packets the device sends to this TACACS server. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

**SHOW TACPLUS SERVER**

**Syntax**
SHOW TACPLUS SERVER

**Description**
This command displays information about the configured TACACS+ servers.

![Example output from the SHOW TACPLUS SERVER command](image)

<table>
<thead>
<tr>
<th>Tacacs Plus Server Information</th>
<th>IP Address</th>
<th>Port</th>
<th>Timeout</th>
<th>Value</th>
<th>Sessions</th>
<th>Single connection</th>
<th>Local Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>172.168.198.254</td>
<td>49</td>
<td>5</td>
<td></td>
<td>1</td>
<td>Yes</td>
<td>local7</td>
</tr>
<tr>
<td></td>
<td>192.168.196.254</td>
<td>49</td>
<td>8</td>
<td></td>
<td>2</td>
<td>No</td>
<td>Not set</td>
</tr>
</tbody>
</table>

Table 8: New parameters displayed in the output of the SHOW TACPLUS SERVER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing TACACS+ messages sent to the TACACS+ server</td>
</tr>
</tbody>
</table>
Introduction

A Loopback interface is one that is always available for higher layer protocols to use and advertise out into a network. A local interface is assigned an IP address, but does not have the usual requirement of needing to be connected to a lower layer physical entity. It is the lack of an attachment to a physical entity that allows the perception of it always being accessible via the network.

Loopback interfaces can be utilised by a number of protocols for various purposes. They can be used to improve access to a device, as well as increasing the reliability, security, scalability and protection offered by the device. In addition they can add flexibility and simplify management, information gathering and filtering.

Loopback interfaces can help improve reliability. For example, having OSPF advertise a Loopback Interface as a interface-route, out into the network irrespective of what other physical links are up or down at the time - ensuring a higher probability of routing traffic being received and subsequently forwarded. Additionally, reliability and performance can be improved if BGP configures parallel paths to a Loopback Interface on a peer device, resulting in the improvement of load sharing.

Access and security can be improved through filtering. Incoming traffic can be filtered by rules that specify Loopback Interfaces as the only acceptable destination addresses.

Information gathering and filtering as well as management can potentially be simplified if protocols such as SNMP use Loopback interfaces for receiving and sending trap and log type information.
New Commands

Use the following new commands to configure Local Interfaces.

ADD IP LOCAL

Syntax

SET IP LOCAL=[1..15][FILTER={filter-number|NONE}]
 [GRE=[0..100 NONE}] [IPADDRESS=ipadd]
 [POLICYFILTER={filter-number|NONE}]
 [PRIORITYFILTER={filter-number|NONE}]

where:

- filter-number is a number from 0 to 299.
- ipadd is an IP address in dotted decimal notation.

Description

This command adds a local interface to the switch. A switch can only have fifteen local interfaces added. The switch has an additional default local interface, which is added on start up. The default local interface can be configured through the SET IP LOCAL command. A local interface is a virtual interface with no physical interface attachment. The interface can be assigned an IP address that can then be used as the source address of IP packets generated internally by IP protocols such as RIP, OSPF, PING and NTP. Higher layer protocols such as RIP, OSPF, PING and NTP must assign a source IP address to packets passed to IP for forwarding.

The following rules are used to determine which IP address to use as the source address:

1. If the higher layer protocol's configuration specifies either a source IP address or a local interface to use, then the configured address is used as the packet's source IP address. For example, the SIPADDRESS parameter of the PING command specifies the source IP address to use in ping packets. While the LOCAL parameter of the ADD BGP PEER command specifies a local interface to use to obtain a source IP address.

2. If the default local interface has been assigned an IP address, then the IP address of the default local interface is used as the packet's source IP address. Otherwise, the IP routing module determines the interface over which the packet is to be transmitted, and assigns the IP address of the interface as the packet's source IP address.

The LOCAL parameter specified is a unique identifying number that is used to identify the interface. The naming convention, or alias, for this interface is the concatenation of the word “local” along this identifying number.

The FILTER parameter specifies the filter to apply to IP packets transmitted or received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface. The default is not to apply a filter.
The GRE parameter specifies the GRE (Generic Routing Encapsulation) entity associated with the interface. The specified GRE entity must have been created previously using the ADD GRE command. The default is NONE. The IPADDRESS parameter specifies the IP address of the interface. The IP address must be the IP address of one of the switch's active IP interfaces. Specifying an IP address of 0.0.0.0 effectively 'unsets' the IP address of the local interface.

The POLICYFILTER parameter specifies the policy filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Policy filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PRIORITYFILTER parameter specifies the priority filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Priority filters are applied to packets as they are transmitted. The default is not to apply a filter.

Examples
To add the local interface 3 with an IP address of 192.168.33.1, use:

```
ADD IP LOCAL=3 IP=192.168.33.1
```

Related Commands
DELETE IP LOCAL
SET IP LOCAL
SHOW IP INTERFACE

DELETE IP LOCAL

Syntax
DELETE IP LOCAL=1..15

Description
This command deletes a local interface from the IP module. The local interface will no longer be used by the IP routing module.

When an IP interface is deleted, any static routes and ARP entries specific to the interface are also deleted

Examples
To delete local interface 5, use:

```
DELETE IP LOCAL=5
```

Related Commands
ADD IP LOCAL
SET IP LOCAL
SHOW IP INTERFACE
**SET SNMP LOCAL**

**Syntax**

```
SET SNMP VERSION={V1|V2|V3|ALL} LOCAL={NONE|1..15}
```

Where

- V1, V2 and V3 represent the version of the SNMP packets.

**Description**

This command sets a local interface to be used with a particular version of SNMP. Once set, the local interface will have its IP address used as the source IP address for all SNMP packets the switch generates and subsequently sends of that SNMP version.

The VERSION parameter specifies version of SNMP packets that the local interface will apply to. The default is ALL.

The LOCAL parameter specifies a local interface to be used as the source IP address for all SNMP packets the switch generates and sends of the specified SNMP version. The local interface IP address will be also be used as the SNMP agent IP address in these outgoing packets. The local interface must already be configured and be in the range 1 to 15. If a local interface is not set for SNMP, the switch will select a source from the current available interfaces.

**Examples**

To set the local interface local5 for SNMPv3 packets, use the command:

```
SET SNMP VERSION=V3 LOCAL=5
```

**Related Commands**

SHOW SNMP
## Modified Commands

The following commands to are changed to enable Local Interfaces to be configured. Changes have been highlighted.

### SET IP LOCAL

This command has been enhanced to enable up to 15 local ports to be modified.

**Syntax**

```plaintext
SET IP LOCAL[=DEFAULT|1..15] [FILTER={filter-number|NONE}]
GRE={0..100|NONE} [IPADDRESS=ipadd]
POLICYFILTER={filter-number|NONE} [PRIORITYFILTER={filter-number|NONE}]
```

where:
- `filter-number` is a number from 0 to 299.
- `ipadd` is an IP address in dotted decimal notation.

**Description**

This command modifies the parameters of one the switch's local interfaces. If the LOCAL argument is either not specified or is set to DEFAULT, the switch's default local interface is modified.

The FILTER parameter specifies the filter to apply to IP packets transmitted or received over the interface. The filter must already have been defined. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface. The default is not to apply a filter.

The GRE parameter specifies the GRE (Generic Routing Encapsulation) entity associated with the interface. The specified GRE entity must have been created previously using the ADD GRE command. The default is NONE.

The IPADDRESS parameter specifies the IP address of the interface. The IP address must be the IP address of one of the switch’s active IP interfaces. Specifying an IP address of 0.0.0.0 effectively ‘unsets’ the IP address of the local interface.

The POLICYFILTER parameter specifies the policy filter to apply to IP packets received over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface. Policy filters are applied to packets as they are transmitted. The default is not to apply a filter.

The PRIORITYFILTER parameter specifies the priority filter to apply to IP packets transmitted over the interface. The filter must already have been defined with the ADD IP FILTER command. An interface may have a maximum of one traffic filter, one policy filter and one priority filter, but the same traffic, policy or priority filter can be assigned to more than one interface.
Priority filters are applied to packets as they are transmitted. The default is not to apply a filter.

**Examples**

To set the IP address of the default local IP interface to 192.168.33.11, use:

```
SET IP LOCAL IP=192.168.33.11
```

To set the local interface 3 to 192.168.33.11, use:

```
SET IP LOCAL=3 IP=192.168.33.1
```

To remove the IP address of the default local IP interface, use:

```
SET IP LOCAL IP=0.0.0.0
```

### SHOW IP INTERFACE

**Syntax**

```
SHOW IP INTERFACE[=interface] [COUNTER[=MULTICAST]]
```

where:

- `interface` is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

**Description**

This command displays interface configuration information for the interfaces that assigned to the IP module using the ADD IP INTERFACE. If an interface is specified, then information for the specified interface is displayed, otherwise information for all IP interfaces is displayed.

A hash symbol (#) after the interface name indicates that the interface has an operational status of “down”. Note that interface routes are propagated by RIP when their status at a physical level is “up”. For VLAN interfaces, this means that the VLAN’s interface route is only propagated if at least one port in e VLAN is active.

The COUNTER parameter displays counters for the specified interface or all interfaces.

---

<table>
<thead>
<tr>
<th>Interface</th>
<th>Type</th>
<th>IP Address</th>
<th>Bc Fr</th>
<th>PArp</th>
<th>Filt</th>
<th>RIP Met.</th>
<th>SAMode</th>
<th>IPSc</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCAL</td>
<td>-</td>
<td>Not Set</td>
<td>-</td>
<td>n</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ppp1</td>
<td>Dynamic</td>
<td>0.0.0.0</td>
<td>1</td>
<td>y</td>
<td>-</td>
<td>-</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>ppp2</td>
<td>Inactive</td>
<td>192.168.23.3</td>
<td>1500</td>
<td>Off</td>
<td>-</td>
<td>-</td>
<td>Pass</td>
<td>-</td>
</tr>
<tr>
<td>vlan2</td>
<td>Static</td>
<td>192.168.1.1.1</td>
<td>1</td>
<td>n</td>
<td>-</td>
<td>-</td>
<td>Pass</td>
<td>No</td>
</tr>
<tr>
<td>vlan3#</td>
<td>Static</td>
<td>192.168.2.1</td>
<td>1</td>
<td>n</td>
<td>-</td>
<td>-</td>
<td>Pass</td>
<td>No</td>
</tr>
</tbody>
</table>

---

Figure 1: Example output from the SHOW IP INTERFACE command.
ADD BGP PEER

Syntax

ADD BGP PEER=ipaddr REMOTEAS=1..65534 [LOCAL={NONE | 1..15}] [other-options]

where:

- ipaddr is an IP address in dotted decimal notation.

Description
This command adds a BGP peer to the switch. This command adds the peer in the disabled state; the switch will not attempt to begin communicating with the peer until the ENABLE BGP PEER command is entered. This allows full configuration of the peer entry before starting to communicate with it.

The LOCAL parameter specifies a local interface to be used as the source for all BGP packets the switch generates and sends to this BGP peer. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

Examples
To add a BGP peer whose IP address is 192.168.1.1, whose AS number is 54321 and whose local interface is local3, use the command:

```
ADD BGP PEER=192.168.1.1 REMOTEAS=54321 DESCRIPTION="test remote BGP peer" LOCAL=3
```
### SET BGP PEER

**Syntax**

```
SET BGP PEER=ipadd [LOCAL={NONE|1..15}] [other-options]
```

where:
- `ipadd` is an IP address in dotted decimal notation.

**Description**

This command modifies parameters for an existing BGP peer in the switch. If the BGP peer is enabled, the BGP peer must be disabled, this command executed, and the BGP peer enabled before the change takes effect. If the BGP peer is disabled, the peer must be enabled before the change takes effect. The PEER parameter specifies the IP address of the peer to be modified. The peer must be an existing BGP peer on this switch.

The LOCAL parameter specifies a local interface to be used as the source for all BGP packets the switch generates and sends to this BGP peer. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

### SHOW BGP PEER

**Syntax**

```
SHOW BGP PEER [=ipadd]
```

where:
- `ipadd` is an IP address in dotted decimal notation.

**Description**

This command displays summary information about all BGP peers, or detailed information about the specified BGP peer.

The PEER parameter specifies the IP address of the BGP peer about which detailed information is to be displayed. If a value is not specified, summary information is displayed for all BGP peers.

Figure 2: Example summary output from the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>BGP peer entries</th>
<th>State</th>
<th>AS</th>
<th>InMsg</th>
<th>OutMsg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>192.168.2.254</td>
<td>Estab</td>
<td>12345</td>
<td>23456</td>
<td>8245</td>
</tr>
<tr>
<td>192.168.3.16</td>
<td>Idle (D)</td>
<td>123</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 2: Example summary output from the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer</td>
<td>The IP address of the BGP peer.</td>
</tr>
<tr>
<td>State</td>
<td>The BGP peer state; one of “Idle”, “Idle(D)”, “Connect”, “Active” “OpenSent”, “OpenConf” and “Estab”. “Idle(D)” indicates that the peer is idle and also disabled.</td>
</tr>
<tr>
<td>As</td>
<td>The number of the autonomous system to which this peer belongs.</td>
</tr>
<tr>
<td>InMsg</td>
<td>The number of messages received from this peer since the TCP connection opened.</td>
</tr>
<tr>
<td>OutMsg</td>
<td>The number of messages sent to this peer since the TCP connection opened.</td>
</tr>
</tbody>
</table>

Figure 3: Example detailed output from the SHOW BGP PEER command.

```
Peer ............... 192.168.2.254
Peer ............... 192.168.2.254
Description ........ Sprint's AS 12345
State .............. Established
Remote AS .......... 12345
Connect Retry ...... 200s
Hold time .......... 90s (actual 0s)
Keep alive .......... 30s (actual 0s - no KEEPALIVES)
Min AS originated ... 20s
Min route advert .... 40s
Filtering
In filter .......... -
In path filter ..... -
In route map ...... -
Out filter .......... 334
Out path filter ... -
Out route map ..... -
Max prefix .......... 2000 (action is WARNING)
External hops ...... 5 (EBGP multihop enabled)
Next hop self ...... No
Send community ...... No
Messages In/Out ..... 23456/3245
Debugging .......... -
Device ............. -
Local Interface ..... local4
```
Table 3: Parameters displayed in the detailed output of the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer</td>
<td>The IP address of the BGP peer.</td>
</tr>
<tr>
<td>Description</td>
<td>The description set for this BGP peer.</td>
</tr>
<tr>
<td>State</td>
<td>The BGP peer state; one of “Idle”, “Idle(D)”, “Connect”, “Active”, “OpenSent”, “OpenConfirm” and “Established”. “Idle(D)” indicates that the peer is idle and also disabled.</td>
</tr>
<tr>
<td>Remote AS</td>
<td>The number of the autonomous system to which this peer belongs.</td>
</tr>
<tr>
<td>Connect Retry</td>
<td>The time interval for retrying the initial TCP connection to this peer in the event of a connection failure.</td>
</tr>
<tr>
<td>Hold Time</td>
<td>The configured and actual hold times for this peer. The actual holdtime is the lower of the configured hold times of the peer and this switch.</td>
</tr>
<tr>
<td>Keep Alive</td>
<td>The configured and actual keep alive times for this peer. The actual keep alive time is set by the actual hold time in such a way that the ratio of actual keep alive to hold time is the same as the ratio of configured keep alive to hold time.</td>
</tr>
<tr>
<td>Min AS originated</td>
<td>The minimum time between advertisements of routes which originate in this autonomous system.</td>
</tr>
<tr>
<td>Min route advert</td>
<td>The minimum time between advertisements of routes which originate outside this autonomous system.</td>
</tr>
<tr>
<td>Filtering</td>
<td>Settings for inward and outward filtering of routing information via BGP.</td>
</tr>
<tr>
<td>In filter</td>
<td>The traffic filter used for filtering incoming routes from this peer.</td>
</tr>
<tr>
<td>In path filter</td>
<td>The AS path filter used for filtering incoming routes from this peer.</td>
</tr>
<tr>
<td>In route map</td>
<td>The route map used for filtering incoming routes from this peer.</td>
</tr>
<tr>
<td>Max prefix</td>
<td>The maximum number of route prefixes that may be received from this peer, and the action taken when this number is exceeded. The action is one of “WARNING” or “TERMINATE”.</td>
</tr>
<tr>
<td>External hops</td>
<td>The number of hops that can be used to reach this peer, if it is an EBGP peer. Having this number exceed 1 allows multihop EBGP.</td>
</tr>
<tr>
<td>Next hop self</td>
<td>Whether or not this switch will advertise to this peer that the next hop for all routes is itself; one of “No” or “Yes”.</td>
</tr>
<tr>
<td>Send community</td>
<td>Whether or not this switch will send the community attribute in the path attributes of UPDATE messages; one of “Yes” or “No”.</td>
</tr>
<tr>
<td>Messages In/Out</td>
<td>The number of incoming/outgoing BGP messages from/to this peer.</td>
</tr>
<tr>
<td>Debugging</td>
<td>The debugging types enabled for this peer; one or more of “MSG”, “STATE”, “UPDATE” or “ALL”.</td>
</tr>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing BGP messages to this peer.</td>
</tr>
</tbody>
</table>
SET BGP

Syntax

```
SET BGP [ROUTERID=ipadd] [other-options]
```

where:

- `ipadd` is an IP address in dotted decimal notation.

Description

This command sets global BGP parameters in the switch.

The ROUTERID parameter specifies a four-byte number that uniquely identifies the switch in a network system. If ROUTERID is not set, the default local interface's IP address is used in the first instance, if it is configured, otherwise the highest interface IP address on the switch is used.

CREATE LOG OUTPUT

Syntax

```
CREATE LOG OUTPUT={TEMPORARY|PERMANENT|output-id} 
  DESTINATION={EMAIL|MEMORY|NVS|ASYN|ROUTER|SYSLOG} 
  [LOCAL={NONE|1..15}] [other-options]
```

where:

- `ipadd` is an IP address

Description

The LOCAL parameter specifies a local interface to be used as the source for all LOG messages sent to a SYSLOG server. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

Example

To create an output definition to forward log messages to a local UNIX host with IP address 192.168.32.77 using the local interface, local3 as the source of the log message, use the command:

```
CREATE LOG OUTPUT=3 DESTINATION=SYSLOG SERVER=192.168.32.77 LOCAL=3
```
SET LOG OUTPUT

Syntax

```bash
SET LOG OUTPUT={TEMPORARY|PERMANENT|output-id} [ASYN=port-number] [DESTINATION={EMAIL|MEMORY|NVS|ASYN|ROUTER|SYSLOG}] [FORMAT={FULL|MSGONLY|SUMMARY}] [MAXQUEUESEVERITY=severity] [MESSAGES=message-count] [QUEUEONLY={FALSE|NO|OFF|ON|TRUE:YES}] [SECURE={FALSE|NO|OFF|ON|TRUE:YES}] [SERVER=ipadd] [TO=email-address] [LOCAL={NONE|1..15}] [FILTER=filter-id [ACTION={PROCESS|IGNORE}]} [ALL] [DATE=[op]dd-mmm-yyyy] [DEVICE=[op]device] [FILE=[op]filename] [MASK=ipadd] [MSGTEXT=[op]string] [MODULE=[op]module-id] [ORIGIN=ipadd] [REFERENCE=[op]string] [SEVERITY=[op]severity] [SOURCELINE=[op]line] [SUBTYPE=[op]subtype-id] [TIME=[op]hh:mm:ss] [TYPE=[op]type-id] [LOCAL={NONE|1..15}]```

Description

The LOCAL parameter specifies a local interface to be used as the source for all LOG messages sent to a SYSLOG server. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

SHOW LOG OUTPUT

Syntax

```bash
SHOW LOG OUTPUT={={TEMPORARY|PERMANENT|output-id}} [] ((FILTER=filter-id|FULL])
```

where:
- output-id is the index number of an output definition, in the range 1 to 20.

Description

This command displays the specified or all output definitions.
Figure 4: Example detailed output from the SHOW LOG OUTPUT command.

<table>
<thead>
<tr>
<th>Output Definition</th>
<th>Enabled</th>
<th>Type</th>
<th>IP Address (Server)</th>
<th>Local Interface</th>
<th>Time Zone</th>
<th>Secure</th>
<th>Queue Only</th>
<th>Filter 1:</th>
<th>Filter 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Syslog</td>
<td>202.36.163.20</td>
<td>local3</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>MODULE != IPX</td>
<td>ALL</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>Router</td>
<td>202.36.163.40</td>
<td></td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>SEVERITY &lt; 7</td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>---&gt; Process</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: New parameters in the output of the SHOW LOG OUTPUT command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in log messages destined for the UNIX SYSLOG server.</td>
</tr>
</tbody>
</table>
ADD RADIUS SERVER

Syntax

ADD RADIUS SERVER=ipadd SECRET=secret PORT=port-number
ACCPORT=port-number [LOCAL={NONE|1..15}]

where:

- **secret** is a character string, 1 to 63 characters long. It may contain uppercase letters (A-Z), lowercase letters (a-z), digits (0-9) and the underscore character ("_"). If the string contains spaces it must be enclosed in double quotes. It is case-sensitive.
- **ipadd** is an IP address in dotted decimal notation.
- **port-number** is a port number in the range 0 to 65535.

This command adds a RADIUS server to the list of known RADIUS servers. RADIUS servers are used for user authentication.

The **SERVER** parameter specifies the IP address of the RADIUS server, in dotted decimal notation. The server must not already be in the list of known RADIUS servers. If SERVER is specified, but PORT and ACCPORT are not, then the RADIUS server is used for both authentication and accounting, and requests are sent to the default ports (1645 and 1646). Use the PORT and ACCPORT parameters to prevent the RADIUS server being used for authentication or accounting, or to specify a different port number to use.

The **SECRET** parameter specifies a shared secret used in communications between the switch and the RADIUS server. The secret is used by the switch to encrypt the password field in authentication requests sent to the RADIUS server, and by the RADIUS server to authenticate the switch’s request. The secret is case-sensitive.

The **PORT** parameter specifies a non-standard port number for communication with the RADIUS server. Setting the port number to zero means that the server will not be used for RADIUS authentication (it may only be required for RADIUS accounting).

The **ACCPORT** parameter specifies a port number for communication with the RADIUS server running RADIUS accounting (RFC 2139). Setting the port number to zero means that the server will not be used for RADIUS accounting (it may only be required for RADIUS authentication).

The **LOCAL** parameter specifies a local interface to be used as the source for all RADIUS packets the switch generates and subsequently sends to this RADIUS server. The local interface IP address will also be used as the NAS IP address in these outgoing packets. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

By default the RADIUS server uses port number 1645 to connect to RADIUS servers for authentication, and port number of 1646 for RADIUS accounting. The RADIUS accounting port is not the official port number (1813) but is the port number used by a number of commonly available packages.
Examples

To add a RADIUS server with an IP address of 192.168.17.11 and “Valid8Me” as the shared secret, use the command:

```
ADD RADIUS SERVER=192.168.17.11 SECRET=Valid8Me
```

To add a RADIUS server for accounting only, with an IP address of 192.168.17.12 and “Valid8Me” as the shared secret, use the command:

```
ADD RADIUS SERVER=192.168.17.11 SECRET=Valid8Me PORT=0 ACCPORT=1813
```

To add a RADIUS server with an IP address of 192.168.17.11, “Valid8Me” as the shared secret and local5 as the local interface use the command:

```
ADD RADIUS SERVER=192.168.17.11 SECRET=Valid8Me LOCAL=5
```

SHOW RADIUS

Syntax

SHOW RADIUS

Description

This command displays the list of known RADIUS servers. RADIUS servers are used for user authentication.

Figure 5: Example output from the SHOW RADIUS command.

<table>
<thead>
<tr>
<th>Server</th>
<th>Port</th>
<th>AccPort</th>
<th>Secret</th>
<th>Local Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.17.11</td>
<td>1645</td>
<td>1646</td>
<td>******</td>
<td>local4</td>
</tr>
<tr>
<td>172.31.253.9</td>
<td>1645</td>
<td>0</td>
<td>******</td>
<td>Not set</td>
</tr>
</tbody>
</table>

Table 5: Example summary output from the SHOW BGP PEER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>The IP address of this RADIUS server.</td>
</tr>
<tr>
<td>Port</td>
<td>The port number used to communicate with the RADIUS authentication server.</td>
</tr>
<tr>
<td>AccPort</td>
<td>The port number used to communicate with the RADIUS accounting server.</td>
</tr>
<tr>
<td>Secret</td>
<td>The shared secret used in communications between the switch and the RADIUS server. Asterisks are displayed to prevent accidental discovery by unauthorised users</td>
</tr>
<tr>
<td>Passcode</td>
<td>The status of the passcode prompt generation; one of “On” or “Off”</td>
</tr>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing messages to the RADIUS server</td>
</tr>
</tbody>
</table>
SHOW SNMP

Syntax

SHOW SNMP

Description
This command displays information about the switch’s SNMP agent.

Figure 6: Example detailed output from the SHOW BGP PEER command.

```
SNMP configuration:
Status ...................... Enabled
Authentication failure traps .... Enabled
Local Interface SNMPv1 ........ Not Set
Local Interface SNMPv2 .......... local5
Local Interface SNMPv3 .......... local2

Community .................. public
Access ....................... read-only
Status ....................... Enabled
Traps ......................... Enabled
Open access ................... Yes
Community .................. Administration
Access ....................... read-write
Status ....................... Disabled
Traps ......................... Disabled
Open access ................... No

SNMP counters:
inPkts .......................... 0 outPkts .......................... 0
inBadVersions .................. 0 outTooBigs .................. 0
inBadCommunityNames ........... 0 outNoSuchNames .......... 0
inBadCommunityUses ............ 0 outBadValues ............ 0
inASNParseErrs ................ 0 outGenErrs ............... 0
inTooBigs ........................ 0 outGetRequests .......... 0
inNoSuchNames ................. 0 outGetNexts .......... 0
inBadValues ........................ 0 outSetRequests .......... 0
inReadOnlys ........................ 0 outGetResponses ....... 0
inGenErrs ........................ 0 outTraps ................. 0
inTotalReqVars .................. 0
inTotalSetVars .................. 0
inGetRequests ................... 0
inGetNexts ....................... 0
inSetRequests ................... 0
inGetResponses ................... 0
inTraps .......................... 0
```

Table 6: New parameters displayed in the output of the SHOW SNMP command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing SNMP messages.</td>
</tr>
</tbody>
</table>
ADD PIM BSRCANDIDATE

Syntax
ADD PIM BSRCANDIDATE [PREFERENCE=0..255]
[BSMINTERVAL={10..15000|DEFAULT}]
[INTERFACE={local-interface|vlan-interface}]

Description
The INTERFACE parameter specifies an interface for the router to use when advertising itself as a candidate bootstrap router. The IP address of this interface will be advertised by the router. The INTERFACE supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified the router will advertise its first active IP interface instead.

ADD PIM RPCANDIDATE

Syntax
ADD PIM RPCANDIDATE [=rp-address] GROUP=group-address
[MASK=ipaddress] [PRIORITY=0..255]
[ADVINTERVAL={10..15000|DEFAULT}]
[INTERFACE={local-interface|vlan-interface}]

Description
The INTERFACE parameter specifies an interface for the router to use when advertising itself as the candidate rendezvous point for a multicast group. The IP address of this interface will be advertised by the router. The INTERFACE supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified the router will advertise its first active IP interface instead.

SET PIM RPCANDIDATE

Syntax
SET PIM RPCANDIDATE [=rp-address] GROUP=group-address
[MASK=ipaddress] [PRIORITY=0..255]
[ADVINTERVAL={10..15000|DEFAULT}]
[INTERFACE={local-interface|vlan-interface}]

Description
The INTERFACE parameter specifies an interface for the router to use when advertising itself as the candidate rendezvous point for a multicast group. The IP address of this interface will be advertised by the router. The INTERFACE supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified the router will advertise its first active IP interface instead.
SHOW PIM RPCANDIDATE

**Syntax**
SHOW PIM RPCANDIDATE

**Description**
This command displays information about multicast groups for which the switch is a PIM-SM Rendezvous Point candidate.

Figure 7: Example output from the SHOW PIM RPCANDIDATE command...

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>The priority for the switch to become the Rendezvous Point for any multicast groups</td>
</tr>
<tr>
<td>Group Address</td>
<td>The multicast groups associated with the specified Rendezvous Point.</td>
</tr>
<tr>
<td>Mask</td>
<td>The mask for the address.</td>
</tr>
<tr>
<td>Interface</td>
<td>The interface the switch will advertise itself as when advertising as a Rendezvous Point for multicast groups.</td>
</tr>
</tbody>
</table>

Table 7: Parameters in the output of the SHOW PIM RPCANDIDATE command

ADD TACPLUS SERVER

**Syntax**
ADD TACPLUS SERVER=ipaddress [KEY=key] [PORT=port] [SINGLECONNECTION=(YES|NO)] [TIMEOUT=1..10] [LOCAL=(NONE|1..15)]

where:
- *ipaddress* is an IP address in dotted decimal notation.
- *key* is a string of up to 64 characters.
- *port* is an integer value.

**Description**
This command adds a TACACS+ server.

The SERVER parameter specifies the IP address of the TACACS+ server to identify. A network can have different TACACS+ servers for the purposes of authentication, authorization and accounting.

The PORT parameter specifies the TCP port number to be used when making connections to the TACACS+ server. The default port number is 49.

The LOCAL parameter specifies a local interface to be used as the source for all TACACS+ packets the device sends to this TACACS+ server. The local
interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

**SET TACPLUS SERVER**

**Syntax**

```
ADD TACPLUS SERVER=ipaddress [KEY=key] [PORT=port]
   [SINGLECONNECTION=(YES|NO)] [TIMEOUT=1..10]
   [LOCAL={NONE|1..15}]
```

where:
- `ipaddress` is an IP address in dotted decimal notation.
- `key` is a string of up to 64 characters.
- `port` is an integer value.

**Description**

This command modifies parameters already set for a TACACS+ server. The SERVER parameter specifies the IP address of the TACACS+ server to be modified.

The LOCAL parameter specifies a local interface to be used as the source for all TACACS+ packets the device sends to this TACACS server. The local interface must already be configured and be in the range 1 to 15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

**SHOW TACPLUS SERVER**

**Syntax**

```
SHOW TACPLUS SERVER
```

**Description**

This command displays information about the configured TACACS+ servers.

Figure 8: Example output from the SHOW TACPLUS SERVER command

<table>
<thead>
<tr>
<th>Tacacs Plus Server Information</th>
<th>IP Address</th>
<th>Port</th>
<th>Timeout Value</th>
<th>Sessions</th>
<th>Single connection</th>
<th>Local Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>172.168.198.254</td>
<td>49</td>
<td>5</td>
<td>1</td>
<td>Yes</td>
<td>local7</td>
</tr>
<tr>
<td></td>
<td>192.168.196.254</td>
<td>49</td>
<td>8</td>
<td>2</td>
<td>No</td>
<td>Not set</td>
</tr>
</tbody>
</table>

Table 8: New parameters displayed in the output of the SHOW TACPLUS SERVER command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Interface</td>
<td>The interface used as the source in outgoing TACACS+ messages sent to the TACACS+ server</td>
</tr>
</tbody>
</table>