Part No. 031729-00, Rev. A
June 2005

OmniSwitch 6600 Family
OmniSwitch 7700/7800
OmniSwitch 8800
User Guide Supplement

Release 5.1.6.R02

www.alcatel.com

This user guide documents OmniSwitch 6600 Family, OmniSwitch 7700/7800, and OmniSwitch 8800 hardware and software.
The information described in this guide are subject to change without notice.

This OmniSwitch product contains components which may be covered by one or more of the following U.S. Patents:

- U.S. Patent No. 6,339,830
- U.S. Patent No. 6,070,243
- U.S. Patent No. 6,061,368
- U.S. Patent No. 5,394,402
- U.S. Patent No. 6,047,024
- U.S. Patent No. 6,314,106
- U.S. Patent No. 6,542,507

26801 West Agoura Road
Calabasas, CA 91301
(818) 880-3500 FAX (818) 880-3505
info@ind.alcatel.com

US Customer Support—(800) 995-2696
International Customer Support—(818) 878-4507
Internet—http://eservice.ind.alcatel.com

Contents

1 User Documentation Addendum

This chapter includes information that should be added to or changed in the **5.1.6 release** of the set of user guides for the OmniSwitch 6600 Family, OmniSwitch 7700/7800, and OmniSwitch 8800.

OmniSwitch CLI Reference Guide

The following modifications should be made:

IPv6 Commands

Please refer to Chapter 2, "IPv6 Commands," in this addendum for CLI commands pertaining to IPv6.

Chapter 40, "High Availability VLAN Commands"

On page 40-2 the following two bullet items should be added to the **Usage Guidelines** section for the **vlan port-mac ingress-port** command:

- Note that removing the last ingress/egress port from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one ingress/egress port left in the VLAN.
- All HA VLAN related ports must first belong to the same default VLAN before they are configured as ingress, egress, or inter-switch ports for the HA VLAN.

On page 40-3 the **MIB Objects** section for the **vlan port-mac ingress-port** command should be replaced with the following:

```
vlanHAPortTable
  vlanHAPortVlanId
  vlanHAPortType
  vlanHAPortIfIndex
```

On page 40-4 the following two bullet items should be added to the **Usage Guidelines** section for the **vlan port-mac egress-port** command:

- Note that removing the last ingress/egress port from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one ingress/egress port left in the VLAN.
- All HA VLAN related ports must first belong to the same default VLAN before they are configured as ingress, egress, or inter-switch ports for the HA VLAN.

On page 40-5 the **MIB Objects** section for the **vlan port-mac egress-port** command should be replaced with the following:

```
vlanHAPortTable
  vlanHAPortVlanId
  vlanHAPortType
  vlanHAPortIfIndex
```

mac-address-table port-mac vlan mac

On page 40-6 the following bullet should be added to the **Usage Guidelines** section for the **mac-address-table port-mac vlan mac** command:

- Note that removing the last MAC address from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one MAC address left.

On page 40-7 the following MIB information should be added to the **MIB Objects** section for the **mac-address-table port-mac vlan mac** command:

```
  vlanHAPortVlanId
```

The following new command should be included in this chapter:

vlan port-mac bandwidth

Configures the bandwidth for the ingress flood queue associated with high availability (HA) VLANs.

vlan *vid* **port-mac bandwidth** *mbps*

Syntax Definitions

vid	An existing HA VLAN ID number (1–4094).
mbps	Bandwidth value for the specified HA VLAN flood queue (1mbps – 1000mbps).

Defaults

By default, the flood queue bandwidth for an HA VLAN is set to 15 mbps.

Platforms Supported

OmniSwitch 7700, 7800, 8800

Usage Guidelines

- The VLAN ID specified with this command must be the ID for an HA VLAN. An HA VLAN contains at least one ingress or egress port and one MAC address.
- The ingress flood queue is created when the first HA VLAN is configured on the switch, and deleted when the last HA VLAN is removed from the switch.

Examples

```
-> vlan 10 port-mac bandwidth 50
-> vlan 200 port-mac bandwidth 1000
```

Release History

Release 5.1.6; command was introduced.

Related Commands

vlan port-mac ingress-port	Adds and removes ingress ports from an HA VLAN.
vlan port-mac egress-port	Adds and removes egress ports from an HA VLAN.
mac-address-table port-mac vlan mac	Adds and removes MAC addresses from an HA VLAN.

MIB Objects

```
vlanTable
   vlanNumber
   vlanHABandwidth
```

On page 40-9 and 40-10 the **Examples** section for the **show mac-address-table port-mac** command should be replaced with the following:

```
-> show mac-address-table port-mac
Port mac configuration for vlan 10

Bandwidth : 15 MB/sec

   Ingress Port list:
           3/5   3/7
   Egress Port list:
           3/9  3/6
   Mac Address list:
           00:DA:95:3C:44:55
           00:13:14:34:5E:78
           01:23:45:C1:17:21

Port mac configuration for vlan 20

Bandwidth : 15 MB/sec

   Ingress Port list:
           1/4  8/2
   Egress Port list:
           4/9  4/6
   Mac Address list:
           00:11:22:33:44:05
           07:23:14:34:31:25
           00:23:45:67:43:04


-> show mac-address-table port-mac vlan 10
Port mac configuration for vlan 10

Bandwidth : 15 MB/sec

   Ingress Port list:
           3/5   3/7
   Egress Port list:
           3/9  3/6
   Mac Address list:
           00:DA:95:3C:44:55
           00:13:14:34:5E:78
           01:23:45:C1:17:21
```

On page 40-10 the following new field definition should be added to the **Output Definitions** table for the **show mac-address-table port-mac** command:

Bandwidth	The bandwidth size for the HA VLAN ingress flood queue. You can change this value with the **vlan port-mac bandwidth**.

On page 40-10 the following line should be added to the **Release History** section for the **show mac-address-table port-mac** command:

Release 5.1.6; **bandwidth** field added.

On page 40-10 the **MIB Objects** section for the **show mac-address-table port-mac** command should be replaced with the following:

```
vlanHAPortTable
   vlanHAPortVlanId
   vlanHAPortType
   vlanHAPortIfIndex
slMacToPortMacTable
   vlanHAPortVlanId
   slMacToPortMacAddress
vlanTable
   vlanNumber
```

Chapter 42, “802.1X Commands”

On page 42-11 replace the **Examples** section for the **show 802.1x** command with the following:

```
-> show 802.1x 1/13

802.1x configuration for slot 1 port 13:

  direction                      = both,
  operational directions         = both,
  port-control                   = auto,
  quiet-period (seconds)         = 60,
  tx-period (seconds)            = 30,
  supp-timeout (seconds)         = 30,
  server-timeout (seconds)       = 30,
  max-req                        = 2,
  re-authperiod (seconds)        = 3600,
  reauthentication               = no
  Guest Vlan ID                  = 20,
  Supplicant polling retry count = 2
```

On page 42-12 the following two new field definitions should be added to the **Output Definitions** table for the **show 802.1x** command:

`Guest VLAN ID`	Indicates if a guest VLAN is configured for non-802.1x traffic received on the port. If so, a VLAN ID number appears in this field. Configured through the **802.1x guest-vlan** command This field does not appear on an OmniSwitch 6800.
`Supplicant polling retry count`	The number of times a device is polled for EAP frames to determine whether or not the device is an 802.1x client. Configured through the **802.1x supp-polling retry** command. This field does not appear on an OmniSwitch 6800.

On page 42-13 the following MIB information should be added to the **MIB Objects** section for the **show 802.1x** command:

```
alaDot1xGuestVlanConfTable
   alaDot1xGuestVlanNumber
   alaDot1xSuppPollingCnt
```

The following three new commands should be included in this chapter:

802.1x guest-vlan

Configures a guest VLAN for an 802.1x port. When non-802.1x traffic is received on the specified port, it is assigned to the guest VLAN.

802.1x *slot/port* **guest-vlan** {*vid* | **disable**}

Syntax Definitions

slot	The slot number of the 802.1x port.
port	The 802.1x port number.
vid	The VLAN ID number that will serve as a guest VLAN for the 802.1x port.
disable	Disables the guest VLAN functionality for the 802.1x port.

Defaults

By default a guest VLAN is not configured for 802.1x ports.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- If a guest VLAN is already configured for the specified 802.1x port, the existing VLAN ID is overwritten with the new value. For example, if VLAN 10 is configured as a guest VLAN for 802.1x port 10/24 and this command is entered specifying VLAN 20, then VLAN 20 becomes the new guest VLAN for the port.
- Using the **disable** pulmotor also removes the guest VLAN association from the 802.1x port. The functionality is enabled again when a new guest VLAN is configured.
- The guest VLAN option is only available for 802.1x ports operating in the **auto** mode.
- Only one guest VLAN per 802.1x port is allowed.
- The VLAN ID specified with this command must already exist. VLANs are created using the **vlan** command.
- Note that on an OmniSwitch 6624/6648, non-802.1x clients learned on the guest VLAN are dropped if an 802.1x client successfully accesses the same port.

Examples

```
-> 802.1x 3/1 guest-vlan 5
-> 802.1x 3/1 guest-vlan disable
```

Release History

Release 5.1.6; command was introduced.

Related Commands

802.1x	Configures 802.1X parameters on a particular slot/port.
802.1x supp-polling retry	Configures the number of times a device is polled for EAP frames.
show 802.1x	Displays information about ports configured for 802.1X.
show 802.1x non-supp	Displays non-802.1x devices learned on the switch and their guest VLAN assignments.

MIB Objects

```
alaDot1xGuestVlanConfTable
  alaDot1xGuestVlanNumber
```

802.1x supp-polling retry

Configures the number of times to poll a device for EAP frames to determine whether or not the device is an 802.1x client.

802.1x *slot/port* **supp-polling retry** *retries*

Syntax Definitions

slot	The slot number of the 802.1x port.
port	The 802.1x port number.
retries	The number of times a device is polled for EAP frames (1–99).

Defaults

By default, the number of retries is set to 2.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guideline

- The polling interval is 0.5 seconds between each retry.
- If no EAP frames are received from a device connected to an 802.1x port, the device is considered a non-802.1x client (non-supplicant).
- If a guest VLAN is configured on the 802.1x port, the non-802.1x client is assigned to the guest VLAN. If a guest VLAN does not exist, the device is blocked from accessing the 802.1x port.

Examples

```
-> 802.1x 3/1 supp-polling retry 5
-> 802.1x 3/1 supp-polling retry 10
```

Release History

Release 5.1.6; command was introduced.

Related Commands

802.1x guest-vlan	Configures a guest VLAN to carry non-802.1x traffic that is received on an 802.1x port.
show 802.1x	Displays information about ports configured for 802.1X.
show 802.1x non-supp	Displays non-802.1x devices learned on the switch and their guest VLAN assignments.

MIB Objects

```
alaDot1xGuestVlanConfTable
  alaDot1xSuppPollingCnt
```

show 802.1x non-supp

Displays a list of all non-802.1x supplicants learned on all 802.1x ports.

show 802.1x non-supp [*slot/port***]**

Syntax Definitions

slot	The slot of the port for which you want to display information.
port	The port for which you want to display 802.1X information.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

If you do not specify a particular slot/port, all non-802.1x supplicants associated with all 802.1X ports are displayed.

Examples

```
->show 802.1x non-supp

Slot  MAC                Vlan
Port  Address            Learned
-----+-----------------+----------
3/1   00:61:4f:11:22:33     2
3/1   00:61:4f:44:55:66     2
3/1   00:61:4f:77:88:99     2
3/3   00:61:22:15:22:33     5
3/3   00:61:22:44:75:66     5

->show 802.1x non-supp 3/3

Slot  MAC                Vlan
Port  Address            Learned
-----+-----------------+----------
3/3   00:61:22:15:22:33     5
3/3   00:61:22:44:75:66     5
```

output definitions

`Slot/Port`	The 802.1X slot and port number that provides access to the non-802.1x device.
`MAC Address`	The source MAC address of the non-802.1x device connected to the 802.1x port.
`VLAN Learned`	The VLAN ID of the guest VLAN in which the source MAC address of the non-802.1x device was learned.

Release History

Release 5.1.6; command was introduced.

Related Commands

show 802.1x	Displays information about ports configured for 802.1X.

MIB Objects

```
alaDot1xPortTable
   alaDot1xNonSupplicantSlotNum
   alaDot1xNonSupplicantPortNum
   alaDot1xNonSupplicantMACAddress
   alaDot1xNonSupplicantVlanID
```

Chapter 22, “IP Commands”

On page 22-6 the following bullet should be added to the **Usage Guidelines** section for the **ip interface** command:

- To create an IP interface for network management purposes, specify **Loopback0** (case sensitive) as the name of the interface. The Loopback0 interface is not bound to any VLAN, so it will always remain operationally active.

OmniSwitch 7700/7800/8800 Network Configuration Guide

The following modifications should be made:

Chapter 13, “Configuring IP”

New Section, page 13-9

The following section should be added to page 13-9:

Configuring a Loopback0 Interface

Loopback0 is the name assigned to an IP interface to identify a consistent address for network management purposes. The Loopback0 interface is not bound to any VLAN, so it will always remain operationally active. This differs from other IP interfaces in that if there are no active ports in the VLAN, all IP interface associated with that VLAN are not active. In addition, the Loopback0 interface provides a unique IP address for the switch that is easily identifiable to network management applications.

This type of interface is created in the same manner as all other IP interfaces, using the **ip interface** command. To identify a Loopback0 interface, enter **Loopback0** for the interface name. For example, the following command creates the Loopback0 interface with an IP address of 10.11.4.1:

```
-> ip interface Loopback0 address 10.11.4.1
```

Note the following when configuring the Loopback0 interface:

- The interface name, “Loopback0”, is case sensitive.
- The **admin** parameter is the only configurable parameter supported with this type of interface.
- The Loopback0 interface is always active and available.
- Only one Loopback0 interface per switch is allowed.
- Creating this interface does *not* deduct from the total number of IP interfaces allowed per VLAN or switch.

Loopback0 Address Advertisement

The Loopback0 IP interface address is automatically advertised by the IGP protocols RIP and OSPF when the interface is created. There is no additional configuration necessary to trigger advertisement with these protocols.

Note the following regarding Loopback0 advertisement:

- RIP advertises the host route to the Loopback0 IP interface as a redistributed (directhost) route.
- OSPF advertises the host route to the Loopback0 IP interface in its Router-LSAs (as a Stub link) as an internal route into all its configured areas.

Configuring a BGP Peer Session with Loopback0

It is possible to create BGP peers using the Loopback0 IP interface address of the peering router and binding the source (i.e., outgoing IP interface for the TCP connection) to its own configured Loopback0 interface. The Loopback0 IP interface address can be used for both Internal and External BGP peer sessions. For EBGP sessions, if the External peer router is multiple hops away, the **ebgp-multihop** parameter may need to be used.

The following example command configures a BGP peering session using a Loopback0 IP interface address:

```
-> ip bgp neighbor 2.2.2.2 update-source Loopback0
```

See the *OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide* for more information.

Chapter 22, “Configuring 802.1X”

Quick Steps for Configuring 802.1X

On page 22-3 the following two new steps should be added to this section:

6 (Optional) Configure a guest VLAN for the 802.1x port using the **802.1x guest-vlan** command.

```
-> 802.1x 3/1 guest-vlan 5
```

7 (Optional) Configure the number of times supplicant devices are polled for identification using the **802.1x supp-polling retry** command.

```
-> 802.1x 3/1 supp-polling retry 10
```

On page 22-3 of this section replace the **Note** information about how to display 802.1x configuration and user information with the following:

Note. Verify the 802.1X port configuration using the **show 802.1x** command:

```
-> show 802.1x 1/13

802.1x configuration for slot 1 port 13:

  direction                   = both,
  operational directions      = both,
  port-control                = auto,
  quiet-period (seconds)      = 60,
  tx-period (seconds)         = 30,
  supp-timeout (seconds)      = 30,
  server-timeout (seconds)    = 30,
  max-req                     = 2,
  re-authperiod (seconds)     = 3600,
  reauthentication            = no
  Guest Vlan ID                  = 20,
  Supplicant polling retry count = 2
```

Optional. To display the number of 802.1x users on the switch, use the **show 802.1x users** command:

```
->show 802.1x users

Slot  MAC                Port                 User
Port  Address            State                Name
-----+------------------+--------------------+------------------------
3/1   00:60:4f:11:22:33  Connecting            user50
3/1   00:60:4f:44:55:66  Held                  user51
3/1   00:60:4f:77:88:99  Authenticated         user52
3/3   00:60:22:15:22:33  Force-authenticated   N/A
3/3   00:60:22:44:75:66  Force-authenticated   N/A
3/3   00:60:22:37:98:09  Force-authenticated   N/A
```

Optional. To display the number of non-802.1x users learned on the switch, use the **show 802.1x non-supp** command:

```
->show 802.1x non-supp

Slot  MAC               Vlan
Port  Address           Learned
-----+-----------------+----------
3/1   00:61:4f:11:22:33     2
3/1   00:61:4f:44:55:66     2
3/1   00:61:4f:77:88:99     2
3/3   00:61:22:15:22:33     5
3/3   00:61:22:44:75:66   5
```

See the *OmniSwitch CLI Reference Guide* for information about the fields in this display.

New Section, page 22-7

The following section should be added to page 22-7:

Guest VLANs for Non-802.1x Supplicants

For those supplicants that are not 802.1x devices—do not send/receive EAP frames—an optional guest VLAN feature is available to allow traffic from these devices on an 802.1x port. If the user-defined guest VLAN is not available, then traffic from a non-802.1x device is dropped.

The switch determines whether or not a device is an 802.1x supplicant by sending EAP-Request/Identity frames on the 802.1x port every 0.5 seconds for a configurable number of times. If no EAP frames are received from a device after the specified number of attempts, the device is determined to be a non-802.1x supplicant and is learned on the guest VLAN configured for that port. If no guest VLAN is available, then the non-802.1x supplicant is blocked from accessing the 802.1x port and no further attempts are made to solicit EAP frames from the device.

Note the following when using guest VLANs:

- 802.1x supplicants that fail authentication are not eligible for guest VLAN access. This type of VLAN access is only for those devices identified as non-802.1x supplicants that have not made any attempt to authenticate.

- Once a non-802.1x supplicant is learned on a guest VLAN, it is no longer eligible for Group Mobility classification and assignment.

- If a non-802.1x supplicant device becomes 802.1x capable when it is a member of a guest VLAN, upon authentication the device is automatically moved from the guest VLAN to the appropriate 802.1x specified VLAN. Disconnecting the device from the 802.1x port is not required in this scenario.

- If an authenticated 802.1x supplicant becomes non-802.1x capable, the device is moved to an existing guest VLAN after the device is rebooted.

By default a guest VLAN is not configured on an 802.1x port. For information about how to configure a guest VLAN, see "Configuring a Guest VLAN" on page 1-14. For information about how to set the number of times an unknown device is polled for identification, see "Configuring the Supplicant Polling Retry Count" on page 1-15.

New Section, page 22-11

The following section should be added to page 22-11:

Configuring a Guest VLAN

To configure a guest VLAN for an 802.1x port, use the **802.1x guest-vlan** command with the relevant slot/port number and specify an existing VLAN ID. For example:

```
-> 802.1x 3/1 guest-vlan 5
```

This command associates guest VLAN 5 with 802.1x port 3/1. When a non-802.1x supplicant is identified on this port, the source MAC address of the supplicant is learned in VLAN 5. This MAC address is then aged according to the aging timer value for VLAN 5.

To remove a guest VLAN from an 802.1x port, use the **disable** option with the **802.1x guest-vlan** command. Note that it is not necessary to specify the guest VLAN ID with this command. For example:

```
-> 802.1x 3/1 guest-vlan disable
```

Note the following when configuring a guest VLAN:

- The guest VLAN option is only available for 802.1x ports operating in the **auto** mode.

- Only one guest VLAN is allowed per 802.1x port.
- The VLAN ID specified must already exist in the switch configuration. Use the **vlan** command to create a VLAN before configuring it as an 802.1x guest VLAN.
- If a guest VLAN is already configured for the specified 802.1x port when the **802.1x guest-vlan** command is used, the existing VLAN ID is overwritten with the new value.

Configuring the Supplicant Polling Retry Count

To configure the number of times the switch polls an unknown device connected to an 802.1x port, use the **802.1x supp-polling retry** command. For example,

```
-> 802.1x 3/1 supp-polling retry 10
```

If after the number of polling attempts specified the device has not responded with EAP frames, then the device is learned as a non-802.1x supplicant in a guest VLAN. If a guest VLAN was not configured for the 802.1x port, the device is blocked from accessing that port and no other attempts are made to solicit EAP frames from the device.

Note that the polling interval is set to 0.5 seconds between each retry and is not a configurable at this time.

Chapter 28, “Configuring High Availability VLANs”

Replace all the contents of Chapter 28 with the contents of Chapter 3, “Configuring High Availability VLANs,” in this addendum.

OmniSwitch 7700/7800/8800 Advanced Routing Configuration Guide

The following modifications should be made:

Chapter 2, “Configuring BGP”

New Section, page 2-29

The following section should be added to page 2-29:

Configuring a BGP Peer with the Loopback0 Interface

Loopback0 is the name assigned to an IP interface to identify a consistent address for network management purposes. The Loopback0 interface is not bound to any VLAN, so it will always remain operationally active. This differs from other IP interfaces in that if there are no active ports in the VLAN, all IP interface associated with that VLAN are not active. In addition, the Loopback0 interface provides a unique IP address for the switch that is easily identifiable to network management applications.

It is possible to create BGP peers using the Loopback0 IP interface address of the peering router and binding the source (i.e., outgoing IP interface for the TCP connection) to its own configured Loopback0 interface. The Loopback0 IP interface address can be used for both Internal and External BGP peer sessions. For EBGP sessions, if the External peer router is multiple hops away, the **ebgp-multihop** parameter may need to be used.

The following example command configures a BGP peering session using a Loopback0 IP interface address:

```
-> ip bgp neighbor 2.2.2.2 update-source Loopback0
```

See the *OmniSwitch 7700/7800/8800 Network Configuration Guide* for more information about configuring an IP Loopback0 interface.

OmniSwitch 6600 Family Network Configuration Guide

The following modifications should be made:

Chapter 21, "Configuring 802.1X"

Quick Steps for Configuring 802.1X

On page 21-3 the following two new steps should be added to this section:

6 (Optional) Configure a guest VLAN for the 802.1x port using the **802.1x guest-vlan** command.

```
-> 802.1x 3/1 guest-vlan 5
```

7 (Optional) Configure the number of times supplicant devices are polled for identification using the **802.1x supp-polling retry** command.

```
-> 802.1x 3/1 supp-polling retry 10
```

On page 22-3 of this section replace the **Note** information about how to display 802.1x configuration and user information with the following:

Note. Verify the 802.1X port configuration using the **show 802.1x** command:

```
-> show 802.1x 1/13

802.1x configuration for slot 1 port 13:

  direction                    = both,
  operational directions       = both,
  port-control                 = auto,
  quiet-period (seconds)       = 60,
  tx-period (seconds)          = 30,
  supp-timeout (seconds)       = 30,
  server-timeout (seconds)     = 30,
  max-req                      = 2,
  re-authperiod (seconds)      = 3600,
  reauthentication             = no
  Guest Vlan ID                   = 20,
  Supplicant polling retry count = 2
```

Optional. To display the number of 802.1x users on the switch, use the **show 802.1x users** command:

```
->show 802.1x users

Slot  MAC                Port                 User
Port  Address            State                Name
-----+------------------+--------------------+-------------------------
3/1   00:60:4f:11:22:33  Connecting            user50
3/1   00:60:4f:44:55:66  Held                  user51
3/1   00:60:4f:77:88:99  Authenticated         user52
3/3   00:60:22:15:22:33  Force-authenticated   N/A
3/3   00:60:22:44:75:66  Force-authenticated   N/A
3/3   00:60:22:37:98:09  Force-authenticated   N/A
```

Optional. To display the number of non-802.1x users learned on the switch, use the **show 802.1x non-supp** command:

```
->show 802.1x non-supp

Slot  MAC               Vlan
Port  Address           Learned
-----+-----------------+----------
3/1   00:61:4f:11:22:33     2
3/1   00:61:4f:44:55:66     2
3/1   00:61:4f:77:88:99     2
3/3   00:61:22:15:22:33     5
3/3   00:61:22:44:75:66    5
```

See the *OmniSwitch CLI Reference Guide* for information about the fields in this display.

New Section, page 21-5

The following section should be added to page 21-5:

Guest VLANs for Non-802.1x Supplicants

For those supplicants that are not 802.1x devices—do not send/receive EAP frames—an optional guest VLAN feature is available to allow traffic from these devices on an 802.1x port. If the user-defined guest VLAN is not available, then traffic from a non-802.1x device is dropped.

The switch determines whether or not a device is an 802.1x supplicant by sending EAP-Request/Identity frames on the 802.1x port every 0.5 seconds for a configurable number of times. If no EAP frames are received from a device after the specified number of attempts, the device is determined to be a non-802.1x supplicant and is learned on the guest VLAN configured for that port. If no guest VLAN is available, then the non-802.1x supplicant is blocked from accessing the 802.1x port and no further attempts are made to solicit EAP frames from the device.

Note the following when using guest VLANs:

- Non-802.1x clients learned on a guest VLAN are dropped if an 802.1x client successfully authenticates on the same port. This is due to a one VLAN per port restriction (either 802.1x VLAN or guest VLAN assignment but not both) As a result, using a hub connection to provide access for multiple users to an 802.1x port is *not* recommended.
- 802.1x supplicants that fail authentication are not eligible for guest VLAN access. This type of VLAN access is only for those devices identified as non-802.1x supplicants that have not made any attempt to authenticate.

- Once a non-802.1x supplicant is learned on a guest VLAN, it is no longer eligible for Group Mobility classification and assignment.
- If a non-802.1x supplicant device becomes 802.1x capable when it is a member of a guest VLAN, upon authentication the device is automatically moved from the guest VLAN to the appropriate 802.1x specified VLAN. Disconnecting the device from the 802.1x port is not required in this scenario.
- If an authenticated 802.1x supplicant becomes non-802.1x capable, the device is moved to an existing guest VLAN after the device is rebooted.

By default a guest VLAN is not configured on an 802.1x port. For information about how to configure a guest VLAN, see "Configuring a Guest VLAN" on page 1-14. For information about how to set the number of times an unknown device is polled for identification, see "Configuring the Supplicant Polling Retry Count" on page 1-15.

New Section, page 21-10

The following section should be added to page 21-10:

Configuring a Guest VLAN

To configure a guest VLAN for an 802.1x port, use the **802.1x guest-vlan** command with the relevant slot/port number and specify an existing VLAN ID. For example:

```
-> 802.1x 3/1 guest-vlan 5
```

This command associates guest VLAN 5 with 802.1x port 3/1. When a non-802.1x supplicant is identified on this port, the source MAC address of the supplicant is learned in VLAN 5. This MAC address is then aged according to the aging timer value for VLAN 5.

To remove a guest VLAN from an 802.1x port, use the **disable** option with the **802.1x guest-vlan** command. Note that it is not necessary to specify the guest VLAN ID with this command. For example:

```
-> 802.1x 3/1 guest-vlan disable
```

Note the following when configuring a guest VLAN:

- The guest VLAN option is only available for 802.1x ports operating in the **auto** mode.
- Only one VLAN is allowed per 802.1x port. If a client successfully authenticates on the port, all guest VLAN users are dropped.
- The VLAN ID specified must already exist in the switch configuration. Use the **vlan** command to create a VLAN before configuring it as an 802.1x guest VLAN.
- If a guest VLAN is already configured for the specified 802.1x port when the **802.1x guest-vlan** command is used, the existing VLAN ID is overwritten with the new value.

Configuring the Supplicant Polling Retry Count

To configure the number of times the switch polls an unknown device connected to an 802.1x port, use the **802.1x supp-polling retry** command. For example,

```
-> 802.1x 3/1 supp-polling retry 10
```

If after the number of polling attempts specified the device has not responded with EAP frames, then the device is learned as a non-802.1x supplicant in a guest VLAN. If a guest VLAN was not configured for the

802.1x port, the device is blocked from accessing that port and no other attempts are made to solicit EAP frames from the device.

Note that the polling interval is set to 0.5 seconds between each retry and is not a configurable at this time.

2 IPv6 Commands

This chapter details Internet Protocol Version 6 (IPv6) commands for the switch (including RIPng commands). IPv6 (documented in RFC 2460) is designed as a successor to IPv 4. The changes from IPv4 to IPv6 fall primarily into the following categories:

Expanded Routing and Addressing Capabilities - IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy and a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses.

Header Format Simplification - Some IPv4 header fields were dropped or made optional, to reduce the common-case processing cost of packet handling and to keep the bandwidth cost of the IPv6 header as low as possible despite the increased size of the addresses. Even though the IPv6 addresses are four times longer than the IPv4 addresses, the IPv6 header is only twice the size of the IPv4 header.

Anycast Addressing - A new type of address called a "anycast address" is defined, to identify sets of nodes where a packet sent to an anycast address is delivered to one of the nodes. The use of anycast addresses in the IPv6 source route allows nodes to control the path which their traffic flows.

Improved Support for Options - Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

Authentication and Privacy Capabilities - IPv6 includes the definition of extensions which provide support for authentication, data integrity, and confidentiality. This is included as a basic element of IPv6 and will be included in all implementations.

IPv6 is supported on 6600/7700/7800/8800 series switches running software Release 5.1.6 and up.

MIB information for the IPv6 and RIPng commands is as follows:

Filename:	Ipv6.mib
Module:	Ipv6-MIB, Ipv6-TCP-MIB, Ipv6-UDP-MIB
Filename:	AlcatelIND1Ipv6.mib
Module:	alcatelIND1IPv6MIB
Filename:	AlcatelIND1Ripng.mib
Module:	alcatelIND1RipngMIB

A summary of the IPv6 commands is listed here:

IPv6	**ipv6 interface**
	ipv6 address
	ipv6 hop-limit
	ipv6 interface tunnel source destination
	ipv6 hop-limit
	ipv6 pmtu-lifetime
	ipv6 host
	ipv6 neighbor
	ipv6 prefix
	ipv6 route
	ping6
	traceroute6
	debug ipv6 packet
	debug ipv6 trace-category
	show ipv6 hosts
	show ipv6 icmp statistics
	show ipv6 interface
	show ipv6 pmtu table
	clear ipv6 pmtu table
	clear ipv6 neighbors
	show ipv6 prefixes
	show ipv6 routes
	show ipv6 tcp ports
	show ipv6 traffic
	clear ipv6 traffic
	show ipv6 tunnel
	show ipv6 udp ports
IPv6 RIP	**ipv6 load rip**
	ipv6 rip status
	ipv6 rip invalid-timer
	ipv6 rip garbage-timer
	ipv6 rip holddown-timer
	ipv6 rip jitter
	ipv6 rip route-tag
	ipv6 rip update-interval
	ipv6 rip triggered-sends
	ipv6 rip interface metric
	ipv6 rip interface recv-status
	ipv6 rip interface send-status
	ipv6 rip interface horizon
	ipv6 rip debug-level
	ipv6 rip debug-type
	show ipv6 rip
	show ipv6 rip interface
	show ipv6 rip peer
	show ipv6 rip routes
	show ipv6 rip debug

ipv6 interface

Configures an IPv6 interface on a VLAN or IPv6 tunnel.

ipv6 interface *if_name* **[vlan** *vid* **| tunnel {***tid* **| 6to4}] [enable | disable]**
[mtu *size***]**
[ra-send {yes | no}]
[ra-max-interval *interval***]**
[ra-managed-config-flag {true | false}]
[ra-other-config-flag {true | false}]
[ra-reachable-time *time***]**
[ra-retrans-timer *time***]**
[ra-default-lifetime *time* **| no ra-default-lifetime]**
[ra-send-mtu] {yes | no}

no ipv6 interface *if_name*

Syntax Definitions

if_name	IPv6 interface name.
vlan	Creates a VLAN interface.
vid	VLAN ID number.
tunnel	Creates a tunnel interface.
tid	Tunnel ID number.
6to4	Enables 6to4 tunneling.
mtu *size*	Maximum Transmission Unit for the interface.
ra-send	Specifies whether the router advertisements are sent on this interface.
ra-max-interval *interval*	Maximum time, in seconds, allowed between the transmission of unsolicited multicast router advertisements in this interface. The range is 4 - 1,800.
ra-managed-config-flag	Value to be placed in the managed address configuration flag field in router advertisements sent on this interface.
ra-other-config-flag	Value to be placed in the other stateful configuration flag in router advertisements sent on this interface.
ra-reachable-time *time*	Value, in milliseconds, to be placed in the reachable time field in router advertisements sent on this interface. The range is 0 - 3,600,000). The special value of zero indicates that this time is unspecified by the router.
ra-retrans-timer *time*	Value, in milliseconds, to be placed in the retransmit timer field in router advertisements sent on this interface. The value zero indicates that the time is unspecified by the router.

ra-default-lifetime *time*

Value, in seconds, to be placed in the router lifetime field in router advertisements sent on this interface. The time must be zero or between the value of “ra-max-interval” and 9,000 seconds. A value of zero indicates that the router is not to be used as a default router. The “no ra-default-lifetime” option will calculate the value using the formula (3 * ra-max-interval).

enable | disable

Administratively enable or disable the interface.

ra-send-mtu

Specifies whether the MTU option is included in the router advertisements sent on the interface.

Defaults

parameter	default
ra-send	yes
ra-max-interval	600
ra-managed-config-flag	false
ra-reachable-time	0
ra-retrans-timer	0
ra-default-lifetime	no
ra-send-mtu	no

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- When you create an IPv6 interface it is enabled by default.
- Use the “no” form of the command to delete an interface.
- All IPv6 VLAN and tunnel interfaces must have a name.
- When creating an IPv6 interface you must specify a VLAN ID, Tunnel ID, or **6to4**. When modifying or deleting an interface, you do not need to specify one of these options unless the name assigned to the interface is being changed. If it is present with a different value from when the interface was created, the command will be in error.
- A 6to4 interface cannot send advertisements (**ra-send**).
- To enable IPv6 routing you must first create a VLAN, then create an IPv6 interface on the VLAN. See Chapter 21, “VLAN Management Commands,” for information on creating VLANs.
- To route IPv6 traffic over an IPv4 network, you must create an IPv6 tunnel using the **ipv6 interface tunnel source destination** command.

Example

```
-> ipv6 interface Test vlan 1

-> ipv6 interface Test_Tunnel tunnel 2

-> ipv6 interface Test_6to4 tunnel 6to4
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 interface	Displays IPv6 Interface Table
show ipv6 tunnel	Displays IPv6 Tunnel information and whether the 6to4 tunnel is enabled.

MIB Objects

```
IPv6IfIndex
alaIPv6InterfaceTable
  alaIPv6InterfaceName
  alaIPv6InterfaceMtu
  alaIPv6InterfaceSendRouterAdvertisements
  alaIPv6InterfaceMaxRtrAdvInterval
  alaIPv6InterfaceAdvManagedFlag
  alaIPv6InterfaceAdvOtherConfigFlag
  alaIPv6InterfaceAdvRetransTimer
  alaIPv6InterfaceAdvDefaultLifetime
  alaIPv6InterfaceAdminStatus
  alaIPv6InterfaceAdvReachableTime
  alaIPv6InterfaceAdvSendMtu
  alaIPv6InterfaceRowStatus
```

ipv6 address

Configures an IPv6 address for an IPV6 interface on a VLAN, configured tunnel, or a 6to4 tunnel. There are different formats for this command depending on the address type.

ipv6 address *ipv6_address /prefix_length* **[anycast]** {*if_name* | **loopback**}

no ipv6 address *ipv6_address /prefix_length* **[anycast]** {*if_name* | **loopback**}

ipv6 address *ipv6_prefix/prefix_length* **eui-64** {*if_name* | **loopback**}

no ipv6 address *ipv6_prefix/prefix_length* **eui-64** {*if_name* | **loopback**}

Syntax Definitions

ipv6_address IPv6 address.

/prefix_length The number of bits that are significant in the IPv6 address (mask). (0...128).

anycast Indicates the address is an anycast address.

eui-64 Append an EUI-64 identifier to the prefix.

if_name Name assigned to the interface.

loopback Configures the loopback interface.

Defaults

parameter	default
/prefix_length	0

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- You can assign multiple IPv6 addresses to an IPv6 interface.
- Use the “no” form of the command to delete an address.
- The “eui” form of the command is used to add or remove an IPv6 address for a VLAN or configured tunnel using an EUI-64 interface ID in the low order 64 bits of the address.
- To enable IPv6 routing you must first create a VLAN, then create an IPv6 interface on the VLAN. See Chapter 21, “VLAN Management Commands,” for information on creating VLANs.
- To route IPv6 traffic over and IPv4 network, you must create an IPv6 tunnel using the **ipv6 interface tunnel source destination** command.

Example

```
-> ipv6 address 4132:86::19A/64 Test_Lab

-> ipv6 address 2002:d423:2323::35/64 Test_6to4
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 interface	Displays IPv6 Interface Table.

MIB Objects

```
IPv6IfIndex
alaIPv6InterfaceAddressTable
   alaIPv6InterfaceAddress
   alaIPv6InterfaceAddressAnycastFlag
   alaIPv6InterfaceEUI64AddressPrefixLength
   alaIPv6InterfaceEUI64AddressrowStatus

For EUI-64 Addresses:
alaIPv6InterfaceEUI64AddresssTable
   alaIPv6InterfaceEUI64Address
   alaIPv6InterfaceEUI64AddressPrefixLength
   alaIPv6InterfaceEUI64AddressRowStatus
```

ipv6 interface tunnel source destination

Configures the source and destination IPv4 addresses for a configured tunnel.

ipv6 interface *if_name* **tunnel {[source** *ipv4_source***] [destination** *ipv4_destination***]}**

Syntax Definitions

if_name	Name assigned to the tunnel interface.
ipv4_source	Source IPv4 address for the configured tunnel.
ipv4_destination	Destination IPv4 address for the configured tunnel.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use the **ipv6 interface** command to create an IPv6 tunnel interface.

Example

```
-> ipv6 interface Test tunnel 2 source 10.255.11.242 destination 10.255.11.242
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 interface	Creates an IPv6 tunnel interface.
show ipv6 tunnel	Displays IPv6 Tunnel information.

MIB Objects

```
IPv6IfIndex
  alaIPv6ConfigTunnelv4Source
  alaIPv6ConfigTunnelv4Dest
  alaIPv6ConfigTunnelRowStatus
```

ipv6 dad-check

Runs a Duplicate Address Detection (DAD) check on an address that was marked as duplicated.

ipv6 dad-check *ipv6_address if_name*

Syntax Definitions

ipv6_address	IPv6 address.
ip_name	Name assigned to the interface.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

The switch performs DAD check when an interface is attached to the stack and its VLAN first enters the active state. Use this command to rerun a DAD check on an address that was marked as duplicated.

Example

```
-> ipv6 dad-check fe80::2d0:95ff:fe6a:f458/64 Test_Lab
```

Release History

Release 5.1.6; command was introduced.

Related Commands

N/A.

MIB Objects

```
alaIPv6InterfaceAddressTable
  alaIPv6InterfaceAddressDADStatus
```

ipv6 hop-limit

Configures the value placed in the hop limit field in the header of all IPv6 packets that are originated by the switch. It also configures the value placed in the hop limit field in router advertisements.

ipv6 hop-limit *value*

no ipv6 hop-limit

Syntax Definitions

value	Hop limit value. The range is 0 - 255.

Defaults

parameter	default
value	64

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use the “no” form of the command to return the hop limit to its default value.

Example

```
-> ipv6 hop-limit 64
```

Release History

Release 5.1.6; command was introduced.

Related Commands

N/A.

MIB Objects

```
ipv6MibObjects
   Ipv6DefaultHopLimit
```

ipv6 pmtu-lifetime

Configures the configure the minimum lifetime for entries in the path MTU Table.

ipv6 pmtu-lifetime *time*

Syntax Definitions

time	Minimum path MTU entry lifetime, in minutes. Valid range is 10 - 1440.

Defaults

parameter	default
time	60

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

N/A.

Example

```
-> ipv6 pmtu-lifetime 30
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 pmtu table	Displays the IPv6 path MTU Table.
clear ipv6 pmtu table	Removes all entries from the IPv6 path MTU Table.

MIB Objects

```
alaIPv6ConfigTable
  alaIPv6PMTUMinLifetime
```

ipv6 host

Configures a static host name to IPv6 address mapping to the local host table.

ipv6 host *name ipv6_address*

no ipv6 host *name ipv6_address*

Syntax Definitions

name	Host name associated with the IPv6 address (1 - 255 characters).
ipv6_address	IPv6 address.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use the "no" form of the command to remove the mapping from the host table.

Example

```
-> ipv6 host Lab 4235::1200:0010
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 hosts	Displays IPv6 Local Hosts Table.

MIB Objects

```
alaIPv6HostTable
   alaIPv6HostName
   alaIPv6HostAddress
   alaIPv6HostRowStatus
```

ipv6 neighbor

Configures a static entry in the IPv6 Neighbor Table.

ipv6 neighbor *ipv6_address hardware_address* {*if_name*} *slot/port*

no ipv6 neighbor *ipv6_address* {*if_name*}

Syntax Definitions

ipv6_address	IPv6 address that corresponds to the hardware address.
hardware_address	MAC address in hex format (e.g., 00:00:39:59:F1:0C).
if_name	Name assigned to the interface on which the neighbor resides.
slot/port	Slot/port used to reach the neighbor.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use the "no" form of the command to remove an entry from the IPv6 Neighbor Table.

Example

```
-> ipv6 neighbor 4132:86::203 00:d0:c0:86:12:07 Test 1/1
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 neighbors	Displays IPv6 Neighbor Table.

MIB Objects

```
IPv6IfIndex
alaIPv6NeighborTable
   alaIPv6NeighborNetAddress
   alaIPv6NeighborPhysAddress
   alaIPv6NeighborSlot
   alaIPv6NeighborPort
   alaIPv6NeighborRowStatus
```

ipv6 prefix

Configures an IPv6 prefix on an interface. Used for configuring prefixes for router advertisements.

ipv6 prefix *ipv6_address /prefix_length if_name*
[valid-lifetime *time***]**
[preferred-lifetime *time***]**
[on-link-flag {true | false}
[autonomous-flag {true | false}] *if_name*

no ipv6 prefix *ipv6_address /prefix_length if_name*

Syntax Definitions

ipv6_address	IPv6 address of the interface.
/prefix_length	The number of bits that are significant in the iPv6 address (mask). (0...128).
valid-lifetime *time*	Length of time, in seconds, that this prefix will remain valid, i.e. time until deprecation. A value of 4,294,967,295 represents infinity.
preferred-lifetime *time*	Length of time, in seconds, that this prefix will remain preferred, i.e. time until deprecation. A value of 4,294,967,295 represents infinity.
on-link-flag	On-link configuration flag. When “true.” this prefix can be used for on-link determination.
autonomous-flag	Autonomous address configuration flag. When “true,” indicates that this prefix can be used for autonomous address configuration (i.e., can be used to form a local interface address).
if_name	Name assigned to the interface.

Defaults

parameter	default
valid-lifetime *time*	2,592,000
preferred-lifetime *time*	604,800
on-link-flag	true
autonomous-flag	true

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use the “no” form of the command to delete a prefix.

Example

```
-> ipv6 prefix 4132:86::/64 Test
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 prefixes	Displays IPv6 prefixes used in router advertisements.

MIB Objects

```
IPv6IfIndex
```

```
alaIPv6InterfacePrefixTable
   alaIP6vInterfacePrefix
   alaIP6vInterfacePrefixLength
   alaIP6vInterfacePrefixValidLifetime
   alaIP6vInterfacePrefixPreferredLifetime
   alaIP6vInterfacePrefixonLinkFlag
   alaIP6vInterfacePrefixAutonomousFlag
   alaIP6vInterfacePrefixRowStatus
```

ipv6 route

Configures a static entry in the IPv6 route.

ipv6 route *ipv6_prefix/prefix_length ipv6_address* [*if_name*]

no ipv6 route *ipv6_prefix/prefix_length ipv6_address* [*if_name*]

Syntax Definitions

ipv6_prefix	IPv6 network that is the destination of this static route.
/prefix_length	The number of bits that are significant in the iPv6 address (mask). (0...128).
ipv6_address	IPv6 address of the next hop used to reach the specified network.
if_name	If the next hop is a link-local address, the name of the interface used to reach it.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800

Usage Guidelines

Use the “no” form of the command to remove a static route.

Example

```
-> ipv6 route 212:95:5::/64 fe80::2d0:95ff:fe6a:f458 v6if-137
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 routes	Displays IPv6 Forwarding Table.

MIB Objects

```
alaIPv6StaticRouteTable
   alaIPv6StaticRouteNextHop
   alaIPv6StaticRouteIfIndex
   alaIPv6StaticRouteDest
   alaIPv6StaticRoutePrefixLength
   alaIPv6StaticRouteRowStatus
```

ping6

Used to test whether an IPv6 destination can be reached from the local switch. This command sends an ICMPv6 echo request to a destination and then waits for a reply. To ping a destination, enter the **ping6** command and enter either the destination's IPv6 address or hostname. The switch will ping the destination using the default frame count, packet size, and interval (6 frames, 64 bytes, and 1 second respectively). You can also customize any or all of these parameters as described below.

ping6 {*ipv6_address* | *hostname*} [*if_name*] **[count** *count***]** **[size** *data_size***]** **[interval** *seconds***]**

Syntax Definitions

ipv6_address	IP address of the system to ping.
hostname	DNS name of the system to ping.
if_name	If the target is a link-local address, the name of the interface used to reach it.
count	Number of packets to be transmitted.
size	Size of the data portion of the packet sent for this ping, in bytes.
seconds	Interval, in seconds, at which ping packets are transmitted.

Defaults

parameter	default
count	6
size	56
interval *seconds*	1

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- If you change the default values they will only apply to the current ping. The next time you use the ping command, the default values will be used unless you again enter different values.
- When the next hop address is a local link address, the name of the interface used to reach the destination must be specified.

Example

```
-> ping6 fe80::2d0:95ff:fe6a:f458/64
```

Release History

Release 5.1.6; command was introduced.

Related Commands

traceroute6	Used to find the path taken by an IPv6 packet from the local switch to a specified destination.

traceroute6

Used to find the path taken by an IPv6 packet from the local switch to a specified destination. This command displays the individual hops to the destination as well as some timing information.

traceroute6 {*ipv6_address* | *hostname*} [*if_name*] **[max-hop** *hop_count***] [wait-time** *time***] [port** *port_number***] [probe-count** *probe***]**

Syntax Definitions

ipv6_address	Destination IPV6 address IPv6 address of the host whose route you want to trace.
hostname	DNS name of the host whose route you want to trace.
if_name	If the target is a link-local address, the name of the interface used to reach it.
hop_count	Maximum hop count for the trace.
time	Delay time, in seconds between probes
port	Specific UDP port destination. By default, the destination port is chosen by traceroute6.
probe	Number of probes to be sent to a single hop.

Defaults

parameter	default
hop_count	30
time	5
probe	3

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- When using this command, you must enter the name of the destination as part of the command line (either the IPv6 address or hostname).
- Use the optional **max-hop** parameter to set a maximum hop count to the destination. If the trace reaches this maximum hop count without reaching the destination, the trace stops.

Example

```
-> traceroute6 41EA:103::65C3
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ping6	Used to test whether an IPv6 destination can be reached from the local switch.

debug ipv6 packet

Configures the display of IPv6 debug messages.

```
debug ipv6 packet
[defaults]
[v6header {concise | verbose}]
[extheader {none | payload | concise | verbose}]
[etherheader {yes | no}]
[raw bytes]
[board {all | cmm | ni [slot_number] | none}]
[ether-filter mac_address | either-filter-pair mac_address mac_address | no ether-filter]
[ipv6-filter ipv6_address [/prefix_length] | ipv6-filter-pair ipv6_address [/prefix_length] | no ipv6-filter]
[direction {all | in | out | from-cmm | from-ipv4 | to-cmm | to-ipv4}]
[output {console | file filename}]
```

```
no debug ipv6 packet
```

Syntax Definitions

defaults Resets all settings to default values.

v6header Sets the display format for the IPv6 header.

extheader Sets the display format for IPv6 extension headers:
none - No extension headers will be displayed
payload - Information on the final payload header only
concise - Concise information on all extension headers
verbose - Verbose information on all extension headers.

etherheader Specifies whether the packet's Ethernet header will be displayed.

raw *bytes* If bytes is not zero, this number of raw hex bytes of the packet will be displayed.

board Specifies the board(s) on which packet debug is enabled.

ether-filter Allows filtering of packets based on their source and destination MAC addresses. If a single MAC address is specified, only packets whose source or destination MAC address match the specified value will be displayed. If a pair of MAC addresses is specified, only those packets being exchanged between the two MAC addresses will be displayed.

ipv6-filter Allows filtering of packets based on their source and destination IPv6 addresses. If a single IPv6 address is specified, only packets sent to or received from that address will be displayed. If a pair of addresses is specified, only those packets being exchanged between the two addresses will be displayed.

direction

Allows filtering of packets based on the direction of flow:
all - debug both incoming and outgoing packets
in - debug incoming IPv6 packets
out - debug outgoing packets
from-cmm - debug packets received from the CMM.
from-ipv4 - debug packets received from an IPv4 interface.
to-cmm - debug packets sent to the CMM.
to-ipv4 - debug packets sent to an IPv4 interface.

output

Specifies the destination for the debug information.
console - write debug information to the console screen or file
file *filename* - write debug information to the specified file.

Defaults

parameter	default
v6header	concise
extheader	payload
etherheader	yes
raw *bytes*	0
board	all
ether-filter	no ether-filter
ipv6-filter	no ipv6-filter
direction	all
output	console

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- Use the **no** form of the command to turn off IPv6 debugging.
- Options are additive across multiple command lines until reset with the “default” option.

Example

```
-> debug ipv6 packet defaults
```

Release History

Release 5.1.6; command was introduced.

Related Commands

debug ipv6 trace-category	Enables/disables specific IPv6 EDR trace categories.

MIB Objects

N/A.

debug ipv6 trace-category

Enables/disables specific IPv6 EDR trace categories. If a category is enabled (e.g., vlan, tunnel), switch log messages generated for that category are written to the switch log.

debug ipv6 trace-category [all | default | general | cmm-control | ni-data | ni-control | vlan | tunnel | neighbor | route | mip | ipc | cd | pm | sm | monitor | rtadv]

no debug ipv6 trace-category [all | default | general | cmmcontrol | nidata | nicontrol | vlan | tunnel | neigh | route | mip | ipc | cd | pm | sm | monitor | rtadv]

Syntax Definitions

all Enable/disable all trace categories.

default Enable the default trace categories (general and monitor).

general Enable/disable the general trace category

cmm-control Enable/disable trace messages pertaining to the CMM control socket.

ni-data Enable/disable trace messages pertaining to the exchange of IPv6 packets with the NIs.

ni-control Enable/disable trace messages pertaining to the control messages exchanged with the NIs.

vlan Enable/disable trace messages pertaining to VLAN interfaces.

tunnel Enable/disable trace messages pertaining to tunnel interfaces.

neighbor Enable/disable trace messages pertaining to the neighbor cache.

route Enable/disable trace messages pertaining to the forwarding table.

mip Enable/disable trace messages pertaining to MIP processing.

ipc Enable/disable trace messages pertaining to IPC communications.

cs Enable/disable trace messages pertaining to chassis supervision.

pm Enable/disable trace messages pertaining to port manager.

sm Enable/disable trace messages pertaining to session manager.

monitor Enable/disable debug and monitoring trace messages.

rtadv Enable/disable router advertisement trace messages.

Defaults

N/A

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- Use the **no** form of the command to disable debug messages for a category.
- The general and monitor categories are the only ones enabled by default.
- Options are additive across multiple command lines until reset with the “default” option.
- This command controls only debug level switch log messages (Debug 1,2,3). Messages at higher levels are always logged.

Example

```
-> debug ipv6 trace-category all
```

Release History

Release 5.1.6; command was introduced.

Related Commands

debug ipv6 packet	Configures the display of IPv6 debug messages.

MIB Objects

N/A.

show ipv6 hosts

Displays IPv6 Local Hosts Table.

show ipv6 hosts [*substring*]

Syntax Definitions

substring	Limits the display to host names starting with the specified substring.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

If you do not specify a substring, all IPv6 hosts are displayed.

Example

```
-> show ipv6 hosts

Name                                              IPv6 Address
--------------------------------------------------+--------------------------------------
ipv6-test1.alcatel.com                            4235::1200:0010
ipv6-test2.alcatel.com                            4235::1200:0020
otheripv6hostname                                 4143:1295:9490:9303:00d0:6a63:5430:9031
```

output definitions

Name	Name associated with the IPv6 address.
IPv6 Address	IPv6 address associated with the host name.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 host	Configures a static host name to IPv6 address mapping to the local host table.

MIB Objects

```
alaIPv6HostTable
   alaIPv6HostName
   alaIPv6HostAddress
```

show ipv6 icmp statistics

Displays IPv6 ICMP statistics.

show ipv6 icmp statistics [*if_name*]

Syntax Definitions

if_name	Display statistics only for this interface.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

The ICMP Table can be used to monitor and troubleshoot the switch.

Example

```
-> show ipv6 icmp statistics

Message                        Received   Sent
------------------------------+----------+----------
 Total                                  0          0
 Errors                                 0          0
 Destination Unreachable                0          0
 Administratively Prohibited            0          0
 Time Exceeded                          0          0
 Parameter Problems                     0          0
 Packet Too Big                         0          0
 Echo Requests                          0          0
 Echo Replies                           0          0
 Router Solicitations                   0          0
 Router Advertisements                  0          0
 Neighbor Solicitations                 0          0
 Neighbor Advertisements                0          0
 Redirects                              0          0
 Group Membership Queries               0          0
 Group Membership Responses             0          0
 Group Membership Reductions            0          0
```

output definitions

Total	Total number of ICMPv6 messages the switch received or attempted to send.
Errors	Number of ICMPv6 messages the switch sent or received but was unable to process because of ICMPv6-specific errors (bad checksums, bad length, etc.).
Destination Unreachable	Number of Destination Unreachable messages that were sent or received by the switch.
Administratively Prohibited	Number of Destination Unreachable/Communication Administratively Prohibited messages sent or received by the switch.
Time Exceeded	Number of Time Exceeded messages sent or received by the switch.
Parameter Problems	Number of Parameter Problem messages sent or received by the switch.
Packet Too Big	Number of Packet Too Big messages sent or received by the switch.
Echo Requests	Number of Echo Request messages sent or received by the switch.
Echo Replies	Number of Echo Reply messages sent or received by the switch.
Router Solicitations	Number of Router Solicitations sent or received by the switch.
Router Advertisements	Number of Router Advertisements sent or received by the switch.
Neighbor Solicitations	Number of Neighbor Solicitations sent or received by the switch.
Neighbor Advertisements	Number of Neighbor Advertisements sent or received by the switch.
Redirects	Number of Redirect messages sent or received by the switch.
Group Membership Queries	Number of Group Membership Queries sent or received by the switch.
Group Membership Responses	Number of Group Membership Responses sent or received by the switch.
Group Membership Reductions	Number of Group Membership Reductions sent or received by the switch.

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 traffic	Displays IPv6 traffic statistics.

MIB Objects

```
ipv6IfIcmpTable
   ipv6IfIcmpInMsgs
   ipv6IfIcmpInErrors
   ipv6IfIcmpInDestUnreachs
   ipv6IfIcmpInAdminProhibs
   ipv6IfIcmpInTimeExcds
   ipv6IfIcmpInParmProblems
   ipv6IfIcmpInPktTooBigs
   ipv6IfIcmpInEchos
   ipv6IfIcmpInEchoReplies
   ipv6IfIcmpInRouterSolicits
   ipv6IfIcmpInRouterAdvertisements
   ipv6IfIcmpInNeighborSolicits
   ipv6IfIcmpInNeighborAdvertisements
   ipv6IfIcmpInRedirects
   ipv6IfIcmpInGroupMembQueries
   ipv6IfIcmpInGroupMembResponses
   ipv6IfIcmpInGroupMembReductions
   ipv6IfIcmpOutMsgs
   ipv6IfIcmpOutErrors
   ipv6IfIcmpOutDestUnreachs
   ipv6IfIcmpOutAdminProhibs
   ipv6IfIcmpOutTimeExcds
   ipv6IfIcmpOutParmProblems
   ipv6IfIcmpOutPktTooBigs
   ipv6IfIcmpOutEchos
   ipv6IfIcmpOutEchoReplies
   ipv6IfIcmpOutRouterSolicits
   ipv6IfIcmpOutRouterAdvertisements
   ipv6IfIcmpOutNeighborSolicits
   ipv6IfIcmpOutNeighborAdvertisements
   ipv6IfIcmpOutRedirects
   ipv6IfIcmpOutGroupMembQueries
   ipv6IfIcmpOutGroupMembResponses
   ipv6IfIcmpOutGroupMembReductions
```

show ipv6 interface

Displays IPv6 Interface Table.

show ipv6 interface [*if_name* | **loopback**]

Syntax Definitions

if_name	Interface name. Limits the display to a specific interface.
loopback	Limits display to loopback interfaces.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- If you do not specify an interface name, all IPv6 interfaces are displayed.
- Specify an interface name (e.g., VLAN 12) to obtain more detailed information about a specific interface.

Example

```
-> show ipv6 interface

Name                 IPv6 Address/Prefix Length                    Status    Device
--------------------+----------------------------------------+---------+--------
smbif-5              fe80::2d0:95ff:fe12:f470/64                   Active    VLAN 955
                     212:95:5::35/64
                     212:95:5::/64
v6if-to-eagle        fe80::2d0:95ff:fe12:f470/64                   Disabled  VLAN 1002
                     195:35::35/64
                     195:35::/64
V6if-6to4-137        2002:d423:2323::35/64                         Active    6to4 Tunnel
                     2002:d423:2323::/64
v6if-tunnel-137      fe80::2d0:95ff:fe12:f470/64                   Disabled  Tunnel 2
                     137:35:35::35/64
                     137:35:35::/64                                Active    loopback
loopback             ::1/128
```

output definitions

Name	Interface name. This is usually the VLAN on which the interface is configured.
IPv6 Address/Prefix Length	IPv6 address and prefix length assigned to the interface. If an interface has more than one IPv6 address assigned to it, each address is shown on a separate line.
Status	Interface status (e.g., Active/Inactive).
Device	The device on which the interface is configured (e.g., VLAN 955).

```
-> show ipv6 interface v6if-6to4-137

v6if-6to4-137
  IPv6 interface index        = 16777216(0x01000000)
  Administrative status       = Enabled
  Operational status          = Active
  Link-local address(es):
  Global unicast address(es):
    2002:d423:2323::35/64
  Anycast address(es):
    2002:d423:2323::/64
  Joined group addresses:
    ff02::1:ff00:0
    ff02::2:93da:68lb
    ff02::1
    ff02::1:ff00:35
  Maximum Transfer Unit (MTU) = 1280
  Send Router Advertisements  = No
  Maximum RA interval (sec)   = 600
  Minimum RA interval (sec)   = 198
  RA managed config flag      = False
  RA other config flag        = False
  RA reachable time (ms)      = 30000
  RA retransmit timer (ms)    = 1000
  RA default lifetime (sec)   = 1800
  Packets received            = 215686
  Packets sent                = 2019
  Bytes received              = 14108208
  Bytes sent                  = 178746
  Input errors                = 0
  Output errors               = 0
  Collisions                  = 0
  Dropped                     = 0
```

```
-> show ipv6 interface v6if-tunnel-137

v6if-tunnel-137
  IPv6 interface index        = 16777216(0x01000000)
  Administrative status       = Disabled
  Operational status          = Inactive
  Link-local address(es):
    fe80::2d0:95ff:fe12:f470/64
  Global unicast address(es):
    137:35:35:35/64
  Anycast address(es):
    137:35:35:35/64
  Joined group addresses:
    ff02::1:ff00:0
    ff02::1:ff00:35
    ff02::2:93da:68lb
    ff02::1
    ff02::1:ff12:f470
  Maximum Transfer Unit (MTU) = 1280
  Send Router Advertisements  = Yes
  Maximum RA interval (sec)   = 600
  Minimum RA interval (sec)   = 198
  RA managed config flag      = False
  RA other config flag        = False
  RA reachable time (ms)      = 30000
  RA retransmit timer (ms)    = 1000
  RA default lifetime (sec)   = 1800
  Packets received            = 0
  Packets sent                = 2
  Bytes received              = 0
  Bytes sent                  = 144
  Input errors                = 0
  Output errors               = 2
  Collisions                  = 0
  Dropped                     = 0
```

output definitions

IPv6 interface index	IPv6IfIndex value that should be used in SNMP requests pertaining to this interface.
Administrative status	Administrative status of this interface (Enabled/Disabled).
Operational status	Indicates whether the physical interface is connected to a device (Active/Inactive).
Hardware address	Interface’s MAC address
Link-local address	Link-local address assigned to the interface.
Global unicast address(es)	Global unicast address(es) assigned to the interface.
Joined group address(es)	Addresses of the multicast groups that this interface has joined.
Maximum Transfer Unit	Interface MTU value.
Send Router Advertisements	Indicates if the router sends periodic router advertisements and responds to router solicitations on the interface.
Maximum RA interval (sec)	Maximum time between the transmission of unsolicited router advertisements over the interface.
Minimum RA interval (sec)	Minimum time between the transmission of unsolicited router advertisements over the interface (0.33 * Maximum RA Interval).

output definitions

RA managed config flag	True/False value in the managed address configuration flag field in router advertisements.
RA other config flag	The True/False value in the other stateful configuration flag field in router advertisements sent over this interface.
RA reachable time (ms)	Value placed in the reachable time field in the router advertisements sent over this interface.
RA retransmit timer (ms)	Value placed in the retransmit timer field in router advertisements sent over this interface.
RA default lifetime (ms)	The value placed in the router lifetime field in the router advertisements sent over this interface.
Packets received	Number of IPv6 packets received since the last time the counters were reset.
Packets sent	Number of IPv6 packets sent since the last time the counters were reset
Bytes received	Number of bytes of data received since the last time the counters were reset.
Bytes sent	Number of bytes of data sent since the last time the counters were reset.
Input errors	Number of input errors received since the last time the counters were reset.
Output errors	Number of output errors received since the last time the counters were reset.
Collisions	Number of collisions since the last time the counters were reset.
Dropped	Number of packets dropped since the last time the counters were reset

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 address	Configures an IPv6 address on a VLAN, configured tunnel, or a 6to4 tunnel.
ipv6 interface	Configures an IPv6 interface on a VLAN.

MIB Objects

```
ipv6InterfaceTable
  ipv6AdminStatus
  ipv6PhysicalAddress
  ipv6InterfaceAddress
  ipv6Address
  ipv6AddressPrefix
  ipv6IfEffectiveMtu
  ipv6IfStatsInReceives
  ipv6IfStatsOutRequests
  ipv6IfStatsOutForwDatagrams
```

```
alaIPv6InterfaceTable
   alaIPv6InterfaceName
   alaIPv6InterfaceAddress
   alaIPv6InterfaceAdminStatus
   alaIPv6InterfaceRowStatus
   alaIPv6InterfaceDescription
   alaIPv6InterfaceMtu
   alaIPv6InterfaceType
   alaIPv6InterfaceAdminStatus
   alaIPv6InterfaceSendRouterAdvertisements
   alaIPv6InterfaceMaxRtrAdvInterval
   alaIPv6InterfaceAdvManagedFlag
   alaIPv6InterfaceAdvOtherConfigFlag
   alaIPv6InterfaceAdvReachableTime
   alaIPv6InterfaceAdvRetransTimer
   alaIPv6InterfaceAdvDefaultLifetime
   alaIPv6InterfaceName
   alaIPv6InterfaceAdvSendMtu
```

show ipv6 pmtu table

Displays the IPv6 Path MTU Table.

show ipv6 pmtu table

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

N/A.

Example

```
-> show ipv6 pmtu table

1-PMTU Entry
Destination Address                                       MTU      Expires
--------------------------------------------------------+--------+-------
fe80::02d0:c0ff:fe86:1207                                 1280     1h 0m
```

output definitions

Destination Address	IPv6 address of the path's destination.
MTU	Path’s MTU.
Expires	Minimum remaining lifetime for the entry.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 pmtu-lifetime	Configures the configure the minimum lifetime for entries in the path MTU Table.
clear ipv6 pmtu table	Removes all entries from the IPv6 path MTU Table.

MIB Objects

```
alaIPv6ConfigTable
   alaIPv6PMTUDest
   alaIPv6PMTUexpire
```

clear ipv6 pmtu table

Removes all entries from the IPv6 path MTU Table.

clear ipv6 pmtu table

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

N/A.

Example

```
-> clear ipv6 pmtu table
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 pmtu-lifetime	Configures the configure the minimum lifetime for entries in the path MTU Table.
show ipv6 pmtu table	Displays the IPv6 path MTU Table.

MIB Objects

```
alaIPv6ConfigTable
   alaIpv6ClearPMTUTable
```

show ipv6 neighbors

Displays IPv6 Neighbor Table.

show ipv6 neighbors [*ipv6_prefix/prefix_length* | *if_name* | **hw** *hardware_address* | **static]**

Syntax Definitions

ipv6_prefix/prefix_length	IPv6 prefix. Restricts the display to those neighbors starting with the specified prefix.
if_name	Interface name. Restricts the display to those neighbors reached via the specified interface.
hardware_address	MAC address. Restricts the display to the specified MAC address.
static	Restricts display to statically configured neighbors.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

If you do not specify an option (e.g., if_name), all IPv6 neighbors are displayed.

Example

```
-> show ipv6 neighbors

IPv6 Address                 Hardware Address    State     Type    Port  Interface
----------------------------+-------------------+----------+-------+-----+---------
fe80::02d0:c0ff:fe86:1207    00:d0:c0:86:12:07   Probe     Dynamic  1/15   vlan_4
fe80::020a:03ff:fe71:fe8d    00:0a:03:71:fe:8d   Reachable Dynamic  1/ 5   vlan_17
```

output definitions

IPv6 Address	The neighbor's IPv6 address.
Hardware Address	The MAC address corresponding to the IPv6 address.
State	The neighbor's state: - **Unknown** - **Incomplete** - **Reachable** - **Stale** - **Delay** - **Probe**.
Type	Indicates whether the neighbor entry is a **Static** or **Dynamic** entry.
Port	The port used to reach the neighbor.
Interface	The neighbor's interface name (e.g., vlan_1)

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 neighbor	Configures a static entry in the IPv6 Neighbor Table.

MIB Objects

```
ipv6IfIndex
alaIPv6NeighborTable
  alaIPv6NeighborNetAddress
  alaIPv6NeighborPhysAddress
  alaIPv6NeighborSlot
  alaIPv6NeighborPort
  alaIPv6NeighborType
  alaIPv6NeighborState
```

clear ipv6 neighbors

Removes all entries, except static entries, from the IPv6 Neighbor Table.

clear ipv6 neighbors

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

This commands only clears dynamic entries. If static entries have been added to the table, they must be removed using the **no** form of the **ipv6 neighbor** command.

Example

```
-> clear ipv6 neighbors
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 neighbor	Configures a static entry in the IPv6 Neighbor Table.
show ipv6 neighbors	Displays IPv6 Neighbor Table.

MIB Objects

```
alaIPv6NeighborTable
   alaIPv6ClearNeighbors
```

show ipv6 prefixes

Displays IPv6 prefixes used in router advertisements.

show ipv6 prefixes

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

N/A.

Example

```
-> show ipv6 prefixes

Legend: Flags: A = Autonomous Address Configuration, L = OnLink

                                                 Valid     Preferred
Name            IPv6 Address/Prefix Length    Lifetime   Lifetime   Flags   Source
--------------+-----------------------------+----------+----------+-------+---------
vlan 955        212:95:5::/64                  2592000    604800     LA     dynamic
vlan 1002       195:35::/64                    2592000    604800     LA     dynamic
6to4tunnel      2002:d423:2323::/64            2592000    604800     LA     dynamic
tunnel 2        137:35:35::/64                 2592000    604800     LA     dynamic
```

output definitions

Name	The interface name. This is usually the VLAN on which the interface is configured.
IPv6 Address/Prefix Length	The IPv6 prefix and prefix length for a Router Advertisement Prefix Option.
Valid Lifetime	Length of time, in seconds, that this prefix will remain valid (i.e., time until deprecation). A value of 4,294,967,295 represents infinity.
Preferred Lifetime	Length of time, in seconds, that this prefix will remain preferred (i.e. time until deprecation). A value of 4,294,967,295 represents infinity.
Flags	**L -** Prefix can be used for onlink determination. **A -** Prefix can be used for autonomous address configuration (i.e., can be used to form a local interface address).
Source	**config -** Prefix has been configured by management. **dynamic -** Router Advertisements are using interface prefixes.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 prefix	Configures an IPv6 prefix on an interface. Used for configuring prefixes for router advertisements.

MIB Objects

```
IPv6AddrPrefixTable
   IPv6AddressPrefixEntry
   IPv6AddressPrefixLength
   IPv6AddressPrefixLinkFlag
   IPv6AddressPrefixAdvvalidLifetime
   IPv6AddressPrefixAdvPreferredLifetime
alaIPv6InterfacePrefixTable
   alaIPv6InterfacePrefix
   alaIPv6InterfacePrefixLength
   alaIPv6InterfacePrefixValidLifetime
   alaIPv6InterfacePrefixPreferredLifetime
   alaIPv6InterfacePrefixOnLinkFlag
   alaIPv6InterfacePrefixsource
```

show ipv6 routes

Displays IPv6 Forwarding Table.

show ipv6 routes [*ipv6_prefix/prefix_length* | **static]**

Syntax Definitions

ipv6_prefix/prefix_length	IPv6 prefix. Restricts the display to those routes starting with the specified prefix.
static	Restricts display to statically configured routes.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

If you do not specify an option (e.g., “static”), all IPv6 interfaces are displayed.

Example

```
-> show ipv6 routes

Legend:Flags:U = Up, G = Gateway, H = Host, S = Static, C = Cloneable, D = Dynamic,
             M = Modified, R = Unreachable, X = Externally resolved, B = Discard,
             L = Link-layer, 1 = Protocol specific, 2 = Protocol specific

Destination Prefix  Gateway Address           Interface         Age          Protocol  Flags
-------------------+----------------+--------+-----------------+------------+---------+-----
::/0                2002:d468:8a89::137       v6if-6to4-137     18h 47m 26s  Static    UGS
137:35:35::/64      fe80::2d0:95ff:fe12:f470  v6if-tunnel-137   18h 51m 55s  Local     UC
195:35::/64         fe80::2d0:95ff:fe12:f470  v6if-to-eagle     18h 51m 55s  Local     UC
212:95:5::/64       fe80::2d0:95ff:fe12:f470  smbif-5           18h 51m 55s  Local     UC
2002::/16           2002:d423:2323::35        v6if-6to4-137     18h 51m 55s  Other     U
```

output definitions

Destination Prefix	IPv6 destination address and prefix.
Gateway Address	IPv6 address of the gateway used to reach the destination network.
Interface	The device the interface is using (e.g., VLAN 6to4tunnel); or loopback.
Age	Age of the entry. Entries less than 1 day old are displayed in hh:mm:ss format. Entries more than 1 day old are displayed in dd:hh format.
Protocol	Protocol by which the route was learned.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 route	Configures a static entry in the IPv6 route.

MIB Objects

```
IPv6RouteTable
   IPv6Routes
   IPv6RoutesPrefix
   IPV6RoutesStatic
alaIPv6StaticRouteTable
   alaIPv6StaticRouteEntry
```

show ipv6 tcp ports

Displays TCP Over IPv6 Connection Table. This table contains information about existing TCP connections between IPv6 endpoints.

show ipv6 tcp ports

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Only connections between IPv6 addresses are contained in this table.

Example

```
-> show ipv6 tcp ports

Local Address        Port Remote Address                   Port   Interface      State
--------------------+-------------------------------------+------+--------------+--------
::                   21 ::                                 0                     listen
::                   23 ::                                 0                     listen
2002:d423:2323::35   21 212:61:61:0:2b0:doff:fe43:d4f8     34144  v6if-6to4-137  established
2002:d423:2323::35   49153 212:61:61:0:2b0:d0ff:fe43:d4f8  34144  v6if-6to4-137  established
```

output definitions

Local Address	Local address for this TCP connection. For ports in the “Listen” state, which accepts connections on any IPv6 interface, the address is ::0.
Port	Local port number for the TCP connection.
Remote Address	Remote IPv6 address for the connection. If the connection is in the “Listen” state, the address is ::0.
Port	Remote port number for the TCP connection. If the connection is in the “Listen” state, the port number is 0.
Interface	Name of the interface (or “unknown”) over which the connection is established.
State	State of the TCP connection as defined in RFC 793.

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 udp ports	Displays the UDP Over IPv6 Listener Table.

MIB Objects

```
IPv6TcpConnTable
   IPv6TcpConnEntry
   IPv6TcpConnLocalAddress
   IPv6TcpConnLocalPort
   IPv6TcpConnRemAddress
   IPv6TcpConnRemPort
   IPv6TcpConnIfIndex
   IPv6TcpConnState
```

show ipv6 traffic

Displays IPv6 traffic statistics.

show ipv6 traffic [*if_name*]

Syntax Definitions

if_name	Interface name. restricts the display to the specified interface instead of global statistics.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

The statistics show the cumulative totals since the last time the switch was powered on, the last reset of the switch was executed or the traffic statistics were cleared using the command.

Example

```
-> show ipv6 traffic

IPv6 statistics
  Packets received
    Total                = 598174
    Header errors        = 0
    Too big              = 12718
    No route             = 4
    Address errors       = 0
    Unknown protocol     = 0
    Truncated packets    = 0
    Local discards       = 0
    Delivered to users   = 582306
    Reassembly needed    = 0
    Reassembled          = 0
    Reassembly failed    = 0
    Multicast Packets    = 118
  Packets sent
    Forwarded            = 3146
    Generated            = 432819
    Local discards       = 0
    Fragmented           = 0
    Fragmentation failed = 0
    Fragments generated  = 0
    Multicast packets    = 265
```

output definitions

Total	Total number of input packets received, including those received in error.
Header errors	Number of input packets discarded due to errors in their IPv6 headers (e.g., version number mismatch, other format errors, hop count exceeded, errors discovered in processing their IPv6 options).
Too big	Number of input packets that could not be forwarded because their size exceeded the link MTU of the outgoing interface.
No route	Number of input packets discarded because no route could be found to transmit them to their destination.
Address errors	Number of input packets discarded because the IPv6 address in their IPv6 header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., ::0) and unsupported addresses (e.g., addresses with unallocated prefixes).
Unknown protocol	Number of locally-addressed packets received successfully but discarded because of an unknown or unsupported protocol.
Truncated packets	Number of input packets discarded because the packet frame did not carry enough data.
Local discards	Number of input IPv6 packets for which no problems were encountered to prevent their continued processing, ut which were discarded (e.g., for lack of buffer space). Note that this counter does not include any packets discarded while awaiting re-assembly.
Delivered to users	Total number of packets successfully delivered to IPv6 user protocols (including ICMP).
Reassembly needed	Number of IPv6 fragments received that needed to be reassembled.
Reassembled	Number of IPv6 packets successfully reassembled.
Reassembly failed	Number of failures detected by the IPv6 reassembly algorithm (for whatever reason: timed out, errors, etc.).
Multicast packets	Number of multicast packets received.
Forwarded	Number of output packets that this entity received and forwarded to their final destinations.
Generated	Total number of IPv6 packets that local IPv6 user-protocols (including ICMP) supplied to IPv6 in requests for transmission. Note that this counter does not include any packets counted by the Forwarded statistic.
Local discards	Number of output IPv6 packets for which no problem was encountered to prevent their transmission to their destination, but were discarded (e.g., for lack of buffer space). Note that this counter would include packets counted by the Forwarded statistic if any such packets met this (discretionary) discard criterion.
Fragmented	Number of IPv6 packets successfully fragmented.
Fragmentation failed	Number of IPv6 packets discarded because they needed to be fragmented but could not be.
Fragments generated	Number of output packet fragments generated as a result of fragmentation.
Multicast packets	Number of multicast packets transmitted.

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 icmp statistics	Displays IPv6 ICMP statistics.

MIB Objects

```
ipv6IfStatsTable
   ipv6IfStatsInReceives
   ipv6IfStatsInHdrErrors
   ipv6IfStatsInTooBigErrors
   ipv6IfStatsInNoRoutes
   ipv6IfStatsInAddrErrors
   ipv6IfStatsInUnknownProtos
   ipv6IfStatsInTruncatedPkts
   ipv6IfStatsInDiscards
   ipv6IfStatsInDelivers
   ipv6IfStatsOutForwDatagrams
   ipv6IfStatsOutRequests
   ipv6IfStatsOutDiscards
   ipv6IfStatsOutFragOKs
   ipv6IfStatsOutFragFails
   ipv6IfStatsOutFragCreates
   ipv6IfStatsReasmReqds
   ipv6IfStatsReasmOKs
   ipv6IfStatsReasmFails
   ipv6IfStatsInMcastPkts
   ipv6IfStatsOutMcastPkts
```

clear ipv6 traffic

Resets all IPv6 traffic counters.

clear ipv6 traffic

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use the **show ipv6 traffic** command to view current IPv6 traffic statistics.

Example

```
-> clear ipv6 traffic
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 traffic	Displays IPv6 traffic statistics..

MIB Objects

```
alaIPv6ConfigTable
  alaIPv6ClearTraffic
```

show ipv6 tunnel

Displays IPv6 Tunnel information and whether the 6to4 tunnel is enabled.

show ipv6 tunnel

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

N/A.

Example

```
-> show ipv6 tunnel

IPv6 6to4 tunnel: Enabled
Configured Tunnels:
Tunnel              IPv6 Address/Prefix Length    Source IPv4     Destination IPv4
-------------------+-----------------------------+---------------+-----------------
1                   2001:0000:0200::101/48        192.16.10.101   192.28.5.254
23                  2001:0000:0200::102/48        192.15.10.102   10.27.105.25
v6if-tunnel-137     fe80::2d0:95ff:fe12:f470/64   212.35.35.35    212.104.138.137
```

output definitions

IPv6 6to4 tunnel	Indicates whether 6to4 tunneling is enabled or disabled on the switch.
Tunnel	Tunnel ID.
IPv6 Address/Prefix Length	IPv6 address associated with the tunnel.
Source IPv4	Source IPv4 address for the tunnel.
Destination IPv4	Destination IPv4 address for the tunnel.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 interface tunnel source destination	Configures the source and destination IPv4 addresses for a configured tunnel.

MIB Objects

```
alaIPv6ConfigTunnelTable
   alaIPv6Tunnel6to4
   alaIPv6ConfigTunnelv4Source
   alaIPv6ConfigTunnelv4Dest
```

show ipv6 udp ports

Displays the UDP Over IPv6 Listener Table. This table contains information about UDP/IPv6 endpoints.

show ipv6 udp ports

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Only endpoints utilizing IPv6 addresses are displayed in this table.

Example

```
-> show ipv6 udp ports

Local Address            Port    Interface
------------------------+-------+--------------------
```

output definitions

Local Address	Local IPv6 address for this UDP listener. If a UDP listener accepts packets for any IPv6 address associated with the switch, the value is ::0.
Port	Local Port number for the UDP connection.
Interface	Name of the interface the listener is using or “unknown.”

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 tcp ports	Displays TCP Over IPv6 Connection Table.

MIB Objects

```
IPv6UdpTable
   IPv6UdpEntry
   IPv6UdpLocalAddress
   IPv6UdpLocalPort
   IPv6UdpIfIndex
```

ipv6 load rip

Loads RIPng into memory. When the switch is initially configured, you must load RIPng into memory to enable RIPng routing.

ipv6 load rip

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- RIPng will support a maximum of 1,000 routes.
- RIPng will support a maximum of 20 interfaces.
- Use the **ipv6 rip status** command to enable RIPng on the switch.

Example

```
-> ipv6 load rip
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip status	Enables/disables RIPng routing on the switch.
show ipv6 rip	Displays RIPng status and general configuration parameters.

MIB Objects

```
alaDrcTmConfig
  alaDrcTmIPRipngStatus
```

ipv6 rip status

Enables/disables RIPng on the switch.

ipv6 rip status {enable | disable}

Syntax Definitions

N/A

Defaults

parameter	default
enable | disable	enable

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

RIPng must be loaded on the switch (**ipv6 load rip**) to enable RIP on the switch.

Example

```
-> ipv6 rip status enable
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 load rip	Loads RIPng into memory.
show ipv6 rip	Displays RIPng status and general configuration parameters.

MIB Objects

```
alaProtocolripng
   alaRipngProtoStatus
```

ipv6 rip invalid-timer

Configures the amount of time a route remains active in RIB before being moved to the "Garbage" state.

ipv6 rip invalid-timer *seconds*

Syntax Definitions

seconds	Time, in seconds, that a route will remain in an “Active” state. Valid range is 1 - 300.

Defaults

parameter	default
seconds	180

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

This timer is reset each time a routing update is received.

Example

```
-> ipv6 rip invalid-timer 300
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip garbage-timer	Configures the RIPng garbage timer value.
ipv6 rip holddown-timer	Configures the amount of time a route is placed in a holddown state.

MIB Objects

```
alaProtocolripng
  alaRipngInvalidTimer
```

ipv6 rip garbage-timer

Configures the RIPng garbage timer value. When a route in the RIB exceeds the configured Invalid Timer Value, the route is moved to a “Garbage” state in the the RIB. The garbage timer is the length of time a route will stay in this state before it is flushed from the RIB.

ipv6 rip garbage-timer *seconds*

Syntax Definitions

seconds	Time, in seconds, that a route will remain in the RIPng Routing Table before it is flushed from the RIB. Valid range is 0 - 180.

Defaults

parameter	default
seconds	120

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use the **ipv6 rip invalid-timer** command to set the Invalid Timer Value.

Example

```
-> ipv6 rip garbage-timer 180
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip invalid-timer	Configures the amount of time a route remains active in RIB before being moved to the "Garbage" state.
ipv6 rip holddown-timer	Configures the amount of time a route is placed in a holddown state.

MIB Objects

```
alaProtocolripng
  alaRipngGarbageTimer
```

ipv6 rip holddown-timer

Configures the amount of time a route is placed in a holddown state. Whenever a route is seen from the same gateway with a higher metric than the route in the RIB, the route goes into holddown. This excludes route updates with an INFINITY metric.

ipv6 rip holddown-timer *seconds*

Syntax Definitions

seconds	Time, in seconds, that a route will remain in a holddown state. Valid range is 0 - 120.

Defaults

parameter	default
seconds	0

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

While in holddown, the route continues being announced as usual and used in the RIB. This interval is used to control route flap dampening.

Example

```
-> ipv6 rip holddown-timer 60
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip invalid-timer	Configures the amount of time a route remains active in RIB before being moved to the "Garbage" state.
ipv6 rip garbage-timer	Configures the RIPng garbage timer value.

MIB Objects

```
alaProtocolripng
   alaRipngHolddownTimer
```

ipv6 rip jitter

Configures an offset value for RIPng updates. This is the maximum (positive or negative) value that can be used to offset the update interval. For example, with an update interval of 30 seconds, and a jitter value of 5 seconds, the RIPng update packet would be sent somewhere (random) between 25 and 35 seconds from the previous update.

ipv6 rip jitter *value*

Syntax Definitions

value	Time, in seconds, that a routing update is offset. Valid range is 0 to one-half the updated interval value (e.g., if the updated interval is 30, the range would be 0 - 300).

Defaults

parameter	default
value	5

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

As you increase the number of RIPng interfaces/peers, it is recommended that you increase the Jitter value to reduce the number of RIPng updates being sent over the network.

Example

```
-> ipv6 rip jitter 10
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip update-interval	Configures the RIPng update interval.
show ipv6 rip	Displays RIPng status and general configuration information.

MIB Objects

```
alaProtocolripng
  alaRipngJitter
```

ipv6 rip route-tag

Configures the route tag value for RIP routes generated by the switch.

ipv6 rip route-tag *value*

Syntax Definitions

value	Route tag value. Valid range is 0 – 65535.

Defaults

parameter	default
value	0

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

This value does not apply to routes learned from other routers. For these routes, the route tag propagates with the route.

Example

```
-> ipv6 rip route-tag 30
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 rip	Displays RIPng status and general configuration information.

MIB Objects

```
alaProtocolripng
  alaRipngRouteTag
```

ipv6 rip update-interval

Configures the RIPng update interval. This is the interval, in seconds, that RIPng routing updates will be sent out.

ipv6 rip update-interval *seconds*

Syntax Definitions

seconds Interval, in seconds, that RIPng routing updates are sent out. Valid range is 0 - 120.

Defaults

parameter	default
seconds	30

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

Use this command, along with the **ipv6 rip jitter** command to configure RIPng updates.

Example

```
-> ipv6 rip update-interval 30
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip jitter	Configures an offset value for RIPng updates.
show ipv6 rip	Displays RIPng status and general configuration information.

MIB Objects

```
alaRipng
   alaRipngUpdateInterval
```

ipv6 rip triggered-sends

Configures the behavior of triggered updates.

ipv6 rip triggered-sends {all | updated-only | none}

Syntax Definitions

all All RIPng routes are added to any triggered updates.

updated-only Only route changes that are causing the triggered update are included in the update packets.

none RIPng routes are not added to triggered updates.

Defaults

parameter	default
all | updated-only | none	updated-only

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- If set to "all", all routes are sent in the update, not just route changes, which increases RIPng traffic on the network.
- If set to "none", no triggered updates are sent, which can cause delays in network convergence.

Example

```
-> ipv6 rip triggered-sends none
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 rip Displays RIPng status and general configuration information.

MIB Objects

```
alaProtocolripng
  alaRipngTriggeredSends
```

ipv6 rip interface

Creates/deletes a RIPng interface.

ipv6 rip interface *if_name*

[no] ipv6 rip interface *if_name*

Syntax Definitions

if_name	IPv6 interface name.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- By default, a RIPng interface is created in the enabled state.
- Routing is enabled on a VLAN when you create a router port. However, to enable RIPng routing, you must also configure and enable a RIPng routing interface on the VLAN's IP router port. For more information on VLANs and router ports, see Chapter 21, "VLAN Management Commands."
- RIPng will support a maximum of 20 interfaces.

Example

```
-> ipv6 rip interface Test_Lab
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 load rip	Loads RIPng into memory.
ipv6 rip status	Enables/disables RIPng on the switch.
ipv6 rip interface recv-status	Configures IPv6 RIPng interface “Receive” status. When this status is set to "enable", packets can be received on this interface.
ipv6 rip interface send-status	Configures IPv6 RIPng interface “Send” status. When this status is set to "enable", packets can be sent on this interface.
show ipv6 rip interface	Displays information for all or specified RIPng interfaces.

MIB Objects

```
alaRipngInterfaceTable
   alaRipngInterfaceStatus
```

ipv6 rip interface metric

Configures the RIPng metric or cost for a specified interface. You can set priorities for routes generated by a switch by assigning a metric value to routes generated by that switch's RIPng interface. For example, routes generated by a neighboring switch may have a hop count of 1. However, you can lower the priority of routes generated by that switch by increasing the metric value for routes generated by the RIPng interface.

ipv6 rip interface *if_name* **metric** *value*

Syntax Definitions

if_name	IPv6 interface name.
value	Metric value. Valid range is 1 - 15.

Defaults

parameter	default
value	1

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

When you configure a metric for a RIPng interface, this metric cost is added to the metric of the incoming route.

Example

```
-> ipv6 rip Test_Lab metric 1
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip interface	Creates/deletes a RIPng interface.
show ipv6 rip interface	Displays information for all or specified RIPng interfaces.

MIB Objects

```
alaRipngInterfaceTable
   alaRipngInterfaceMetric
```

ipv6 rip interface recv-status

Configures IPv6 RIPng interface “Receive” status. When this status is set to "enable", packets can be received on this interface. When it is set to "disable", packets will not be received on this interface.

ipv6 rip interface *if_name* **recv-status {enable | disable}**

Syntax Definitions

if name	IPv6 interface name.
enable | disable	Interface “Receive” status.

Defaults

parameter	default
enable | disable	enable

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

RIPng must be loaded (**ipv6 load rip**) and enabled (**ipv6 rip status**)on the switch to send or receive packets on the interface.

Example

```
-> ipv6 rip interface Test_Lab recv-status disable
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 load rip	Loads RIPng into memory.
ipv6 rip status	Enables/disables RIPng on the switch.
ipv6 rip interface send-status	Configures IPv6 RIPng interface “Send” status.

MIB Objects

```
alaRipngInterfaceTable
   alaRipngInterfaceRecvStatus
```

ipv6 rip interface send-status

Configures IPv6 RIPng interface “Send” status. When this status is set to "enable", packets can be sent from this interface. When it is set to "disable", packets will not be sent from this interface.

ipv6 rip interface *if_name* **send-status {enable | disable}**

Syntax Definitions

if name	IPv6 interface name.
enable | disable	Interface “Send” status.

Defaults

parameter	default
enable | disable	enable

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

RIPng must be loaded (**ipv6 load rip**) and enabled (**ipv6 rip status**)on the switch to send or receive packets on the interface.

Example

```
-> ipv6 rip interface Test_Lab send-status enable
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 load rip	Loads RIPng into memory.
ipv6 rip status	Enables/disables RIPng on the switch.
ipv6 rip interface recv-status	Configures IPv6 RIPng interface “Receive” status.

MIB Objects

```
alaRipngInterfaceTable
   alaRipngInterfaceSendStatus
```

ipv6 rip interface horizon

Configures the routing loop prevention mechanisms.

ipv6 rip interface *if_name* **horizon {none | split-only | poison}**

Syntax Definitions

if_name IPv6 interface name.

none | split-only | poison

none - Disables loop prevention mechanisms.
split-only - Enables split-horizon, without poison-reverse.
poison - Enables split-horizon with poison-reverse.

Defaults

parameter	default
none | split-only | poison	poison

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- If set to “none” the route is not sent back to the peer.
- If set to ‘split-only”, the route received from the peer is sent back with an increased metric.
- If set to “poison” the route received from the peer is sent back with an “infinity” metric.

Example

```
-> ipv6 rip interface Test_Lab none
```

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 rip interface	Displays information for all or specified RIPng interfaces.
show ipv6 rip routes	Displays all or a specific set of routes in the RIPng Routing Table.

MIB Objects

```
alaRipngInterfaceTable
  alaRipngInterfaceHorizon
```

ipv6 rip debug-level

Configures the RIPng debug level for all debug types.

ipv6 rip debug-level *level*

Syntax Definitions

level	Debug level. Valid range is 0 - 255.

Defaults

parameter	default
level	0

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- This command sets the debug level for **all** configured types. You cannot set different levels for each type.
- Use the **ipv6 rip debug-type** command to specify the type of RIPng messages to debug.
- When the debug level is set to 0, the log is turned off.

Example

```
-> ipv6 rip debug-level 50
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip debug-type	Configures the type of RIPng messages to debug.

MIB Objects

```
alaRipngDebug
  alaRipngDebugLevel
```

ipv6 rip debug-type

Configures the type of RIPng messages to debug.

ipv6 rip debug-type [error] [warning] [recv] [send] [rdb] [age] [mip] [info] [setup] [time] [tm] [all]

Syntax Definitions

error	Includes error conditions, failures, processing errors, etc.
warning	Includes general warnings, non-fatal conditions.
recv	Enables debugging in the receive flow path of the code.
send	Enables debugging in the send flow path of the code.
rdb	Debugs RIP database handling.
age	Debugs code handling database entry aging/timeouts.
mip	Debugs RIPng MIP messages.
info	Provides general information.
setup	Provides information during initialization.
time	Debugs timeout handler.
tm	Debugs RIPng Task Manager messages.
all	Enables all debug options.

Defaults

N/A

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

- Use the **no** form of the command to disable a debug type.
- You can configure more than on debug type in the same command (see example below).
- Use the **ipv6 rip debug-level** command to set the debug level. This command sets the debug level for **all** configured types. You cannot set different levels for each type.

Example

```
-> ipv6 rip debug-type error warning recv send
```

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip debug-level	Configures the RIPng debug level.

MIB Objects

```
alaRipngDebug
   alaRipngDebugError
   alaRipngDebugWarn
   alaRipngDebugRecv
   alaRipngDebugSend
   alaRipngDebugRdb
   alaRipngDebugAge
   alaRipngDebugMip
   alaRipngDebugInfo
   alaRipngDebugSetup
   alaRipngDebugTime
   alaRipngDebugTm
   alaRipngDebugAll
```

show ipv6 rip

Displays RIPng status and general configuration parameters.

show ipv6 rip

Syntax Definitions

N/A

Defaults

N/A

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

N/A

Examples

```
-> show ipv6 rip

Status              = Enabled,
Number of routes    = 10,
Route tag           = 0,
Update interval     = 30,
Invalid interval    = 180,
Garbage interval    = 120,
Holddown interval   = 0,
Jitter interval     = 5,
Triggered Updates   = All Routes,
```

output definitions

Status	RIPng protocol status (enabled or disabled).
Number of routes	Number of RIPng routes in Forwarding Information Base (FIB).
Route tag	Route tag value for RIP routes generated by the switch. Valid range is 0-65535. Default is 0.
Invalid interval	Invalid Timer setting, in seconds.
Garbage interval	Garbage Timer setting, in seconds.
Holddown interval	Holddown Timer setting, in seconds.
Jitter interval	Jitter setting.
Triggered updates	Triggered Updates setting (All Routes, Updated Routes, None).

Release History

Release 5.1; command was introduced.

Related Commands

ipv6 rip status	Enables/disables RIPng routing on the switch.
ipv6 rip route-tag	Configures the route tag value for RIP routes generated by the switch.
ipv6 rip update-interval	Configures the Interval, in seconds, that RIPng routing updates are sent out.
ipv6 rip invalid-timer	Configures the amount of time a route remains active in RIB before being moved to the "garbage" state.
ipv6 rip invalid-timer	Configures the RIPng garbage timer value. Routes move into the garbage collection state because the timer expired or a route update with an INFINITY metric was received.
ipv6 rip holddown-timer	Configures the amount of time a route is placed in a holddown state.
ipv6 rip jitter	Configures an offset value for RIPng updates. This is the maximum (positive or negative) value that can be used to offset the update interval.
ipv6 rip triggered-sends	Configures the behavior of triggered updates.

MIB Objects

```
alaRipngInterfaceTable
   alaRipngInterfaceStatus
   alaRipngRouteTag
   laRipngInvalidTimer
   alaRipngGarbageTimer
   alaRipngHolddownTimer
   alaRipngJitter
   alaRipngTriggeredSends
```

show ipv6 rip interface

Displays information for all or specified RIPng interfaces.

show ipv6 rip interface [*if_name*]

Syntax Definitions

if_name	IPv6 interface name.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

If you do not specify an interface, all IPv6 RIP interfaces are displayed.

Example

```
-> show ipv6 rip interface

    Interface                       Packets
       Name          Status     Recvd   Sent     Metric
--------------------+----------+---------+--------+---------
Test_Lab            Active     12986     12544    1
Test_Lab_2          Active     12556     12552    1


-> show ipv6 rip interface if3

Name                           = Test_Lab,
IPv6 interface index           = 3,
Interface status               = Active,
Next Update                    = 27 secs,
Horizon Mode                   = Split and Poison-reverse,
MTU size                       = 1500,
Metric                         = 1,
Send status                    = Enabled,
Receive status                 = Enabled,
Packets received               = 12986,
Packets sent                   = 12544,
```

output definitions

Interface name	Interface name.
IPv6 interface index	IPv6 index of this interface.
Status	Interface status (Active/Inactive).
Packets Recvd	Number of packets received by the interface.

output definitions

Packets Sent	Number of packets sent by the interface.
Metric	RIPng metric (cost) configured for the interface.
IPv6 interface index	IPv6 interface index number.
Interface status	Interface status (Active/Inactive).
Next update	Seconds remaining until the next update on this interface.
Horizon mode	Interface Horizon Mode (routing loop prevention mechanisms). Displayed modes are none/split-only/poison-reverse.
MTU size	Maximum transmission size for RIPng packets on the interface.
Send status	Interface “Send” status. When this status is set to "enable", packets can be sent from this interface. When it is set to "disable", packets will not be sent from this interface.
Receive status	Interface “Receive” status. When this status is set to "enable", packets can be received by this interface. When it is set to "disable", packets cannot be received by this interface.
Packets received	Number of packets received by the interface.
Packets sent	Number of packets sent by the interface.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip interface	IPv6 interface name.
ipv6 rip status	Enables/disables RIPng routing on the switch.
ipv6 rip interface recv-status	Configures the interface “Receive” status. When this status is set to "enable", packets can be received by this interface. When it is set to "disable", packets cannot be received by this interface.
ipv6 rip interface send-status	Configures the interface “Send” status. When this status is set to "enable", packets can be sent from this interface. When it is set to "disable", packets will not be sent from this interface.
ipv6 rip interface metric	Configures the RIPng metric (cost) for the interface.
ipv6 rip interface horizon	Configures the interface Horizon Mode (routing loop prevention mechanisms).
show ipv6 rip	Displays RIPng status and general configuration parameters (e.g., force holddown timer).

MIB Objects

```
alaRipngInterfaceTable
  alaRipngInterfaceEntry
  alaRipngInterfaceStatus
  alaRipngInterfacePacketsRcvd
  alaRipngInterfacePacketsSent
  alaRipngInterfaceMetric
  alaRipngInterfaceIndex
  alaRipngInterfaceNextUpdate
  alaRipngInterfaceHorizon
  alaRipngInterfaceMTU
  alaRipngInterfaceSendStatus
  alaRipngInterfaceRecvStatus
```

show ipv6 rip peer

Displays a summary of the observed RIPng peers, or specific information about a peer when a peer address is provided.

show ipv6 rip peer [*ipv6_addresss*]

Syntax Definitions

ipv6_addresss	IPv6 address of the peer.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

If you do not specify a peer, all IPv6 RIP peers are displayed.

Example

```
-> show ipv6 peer
                              Seen on       Packets   Last
Address                       Interface     Recv      Update
-----------------------------+-------------+--------+----------
fe80::200:39ff:fe1f:710c      vlan172       23       20
fe80::2d0:95ff:fe12:da40      bkbone20      33       2
fe80::2d0:95ff:fe12:da40      vlan150       26       25
fe80::2d0:95ff:fe6a:5d41      nssa23        20       25



-> show ipv6 rip peer fe80::2d0:95ff:fe12:da40

Peer#1 address          = fe80::2d0:95ff:fe12:da40,
Seen on interface       = bkbone20,
Last Update             = 8 secs,
Received packets        = 33,
Received bad packets    = 0
Received routes         = 5,
Received bad routes     = 0

Peer#2 address          = fe80::2d0:95ff:fe12:da40,
Seen on interface       = vlan150,
Last Update             = 1 secs,
Received packets        = 27,
Received bad packets    = 0
Received routes         = 2,
Received bad routes     = 0
```

output definitions

Address	IPv6 address of the peer.
Seen on Interface	Interface used to reach the peer.
Packets Recvd	Number of packets received from the peer.
Last Update	Number of seconds since the last updated was received from the peer.
Peer address	Peer IPv6 address.
Received packets	Number of packets received from the peer.
Received bad packets	Number of bad packets received from the peer.
Received routes	Number of RIPng routes received from the peer.
Received bad routes	Number of bad RIPng routes received from the peer.

Release History

Release 5.1.6; command was introduced.

Related Commands

show ipv6 rip interface	Displays all or specified RIPng interface status
show ipv6 rip routes	Displays all or a specific set of routes in the RIPng Routing Table.

MIB Objects

```
alaRipngPeerTable
   alaRipngPeerEntry
   alaRipngPeerAddress
   alaRipngPeerIndex
   alaRipngPeerLastUpdate
   alaRipngPeerNumUpdates
   alaRipngPeerBadPackets
   alaRipngPeerNumRoutes
   alaRipngPeerBadRoutes
```

show ipv6 rip routes

Displays all or a specific set of routes in the RIPng Routing Table.

show ipv6 rip routes [dest <*ipv6_prefix/prefix_length*>**] | [gateway** <*ipv6_addr*>**] | [detail** <*ipv6 prefix/prefix_length*>**]**

Syntax Definitions

dest	Displays all routes whose destination matches the IPv6 prefix/prefix length.
gateway	Displays all routes whose gateway matches the specified IPv6 address.
detail	Displays detailed information about a single route matching the specified destination.
ipv6_addr	IPv6 address.
ipv6_prefix/prefix length	IPv6 address and prefix/prefix length.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

If you do not enter one of the optional parameters, all IPv6 RIP routes are displayed.

Example

```
-> show ipv6 rip routes

Legends: State: A = Active, H = Holddown, G = Garbage
Destination     Gateway                    State    Metric Proto
---------------+--------------------------+-------+------+------
100::1/128      +fe80::200:39ff:fe1f:710c     A       2      Rip
100::100:1/128  +fe80::200:39ff:fe1f:710c     A       2      Rip
400::/100       +fe80::2d0:95ff:fe12:e050     A       1      Local
900::/100       +fe80::2d0:95ff:fe12:e050     A       1      Local
8900::/100      +fe80::2d0:95ff:fe12:da40     A       2      Rip
9800::/100      +fe80::2d0:95ff:fe12:da40     A       2      Rip
9900::/100      +fe80::2d0:95ff:fe12:e050     A       1      Local
```

```
-> show ipv6 rip routes detail 9900::/100

Destination     = 9900::,
Mask length     = 100,
Gateway(1)      = fe80::2d0:95ff:fe12:e050,
Protocol        = Local,
Out Interface   = nssa23,
Metric          = 1,
Status          = Installed,
State           = Active,
Age             = 10544s,
Tag             = 0,
Gateway(2)      = fe80::2d0:95ff:fe12:da40,
Protocol        = Rip,
Out Interface   = bkbone20,
Metric          = 2,
Status          = Not Installed,
State           = Active,
Age             = 15s,
Tag             = 0,
```

output definitions

Destination	IPv6 address/address length of the destination.
Gateway	IPv6 gateway used to reach the destination.
State	Route status (Active/Inactive).
Metric	Routing metric for this route
Protocol	Protocol used to learn the route.
Mask Length	Prefix Length.
Out Interface	The interface used to reach the destination.
Status	Route status (Active/Inactive)
Age	The number of seconds since the route was last updated.
Tag	The route tag value for the route.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip interface	Creates/deletes a RIPng interface.
ipv6 rip interface metric	Configures the RIPng metric or cost for a specified interface.
show ipv6 rip interface	Displays all or specified RIPng interface status.

MIB Objects

```
alaRipngRouteTable
  alaRipngRouteEntry
  alaRipngRoutePrefixLen
  alaRipngRouteNextHop
  alaRipngRouteType
  alaRipngRouteAge
  alaRipngRouteTag
  alaRipngRouteStatus
  alaRipngRouteMetric
```

show ipv6 rip debug

Displays the current RIPng debug level and types.

show ipv6 rip debug

Syntax Definitions

N/A.

Defaults

N/A.

Platforms Supported

OmniSwitch 6624, 6648, 7700, 7800, 8800

Usage Guidelines

N/A.

Example

```
-> show ipv6 rip debug

Debug Level = 0,
error       = on,
warning     = off,
recv        = off,
send        = off,
rdb         = off,
age         = off,
mip         = off,
info        = off,
setup       = off,
time        = off,
tm          = off,
```

output definitions

Debug Level	Debug level. Valid range is 0 - 255. Default is 0.
Debug Type Status (on/off)	**error** - Includes error conditions, failures, processing errors, etc. **warning** - Includes general warnings, non-fatal conditions. **recv** - Enables debugging in the receive flow path of the code. **send** - Enables debugging in the send flow path of the code. **rdb** - Debugs RIP database handling. **age** - Debugs code handling database entry aging/timeouts. **mip** - Debugs RIPng MIP messages. **info** - Provides general information. **setup** - Provides information during initialization. **time** - Debugs timeout handler. **tm** - Debugs RIPng Task Manager messages. **all** - Enables all debug options.

Release History

Release 5.1.6; command was introduced.

Related Commands

ipv6 rip debug-level	Configures the RIPng debug level.
ipv6 rip debug-type	Configures the type of RIPng messages to debug.

MIB Objects

```
alaRipngDebug
  alaRipngDebugLevel
  alaRipngDebugError
  alaRipngDebugWarn
  alaRipngDebugRecv
  alaRipngDebugSend
  alaRipngDebugRdb
  alaRipngDebugAge
  alaRipngDebugMip
  alaRipngDebugInfo
  alaRipngDebugSetup
  alaRipngDebugTime
  alaRipngDebugTm
  alaRipngDebugAll
```

3 Configuring High Availability VLANs

High availability (HA) VLANs, unlike standard VLANs, allow you to send traffic intended for a single destination MAC address to multiple switch ports. These high availability VLANs can be used to manage server clusters.

In This Chapter

This chapter describes the basic components of high availability VLANs and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Creating a VLAN on page 3-9.
- Adding ingress ports to a high availability VLAN on page 3-10.
- Adding egress ports to a high availability VLAN on page 3-12.
- Adding MAC addresses to a high availability VLAN on page 3-13.
- Configuring inter-switch ports for HA VLANs on page 3-14.
- Configuring the ingress flood queue bandwidth on page 3-15

Note. You can also configure and monitor high availability VLANs with WebView, Alcatel's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to WebView's online documentation for more information on configuring and monitoring high availability VLANs with WebView.

High Availability VLANs Specifications

The table below lists specifications for high availability VLAN software.

RFCs Supported	2674—*Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*
IEEE Standards Supported	802.1D—*Media Access Control Bridges* 802.1w—*Rapid Reconfiguration (802.1D Amendment 2*) 802.1s—*Multiple Spanning Trees (802.1Q Amendment 3)*
Maximum high availability VLANs per switch	32
Switch ports eligible for high availability VLAN assignment.	Fixed ports on second-generation Network Interface (NI) modules.
Switch port *not* eligible for high availability VLAN assignment.	Mobile, 802.1Q tagged, link aggregate, Learned Port Security (LPS), mirrored or mirroring ports, and all ports on first-generation NI modules.
Spanning Tree modes supported.	Flat (one Spanning Tree instance per switch). 1x1 (one Spanning Tree instance per VLAN).
CLI Command Prefix Recognition	All high availability VLAN configuration commands with the **vlan** prefix support prefix recognition. (However, the **show mac-address-table port-mac** command does not support prefix recognition.) See the “Using the CLI” chapter in the *OmniSwitch 7700/ 7800/8800 Switch Management Guide* for more information.

High Availability Default Values

The table below lists default values for high availability VLAN software.

Parameter Description	Command	Default Value/Comments
Ingress ports assigned.	**vlan port-mac ingress-port**	No ingress ports assigned.
Egress ports assigned.	**vlan port-mac egress-port**	No egress ports assigned.
MAC addresses assigned.	**mac-address-table port-mac** **vlan mac**	No MAC addresses assigned.
high availability VLAN ingress Flood queue bandwidth	**vlan port-mac bandwidth**	15 Mbps

Quick Steps for Creating High Availability VLANs

Follow the steps below for a quick tutorial on configuring high availability (HA) VLANs. Additional information on how to configure each command is given in the sections that follow.

1 Create a default VLAN for the HA VLAN ports with the **vlan** command as shown below:

```
-> vlan 10
```

2 Assign ports to the new default VLAN with the **vlan port default** command as shown below:

```
-> vlan 10 port default 1/1, 3/2, 3/7, 6/1, 6/2
```

3 Create a VLAN that will become the HA VLAN once configured with ingress and egress ports. For example:

```
-> vlan 200
```

4 Configure ports 1/1 and 3/2 as ingress ports for HA VLAN 200. Use the **vlan port-mac ingress-port** command as shown below:

```
-> vlan 200 port-mac ingress-port 1/1 3/2
```

5 Configure ports 6/1 and 6/2 as egress ports for HA VLAN 200. Use the **vlan port-mac egress-port** command as shown below:

```
-> vlan 200 port-mac egress-port 6/1 6/2
```

6 Assign a MAC address to this high availability VLAN with the **mac-address-table port-mac vlan mac** command. For example:

```
-> mac-address-table port-mac vlan 200 mac 00:DA:29:3C:11:20
```

7 Configure port 3/7 as an inter-switch port for the HA VLAN with the **vlan 802.1q** command as shown below:

```
-> vlan 200 802.1q 3/7
```

Note that Step 7 is only done when the HA VLAN is configured across two switches. The ports that provide the connection between the two switches must be tagged with the HA VLAN ID.

8 Set the HA VLAN ingress flood queue bandwidth size to 100 Mbps. Use the **vlan port-mac bandwidth** command as shown below:

```
-> vlan 200 port-mac bandwidth 100
```

Note. *Optional*. You can display the configuration of high availability VLANs with the **show mac-address-table port-mac** command. For example:

```
-> show mac-address-table port-mac 200
Port mac configuration for vlan 200

Bandwidth : 100 MB/sec

   Ingress Port list:
           1/1  3/2
   Egress Port list:
           6/1  6/2
   Mac Address list:
           00:DA:29:3C:11:20
```

An example of what these commands look like entered sequentially on the command line:

```
-> vlan 10
-> vlan 10 port default 1/1 3/2 3/7 6/1 6/2
-> vlan 200 port-mac ingress-port 1/1 3/2
-> vlan 200 port-mac egress-port 6/1 6/2
-> mac-address-table port-mac vlan 200 mac 00:DA:29:3C:11:20
-> vlan 200 802.1q 3/7
-> vlan 200 port-mac bandwidth 100
```

High Availability VLAN Overview

High availability (HA) VLANs send traffic intended for a single destination MAC address to multiple switch ports. This section provides a brief overview on how traffic flows in and out of high availability VLANs and how high availability VLANs can manage third-party high availability firewall clusters (see "High Availability Firewall Clusters" on page 3-6 for information).

An HA VLAN is configured by creating a standard VLAN and then assigning ingress or egress ports to the VLAN. Once these types of ports are assigned, the standard VLAN automatically becomes an HA VLAN. When this occurs, standard VLAN commands no longer apply.

Destination MAC addresses (unicast and multicast) are also assigned to high availability VLANs. These addresses identify ingress port traffic that the switch will send out on all egress ports that belong to the same VLAN

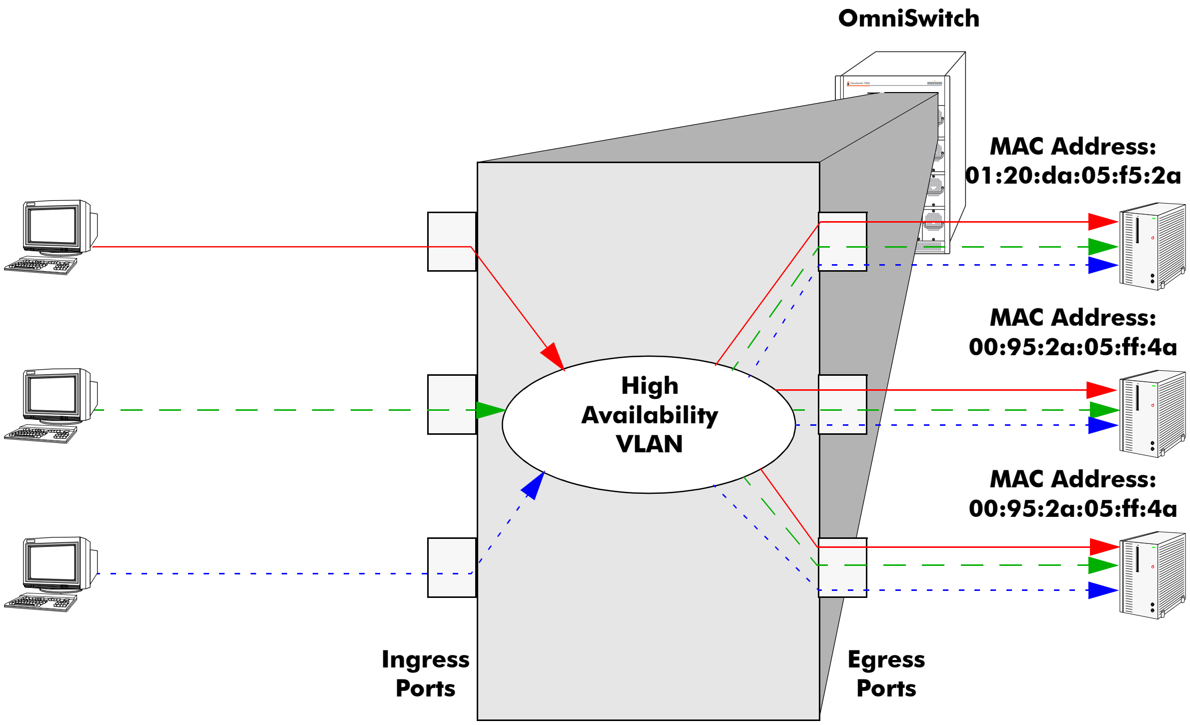
In addition to assigning ingress and egress ports, tagging inter-switch link ports with an HA VLAN ID is allowed. Ingress port traffic destined for an HA VLAN MAC address is sent out on all egress *and* inter-switch link ports that belong to the same VLAN. Traffic forwarded on inter-switch link ports is done so in accordance with the Spanning Tree state of the port.

It is also possible to configure the ingress flood queue bandwidth size for HA VLANs. See "Configuring the Flood Queue Bandwidth" on page 3-15 for more information.

Note. Once a VLAN becomes an HA VLAN, only ingress, egress, and tagged inter-switch link ports are allowed as members of that VLAN.

Ingress and Egress Traffic Flows

The figure below shows how ingress traffic is handled by high availability VLANs.



Ingress to Egress Port Flow

In the above example, packets received on the ingress ports that are destined for the high availability VLAN MAC address are sent out the egress ports that are members of the same VLAN. Since all three servers are connected to egress ports, they all receive the ingress port traffic. This provides a high level of availability in that if one of the server connections goes down, the other connections still forward traffic to one of the redundant servers.

Note the following regarding ingress and egress port traffic flow:

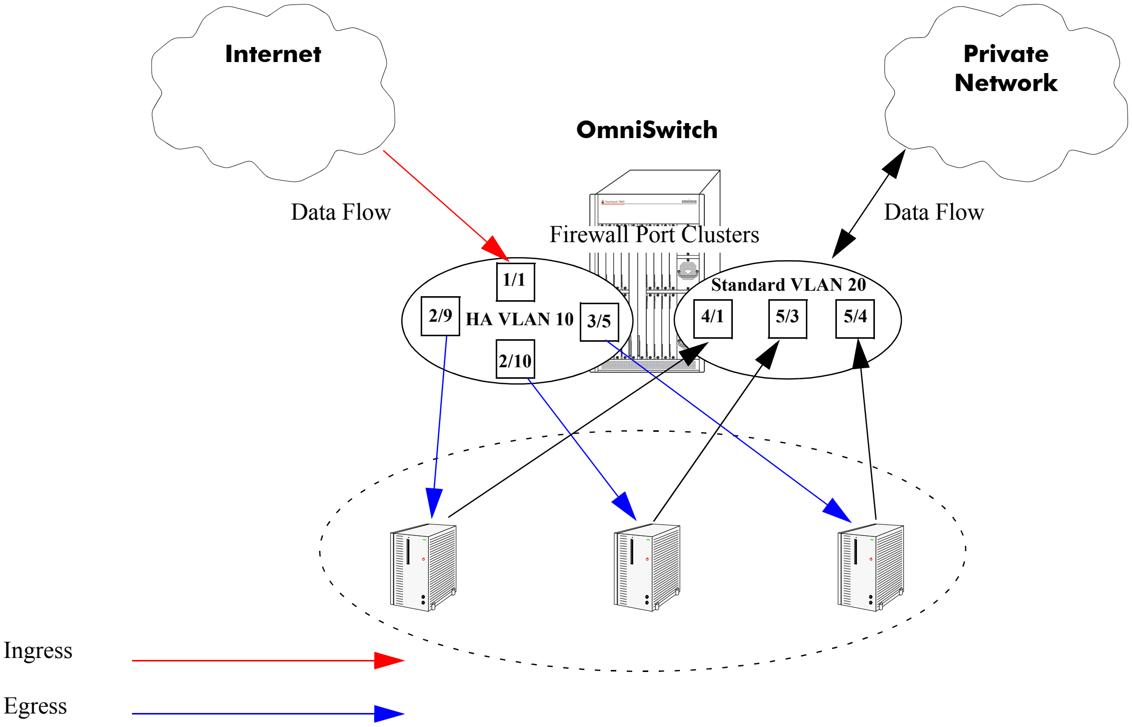
- Ingress port traffic destined for the high availability VLAN MAC address is only sent out on egress ports and not on any other ingress ports.
- If a packet received on an ingress port is not destined for the high availability VLAN MAC address, the packet is bridged as regular traffic to all ports in the VLAN, not just egress ports.
- Traffic received on egress ports is bridged as regular traffic to all ports assigned to the VLAN, regardless of their ingress or egress port state.

High Availability Firewall Clusters

One key application of high availability VLANs is interfacing with third-party high availability firewall clusters, which allow two or more servers running a common firewall application to work as if they were one system. The following subsection describes an example HA VLAN implementation that is used to interface with a third-party high availability firewall cluster.

Traditional Firewall Implementation

The figure below shows two high availability VLANs that are used to manage a third-party high availability firewall cluster. Unsecure traffic from the Internet comes into the OmniSwitch through the ingress port 1/1 of high availability VLAN 10. This traffic is sent to the high availability cluster through the egress ports that belong to HA VLAN 10 (2/9, 2/10, and 3/5).



Firewall and High Availability Cluster

The third-party high availability firewall cluster sends authorized traffic to ports 4/1, 5/3, and 5/4 that belong to standard VLAN 20. This traffic is then forwarded on VLAN 20 to the private network.

See "Application Example 1: Firewall Cluster" on page 3-16 for instructions on how to configure the high availability VLANs in the example above.

Configuring High Availability VLANs on a Switch

This section describes how to use the Command Line Interface (CLI) commands to configure high availability (HA) VLANs on a switch. For a brief tutorial on configuring HA VLANs, see "Quick Steps for Creating High Availability VLANs" on page 3-3.

When configuring HA VLANs, you must perform the following steps:

1 **Create a VLAN**. To create a VLAN use the **vlan** command, which is described in "Creating and Deleting VLANs" on page 3-9.

2 **Assign Ingress Ports.** To assign ingress ports to the high availability VLAN, use the **vlan port-mac ingress-port** command, which is described in "Assigning and Removing Ingress Ports" on page 3-10.

3 **Assign Egress Ports.** To assign egress ports to the high availability VLAN, use the **vlan port-mac egress-port** command, which is described in "Assigning and Removing Egress Ports" on page 3-12.

4 **Assign MAC Addresses**. To assign MAC addresses to the high availability VLAN, use the **mac-address-table port-mac vlan mac** command, which is described in "Assigning and Removing MAC Addresses" on page 3-13.

5 **Configure Inter-Switch Ports.** To configure an HA VLAN across two switches, use the **vlan 802.1q** command to tag the connection ports with the HA VLAN ID. This procedure is described in "Configuring Inter-switch Ports for HA VLANs" on page 3-14.

6 **Configure The Flood Queue Bandwidth**. To configure the size of the HA VLAN ingress flood queue bandwidth, use the **vlan port-mac bandwidth** command, which is described in "Configuring the Flood Queue Bandwidth" on page 3-15.

Note. You must have write access to the VLAN family of commands (i.e., the **domain-layer2** domain) to use the commands described in the following subsections. See the "Managing Switch User Accounts" in the *OmniSwitch 7700/7800/8800 Switch Management Guide* for more information.

Note the following when configuring HA VLANs:

- Only fixed ports on second-generation Network Interface (NI) modules are eligible for HA VLAN assignment. Mobile ports, 802.1Q tagged ports, link aggregate ports, Learned Port Security (LPS) ports, and ports that mirror or are mirrored are not eligible for HA VLAN use.
- All HA VLAN related ports must first belong to the same default VLAN before they are configured as ingress, egress, or inter-switch ports for the HA VLAN.
- Only ingress/egress and tagged inter-switch ports are allowed in an HA VLAN. See "Configuring Inter-switch Ports for HA VLANs" on page 3-14 for more information about inter-switch ports.
- Do not assign an HA VLAN as the default VLAN for a port. Any attempt to do so is not allowed.
- When a port is assigned to an HA VLAN as an ingress or egress port, the default VLAN assignment remains the same. For example, if VLAN 10 is the default VLAN for port 3/10 and this same port is associated with HA VLAN 200 as an ingress or egress port, VLAN 10 still remains the default VLAN for port 3/10. In addition, the **show vlan port** command only shows the VLAN 10 assignment.
- It is possible to designate a port as both an ingress and egress port.

- It is highly recommended that all switches that participate in an HA VLAN configuration run in the same Spanning Tree mode and use the same Spanning Tree protocol (STP, RSTP, MSTP).

Use the **show mac-address-table port-mac** command to verify the HA VLAN configuration on the switch. See "Displaying High Availability VLAN Status and Statistics" on page 3-19 for more information.

Creating and Deleting VLANs

The following subsections describe how to create and delete a VLAN with the **vlan** command.

Note. This chapter provides only a basic description of creating and deleting VLANs. For a complete description of configuring and monitoring VLANs on a switch, please refer to Chapter 5, "Configuring VLANs."

Creating a VLAN

To create a new VLAN use the **vlan** command by entering **vlan** followed by the VLAN ID number, which can be any integer from 2 to 4094. (Default VLAN 1 is part of the standard switch configuration and does not need to be created.) For example, to create a VLAN with a VLAN ID number of 10 enter

```
-> vlan 10
```

You can also specify the administrative status and a name for the VLAN with the **vlan** command. For example, to administratively enable (the default) a VLAN when you configure it enter **vlan** followed by the VLAN ID number and **enable**.

For example, to create vlan 10 and administratively enable it enter

```
-> vlan 10 enable
```

To administratively disable a VLAN when you configure it enter **vlan** followed by the VLAN ID number and **disable**.

For example, to create vlan VLAN 10 and administratively disable it enter

```
-> vlan 10 disable
```

To assign a name to a VLAN when you configure it enter **vlan** followed by the VLAN ID number, **name**, and a text description, which can be up to 32 characters long.

Note. If a text description has spaces the name must be enclosed within quotes (e.g., **"VLAN 10"**)

For example, to create VLAN 10 and name it "VLAN10" enter

```
-> vlan 10 name VLAN10
```

Note. You can use the **name** keyword with the **enable** and **disable** keywords (e.g., **vlan 10 enable name VLAN10**).

Deleting a VLAN

To delete a VLAN use the **no** form of the **vlan** command by entering **no vlan** followed by the VLAN's ID number. For example, to delete high availability VLAN 10 enter:

```
-> no vlan 10
```

Assigning and Removing Ingress Ports

The following subsections describe how to assign to and remove ingress ports from a high availability VLAN with the **vlan port-mac ingress-port** command.

Note. Using the **vlan port-mac ingress-port** command will change a standard VLAN to a high availability VLAN. Standard VLAN commands do not apply to high availability VLANs.

Assigning Ingress Ports

To assign ingress ports to a high availability VLAN you use the **vlan port-mac ingress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac ingress-port**, the slot number of the port, a slash (/), and the port number.

For example, to add ingress port 3/2 to high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac ingress-port 3/2
```

You can also add multiple ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac ingress-port**, the slot number of the first port, a slash (/), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be added by specifying their slot number, a slash, and their port number preceded by a space.

Note. Ingress ports that belong to the same high availability VLAN do not need to be sequential and can be on different second-generation NI modules.

For example, to add ingress port 3/2, 5/1, and 7/4 to high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac ingress-port 3/2 5/1 7/4
```

You can also add a range of ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac ingress-port**, the slot number of the first port, a slash (/), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to add ingress port 2/1 through 2/5 to high availability VLAN 10 enter:

```
-> vlan 10 port-mac ingress-port 2/1-5
```

You can add multiple and ranges of ingress ports in the same command line. For example, to add ingress ports 2/1 through 2/5, 3/2, 5/1, and 7/4 to high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac ingress-port 2/1-5 3/2 5/1 7/4
```

Removing Ingress Ports

To remove ingress ports from a high availability VLAN you use the **no** form of the **vlan port-mac ingress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac no ingress-port**, the slot number of the port, a slash (/), and the port number.

For example, to remove ingress port 3/2 from high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac no ingress-port 3/2
```

You can also remove multiple ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no ingress-port**, the slot number of the first port, a slash (/), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be removed by specifying their slot number, a slash, and their port number preceded by a space.

For example, to remove ingress port 3/2, 5/1, and 7/4 from high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac no ingress-port 3/2 5/1 7/4
```

You can also remove a range of ingress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no ingress-port**, the slot number of the first port, a slash (/), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to remove ingress port 2/1 through 2/5 from high availability VLAN 10 enter:

```
-> vlan 10 port-mac no ingress-port 2/1-5
```

You can remove multiple and ranges of ingress ports in the same command line. For example, to remove ingress ports 2/1 through 2/5, 3/2, 5/1, and 7/4 from high availability VLAN 10 you would enter:

```
-> vlan 10 port-mac no ingress-port 2/1-5 3/2 5/1 7/4
```

Note. Removing the last ingress/egress port from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one ingress/egress port left in the VLAN.

Assigning and Removing Egress Ports

The following subsections describe how to assign to and remove egress ports from a high availability VLAN with the **vlan port-mac egress-port** command.

Note. Using the **vlan port-mac egress-port** command will change a standard VLAN to a high availability VLAN. Standard VLAN commands do not apply to high availability VLANs.

Assigning Egress Ports

To assign egress ports to a high availability VLAN you use the **vlan port-mac egress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac egress-port**, the slot number of the port, a slash (/), and the port number.

For example, to add egress port 1/5 to high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac egress-port 1/5
```

You can also add multiple egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac egress-port**, the slot number of the first port, a slash (/), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be added by specifying their slot number, a slash, and their port number preceded by a space.

Note. Egress ports that belong to the same high availability VLAN do not need to be sequential and can be on different second-generation NI modules.

For example, to add egress port 1/5, 6/2, and 8/3 to high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac egress-port 1/5 6/2 8/3
```

You can also add a range of egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac egress-port**, the slot number of the first port, a slash (/), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to add egress port 3/4 through 3/8 to high availability VLAN 20 enter:

```
-> vlan 20 port-mac egress-port 3/4-8
```

You can add multiple and ranges of egress ports in the same command line. For example, to add egress ports 1/5, 3/4 through 3/8, 6/2, and 8/3 to high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac egress-port 1/5 3/4-8 6/2 8/3
```

Removing Egress Ports

To remove egress ports from a high availability VLAN, use the **no** form of the **vlan port-mac egress-port** command by entering **vlan**, followed by the VLAN's ID number, **port-mac no egress-port**, the slot number of the port, a slash (/), and the port number.

For example, to remove egress port 1/5 from high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac no egress-port 1/5
```

You can also remove multiple egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no egress-port**, the slot number of the first port, a slash (/), the port number of the first port, a space, the slot number of the second port, a slash, and the port number of the second port. Additional ports can also be removed by specifying their slot number, a slash, and their port number preceded by a space.

For example, to remove egress port 1/5, 6/2, and 8/3 from high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac no egress-port 1/5 6/2 8/3
```

You can also remove a range of egress ports by entering **vlan**, followed by the VLAN's ID number, **port-mac no egress-port**, the slot number of the first port, a slash (/), the port number of the first port on the NI, a hyphen (**-**), and the last port on the NI.

For example, to remove egress port 3/4 through 3/8 from high availability VLAN 20 enter:

```
-> vlan 20 port-mac no egress-port 3/4-8
```

You can remove multiple and ranges of egress ports in the same command line. For example, to remove egress ports 1/5, 3/4 through 3/8, 6/2, and 8/3 from high availability VLAN 20 you would enter:

```
-> vlan 20 port-mac no egress-port 1/5 3/4-8 6/2 8/3
```

Note. Removing the last ingress/egress port from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one ingress/egress port left in the VLAN.

Assigning and Removing MAC Addresses

The following subsections describe how to assign and remove MAC addresses from a high availability VLAN with the **mac-address-table port-mac vlan mac** command. Traffic that is received on ingress ports that contains a destination MAC address that matches the high availability VLAN address is sent out all egress ports that belong to the high availability VLAN.

Note. Using the **mac-address-table port-mac vlan mac** command will change a standard VLAN to a high availability VLAN. Standard VLAN commands do not apply to high availability VLANs.

Assigning MAC Addresses

To assign a MAC address to a high availability VLAN, use the **mac-address-table port-mac vlan mac** command by entering **mac-address-table port-mac vlan**, followed by the VLAN's ID number, **mac**, and the MAC address. Note that both unicast and multicast addresses are supported.

For example, to assign the MAC address 00:25:9a:5c:2f:10 to high availability VLAN 20 you would enter:

```
-> mac-address-table port-mac vlan 20 mac 00:25:9a:5c:2f:10
```

To add more than one MAC address to a high availability VLAN, enter each address on the same command line separated by a space. For example, to assign MAC addresses 00:25:9a:5c:2f:11, 00:25:9a:5c:12, and 01:00:00:3f:4c:10, to high availability VLAN 30, you would enter:

```
-> mac-address-table port-mac vlan 30 mac 00:25:9a:5c:2f:11 00:25:9a:5c:12
01:00:00:3f:4c:10.
```

Removing MAC Addresses

To remove a MAC address associated with a high availability VLAN, use the **no** form of the **mac-address-table port-mac vlan mac** command. For example, the following command removes MAC address 00:25:9a:5c:2f:10 from VLAN 20:

```
-> mac-address-table port-mac vlan 20 no mac 00:25:9a:5c:2f:10
```

To remove more than one MAC address from a high availability VLAN using a single command, enter each address on the same command line separated by a space. For example, to remove MAC addresses 00:25:9a:5c:2f:11, 00:25:9a:5c:12, and 01:00:00:3f:4c:10, from high availability VLAN 30, you would enter:

```
-> mac-address-table port-mac vlan 30 no mac 00:25:9a:5c:2f:11 00:25:9a:5c:12
01:00:00:3f:4c:10.
```

Note. Removing the last MAC address from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one MAC address left.

Configuring Inter-switch Ports for HA VLANs

One scenario using HA VLANs involves configuring ingress ports on one switch and egress ports on a second switch. In this case, the ports that connect the two switches require an 802.1Q tag for the HA VLAN.

The **vlan 802.1q** command is used to tag a port with an HA VLAN. Once this is done, the tagged ports are identified as inter-switch ports and will carry HA VLAN traffic between the two switches. See "Application Example 2: Inter-Switch HA VLANs" on page 3-17 for an example of using inter-switch ports.

Note the following regarding inter-switch ports:

- Fixed ports, 802.1Q tagged ports, and link aggregates on a second-generation module are eligible to become inter-switch ports for HA VLANs.
- Once a link aggregate is tagged with an HA VLAN, it is not possible to add any more member ports to the aggregate.
- The HA VLAN and the default VLAN for an inter-switch port should participate in the same Spanning Tree instance, especially when there are redundant inter-switch connections. If this is not possible—such as when the 1x1 mode is active and there is one instance per VLAN—then ensure that the Spanning Tree state of the inter-switch port remains the same for both VLANs.

For more information about 802.1Q tagging, see Chapter 10, "Configuring 802.1Q." For more information about Spanning Tree modes and instances, see Chapter 6, "Configuring Spanning Tree Parameters," and Chapter 3, "Using 802.1s Multiple Spanning Tree."

Configuring the Flood Queue Bandwidth

When the first HA VLAN is created on the switch, an ingress flood queue is automatically created for HA VLAN traffic. By default, the bandwidth size of this queue is set to 15mbps. To change the bandwidth size, use the **vlan port-mac bandwidth** command. For example, the following command sets the bandwidth value for HA VLAN 200 to 100mbps:

```
-> vlan 200 port-mac bandwidth 100
```

Note that when removing HA VLANs from the switch configuration, the flood queue remains in existence until the last HA VLAN is removed.

Application Example 1: Firewall Cluster

This section describes how to configure the traditional firewall implementation, which uses a third-party high availability firewall cluster, described in "Traditional Firewall Implementation" on page 3-7. As shown in the figure on page 3-7, traffic from the Internet comes into the switch through high availability VLAN 10 ingress ports. This VLAN has three egress ports (2/9, 2/10, and 3/5) that connect to the third-party high availability firewall cluster. The firewall cluster is connected to three ports (4/1, 5/3, 7/6) that belong to standard VLAN 20. This VLAN connects to devices within a private network.

Follow the steps below to configure the necessary high availability VLAN on an OmniSwitch.

1 Create a default VLAN for HA VLAN 10 ports with the **vlan** command as shown below:

```
-> vlan 5
```

2 Assign ports to the new default VLAN with the **vlan port default** command as shown below:

```
-> vlan 5 port default 1/1 2/9 2/10 3/5
```

3 Configure VLAN 10, which will have the ingress ports, with the **vlan** command as shown below:

```
-> vlan 10
```

4 Assign the ingress port 1/1 to VLAN 10 with the **vlan port-mac ingress-port** command as shown below:

```
-> vlan 10 port-mac ingress-port 1/1
```

5 Assign the egress ports 2/9, 2/10, and 3/5 to VLAN 10 with the **vlan port-mac egress-port** command as shown below:

```
-> vlan 10 port-mac egress-port 2/9-10 3/5
```

6 Configure standard VLAN 20, which will carry authorized traffic to the private network, with the **vlan** command as shown below:

```
-> vlan 20
```

7 Assign destination MAC addresses to VLAN 10 with the **mac-address-table port-mac vlan mac** command as shown below:

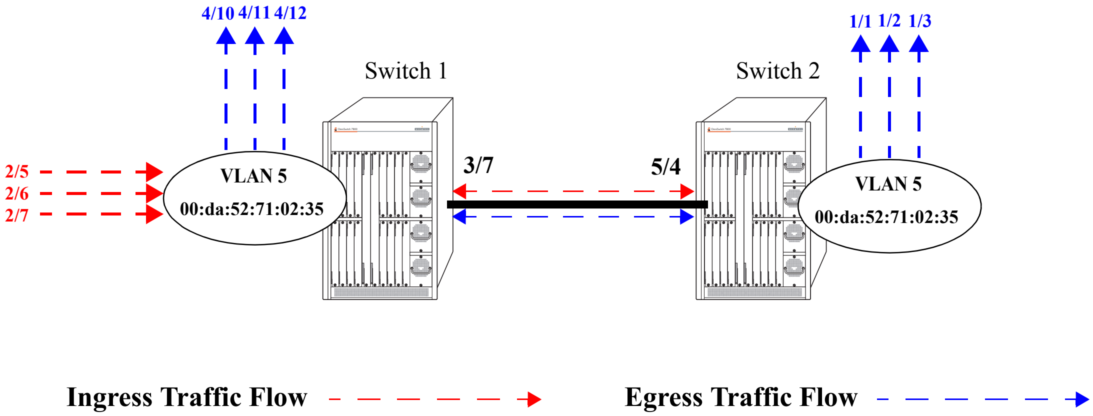
```
-> mac-address-table port-mac vlan 10 mac 00:95:2A:01:3C:10
```

Application Example 2: Inter-Switch HA VLANs

This section describes how to implement an HA VLAN configuration across two switches. As shown in the figure below:

- Ports 3/7 and 5/4 connect Switch 1 and Switch 2. Because these ports are both tagged with HA VLAN 5 (not shown), they function as inter-switch ports for VLAN 5.
- Traffic from VLAN 5 will flow in both directions through the inter-switch link provided by the 3/4 and 5/7 connection.
- VLAN 5 has three ingress ports (2/5, 2/6, and 2/7) and three egress ports (4/10, 4/11, and 4/12) on Switch 1 and three egress ports (1/1, 1/2, and 1/3) on Switch 2. In addition, VLAN 5 is also configured with an HA VLAN destination MAC address (00:da:53:71:02:35) on both switches.
- The HA VLAN ingress flood queue bandwidth size is limited to 100 Mbps.
- Traffic destined for 00:da:52:71:02:35 that is received on VLAN 5 ingress ports is forwarded to the VLAN 5 egress ports on Switch 1 *and* across the inter-switch link to VLAN 5 egress ports on Switch 2.

Note that traffic received on any ingress ports that is not destined for the HA VLAN MAC is forwarded according to switching rules.



HA VLAN Inter-Switch Configuration

Follow the steps below to configure this example inter-switch HA VLAN implementation:

1 Create a default VLAN for HA VLAN 5 ports on both Switch 1 and Switch 2 with the **vlan** command as shown below:

```
-> vlan 2
```

2 Assign ports to the new default VLAN on Switch 1 with the **vlan port default** command as shown below:

```
-> vlan 2 port default 2/5-7 3/7 4/10-12
```

3 Assign ports to a new default VLAN on Switch 2 with the **vlan port default** command as shown below:

```
-> vlan 2 port default 1/1-3 5/4
```

4 Configure VLAN 5, which will become an HA VLAN, on both Switch 1 and Switch 2 with the **vlan** command as shown below:

```
-> vlan 5
```

5 Assign the ingress ports 2/5, 2/6, and 2/7 to VLAN 5 on Switch 1 with the **vlan port-mac ingress-port** command as shown below:

```
-> vlan 5 port-mac ingress-port 2/5-7
```

6 Assign the egress ports 4/10, 4/11, and 4/12 to VLAN 5 on Switch 1 with the **vlan port-mac egress-port** command as shown below:

```
-> vlan 5 port-mac egress-port 4/10-12
```

7 Tag port 3/7 on Switch 1 with VLAN 5 using the **vlan 802.1q** command as shown below:

```
-> vlan 5 802.1q 3/7
```

8 Assign the egress ports 1/1, 1/2, and 1/3 to VLAN 5 on Switch 2 with the **vlan port-mac egress-port** command as shown below:

```
-> vlan 5 port-mac egress-port 1/1-3
```

9 Tag port 5/4 on Switch 2 with VLAN 5 using the **vlan 802.1q** command as shown below:

```
-> vlan 5 802.1q 5/4
```

10 Assign the HA VLAN destination MAC address to VLAN 5 on both Switch 1 and Switch 2 using the **mac-address-table port-mac vlan mac** command as shown below:

```
-> mac-address-table port-mac vlan 10 mac 00:da:52:71:02:35
```

11 Set the HA VLAN ingress flood queue bandwidth size to 100 Mbps using the **vlan port-mac bandwidth** command as shown below:

```
-> vlan 5 port-mac bandwidth 100
```

Displaying High Availability VLAN Status and Statistics

You can use CLI **show** commands to display the current configuration and statistics of high availability VLANs on a switch. These commands include the following:

show mac-address-table port-mac Displays the status and configuration of high availability VLANs.

show vlan Displays a list of all VLANs configured on the switch and the status of related VLAN properties (e.g., admin and Spanning Tree status and router port definitions).

show vlan port Displays a list of VLAN port assignments.

To display the status and configuration of high availability VLANs you use the **show mac-address-table port-mac** command. To display the status and configuration of all high availability VLANs on a switch enter:

```
-> show mac-address-table port-mac
```

A screen similar to the following will be displayed:

```
Port mac configuration for vlan 10

Bandwidth : 15 MB/sec

   Ingress Port list:
           3/5  3/7
   Egress Port list:
           3/9  3/6
   Mac Address list:
           00:DA:95:3C:44:55
           00:13:14:34:5E:78
           01:23:45:C1:17:21

Port mac configuration for vlan 20

Bandwidth : 15 MB/sec

   Ingress Port list:
           1/4  8/2
   Egress Port list:
           3/9  3/6
   Mac Address list:
           00:11:22:33:44:55
           07:23:14:34:31:25
           00:23:45:67:43:04
```

To display the status and configuration of a single high availability VLAN enter **show mac-address-table port-mac vlan** followed by the VLAN's ID number. For example, to display the status and configuration of high availability VLAN 10 enter

```
-> show mac-address-table port-mac vlan 10
```

A screen similar to the following will be displayed:

```
Port mac configuration for vlan 10

Bandwidth : 15 MB/sec

  Ingress Port list:
          3/5  3/7
  Egress Port list:
          3/9  3/6
  Mac Address list:
          00:11:22:33:44:55
          00:13:14:34:34:78
          01:23:45:67:11:21
```

Note. See the *OmniSwitch CLI Reference Guide*.for complete syntax for the **show mac-address-table port-mac** command.