ACOS 6.0.8
Network Configuration Guide

December, 2025

Information in this document is subject to change without notice.

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at:
a10-virtual-patent-marking.

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Layer 2 Networking

The following chapters are covered in this part/section:

Link Trunking

Link Layer Discovery Protocol

L2 Protocols: STP / RSTP and MSTP

Virtual LAN Support

Virtual Wire

Link Trunking

This chapter describes how to configure trunk links on the ACOS device.

The following topics are covered:

Overview

The ACOS device supports aggregation of multiple Ethernet data ports into logical links, called "trunks". Trunks can enhance performance by providing higher throughput and greater link reliability.

Higher throughput is provided by the aggregate throughput of the individual links in the trunk. Greater link reliability is provided by the multiple links in the trunk. If an individual port in the trunk goes down, the trunk link continues to operate using the remaining up ports in the trunk.

You can configure the following types of trunks:

- Static trunks

- Dynamic trunks – You can enable Link Aggregation Control Protocol (LACP) on Ethernet data interfaces, to make those interfaces candidate members of dynamically configured trunks.

  Link Aggregation Control Protocol (LACP) dynamically creates trunk links. The ACOS implementation of LACP is based on the 802.3ad IEEE specification. You can configure a maximum of 16 LACP trunks on an ACOS device. An interface can belong to a single LACP trunk.

NOTE: The number of trunks supported and number of ports that can be configured per trunk vary depending on the specific device. In the CLI, use the ? help command to determine the allowable values. In the GUI, the allowable ranges are visible in the configurable fields.

For example, for ACOS Thunder Bare Metal:

- The maximum number of ethernet ports that can be put into a trunk group is: 8

- The maximum number of trunk groups that can be supported on a Bare Metal is: 11.

NOTE: Interface parameters for a trunk apply collectively to all ports in trunk, as a single interface. For example, IP addresses and other IP parameters apply to the entire trunk as a single interface.

Feedback

Trunk Parameters

This section describes the parameter that can be configured for a trunk.

The following topics are covered:

Interface-Level Parameters for Trunks

After you add a trunk to the configuration, you can configure the trunk as an Ethernet data interface. The following interface-level parameters can be configured on trunk interfaces.

- Trunk Interface Name – You can assign a name to the trunk, in addition to the numeric ID you specify when you create the trunk. The name can be 1-63 characters in length, can contain numbers, upper case and lower case characters, and must not include the following symbols: ~!@#$%^&*()_+|}{:"<>?

- IPv4 and IPv6 parameters – You can assign one or more IPv4 and IPv6 addresses, and configure other IP-related parameters such as IP helper or IPv6 neighbor discovery.

- Dynamic routing – You can configure interface-level OSPF and IS-IS parameters.

- Access list (ACL) – You can filter incoming traffic based on source and destination IPv4 or IPv6 address and protocol port, as well as additional parameters such as ICMP type and code or VLAN ID.

NOTE:

○ Ethernet information is only retrieved from the `enable-management` service and not from the access-list.

○ For access-list configurations, only the `eq` operator is supported for IP ports. Currently, operators such as `lt` or `gt` are not supported and will be ignored.

- ICMP rate limiting – You can enable protection against distributed denial-of-service (DDoS) attacks such as Smurf attacks, which consist of floods of spoofed broadcast ping messages.

- Layer 3 forwarding – Layer 3 forwarding is enabled by default. You can disable it.

  If you want to allow Layer 3 forwarding except between VLANs, a separate option allows you to disable Layer 3 forwarding between VLANs.

- Port threshold – Minimum number of individual member ports that must be Up in order for the trunk to be Up. (See Port-Threshold Parameters.)

- Virtual Wire Interface - You can enable ethernet as a virtual wire interface and configure virtual wire endpoints as ethernet, trunk, or virtual wire ethernet group.

NOTE: The **disable** and **enable** commands at the interface configuration level for the trunk control Layer 3 forwarding on the trunk but do not completely disable the trunk. To control all forwarding on the trunk, use the **disable** or **enable** command at the trunk configuration level instead.

For more information about these commands, see *Command Line Interface Reference*.

Port-Threshold Parameters

By default, a trunk's status remains UP so long as at least one of its member ports is up. You can change the ports threshold of a trunk to 2-8 ports.

If the number of up ports falls below the configured threshold, the ACOS device automatically disables the trunk's member ports. The ports are disabled in the

running-config. The ACOS device also generates a log message and an SNMP trap, if these services are enabled.

NOTE:	After the feature has disabled the members of the trunk group, the ports are not automatically re-enabled. The ports must be re-enabled manually after the issue that caused the ports to go down has been resolved.

In some situations, a timer is used to delay the ports-threshold action. The configured port-threshold is not enforced until the timer expires. The ports-threshold timer for a trunk is used in the following situations:

- When a member of the trunk links up.

- A port is added to or removed from the trunk.

- The port-threshold for the trunk is configured during run time. (If the threshold is set in the startup-config, the timer is not used.)

- The port-threshold must be configured on one side of the interface trunk. If the port-threshold is configured on both sides, when any of the member interfaces is Up or Down, the port-threshold timer runs on both sides. When the timer expires, the trunk interface flaps between the port-threshold timers of the two endpoints.

LACP Parameters

By default, a trunk's status remains Up so long as at least one of its member ports is up. You can change the ports threshold of a trunk to 2-8 ports.

Since a trunk comprises of several member links, if the number of operational members of a trunk goes below the configured threshold value, the remaining member links are automatically marked as "blocked" and the trunk is considered non--operational. When the down link is functional again, the remaining links that were marked blocked are also operational again, making the trunk available for use.

NOTE:	If you administratively disable the LACP feature from members of the trunk group, the links are not automatically re-enabled. The links must be re-enabled manually after the issue that caused the links to go down has been resolved.

The following LACP parameters are configurable:

- Global LACP Parameter

- Interface-Level LACP Parameters

- Unidirectional Link Detection

Global LACP Parameter

- LACP system priority – Specifies the LACP priority of the ACOS device. In cases where LACP settings on the local device (the ACOS device) and the remote device at the other end of the link differ, the settings on the device with the higher priority are used.

You can specify 1-65535. A low priority number indicates a high priority value. The highest priority is 1 and the lowest priority is 65535. The default is 32768.

Interface-Level LACP Parameters

In addition to the interface-level parameters you can configure on static trunk interfaces, LACP trunk interfaces have the following parameters:

- LACP trunk ID – ID of a dynamic trunk. Adding an interface to an LACP trunk makes that interface a candidate for membership in the trunk. During negotiation with the other side of the link, LACP selects the interfaces to actively participate in the link. When you add an interface, you must specify whether LACP will run in active or passive mode on the interface. Active mode initiates link formation with the other end of the link. Passive mode waits for the other end of the link to initiate link formation. The admin key must match on all interfaces in the trunk. The value can be 1-4096.

- LACP port priority – Priority of the interface for selection as an active member of a link. If the LACP trunk has more candidate members than are allowed by the device at the other end of the link, LACP selects the interfaces with the highest port priority values as the active interfaces. The other interfaces are standbys, and are used only if an active interface goes down. You can specify 1-65535. A low priority number indicates a high priority value. The highest priority is 1 and the lowest priority is 65535. The default is 32768.

- LACP timeout – Aging timeout for LACP data units from the other end of the LACP link. You can specify short (3 seconds) or long (90 seconds). The default is long.

- Mode – Indicate whether you want LACP to operate in Active or Passive Mode. The Active mode initiates link formation with the other end of the link. In this case, the ACOS device will send the LACP frame to its link partner. Passive mode waits for the other end of the link to initiate link formation. In this case, the ACOS device will only send an LACP frame if it receives an LACP frame from the link partner.

- Admin Key – The admin key must match on all interfaces in the trunk. The value can be 10000-65535.

- Unidirectional Link Detection (UDLD) – UDLD checks the links in LACP trunks to ensure that both the send and receive sides of each link are operational. UDLD can only be configured on the single port LACP trunk. UDLD is not supported on multilink LACP trunks. (For more information, see Unidirectional Link Detection.)

Unidirectional Link Detection

When UDLD is enabled, the UDLD uses LACP protocol packets as heartbeat messages. If an LACP link on the ACOS device does not receive an LACP protocol packet within a specified timeout, LACP blocks traffic on the port. This corrects the problem by forcing the devices connected by the non-operational link to use other, fully operational links.

A link that is blocked by LACP can still receive LACP protocol packets but blocks all other traffic.

UDLD is disabled by default on LACP trunk links. You can enable UDLD on individual LACP trunk interfaces.

Heartbeat Timeout

The local port waits for a configurable timeout to receive an LACP protocol packet from the remote port. If an LACP protocol packet does not arrive before the timeout expires, LACP disables the local port. You can set the timeout to 1-60 seconds (slow timeout) or 100-1000 milliseconds (fast timeout). The default is 1 second.

If the remote port begins sending LACP protocol packets again, LACP on the local port re-enables the port.

Feedback

Requirements

To operate properly, UDLD must be supported and enabled on both devices that are using LACP trunk links.

It is recommended to use auto-negotiation on each end of the link to establish the mode (half duplex or full duplex). Auto-negotiation helps ensure link bidirectionality at Layer 1, while UDLD helps at Layer 2.

Monitor Trunk Interface Statistics

The trunk interface statistics help monitoring the performance and usage of trunk interfaces. The statistics for the trunk are based on incoming and outgoing packets. ACOS displays the trunk interface statistics for Layer 2 (L2) and Layer 3 (L3) trunks.

In multi-PU platforms, the aggregated statistics from Processing Unit 1 (PU1) and Processing Unit 2 (PU2) are displayed on PU1, while the statistics specific to PU2 are displayed only on PU2.

To enable the generation of statistics for trunk interfaces, you can use the `trunk-stats` command. For more information, see *Command Line Interface Reference*. The command to enable trunk statistics generation is:

```
ACOS(config)# trunk-stats enable
```

NOTE: The interface trunk must be configured along with the trunk-group to view the interface statistics under the command `show interface statistics`.

The following example shows L2 trunk interface statistics:

```
ACOS#show interfaces trunk 1
Trunk 1 is down, line protocol is down
 Members: 4
 0 packets input  0 bytes
 Received  0 broadcasts, Received 0 multicasts, Received 0 unicasts
 0 packets output  0 bytes
 Transmitted  0 broadcasts, Transmitted 0 multicasts, Transmitted 0
unicasts
```

The following example shows L3 trunk interface statistics:

```
ACOS(config-if:trunk:2)#show interfaces trunk 2
Trunk 2 is down, line protocol is down
 Hardware is TrunkGroup, Address is 001f.a020.04db
 Internet address is 1.1.1.1, Subnet mask is  255.255.255.0
 IP MTU is 1500 bytes 0 packets input  0 bytes
 Received  0 broadcasts, Received 0 multicasts, Received 0 unicasts
 0 packets output  0 bytes
 Transmitted  0 broadcasts, Transmitted 0 multicasts, Transmitted 0
unicasts
```

The following command clears the statistics on trunk interface:

```
ACOS#clear statistics interface trunk <num>
```

Static Trunk Configuration

This section provides steps for configuring a static trunk:

- Using the GUI to Configure a Static Trunk

- Using the CLI to Configure a Static Trunk

An overview of the procedure for creating a trunk:

1. Add individual Ethernet data ports to the trunk.

2. Configure the trunk as a single interface.

Using the GUI to Configure a Static Trunk
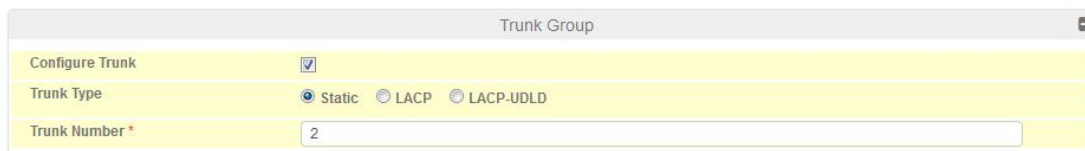
To configure a static trunk on an Ethernet interface:

1. Configuring the Trunk

2. Configuring the Minimum Port Threshold

Configuring the Trunk

1. Hover over **Network** in the navigation bar, and select **Interface**.

2. Check the menu bar to be sure you're on the LAN page.

3. Click **Edit** in the Actions column for an Ethernet interface.

4. Find the Trunk Group section and click the plus sign (+) icon to expand it.

   a. Click the **Configure Trunk** radio button.

   b. Select Static in the Trunk Type field.

   c. Specify a Trunk Number.

5. Repeat as needed to configure trunks on additional Ethernet interfaces.

Trunk Group	
Configure Trunk	☑
Trunk Type	◉ Static ○ LACP ○ LACP-UDLD
Trunk Number *	2

6. Click **Update** button.

Configuring the Minimum Port Threshold

To configure the trunk's port threshold and port threshold timer:

1. Click **Trunk** on the menu bar.

2. Click **Edit** in the Actions column for the trunk interface.

3. In the General fields section, do the following:

   a. In the Port Threshold field, specify a value of 2-8.

   b. In the Port Threshold Timer field, indicate a timer value from 1-300 seconds.

4. Click **Update Trunk**.

Using the CLI to Configure a Static Trunk

To configure a static trunk, use the commands in this section.

1. Change the CLI to the configuration level for the interface.

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)#
```

2. Assign the interface to the trunk, using the following command:

```
ACOS(config-if:ethernet:1)# trunk-group 7
AOCS(config-if:ethernet:1-trunk-group:7)#
```

You must repeat this series of commands for each interface you want to add to a trunk.

The following commands configure trunk 7 with ports 1and 2, and verify the configuration:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# trunk-group 7
ACOS(config-if:ethernet:1-trunk-group:7)# exit
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# trunk-group 7
ACOS(config-if:ethernet:2-trunk-group:7)# show trunk
Trunk ID         : 7           Member Count: 2
Trunk Name       : None
Trunk Status     : Up
Trunk Type       : Static
Members          : 1    2
Cfg Status       : Enb Enb
Oper Status      : Up   Up
Ports-Threshold  : None
Working Lead     : 2
ACOS(config-if:ethernet:2-trunk-group:7)# exit
ACOS(config-if:ethernet:2)# exit
ACOS(config)#
```

Configuring Interface-Level Trunk Parameters

The following commands access the interface configuration level for the trunk and assign a name, an IPv6 address along with port threshold parameters to the trunk interface:

```
ACOS(config)# interface trunk 7
ACOS(config-if:trunk:7)# name exampletrunk7
ACOS(config-if:trunk:7)# ipv6 address 2001:db8::7/32
ACOS(config-if:trunk:7)# ports-threshold 2
ACOS(config-if:trunk:7)# ports-threshold-timer 100
```

Dynamic Trunk Configuration

This section provides steps for configuring a dynamic trunk.

The following topics are covered:

Using the GUI to Configure an LACP Trunk

To configure an LACP trunk:

1. Navigate to **Network > Interfaces > LAN**.

2. Click Edit in the Actions column for the Ethernet.

3. Scroll down and click Trunk Group to reveal trunk configuration options.

4. Enter the Trunk ID.

5. To configure the LACP trunk without uni-directional detection:

    a. Specify LACP as the type for the Trunk Type.

6. Click the checkbox for Uni-directional Detection:

    a. Specify LACP-UDLD for the Trunk Type.

    b. Choose Slow or Fast for UDLD Timeout. If you select Slow, specify a UDLD timeout of 1-60 seconds. If you select Fast, specify a UDLD timeout of 100-1000ms.

7. Specify Active or Passive mode in the Mode field.

8. Specify an Admin Key.

9. Choose a Timeout value of Long or Short.

10. Specify the LACP priority in the Port Priority field.

11. Click **Update**.

Configuring the LACP System Priority

To configure the LACP system priority, follow these steps:

1. Hover over **Network** in the navigation bar, and select **LACP**.

2. You can specify an LACP system priority of 1-65535. The default priority setting is 2.

3. Click **OK**.

Configuring the Minimum Port Threshold

To configure the port threshold parameters for LACP trunks, do the following:

NOTE:	These steps assume that you have already created an LACP dynamic trunk. See Using the GUI to Configure an LACP Trunk.

1. Navigate to **Network > Interfaces > Trunk**.

2. Click **Edit** in the Actions column for an existing LACP Trunk 1. The Create Trunk window appears.

3. In the Ports Threshold section, enter a value from 2-8.

4. In the Port Threshold Timer field, indicate a timer value from 1-300 seconds.

5. Click **Update Trunk**.

Verifying Port Threshold Configuration in the GUI

To verify your LACP configuration of the Port Threshold and the Port Threshold Timer, do the following:

1. Navigate to **Network > Interfaces > Trunk**.

2. The configured trunks table appears.

3. The Ports Threshold field displays the configured ports threshold.

4. The Timer field displays the configured port threshold timer.

Network >> Interface >> Trunk   Help

Search					Refresh	Clear	Enable	Disable

☐	Status	Trunk Number	Type	Members(Config/Oper Status)	Ports Threshold	Timer	Time Running	VLAN	Actions
☐	⬇	2	Dynamic (LACP)	1 ✓		2	100	Yes	✗	Edit

Previous 1 Next   1 item   Items per page: 10 ▼

Using the CLI to Configure an LACP Trunk

To configure a dynamic, use the commands in this section.

Configuring Each Interface

1. Change the CLI to the configuration level for the interface.

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)#
```

2. Assign the interface to the LACP trunk, using the following command:

```
ACOS(config-if:ethernet:1)# trunk-group 4 lacp
ACOS(config-if:ethernet:1-trunk-group:4)#
```

3. (Optional) Specify the LACP priority of the interface, using the following command:

```
ACOS(config-if:ethernet:1-trunk-group:4)# port-priority 100
```

You can specify 1-65535. The default is 32768.

4. (Optional) Specify the aging timeout for LACP data units from the other end of the LACP link, using the following command:

```
ACOS(config-if:ethernet:1-trunk-group:4)# timeout short
```

You can specify `short` (3 seconds) or `long` (90 seconds). The default is `long`.

5. (Optional) Specify the UDLD aging timeout, using the following command:

```
ACOS(config-if:ethernet:1-trunk-group:4)# udld timeout slow 1
```

You can specify `fast` (100-1000 milliseconds) or `slow` (1-60 seconds). The default is `slow 1`.

6. (Optional) Configure ports-threshold settings. Specify the minimum number of ports that must remain up, using the `ports-threshold` command at the LACP trunk configuration level of the CLI:

```
ACOS(config)# interface trunk 4
ACOS(config-if:trunk:4)# ports-threshold 2 timer 100 do-auto-recovery
ACOS(config-if:trunk:4)# exit
ACOS(config)#
```

You can specify 2-8 ports.

You can set the ports-threshold timer to 1-300 seconds. The default is 10 seconds. The **do-auto-recovery** option in this command enables automatic recovery of the trunk when the required number of ports come back up. If you omit this option, the trunk remains disabled until you re-enable it.

Configuring LACP System Priority

1. (Optional) Set the LACP system priority, using the following command at the global configuration level of the CLI:

```
ACOS(config)# lacp system-priority 32768
```

You can specify 1-65535. The default is 32768.

Configuring Interface-Level Parameters on an LACP Trunk

To configure interface-level parameters for the trunk, use the following command to access the interface configuration level for the trunk.

1. Change the CLI to the configuration level for the trunk interface.

```
ACOS(config)# interface trunk 4
ACOS(config-if:trunk:4)#
```

2. For a list of the commands applicable at this level. (For information, see *Command Line Interface Reference*.)

```
vThunder(config-if:trunk:4)# ?
access-list          Apply ACL rules to incoming packets on this
interface
 bfd                 Configure BFD (Bidirectional Forwarding
Detection)
 clear               Clear or Reset Functions
 do                  To run exec commands in config mode
 end                 Exit from configure mode
 exit                Exit from configure mode or sub mode
 icmp-rate-limit     Limit ICMP traffic to this interface
 icmpv6-rate-limit   Limit ICMPv6 traffic to this interface
 ip                  Global IP configuration subcommands
 ipv6                Global IPv6 configuration subcommands
```

```
  isis                ISIS
  l3-vlan-fwd-disable  Disable L3 forwarding between VLANs
  lw-4o6              Configure LW-4over6 interface
  mtu                 Interface mtu
  name                Name for the interface
  no                  Negate a command or set its defaults
  ports-threshold     Threshold for the minimum number of ports that
need to
                      be UP for the trunk to remain UP
  show                Show Running System Information
  snmp-server         SNMP trap source
  write               Write Configuration
  enable              Enable
  disable             Disable
vThunder(config-if:trunk:4)#
```

NOTE: The commands listed at this level depend on the device model and the ACOS software release.

For more information about these commands, see *Command Line Interface Reference*.

Link Layer Discovery Protocol

The Link Layer Discovery Protocol (LLDP) enables network devices to advertise their identity, capabilities, and neighbors on the network. This feature is based on the IEEE 802.1AB standard and the standard MIB called "LLDP-V2-MIB."

For more information, refer to the following URLs:

- http://www.mibdepot.com/cgi-bin/getmib3.cgi?win=mib_ a&i=1&n=IP-MIB&r=vmware&f=LLDP-V2- MIB.mib&v=v2&t=def

- http://www.ieee802.org/1/files/public/MIBs/LLDP-V2-MIB-200906080000Z.txt

The following topics are covered:

Overview of LLDP

LLDP allows ACOS devices to discover directly-connected LAN neighbors and allows these neighbors to discover the ACOS devices. Configure LLDP only in the shared partition.

Use the LLDP protocol to assist in the following ways:

- To discover remote networks.

- To facilitate port association.

- To help identify which port a switch or a host is connected to.

- To help design and troubleshoot network topologies.

Since the LLDP protocol can transmit or receive information on system capabilities, but cannot request specific information from an LLDP agent or acknowledge receipt of information, it is called a "one-way protocol."

NOTE:	This feature does not support aXAPI.

The Link Layer Discovery Protocol Data Unit (LLDPDU) contains several elements of variable lengths that comprise the LLCP frame. They carry information on the type, length, and value fields (TLVs), where type identifies the kind of information that is transmitted, length contains the string of octets, and value is the actual content that is being transmitted. The mandatory information that is transmitted identifies the TLV for the chassis ID, the port ID, the Time to Live, and the end of the LLDP data packet. It can also contain zero or more optional TLVs. For the duration of an operational port, the chassis ID and the port ID information will remain the same.

A Time to Live TLV or a non-zero TLV informs the receiving LLDP agent to discard the LLDP data packet after the indicated time expires. A zero TLV directs the receiving LLDP agent to discard the LLDP packet immediately. As the name suggests, the End of LLDP data packet indicates that completion of the LLDP packet.

Configuring LLDP

This section describes how to configure LLDP.

The following topics are covered:

Using the GUI to Configure LLDP

To configure this feature using the GUI:

1. To enable the LLDP feature globally:

    a. Navigate to **Network > Interfaces > LLDP**.

    b. Select the **Enable** checkbox in the Enable field.

    c. Optionally, enable RX using the Rx field.

    d. Optionally, enable TX using the Tx field.

2. To enable LLDP on the interface:

    a. Navigate to **Network > Interfaces > LAN**.

    b. Click **Edit** in the Actions column for the interface.

    c. Click LLDP to expand additional configuration options.

    d. Select the Rt Enable field.

    e. Optionally, select the Rx field.

    f. Optionally, select the Tx field.

Using the CLI to Configure LLDP

To enable the LLDP feature via the CLI, enable the feature from the global level:

```
ACOS(config)# lldp enable rx tx
```

The example below shows how to enable LLDB on an interface (Ethernet 2):

```
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# lldp enable rx tx
```

The following example shows your LLDP configuration:

```
ACOS(config)# show run | inc lldp
```

Feedback

```
lldp enable rx tx
lldp notification interval 20
lldp tx interval 10
lldp tx fast-count 2
lldp tx fast-interval 2
```

The following example shows your LLDP interface configuration:

```
ACOS(config)# show run int eth 1
interface ethernet 1
ip address 7.1.1.169 255.255.255.0
lldp enable rx tx
lldp notification enable
```

L2 Protocols: STP / RSTP and MSTP

The L2 protocols (STP, RSTP, and MSTP) are supported on the ACOS platform. The xSTP protocols help in avoiding loops in L2 topologies.

The following commands configure the spanning-tree mode as STP:

```
ACOS(config)# spanning-tree mode stp
ACOS(config-stp)# enable
ACOS(config-stp)# priority 4096
```

The following command configures the VLAN in STP mode:

```
ACOS(config-stp)# vlan 100
```

The following commands configure the spanning-tree mode as RSTP:

```
ACOS(config)# spanning-tree mode rstp
ACOS(config-rstp)# enable
ACOS(config-rstp)# priority 4096
ACOS(config-rstp)# vlan 200
```

The following commands configure the spanning-tree mode as MSTP:

```
ACOS(config)# spanning-tree mode mstp
ACOS(config-mstp)# enable
ACOS(config-mstp)# revision 1 name tree
ACOS(config-mstp)# priority 4096
ACOS(config-mstp)# instance 1
ACOS(config-mstp-instance:1)# vlan 300
ACOS(config-mstp-instance:1)# priority 4096
```

Configuring Spanning-tree options on Interfaces

The Path-Cost is configured for Designated Port (DP) or Blocked Port (BP) election per layer-2 segment. The following commands configure the Ethernet interface as an admin-edge port and set the path cost on the interface:

```
ACOS(config-if:ethernet:1)#spanning-tree admin-edge
ACOS(config-if:ethernet:1)#spanning-tree path-cost 110
```

The following command configures the trunk interface as an edge-port:

```
ACOS(config-if:trunk:1)#spanning-tree admin-edge
```

The following command disables the auto-edge detection on the interface:

```
ACOS(config-if:trunk:1)#no spanning-tree auto-edge
```

The following command configures path cost on the trunk interface:

```
ACOS(config-if:trunk:1)#spanning-tree path-cost 110
```

Configuring MSTP Spanning-tree options on Interfaces

The following command configures the path cost for the MST instance ID on the interface:

```
ACOS(config-if:ethernet:1)#spanning-tree instance 1 path-cost 110
```

The following command sets the path cost for Common and Internal Spanning Tree (CIST):

```
ACOS(config-if:ethernet:1)#spanning-tree path-cost 120
```

The following command sets the no auto-edge for ethernet 1:

```
ACOS(config-if:ethernet:1)#no spanning-tree auto edge
```

Show Commands

The show command can be used to view the spanning-tree operational status:

```
ACOS# show spanning-tree
```

```
Spanning-tree Mode : RSTP
-------------
FWD ports : 2 3 4
BLK ports : 5
-------------
```

The following table describes the fields displayed under the `show spanning-tree` command:

Field	Description
Spanning-tree Mode	Configured mode on the system (STP or RSTP).
FWD ports	Ports in forward state.
BLK ports	Ports in blocked state.

The following example shows the sample output for `show spanning-tree detail`:

```
ACOS# show spanning-tree detail
Spanning-tree Mode        : RSTP
-------------
FWD ports                 : 2 3 4
BLK ports                 : 5
Bridge ID                 : 8.000.00:1F:A0:05:A2:09 (priority 32768)
Designated root           : 8.000.00:1F:A0:02:3D:99 (priority 32768)
Regional root             : 8.000.00:1F:A0:05:A2:09 (priority 32768)
Root_port                 : ethernet2
Path cost                 : 20000     Internal Path cost  : 0
Max age                   : 20        Internal max age    : 20
Root forward delay        : 15        Bridge forward delay : 15
Tx hold count             : 6
Max hops                  : 20
Bridge hello time         : 2
Age time                  : 300
Time since topology change : 977
Topology change count     : 1
Topology change port      :
Last topology change port : ethernet2
--------------
```

The following table describes the fields displayed under the `show spanning-tree detail` command:

Field	Description
Spanning-tree Mode	Configured mode on the system (STP or RSTP).
FWD ports	Ports in forward state.
BLK ports	Ports in blocked state.
Bridge ID	MAC address and priority of the switch.
Designated root	MAC address and priority of the designated port.
Regional root	MAC address and priority of the regional port.
Root port	Port with the lowest path cost to reach the root bridge.
Path cost	Metric used to calculate the shortest path to reach the root bridge.
Internal Path cost	Internal path cost.
Max age	Maximum period for which STP saves the Bridge Protocol Data Unit (BPDU) information.
Internal max age	Internal maximum period for which STP saves the BPDU information.
Root forward delay	Time spent in the listening and learning state.
Bridge forward delay	Time spent by the bridge to listen.
Tx hold count	Maximum number of BPDUs (that an interface can send in a second).
Max hops	Maximum hops for a STP.
Bridge hello time	Time interval at which a port sends each BPDU.
Age time	Time period for which STP saves the BPDU information.
Time since topology change	Time duration when the topology was changed.
Topology change	Port on which the topology is changed.

Field	Description
port	
Last topology change port	Port on which the topology was changed last.

The show command can be used to view the spanning-tree interfaces:

```
ACOS# show spanning-tree interfaces
detail: Detail
ethernet: Ethernet interface
instance: MST Instance
trunk: Trunk interface
|       : Output modifiers
```

The following table describes the fields displayed under the `show spanning-tree interfaces` command:

Field	Description
detail	Details about spanning-tree interfaces.
ethernet	Ethernet interface running the spanning-tree protocol.
trunk	Trunk interface running the spanning-tree protocol.

The following example shows the sample output for `show spanning-tree interfaces`

```
ethernet2
  ------------------------------------------------
  CIST
  ----------------
  Enabled           yes                role                 Root
  Num TX BPDU       4                  Num TX TCN           3
  Num RX BPDU       614                Num RX TCN           2
  Num Transition FWD 1                 Num Transition BLK   1
  Rcvd BPDU         no                 Rcvd STP             no

ethernet3
  ------------------------------------------------
  CIST
  ----------------
```

```
  Enabled            yes                      role
Designated
  Num TX BPDU        615                      Num TX TCN          0
  Num RX BPDU        2                        Num RX TCN          0
  Num Transition FWD 1                        Num Transition BLK  1
  Rcvd BPDU          no                       Rcvd STP            no

ethernet4
  -------------------------------------------------
  CIST
  ---------------
  Enabled            yes                      role
Designated
  Num TX BPDU        615                      Num TX TCN          0
  Num RX BPDU        0                        Num RX TCN          0
  Num Transition FWD 1                        Num Transition BLK  1
  Rcvd BPDU          no                       Rcvd STP            no

ethernet5
  -------------------------------------------------
  CIST
  ---------------
  Enabled            yes                      role                Backup
  Num TX BPDU        2                        Num TX TCN          0
  Num RX BPDU        615                      Num RX TCN          0
  Num Transition FWD 0                        Num Transition BLK  1
  Rcvd BPDU          no                       Rcvd STP            no
```

The following table describes the fields displayed under the `show spanning-tree interfaces` command:

Field	Description
enabled	Interface state. If yes (port is up) or if no (port is down).
role	Role of the interface. For example, Designated and so on.
Num TX BPDU	Number of BPDUs transmitted.
Num TX TCN	Number of Topology Change Notification (TCN)

Field	Description
	transmitted.
Num RX BPDU	Number of BPDUs received.
Num RX TCN	Number of TCNs received.
Num Transition FWD	Number of ports transitioned in forward state.
Num Transition BLK	Number of ports transitioned in blocked state.
Rcvd BPDU	Information about the BPDUs received. The output can be yes or no.
Rcvd STP	Information about the STP received. The output can be yes or no.

The following example shows the sample output for `show spanning-tree interfaces ethernet 3`:

```
ethernet3
  ------------------------------------------------
  CIST
  ---------------
  Enabled             yes                         role
Designated
  Num TX BPDU         716                         Num TX TCN          0
  Num RX BPDU         2                           Num RX TCN          0
  Num Transition FWD 1                            Num Transition BLK  1
  Rcvd BPDU           no                          Rcvd STP            no
```

The show command can be used to view the spanning-tree interfaces detail:

```
ACOS# show spanning-tree interfaces detail
show spanning-tree interfaces ethernet <> / trunk <> detail:
```

The following example shows the sample output for `show spanning-tree interfaces ethernet 3 detail`:

```
ethernet3
  ------------------------------------------------
  CIST
  ---------------
```

```
  Enabled             yes                       role
Designated
  port id             8.008                     state
forwarding
  external port cost 20000                      admin external cost  0
  internal port cost 20000                      admin internal cost  0
  designated root     8.000.00:1F:A0:02:3D:99  dsgn external cost   20000
  priority 32768
  designated bridge   8.000.00:1F:A0:05:A2:09  dsgn internal cost   8.008
  priority 32768
  admin edge port     no                        auto edge port       yes
  oper edge port      yes                       topology change ack  no
  point-to-point      yes                       admin point-to-point auto
  restricted role     no                        restricted TCN       no
  port hello time     2                         disputed             no
  bpdu guard port     no                        bpdu guard error     no
  network port        no                        BA inconsistent      no
  Num TX BPDU         1062                      Num TX TCN           0
  Num RX BPDU         2                         Num RX TCN           0
  Num Transition FWD 1                          Num Transition BLK   1
  Rcvd BPDU           no                        Rcvd STP             no
  Rcvd RSTP           no                        Send RSTP            yes
  Rcvd TC Ack         no                        Rcvd TCN             no
```

The following table describes the fields displayed under the `show spanning-tree interfaces ethernet 3 detail` command:

Field	Description
Enabled	Interface state. If yes (port is up) or if no (port is down).
role	Role of the interface. For example, Designated and so on.
port id	Port ID of the interface.
state	State of the interface (forwarding or blocking).
external port cost	External port cost according to the port's link speed.
admin external cost	Administrator configured external cost.
internal port cost	Internal port cost according to the port's link speed. This is used in MSTP and not with STP/RSTP.

Field	Description
admin internal cost	Administrator configured internal cost. This is used in MSTP and not with STP/RSTP.
designated root	Designated root ID.
dsgn external cost	Designated root port external path cost.
priority	Priority of the path cost.
dsgn regional root	Designated regional root.
dsgn internal cost	Designated internal cost.
priority	Priority of the path cost.
designated bridge	Designated bridge ID.
dsgn internal cost	Designated root port internal path cost used for MSTP.
priority	Designated bridge ID priority.
admin edge port	Information about the administrator configured edge port. The output can be yes or no.
auto edge port	Information about the auto edge port configured. The output can be yes or no.
oper edge port	Information about the operational status of port. The output can be yes or no.
topology change ack	Information about the topology change acknowledgment sent. The output can be yes or no.
point-to-point	Information about the port being point-to-point. The output can be yes or no.
admin point-to-point	Information about the administrator configured point-to-point link. The output can be yes or no.
restricted role	To restrict the port configuration as Root Port. Currently, this configuration is not supported.
restricted TCN	To restrict the port from propagating topology change. Currently, this configuration is not supported.
port hello time	Time interval between BPDUs in port.
disputed	Dispute occurred on the port.
bpdu guard port	Information about the BPDU guard configuration. The

Field	Description
	output can be yes or no.
bpdu guard error	Information about receiving the BPDU on the BPDU guard configured port. The output can be yes or no.
network port	Information about configuring the port as the network port. The output can be yes or no.
BA inconsistent	Information about BPDUs received on the port. The output can be yes or no.
Num TX BPDU	Number of BPDUs transmitted.
Num TX TCN	Number of TCNs transmitted.
Num RX BPDU	Number of BPDUs received.
Num RX TCN	Number of TCNs received.
Num Transition FWD	Number of ports transitioned in forward state.
Num Transition BLK	Number of ports transitioned in blocked state.
Rcvd BPDU	Information about the BPDUs received. The output can be yes or no.
Rcvd STP	Information about the STP received. The output can be yes or no.
Rcvd RSTP	Information about the RSTP received. The output can be yes or no.
Send RSTP	Information about the RSTP sent. The output can be yes or no.
Rcvd TC Ack	Information about the topology change acknowledgment received. The output can be yes or no.
Rcvd TCN	Information about the topology change notification received. The output can be yes or no.

MSTP Show Commands

The show command can be used to view the MSTP spanning-tree configuration:

```
ACOS# show spanning-tree
Spanning-tree Mode : MSTP
-------------
MST Instance 0
FWD ports : 2 3 4
BLK ports : 5
VLANs : 1-4094
-------------
```

The following table describes the fields displayed under the `show spanning-tree` command:

Field	Description
Spanning-tree Mode	Configured mode on the system (MSTP).
Instance	MSTP instance.
FWD ports	Ports in forward state.
BLK ports	Ports in blocked state.
VLAN	VLAN of the MSTP.

The show command can be used to view the spanning-tree instance:

```
ACOS# show spanning-tree instance 0
Spanning-tree Mode : MSTP
-------------
MST Instance 0
FWD ports : 2 3 4
BLK ports : 5
VLANs : 1-4094
-------------
```

The following example shows the sample output for `show spanning-tree detail`:

```
ACOS# show spanning-tree detail
Spanning-tree Mode        : MSTP
-------------
MST Instance 0
FWD ports                 : 1 3 4
BLK ports                 : 5
VLANs                     : 1-9,11-4094
```

```
Bridge ID                 : 8.000.00:1F:A0:05:A2:08 (priority 32768)
Regional root             : 8.000.00:1F:A0:05:A2:08 (priority 32768)
Root_port                 : none
Time since topology change : 31889
Topology change count     : 1
Topology change port      :
Last topology change port : ethernet3
-------------
MST Instance 1
FWD ports                 : 1 3 4
BLK ports                 : 5
VLANs                     : 10
Bridge ID                 : 8.000.00:1F:A0:05:A2:08 priority 32769
(priority 32768
sys-id-ext 1)
Regional root             : 8.000.00:1F:A0:05:A2:08 priority 32769
(priority 32768
sys-id-ext 1)
Root_port                 : none
Time since topology change : 31889
Topology change count     : 1
Topology change port      :
Last topology change port : ethernet3
-------------
```

The following table describes the fields displayed under the `show spanning-tree detail` command:

Field	Description
Spanning-tree Mode	Configured mode on the system (MSTP).
MST Instance	Instance of the MSTP.
FWD ports	Ports in forward state.
BLK ports	Ports in blocked state.
VLANs	VLAN of the MSTP.
Bridge ID	MAC address and priority of the switch.
Regional root	MAC address and priority of the regional port.

Field	Description
Root port	Port with the lowest path cost to reach the root bridge.
Time since topology change	Time duration when the topology was changed.
Topology change count	Count of the topology changes.
Topology change port	Port on which the topology is changed.
Last topology change port	Port on which the topology was changed last.

The show command can be used to view the `spanning-tree instance detail`:

```
ACOS# show spanning-tree instance 1 detail
Spanning-tree Mode                            : MSTP
------------------
MST Instance 1
FWD ports                 : 1 3 4
BLK ports                 : 5
VLANs                     : 10
Bridge ID                 : 8.001.00:1F:A0:05:A2:08 priority 32769
(priority 32768 sys-id-ext 1)
Regional root             : 8.001.00:1F:A0:05:A2:08 priority 32769
(priority 32768 sys-id-ext 1)
Root_port                 : none
Time since topology change : 32194
Topology change count     : 1
Topology change port      :
Last topology change port  : ethernet3
-------------------
```

The show command can be used to view the spanning-tree interfaces:

```
ACOS# show spanning-tree interfaces
detail: Detail
ethernet: Ethernet interface
instance: MSTI configured on the system
trunk: Trunk interface
```

```
|       : Output modifiers
```

The following table describes the fields displayed under the `show spanning-tree interfaces` command:

Field	Description
detail	Details about spanning-tree interfaces.
ethernet	Ethernet interface running the spanning-tree protocol.
instance	Interface instance running the spanning-tree protocol.
trunk	Trunk interface running the spanning-tree protocol.

The following example shows the sample output for `show spanning-tree interfaces` :

```
ACOS#show spanning-tree interfaces
ethernet1
  ------------------------------------------------
  CIST
  ----------------
  Enabled           yes                        role
Designated
  Num TX BPDU        14077                      Num TX TCN          0
  Num RX BPDU        0                          Num RX TCN          0
  Num Transition FWD 4                          Num Transition BLK  4
  Rcvd BPDU          no                         Rcvd STP            no
  ------------------------------------------------
  MSTI Instance: 0
  ----------------
  role              Designated                 port id            8.006
  state             forwarding                 disputed           no
  internal port cost 20000                     admin internal cost 0
  dsgn regional root 8.000.00:1F:A0:05:A2:08  dsgn internal cost  0
  priority 32768
  designated bridge  8.000.00:1F:A0:05:A2:08  dsgn internal cost  8.006
  priority 32768
  ------------------------------------------------
  MSTI Instance: 1
  ----------------
```

```
  role              Designated              port id          8.006
  state             forwarding              disputed         no
  internal port cost 20000                  admin internal cost 0
  dsgn regional root 8.001.00:1F:A0:05:A2:08 dsgn internal cost 0
  priority 32769 (priority 32768, sys-id-ext 1)
  designated bridge 8.001.00:1F:A0:05:A2:08 dsgn internal cost 8.006
  priority 32769 (priority 32768, sys-id-ext 1)
  --------------------------------------------------

ethernet3
  --------------------------------------------------
  CIST
  ----------------
  Enabled           yes                     role
Designated
  Num TX BPDU       14078                   Num TX TCN       2
  Num RX BPDU       3                       Num RX TCN       0
  Num Transition FWD 4                      Num Transition BLK 4
  Rcvd BPDU         no                      Rcvd STP         no
  --------------------------------------------------
  MSTI Instance: 0
  ----------------
  role              Designated              port id          8.008
  state             forwarding              disputed         no
  internal port cost 20000                  admin internal cost 0
  dsgn regional root 8.000.00:1F:A0:05:A2:08 dsgn internal cost 0
  priority 32768
  designated bridge 8.000.00:1F:A0:05:A2:08 dsgn internal cost 8.008
  priority 32768
  --------------------------------------------------
  MSTI Instance: 1
  ----------------
  role              Designated              port id          8.008
  state             forwarding              disputed         no
  internal port cost 20000                  admin internal cost 0
  dsgn regional root 8.001.00:1F:A0:05:A2:08 dsgn internal cost 0
  priority 32769 (priority 32768, sys-id-ext 1)
  designated bridge 8.001.00:1F:A0:05:A2:08 dsgn internal cost 8.008
  priority 32769 (priority 32768, sys-id-ext 1)
  --------------------------------------------------
```

```
ethernet4
  --------------------------------------------------
  CIST
  ----------------
  Enabled          yes                      role
Designated
  Num TX BPDU      14077                    Num TX TCN          0
  Num RX BPDU      0                        Num RX TCN          0
  Num Transition FWD 4                      Num Transition BLK  4
  Rcvd BPDU        no                       Rcvd STP            no
  --------------------------------------------------
  MSTI Instance: 0
  ----------------
  role             Designated               port id             8.009
  state            forwarding               disputed            no
  internal port cost 20000                  admin internal cost 0
  dsgn regional root 8.000.00:1F:A0:05:A2:08  dsgn internal cost  0
  priority 32768
  designated bridge  8.000.00:1F:A0:05:A2:08  dsgn internal cost  8.009
  priority 32768
  --------------------------------------------------
  MSTI Instance: 1
  ----------------
  role             Designated               port id             8.009
  state            forwarding               disputed            no
  internal port cost 20000                  admin internal cost 0
  dsgn regional root 8.001.00:1F:A0:05:A2:08  dsgn internal cost  0
  priority 32769 (priority 32768, sys-id-ext 1)
  designated bridge  8.001.00:1F:A0:05:A2:08  dsgn internal cost  8.009
  priority 32769 (priority 32768, sys-id-ext 1)
  --------------------------------------------------

ethernet5
  --------------------------------------------------
  CIST
  ----------------
  Enabled          yes                      role                Backup
  Num TX BPDU      3                        Num TX TCN          0
  Num RX BPDU      14078                    Num RX TCN          2
```

```
Num Transition FWD 0                              Num Transition BLK   2
Rcvd BPDU           no                            Rcvd STP            no
--------------------------------------------------
MSTI Instance: 0
---------------
role               Backup                         port id             8.00A
state              discarding                     disputed            no
internal port cost 20000                          admin internal cost 0
dsgn regional root 8.000.00:1F:A0:05:A2:08  dsgn internal cost   0
priority 32768
designated bridge  8.000.00:1F:A0:05:A2:08  dsgn internal cost   8.008
priority 32768
-------------------------------------------------
MSTI Instance: 1
---------------
role               Backup                         port id             8.00A
state              discarding                     disputed            no
internal port cost 20000                          admin internal cost 0
dsgn regional root 8.001.00:1F:A0:05:A2:08  dsgn internal cost   0
priority 32769 (priority 32768, sys-id-ext 1)
designated bridge  8.001.00:1F:A0:05:A2:08  dsgn internal cost   8.008
priority 32769 (priority 32768, sys-id-ext 1)
```

The following table describes the fields displayed under the `show spanning-tree interfaces` command:

Field	Description
Instance	Instance of the interface.
role	Role of the interface. For example, Designated and so on.
port id	Port ID of the interface.
state	State of the interface (forwarding or discarding).
disputed	Dispute occurred on the port.
internal port cost	Internal port cost according to the port's link speed.
admin internal cost	Administrator configured internal cost.
dsgn regional root	Designated regional root.
dsgn internal cost	Designated internal cost.

Field	Description
priority	Priority of the path cost.
designated bridge	Designated bridge ID.
dsgn internal cost	Designated root port internal path cost used for MSTP.
priority	Designated bridge ID priority.

The show command can be used to view the spanning-tree interfaces ethernet 3 instance 0:

```
ethernet3
  -------------------------------------------------
  MSTI Instance: 0
  ----------------
  role              Designated                port id              8.008
  state             forwarding                disputed             no
  internal port cost 20000                    admin internal cost  0
  dsgn regional root 8.000.00:1F:A0:05:A2:09  dsgn internal cost   0
  priority 32768
  designated bridge  8.000.00:1F:A0:05:A2:09  dsgn internal cost   8.008
  priority 32768
  -------------------------------------------------
```

The show command can be used to view the spanning-tree interfaces ethernet 3 detail:

```
ethernet3
  -------------------------------------------------
  CIST
  ----------------
  Enabled           yes                       role
Designated
  port id           8.008                     state
forwarding
  external port cost 20000                    admin external cost  0
  internal port cost 20000                    admin internal cost  0
  designated root    8.000.00:1F:A0:05:A2:08  dsgn external cost   0
  priority 32768
  designated bridge  8.000.00:1F:A0:05:A2:08  dsgn internal cost   8.008
  priority 32768
  admin edge port    no                       auto edge port       yes
```

```
oper edge port     yes                        topology change ack  no
point-to-point     yes                        admin point-to-point auto
restricted role    no                         restricted TCN       no
port hello time    2                          disputed             no
bpdu guard port    no                         bpdu guard error     no
network port       no                         BA inconsistent      no
Num TX BPDU        15310                      Num TX TCN           2
Num RX BPDU        3                          Num RX TCN           0
Num Transition FWD 4                          Num Transition BLK   4
Rcvd BPDU          no                         Rcvd STP             no
Rcvd RSTP          no                         Send RSTP            yes
Rcvd TC Ack        no                         Rcvd TCN             no
-------------------------------------------------
MSTI Instance: 0
----------------
role               Designated                 port id              8.008
state              forwarding                 disputed             no
internal port cost 20000                      admin internal cost  0
dsgn regional root 8.000.00:1F:A0:05:A2:08  dsgn internal cost   0
priority 32768
designated bridge  8.000.00:1F:A0:05:A2:08  dsgn internal cost   8.008
priority 32768
-------------------------------------------------
MSTI Instance: 1
----------------
role               Designated                 port id              8.008
state              forwarding                 disputed             no
internal port cost 20000                      admin internal cost  0
dsgn regional root 8.001.00:1F:A0:05:A2:08  dsgn internal cost   0
priority 32769 (priority 32768, sys-id-ext 1)
designated bridge  8.001.00:1F:A0:05:A2:08  dsgn internal cost   8.008
priority 32769 (priority 32768, sys-id-ext 1)
-------------------------------------------------
```

Limitation

Spanning-tree (MSTP/STP and RSTP) protocols are not supported on the Multi-PU platform.

Virtual LAN Support

This chapter describes support for VLAN and for VLAN-to-VLAN bridging.

The following topics are covered:

VLAN Overview

A VLAN is a Layer 2 broadcast domain. MAC-layer broadcast traffic can be flooded within the VLAN but does not cross to other VLANs. For traffic to go from one VLAN to another, it must be routed.

You can segment the ACOS device into multiple VLANs. Each Ethernet data port can be a member of one or more VLANs, depending on whether the port is tagged or untagged:

- Tagged: Tagged ports can be members of multiple VLANs. The port can recognize the VLAN to which a packet belongs based on the VLAN tag included in the packet.

- Untagged: Untagged ports can belong to only a single VLAN. By default, all Ethernet data ports are untagged members of VLAN 1.

NOTE: A tagged port is a physical port to which a tagged VLAN is bound, while an untagged port is a physical port to which an untagged VLAN is bound. See the Example of Tagged and Untagged Ports section for how these ports are configured.

Default VLAN (VLAN 1)

By default, all the ACOS device's Ethernet data ports are members of a single virtual LAN (VLAN), VLAN 1.

On a new or non-configured ACOS device, as soon as you configure an IP address on any individual Ethernet data port or trunk interface, Layer 2 forwarding on VLAN 1 is disabled.

When Layer 2 forwarding on VLAN 1 is disabled, broadcast, multicast, and unknown unicast packets are dropped instead of being forwarded. Learning is also disabled on the VLAN. However, packets for the ACOS device itself (for example, LACP or OSPF) are not dropped.

NOTE:	When an IP address is not configured on an Ethernet data port or trunk interface, which is belonging to the default VLAN (VLAN 1), broadcast, multicast, and unknown unicast packets received by these interfaces, then it is flooded to the data ports or the trunk interface, which belongs to the default VLAN.

To re-enable Layer 2 forwarding on VLAN 1, use the following command at the global configuration level of the CLI:

```
ACOS(config)# vlan-global enable-def-vlan-l2-forwarding
```

NOTE:	Configuring an IP address on an individual Ethernet interface indicates you are deploying in routed mode (also called "gateway mode"). If you deploy in transparent mode instead, in which the ACOS device has a single IP address for all data interfaces, Layer 2 forwarding is left enabled by default on VLAN 1.

Virtual Ethernet Interfaces

On ACOS devices deployed in routed mode (Layer 3 mode), you can configure IP addresses on VLANs. To configure an IP address on a VLAN, add a Virtual Ethernet (VE) interface to the VLAN, then assign the IP address to the VE.

Each VLAN can have one VE. The VE ID must be the same as the VLAN ID. For example, VLAN 2 can have VE 2, VLAN 3 can have VE 3, and so on.

Maximum Number of Supported Virtual Ethernet Interfaces

The number of VE interfaces supported on a single port varies depending on the specific platform.

Example of Tagged and Untagged Ports

In the following example, two physical Ethernet ports are enabled. The first Ethernet port (`interface ethernet 1`) will be configured as a tagged port with two network

interfaces, while the second Ethernet port (`interface ethernet 7`) will be configured as an untagged port with one network interface.

1. Enable the physical Ethernet ports:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# exit
```

```
ACOS(config)# interface ethernet 7
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# exit
```

2. Configure VLAN 10. Bind Ethernet port 1 to a tagged VLAN 10. The 802.1Q tag is 10. Bind a network interface to the tagged port:

```
ACOS(config) #vlan 10
ACOS(config-vlan:10)# tagged ethernet 1
ACOS(config-vlan:10)# router-interface ve 10
ACOS(config-vlan:10)# exit
```

3. Configure VLAN 11. Bind Ethernet port 1 to a tagged VLAN 11. The 802.1Q tag is 11. Bind a network interface to the tagged port:

```
ACOS(config)# vlan 11
ACOS(config-vlan:11)# tagged ethernet 1
ACOS(config-vlan:11)# router-interface ve 11
ACOS(config-vlan:11)# exit
```

4. Configure VLAN 5. Bind Ethernet port 7 to an untagged VLAN 5. Bind a network interface to the untagged port:

```
ACOS(config)# vlan 5
ACOS(config-vlan:5)# untagged ethernet 7
ACOS(config-vlan:5)# router-interface ve 5
ACOS(config-vlan:5)# exit
```

5. Show the VLAN configuration:

```
ACOS# show config vlan
...
vlan 5
  untagged ethernet 7
```

```
   router-interface ve 5
!
vlan 10
  tagged ethernet 1
  router-interface ve 10
!
vlan 11
  tagged ethernet 1
  router-interface ve 11
!
```

6. Show the VLANs:

```
ACOS# show vlans
Total VLANs: 4
VLAN 1, Name [DEFAULT VLAN]:
Untagged Ethernet Ports:    2   3   4   5   6   8
  Tagged Ethernet Ports:   None
 Untagged Logical Ports:   None
   Tagged Logical Ports:   None


VLAN 5, Name [None]:
Untagged Ethernet Ports:    7
  Tagged Ethernet Ports:   None
 Untagged Logical Ports:   None
   Tagged Logical Ports:   None


       Router Interface:   ve 5


VLAN 10, Name [none]:
Untagged Ethernet Ports:   None
  Tagged Ethernet Ports:    1
 Untagged Logical Ports:   None
   Tagged Logical Ports:   None


       Router Interface:   ve 10


VLAN 11, Name [none]:
Untagged Ethernet Ports:   None
  Tagged Ethernet Ports:    1
```

Feedback

```
   Untagged Logical Ports:    None
     Tagged Logical Ports:    None


        Router Interface:    ve 11
VLAN 1, Name [DEFAULT VLAN]:
Untagged Ethernet Ports:     2   3   4   5   6   8
  Tagged Ethernet Ports:    None
 Untagged Logical Ports:    None
   Tagged Logical Ports:    None


VLAN 5, Name [None]:
Untagged Ethernet Ports:     7
  Tagged Ethernet Ports:    None
 Untagged Logical Ports:    None
   Tagged Logical Ports:    None


        Router Interface:    ve 5


VLAN 10, Name [none]:
Untagged Ethernet Ports:    None
  Tagged Ethernet Ports:     1
 Untagged Logical Ports:    None
   Tagged Logical Ports:    None


        Router Interface:    ve 10


VLAN 11, Name [none]:
Untagged Ethernet Ports:    None
  Tagged Ethernet Ports:     1
 Untagged Logical Ports:    None
   Tagged Logical Ports:    None


        Router Interface:    ve 11
```

VLAN-to-VLAN Bridging

The following topics are covered:

VLAN-to-VLAN Bridging Overview

VLAN-to-VLAN bridging allows an ACOS device to selectively bridge traffic among multiple VLANs. The ACOS device selectively forwards packets from one VLAN to another based on the VLAN-to-VLAN bridging configuration on the ACOS device. This feature allows the traffic flow between VLANs to be tightly controlled through the ACOS device without the need to reconfigure the hosts in the separate VLANs.

VLAN-to-VLAN bridging is useful in cases where reconfiguring the hosts on the network either into the same VLAN, or into different IP subnets, is not desired or is impractical.

You can configure a bridge VLAN group to forward one of the following types of traffic:

- IP traffic only (the default) – This option includes typical traffic between end hosts, such as ARP requests and responses.

  This option does not forward multicast packets.

- All traffic – This option forwards all types of traffic.

Figure 1 shows an example topology of VLAN-to-VLAN bridging:

Figure 1 : VLAN-to-VLAN Bridging (with VRRP-A)
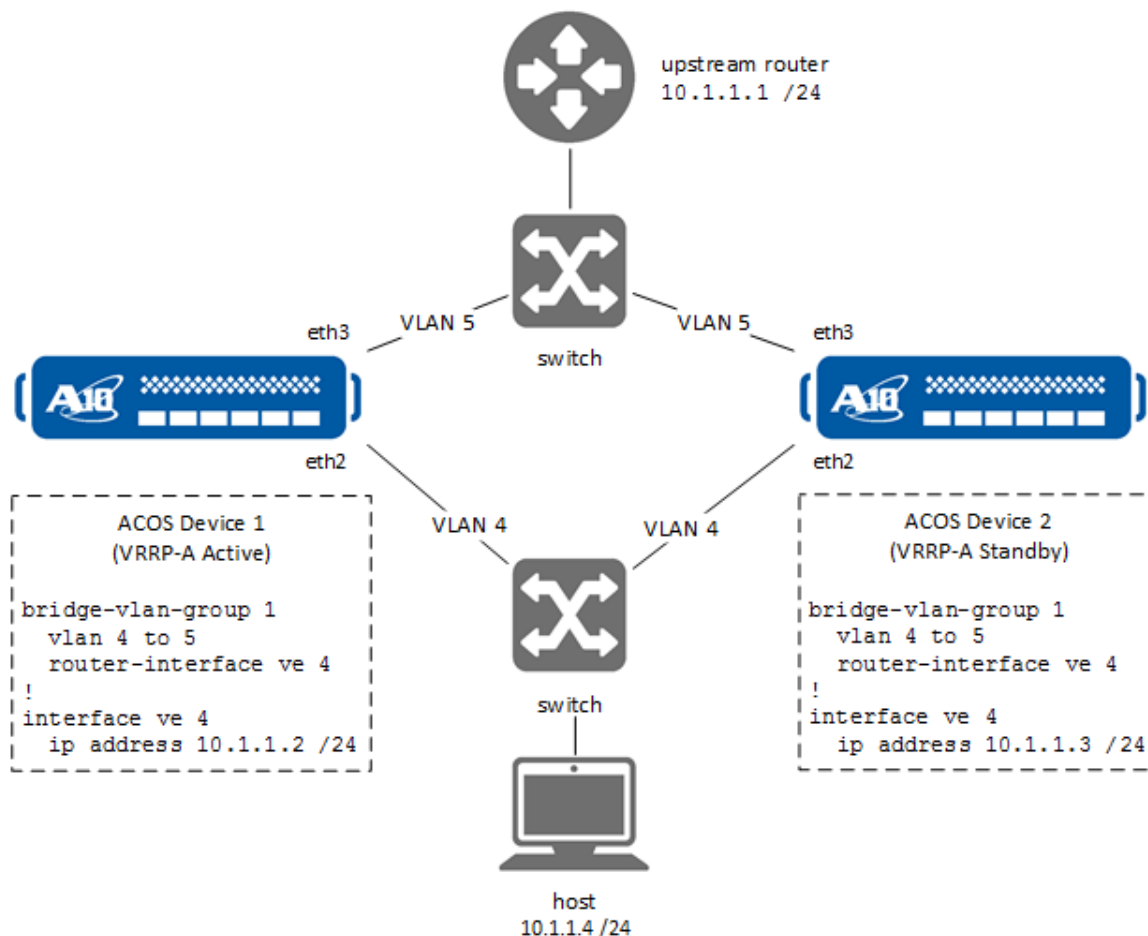
In this example, the ACOS devices are bridging traffic between VLAN 4 and VLAN 5.

VLAN-to-VLAN Bridging Configuration Notes

VLAN-to-VLAN bridging is supported on ACOS devices deployed in transparent mode (Layer 2) or in gateway mode (Layer 3).

Each VLAN to be bridged must be configured on the ACOS device. The normal rules for tagging apply:

- If an interface belongs to only one VLAN, the interface can be untagged.

- If the interface belongs to more than one VLAN, the interface must be tagged.

Each VLAN can belong to only a single bridge VLAN group.

Each bridge VLAN group can have a maximum of 8 member VLANs. Traffic from any VLAN in the group is bridged to all other VLANs in the group. The total number of bridge VLAN groups on the system (including those in L3V partitions) cannot exceed 255.

If the ACOS device is deployed in gateway mode, a Virtual Ethernet (VE) interface is required in the bridge VLAN group.

VLAN-to-VLAN Bridging Configuration Examples

To configure VLAN-to-VLAN bridging:

1. Configure each of the VLANs to be bridged. In each VLAN, add the ACOS device's interfaces to the VLAN.

2. Configure a bridge VLAN group. Add the VLANs to the group.

   If the ACOS device is deployed in routed mode, add a Virtual Ethernet (VE) interface to the group.

   Optionally, you can assign a name to the group. You also can change the types of traffic to be bridged between VLANs in the group.

3. If the ACOS device is deployed in routed mode, configure an IP address on the VE to place the ACOS device in the same subnet as the bridged VLANs.

CLI Example – Transparent Mode

The commands in this section configure an ACOS device deployed in transparent mode to forward IP traffic between VLANs 2 and 3.

The following commands configure the VLANs:

```
ACOS(config)# vlan 2
ACOS(config-vlan:2)# tagged ethernet 2
ACOS(config-vlan:2)# exit
ACOS(config)# vlan 3
ACOS(config-vlan:3)# tagged ethernet 3
ACOS(config-vlan:3)# exit
```

The following commands configure the bridge VLAN group:

```
ACOS(config)# bridge-vlan-group 1
```

```
ACOS(config-bridge-vlan-group:1)# vlan 2 to 3
ACOS(config-bridge-vlan-group:1)# exit
```

CLI Example – Routed Mode with VRRP-A

VLAN-to-VLAN bridging can also be configured with VRRP-A by specifying a VRID under the bridge VLAN configuration. Using the topology defined in VLAN-to-VLAN Bridging (with VRRP-A):

- Only the active device in the VRID will respond to ARP requests from devices in the bridged VLAN.

- The active VRRP-A device forwards any traffic passing through the bridge VLAN (destined for 10.1.1.1), and processes any traffic destined for the bridge VLAN VE IP address (10.1.1.2).

- The standby VRRP-A device drops any traffic passing through the bridge VLAN (destined for 10.1.1.1), but will processes any traffic destined for the bridge VLAN VE IP address (10.1.1.2).

- On a failover, the new active device will forward any traffic passing through the bridge VLAN (destined for 10.1.1.3).

The commands in this section configure the topology shown in VLAN-to-VLAN Bridging (with VRRP-A); two ACOS devices deployed in routed mode to forward IP traffic between VLANs 4 and 5 on IP subnet 10.10.1.x.

Configure VRRP-A, for Device 1:

```
ACOS1(config)# vrrp-a common
ACOS1(config-common)# device-id 1
ACOS1(config-common)# set-id 1
ACOS1(config-common)# enable
ACOS1(config-common)# exit
ACOS1(config)# vrrp-a l3-inline-mode
ACOS1(config)# vrrp-a restart-port-list
ACOS1(config-restart-port-list)# ethernet 7 to 8
ACOS1(config-restart-port-list)# exit
ACOS1(config)# vrrp-a vrid-lead lead
ACOS1(config-vrid-lead:lead)# partition shared vrid 0
ACOS1(config-vrid-lead:lead)# exit
ACOS1(config)#
```

Enabling `l3-inline-mode` and `restart-port-list` in the configuration are mandatory for VLAN-to-VLAN bridging with VRRP-A. All interfaces which are part of the bridge VLAN group must be included in the `restart-port-list`.

NOTE:	You must omit at least one port connecting the ACOS Devices from the restart port-list, and heartbeat messages must be enabled on the port. This is to ensure that heartbeat messages between the ACOS Devices are maintained; otherwise, flapping might occur.

The `vrid-lead` configuration is used for L3V partitions to follow the vrid-lead of the shared partition. Since only one VRID can be configured in a given partition when `l3-inline-mode` is enabled, all L3V partitions will end up following same VRID of the shared partition.

To configure the vrid-lead in an L3V partition (for example, partition p1):

```
ACOS[p1](config-vrid:0)# vrrp-a vrid 0
ACOS[p1](config-vrid:0)# follow vrid-lead lead
ACOS[p1](config-vrid:0)#
```

Configure VRRP-A for Device 2:

```
ACOS2(config)# vrrp-a common
ACOS2(config-common)# device-id 2
ACOS2(config-common)# set-id 1
ACOS2(config-common)# enable
ACOS2(config-common)# exit
ACOS2(config)# vrrp-a l3-inline-mode
ACOS2(config)# vrrp-a restart-port-list
ACOS2(config-restart-port-list)# ethernet 2 to 3
ACOS2(config-restart-port-list)# exit
ACOS2(config)# vrrp-a vrid-lead lead
ACOS2(config-vrid-lead:lead)# partition shared vrid 0
ACOS2(config-vrid-lead:lead)# exit
ACOS2(config)#
```

On each ACOS device, the following commands configure the VLANs (example shown for Device 1):

```
ACOS1(config)# vlan 4
ACOS1(config-vlan:4)# tagged ethernet 2
ACOS1(config-vlan:4)# exit
```

```
ACOS1(config)# vlan 5
ACOS1(config-vlan:5)# tagged ethernet 3
ACOS1(config-vlan:5)# exit
```

On each ACOS device, the following commands configure the bridge VLAN group, which includes a VE (example shown for Device 1):

```
ACOS1(config)# bridge-vlan-group 1
ACOS1(config-bridge-vlan-group:1)# vlan 4 to 5
ACOS1(config-bridge-vlan-group:1)# router-interface ve 4
ACOS1(config-bridge-vlan-group:1)# exit
```

On ACOS device 1, The following commands assign an IP address to the VE:

```
ACOS1(config)# interface ve 4
ACOS1(config-if:ve:4)# ip address 10.1.1.2 /24
ACOS1(config-if:ve:4)# exit
```

On ACOS device 2, The following commands assign an IP address to the VE:

```
ACOS2(config)# interface ve 4
ACOS2(config-if:ve:4)# ip address 10.1.1.3 /24
ACOS2(config-if:ve:4)# exit
```

CLI Example – Preventing Loops in Networks

Packet brokers (active and passive paths) are used in some enterprise network topologies/environments to route traffic to security devices as service chains. A network loop upstream may occur when the firewall performs a fail-over to the passive paths. Since ACOS retains VLAN layer 2 information, it cannot be updated after a failover. Hence, it sends the old VLAN information to the sessions and sends packets over the passive route. This results in an indefinite traffic loop and the network never recovers by itself.

To prevent such loops, use the option update-l2-info, that detects the changes in the session's L2 information and updates the cache accordingly. L2 information includes the VLAN tag and MAC address.

The topology in the following figure demonstrates this scenario. It is a modified form of SSLi IP-less deployment (refer to the *SSLi Configuration Guide* for more details).
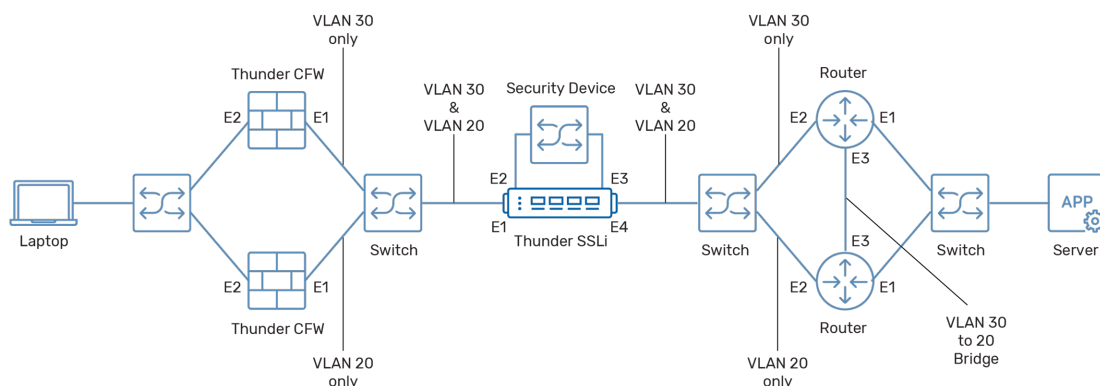
The following commands configure the above shown topology:

1. Configure the active and passive VLANs (20 and 30) in a **bridge-vlan-group.**

```
vlan 20
!
vlan 30
!
bridge-vlan-group 1
vlan 20
vlan 30
!
```

2. Configure the ethernet interface as a virtual wire endpoint and configure the
    update-l2-info option.

    In case of ethernet 2 and 3, **vlan-learning** option is disabled because they form a
    loop inside the Thunder itself, unlike ethernet e1 and e4 that face the client and
    the server.

```
interface ethernet 1
enable
virtual-wire update-l2-info vlan-learning enable mac-learning enable
ip allow-promiscuous-vip
!
interface ethernet 2
enable
virtual-wire update-l2-info vlan-learning disable mac-learning enable
!
interface ethernet 3
```

```
enable
virtual-wire update-l2-info vlan-learning disable mac-learning enable
ip allow-promiscuous-vip
!
interface ethernet 4
enable
virtual-wire update-l2-info vlan-learning enable mac-learning enable
!
```

3. Configure the virtual wire.

```
virtual-wire 1
ethernet 1 ethernet 2
!
virtual-wire 2
ethernet 3 ethernet 4
!
```

4. Configure the SLB server and the service group components.

```
slb template server-ssl sssl
forward-proxy-enable
!
slb server e2 ethernet 2
port 0 tcp
port 0 udp
port 8080 tcp
!
slb server e4 ethernet 4
port 0 tcp
port 0 udp
port 443 tcp
port 8080 tcp
!
slb service-group sg_e2_0_tcp tcp
member e2 0
!
slb service-group sg_e2_0_udp udp
member e2 0
!
slb service-group sg_e2_8080_tcp tcp
```

```
member e2 8080
!
slb service-group sg_e4_0_tcp tcp
member e4 0
!
slb service-group sg_e4_0_udp udp
member e4 0
!
slb service-group sg_e4_443_tcp tcp
member e4 443
!
```

5. Configure the SLB Virtual Server and templates. ACLs are added for permitting traffic to the VIPs from client side and server side.

```
access-list 198 permit ip any any ethernet 1
!
access-list 199 permit ip any any ethernet 3
!
slb template server-ssl sssl
forward-proxy-enable
!
slb template client-ssl cssl
forward-proxy-ca-cert ax.crt
forward-proxy-ca-key ax.crt
forward-proxy-enable
!
slb virtual-server ssli_ig 0.0.0.0 acl 198
port 0 tcp
service-group sg_e2_0_tcp
use-rcv-hop-for-resp
no-dest-nat
port 0 udp
service-group sg_e2_0_udp
use-rcv-hop-for-resp
no-dest-nat
port 443 https
service-group sg_e2_8080_tcp
use-rcv-hop-for-resp
template client-ssl cssl
```

```
no-dest-nat port-translation
!
slb virtual-server ssli_o 0.0.0.0 acl 199
port 0 tcp
service-group sg_e4_0_tcp
use-rcv-hop-for-resp
no-dest-nat
port 0 udp
service-group sg_e4_0_udp
use-rcv-hop-for-resp
no-dest-nat
port 443 tcp
service-group sg_e4_443_tcp
use-rcv-hop-for-resp
no-dest-nat
port 8080 http
service-group sg_e4_443_tcp
use-rcv-hop-for-resp
template server-ssl sssl
no-dest-nat port-translation
!
```

802.1Q-in-Q VLAN Tagging

802.1Q-in-Q (Q-in-Q) is an Ethernet networking standard. This capability expands the VLAN space by tagging the tagged packets, thus creating a double-tagged frame.

It allows the service provider to separate the traffic from different customers and transparently transfer it throughout the network. It also provides certain services on specific VLANs for specific customers. It is supported on all FPGA platforms and only on certain non-FPGA platforms.

The double tagging enables the device to process the following two Q-in-Q VLAN tags:

- C-TAG- It is a Customer Tag to identify & isolate customer traffic in a multi-customer environment.

● S-TAG- It is a Service Tag to identify and isolate the traffic in a multi-service environment. For example, Firewall, SSLi (encrypt/decrypt).

Supported Features

● Supports virtual wire and L2-IPless topologies only.

● Supports shared partition only. An L3V partition can support Q-in-Q only if the configuration is set on a shared partition.

● Supports all FPGA platforms. For non-FPGA, Q-in-Q is supported only on Thunder 3350 series platforms. Only X710 ports (ports 9 to 20) on the Thunder 3350 platform support Q-in-Q.

Limitations

● Q-in-Q support is handled by NIC cards, it drop the packets if there is TPID mismatch. However, there are no drop counters to show if a TPID mismatch takes place.

● Virtual wire deployment does not support packets with mismatched TPID.

Deployment Example

After configuring 802.1Q-in-Q, the Thunder device supports the following topologies:

● Transit traffic passing through the Thunder device using C-TAG and -STAG

Figure 2 : Double tagged transit traffic
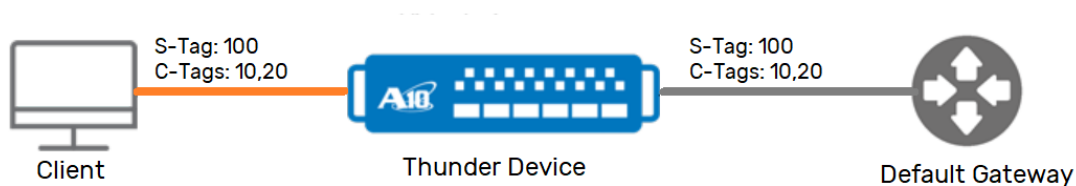
● Network Packet Brokers (NPB) using the two tags (C-TAG and -STAG) to interface with various network security devices (Firewall, encryption/decryption solutions).

Figure 3 : Double tagged traffic from a Network Packet Broker
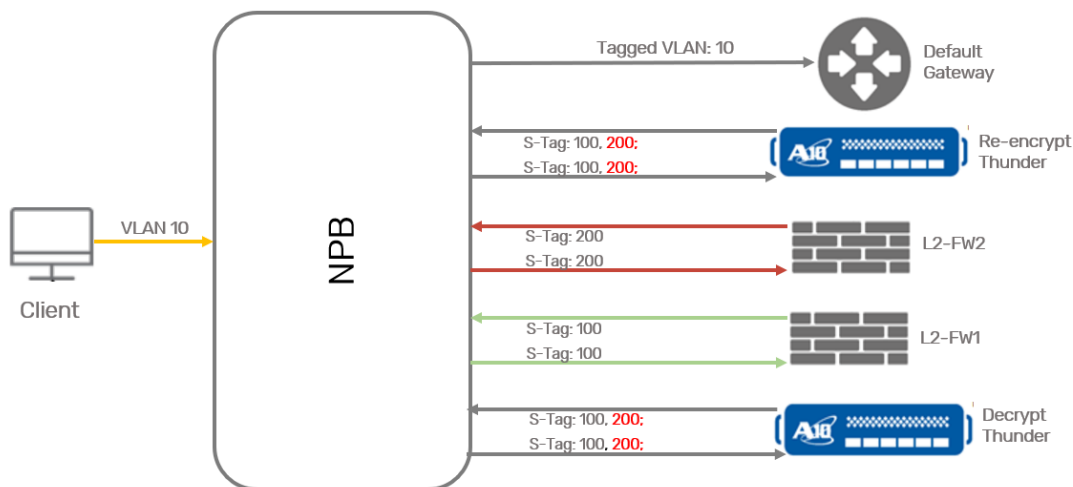
Q-in-Q VLAN Tagged Frame Format

A 802.1q tag is a four byte field in an ethernet frame, positioned right after the source MAC address. It consists of the following:

- Tag Protocol Identifier (TPID), a 16 bit value identifying the frame is IEEE 802.1Q tagged frame. It replaces the EtherType field in untagged frames.

- Tag Control Info (TCI), a 16 bit field containing the VLAN Identifier (12 bit) and other miscellaneous information (4 bits).

In a double tagged packet, two 802.1q tags are placed one after another as shown in the following figure,

Figure 4 : Q-in-Q VLAN tagged frame
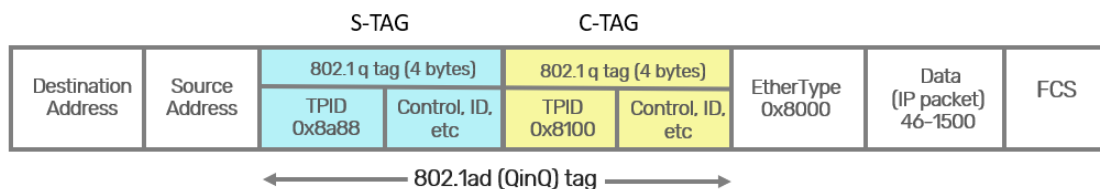
CLI Configuration

- To configure the support on all the physical ports, use the following command:

```
ACOS(config)# system q-in-q
ACOS(config-q-in-q)# enable-all-ports
```

  - For non-FPGA platform (Thunder 3350 series), when this command is set, ports 1 to 8 accept only VLAN traffic, whereas ports 9 to 20 accept both Q-in-Q and VLAN traffic.

NOTE:	Once activated, Q-in-Q is enabled across all supported ports on a Thunder device.

- To disable the support on all the physical ports, use the following command:

```
ACOS(config)# system q-in-q
ACOS(config-q-in-q)# no enable-all-ports
```

- To customize the TPID values for inner or outer VLANs, use the following command:

```
ACOS(config)# system q-in-q
ACOS(config-q-in-q)# inner-tpid 8100
ACOS(config-q-in-q)# outer-tpid 8100
ACOS(config-q-in-q)# enable-all-ports
```

For more information on the `system q-in-q` command, see *Command Line Interface Reference*.

Virtual Wire

The following topics are covered:

Feedback

Virtual Wire Overview

A virtual wire or bump-in-the-wire deployment consists of two interfaces that are logically bridged, forming a virtual wire pair. For packets that traverse the virtual wire endpoints, there is no Layer 2 modification, and the endpoints do not participate in MAC learning, ARP, or route lookups. The Thunder virtual wire deployment is an IP-Less configuration, without VLAN membership, and changes to the route tables on the neighboring devices are not required.

Virtual wire endpoints can be configured as:

- Ethernet interfaces
- Trunk ports (static or LACP)
- Virtual wire Ethernet group ports

The following diagrams are examples of virtual wire deployments:

Figure 5 : Virtual Wire - Endpoints: E1 and E2
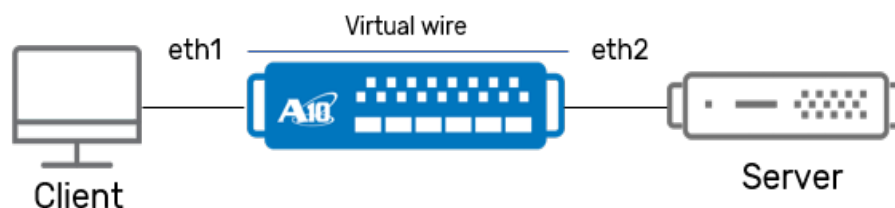
Figure 6 : Virtual Wire – Endpoints: E1 and E2 with mixed L3 environment
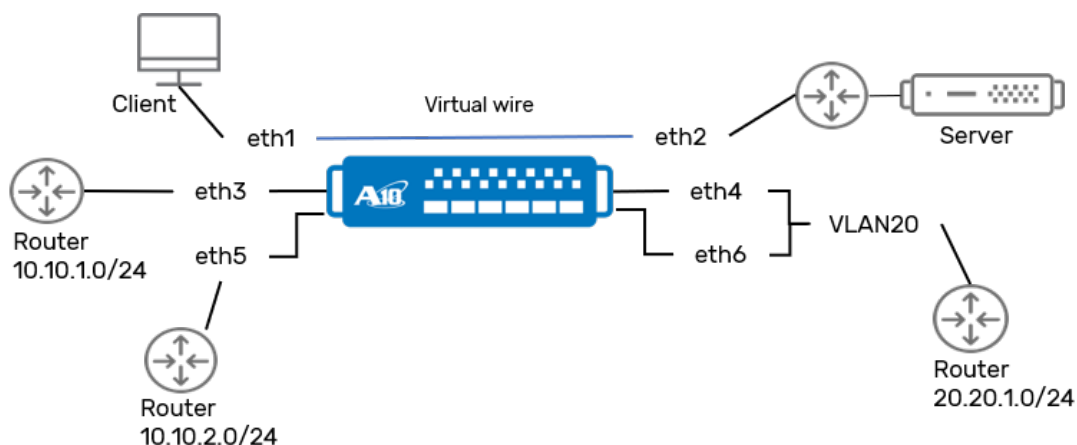
Configuration Overview

The following are key points of the virtual wire configuration on Thunder devices:

- Two interfaces are configured as virtual wire interface endpoints.

- Two virtual wire interfaces are bound to a virtual wire pair with a maximum of 32 pairs.

- When a packet comes in from one virtual wire interface endpoint, the packet leaves on the other virtual interface endpoint of the pair, based on the load balancing decision or SSLi decision.

- There is no MAC or ARP learning for the virtual wire pair endpoints.

- Virtual wire is IP-Less and does not support source NAT.

- The Layer 2 header of the packet is not modified when traversing the virtual wire pair (bidirectionally).

- The MTU of the virtual wire endpoints (ingress and egress) must be the same. If an Ethernet interface is added to a virtual wire port group, all members of the group must have the same MTU.

- The link status of the virtual wire is down if only one endpoint of the virtual wire is down; if one side is set to disable or is down, the state of the other endpoint is down. If the endpoint is configured as a trunk or virtual wire port group, the

endpoint is marked down when all Ethernet ports of the trunk or virtual wire port group are down, for the endpoint to be down.

- The following cannot be configured on the virtual wire interface:
  - IP and IPv6 addressing
  - BFD
  - LLDP
  - IS-IS router interface
  - NTPv6
  - snmp-server (trap source support).
  - Spanning-tree Protocol

Virtual wire is supported for Layer 4-7 load balancing and SSLi. For more information, refer to the *Application Delivery and Server Load Balancing Guide* or the *SSLi Configuration Guide*.

Configuration Examples

This section provides configuration examples for a virtual wire deployment.

To configure a virtual wire use the following command and configure the ID number.

```
ACOS(config)# virtual-wire <1-32>
```

The maximum number of virtual wire pairs is 32.

The following command configures Ethernet 1 and Ethernet 2 as virtual wire endpoints with ID 10:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# virtual-wire
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# virtual-wire
ACOS(config)# virtual-wire 10
ACOS(config-virtual-wire:10)# ethernet 1 ethernet 2
```

The following command configures E1 interface as an endpoint and an LACP trunk group 20 (E2 and E3) as the other endpoint for the virtual wire pair (5).

```
ACOS(config)# interface ethernet 1
```

```
ACOS(config-if:ethernet:1)# virtual-wire
ACOS(config-if:ethernet:1)# interface ethernet 2
ACOS(config-if:ethernet:2)# trunk-group 20 lacp
ACOS(config-if:ethernet:2)# interface ethernet 3
ACOS(config-if:ethernet:3)# trunk-group 20 lacp
ACOS(config-if:ethernet:3)# exit
ACOS(config)# interface trunk 20
ACOS(config-if:trunk:20)# virtual-wire
ACOS(config-if:trunk:20)# virtual-wire 5
ACOS(config-virtual-wire:5)# ethernet 1 trunk 20
```

The following show commands display virtual wire and virtual wire Ethernet group statistics. If a virtual wire has no interface member, it will not be shown:

```
ACOS(config)# show virtual-wire
Virtual Wire 1 state: Up
Virtual Wire 2 state: Down
ACOS(config)# show virtual-wire [number] statistics
Virtual Wire 1
----------------------------------------------------------
Counter Value
----------------------------------------------------------
ethernet 1 Input Packets 70
ethernet 1 Input Bytes 7154
ethernet 1 Output Packets 66
ethernet 1 Output Bytes 11619
ethernet 1 Dropped Packets 0
ethernet 2 Input Packets 69
ethernet 2 Input Bytes 11799
ethernet 2 Output Packets 67
ethernet 2 Output Bytes 6974
ethernet 2 Dropped Packets 0
ACOS(config)# show virtual-wire-ethernet-group [number]
Virtual wire ethernet group 4
----------------------------------------------------------
Members : 5 6
Status : Up Down
Leader port : 5
Group status : Up
```

For additional information on virtual wire solutions, refer to the *Application Delivery and Server Load Balancing Guide* or the *SSLi Configuration Guide*.

Limitation

Virtual wire is not supported with Virtual Chassis System (aVCS).

Virtual Wire Layer 2 Health Monitoring

ACOS provides the layer 2 health monitoring infrastructure support for Virtual wire, Layer 2, or Virtual Wire with SSLi (IP-less) deployments where monitoring other network devices is required for failover. With this feature, the system will monitor the end-to-end path health of Layer 2 and enables service resilience in event of path failure and allows faster corrective actions (such as switchover).

In a virtual wire deployment, various topologies can be considered. For example, you can invisibly build a firewall on a network environment by connecting two firewall ports (interfaces). The virtual wire connects the two interfaces logically; thus, the virtual wire is an 'internal path' to the firewall.

Figure 7 : Virtual Wire Firewall Deployment Example
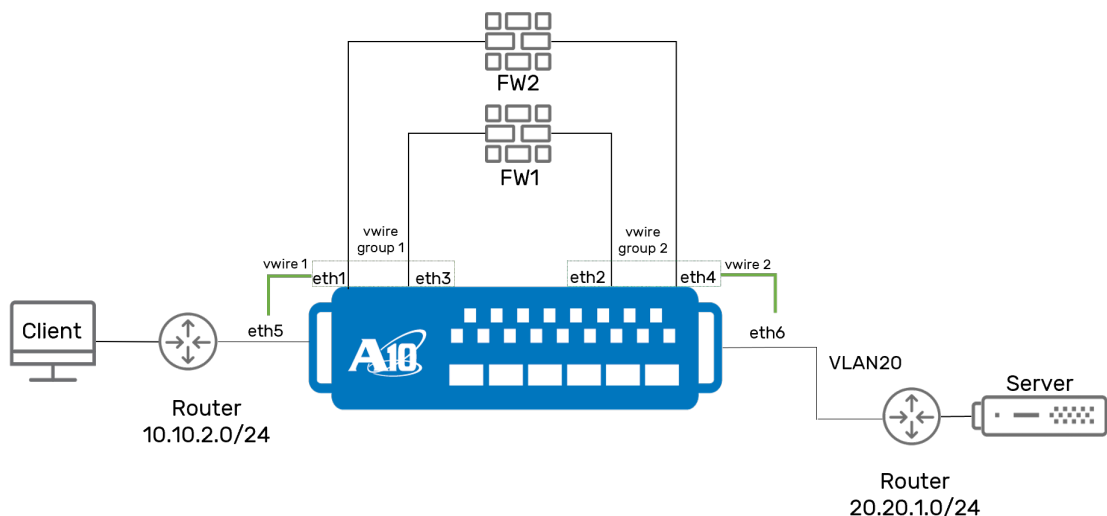
Heartbeats are used to communicate between the two nodes and to monitor the health of the internal paths. When the layer 2 health monitoring infrastructure is configured, it causes the endpoint interfaces in the path to continue transmitting

heartbeat messages to one another. If several heartbeat packets are not received on either side, the path is considered down.

In the case of firewall load balance by virtual wire or firewall service in SSLi deployment, this feature can detect if the defined internal path is up or down. Similarly, in firewall load balance by virtual wire, you can utilize an interface real server to achieve load balance.

This functionality can also be utilized to connect the path status to the real server interface. For example, if the internal path is blocked due to a firewall failure, ACOS can detect the change in the state and notify the interface server on the path to reflect the condition. Once the real server's state has been updated, the application can select a working server to process the subsequent request.

Configuration Overview

This topic describes the configuration workflow of layer 2 health monitoring feature.

1. The interface servers are configured.

2. The BFD feature must be enabled on the interface.

3. The Virtual wire is configured with the SSLi (IP-less) Deployment.

   For more information, see *Secure Socket Layer Insight (SSLi) Configuration Guide*.

4. The individual layer 2 health monitoring object and the layer 2 BFD are configured to provide a fast-forwarding path failure detection.

   Optionally, the layer 2 health monitoring feature (using layer 2 BFD) can be enabled at the global configuration mode. This setting reduces the overall configuration size by abstracting the common settings and overriding the individual setting

5. The layer 2 health monitoring system path object is defined at the global configuration mode.

6. The interface real server is configured for load balancing and the system path object is associated with the interface real server.

   The path object will notify the associated server when its state changes and the real server can synchronize it state according to the received event.

Configuration Examples

This topic describes how to enable and configure the layer 2 health monitoring infrastructure.

For detailed information on the CLI commands, see *Command Line Reference Guide*.

- Enable BFD on the global configuration level.

```
ACOS(config)# bfd enable
```

- Configure the individual health monitor/check object and then configure the layer 2 BFD settings.

```
ACOS(config)# system health-check hm-test
ACOS(config-health-check)# l2-bfd rx-interval 50 tx-interval 50
multiplier 3
```

Optionally, configure the global-level layer 2 health monitoring feature using layer 2 BFD.

```
ACOS(config)# l2-bfd
ACOS(config-l2-bfd)# ether-type 88B6
ACOS(config-l2-bfd)# rx-interval 50
ACOS(config-l2-bfd)# tx-interval 50
ACOS(config-l2-bfd)# multiplier 3
```

- Configure the layer 2 health monitor path object.

```
ACOS(config)#system path hm-obj-test
ACOS(config-path)#interface-pair ethernet 1 ethernet 2
ACOS(config-path)#use hm-test
```

- Configure the interface real server and associate the layer 2 health check path to indicate that the server state needs to be synchronized with the system path object.

```
ACOS(config)# slb server S1 ethernet 3
ACOS(config-real server)# l2-health-check-path hm-obj-test
ACOS(config-real server)# port 80 tcp
```

NOTE:	If the system path object setting is modified or removed, the interface real server setting will automatically be modified or removed.

Show Command Examples

This topic describes the show commands that can be used to view the various statistics of layer 2 health monitoring.

- View the server state synchronize with the system path object:

```
ACOS(config)#show system path
L2HM Path                          Health-Check      Path-State
Apps(Instances)
hm-obj-test hm-test                UP                SLB(625)

ACOS(config)#show slb server
Total Number of Servers configured: 2
Total Number of Services configured: 3
Current = Current Connections, Total = Total Connections
Fwd-pkt = Forward packets, Rev-pkt = Reverse packets
Service        Current     Total       Fwd-pkt    Rev-pkt    Peak-conn
State
---------------------------------------------------------------------------
---
gw:443/tcp     0           5           21         41         0          Up
gw: Total      0           5           21         41         0          Up
st:443/tcp     0           0           0          0          0          Up
st:8080/tcp    0           5           19         13         0          Up
st: Total      0           5           19         13         0          Up
....
```

- View the BFD neighbors statistics:

```
ACOS(config)#show bfd neighbors details
Our Address       00:0d:48:0a:83:0c
Neighbor Address 00:0d:48:0a:83:0b
Vlan 1
Clients Static
```

```
Singlehop, Echo disabled, Demand disabled, UDP source port 0
Asynchronous mode, Authentication None
CPU ID 3, Interface index 12
Local State Up, Remote State Up, 0h:1m:41s up
Local discriminator 0x00000004, Remote discriminator 0x00000005
Config DesiredMinTxInterval 800 milliseconds, RequiredMinRxInterval 800
milliseconds
Local DesiredMinTxInterval 800 milliseconds, RequiredMinRxInterval 800
milliseconds
Remote DesiredMinTxInterval 800 milliseconds, RequiredMinRxInterval 800
milliseconds
Local Multiplier 4, Remote Multiplier 4
Hold Down Time 3200 milliseconds, Transmit Interval 800 milliseconds
Local Diagnostic: No Diagnostic(0)
Remote Diagnostic: No Diagnostic(0)
Last sent echo sequcne number 0x00000000
Control Packet sent 143, received 143
Echo Packet sent 0, received 0
Dcmsg sent 4 Dcmsg rcvd 4
Session UP 2 times DOWN 0 times
```

Virtual Wire VLANs for Failover

ACOS supports the detection and tracking of active VLAN pairs in the virtual wire environment. Traffic is thus distributed seamlessly if the VLAN is changed during failover scenarios or in topologies where the traffic uses a different VLAN for forward and reverse traffic.

In a virtual wire with SSLi (IP-less) deployment, various topologies can be considered. See the Virtual Wire Layer 2 Health Monitoring from the previous section.

In enterprise SSLi deployments, packet brokers play a crucial role in managing network traffic. Packet brokers have two types of ports: network ports and tool ports. The network ports route the traffic to the network, and the tool ports send the service traffic to the security devices. Network ports are in pairs, with one facing the upstream port (for example, a switch) and the other facing the downstream port (for example, a firewall). The packet broker is inserted between the switch and the firewall, which acts as a "bump in the wire," see .

When there are multiple upstream and downstream ports, for example, multiple switches or firewall devices in a cluster to act as active-standby for failover scenarios, packet brokers use VLAN tagging to determine the source and destination of each packet. Packet broker inserts an outer VLAN tag using Q-in-Q technology for each network port pair. This tag acts as a signature for that specific port pair, allowing the packet broker to route traffic correctly.

In this kind of topology, it is important for ACOS to know about the VLAN pairs in the network and keep a track of which VLAN pair is active during failover. The virtual wire VLAN pairs facilitate ACOS to track the active VLAN pairs for virtual wire sessions. ACOS detects and updates the active VLAN pair in the network and forwards the traffic to the active server.

Figure 8 : Virtual Wire SSLi Firewall Deployment Example
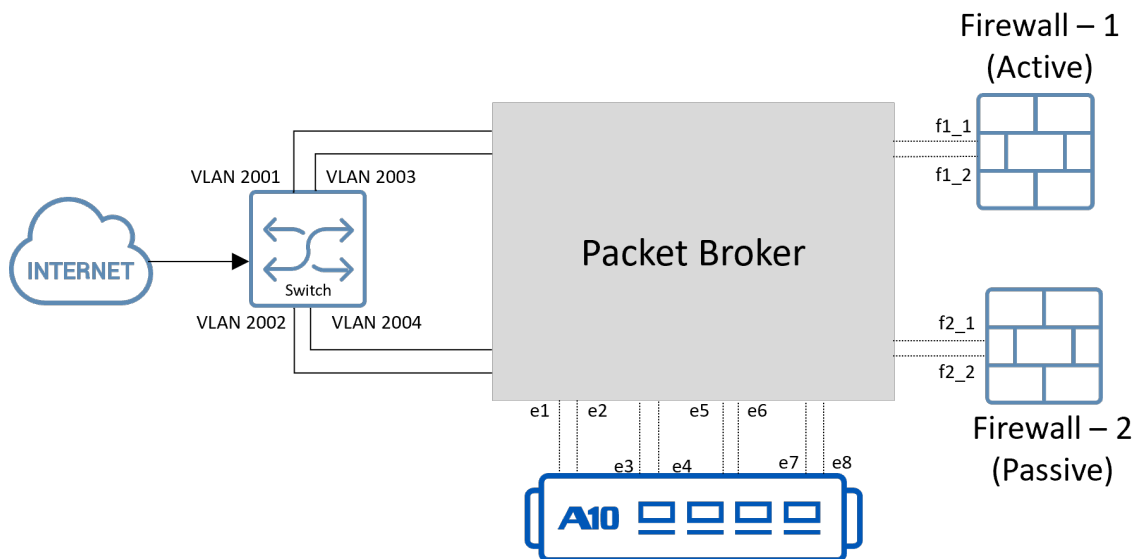
In the above example,

- The switch has 2 VLAN tags in a stable state that work concurrently. VLAN 2001 and 2003 maps with Firewall - 1 and VLAN 2002 and 2004 maps with Firewall – 2. After firewall failover, ACOS chooses another pair of VLANs to route the traffic.

- The switch inserts an outer VLAN tag into each network port pair. It inserts outer VLAN tag 2001 for the link between Firewall - 1 and switches; and outer VLAN tag 2002 for the connection between Firewall - 2 and switches.

There are four network port pairs: f1_1, f1_2, f2_1, and f2_2. So, the Q-in-Q VLAN mapping will be as follows:

- f1_1:2001

- f1_2:2003

- f2_1:2002

- f2_2:2004

- Each firewall consists of two ports that are used as a trunk. The traffic flows either on f1_1 and f1_2 or f2_1 and f2_2. In this case, Firewall-1 can choose either f1_1 or f1_2.

- There are two separate virtual-wire paths: e1 to e4 and e5 to e8. The packet broker distributes the traffic seamlessly to both these paths.

To ensure that traffic is forwarded to the active VLAN during failover, you can configure virtual wire VLAN pairs to facilitate ACOS to update and track the active VLAN tags for virtual wire sessions.

Configuration Overview

This topic describes the configuration workflow of the virtual wire VLAN feature.

1. The interface servers are configured as a virtual wire endpoint.

2. VLAN learning is enabled on the interface.

3. The active and passive VLAN pairs are configured.

4. The bridge VLAN groups are configured, and VLAN pairs are bound to this group.

5. The virtual wire VLAN pairs are configured, and the two active VLAN tags are mapped. Similarly, the two standby VLAN tags are mapped, which will be used during failover.

6. The virtual wire VLAN set is configured, and the virtual wire VLAN pairs are mapped to this set.

7. The virtual wire is configured with the SSLi (IP-less) Deployment.

    For more information, see *Secure Socket Layer Insight (SSLi) Configuration Guide*.

Configuration Examples

This topic describes how to enable and configure the virtual wire VLAN for failover.

For detailed information on the CLI commands, see *Command Line Interface Reference*.

1. Configure the ethernet interface(s) as a virtual wire endpoint. Additionally, configure the update-l2-info and VLAN learning options to let ACOS detect the active VLAN in the network.

   The **interface ethernet 1** to **4** are configured. Two for the client-side and two for the server-side.

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# name client-side
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# virtual-wire update-l2-info vlan-learning
enable mac-learning disable
ACOS(config-if:ethernet:1)# exit
ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# name client-side-path2
ACOS(config-if:ethernet:2)# enable
ACOS(config-if:ethernet:2)# virtual-wire update-l2-info vlan-learning
enable mac-learning disable
ACOS(config-if:ethernet:2)# exit
....
ACOS(config-if:ethernet:4)# exit
```

2. Configure the active and passive VLANs.

   The two VLAN pairs are configured. Active **VLAN 2001** and **2003**, and passive **VLAN 2002** and **2004**.

```
ACOS(config)# vlan 2001
ACOS(config-vlan:2001)# exit
ACOS(config)# vlan 2002
ACOS(config-vlan:2002)# exit
ACOS(config)# vlan 2003
ACOS(config-vlan:2003)# exit
ACOS(config)# vlan 2004
```

```
ACOS(config-vlan:2004)# exit
```

3. Configure the global options in the virtual wire. This is an optional configuration and affect how ACOS updates and tracks the active VLAN.

   The virtual wire update period option is set to **60** seconds. The update active VLAN option is set to **all**, forcing ACOS to update VLAN by any packet every 60 seconds.

```
ACOS(config)# virtual-wire-global
ACOS(config-virtual-wire-global)# vlan-update-period 60
ACOS(config-virtual-wire-global)# update-active-vlan all
```

4. Configure the bridge VLAN group to match the session for packets with VLAN in both VLAN pairs.

   The bridge VLAN group **1** is configured. The previously defined active and passive **VLAN 2001 to 2004** is configured for VLAN-to-VLAN bridging.

```
ACOS(config)# bridge-vlan-group 1
ACOS(config-bridge-vlan-group:1)# vlan 2001 to 2004
```

5. Configure and map the two active VLAN with the virtual wire.

   The virtual wire VLAN pair **1** is configured. The previously defined active **VLAN 2001** and **2003** are bound to the virtual wire VLAN pair.

```
ACOS(config)# virtual-wire-vlan-pair 1
ACOS(config-virtual-wire-vlan-pair:1)# vlan 2001 vlan 2003
```

6. Configure and map the two passive VLAN with the virtual wire, which will be used after failover.

   The virtual wire VLAN pair **2** is configured. The previously defined passive **VLAN 2002** and **2004** are bound to the virtual wire VLAN pair.

```
ACOS(config)# virtual-wire-vlan-pair 2
ACOS(config-virtual-wire-vlan-pair:2)# vlan 2002 vlan 2004
```

7. Configure and map the two VLAN pairs in a virtual wire pair set.

   The virtual wire VLAN pair set **1** is configured. The previously defined active and passive virtual wire VLAN pairs **1** and **2** are configured.

```
ACOS(config)# virtual-wire-vlan-pair-set 1
```

```
ACOS(config-virtual-wire-vlan-pair-set:1)# virtual-wire-vlan-pair 1
ACOS(config-virtual-wire-vlan-pair-set:1)# virtual-wire-vlan-pair 2
```

Additional Notes

- ACOS detects and updates the active VLAN pair in the network and forwards the traffic to the active server. Moreover, ACOS matches the packet to the correct virtual wire session even if the forward and reverse traffic use a different VLAN or if the VLAN is changed due to firewall failover.

- The active VLAN is updated in the order specified in the virtual wire VLAN pair. For example, if a virtual-wire session used active **VLAN 2001**, and the active VLAN for **virtual-wire-vlan-pair 2** needs to be updated, then ACOS updates VLAN to **2002** because 2001 and 2002 are the first VLAN in the pairs.

Show Command Examples

This topic describes the show commands that can be used for monitoring virtual wire VLAN feature.

- To view the active VLAN pair, use the `show virtual-wire-global vlan-set-active-member` command:

```
ACOS(config)#show virtual-wire-global vlan-set-active-member
virtual-wire-vlan-set: 3
active vlan-pair : 2
```

- To track the active VLAN pair update count, use the `show virtual-wire-global counter` command:

```
ACOS(config)#show virtual-wire-global counter
VLAN update : 0
MAC update : 0
VLAN pair update: 2
```

Limitations

- This feature is supported only for slow path.

  For example, the VLAN tagging with Q-in-Q technology and SSL inspection are often used in 'slow paths,' where the traffic needs to be inspected, processed, and managed more thoroughly.

- Making changes to `virtual-wire-vlan-pair` or `virtual-wire-vlan-pair-set` configurations in production may cause some virtual wire sessions to use the incorrect active VLAN.

Virtual Wire Layer 3 Health Monitoring

ACOS provides layer 3 health monitoring infrastructure support for virtual wire and virtual wire with SSLi (IP-less), which operates transparently in high-availability deployments. In these deployments, two firewall devices manage the active and standby paths during failover scenarios.

Consider a deployment where Thunder SSLi A and Firewall A are active, while Thunder SSLi B and Firewall B are on standby. During a switch from active to standby, ACOS might continue to send packets for previously established sessions, causing traffic to route incorrectly and disturbing network traffic.

To address these scenarios, Layer 3 health checks can be configured on the VLAN that needs monitoring. The health check uses ping, gratuitous ARP (GARP), and packet count methods on the target device, allowing ACOS to detect which VLAN is in active or standby mode.

When this feature is implemented, the Thunder SSLi and Firewall devices (A and B) are aware of their active or standby status for each VLAN. Additionally, the health check prevents ACOS from sending the packets to the ACOS standby devices, thus avoiding network disturbances. The health check works on ACOS device only.

For detailed deployment and implementation of layer 3 health monitoring in Virtual wire with SSLi (IP-Less) deployment, see SSL Insight (SSLi) Configuration Guide.

Configuration Overview

This topic describes the configuration workflow of layer 3 health monitoring feature.

1. The interface servers are configured.

2. The VLANs are configured in the gateway and the Firewall servers.

3. The firewall servers are configured with VRRP-A for high availability, where one is in active path and the other is in standby path.

   For more information, see Firewall Configuration Guide and Configuring VRRP-A High Availability.

4. The Virtual wire is configured with the SSLi (IP-less) Deployment, which is transparently integrated in the network path.

   For more information, see SSL Insight (SSLi) Configuration Guide.

5. The layer 3 health check is set up on the specific VLANs using different methods to monitor the status of network paths.

6. The health check detects the status of VLAN in the network and prevents ACOS from sending packet for session in the standby VLAN.

Configuration Examples

This topic describes how to enable and configure layer 3 health monitoring with different health check methods. Health monitoring is implemented to the VLAN that needs monitoring.

For detailed information on CLI commands and sub-commands, see the virtual-wire-health-check command in the *Command Line Interface Reference*.

Example 1: Configure Ping Health Check Method

The ping method monitors the traffic of the virtual port on the VLAN. ACOS sends an ICMP echo request to the target IP address and checks for the response. If it does not receive a response to two consecutive echo requests, the VLAN status is marked as standby.

To configure the ping method health check, use the following commands:

```
ACOS(config)# virtual-wire-health-check vlan 10 method ping
ACOS(config-virtual-wire-health-check:10)# interface ethernet 4
ACOS(config-virtual-wire-health-check:10)# interval 5
ACOS(config-virtual-wire-health-check:10)# nexthop-ip 10.10.10.3
ACOS(config-virtual-wire-health-check:10)# enable
```

In the above example, the ping health check is enabled for **vlan 10**. The **interface** is the output interface through which the echo request (ping) will be sent. The **nexthop-ip** is the IPv4 target address. The time interval is set to **5** seconds, which determines how frequently the health check entry will be checked.

Example 2: Configure Ping Health Check Method using Source and Nexthop Addresses

Optionally, the ping method can be configured with the IP and MAC addresses of the source and nexthop. This means that you can specify the nexthop IPv4 and MAC address of the target device. Similarly, you can specify the source IP address and MAC address that should be used for the ping request. ACOS will send the ping request directly using this information without inspecting the traffic.

NOTE:	If only part of the information (**source-ip** or **nexthop-ip**) is configured, ACOS will analyze the traffic to gather the remaining fields and then send the ping request.

To configure the ping method health check using source and nexthop addresses, use the following commands:

```
ACOS(config)# virtual-wire-health-check vlan 10 method ping
ACOS(config-virtual-wire-health-check:10)# interface ethernet 4
ACOS(config-virtual-wire-health-check:10)# interval 5
ACOS(config-virtual-wire-health-check:10)# nexthop-ip 10.10.10.3
nexthop-mac 000c.2991.3b76
ACOS(config-virtual-wire-health-check:10)# source-ip 10.10.10.123
source-mac 1122.3344.5566
ACOS(config-virtual-wire-health-check:10)# enable
```

Example 3: Configure GARP Health Check Method

The Gratuitous ARP (GARP) method monitors the GARP packets of the target address or interface for a specified interval. If GARP packets are not received within the configured interval, the VLAN status is marked as standby.

To configure the GARP method health check, use the following commands:

```
ACOS(config)# virtual-wire-health-check vlan 10 method garp
ACOS(config-virtual-wire-health-check:10)# interface ethernet 4
ACOS(config-virtual-wire-health-check:10)# interval 5
ACOS(config-virtual-wire-health-check:10)# nexthop-ip 10.10.10.3
ACOS(config-virtual-wire-health-check:10)# standby-interface ethernet 5
ACOS(config-virtual-wire-health-check:10)# enable
```

In the above example, the GARP health check is enabled for **vlan 10**. The `interface` will receive the GARP packets when the device is in active mode. Similarly, if ACOS receives GARP packet from the `standby-interface`, the VLAN is changed to standby immediately without waiting for the configured interval. The `nexthop-ip` is the IPv4 address of the target device. The time interval is set to **5** seconds, which determines how frequently the health check entry will be checked.

Example 4: Configure Packet Count Health Check Method

The packet count method monitors the packet rate on the VLAN for a specified active threshold. If it falls below this threshold, the VLAN status is marked as standby.

To configure the packet count method health check, use the following commands:

```
ACOS(config)# virtual-wire-health-check vlan 10 method packet-count
ACOS(config-virtual-wire-health-check:10)# interface ethernet 4
ACOS(config-virtual-wire-health-check:10)# interval 5
ACOS(config-virtual-wire-health-check:10)# active-threshold 5
ACOS(config-virtual-wire-health-check:10)# enable
```

In the above example, the packet count health check is enabled for **vlan 10**. The `interface` is the target interface to monitor the packet count. The `interval` is set to **5** seconds, which determines how frequently ACOS checks the packet rate on the specified interface. The `active-threshold` is set to 5 (packet per second).

- If the packet rate on the target interface falls below this threshold, the VLAN status is marked as standby.

- If the packet rate is above this threshold, the VLAN status is marked as active.

Example 5: Bind Partition Health Check to the Health Check Methods

Optionally, the partition health check can be bound to the ping, GARP, and packet count health check methods. This health check monitors the VLAN status of the

Feedback

entire partition, which is considered in standby mode. All the VLAN configured with virtual-wire health-check within the partition will be considered as standby mode.

```
ACOS(config)# virtual-wire-health-check vlan 10 method ping
ACOS(config-virtual-wire-health-check:10)# interface ethernet 4
ACOS(config-virtual-wire-health-check:10)# interval 5
ACOS(config-virtual-wire-health-check:10)# nexthop-ip 10.10.10.3
ACOS(config-virtual-wire-health-check:10)# partition-health-check
ACOS(config-virtual-wire-health-check:10)# enable
```

Example 6: Configure Additional Health Check Options

- Optionally, if you want to count only the Layer 3 packets in the packet count health check method, you can enable `l3-packet` command.

  To configure the layer 3 packets in GARP method, use the following commands:

```
ACOS(config)# virtual-wire-health-check vlan 10 method garp
ACOS(config-virtual-wire-health-check:10)# interface ethernet 4
ACOS(config-virtual-wire-health-check:10)# interval 5
ACOS(config-virtual-wire-health-check:10)# nexthop-ip 10.10.10.3
ACOS(config-virtual-wire-health-check:10)# standby-interface ethernet 5
ACOS(config-virtual-wire-health-check:10)# l3-packet
ACOS(config-virtual-wire-health-check:10)# enable
```

NOTE:	In case of packet count health check method, ACOS only counts the packet rate for IPv4 and IPv6 packets, excluding other types of packets from the count.

- Optionally, if you want to enable inner VLAN ID in an 802.1Q-in-Q environment for ping and packet count health check methods, you can use `inner-vlan` command.

  To configure the inner VLAN in ping method, use the following commands:

```
ACOS(config)# virtual-wire-health-check vlan 10 method ping
ACOS(config-virtual-wire-health-check:10)# interface ethernet 4
ACOS(config-virtual-wire-health-check:10)# interval 5
ACOS(config-virtual-wire-health-check:10)# nexthop-ip 10.10.10.3
ACOS(config-virtual-wire-health-check:10)# inner-vlan 2
```

```
ACOS(config-virtual-wire-health-check:10)# enable
```

- In case of ping health check method, ACOS sends echo packets with inner and outer VLAN tags to ensure correct echo requests reach their target devices within the Q-in-Q environment.

- In case of packet count health check method, ACOS only counts the packet rate for packets with the specified inner VLAN ID.

Show Command Examples

This topic describes show commands that can be used to view the various statistics of layer 3 health monitoring.

- To view the health check entry status of all VLANs, use the following show command:

```
ACOS(config)#show virtual-wire-health-check statistics
Entry VLAN: 10
Entry state: Enabled
VLAN state: Active

Entry VLAN: 20
Entry state: Disabled
VLAN state: Active

Entry VLAN: 30
Entry state: Enabled
VLAN state: Standby
```

- To view the specific VLAN statistics for tracking active or standby events, use the following show command:

```
ACOS(config)#show virtual-wire-health-check vlan 40 statistics
Entry VLAN: 20
Active event: 0
Standby event: 1
```

- To view the 'standby drop' counter for packets dropped due to VLAN in standby mode, use the following show command:

```
ACOS(config)# show virtual-wire-global counters
VLAN update      : 0
MAC update       : 0
VLAN pair update: 0
standby drop     : 2
```

Limitation

IPv6 address is not supported and cannot be configured for target device or interface.

Traffic Distribution Mode

ACOS supports traffic distribution mode configuration, which uses client-side and server-side VLAN or interface ethernet to evenly distribute traffic to a specific destination such as two servers or processing units or modules. This feature is mainly used to improve layer 4 load distribution in platforms with multiple processing units (multi-PU).

Thunder 7650/7655S multi-PU platforms are designed for large-scale traffic distribution, load balancing, and scalability. PU1 and PU2 are the two active PUs in the multi-PU architecture. The traffic for the same object is distributed among the two PUs running the specified object.

ACOS offers a variety of traffic distribution modes or options to fulfill your business requirements and distribute the load across PUs effortlessly.

The following topics are covered:

Traffic Flow

This section explains the traffic distribution flow and key configuration guidelines.

The traffic distribution uses a two-tuple (source IP and destination IP) or three-tuple (source IP, destination IP, and protocol type) hash mechanism to route the traffic to the backend server or instances. This means the traffic from the same client is sent to the same backend instance.

Figure 9 : Traffic Distribution Flow Example
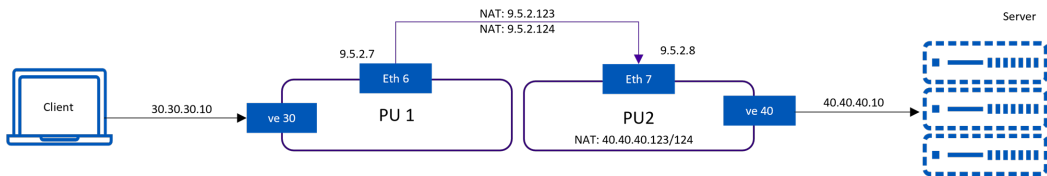
In this example, the input traffic (ve30, eth7) is hashed based on the source IP. Similarly, the output traffic (eth6, ve40) is hashed based on the destination IP. This ensures that the same traffic flow is processed on the same PU based on the client's IP. This is achieved by configuring the `traffic-distribution-mode` command in the interface or VLAN. Here, `traffic-distribution-mode sip` is configured for eth7, and `traffic-distribution-mode dip` is configured for eth6.

Since, the traffic for the same object is distributed to the two PUs running the specific object, you must set the application type on the multi-PU platform using the `application-type[adc | cgn]` command. For more information on implementing the multi-PU platforms for ACOS products (such as ADC, SSLi, GSLB, etc.), see the respective Configuration Guide.

CLI Configuration

This section describes the CLI configuration examples for implementing traffic distribution.

The `traffic-distribution-mode` command can invoke multi-PU traffic hashing to both PUs. Based on your business requirements, the following options can be configured:

- `sip` - Distribute the traffic evenly to the multi-PUs (PU1 and PU2) based on the source IP address.
- `dip` - Distribute the traffic evenly to the multi-PUs (PU1 and PU2) based on the destination IP address.
- `l3-lookup` - Distribute the traffic evenly for a DMZ-facing interface in L2 mode.
- `primary` - Distribute the traffic to the primary (master) unit.
- `blade` - Distribute the traffic to the secondary (blade) unit.
- `l4-src-port` - Distribute the traffic to the multi-PUs based on the source port of Layer 4 Protocol (TCP/UDP).
- `l4-dst-port` - Distribute the traffic to the multi-PUs based on the destination port of Layer 4 Protocol (TCP/UDP).

For more information, see *Command Line Interface Reference*.

- Configuring traffic distribution mode on the interface ethernet.

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# speed-forced-40g
ACOS(config-if:ethernet:1)# enable
ACOS(config-if:ethernet:1)# traffic-distribution-mode sip
ACOS(config-if:ethernet:1)# exit

ACOS(config)# interface ethernet 2
ACOS(config-if:ethernet:2)# exit
...
ACOS(config)# interface ethernet 5
ACOS(config-if:ethernet:5)# speed-forced-40g
ACOS(config-if:ethernet:5)# enable
ACOS(config-if:ethernet:5)# traffic-distribution-mode dip
ACOS(config-if:ethernet:5)# exit
```

- Configuring traffic distribution mode on the VLAN.

```
ACOS(config)# vlan 10
ACOS(config-vlan:10)# tagged ethernet 1
ACOS(config-vlan:10)# router-interface ve 10
ACOS(config-vlan:10)# traffic-distribution-mode sip
ACOS(config-vlan:10)# exit

ACOS(config)# vlan 20
ACOS(config-vlan:20)# tagged ethernet 1
ACOS(config-vlan:20)# router-interface ve 20
ACOS(config-vlan:20)# traffic-distribution-mode dip
ACOS(config-vlan:20)# exit

ACOS(config)# interface ve 10
ACOS(config-if:ve:10)# ip address 10.31.8.35 255.255.255.0
ACOS(config-if:ve:10)# exit
ACOS(config)# interface ve 20
ACOS(config-if:ve:20)# ip address 10.31.9.35 255.255.255.0
ACOS(config-if:ve:20)# ipv6 address 2602:803:c002::500:142/126
ACOS(config-if:ve:20)# ipv6 enable
ACOS(config-if:ve:20)# exit
```

Layer 3 Networking

The following chapters are covered in this part/section:

Dynamic Host Configuration Protocol (DHCP)

Two Way Active Measurement Protocol (TWAMP)

Dynamic Host Configuration Protocol (DHCP)

The following topics are covered:

Overview of DHCP

Dynamic Host Configuration Protocol (DHCP) is a mechanism commonly used by clients to auto-discover their addressing and other configuration information when connected to a network. On ACOS devices, DHCP configuration supports IP address, subnet masks, default gateway, and classless static routes (option 121) from the DHCP server.

You can enable use of DHCP to dynamically configure IP addresses on the following types of interfaces:

- Management interface – A single IP address can be assigned.

- Ethernet data interfaces – Multiple IP addresses can be assigned.

- Virtual ethernet interfaces – Multiple IP addresses can be assigned.

- Trunk interfaces – Multiple IP addresses can be assigned.

Virtual servers and IP NAT pools are also able to use the DHCP-assigned address of a given data interface. If this option is enabled, ACOS updates the VIP or pool address any time the specified data interface's IP address is changed by DHCP.

Notes

- DHCP can be enabled on an interface only if that interface does not already have any statically assigned IP addresses.

- On ACOS devices deployed in gateway (Layer 3) mode, Ethernet data interfaces can have multiple IP addresses. An interface can have a combination of dynamically assigned addresses (by DHCP) and statically configured addresses. However, if you plan to use both methods of address configuration, static addresses can be configured only after you finish using DHCP to dynamically configure addresses. To use DHCP in this case, you must first delete all the statically configured IP addresses from the interface.

- On vThunder models, if single-IP mode is used, DHCP can be enabled only at the physical interface level.

- On devices deployed in Transparent (Layer 2) mode:

    o You can enable DHCP on the management interface and at the global level.

    o The VIP address and pool NAT address (if used) should match the global data IP address of the device. Make sure to enable this option when configuring the VIP or pool.

Enabling DHCP

Using the GUI

1. Hover over **Network** in the navigation bar, and select **Interface** from the drop-down menu.

2. Depending on the type of interface on which to configure this feature, select LAN, Virtual Ethernet or Trunk from the menu bar.

3. Click **Edit** in the actions column for the interface on which to configure this feature.

4. Expand the IP section to reveal additional configuration options.

5. Select the checkbox in the DHCP field.

6. Click **Update**.

Using the CLI

To enable DHCP on an interface, use the `ip address dhcp` command at the configuration level for the interface:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip address dhcp
```

Configuring DHCP Relays

The following topics are covered:

Overview of DHCP Relays

This section describes DHCP relay support and how to configure it.

You can configure the ACOS device to relay DHCP traffic between DHCP clients and DHCP servers located in different VLANs or subnets.

DHCP relay is supported only for the standard DHCP protocol ports:

- Boot protocol server (BOOTPS) – UDP port 67

- Boot protocol client (BOOTPC) – UDP port 68

DHCP relay service is supported for IPv4 and IPv6.

DHCP is a Client-Server protocol and relies on broadcast communication between the client and server for packet exchanges. Accordingly, the clients and the servers must be in the same broadcast domain (Layer 2 VLAN) for this to work, since Layer 3 routers typically do not forward broadcasts. However, in most deployments it is not practical to have a DHCP server in each Layer 2 VLAN. Instead, it is typical to use a common DHCP server for all VLANs and subnets in the network.

Notes

- In the current release, the helper-address feature provides service for DHCP packets only.

- The interface on which the helper address is configured must have an IP address.

- The helper address cannot be the same as the IP address on any interface or an IP address used for SLB.

Configuring DHCP Relays

To enable DHCP communication between different VLANs or subnets, you can use a DHCP relay. A DHCP relay acts as a mediator between the DHCP client and the DHCP server when they are not in the same broadcast domain.

To configure the ACOS device as a DHCP relay, configure the DHCP server IP address as a helper address on the IP interface connected to DHCP clients. The ACOS device intercepts broadcast DHCP packets sent by clients on the interface configured with the helper address.

The ACOS device then places the receiving interface's IP address (not the helper address) in the relay gateway address field, and forwards the DHCP packet to the server. When the DHCP server replies, the ACOS device forwards the response to the client.

Using the GUI to Configure a DHCP Relay

To configure a helper address for the IP interface connected to the DHCP clients:

1. Hover over **Network** in the navigation bar, and select **Interface** from the drop-down menu.

2. Depending on the type of interface on which to configure this feature, select LAN, Virtual Ethernet or Trunk from the menu bar.

3. Click **Edit** in the actions column for the interface on which to configure this feature.

4. Expand the IP section to reveal additional configuration options.

5. Specify an IP address for the IP Helper Address field.

6. Click **Add**.

7. You can add up to 2 helper addresses per interface.

8. Click **Update**.

Using the CLI to Configure a DHCP Relay

The following commands configure two helper addresses. The helper address for DHCP server 100.100.100.1 is configured on Ethernet interface 1 and on Virtual Ethernet (VE) interfaces 5 and 7. The helper address for DHCP server 20.20.20.102 is configured on VE 9.

NOTE:	You can configure up to 2 IP helper addresses per Ethernet interface.

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip helper-address 100.100.100.1
ACOS(config-if:ethernet:1)# exit
```

```
ACOS(config)# interface ve 5
ACOS(config-if:ve:5)# ip helper-address 100.100.100.1
ACOS(config-if:ve:5)# exit
ACOS(config)# interface ve 7
ACOS(config-if:ve:7)# ip helper-address 100.100.100.1
ACOS(config-if:ve:7)# exit
ACOS(config)# interface ve 9
ACOS(config-if:ve:9)# ip helper-address 20.20.20.102
```

Use the `show ip helper-address` command shows summary DHCP relay information:

```
ACOS(config)# show ip helper-address
Interface   Helper-Address          RX           TX         No-Relay
 Drops
---------   --------------   ------------  ------------  ------------  -----
-------
eth1        100.100.100.1              0            0             0
      0
ve5         100.100.100.1           1669         1668             0
      1
ve7                                  1668         1668             0
      0
ve8         100.100.100.1              0            0             0
      0
ve9         20.20.20.102               0            0             0
      0
```

Use the `detail` parameter to view additional detailed DHCP relay information:

```
ACOS# show ip helper-address detail
IP Interface: eth1
------------
  Helper-Address: 100.100.100.1
  Packets:
          RX: 0
             BootRequest Packets : 0
             BootReply Packets   : 0
          TX: 0
             BootRequest Packets : 0
             BootReply Packets   : 0
  No-Relay: 0
  Drops:
```

```
              Invalid BOOTP Port  : 0
              Invalid IP/UDP Len  : 0
              Invalid DHCP Oper   : 0
              Exceeded DHCP Hops  : 0
              Invalid Dest IP     : 0
              Exceeded TTL        : 0
              No Route to Dest    : 0
              Dest Processing Err : 0


IP Interface: ve5
------------
  Helper-Address: 100.100.100.1
  Packets:
              RX: 16
                  BootRequest Packets : 16
                  BootReply Packets   : 0
              TX: 14
                  BootRequest Packets : 0
                  BootReply Packets   : 14
  No-Relay: 0
  Drops:
              Invalid BOOTP Port  : 0
              Invalid IP/UDP Len  : 0
              Invalid DHCP Oper   : 0
              Exceeded DHCP Hops  : 0
              Invalid Dest IP     : 0
              Exceeded TTL        : 0
              No Route to Dest    : 2
              Dest Processing Err : 0


IP Interface: ve7
------------
  Helper-Address: None
  Packets:
              RX: 14
                  BootRequest Packets : 0
                  BootReply Packets   : 14
              TX: 14
                  BootRequest Packets : 14
                  BootReply Packets   : 0
```

```
No-Relay: 0
Drops:
          Invalid BOOTP Port  : 0
          Invalid IP/UDP Len  : 0
          Invalid DHCP Oper   : 0
          Exceeded DHCP Hops  : 0
          Invalid Dest IP     : 0
          Exceeded TTL        : 0
          No Route to Dest    : 0
          Dest Processing Err : 0
```

Descriptions for the fields in both outputs are available in the *Command Line Interface Reference*.

The following command clears the DHCP relay counters:

```
ACOS# clear ip helper-address statistics
```

Two Way Active Measurement Protocol (TWAMP)

The Two-Way Active Measurement Protocol or **TWAMP Light Version Responder** is used for the implementation of both IPv4 and IPv6. The RFC 5357 Two-Way Active Measurement Protocol (TWAMP) is an IP QoS network measurement protocol. TWAMP provides QoS scrutiny of circular-tour performance between two network endpoints. The Two-Way Active Management Protocol (TWAMP), defined in RFC 5357, is the next ideal version or an extension of the One-Way Active Management Protocol (OWAMP) that provisions two-way or circular-tour mode quantities instead of unidirectional competences.

The following topics are covered:

Feedback

Overview

The following topics are covered:

What is TWAMP?

The TWAMP (Two Way Active Measurement Protocol) is a key function to keep track of the service quality of 5G mobile service by monitoring some of the key L3 networking indicators such as latency/delay and abnormal packet drops. The TWAMP has an edge over OWAMP (One Way Active Measurement Protocol) with more robust features. As a cutting-edge technology, 5G requires more stable and low-latency between massive mobile devices such as connected cars or any other devices. It requires any ultra-lower latency like remote operation.

Advantages of TWAMP

The significant advantages of the TWAMP feature are as the following:

- An easy feature to set-up and use with a great customer benefits.

- Time synchronization is not mandatory for TWAMP, whereas for OWAMP it is required.

- It is implemented by most of the major network equipment manufacturer already or in process of implementing it.

- This protocol is used to measure the QoS (Quality of Service) KPIs between any two point of IP (Layer 3) Network.

KPIs of TWAMP

The following are the TWAMP KPIs:

- Lost Packets

- Out of Order Packets

- Duplicate Packets

- Latency/Delay

- Packet Delay Variation – PDV (as defined in the ITU-T Y.1540 standard)

- IP Delay Variation – IPDV (as defined in the RFC-3550 standard)

- Inter Delay - time taken by the TWAMP responder to process the incoming packets and respond.

- TTL

NOTE: All these metrics are available for One Way or Two Way (Round Trip) Measurements. The Two Way (Round Trip) measurement do not require the time synchronization for both end points, whereas the One-Way Delay/Latency measurement, for that specific case, both end points must be time synchronized.

Feature Description

The following are the various aspects of this new feature.

The following topics are covered:

Time Stamp Computation

The RFC mandates to use the following format for time stamp computation.

The format of the timestamp is the same as in [RFC1305] and is as follows: the first 32 bits represent the unsigned integer number of seconds elapsed since 0h on 1 January 1900; the next 32 bits represent the fractional part of a second that has elapsed since then.

The following is a representation of the Timestamp:

```
0                             1                             2                             3
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9   0   1   2 3 4 5 6 7 8 9 0 1
```

Integer part of seconds

Fractional part of seconds
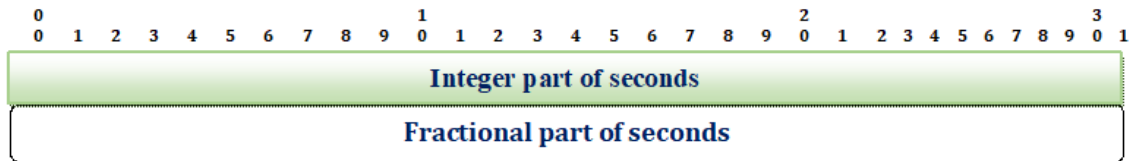
Error Estimate

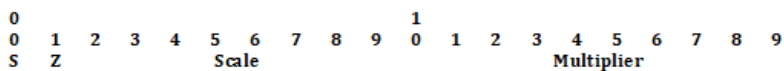The following is a description for a sample error estimate, which is defined in the RFC. It specifies the estimate of the error and the synchronization. It has the following format:

```
0                             1
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5   6   7   8   9
S   Z                   Scale                           Multiplier
```

The first bit, S, must be set if the party generating the timestamp has a clock that is synchronized to UTC using an external source. The following are the various aspects of this error estimation and the related tasks.

1. The bit must be set if the GPS hardware is used. It indicates an acquired it has current position and time or if NTP is used, it indicates that it has synchronized to an external source, which includes stratum 0 source, and so on.

   - If there is no notion of external synchronization for the time source, the bit must not be set.

   - The next bit has the same semantics as MBZ fields elsewhere: it must be set to zero by the sender and ignored by everyone else.

2. The next six bits, Scale, form an unsigned integer; Multiplier is an unsigned integer, which are interpreted as the following:

   - The error estimate is equal to Multiplier*2^(-32)*2^Scale (in seconds).

   - Notation clarification: 2^Scale is two to the power of Scale.

- Multiplier must not be set to zero.
- If the Multiplier is zero, the packet must be considered corrupt and discarded.

3. As the existing devices are not synchronized with any external time source by default, the bit 's' is set to the value 0.

4. The error estimation is calculated as the following:

```
before_jiffies = jiffies();

clock_get_time(); // Get linux time clock and compute time from 1900 Jan
1st

after_jiffies = jiffies();


 error_estimate_compute(after_jiffies-before_jiffies)

{

 Time_in_msec = jiffies_to_msecs(diff);

}
```

5. The mathematical formula outlined in RFC is used to compute the scale and multiplier values.

6. These values are static and are stored in `g_sto_data`.

Firewall Implicit Rule

If the device running TWAMP responder also supporting firewall functionality, TWAMP test packets may be dropped by default firewall configuration. To circumvent this, the device is programmed with firewall implicit rules to allow TWAMP test packets. TWAMP test packets are identified by destination port of the UPD packet that is available in TWAMP configuration.

The following table lists out events and action details for firewall implication.

Table 1 : Firewall Implicit Rule with Details for Events and Action

Sl. No.	Event	Action
1	Firewall is configured	If TWAMP configuration is set, configure implicit rule to allow the packets.
2	Firewall configurations removed	No action
3	TWAMP is configured	If firewall is configured, configure implicit rule to allow packets.
4	TWAMP configuration is updated	Delete the previous implicit rule and add new implicit rule to allow packets with new UDP port.
5	TWAMP configuration is deleted	Remove the implicit rule which allows the packets.

Limitations or Known Issues or Dependencies

The following is a list of known issues or limitations or dependencies for this feature.

- The UDP port number used for the twamp test packet reception must not be used by any other application in ACOS.

- Only TWAMP light version responder functionality is supported in this version.

- If the user changes the clock on the device, the first run of the test may show bad time stamp values. This is an expected behavior. The second run of the test must fix the problem as the user takes the note of the reading the clock time stamp again.

Routing Protocols

The following chapters are covered in this part/section:

Open Shortest Path First (OSPF)

The ACOS device supports the following OSPF versions:

- OSPFv2 for IPv4

- OSPFv3 for IPv6

This chapter provides configuration examples.

The following topics are covered:

For detailed CLI syntax information, see *Command Line Interface Reference*.

NOTE: It is recommended to set a fixed router-ID for all dynamic routing protocols you plan to use on the ACOS device, to prevent router-ID changes caused by VRRP-A failover.
Conditional routing is supported in OSPFv2 and OSPFv3. For more information on Conditional routing, see Conditional Routing using Route Maps.

Support for Multiple OSPFv2 and OSPFv3 Processes

The ACOS device supports multiple OSPFv2 processes on a single ACOS device. Only a single OSPFv2 process can run on a given interface.

Each IPv6 link can run up to 65535 OSPFv3 processes, on the same link.

Each OSPF process is completely independent of the other OSPF processes on the device. They do not share any information directly. However, you can configure redistribution of routes between them.

Support for OSPFv2 and OSPFv3 on the Same Interface or Link

You can configure OSPFv2 and OSPFv3 on the same interface or link. OSPFv2 configuration commands affect only the IPv4 routing domain, while OSPFv3 configuration commands affect only the IPv6 routing domain.

OSPF MIB Support

The following OSPF MIBs are supported:

- RFC 1850 – OSPFv2 Management Information Base
- draft-ietf-ospf-ospfv3-mib-08 – OSPFv3 Management Information Base

OSPF Configuration Example

The configuration excerpts in this example configure OSPFv2 and OSPFv3 on an ACOS device.

The following topics are covered:

Interface Configuration

The following commands configure two physical Ethernet data interfaces. Each interface is configured with an IPv4 address and an IPv6 address. Each interface also is added to OSPF area 0 (the backbone area).

The link-state metric (OSPF cost) of Ethernet 2 is set to 30, which is higher than the default, 10. Based on the cost difference, OSPF routes through Ethernet 1 will be favored over OSPF route through Ethernet 2, because the OSPF cost of Ethernet 1 is lower.

```
interface ethernet 1
 ip address 2.2.10.1 255.255.255.0
 ipv6 address 5f00:1:2:10::1/64
 ipv6 router ospf area 0 tag 1
!
interface ethernet 2
 ip address 3.3.3.1 255.255.255.0
 ipv6 address 5f00:1:2:20::1/64
 ip ospf cost 30
 ipv6 router ospf area 0 tag 1
```

The following commands configure two Virtual Ethernet (VE) interfaces. On VE 3, an IPv4 address is configured. On VE 4, an IPv4 address and an IPv6 address are configured.

OSPFv2 authentication is configured on VE 3, and the OSPF cost is set to 20.

On VE 4, the OSPF cost is set to 15.

```
interface ve 3
 ip address 1.1.1.2 255.255.255.0
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 abc
 ip ospf cost 20
!
interface ve 4
 ip address 1.1.60.2 255.255.255.0
 ipv6 address 5f00:1:1:60::2/64
 ip ospf cost 15
```

Global OSPF Parameters

The following commands configure global settings for OSPFv2 process 2. The router ID is set to 2.2.2.2. Subnets 1.1.x.x, 2.2.10.x, and 3.3.3.x are added to the backbone area. Redistribution is enabled for static routes, routes to VIPs, IP source NAT addresses, and floating IP addresses. In addition, an extra VRRP-A priority cost is configured, and the SPF timer is changed.

```
router ospf 2
 router-id 2.2.2.2
 ha-standby-extra-cost 25
 timers spf exp 500 50000
 redistribute static metric 5 metric-type 1
 redistribute vip only-flagged 500 metric-type 1
 redistribute ip-nat
 redistribute floating-ip metric-type 1
 network 1.1.0.0 0.0.255.255 area 0
 network 2.2.10.0 0.0.0.255 area 0
 network 3.3.3.0 0.0.0.255 area 0
```

The following commands configure global settings for OSPFv3 process 1. The router ID is set to 3.3.3.3. A stub area is added, redistribution is enabled, and the SPF timer is changed.

```
router ipv6 ospf 1
 router-id 3.3.3.3
 redistribute static metric 5 metric-type 1
 redistribute ip-nat
 redistribute floating-ip
 area 1 stub
 timers spf exp 500 50000
```

Clearing Specific OSPF Neighbors

The OSPF feature provides the option to clear all or specific OSPF neighbors.

You can clear neighbors by specifying various filters:

```
clear ip ospf [process-id]
{
process |
neighbor {all | neighbor-id | interface {interface-ip-address [neighbor-
ip-address]}}
}


clear ipv6 ospf [process-tag]
{
process |
neighbor {all | neighbor-id | interface-name [neighbor-id]}
}
```

The options listed in the syntax stand for following:

- `process-id` — Specifies the IPv4 OSPFv2 process to run on the device, and can be 1-65535.

- `process-tag` — Specifies the IPv6 OSPFv3 process to run on the IPv6 link, and can be 1-65535.

- `neighbor-id` — Specified the router-id of the OSPF device.

- `neighbor-ip-address` — Specifies the IP address of the interface for the neighboring device.

- `interface-ip-address` — Specifies the IP address of the interface of the device on which the OSPF neighbor exists.

Using OSPFv2, the CLI enables you to indicate an interface IP Address of the ACOS device. Using OSPFv3, the CLI enables you to specify the interface name for a specific neighbor.

Use the following commands to effect changes to clear OSPF neighbor information:

The following command clears all OSPF neighbors:

```
clear ip ospf [process-id] neighbor all
```

To clear all neighbors to a specific router:

```
clear ip ospf [process-id] neighbor neighbor-router-id
```

To clear all neighbors on an IPv4 interface:

```
clear ip ospf [process-id] neighbor interface interface-ip-address
```

To clear a neighbor on a specified interface to a specified router:

```
clear ip ospf [process-id] neighbor interface interface-ip-address
neighbor-router-id
```

To clear all IPv6 neighbors:

```
clear ipv6 ospf [process-tag] neighbor all
```

To clear all neighbors to a specific router:

```
clear ipv6 ospf [process-tag] neighbor neighbor-router-id
```

To clear all neighbors on a specified interface:

```
clear ipv6 ospf [process-tag] neighbor interface-name
```

To clear all neighbors on a specified interface to a specific router:

```
clear ipv6 ospf [process-tag] neighbor interface-name neighbor-router-id
```

Configuration Examples

The following command clears all OSPFv2 neighbors:

```
ACOS(config)#clear ip ospf neighbor all
```

The following command clears all neighbors to a specific router:

```
ACOS(config)#clear ip ospf neighbor 192.1.1.1
```

The following command clears all neighbors on an interface:

```
ACOS(config)#clear ip ospf neighbor interface 10.1.1.10
```

The following command clears a neighbor on a specified interface to a specified router:

```
ACOS(config)#clear ip ospf neighbor interface 10.1.1.10 192.1.1.10
```

The following command clears all OSPFv3 neighbors:

```
ACOS(config)#clear ipv6 ospf 5 neighbor all
```

The following command clears all neighbors to a specific router:

```
ACOS(config)#clear ipv6 ospf neighbor 192.1.1.1
```

The following command clears all OSPFv3 neighbors on a specified interface:

```
ACOS(config)#clear ipv6 ospf neighbor ethernet 1
```

The following command clears all neighbors on a specified interface to a specific router:

```
ACOS(config)#clear ipv6 ospf neighbor ethernet 1 192.1.1.1
```

OSPF Logging

The following topics are covered:

Overview

Router logging is disabled by default. You can enable router logging to one or more of the following destinations:

- CLI terminal (stdout)
- Local logging buffer
- Local file
- External log servers

NOTE: Log file settings are retained across reboots but debug settings are not. Enabling debug settings that produce lots of output, or enabling all debug settings, is not recommend for normal operation.

Configuring Router Logging for OSPF

To configure router logging for OSPF:

1. Enable output options.

2. Set severity level and facility.

3. Enable debug options to generate output.

NOTE:	For additional syntax information, including **show** and **clear** commands for router logging, see the *Command Line Interface Reference*.

Enabling Output Options

To enable output to the local logging buffer, use the following command at the global configuration level of the CLI:

```
router log log-buffer
```

To enable output to a local file, use the following command at the global configuration level of the CLI:

```
[no] router log file {name string | per-protocol | rotate num | size Mbytes}
```

To enable output to a remote log server, use the following command at the global configuration level of the CLI:

```
logging host ipaddr [ipaddr...] [port protocol-port]
```

Up to 10 remote logging servers are supported.

Setting Severity Level and Facility

The default severity level for router logging is 7 (debugging). The default facility is local0.

To change set the severity level for messages output to the terminal, use the following command at the global configuration level of the CLI:

```
logging monitor severity-level
```

The severity-level can be one of the following:

- **0** or **emergency**

- **1** or **alert**

- **2** or **critical**

- **3** or **error**

- **4** or **warning**

- **5** or **notification**

- **6** or **information**

- **7** or **debugging**

To change the severity level for messages output to the local logging buffer, use the following command at the global configuration level of the CLI:

```
logging buffered severity-level
```

To change the severity level for messages output to external log servers, use the following command at the global configuration level of the CLI:

```
logging syslog severity-level
```

To change the facility, use the following command at the global configuration level of the CLI:

```
logging facility facility-name
```

The facility-name can be one of the following:

- **local0**

- **local1**

- **local2**

- **local3**

- **local4**

- **local5**

- **local6**

- **local7**

Enabling Debug Options to Generate Output

To enable debugging for OSPF, use the following commands at the global configuration level or Privileged EXEC level of the CLI:

```
debug [ipv6] ospf type
```

The **ipv6** option enables debugging for OSPFv3. Without the **ipv6** option, debugging is enabled for OSPFv2.

The *type* specifies the types of OSPF information to log, and can be one or more of the following:

- **all** – Enables debugging for all information types listed below.
- **bfd** – Enables debugging for Bidirectional Forwarding Detection (BFD).
- **events** – Enables debugging for OSPF events.
- **ifsm** – Enables debugging for the OSPF Interface State Machine (IFSM).
- **lsa** – Enables debugging for OSPF Link State Advertisements (LSAs).
- **nfsm** – Enables debugging for the OSPF Neighbor State Machine (NFSM).
- **nsm** – Enables debugging for the Network Services Module (NSM). The NSM deals with use of ACLs, route maps, interfaces, and other network parameters.
- **packet** – Enables debugging for OSPF packets.
- **route** – Enables debugging for OSPF routes.

For each level, both **debug** commands are required.

CLI Example

The following commands configure OSPFv2 logging to a local file.

```
ACOS(config)#router log file name ospf-log
ACOS(config)#router log file per-protocol
ACOS(config)#router log file size 100
ACOS(config)#debug ospf all
ACOS(config)#debug ospf packet
```

These commands create a router log file named "ospf-log". The **per-protocol** option will log messages for each routing protocol separately. The log file will hold a maximum 100 MB of data, after which the messages will be saved in a backup and the log file will be cleared.

The following command displays the contents of the local router log file:

```
ACOS(config)#show router log file ospfd
2010/04/21 09:57:20 OSPF: IFSM[ve 3:1.1.1.2]: Hello timer expire
2010/04/21 09:57:20 OSPF: SEND[Hello]: To 224.0.0.5 via ve
3:1.1.1.2,
length
64
2010/04/21 09:57:20 OSPF:
-----------------------------------------------------
2010/04/21 09:57:20 OSPF: Header
2010/04/21 09:57:20 OSPF:   Version 2
2010/04/21 09:57:20 OSPF:   Type 1 (Hello)
2010/04/21 09:57:20 OSPF:   Packet Len 48
2010/04/21 09:57:20 OSPF:   Router ID 2.2.2.2
2010/04/21 09:57:20 OSPF:   Area ID 0.0.0.0
2010/04/21 09:57:20 OSPF:   Checksum 0x0
2010/04/21 09:57:20 OSPF:   Instance ID 0
2010/04/21 09:57:20 OSPF:   AuType 2
2010/04/21 09:57:20 OSPF:   Cryptographic Authentication
2010/04/21 09:57:20 OSPF:   Key ID 1
2010/04/21 09:57:20 OSPF:   Auth Data Len 16
2010/04/21 09:57:20 OSPF:   Sequence number 1271830931
2010/04/21 09:57:20 OSPF: Hello
2010/04/21 09:57:20 OSPF:   NetworkMask 255.255.255.0
2010/04/21 09:57:20 OSPF:   HelloInterval 10
2010/04/21 09:57:20 OSPF:   Options 0x2 (-|-|-|-|-|-|E|-)
2010/04/21 09:57:20 OSPF:   RtrPriority 1
2010/04/21 09:57:20 OSPF:   RtrDeadInterval 40
2010/04/21 09:57:20 OSPF:   DRouter 1.1.1.200
2010/04/21 09:57:20 OSPF:   BDRouter 1.1.1.2
2010/04/21 09:57:20 OSPF:   # Neighbors 1
2010/04/21 09:57:20 OSPF:     Neighbor 31.31.31.31
2010/04/21 09:57:20 OSPF:
-----------------------------------------------------
2010/04/21 09:57:21 OSPF: IFSM[ethernet 2:3.3.3.1]: Hello timer
```

```
expire
2010/04/21 09:57:21 OSPF: SEND[Hello]: To 224.0.0.5 via ethernet
2:3.3.3.1,
length 48
2010/04/21 09:57:21 OSPF:
------------------------------------------------------
2010/04/21 09:57:21 OSPF: Header
2010/04/21 09:57:21 OSPF:    Version 2
2010/04/21 09:57:21 OSPF:    Type 1 (Hello)
2010/04/21 09:57:21 OSPF:    Packet Len 48
2010/04/21 09:57:21 OSPF:    Router ID 2.2.2.2
2010/04/21 09:57:21 OSPF:    Area ID 0.0.0.0
2010/04/21 09:57:21 OSPF:    Checksum 0x49eb
2010/04/21 09:57:21 OSPF:    Instance ID 0
2010/04/21 09:57:21 OSPF:    AuType 0
2010/04/21 09:57:21 OSPF: Hello
2010/04/21 09:57:21 OSPF:    NetworkMask 255.255.255.0
2010/04/21 09:57:21 OSPF:    HelloInterval 10
2010/04/21 09:57:21 OSPF:    Options 0x2 (-|-|-|-|-|-|E|-)
2010/04/21 09:57:21 OSPF:    RtrPriority 1
2010/04/21 09:57:21 OSPF:    RtrDeadInterval 40
2010/04/21 09:57:21 OSPF:    DRouter 3.3.3.2
2010/04/21 09:57:21 OSPF:    BDRouter 3.3.3.1
2010/04/21 09:57:21 OSPF:    # Neighbors 1
2010/04/21 09:57:21 OSPF:      Neighbor 81.81.81.81
...
```

Intermediate System to Intermediate System (IS-IS)

This chapter describes how to integrate your ACOS device in an IS-IS network environment.

The following topics are covered:

This chapter provides IS-IS configuration examples.

For detailed CLI syntax information, see *Command Line Interface Reference*.

| NOTE:	It is recommended to set a fixed router-ID for all dynamic routing protocols you plan to use on the ACOS device, to prevent router-ID changes caused by VRRP-A failover.
	Conditional routing is supported in IS-ISv4/v6. For more information on Conditional routing, see Conditional Routing using Route Maps.

Basic IS-IS Example Topology

The example topology in the Figure 10 shows the ACOS device in a level-1 IS-IS topology.

Figure 10 : ACOS Device in a Basic IS-IS Topology
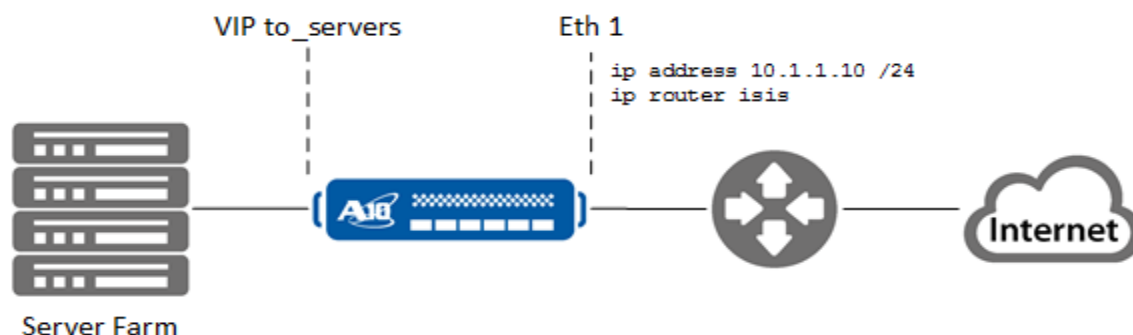
Configuring IS-IS

To configure IS-IS in the sample topology (this figure)ACOS Device in a Basic IS-IS Topology , first enable IS-IS in the ACOS device, enabling it to send Hello packets to other IS-IS devices in the same area:

```
ACOS(config)# router isis
ACOS(config-isis)# net 47.0000.0000.0000.0001.00
ACOS(config-isis)# is-type level-1
ACOS(config-isis)# redistribute vip only-flagged level-1
ACOS(config-isis)# exit
ACOS(config)#
```

The `router isis` command places you in IS-IS configuration mode. The `net` command configures the IS-IS instance on the ACOS device to be in the same area as the upstream router (in this case, 47.0000 as the area-id and 0000.0000.0001 as the system-id). The ACOS device must have the same area-id as the one configured on the router in order for it to bring up level-1 adjacencies.

The `is-type` command configures this instance as a level-1 instance; the same is accomplished by making sure the area-id in the `net` command matches the area-id on the router.

The `redistribute` command allows the VIP to the server farm to be advertised as a route in this IS-IS area.

NOTE:	If you are configuring IS-IS for IPv6, you should also add the `metric-style wide` command in your basic configuration.

Next, configure IS-IS on the individual interfaces. To configure IS-IS on an interface, use the `interface` command to access the configuration level for the interface, then use the `ip router isis | ipv6 router isis` commands. Below is an example to enable IS-IS for IPv4:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip address 10.1.1.10 /24
ACOS(config-if:ethernet:1)# ip router isis
```

To enable IS-IS for IPv6, use IPv6 commands. For example:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ipv6 address 2000::1/64
ACOS(config-if:ethernet:1)# ipv6 router isis
```

Verifying IS-IS Configuration

To view IS-IS settings, use the commands described in the Show Commands of the Network Configuration Chapter in the *Command Line Interface Reference*.

Border Gateway Protocol (BGP)

The ACOS device supports BGP4+ for both IPv4 and IPv6.

This chapter provides configuration examples. For more information on CLI syntax details, see *Command Line Interface Reference*.

NOTE: It is recommended to set a fixed router-ID for all dynamic routing protocols you plan to use on the ACOS device, to prevent router-ID changes caused by VRRP-A failover.

The following topics are covered:

BGP Route Redistributions

The routers in a BGP autonomous system (AS) advertise their routes to other BGP speakers (either internally or externally) through updates exchanged during peering sessions. These updates, or BGP route redistributions, can be used to distribute information about the topology and metrics for the neighboring routers.

The route redistributions can be for either static routes, which are manually-configured by an admin, or the route redistributions can be for dynamic routes that the router has acquired through the normal operation of the BGP protocol, such as routes learned through BGP peering sessions with other routers.

Using BGP Communities as Routing Policy Match Conditions

BGP community is a route attribute or a group of destinations that shares common property. It allows you tag the peers and control the redistribution scope of the routes within the network and to its peers. The BGP router can receive the communities from one of its peers and it can decide to take specific action on these routes through route-map or announce these routes with the communities to the neighbors in the network. The BGP router could also use the route maps to tag the routes within the community.

ACOS supports configuring the following BGP community:

Standard BGP Community

A BGP standard community consists of a set of 4-octet values, each of which specifies a community. All routes with this attribute belong to the communities listed in the attribute. For more information, see RFC 4384.

- The first 16 bits of the value encode the AS number of the network that originates the community.

- The last 16 bits carry a unique number assigned by the autonomous system (AS).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            <AS>               |            <Value>            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
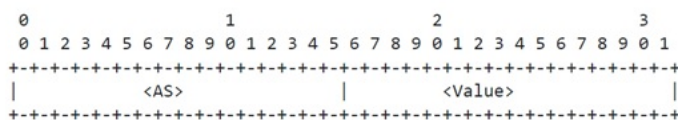
Extended BGP Community

A BGP extended community provides a larger range for grouping or categorizing communities. This attribute provides a mechanism for labelling information carried in BGP-4. For more information, see RFC 4360.

It consists of an 8-octet value (32-bit field) that is divided into two main sections. The first 2 octets of the community encode a 'type' field, while the last 6 octets carry a unique set of data in a format defined by the type field.

Large BGP Community

For more information, see RFC 8195 or RFC 8092.

A BGP large community is encoded as a 12-octet number (as shown below.)

An example of a Large Community is 123456:12345:1234567. Here,

- 123456 – Global administrator

Autonomous System Numbers (ASNs) can fit into the global administrator field.

- 12345 – Local data part 1
- 1234567 – Local data part 2

CLI Example

The following command configures the IP large community list. In first example, the community list "1" permits the BGP large community 4445:2345:2345. Similarly, in the second example, the standard large community list "list1" permits the large community 123456:12345:1234567.

```
ip large-community-list 1 permit 4445:2345:2345
!
ip large-community-list standard list1 permit 123456:12345:1234567
!
```

The following commands configure a route map called "b". The sequence number for this route-map is "10". The rule looks for route updates that have a local preference value of exactly "55". If a match occurs, then the action for this route map is to "permit" BGP updates to occur with this router.

```
ACOS(config)# route-map b permit 10
ACOS(config-route-map:10)# match large-community 1
ACOS(config-route-map:10)# set local-preference 55
```

The following commands configure a route map called "c". The rule looks for route updates that have a local preference value of exactly "77" and the large-community is "list1". If a match occurs, then the action for this route map is to "permit" BGP updates to occur with this router.

```
ACOS(config)# route-map c permit 15
ACOS(config-route-map:10)# match large-community list1
ACOS(config-route-map:10)# set local-preference 77
ACOS(config-route-map:10)# set large-community 123456:12345:1234567
```

The following commands apply the route map to an ACOS device that has BGP enabled. You could specify the AS that this ACOS device belongs to ("101"), the BGP neighbor (2.1.1.1 and 2.1.1.2), the name of the route map ("b" and "c"), and specify whether this route map is affecting inbound or outbound route updates ("in" or "out"), as shown in the sample commands below.

```
router bgp 101
 bgp router-id 1.1.1.1
 maximum-paths 2
 network 77.1.1.1/32 large-community 4445:2345:2345
 neighbor 2.1.1.1 remote-as 101
 neighbor 2.1.1.1 route-map b out
 neighbor 2002::2 remote-as 101
 redistribute connected
 address-family ipv6
  network 6006::2/128 large-community 123456:12345:1234567
  neighbor 2002::2 activate
  redistribute connected
!
router bgp 101
 bgp router-id 1.1.1.2
 network 99.1.1.1/32
 network 77.1.1.1/32 large-community 4445:2345:2345
```

```
neighbor 10.1.1.1 remote-as 101
neighbor 2.1.1.1 send-community large
neighbor 2.1.1.2 remote-as 101
neighbor 2.1.1.2 route-map c in
neighbor 2002::2 remote-as 101
redistribute static
address-family ipv6
 neighbor 2002::2 activate
 neighbor 2002::2 send-community large
!
```

Using Route Maps to Permit or Deny Updates

A BGP route map functions much like a filter. Route maps offer a way to permit or deny the exchange of information to neighboring BGP peers, and route maps can be used by network administrators to reduce the amount of information that is exchanged during BGP peering sessions.

Without route maps, every router on the Internet would share all of its information about every other router to which it is connected, and the sheer volume of traffic would bring the Internet to a grinding halt, so route maps offer a way to throttle the amount of information that is shared among BGP peers.[1]

Route maps are configured with one or more rules. Each rule consists of a set of match criteria and an associated action (permit or deny). The route map can have multiple rules, which are categorized in ascending order. Once the BGP route map is placed into action, it can be used to filter inbound or outbound routing traffic. If traffic is received and there is a positive match for the criteria in one of the rules, then the action associated with that match criteria will be applied. Assuming the associated action is to alter the local preference for routes from that peer, then ACOS will make this change before redistributing these route to other BGP peers.

---

[1]BGP route summarization, or route aggregation, offers another way to reduce the number of routes that are shared by consolidating blocks of IP addresses before redistribution. This prevents excessive fragmentation of blocks of IP addresses and gives ISPs more control over the blocks of IP addresses they own. Route aggregation also helps to conserve the limited number of IPv4 addresses.

Using Route Maps for Traffic Engineering

The rules in the route map are not just used to "permit" or "deny" peering sessions in the binary manner described above. Route maps can also be used for "traffic engineering". This is accomplished by modifying the information a BGP speaker receives from other BGP peers before the altered information is propagated via the route redistribution process. In other words, route maps can be configured to modify the properties of the routing information they receive before sending that modified data on its way.

For example, if you know that a neighboring autonomous system has old equipment that could impede or slow your network's traffic, it might be beneficial if you could administratively tell the equipment in your autonomous system to avoid that other network.

Route maps allow you to accomplish this goal by rewriting the properties or metrics associated with the paths to this other network.

You could set up one or more match criteria to identify traffic from this slower and older network, such that if a positive match occurs, ACOS would increase the cost (or decrease the weight) for the paths to this other network. Doing so would bias traffic away from these paths and encourage the use of other paths capable of circumventing the slow network.

In this way, ACOS does not simply refuse to accept the route redistributions received from BGP peers in the slower network. Instead of accepting the routing information received at face value, ACOS "tweaks" or rewrites the metrics associated with the paths to make them less attractive before passing them along to the surrounding BGP peers.

Route Selection Based on Local Preference

Route selection can use the local preference as a match criteria in a route map. While vetting route updates, if there is a positive match for the criteria, this triggers an action associated with the match criteria and helps determine whether BGP updates will be sent to one or more BGP peers.

A route map acts as a filter for the redistribution of BGP routes sent to peers. Rules are set up within the route map, consisting of match criteria (the metric upon which we are searching) and an associated action (for example, setting the local preference value). If a positive match is found then the action associated with that rule is applied.

For example, you could set a rule within a route map to look for updates from a particular BGP peer (based on IP address, router ID, or perhaps all routers in a particular Autonomous System Number), and you could then prevent ACOS from propagating, or redistributing, these updates to the other BGP peers in its ASN.

Instead of completely blocking routing updates from a nearby ASN, you could specify an action within the route map that would modify the various metrics to make the associated paths less preferred. For example, if you knew that a particular BGP peer is an older router that could hinder network performance, you could increase the cost of the paths to/from that router by increasing the cost of those paths by increasing the metric number. Similarly, you could achieve the same goal (of reducing the attractiveness of the paths associated with this older router and thus directing traffic away from it) by decreasing the weight for routes learned from this router.

CLI Example

The following commands configure a route map called "RED". The sequence number for this route-map is "10". The rule looks for route updates that have a local preference value of exactly 5000. If a match occurs, then the action for this route map is to "permit" BGP updates to occur with this router.

```
ACOS(config)# route-map RED permit 10
ACOS(config-route-map)# match local-preference 5000
```

At this point, you could apply the route map to an ACOS device that has BGP enabled. You could specify the AS that this ACOS device belongs to ("333"), the BGP neighbor (10.1.1.1), the name of the route map ("RED"), and specify whether this route map is affecting inbound or outbound route updates (in), as shown in the sample commands below.

```
router bgp 333
neighbor 10.1.1.1 remote-as 333
neighbor 10.1.1.1 route-map RED in
```

Conditional Routing using Route Maps

The routers in a BGP autonomous system (AS) advertise their routes to other BGP speakers (either internally or externally) through updates exchanged during peering sessions. The internal peers are updated with default routes even when the external peers are down. This results in unwanted traffic though there is no way to forward the traffic to the external peer.

The route-map match commands are extended to add new rules to match the existence / reachability of any IP prefixes / addresses (default or otherwise) in the Routing Information Base (RIB). Using these rules in the route-maps, the routing protocols may conditionally advertise routes to peers based on the RIB. As the RIB changes the routing protocols will advertise / withdraw the routes based on the route-map rules.

Conditional routing is also supported in IGP protocols such as OSPFv2, OSPFv3, IS-ISv4/v6.

You can configure the following conditions:

- **Exact**: This rule is True when there is an exact match of the prefix in the RIB.

CLI Example:

```
route-map abc permit 1
  match ip rib exact 1.2.3.4/32
  match ipv6 rib exact 1:2:3::4/128
!
```

- **Reachable**: This rule is True when the IP address is reachable based on the routes in the RIB. For example, the matching reachability for 11.21.31.41 is true, if the RIB has either a default route or a route such as 11.21.31.0/24.

CLI Example:

```
route-map abc permit 2
  match ip rib reachable 11.21.31.41
  match ipv6 rib reachable 11:21:31::41
!
```

- **Unreachable**: This rule is True when the IP address is not reachable based on the routes in the RIB.

CLI Example:

```
route-map abc permit 3
  match ip rib unreachable 12.22.32.42
  match ipv6 rib unreachable 12:22:32::42
```

Globally-Enabled Default Route Origination

When you are in router BGP mode, the `default-information originate` CLI command is available to advertise the default route.

Using the GUI to Configure Globally-Enabled Default Route Origination

BGP configuration is not supported in the GUI.

Using the CLI to Configure Globally-Enabled Default Route Origination

To configure a BGP routing process to distribute a default route, use the `default-information originate` command in the address family or router configuration mode. A valid default route must exist and be verified to complete this configuration or the default route will not be advertised:

```
ACOS(config)# router bgp 10
ACOS(config-bgp:10)# default-information originate
```

Equal-Cost Multi-path ECMP Support

Equal-cost multi-path (ECMP) support for BGP is available; by default, ECMP support is disabled. You can enable support for up to 64 equal-cost paths per route destination. Traffic to the destination prefix is then shared across all the installed paths.

Based on your configuration, BGP will install up to the maximum number of routes in the forwarding information base (FIB).

Use the `maximum-paths` command at the BGP configuration level to specify the maximum number of ECMP paths to a given route destination allowed for BGP: The default maximum-path value is 1. This value will not be displayed in the `show running-config` command. With the default setting (maximum-paths 1), BGP will install the single best ECMP route into the FIB used by the ACOS device to forward traffic.

NOTE:	For more information about enabling this feature at the global configuration level for all protocols, see the `maximum-paths` command in the *Command Line Interface Reference*.

The example below shows the BGP portion of an ACOS device configuration. The first set of output shows a device running IPv4 while the second set of output shows a device running IPv6. In the IPv4 output, the lines of output "neighbor 10.10.10.197 remote-as 197" through "neighbor 60.60.60.197 remote-as 197" show that the ACOS routing engine learned of this route from multiple neighbors.

```
ACOS(config)# router bgp 100
ACOS(config-bgp:100)# bgp router-id 100.100.100.100
ACOS(config-bgp:100)# maximum-paths 8
ACOS(config-bgp:100)# neighbor 10.10.10.197 remote-as 197
ACOS(config-bgp:100)# neighbor 20.20.20.197 remote-as 197
ACOS(config-bgp:100)# neighbor 30.30.30.197 remote-as 197
ACOS(config-bgp:100)# neighbor 40.40.40.197 remote-as 197
ACOS(config-bgp:100)# neighbor 50.50.50.197 remote-as 197
ACOS(config-bgp:100)# neighbor 60.60.60.197 remote-as 197
ACOS(config-bgp:100)# neighbor 3310::197 remote-as 197
ACOS(config-bgp:100)# neighbor 3320::197 remote-as 197
ACOS(config-bgp:100)# neighbor 3330::197 remote-as 197
ACOS(config-bgp:100)# neighbor 3340::197 remote-as 197
ACOS(config-bgp:100)# neighbor 3350::197 remote-as 197
ACOS(config-bgp:100)# neighbor 3360::197 remote-as 197
ACOS(config-bgp:100)# address-family ipv6
ACOS(config-bgp:100-ipv6)# maximum-paths 7
ACOS(config-bgp:100-ipv6)# neighbor 3310::197 activate
ACOS(config-bgp:100-ipv6)# neighbor 3320::197 activate
ACOS(config-bgp:100-ipv6)# neighbor 3330::197 activate
```

```
ACOS(config-bgp:100-ipv6)# neighbor 3340::197 activate
ACOS(config-bgp:100-ipv6)# neighbor 3350::197 activate
ACOS(config-bgp:100-ipv6)# neighbor 3360::197 activate
ACOS(config-bgp:100-ipv6)# exit-address-family
ACOS(config-bgp:100)#
```

The `show ip fib` command shows that the ACOS device's forwarding information base (FIB) was able to learn of 6 different routes to the same destination (7.7.7.0/ 24). Each route had an equal cost (distance = 20), and each route was learned through a different Ethernet port.

```
ACOS# show ip fib
Prefix               Next Hop         Interface      Distance
-----------------------------------------------------------------
7.7.7.0 /24          60.60.60.197     ethernet6      20
7.7.7.0 /24          50.50.50.197     ethernet5      20
7.7.7.0 /24          40.40.40.197     ethernet4      20
7.7.7.0 /24          30.30.30.197     ethernet3      20
7.7.7.0 /24          20.20.20.197     ethernet2      20
7.7.7.0 /24          10.10.10.197     ethernet1      20
```

The `show ip bgp` command displays paths learned through BGP. The ACOS device was connected to 6 different routes, and the Metric column shows that the cost is the same for all routes.

```
ACOS# show ip bgp
BGP table version is 14, local router  is 98.98.98.98
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal, l - labeled
              S Stale, m multipath
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 7.7.7.0/24       10.10.10.197           0             0 197 ?
*m                  20.20.20.197           0             0 197 ?
*m                  30.30.30.197           0             0 197 ?
*m                  40.40.40.197           0             0 197 ?
*m                  50.50.50.197           0             0 197 ?
*m                  60.60.60.197           0             0 197 ?
```

The `show ip route database` command displays essentially the same information as shown above. The ACOS device has a FIB that is populated with 6 different routes, of equal cost, to the same destination.

```
ACOS# show ip route database
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
 > - selected route, * - FIB route, p - stale info

B    *> 7.7.7.0/24 [20/0] via 10.10.10.197, ethernet 1, 00:13:38
*>            [20/0] via 20.20.20.197, ethernet 2, 00:13:38
*>            [20/0] via 30.30.30.197, ethernet 3, 00:13:38
 *>             [20/0] via 40.40.40.197, ethernet 4, 00:13:38
*>            [20/0] via 50.50.50.197, ethernet 5, 00:13:38
*>            [20/0] via 60.60.60.197, ethernet 6, 00:13:38
```

Route-Map High Availability for Interior Gateway Protocols

Feature History

ACOS supports a route-map option that performed matching based on the HA or VRRP-A VRID group, and also based on whether the device was the active or standby in the group. This option was used to control BGP route redistribution and advertisement decisions using the ACOS device's high availability state.

ACOS extended this feature to support all Interior Gateway Protocols (IGPs) such as OSPFv2, OSPFv3, ISISv4/6, RIP and RIPng.

This feature is now supported in ACOS 4.0.1 and beyond.

NOTE:	A route map could perform filtering based on metrics such as BGP community, IP address, or metric value. However, the filtering (or matching) could be performed based on the status of an ACOS device in a high availability configuration.
	High availability configuration is only available with VRRP-A beginning with ACOS 4.0 and beyond; the legacy HA configuration is no longer supported.

Route-Map High Availability Overview

This mechanism can be useful in certain network environments; for example, when a network uses VRRP-A for redundancy and the active ACOS device in the VRRP-A group will be upgraded. Such an upgrade requires the active ACOS device to change its status to standby, and the standby device must become active.

In this scenario, the ability to perform route map matching based on high availability status offers a unique way to use BGP (or other IGPs) route redistribution to advertise the paths to the newly-active ACOS device after switchover has occurred.

You can use the BGP protocol to modify some of the route settings by way of the route map. By changing the weights or local preference of certain routing paths, you can influence the routes that are advertised or withdrawn in route updates from the ACOS device to its BGP neighbors.

Alternatively, you can just wait for the old routes to time out, at which point they will be automatically withdrawn from the routing table of the neighboring routers. This will have the effect of directing network traffic to the newly-active ACOS device.

VRRP-A VRID Group Matching

Figure 11 shows a hypothetical network topology with two ACOS devices using VRRP-A for redundancy.

Here are a few other noteworthy points:

- The leftmost ACOS device is Active and the rightmost ACOS device is Standby.

- The diagram shows a Layer 3 router above the ACOS devices. The router is in autonomous system 200, and it is using BGP to share routing updates with the

ACOS load balancers. The ACOS devices are also running BGP and are located within AS 100.

- Static routes connect the ACOS devices to a Layer 3 router, which directs traffic to and from the real servers.

Figure 11 : Topology Using BGP Route Map (with VRRP-A High Availability Matching)
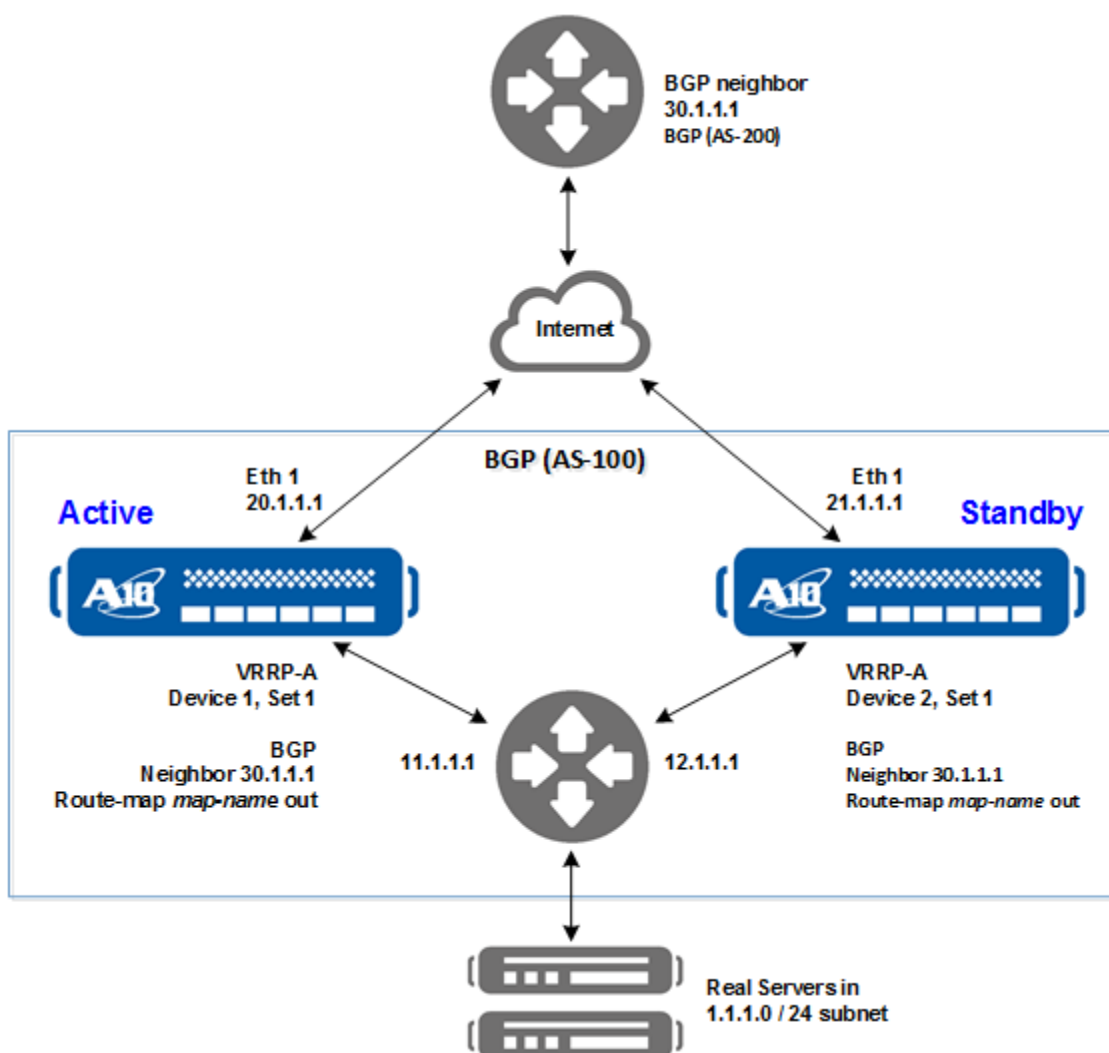
In a network environment like that shown above in Figure 11, the Active ACOS device must be relegated to "standby" mode before it can be upgraded. In turn, the Standby device must also be made "active". When this switchover occurs, it is imperative that the routers running BGP receive updated routing information. This updated routing information will cause the routes to the formerly-active ACOS device to be avoided,

and the routers must also be provided with new routing information about the paths traffic can use to reach the newly active ACOS device.

CLI Example

The following gives an example of a route map configuration. It is based on the network diagram shown in this figure, which has two ACOS devices using VRRP-A for redundancy. To upgrade one of the active ACOS devices, its status must be changed to standby (and the standby device must be made active). Then, the new routing information must be pushed to the router above, which is also running BGP.

Configurations on the Active ACOS device

The CLI commands below are used to configure VRRP-A on the first (Active) ACOS device.

```
vrrp-a common
  device-id 1
  set-id 1
  enable
```

The following CLI commands assign an IP address of 20.1.1.1 to Ethernet interface 1 on the ACOS device.

```
interface eth 1
  ip address 20.1.1.1
```

The following CLI commands are used to create a route map called "test1" with a sequence number of 10. A rule is added that checks for a positive match for the active ACOS device in the VRRP-A group 1. If a positive match is found, then this ACOS device can share its route redistributions with any BGP peers that pass the match criteria.

```
route-map test1 permit 10
  match group 1 active
```

The following CLI commands are used at the global configuration level to enable the BGP protocol and specify the Autonomous System (AS) number of "100" for the Active ACOS device. The BGP peer is specified in remote AS 200, and the hop count needed to reach this external BGP router is not to exceed 255 hops. The outbound redistribution of static routes would be allowed to the BGP peer at 30.1.1.1, based upon the match criteria (and associated actions) in the route-map called "test1".

```
router bgp 100
  redistribute static
  neighbor 30.1.1.1 remote-as 200
  neighbor 30.1.1.1 ebgp-multihop 255
  neighbor 30.1.1.1 route-map test1 out
```

The following CLI commands are used to configure a static route from the Active ACOS device to the real servers in the subnet 1.1.1.0 /24, by way of the next-hop router at IP 11.1.1.1.

```
ip route 1.1.1.0 /24 11.1.1.1
```

Configurations on the Standby ACOS device

The command below configure VRRP-A on the Standby ACOS device.

```
vrrp-a common
  device-id 2
  set-id 1
  enable
```

The following CLI commands assign the IP 21.1.1.1 to Ethernet interface 1 on the Standby ACOS device.

```
interface eth 1
  ip address 21.1.1.1
```

The CLI commands below create a route map called "test1" with a sequence number of 10. A rule is added to check for a match for the active ACOS device in the HA (or VRRP-A) group 1. If a positive match is found, then this ACOS device may share its route redistributions with its BGP peers.

```
route-map test1 permit 10
  match group 1 active
```

The following CLI commands are used at the global configuration level to enable the BGP protocol and specify an Autonomous System (AS) number of "100" for the Standby ACOS device. The BGP peer is specified in remote AS 200, and the hop count needed to reach this external BGP router is not to exceed 255 hops. The outbound redistribution of static routes could be sent to the BGP peer at 30.1.1.1, based upon the match criteria (and the associated actions) in route-map "test1".

```
router bgp 100
  redistribute static
```

```
neighbor 30.1.1.1 remote-as 200
neighbor 30.1.1.1 ebgp-multihop 255
neighbor 30.1.1.1 route-map test1 out
```

The following CLI commands are used to configure a static route from the Standby ACOS device to the real servers in the subnet 1.1.1.0 /24, by way of the next-hop router at IP 12.1.1.1.

```
ip route 1.1.1.0 /24 12.1.1.1
```

NOTE: In the above configuration, only an Active ACOS device can redistribute its static routes. The Standby ACOS device does not redistribute its static routes. The reason for this is that the match criteria "permits" the Active device in an HA (or VRRP-A) pair to send out (redistribute) its routes. There is no rule in the route map with an explicit "deny" action, but the deny is implicit, because any Standby HA devices would fail to match the criteria in the route map, so the Standby HA device would fail to match the criteria and its routing updates would not be shared.

Unnumbered Interfaces Support in BGP

ACOS supports enabling unnumbered IPv4 interfaces in BGP. As per RFC 5549, ACOS BGP supports exchanging IPv4 routes over an unnumbered IPv4 interface. This is achieved by BGP neighbor communication over IPv6 Link Local addresses and requires enabling IPv6 on the interface. ACOS uses the Router Advertisement (RA) link-level protocol to automatically discover the neighboring router's IPv6 Link Local address. There will only be a single BGP neighborship supported per IPv4 unnumbered interface.

The following command under 'router bgp' enables BGP unnumbered support on an unnumbered IPv4 interface:

```
neighbor ve 104 unnumbered
```

The following command enables IPv4 prefix with IPv6 next hop on IPv6 BGP or TCP session:

```
neighbor 3101::197 capability extended-nexthop
```

When an ACOS device has unnumbered interfaces, the following configuration is required to set the Source IP address of services such as Health Monitoring and any other management services that require IPv4 communication:

```
ip unnumbered
  use-source-ip  12.12.12.12
```

NOTE: The IP address must match an existing interface address (typically, a Loopback interface).

When an ACOS device has IPv6 interfaces with only IPv6 Link Local addresses, the following configuration is required to set the Source IPv6 address for management services that require IPv6 communication more than one hop away:

```
ipv6 unnumbered
 use-source-ipv6 2001:db8:8:8::37
```

NOTE: The IPv6 address must match an existing interface's IPv6 address (typically, a Loopback interface).

ACOS allows users to configure an Access List (ACL) to control the source IPv6 addresses for the control packets. IPv6 ACL's `permit` action allows the source IPv6 address to pass through without any NAT action. When the source IPv6 address is not permitted, ACOS will NAT using the source IPv6 address specified under `use-source-ipv6`.

The following command under 'ipv6-unnumbered' is used to configure access-list to permit or deny (NAT) host path traffic with given source IPv6 addresses.

```
ACOS (config) # ipv6 unnumbered
ACOS (config-unnumbered) # use-source-acl?
NAME<length: 1-16> ACL Name
```

Consider the following:

- If the `use_source_ipv6` is not configured, then the ACL is bypassed, and the ACOS will not perform NAT.

- Selection of Source IPv6 address and addresses in ACL must be consistent with other system or network configuration.

- If ACL is not configured, then host traffic will be processed normally as before (NATed).

- If ACL exists but empty, then the default action will be `deny` (NAT).

- Floating IP addresses (like VIP, NAT Pool, etc) that are shared among multiple systems, cannot be used for control plane traffic such as Health Monitoring traffic.

Configuring BGP Unnumbered Interface

The following commands configure BGP Unnumbered support:

Device1:

```
ACOS1(config)#interface eth 1
ACOS1(config-if:ethernet:1)#ipv6 enable
ACOS1(config-if:ethernet:1)#ipv6 ndisc router-advertisement enable
ACOS1(config)#router bgp 97
ACOS1(config-bgp:97)#bgp router-id 97.97.97.97
ACOS1(config-bgp:97)#neighbor eth 1 unnumbered
```

Device 2:

```
ACOS2(config)#interface eth 1
ACOS2(config-if:ethernet:1)#ipv6 enable
ACOS2(config-if:ethernet:1)#ipv6 ndisc router-advertisement enable
ACOS2(config)#router bgp 98
ACOS2(config-bgp:98)#bgp router-id 98.98.98.98
ACOS2(config-bgp:98)#neighbor eth 1 unnumbered
```

BGP Unnumbered Support Limitations

Consider the following limitations:

- BFD authentication is not supported.

- L3v is not supported.

- To simplify CLI, use BGP peer-groups to add peer level settings such as route-maps, neighbor timers, BFD, and so on.
  In the example below, BGP peer-group PG1 has additional settings for IPv4 and

IPv6 address families including route-map and password. The BGP peer-group PG1 may be applied to the unnumbered BGP peer on interface ethernet 1.

```
router bgp 100
neighbor PG1 peer-group
 neighbor PG1 password a10
 neighbor PG1 route-map rmapv4in in
 neighbor PG1 route-map rmapv4out out
 neighbor ethernet 1 unnumbered
neighbor ethernet 1 peer-group PG1   // this applies to peer-group's
IPv4 address family settings
address-family ipv6
    neighbor PG1 route-map rmapv6in in
    neighbor PG1 route-map rmapv6out out
    neighbor ethernet 1 peer-group PG1  // this applies to peer-group's
IPv6 address family settings
```

Displaying BGP Unnumbered Information

The following show commands are supported:

- Use the following command to display the brief interface information. The "Flags" column indicates "U" if the BGP protocol is configured with an unnumbered interface and the interface is operational.

NOTE: IP unnumbered flag (U) is associated with the IPv4 address configuration.

```
ACOS(config)# show interfaces brief
ACOS#show interfaces brief
Port Link Dupl Speed Trunk Vlan MAC
                                        IPs Flags Name
-----------------------------------------------------------------------
---------
mgmt Up   Full 1000  N/A   N/A  001f.a008.0fb0 10.65.19.117/24 1
1    Up   Full 10000 none  1     001f.a008.0fb2 8.8.8.8/24      1
2    Disb None None  none  1     001f.a008.0fb3 0.0.0.0/0       0
3    Disb None None  none  1     001f.a008.0fb  0.0.0.0/0       0
4    Disb None None  none  1     001f.a008.0fb5 0.0.0.0/0       0
5    Disb None None  none  1     001f.a008.0fb6 0.0.0.0/0       0
```

```
6     Disb None None   none  1     001f.a008.0fb7 0.0.0.0/0       0
7     Disb None None   none  1     001f.a008.0fb8 0.0.0.0/0       0
8     Disb None None   none  1     001f.a008.0fb9 0.0.0.0/0       0
9     Up   Full 10000  none  1     001f.a008.0fba 0.0.0.0/0       0U
10    Disb None None   none  1     001f.a008.0fbb 0.0.0.0/0       0
11    Disb None None   none  1     001f.a008.0fbc 0.0.0.0/0       0
12    Disb None None   none  1     001f.a008.0fbd 0.0.0.0/0       0
13    Up   Full 10000  6     1     001f.a008.0fbe 0.0.0.0/0       0
14    Up   Full 10000  6     1     001f.a008.0fbf 0.0.0.0/0       0
15    Disb None None   none  1     001f.a008.0fc0 0.0.0.0/0       0
16    Down None None   none  1     001f.a008.0fc1 0.0.0.0/0       0
17    Disb None None   none  1     001f.a008.0fc2 0.0.0.0/0       0
18    Disb None None   none  1     001f.a008.0fc3 0.0.0.0/0       0
19    Disb None None   none  1     001f.a008.0fc4 0.0.0.0/0       0
20    Disb None None   none  1     001f.a008.0fc5 0.0.0.0/0       0
lo1   Up   N/A  N/A    N/A   N/A   N/A            4.4.4.4/32      1
lo2   Up   N/A  N/A    N/A   N/A   N/A            5.5.5.1/24      1
```

- Use the following command to view the information for interface, if the unnumbered is configured and running or operational on that interface:

```
ACOS(config)# show interface ethernet/ve/trunk <num>


Ethernet 9 is up, line protocol is up
Hardware is 10Gig, Address is 001f.a008.0fba
Unnumbered is configured and active, peer is fe80::21f:a0ff:fe07:635a
Internet address is 0.0.0.0, Subnet mask is 0.0.0.0
IPv6 link-local address is fe80::21f:a0ff:fe08:fba Prefix 64 Type:Link-
Local
Configured Speed auto, Actual 10Gbit, Configured Duplex auto, Actual fdx
Member of L2 Vlan 1, Port is Untagged
Flow Control is disabled, IP MTU is 1500 bytes
Port as Mirror disabled, Monitoring this Port disabled
Interface name is IBGP-to-SLBAX1-L3v
7675141 packets input 950488101 bytes
Received 0 broadcasts, Received 9867 multicasts, Received 7665274 unicasts
0 input errors 0 CRC 0 frame
0 runts 0 input giants
9968713 packets output 922172489 bytes
Transmitted 0 broadcasts, Transmitted 9895 multicasts, Transmitted 9958818
unicasts
```

```
0 output errors 0 output giants 0 collisions
300 second input rate: 11424 bits/sec, 11 packets/sec, 0% utilization
300 second output rate: 16440 bits/sec, 20 packets/sec, 0% utilization
```

Advertising IPv4 Routes Using a Global IPv6 Next Hop Over an IPv6 BGP Connection

ACOS supports advertising IPv4 routes with an IPv6 BGP connection using Global IPv6 next hops.

The following commands advertise IPv4 prefixes over IPv6 next hop over an IPv6 BGP peering:

BGP Configuration:

```
ACOS(config)#router bgp 97
ACOS(config-bgp:97)#network 1.1.1.3/32
ACOS(config-bgp:97)#neighbor 4f90::3 remote-as 98
ACOS(config-bgp:97)#neighbor 4f90::3 capability extended-nexthop
```

If you do not want to configure an IPv4 address at the BGP endpoints, then you can configure ip unnumbered on both endpoint interfaces of the BGP peers.

The following configuration is required to set the Source IP address of services such as Health Monitoring and any other management services that require IPv4 communication:

```
ip unnumbered
 use-source-ip  12.12.12.12
```

NOTE: The IP address must match an existing interface address (typically, a Loopback interface).

The following command under 'interface' enables IPv4 unnumbered support:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip unnumbered
```

Configuring IPv6 Global BGP Peer With IP Unnumbered

The following commands configure IPv6 Global BGP peering to advertise IPv4 prefix with IP unnumbered support:

```
ACOS(config)#interface eth 1
ACOS(config-if:ethernet:1)#enable
ACOS(config-if:ethernet:1)#ip unnumbered
ACOS(config-if:ethernet:1)#ipv6 address 4f90::2
ACOS(config-if:ethernet:1)#ipv6 enable
ACOS(config-if:ethernet:1)#ipv6 ndisc router-advertisement enable
ACOS(config-if:ethernet:1)#ipv6 ndisc router-advertisement max-interval 10
ACOS(config-if:ethernet:1)#ipv6 ndisc router-advertisement min-interval 3
ACOS(config-if:ethernet:1)#exit
ACOS(config)#router bgp 97
ACOS(config-bgp:97)#network 1.1.1.3/32
ACOS(config-bgp:97)#neighbor 4f90::3 remote-as 98
ACOS(config-bgp:97)#neighbor 4f90::3 capability extended-nexthop
```

BGP Extended Next Hop Capability Limitation

Consider the following limitation:

The extended-nexthop capability cannot be configured for the peer group. The individual BGP neighbor must be configured with the extended-nexthop capability.

Displaying BGP Information

The following show commands are supported:

- Use the following command to view BGP IPv4 routes that are learned from the IPv6 global peers:

```
ACOS(config)# show ip bgp
BGP Address Family IPv4 Unicast
BGP table version is 569, local router ID is 12.12.12.12
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              l - labeled
S Stale, m multipath
Origin codes: i - IGP, e - EGP, ? - incomplete
Network       Next Hop                          Metric LocPrf Weight
Type Path
*> 1.1.1.3/32 4f90::3(fe80::20c:29ff:fe99:3b19) ethernet 1    0      0
 98 ?
```

- Use the following command to view the information for the interface, if the unnumbered is configured and running or operational on that interface:

```
ACOS(config)# show interface ethernet/ve/trunk <num>
Ethernet 1 is up, line protocol is up
 Hardware is 10Gig, Address is 000c.29a2.f5a3
 Unnumbered is configured and active, peer is fe80::20c:29ff:fe99:3b19
 Internet address is 0.0.0.0, Subnet mask is 0.0.0.0
 IPv6 address is 4f90::2 Prefix 64 Type: unicast
 IPv6 link-local address is fe80::20c:29ff:fea2:f5a3 Prefix 64 Type:
Link-Local
 Configured Speed auto, Actual 10Gbit, Configured Duplex auto, Actual
fdx
 Member of L2 Vlan 1, Port is Untagged
 Flow Control is disabled, IP MTU is 1500 bytes
 Port as Mirror disabled, Monitoring this Port disabled
 Interface name is Towards_Core_network
 Last Up Change:    0 Day, 8 Hour, 8 Min,34 Sec ago
 Last Port Counters Cleared: Never
 417021 packets input  53493180 bytes
 Received  0 broadcasts, Received 305649 multicasts, Received 111372
unicasts
 0 input errors  0 CRC  0 frame
 0 runts  0 input giants
 119920 packets output  10613695 bytes
 Transmitted  0 broadcasts, Transmitted 7928 multicasts, Transmitted
111992 unicasts
 0 output errors  0 output giants 0 collisions
```

```
300 second input rate: 14584 bits/sec, 13 packets/sec, 0% utilization
300 second output rate: 2896 bits/sec, 3 packets/sec, 0% utilization
```

BGP IPv6 Next Hop Address Preference

By default, when a BGP neighbor advertises a route with both a global IPv6 address and an IPv6 link-local address in the Next Hop field, ACOS installs the link-local address as the forwarding next hop in the Routing Information Base (RIB) and Forwarding Information Base (FIB).

ACOS provides an option to prefer the global IPv6 next hop address over an IPv6 link-local address in BGP. you can enable this behavior by using the `nexthop prefer-global` command. When enabled, ACOS selects the global IPv6 address as the next hop for forwarding.

Consider the following:

- When the `nexthop prefer-global` command is configured or removed, ACOS reinstalls IPv6 routes with the updated next hops in the RIB and FIB.

- Existing BGP sessions are not affected by configuring or removing the `nexthop prefer-global` command.

Configuring Global IPv6 Next Hop Address

The following commands configure global IPv6 as the preferred next hop address:

```
ACOS(config)#router bgp 100
ACOS(config-bgp:100)#bgp router-id 100.100.100.100
ACOS(config-bgp:100)#neighbor 3340::197 remote-as 197
ACOS(config-bgp:100)#address-family ipv6
ACOS(config-bgp:100-ipv6)#nexthop prefer-global
ACOS(config-bgp:100-ipv6)#neighbor 3340::197 activate
```

For more information, see the `nexthop` command in the *Command Line Interface Reference*.

Limitation

This feature is applicable only to the IPv6 address family.

Class E Addressing and Advertising

ACOS supports Class E address configuration and assigns IP addresses from the 240.0.0.0/4 range to interfaces, such as Management interfaces, Loopback interfaces, and Virtual Ethernet interfaces.

Class E IP addresses belong to the range 240.0.0.0/4. As per RFC 1700 and RFC 3330, the Internet Engineering Task Force (IETF) has reserved them for experimental purposes and future use. Recently, Class E addresses have been used in private datacenter deployment and public cloud environments.

The `system ip class-e-address-range-enable` command enables assigning the Class E addresses to network interfaces, advertising through BGP, and installing routes into the Routing Information Base (RIB) and Forwarding Information Base (FIB).

Additionally, routing protocols such as OSPF, IS-IS, BFD, and static routing support Class E addresses.

Configuring Class E Address

The following system command enables configuring Class E addresses:

```
ACOS(config)#system ip class-e-address-range-enable
```

NOTE: For changes to take effect, the system must be reloaded after configuring or deconfiguring the command.

The following commands configure Class E IP on the data interface after enabling the `system ip class-e-address-range-enable` command:

```
ACOS(config)#interface ethernet 3
ACOS(config-if:ethernet:3)#enable
ACOS(config-if:ethernet:3)#ip address 249.100.50.69 255.0.0.0
```

```
ACOS(config)#enable-management service ssh
ACOS(config-enable-management ssh)#ethernet 3
```

The following commands configure the BGP with BFD after enabling the `system ip class-e-address-range-enable` command:

```
Device 1
ACOS(config)#bfd enable
ACOS(config)#router bgp 100
ACOS(config-bgp:100)# neighbor 249.254.32.92 remote-as 100
ACOS(config-bgp:100)# neighbor 249.254.32.92 fall-over bfd

Device 2
ACOS(config)#bfd enable
ACOS(config)#router bgp 100
ACOS(config-bgp:100)# neighbor 249.254.32.126 remote-as 100
ACOS(config-bgp:100)# neighbor 249.254.32.126 fall-over bfd
ACOS(config-bgp:100)# redistribute connected
```

IPv4 Unnumbered for IP Tunnels

ACOS allows you to configure Logical Interfaces (LIFs) as unnumbered interfaces. The packets can originate from or terminate to the unnumbered interface. Logical Interfaces can be configured in shared or L3V partitions.

Any packet originated or destined to unnumbered interface will use the following IP address as the source IP address:

```
ip unnumbered
  use-source-ip 12.12.12.12
```

The following command is used to configure ip unnumbered on the logical interface.

```
ACOS (config) # int lif 123
ACOS (config-if:lif:123) # ip unnumbered
```

Configuring IP Unnumbered Interface

The following commands configure IP Unnumbered support:

```
ACOS(config)#ip unnumbered
```

```
ACOS(config-unnumbered)#use-source-ip 1.2.3.4
ACOS(config-unnumbered)#interface loopback 1
ACOS(config-if:loopback:1)#ip address 1.2.3.4 /32
ACOS(config-if:loopback:1)#interface lif 123
ACOS(config-if:lif:123)#ip unnumbered
ACOS(config-if:lif:123)#
```

IP Unnumbered Support Limitations

Consider the following limitations:

- There can be a maximum of 200 unnumbered links (IP unnumbered and BGP unnumbered) system-wide (Shared + L3V).

- IP unnumbered and BGP unnumbered can exist together but not on the same interface.

- Only IP and GRE encapsulation on the IP unnumbered interfaces are supported.

- Only BGP and static routing on the IP unnumbered interface are supported.

Displaying IP Unnumbered Information

The following show commands are supported:

- Use the following command to display the brief interface information. The "Flags" column indicates "U" if the Logical Interface (LIF) is configured as an unnumbered interface.

```
ACOS#show interfaces brief
Port    Link  Dupl  Speed  Trunk Vlan Encap  MAC              IP Address
     IPs Flags  Name
-------------------------------------------------------------------------
------------------
mgmt    Up    auto  auto   N/A   N/A  N/A    728a.41ab.c4b7
10.64.19.105/24  1
1       Up    Full  10000  none  Tag  N/A    7283.3fa5.cf02  0.0.0.0/0
     0
2       Up    Full  10000  none  Tag  N/A    625d.d6f7.206b  0.0.0.0/0
     0
```

```
3       Disb  None  None   none  1    N/A     5a1f.6334.0f6e
57.1.1.20/24     1
4       Up    Full  10000  none  1    N/A     8a52.7e33.a4ea
56.1.1.13/24     1
lif 124 Up    N/A   N/A    N/A   1    N/A     625d.d6f7.206b  0.0.0.0/0
     0   U
lif_1   Up    N/A   N/A    N/A   1    N/A     5a1f.6334.0f6e  0.0.0.0/0
     0   U
ve2     Up    N/A   N/A    N/A   2    N/A     7283.3fa5.cf02  2.2.2.1/24
     1
ve102   Up    N/A   N/A    N/A   102  N/A     625d.d6f7.206b
102.102.102.1/24 1
Global Throughput:0 bits/sec (0 bytes/sec)
Throughput:0 bits/sec (0 bytes/sec)
```

- Use the following command to view the information for the logical interface:

```
ACOS(config-if:lif:124)#show int lif 124
lif 124 is up, line protocol is up
Logical interface Lif, Address is 625d.d6f7.206b
IP Unnumbered is configured and active
IP Encap, Local endpoint 165.165.165.2, Remote endpoint 165.165.165.1
IP MTU is 1500 bytes
0 packets input  0 bytes
Received  0 broadcasts, Received 0 multicasts, Received 0 unicasts
0 packets output  0 bytes
Transmitted  0 broadcasts, Transmitted 0 multicasts, Transmitted 0
unicasts
```

Bidirectional Forwarding Detection (BFD)

The following topics are covered:

Overview

Bidirectional Forwarding Detection (BFD) provides very fast failure detection for routing protocols. When BFD is enabled, the ACOS device periodically sends BFD control packets to the neighboring devices that are also running BFD. If a neighbor stops sending BFD control packets, the ACOS device quickly brings down the BFD session(s) with the neighbor, and recalculates paths for routes affected by the down neighbor.

BFD provides a faster failure detection mechanism than the timeout values used by routing protocols. Routing protocol timers are multiple seconds long, whereas BFD provides sub-second failover.

The A10 implementation of BFD is based on the following RFCs:

- RFC 5880, Bidirectional Forwarding Detection (BFD)

- RFC 5881, Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)

- RFC 5882, Generic Application of Bidirectional Forwarding Detection (BFD)

- RFC 5883, Bidirectional Forwarding Detection (BFD) for Multihop Paths

BFD may also be initiated with a VRRP-A floating IPv4 or IPv6 address as a source address. When BFD is running with BGP and both BGP and BFD are configured to use VRRP-A floating IPv4 or IPv6 addresses as the source address, and a VRRP-A switchover event occurs, the newly active device with the activated floating IPv4 or IPv6 address will reinitiate BGP and BFD sessions with the router causing a BGP and BFD flap during the failover. When the BFD is running with a floating IPv4 or IPv6 address that is not in an interface subnet, it will require a multi-hop setting.

Support in this Release

The current release has the following BFD support:

- Basic BFD protocol (packet processing, state machine, and so on)

- BGP client support

- Multihop

- BFD Asynchronous mode

- OSPFv2/v3 client support

- Static route support

- IS-IS client support

- BFD Demand mode

- Full Echo function support

- Authentication

BFD Parameters

The following topics are covered:

BFD is disabled by default. You can enable it on a global basis.

BFD Echo

BFD echo enables a device to test data path to the neighbor and back. When a device generates a BFD echo packet, the packet uses the routing link to the neighbor device to reach the device. The neighbor device is expected to send the packet back over the same link.

NOTE:

The BFD echo packets will be dropped if any of the following commands is configured:

- `ip anomaly-drop drop-all`

- `ip anomaly-drop security-attack layer-3`

- `ip anomaly-drop land-attack`

BFD Timers

You can configure BFD timers at the following configuration levels:

- Global

- Interface

If you configure the timers on an individual interface, the interface's settings are used instead of the global settings. Likewise, if the BFD timers are not set on an interface, that interface uses the global settings. For BGP loopback neighbors, BFD always uses the global timer.

The DesiredMinTXInterval, RequiredMinRxInterval and DetectMult timer fields can be configured at the interface and the global configuration level. However, the actual timer will vary depending on the Finite State Machine (FSM) state, through negotiation, and whether or not echo has been enabled.

BGP Support

If you run BGP on the ACOS device, you can enable BFD-based fallover for individual BGP neighbors.

Configuring BFD

Static Route Support

A static route flap can occur when you enable BFD in global mode or when you configure a static BFD session.

In the following example, you will see that the static routes experience a flap when BFD is enabled. The fields to note are flagged in bold:

```
ACOS(config)# show ipv6 route
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       i - IS-IS, B - BGP
Timers: Uptime
```

```
C    3ffe:100::/64 via ::, ve 10, 00:01:28
C    3ffe:1111::/64 via ::, loopback 1, 00:01:30
S    3ffe:2222::/64 [1/0] via 3ffe:100::20, ve 10, 00:00:25  <===value
before flap
timer
C    3ffe:3333::/64 via ::, loopback 2, 00:01:30
ACOS(config)#bfd enable<===enable BFD
ACOS(config)# show ipv6 route
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       i - IS-IS, B - BGP
Timers: Uptime


C    3ffe:100::/64 via ::, ve 10, 00:01:32
C    3ffe:1111::/64 via ::, loopback 1, 00:01:34
S    3ffe:2222::/64 [1/0] via 3ffe:100::20, ve 10, 00:00:01   <==value
after flap
C    3ffe:3333::/64 via ::, loopback 2, 00:01:34
ACOS(config)#
```

To enable BFD, use the following command at the global configuration level of the CLI:

```
ACOS(config)#bfd enable
```

To enable BFD echo, use the following command at the global configuration level of the CLI:

```
ACOS(config)#bfd echo
```

To configure BFD timers, use the following commands. These commands are available at the global configuration level and at the configuration level for individual interfaces.

```
[no] bfd interval ms min-rx ms multiplier num
```

The **interval** value can be 48-1000 ms, and is 800 ms by default. The **min-rx** value can be 48-1000 ms, and is 800 ms by default. The **multiplier** value can be 3-50 and is 4 by default.

Configuring BFD Parameters for BGP

To enable BFD-based fallover for a BGP neighbor, use the following command at the BGP configuration level:

```
[no] neighbor ipaddr fall-over bfd [multihop]
```

To display BFD information for BGP neighbors, use the following command:

```
show ip bgp neighbor
```

Displaying BFD Information

To display summarized BFD neighbor information, use the following command:

```
show bfd neighbors
```

To display detailed BFD neighbor information, use the following command:

```
show bfd neighbors detail
```

To display BFD statistics, use the following command:

```
show bfd statistics
```

To clear BFD statistics, use the following command:

```
clear bfd statistics
```

Disabling BFD

To disable BFD, enter the following command in global configuration mode:

```
ACOS(config)# no bfd enable
```

Enter the command to stop processing all BFD packets.

Configuring BFD with OSPF (for IPv4)

To enable BFD with OSPF on an interface, enter one of the following sets of commands:

To enable BFD on an individual interface:

```
ACOS(config)# interface ethernet 1
```

```
ACOS(config-if:ethernet:1)# ip address 20.0.0.1 255.255.255.0
ACOS(config-if:ethernet:1)# ip ospf bfd
```

To enable BFD on a virtual interface:

```
ACOS(config)# interface ve 100
ACOS(config-if:ve:100)# ip ospf bfd
```

To enable BFD on a trunk:

```
ACOS(config)# interface trunk 1
ACOS(config-if:trunk:1)# ip ospf bfd
```

To enable BFD for all OSPF-enabled interfaces, enter the following commands:

```
ACOS(config)# router ospf 1
ACOS(config-ospf:1)# bfd all-interfaces
```

To selectively disable BFD per interface, enter the following command:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip ospf bfd disable
```

To configure a multihop neighbor over a virtual-link, enter the following command:

```
ACOS(config-ospf:1)# area 1 virtual-link 40.0.0.1 fall-over bfd
```

Sample Configuration

Your running configuration will display your current BFD with OSPF configuration:

```
!
interface ethernet 1
ipv6 router ospf area 0 tag 1
ip address 20.0.0.1 255.255.255.0
 ip ospf bfd
!
interface ethernet 2
ipv6 router ospf area 0 tag 1
ip address 30.0.0.1 255.255.255.0
```

```
!
!
router ospf 1
 bfd all-interfaces
 network 20.0.0.0/24 area 0
 network 30.0.0.0/24 area 0
 area 1 virtual-link 40.0.0.1 fall-over bfd
!
!
bfd enable
!
```

Configuring BFD with OSPF (for IPv6)

To enable BFD with OSPF for IPv6 support on an interface, enter one of the following sets of commands:

To enable BFD on an individual interface:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ipv6 address 2001::1/64
ACOS(config-if:ethernet:1)# ipv6 router ospf area 0 tag 1
ACOS(config-if:ethernet:1)# ipv6 ospf bfd
```

To enable BFD on a virtual interface:

```
ACOS(config)# interface ve 100
ACOS(config-if:ve:100)# ipv6 ospf bfd
```

To enable BFD on a trunk:

```
ACOS(config)# interface trunk 1
ACOS(config-if:trunk:1)# ipv6 ospf bfd
```

To enable BFD for all OSPFv3-enabled interfaces, enter the following commands:

```
ACOS(config)# router ipv6 ospf 1
ACOS(config-ospf:1)# bfd all-interfaces
```

To selectively disable BFD per interface, enter the following command:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ipv6 ospf bfd disable
```

To configure a multihop neighbor over a virtual-link, enter the following command:

```
ACOS(config-ospf:1)# area 1 virtual-link 2.2.2.2 fall-over bfd
```

Sample Configuration

Your running configuration will display your current BFD with OSPF for IPv6 configuration:

```
!
interface ethernet 1
 ipv6 address 2001::1/64
ipv6 router ospf area 0 tag 1
 ipv6 ospf bfd
!
interface ethernet 2
ipv6 router ospf area 0 tag 1
 ipv6 address 3001::1/64
!
!
router ipv6 ospf 1
 router-id 1.1.1.1
 bfd all-interfaces
 area 1 virtual-link 2.2.2.2 fall-over bfd
!
!
bfd enable
!
```

Configuring BFD with IS-IS (for IPv4)

To enable BFD with ISIS on an interface, enter one of the following sets of commands:

To enable BFD on an individual interface:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# ip address 20.0.0.1 255.255.255.0
ACOS(config-if:ethernet:1)# ip router isis
ACOS(config-if:ethernet:1)# isis bfd
```

To enable BFD on a virtual interface:

```
ACOS(config)# interface ve 100
ACOS(config-if:ve:100)# isis bfd
```

To enable BFD on a trunk:

```
ACOS(config)# interface trunk 1
ACOS(config-if:trunk:1)# isis bfd
```

To enable BFD for all IS-IS-enabled interfaces, enter the following commands:

```
ACOS(config)# router isis
ACOS(config-isis)# bfd all-interfaces
ACOS(config-isis)# net 49.0001.0000.0000.0001.00
```

To selectively disable BFD per interface, enter the following command:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# isis bfd disable
```

Sample Configuration

Your running configuration will display your current BFD with ISIS configuration:

```
!
interface ethernet 1
 ip address 20.0.0.1 255.255.255.0
 ip router isis
 isis bfd
!
interface ethernet 2
 ip address 30.0.0.1 255.255.255.0
 ip router isis
 isis bfd
!
!
router isis
 bfd all-interfaces
 net 49.0001.0000.0000.0001.00
!
!
bfd enable
!
```

Configuring BFD with IS-IS (for IPv6)

To enable BFD with ISIS for IPv6 support on an interface, enter one of the following sets of commands:

To enable BFD on an individual interface:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# isis bfd
```

To enable BFD on a virtual interface:

```
ACOS(config)# interface ve 100
ACOS(config-if:ve:100)# ipv6 address 2ffe:123::1/64
ACOS(config-if:ve:100)# ipv6 router isis
ACOS(config-if:ve:100)# isis bfd
```

To enable BFD on a trunk:

```
ACOS(config)# interface trunk 1
ACOS(config-if:trunk:1)# isis bfd
```

To enable BFD for all IS-IS-enabled interfaces, enter the following commands:

```
ACOS(config)# router isis
ACOS(config-isis)# bfd all-interfaces
ACOS(config-isis)# net 49.0001.0000.0000.0002.00
```

To selectively disable BFD per interface, enter the following command:

```
ACOS(config)# interface ethernet 1
ACOS(config-if:ethernet:1)# isis bfd disable
```

Sample Configuration

Your running configuration will display your current BFD with ISIS (for IPv6 support) configuration:

```
!
interface ve 100
ipv6 address 2ffe:123::1/64
ipv6 router isis
isis bfd
!
router isis
```

```
 bfd all-interfaces
 net 49.0001.0000.0000.0002.00
!
bfd enable
```

Configuring BFD with BGP

When BFD is configured with BGP, it is configured on a per neighbor basis. This is different from the OSPF or ISIS configuration with BFD. Use the following commands to configure BFD with BGP:

```
ACOS(config)# router bgp 1
ACOS(config-bgp:1)# neighbor 1.2.3.4 fall-over bfd
```

To configure a multihop BFD neighbor, use the following command:

```
ACOS(config-bgp:1)# neighbor 1.2.3.4 fall-over bfd multihop
```

Sample Configuration

Your running configuration will display your current BFD with BGP configuration:

```
!
router bgp 1
 neighbor 1.2.3.4 remote-as 2
 neighbor 1.2.3.4 fall-over bfd multihop
!
!
bfd enable
!
```

Configuring Static BFD

The following topics are covered:

The following sections describe how to configure global IPv4 static BFD and both global and link-local IPv6 static BFD.

IPv4 Static BFD (Global)

From the global configuration mode, use the following command to add a static BFD entry for the specified IPv4 nexthop:

```
ACOS(config)# ip route static bfd 20.0.0.1 20.0.0.2
```

In the above command, the first parameter is the IPv4 address of the local interface. You can only use the IP addresses for interfaces to setup the BFD session. The second parameter is the IPv4 address of the remote interface that serves as the gateway for the static route.

IPv4 Static BFD with Resource Tracking Template

From the global configuration mode, use the following command to integrate the resource-tracking template.

```
ACOS(config)# ip route static bfd 20.0.0.1 20.0.0.2 template res_track_
temp_1 threshold 40 down
```

In the above command, if the resources tracked by the template `res_track_temp_1` change their state and exceed the configured threshold, then the BFD session will be brought down.

IPv6 Static BFD (Global)

From the global configuration mode, use the following command to add a static BFD entry for the specified IPv6 nexthop:

```
ACOS(config)#ipv6 route static bfd 2001::1 2001::2
```

In the above command, the first parameter is the IPv6 address of the local interface. You can only use the IP addresses for interfaces to setup the BFD session. The second parameter is the IPv6 address of the remote interface that serves as the gateway for the static route.

IPv6 Static BFD with Resource Tracking Template

From the global configuration mode, use the following command to integrate the resource-tracking template.

```
ACOS(config)#ipv6 route static bfd 20.0.0::1 20.0.0::2 template res_track_
temp_1 threshold 40 down
```

In the above command, if the resources tracked by the template `res_track_temp_1` change their state and exceed the configured threshold, then the BFD session will be brought down.

IPv6 Static BFD (Link-Local)

From the global configuration mode, use the following command to add a static BFD entry for the specified link-local IPv6 nexthop:

```
ACOS(config)# ipv6 route static bfd ve 100 fe80::1
```

In the above command, the first parameter is the local interface name (Ethernet, VE, or a specified trunk), and the second parameter is the remote link-local IPv6 address that serves as the gateway.

Configuring BFD Intervals

The following topics are covered:

Global Interval Configuration

From the global configuration mode, use the following command to modify the global interval timer values:

```
ACOS(config)# bfd interval 500 min-rx 500 multiplier 4
```

This command will help configure the interval for any one of the following three parameters and will be applied to all BFD sessions:

- DesiredMinTxInterval
- RequiredMinRxInterval

- Multiplier

Interface Interval Configuration

From the interface configuration mode, use the following command to modify the interface interval timer values:

```
ACOS(config)# interface ve 10
ACOS(config-if:ve:10)# bfd interval 500 min-rx 500 multiplier 4
```

NOTE:	For a BFD session for BGP using a loopback address, for an OSPFv2 virtual link, and for an OSPFv3 virtual link, the ACOS device will always use the global timer configuration, immaterial of the timer that is configured at the interface level.

Enable Authentication

The following topics are covered:

Authentication Per Interface

To configure authentication per interface, from the interface configuration mode, apply one of the following authentication schemes to OSPF, OSPFv3, IS-IS, or static BFD neighbors.

```
bfd authentication 1 md5 password-string
```

You may choose an authentication method from the following available choices:

- Simple password

- Keyed MD5

- Meticulous Keyed MD5

- Keyed SHA1

- Meticulous Keyed SHA1

Authentication Per Neighbor (For BGP Only)

The following command is configured under the BGP configuration mode:

```
ACOS(config)# router bgp 10
ACOS(config-bgp:10)# neighbor 1.2.3.4 fall-over bfd authentication 1 md5
password-string
```

Enabling Echo and Demand Function

The following topics are covered:

Enabling the Echo Function

From the global configuration mode, enable the BFD echo:

```
ACOS(config)# bfd echo
```

Enabling the Echo Function Per Interface

After you configure the global BFD echo, from the interface configuration mode, you can enable BFD echo on a per interface basis using the following command:

```
ACOS(config-if:ethernet:1)# bfd echo
```

Enabling Demand Mode

From the interface configuration mode, you can enable the demand mode to work in conjunction with the echo function using the following command:

```
ACOS(config-if:ethernet:1)# bfd echo demand
```

When demand mode is enabled, after a BFD session is established, a system will be able to verify connectivity with another system at will instead of routinely. Instead of constantly receiving BFD control packets, the system will request that the other system stop sending BFD Control packets. To verify connectivity again, the system

will explicitly send a short sequence of BFD Control packets to the other system and receive a response. Demand mode can be configured to work either independently in each direction, or bidirectionally at the same time.

Asynchronous Mode

The Asynchronous mode is the default mode of operation for BFD. In this mode, systems establish connectivity and know of each other's existence by periodically exchanging BFD Control packets. A session between two connected systems is only declared down after several packets in a row are not received by the other system. BFD will operate in this mode if you do not configure or enable echo or demand.

Viewing BFD Status

BFD status information and details can be viewed using the `show bfd` command along with additional options.

For more information, see `show bfd` in the *Command Line Interface Reference* document.

Micro-BFD for Trunk Ports

Bidirectional Forwarding Detection (BFD) provides fast failure detection for control protocols. RFC 7130 provides a mechanism to run BFD on Link Aggregation Group (LAG) interfaces, also known as Trunks, by running an independent asynchronous mode BFD session on every LAG member link. This feature enables the verification of member link continuity in the absence of LACP. It also provides a faster failure-detection time as compared to the LACP. Additionally, it provides the ability to verify each member link to be able to forward L3 packets.

ACOS supports the execution of a BFD session on each LAG member link, called per-LAG-member-link BFD sessions "micro-BFD sessions". It supports static and LACP trunk ports. Only one type of micro-BFD session may exist per member link: an IPv4 or IPv6 session. In addition, you may configure authentication over the Micro-BFD sessions.

The micro-BFD sessions on the member links are independent BFD sessions with unique local discriminator values, a set of state variables, and independent state machines. When the micro-BFD session is in the Down, AdminDown, or Init states, the initial BFD packets of this session use the dedicated multicast MAC 01-00-5E-90-00-01 as the destination MAC. When a micro-BFD session is in the UP state, it sends the packet with the learned remote LAG-interface MAC. The Micro-BFD sessions use port 6784 as the UDP destination port.

NOTE:	BFD should be enabled on all partitions running Micro-BFD sessions.

ACOS supports the following for Micro-BFD:

Static Trunk

- For Static trunk, after configuring local and neighbor addresses, the BFD sessions start on all trunk member ports.

- When a specific member port BFD session goes down, the trunk member port goes into the block state, and only Micro-BFD packets pass through.

- If the BFD session is UP again, it brings the member port UP.

- The BFD sessions are not removed when they are in up or down states.

LACP Trunk

- The BFD session is initiated once the LACP on the trunk member port becomes UP.

- When the BFD session goes down, it brings down the LACP state on the trunk member port and the member port goes into block state, and the BFD session is deleted.

- If the LACP on the member port goes down, it deletes the BFD session.

Configuring Micro-BFD

The following commands configure Micro-BFD:

- To establish a micro-BFD session on the trunk ports for IPv4:

```
ACOS(config)#interface trunk 2
ACOS(config-if:trunk:2)#bfd per-member-port
ACOS(config-if:trunk:2-per-member-port)#ipv4 local-address 10.10.10.1
```

```
ACOS(config-if:trunk:2-per-member-port)#ipv4 neighbor-address 10.10.10.2
```

- To establish a micro-BFD session on the trunk ports for IPv6:

```
ACOS(config)#interface trunk 3
ACOS(config-if:trunk:3)#bfd per-member-port
ACOS(config-if:trunk:3-per-member-port)#ipv6 local-address 2001:1::1
ACOS(config-if:trunk:3-per-member-port)#ipv6 neighbor-address 2001:1::2
```

Micro-BFD Limitations

The following are the limitations of Micro-BFD:

- Only one type of Micro-BFD session may be configured per trunk – either IPv4 or IPv6.

- BFD echo and echo-demand are not supported for the Micro-BFD sessions.

- When a Firewall rule is configured, you must either disable the Unicast Reverse Path Forwarding (URPF) check or configure the micro-BFD source address on an interface. Otherwise, the BFD session state will not be UP.

Displaying Micro-BFD Information

- To view the trunk type, Static or Link Aggregation Control Protocol (LACP), a new option Trunk Type is introduced in the following show command:

```
show trunk
```

- To view if the BFD sessions are Micro-BFD, the interface (trunk member port) name will be added next to the ACOS interface (Our Address) in the following show command:

```
show bfd neighbors
```

Internet Group Multicast Protocol (IGMP) Queries

The following topics are covered:

Overview

The current implementation of the ACOS software supports the generation of generic Internet Group Multicast Protocol version 2 (IGMPv2) membership query requests. ACOS devices will now generate IGMP membership queries and facilitate multicast deployments.

NOTE: The ACOS software does not support the complete IGMP protocol or the generation of generic membership queries for IGMPv3 or Multicast Listener Discovery (MLDv2).

Previous releases of the ACOS software did not provide support for the IGMPv2 protocol at all, hence it did not provide IGMP membership query support.

IGMPv2 provides the following capabilities:

- IGMP membership queries are only generated when IPv4 addresses are configured. If any IPv6 interface addresses are recognized, no queries will be generated.

- Generates generic IGMPv2 membership query request packets.

- The devices will not process any responses for this query request.

- Uses the default values for membership query request wherever possible.

- Provides the ability to configure the time interval for generation of these membership queries per interface.

- Provides support for this feature with Layer 3 Virtualization (L3V).

IGMP membership queries are supported in routed mode only and will not be supported in non-routed mode.

Figure 12 : IGMP Membership Queries (Routed and Non-Routed Mode)

In Routed Mode

In IGMP Membership Queries (Routed and Non-Routed Mode), the interface for devices 1 and 2 are acting in routed mode, that is, the IP address has been configured on the interface. When the interface is in routed mode, the device can be configured to generate IGMPv2 membership queries out of this interface. However, when an IGMP membership query is received on an interface in routed mode, it will be ignored.

In Non-Routed Mode

In this figure, the Device 2 device is acting as a switch and both Eth 11 and Eth12 on the Device 2 device are in non-routed mode. Eth1 on the Device 1 device and Eth2 on the Device 2 device are configured in routed mode. Hence Eth1 interface on the Device 1 device and Eth2 on the Device 3 device can be configured to generate IGMP Membership Queries.

In this case, when the Device 2 device receives IGMP Membership Queries on Eth11 (generated by the Device 1 device) and Eth 12 (generated by the Device 3 device) it will accept these packets and just switch them as it would any other packet. More importantly, it will not drop these packets since Eth11 and Eth12 on Device 2 are acting in non-routed (switched) mode.

Configuring IGMP Membership Queries

The GUI and the CLI provide a way to configure IGMPv2 membership request queries from the physical, virtual or trunk interface configuration level.

Use the GUI to Configure IGMP Membership Queries

To configure IGMPv2 membership request queries on an interface:

1. Hover over **Network** in the navigation bar, and select **Interface** from the drop-down menu.

2. Depending on the type of interface on which to configure this feature, select LAN, Virtual Ethernet or Trunk from the menu bar.

3. Click **Edit** in the actions column for the interface on which to configure this feature.

4. Expand the IP section to reveal additional configuration options.

5. Select the Generate Membership Query field.

6. In the Membership Query Interval field, specify the time interval (1-255 seconds) after which the device using this interface will initiate an IGMP membership query request.

7. In the Maximum Response Time field, specify the time interval, in 1/10 of a second, before which receiving devices will send the ICMP query message response.

8. Click the **Update** button.

NOTE:	These timers are valid only for a particular interface. They must be configured per interface.

Use the CLI to Configure IGMP Membership Queries

To configure IGMP membership request queries on a physical interface, use the ip igmp command from interface configuration level. For example:

```
ACOS(config-if)# interface ethernet 2
ACOS(config-if:ethernet:2)# ip address 192.168.1.1 /24
ACOS(config-if:ethernet:2)# ip igmp generate-membership-query 10 max-resp-
time 50
```

To view your IGMP membership request query configuration for a a physical interface, do the following:

```
ACOS(config)# show interfaces ethernet 2
  Ethernet 2 is up, line protocol is up
  Hardware is GigabitEthernet, Address is 001f.a004.2e71
  Internet address is 192.168.1.1, Subnet mask is 255.255.255.0
  Configured Speed auto, Actual 1Gbit, Configured Duplex auto, Actual fdx
  IGMP Membership Query is enabled, IGMP Membership Queries sent 3
  Flow Control is disabled, IP MTU is 1500 bytes
```

```
Port as Mirror disabled, Monitoring this Port disabled
0 packets input,  0 bytes
Received 0 broadcasts,  Received 0 multicasts,  Received 0 unicasts
0 input errors,  0 CRC  0 frame
0 runts  0 giants
3003 packets output  264264 bytes
Transmitted 0 broadcasts  3003 multicasts  0 unicasts
0 output errors  0 collisions
300 second input rate: 0 bits/sec, 0 packets/sec, 0% utilization
300 second output rate: 12768 bits/sec, 18 packets/sec, 0% utilization
```

To configure IGMP membership request queries on an virtual Ethernet interface, do the following:

```
ACOS(config)# vlan 50
ACOS(config-vlan:50)# tagged ethernet 1
ACOS(config-vlan:50)# router-interface ve 50
ACOS(config-vlan:50)# exit
ACOS(config)# interface ve 50
ACOS(config-if:ve:50)# ip address 10.10.10.219 /24
ACOS(config-if:ve:50)# ip igmp generate-membership-query 10 max-resp-time
50
```

To view your IGMP membership request query configuration for a virtual Ethernet interface, do the following:

```
ACOS(config)# show interfaces ve 50
VirtualEthernet 50 is up, line protocol is up
  Hardware is VirtualEthernet, Address is 001f.a004.2e72
  Internet address is 10.10.10.219, Subnet mask is 255.255.255.0
  Router Interface for L2 Vlan 50
  IP MTU is 1500 bytes
  IGMP Membership Query is enabled, IGMP Membership Queries sent 32
  0 packets input  0 bytes
  Received  0 broadcasts, Received 0 multicasts, Received 0 unicasts
  0 packets output  0 bytes
  Transmitted  0 broadcasts, Transmitted 0 multicasts, Transmitted 0
unicasts
  300 second input rate: 0 bits/sec, 0 packets/sec
  300 second output rate: 0 bits/sec, 0 packets/sec
```