Configuring VXLAN OAM

This chapter contains these sections:

VXLAN OAM Overview

The VXLAN operations, administration, and maintenance (OAM) protocol is a protocol for installing, monitoring, and troubleshooting Ethernet networks to enhance management in VXLAN based overlay networks.

Similar to ping, traceroute, or pathtrace utilities that allow quick determination of the problems in the IP networks, equivalent troubleshooting tools have been introduced to diagnose the problems in the VXLAN networks. The VXLAN OAM tools, for example, ping, pathtrace, and traceroute provide the reachability information to the hosts and the VTEPs in a VXLAN network. The OAM channel is used to identify the type of the VXLAN payload that is present in these OAM packets.

There are two types of payloads supported:

- Conventional ICMP packet to the destination to be tracked
- Special NVO3 draft Tissa OAM header that carries useful information

The ICMP channel helps to reach the traditional hosts or switches that do not support the new OAM packet formats. The NVO3 draft Tissa channels helps to reach the supported hosts or switches and carries the important diagnostic information. The VXLAN NVO3 draft Tissa OAM messages may be identified via the reserved OAM EtherType or by using a well-known reserved source MAC address in the OAM packets depending on the implementation on different platforms. This constitutes a signature for recognition of the VXLAN OAM packets. The VXLAN OAM tools are categorized as shown in table below.

Table 1: VXLAN OAM Tools

Category	Tools
Fault Verification	Loopback Message
Fault Isolation	Path Trace Message
Performance	Delay Measurement, Loss Measurement
Auxiliary	Address Binding Verification, IP End Station Locator, Error Notification, OAM Command Messages, and Diagnostic Payload Discovery for ECMP Coverage

Loopback (Ping) Message

The loopback message (The ping and the loopback messages are the same and they are used interchangeably in this guide) is used for the fault verification. The loopback message utility is used to detect various errors and the path failures. Consider the topology in the following example where there are three core (spine) switches labeled Spine 1, Spine 2, and Spine 3 and five leaf switches connected in a Clos topology. The path of an example loopback message initiated from Leaf 1 for Leaf 5 is displayed when it traverses via Spine 3. When the loopback message initiated by Leaf 1 reaches Spine 3, it forwards it as VXLAN encapsulated data packet based on the outer header. The packet is not sent to the software on Spine 3. On Leaf 3, based on the appropriate loopback message signature, the packet is sent to the software VXLAN OAM module, that in turn, generates a loopback response that is sent back to the originator Leaf 1.

The loopback (ping) message can be destined to VM or to the (VTEP on) leaf switch. This ping message can use different OAM channels. If the ICMP channel is used, the loopback message can reach all the way to the VM if the VM's IP address is specified. If NVO3 draft Tissa channel is used, this loopback message is terminated on the leaf switch that is attached to the VM, as the VMs do not support the NVO3 draft Tissa headers in general. In that case, the leaf switch replies back to this message indicating the reachability of the VM. The ping message supports the following reachability options:

Ping

Check the network reachability (**Ping** command):

- From Leaf 1 (VTEP 1) to Leaf 2 (VTEP 2) (ICMP or NVO3 draft Tissa channel)
- From Leaf 1 (VTEP 1) to VM 2 (host attached to another VTEP) (ICMP or NVO3 draft Tissa channel)

Figure 1: Loopback Message

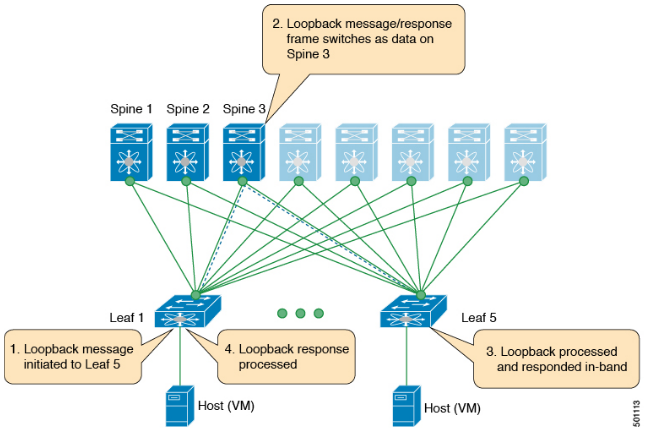
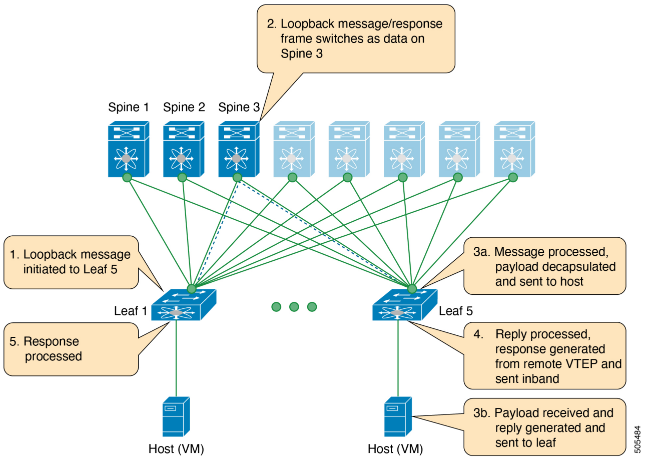

2. Loopback message/response frame switches as data on Spine 3
Spine 1
Spine 2
Spine 3
Leaf 1
Leaf 5
1. Loopback message initiated to Leaf 5
4. Loopback response processed
3. Loopback processed and responded in-band
Host (VM)
Host (VM)
501113

Figure 2: NVO3 Draft Tissa Ping to Remote VM


2. Loopback message/response frame switches as data on Spine 3
Spine 1
Spine 2
Spine 3
1. Loopback message initiated to Leaf 5
3a. Message processed, payload decapsulated and sent to host
Leaf 1
Leaf 5
5. Response processed
4. Reply processed, response generated from remote VTEP and sent inband
3b. Payload received and reply generated and sent to leaf
Host (VM)
Host (VM)
505484

Traceroute and Pathtrace Message

The traceroute and pathtrace message are used for the fault isolation. In a VXLAN network, it may be desirable to find the list of switches that are traversed by a frame to reach the destination. When the loopback test from a source switch to a destination switch fails, the next step is to find out the offending switch in the path. The operation of the path trace message begins with the source switch transmitting a VXLAN OAM frame with a TTL value of 1. The next hop switch receives this frame, decrements the TTL, and on finding that the TTL is 0, it transmits a TTL expiry message to the sender switch. The sender switch records this message as an indication of success from the first hop switch. Then the source switch increases the TTL value by one in the next path trace message to find the second hop. At each new transmission, the sequence number in the message is incremented. Each intermediate switch along the path decrements the TTL value by 1 as is the case with regular VXLAN forwarding.

This process continues until a response is received from the destination switch, or the path trace process timeout occurs, or the hop count reaches a maximum configured value. The payload in the VXLAN OAM frames is referred to as the flow entropy. The flow entropy can be populated so as to choose a particular path among multiple ECMP paths between a source and destination switch. The TTL expiry message may also be generated by the intermediate switches for the actual data frames. The same payload of the original path trace request is preserved for the payload of the response.

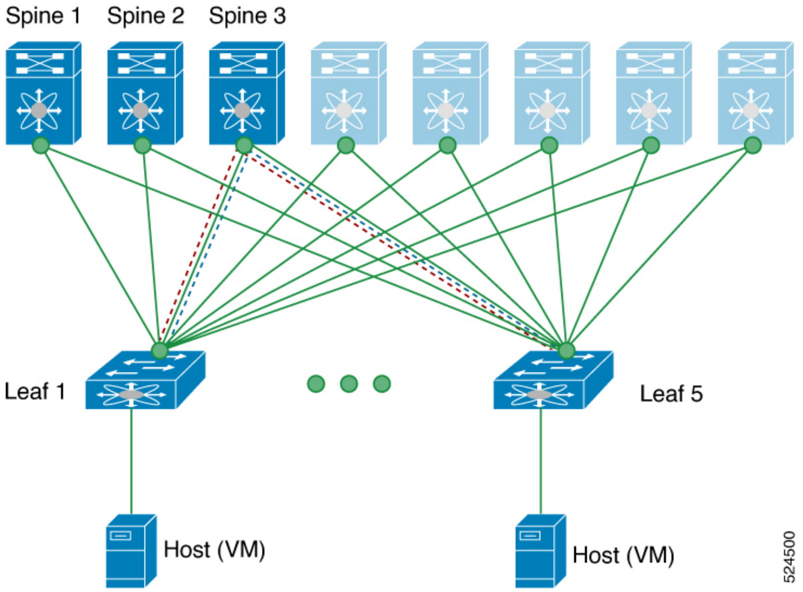
The traceroute and pathtrace messages are similar, except that traceroute uses the ICMP channel, whereas pathtrace use the NVO3 draft Tissa channel. Pathtrace uses the NVO3 draft Tissa channel, carrying additional diagnostic information, for example, interface load and statistics of the hops taken by these messages. If an intermediate device does not support the NVO3 draft Tissa channel, the pathtrace behaves as a simple traceroute and it provides only the hop information. For hops that are not supported, an error message "non OAM capable switch" will be displayed.

Traceroute

Trace the path that is traversed by the packet in the VXLAN overlay using **Traceroute** command:

- Traceroute uses the ICMP packets (channel-1), encapsulated in the VXLAN encapsulation to reach the host

Figure 3: Traceroute Message

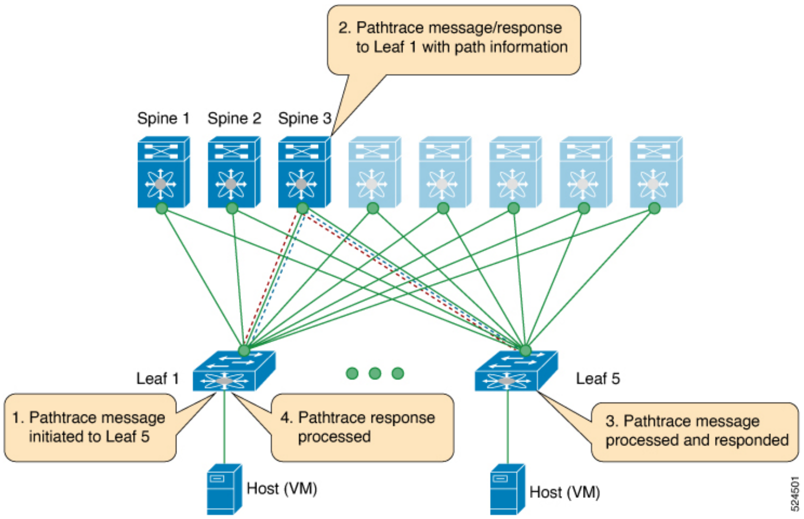


Pathtrace

Trace the path that is traversed by the packet in the VXLAN overlay using the NVO3 draft Tissa channel with **Pathtrace** command:

- Pathtrace uses special control packets like NVO3 draft Tissa or TISSA (channel-2) to provide additional information regarding the path (for example, ingress interface and egress interface). These packets terminate at VTEP and they does not reach the host. Therefore, only the VTEP responds.
- Beginning with NX-OS release 9.3(3), the `Received` field of the **show ngoam pathtrace statistics summary** command indicates all pathtrace requests received by the node on which the command is executed regardless of whether the request was destined to that node.

Figure 4: Pathtrace Message


2. Pathtrace message/response to Leaf 1 with path information
Spine 1
Spine 2
Spine 3
Leaf 1
Leaf 5
1. Pathtrace message initiated to Leaf 5
4. Pathtrace response processed
3. Pathtrace message processed and responded
Host (VM)
Host (VM)
524501

VXLAN EVPN Loop Detection and Mitigation

Causes and Impacts of Loop

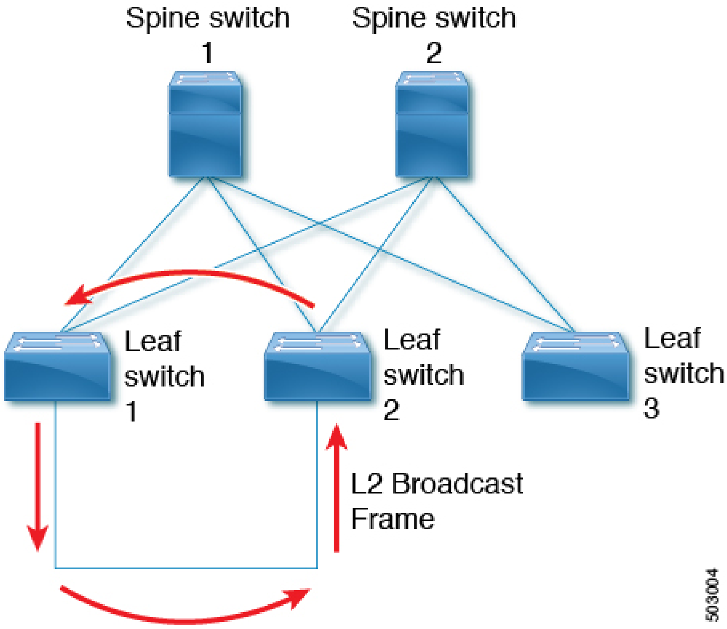
Loops usually occur in a VXLAN EVPN fabric due to incorrect cabling on the south side (access side) of the fabric. When broadcast packets are injected into a network with a loop, the frame remains bridged in the loop. As more broadcast frames enter the loop, they accumulate and can cause a serious disruption of services.

About VXLAN EVPN Loop Detection and Mitigation

Cisco NX-OS Release 9.3(5) introduces VXLAN EVPN loop detection and mitigation. This feature detects Layer 2 loops in a single VXLAN EVPN fabric or a Multi-Site environment. It operates at the port/VLAN level and disables the VLAN(s) on each port where a loop is detected. Administrators are also notified (via syslog) about the condition. In this way, the feature ensures that the network remains up and available.

The following figure shows an EVPN fabric in which two leaf devices (Leaf1 and Leaf2) are directly connected on the south side due to incorrect cabling. In this topology, Leaf3 forwards an L2 broadcast frame to Leaf1. Then the broadcast frame is repeatedly forwarded between Leaf1 and Leaf2 through the south side and the fabric. The forwarding continues until the incorrect cabling is fixed.

Figure 5: Two Leaf Nodes Directly Connected



This feature operates in three phases:

1. Loop Detection: Sends a loop detection probe under the following circumstances: when requested by a client, as part of a periodic probe task, and as soon as any port comes up.

2. Loop Mitigation: Blocks the VLANs on a port once a loop has been discovered and displays a syslog message similar to the following:

   ```
   2020 Jan 14 09:58:44 Leaf1 %NGOAM-4-SLD_LOOP_DETECTED: Loop detected - Blocking vlan
   1001 :: Eth1/3
   ```

   or

   ```
   2024 Sep  9 15:28:01 Node-11 %ETHPORT-3-IF_ERROR_VLANS_SUSPENDED: VLANs 2704 on Interface
    Ethernet1/49/1 are being suspended. (Reason: SUCCESS)
   ```

   Because loops can lead to incorrect local MAC address learning, this phase also flushes the local and remote MAC addresses. Doing so removes any MAC addresses that are incorrectly learned.

   In the previous figure, MAC addresses can be incorrectly learned because packets from hosts sitting behind the remote leaf (Leaf3) can reach both Leaf1 and Leaf2 from the access side. As a result, the hosts incorrectly appear local to Leaf1 and Leaf2, which causes the leafs to learn their MAC addresses.

3. Loop Recovery: Once a loop is detected on a particular port or VLAN and the recovery interval has passed, recovery probes are sent to determine if the loop still exists. When NGOAM recovers from the loop, a syslog message similar to the following appears:

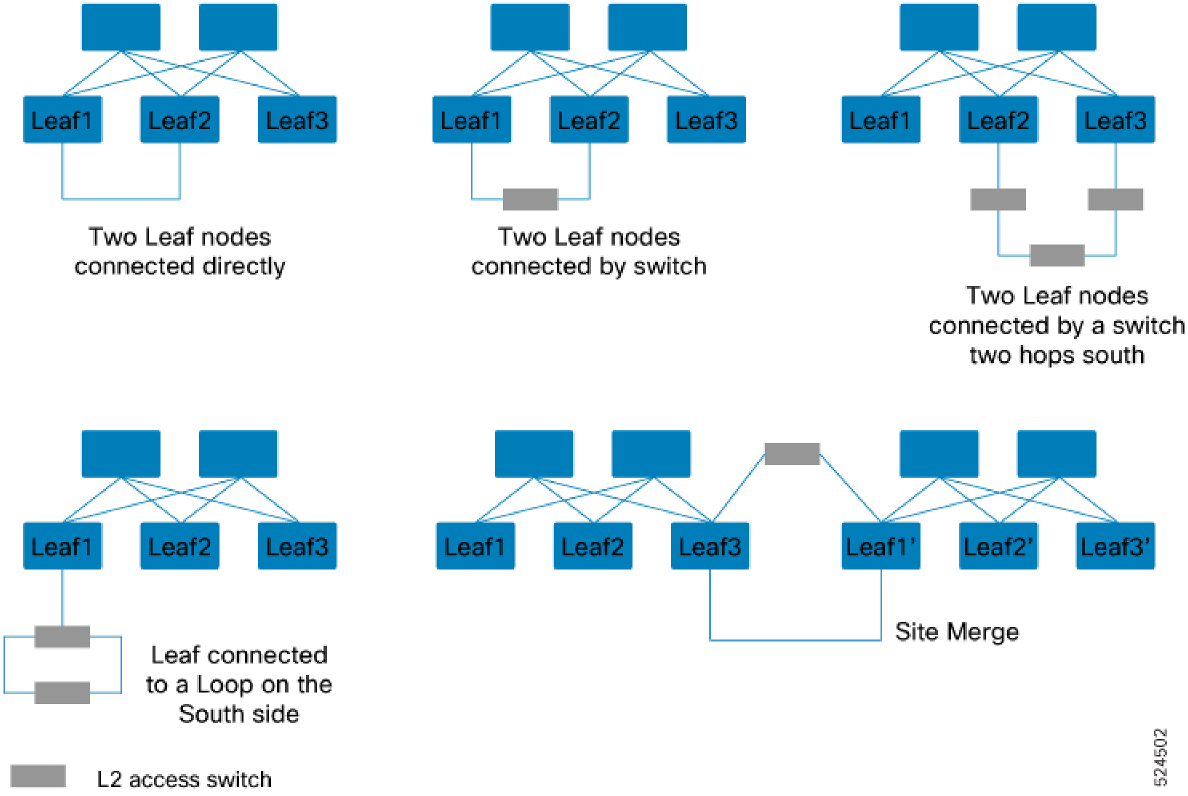
   ```
   2020 Jan 14 09:59:38 Leaf1 %NGOAM-4-SLD_LOOP_GONE: Loop cleared - Enabling vlan 1001
   :: Eth1/3
   ```

   or

   ```
   2024 Sep  9 15:24:23 Node-11 %ETHPORT-3-IF_ERROR_VLANS_REMOVED: VLANs 384 on Interface
   Ethernet1/49/1 are removed from suspended state.
   ```

Note The default logging level for NGOAM does not generate a syslog message. Modifying the logging level of NGOAM to 5 with "logging level ngoam 5" will result in a syslog message being generated when a loop is detected.

Various Loop Scenarios


Leaf1
Leaf2
Leaf3
Two Leaf nodes connected directly
Leaf1
Leaf2
Leaf3
Two Leaf nodes connected by switch
Leaf1
Leaf2
Leaf3
Two Leaf nodes connected by a switch two hops south
Leaf1
Leaf2
Leaf3
Leaf connected to a Loop on the South side
Leaf1
Leaf2
Leaf3
Leaf1'
Leaf2'
Leaf3'
Site Merge
L2 access switch
524502

About Southbound Loop Detection on Layer-3 Interface

Beginning with NX-OS release 10.4(3)F, Cisco Nexus switches support Southbound Loop Detection (SLD) on a Layer-3 (L3) Ethernet and L3 port-channel interfaces in a single VXLAN EVPN fabric or a Multi-Site environment. Before this release, the SLD feature was supported only on Layer-2 interfaces.

This feature detects loops in southbound side (L2 access switches) that are connected to a single leaf switch through an L3 interface or port channel.

When the SLD feature is enabled on the L3 interface, it sends periodic SLD probes to detect loops in a downstream tenant's Layer-2 domain. It continues to monitor for loops and blocks the L3 interface on detection until the user takes action to correct the condition in the downstream L2 domain.

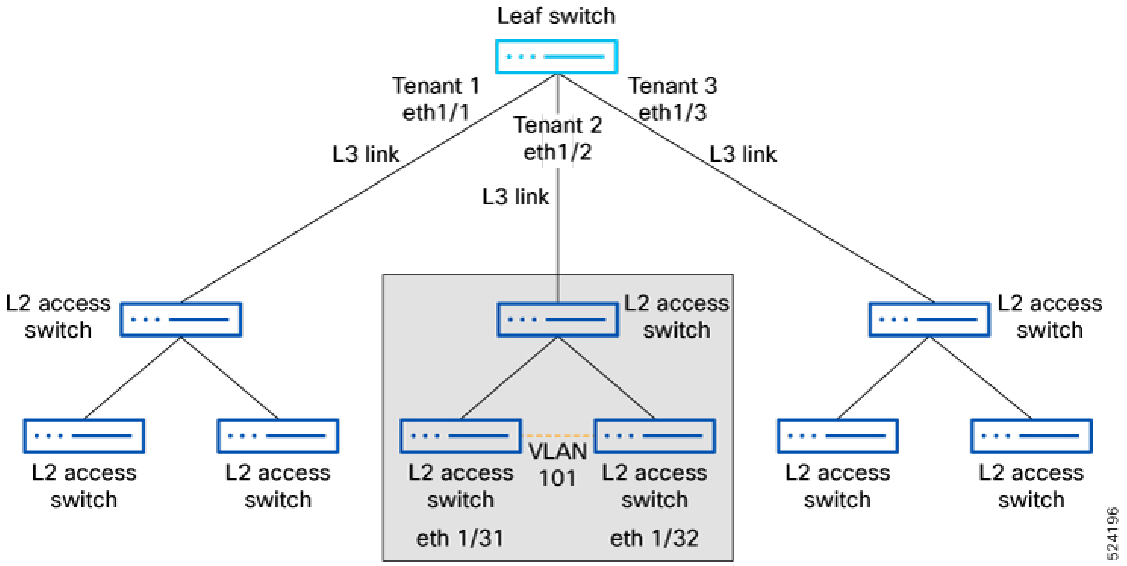
Functionalities of SLD on Layer-3 Interface

- Isolates a single L3 attached tenant to prevent the impact of a storm from propagating beyond a single L3 boundary due to control-plane policing congestion.

- Detects downstream L2 loops and blocks attached L3 interface or L3 port-channel if a loop is detected by receipt of an originated NGOAM probe.
- Unblocks the L3 port if the originated NGOAM probes are no longer detected.

Topology Overview of SLD on Layer-3 Interface

The following figure shows an EVPN fabric with a leaf switch configured with three VRFs (Tenant 1, Tenant 2, and Tenant 3). These VRFs are connected to L2 access switches on the south side using different L3 ports and their respective L3 interfaces.



This feature operates in three phases:

- **Loop Detection**: The SLD L3 feature sends periodic probes to detect loops in the downstream tenant's Layer-2 domain (L2 access switches).

  SLD sends a loop detection probe under the following circumstances: when requested by a client, as part of a periodic probe task, and when any port comes up.

  For example: Tenant 2 accidentally creates a bridging loop due to a cabling error while disabling STP on local VLAN 101. This triggers an ARP storm toward Eth1/2, consuming the entire CoPP Class Normal policer, which causes CoPP policer saturation in Tenant 1 and Tenant 3.

  ```
  2024 Jun 27 02:34:39 tenant2 %L2FM-2-L2FM_CONTINUOUS_MAC_MOVE: Mac
  Address (f80f.6f96.a127) in Vlan 101 is moving continuously. Mac moved
  between Eth1/32 to Eth1/31. Please enable 'logging level l2fm 4' for
  verbose output.
  ```

- **Loop Mitigation**: Blocks the L3 port when a loop has been discovered and displays a syslog message like the following indicating the loop detection and port status changes.

  ```
  2024 Jun 27 02:37:50 leaf %ETHPORT-5-IF_DOWN_NONE: Interface Ethernet1/2 is down (None)
  2024 Jun 27 02:37:50 leaf %ETHPORT-5-IF_DOWN_ERROR_DISABLED: Interface Ethernet1/2 is
  down (Error disabled. Reason:error)
  2024 Jun 27 02:38:52 leaf %ETHPORT-5-IF_ERRDIS_RECOVERY: Interface Ethernet1/2 is being
   recovered from error disabled state (Last Reason:error)
  2024 Jun 27 02:38:54 leaf %ETHPORT-5-IF_DOWN_ERROR_DISABLED: Interface Ethernet1/2 is
  ```

```
down (Error disabled. Reason:error)
!
leaf# show ngoam loop-detection status l3
Port       Status       NumLoops      DetectionTime             ClearedTime
=======================================================================================

Eth1/2     BLOCKED      2             Tue Jun 27 02:38:54 2024  Tue Jun 27 02:38:52
2024
```

After each probe error recovery interval, the blocked L3 port is brought up to send probes and recheck for the loop. Now, the Eth1/2 L3 interface is moved from the **Blocked** state to the **Forwarding** state. The probe checks for the loop, and if the loop still exists, it moves the eth1/2 L3 interface back to the **Blocked** state. This process continues until the user corrects the bridging loop within the L2 domain.

The following sample output displays the state (blocking and unblocking) based on the probe generated:

```
2024 Jun 27 20:26:56 leaf %NGOAM-4-SLD_L3_LOOP_DETECTED: Loop detected - Blocking port
 Eth1/2
2024 Jun 27 20:26:56 leaf %ETHPORT-5-IF_DOWN_NONE: Interface Ethernet1/2 is down (None)
2024 Jun 27 20:26:56 leaf %ETHPORT-5-IF_DOWN_ERROR_DISABLED: Interface Ethernet1/2 is
down (Error disabled. Reason:error)
2024 Jun 27 20:27:58 leaf %ETHPORT-5-IF_ERRDIS_RECOVERY: Interface Ethernet1/2 is being
 recovered from error disabled state (Last Reason:error)
2024 Jun 27 20:27:58 leaf %NGOAM-4-SLD_L3_LOOP_GONE: Loop cleared - Enabling port Eth1/2
2024 Jun 27 20:28:00 leaf %NGOAM-4-SLD_L3_LOOP_DETECTED: Loop detected - Blocking port
 Eth1/2
2024 Jun 27 20:28:01 leaf %ETHPORT-5-IF_DOWN_ERROR_DISABLED: Interface Ethernet1/2 is
down (Error disabled. Reason:error)
```

- **Loop Recovery**: When the cabling error is fixed, the loops on the southbound side will be removed. After the recovery interval has passed, recovery probes will be sent from the L3 interface on the leaf switch to determine whether a loop exists. If the loop is resolved, the port will remain in the forwarding state, and the following syslog message will be generated.

```
2024 Jun 27 22:39:26 tenant2 %ETHPORT-5-IF_DOWN_ADMIN_DOWN: Interface Ethernet1/32 is
down (Administratively down)
2024 Jun 27 22:39:56 tenant2 %ETHPORT-5-SPEED: Interface Ethernet1/2, operational speed
 changed to 10 Gbps
2024 Jun 27 22:39:56 tenant2 %ETHPORT-5-IF_DUPLEX: Interface Ethernet1/2, operational
duplex mode changed to Full
2024 Jun 27 22:39:56 tenant2 %ETHPORT-5-IF_RX_FLOW_CONTROL: Interface Ethernet1/2,
operational Receive Flow Control state changed to off
2024 Jun 27 22:39:56 tenant2 %ETHPORT-5-IF_TX_FLOW_CONTROL: Interface Ethernet1/2,
operational Transmit Flow Control state changed to off
2024 Jun 27 22:39:56 tenant2 %ETHPORT-5-IF_UP: Interface Ethernet1/17 is up
2024 Jun 27 22:41:03 tenant2 %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured from vty by admin
 on 10.82.195.201@pts/2
```

Note The default logging level for NGOAM does not generate a syslog message. Modifying the logging level of NGOAM to 5 with "logging level ngoam 5" will result in a syslog message being generated when a loop is detected.

L2 and L3 SLD Feature Functionality Comparison

Features	SLD on L2 Interface	SLD on L3 Interface
Operation level	Port and VLAN level	Ethernet and L3 port-channel

Features	SLD on L2 Interface	SLD on L3 Interface
Environment	Single-Site and Multi-Site	Single-Site and Multi-Site
Loop detection	Detects the loop of a particular port or VLAN	Detects downstream L2 loops and blocks the L3 interfaces or L3 port channels
Loop mitigation	Blocks the VLANs on a port once a loop has been discovered and displays a syslog message	Isolates a single L3 attached tenant to prevent the impact of a storm from propagating beyond a single L3 boundary by consuming shared CoPP policer resources
Loop blocking	Breaks the southbound loops	Isolates detected loops from impacting the control plane by shedding storm-related traffic
Loop recovery	Sends recovery probes, re-enables VLANs, and logs syslog messages once the loop is cleared	Sends recovery probes, re-enables the port or ethernet interfaces if the NGOAM process no longer sees NGOAM probes, and logs syslog messages once the loop is cleared

Guidelines and Limitations for VXLAN NGOAM

VXLAN NGOAM has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.2(3)F, you do not have to enable the VXLAN feature using the **feature nv overlay** command to use the NGOAM feature on intermediate nodes.
- ***** The Cisco Nexus 9800 switches support only NGOAM ping, traceroute, and pathtrace, but Xconnect and Southbound Loop Detection (SLD) are not supported.

Supported Platform and Release for VXLAN NGOAM

Supported Release	Supported Platform
9.3(3) and later	Cisco Nexus 9300-FX/FX2/GX Series switches
9.3(5) and later	Cisco Nexus 9300-FX3 Series switches
10.2(3)F and later	Cisco Nexus 9300-GX2 Series switches
10.4(1)F and later	Cisco Nexus 9332D-H2R switches
10.4(2)F and later	Cisco Nexus 93400LD-H1 switches
10.4(3)F and later	Cisco Nexus 9364C-H1 switches Cisco Nexus 9800 Series switches*

Guidelines and Limitations for VXLAN EVPN Loop Detection and Mitigation

VXLAN EVPN loop detection and mitigation has the following guidelines and limitations:

- VXLAN EVPN loop detection and mitigation is supported in both STP and STP-less environments.
- To be able to detect loops across sites for VXLAN EVPN Multi-Site deployments, the **ngoam loop-detection** command needs to be configured on all border gateways in the site where the feature is being deployed.
- VXLAN EVPN loop detection and mitigation isn't supported with the following features:
  - Private VLANs
  - VLAN translation
  - ESI-based multihoming
  - VXLAN Cross Connect
  - Q-in-VNI
  - EVPN segment routing (Layer 2)

Note Ports or VLANs configured with these features must be excluded from VXLAN EVPN loop detection and mitigation. You can use the **disable** {**vlan** *vlan-range*} [**port** *port-range*] command to exclude them.

Supported Platform and Release for VXLAN EVPN Loop Detection and Mitigation

Supported Release	Supported Platform
9.3(5) and later	Cisco Nexus 9300-EX/FX/FX2 and 9332C and 9364C Series switches Cisco Nexus 9500 platform switches with 9700-EX/FX line cards
10.1(1) and later	Cisco Nexus 9300-FX3/GX Series switches
10.2(3)F and later	Cisco Nexus 9300-GX2 Series switches
10.4(1)F and later	Cisco Nexus 9332D-H2R Series switches
10.4(2)F and later	Cisco Nexus 93400LD-H1 Series switches
10.4(3)F and later	Cisco Nexus 9364C-H1 Series switches

Guidelines and Limitations for SLD on L3 Interface

- SLD is supported only on L3 ethernet and L3 port-channel interfaces. It is not supported on L3 sub-interfaces.

Supported Platform and Release for SLD on L3 Interface

Release	Platform
10.4(3)F and later	Cisco Nexus 9300-EX/FX/FX2/GX/GX2/H2R/H1, 9332C and 9364C Series switches Cisco Nexus 9500 platform switches with 9700-EX/FX/GX line cards

Configuring VXLAN OAM

Before you begin

As a prerequisite, ensure that the VXLAN configuration is complete.

Note Beginning with Cisco NX-OS Release 10.2(3), you do not have to enable the VXLAN feature for configuring the NGOAM feature on intermediate nodes.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature ngoam**
3. switch(config)# **ngoam install acl**

DETAILED STEPS

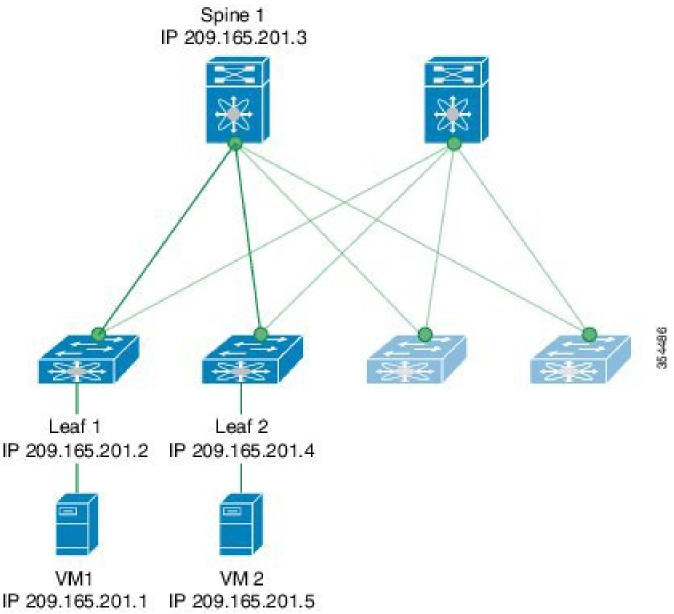
Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature ngoam**	Enters the NGOAM feature.
Step 3	switch(config)# **ngoam install acl**	Installs the NGOAM Access Control List (ACL). **Note** This command is deprecated beginning with Cisco NX-OS Release 9.3(5) and is required only for earlier releases.

Example

See the following examples of the configuration topology.

Figure 6: VXLAN Network



VXLAN OAM provides the visibility of the host at the switch level, that allows a leaf to ping the host using the **ping nve** command.

The following examples display how to ping from Leaf 1 to VM2 via Spine 1 with channel 1 (unique loopback) and with channel 2 (NVO3 Draft Tissa):

```
switch# ping nve ip 209.165.201.5 vrf vni-31000 source 1.1.1.1 verbose

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response


Sender handle: 34
! sport 40673 size 39,Reply from 209.165.201.5,time = 3 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
! sport 40673 size 39,Reply from 209.165.201.5,time = 1 ms
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/18 ms
Total time elapsed 49 ms

switch# ping nve ip unknown vrf vni-31000 payload ip 209.165.201.5 209.165.201.4 payload-end
 verify-host
<snip>
Sender handle: 34
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/18 ms
Total time elapsed 49 ms
```

Note The source ip-address 1.1.1.1 used in the above example is a loopback interface that is configured on Leaf 1 in the same VRF as the destination ip-address. For example, the VRF in this example is vni-31000.

The following example displays how to traceroute from Leaf 1 to VM 2 via Spine 1.

```
switch# traceroute nve ip 209.165.201.5 vrf vni-31000 source 1.1.1.1 verbose


Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response


Traceroute request to peer ip 209.165.201.4 source ip 209.165.201.2
Sender handle: 36
  1 !Reply from 209.165.201.3,time = 1 ms
  2 !Reply from 209.165.201.4,time = 2 ms
  3 !Reply from 209.165.201.5,time = 1 ms
```

The following example displays the output of the pathtrace from Leaf 2 to Leaf 1.

```
switch# pathtrace nve ip 209.165.201.4 vni 31000 verbose

Path trace Request to peer ip 209.165.201.4 source ip 209.165.201.2

Sender handle: 42
Hop Code    ReplyIP            IngressI/f    EgressI/f     State
======================================================================
1   !Reply from 209.165.201.3,  Eth5/5/1      Eth5/5/2     UP/UP
2   !Reply from 209.165.201.4,  Eth1/3        Unknown      UP/DOWN
```

The following example displays the output of the MAC ping from Leaf 2 to Leaf 1 using NVO3 draft Tissa channel:

```
switch# ping nve mac 0050.569a.7418 2901 ethernet 1/51 profile 4 verbose

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response

Sender handle: 408
!!!!Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/5 ms
Total time elapsed 104 ms

switch# show run ngoam
feature ngoam
ngoam profile 4
oam-channel 2
ngoam install acl
```

The following example displays how to pathtrace based on a payload from Leaf 2 to Leaf 1:

```
switch# pathtrace nve ip unknown vrf vni-31000 payload mac-addr 0050.569a.d927 0050.569a.a4fa
ip 209.165.201.5 209.165.201.1 port 15334 12769 proto 17 payload-end
```

```
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response

Path trace Request to peer ip 209.165.201.4 source ip 209.165.201.2
Sender handle: 46
Hop Code Reply            IngressI/f EgressI/f State
======================================================================
1 !Reply from 209.165.201.3, Eth5/5/1 Eth5/5/2 UP/UP
2 !Reply from 209.165.201.4, Eth1/3    Unknown UP/DOWN
```

Note When the total hop count to final destination is more than 5, the path trace default TTL value is 5. Use **max-ttl** option to finish VXLAN OAM path trace completely.

For example: **pathtrace nve ip unknown vrf vrf-vni13001 payload ip 200.1.1.71 200.1.1.23 payload-end verbose max-ttl 10**

The following example displays how to pathtrace NVE MAC:

```
pathtrace nve mac 0050.569a.d927 11 payload mac-addr 0050.569a.d927 0050.569a.a4fa payload-end
 vni 31000 verbose

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response,
'v' - Other - Use verbose to see the result

Path trace Request to peer ip 209.165.201.4 source ip 209.165.201.2
Sender handle: 46
Hop Code Reply             IngressI/f EgressI/f State
======================================================================
1 !Reply from 209.165.201.3, Eth5/5/1 Eth5/5/2 UP/UP
2 !Reply from 209.165.201.4, Eth1/3    Unknown UP/DOWN
```

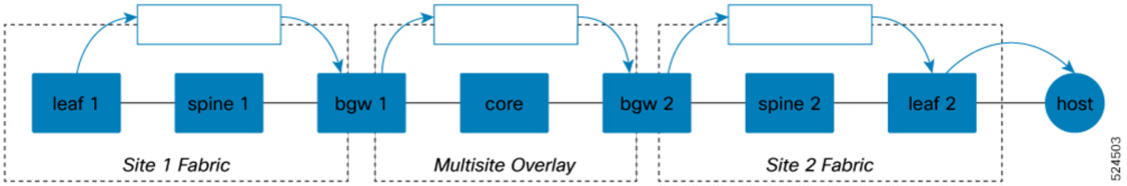
Note When the total hop count to final destination is more than 5, the path trace default TTL value is 5. Use **max-ttl** option to finish VXLAN OAM path trace completely.

For example: `pathtrace nve ip unknown vrf vrf-vni13001 payload ip 200.1.1.71 200.1.1.23 payload-end verbose max-ttl 10`

For Single site and multisite, there will be difference in pathtrace and traceroute behaviour and the the output may vary in different scenarios as mentioned below. It's important to understand the distinction between using NVE traceroute and pathtrace. Pathtrace reveals more nodes because it requires each node to support NGOAM, limiting the trace to the extent of a Cisco N9K VXLAN fabric. In contrast, NVE traceroute can trace beyond the VXLAN fabric into the IP network, as it is RFC compliant and can trace the IP network of any vendor.

- **Traceroute (IP)**: As the diagram shows, the multiple probes are sent with TTL expiry or ACL hits, where:
  - The arrows pointing to a node indicate which hops the trace will appear to hit (you will see a line in the trace output from those nodes).

- The arrows pointing into a pipe represent the packet being encapsulated in VXLAN. When encapsulated, you won't see responses from nodes until it drops out of the pipe, as encapsulation adds a higher TTL to the outer packet, meaning that the TTL expiry upon which a traceroute depends doesn't occur inside the pipe.

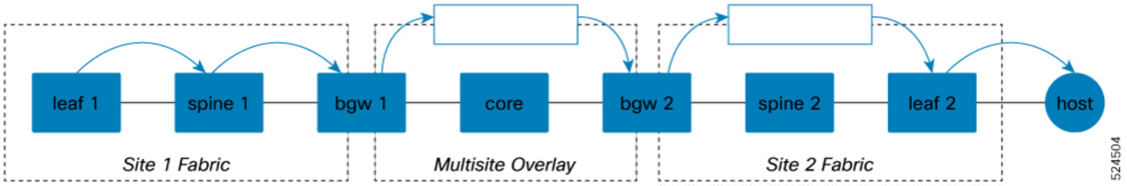


The process described for the IP traceroute involves the following steps:

1. A regular UDP packet is initiated and encapsulated in VXLAN at the leaf switch.
2. The packet travels through the network and is decapsulated at the Border Gateway (BGW) of Site 1.
3. The BGW of Site 1 receives the packet and sends a response.
4. The packet is then re-encapsulated at BGW 1 and continues its journey through the network.
5. The packet exits the tunnel at BGW 2, where another response is received.
6. The packet is encapsulated once more and exits at a leaf in Site 2, prompting another response.
7. Finally, the packet reaches the leaf, and the final response is seen.

This sequence ensures that the packet is properly encapsulated and decapsulated as it traverses through different sites and network components, allowing for accurate tracing of the packet's path.

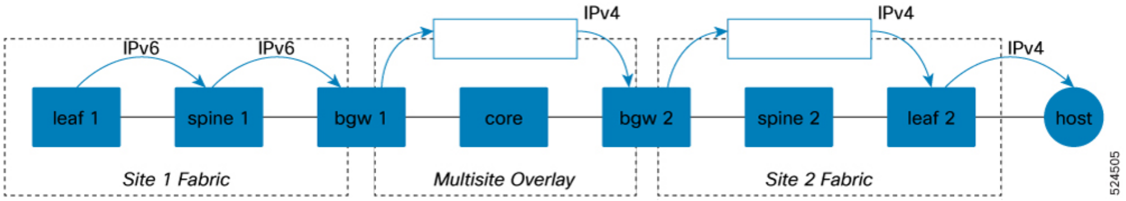
- **Traceroute (NVE):** As the diagram shows, in an NVE traceroute, NGOAM is intelligent enough to recognize that it is being generated from a VTEP. Consequently, it will first trace the underlay up to the remote VTEP that the destination is beyond. From there, it will switch to an overlay traceroute, which will function similarly to an IP traceroute. The NGOAM channel used here employs plain UDP and ICMP (UDP requests for the underlay, followed by ICMP requests for the overlay portion after the remote VTEP). Due to this advanced functionality, the probes in the local fabric will not be encapsulated in VXLAN and will not enter a pipe, allowing us to see the nodes in the local fabric.



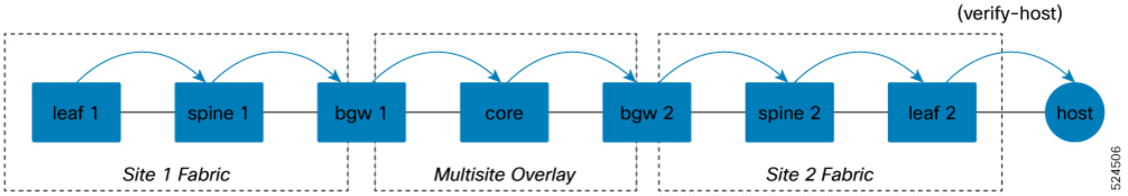
Once the probes reach the remote VTEP, they are encapsulated in VXLAN to continue the probing. After the local BGW, the output resembles a normal IP traceroute as the probes enter the pipes in the multi-site and site2 fabrics. This hybrid underlay and overlay tracing explains why we see a mix of IPv6 and IPv4 responses.

Let's examine the NVE traceroute more closely, considering both the underlay and overlay:

- **Traceroute (NVE – IPv4 over IPv6)**: Since the local underlay fabric is IPv6, NGOAM generates probes as IPv6 within the local fabric. You receive IPv6 responses from the local spine and BGW. However, once the trace reaches the BGW, NGOAM switches to overlay tracing. As the overlay is IPv4, you will receive IPv4 responses from the visible nodes beyond the BGW, as the packet effectively becomes IPv4.



- **Pathtrace**: As shown in the diagram, pathtrace will generate a response from each node in the fabric. It uses a different channel (NVO3), which allows VXLAN-capable nodes in the fabric to process the packet specially due to ACL hits rather than TTL expiry. This makes it more capable of capturing the packet for processing, as long as the node supports NGOAM. Additionally, pathtrace receives special handling by NGOAM on the BGW, which adjusts the probe to allow it to continue into the next fabric.



Configuring NGOAM Profile

Complete the following steps to configure NGOAM profile.

SUMMARY STEPS

1. switch(config)# **[no] feature ngoam**
2. switch(config)# **[no] ngoam profile <profile-id>**
3. switch(config-ng-oam-profile)# **?**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **[no] feature ngoam**	Enables or disables NGOAM feature
Step 2	switch(config)# **[no] ngoam profile <profile-id>**	Configures OAM profile. The range for the profile-id is <1 – 1023>. This command does not have a default value.

	Command or Action	Purpose
		Enters the **config-ngoam-profile submode** to configure NGOAM specific commands. **Note** All profiles have default values and the **show run all** CLI command displays them. The default values are not visible through the **show run** CLI command.
Step 3	switch(config-ng-oam-profile)# **?** **Example:** switch(config-ng-oam-profile)# ? description Configure description of the profile dot1q Encapsulation dot1q/bd flow Configure ngoam flow hop Configure ngoam hop count interface Configure ngoam egress interface no Negate a command or set its defaults oam-channel Oam-channel used payload Configure ngoam payload sport Configure ngoam Udp source port range	Displays the options for configuring NGOAM profile.

Example

See the following examples for configuring an NGOAM profile and for configuring NGOAM flow.

```
switch(config)#
ngoam profile 1
oam-channel 2
flow forward
payload pad 0x2
sport 12345, 54321

switch(config-ngoam-profile)#flow {forward }
Enters config-ngoam-profile-flow submode to configure forward flow entropy specific
information
```

See the following examples for configuring an Oam channel 2, ping and pathtrace using NGOAM oam channel.

```
switch(config)#
ngoam profile 1
oam-channel 2

!Ping nve using oam channel 2
ping nve ip 100.100.100.1 profile 1 vni 201011 verbose count 5

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response,
'v' - Other - Use verbose to see the result

Sender handle: 26
! size 300,Reply from Node-01 (100.100.100.1),time = 7 ms
 Pkt sent on sport = 61273
! size 300,Reply from Node-01 (100.100.100.1),time = 6 ms
 Pkt sent on sport = 61273
! size 300,Reply from Node-01 (100.100.100.1),time = 6 ms
 Pkt sent on sport = 61273
! size 300,Reply from Node-01 (100.100.100.1),time = 6 ms
 Pkt sent on sport = 61273
! size 300,Reply from Node-01 (100.100.100.1),time = 6 ms
 Pkt sent on sport = 61273

Sent 5, Received 5, Success rate is 100 percent Round-trip min/avg/max = 6/6/7 ms
Total time elapsed 115 ms

!Pathtrace nve using oam channel 2
pathtrace nve ip 100.100.100.1 vni 201011 verbose

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'D' - Destination Unreachable, 'X' - unknown return code,
'm' - malformed request(parameter problem),
'c' - Corrupted Data/Test, '#' - Duplicate response,
'v' - Other - Use verbose to see the result

Path trace Request to peer ip 100.100.100.1 source ip 100.100.100.2
Sender handle: 132

Hop   Code   ReplyIP   IngressI/f  EgressI/f   State
======================================================
  1 !Reply from 2.3.1.2, (Node-03) Eth1/53/1  Eth1/43  UP / UP
  2 !Reply from 100.100.100.1, (Node-01) Eth1/43  Unknown  UP / DOWN
```

ConfiguringConfigure NGOAM Southbound Loop Detection on Layer-2 Interfaces

Follow these steps to configure NGOAM Southbound loop detection and mitigation.

Before you begin

Enable the NGOAM feature.

Use the following command to create space for the TCAM ing-sup region:

```
hardware access-list tcam region ing-sup 768
```

Note

- Ensure that additional TCAM entries are freed up before increasing the allocation for the ing-sup region.
- Configuring the TCAM region requires the node to be rebooted.

Procedure

Step 1 Run the [**no**] **ngoam loop-detection** command in global configuration mode, to enable NGOAM Southbound loop detection and mitigation for all VLANs or ports.

Example:

```
switch# configure terminal
switch(config)# ngoam loop-detection
switch(config-ng-oam-loop-detection)#
```

This feature is disabled by default.

The **no** form of this command disable the NGOAM Southbound loop detection and mitigation.

Step 2 (Optional) Run the [**no**] **disable** {**vlan** *vlan-range*} [**port** *port-range*] command to disable NGOAM Southbound loop detection and mitigation for specific VLANs or ports and brings up any loop-detected ports.

Example:

Disables on specific VLAN ports:

```
switch(config-ng-oam-loop-detection)# disable vlan 1200 port ethernet 1/1
```

Disables on specific VLANs:

```
switch(config-ng-oam-loop-detection)# disable vlan 1300
```

The **no** form of this command resumes active monitoring of these VLANs or ports.

Step 3 (Optional) Run the [**no**] **periodic-probe-interval** *value* command to specify how often periodic loop-detection probes are sent.

Example:

```
switch(config-ng-oam-loop-detection)# periodic-probe-interval 200
```

Range: 60 seconds to 3600 seconds (60 minutes). Default: 300 seconds (5 minutes).

Step 4 (Optional) Run the [**no**] **port-recovery-interval** *value* command to specify how often recovery probes are sent when a port or VLAN is shut down.

Example:

```
switch(config-ng-oam-loop-detection)# port-recovery-interval 300
```

Range: 300 seconds to 3600 seconds (60 minutes). Default value: 600 seconds (10 minutes).

Step 5 (Optional) Run the **show ngoam loop-detection summary** command to verify the loop-detection configuration and current loop summary.

Example:

```
switch# show ngoam loop-detection summary
Loop detection:enabled
Periodic probe interval: 200
Port recovery interval: 300
Number of vlans: 1
Number of ports: 1
Number of loops: 1
Number of ports blocked: 1
Number of vlans disabled: 0
Number of ports disabled: 0
```

```
Total number of probes sent: 214
Total number of probes received: 102
Next probe window start: Thu May 14 15:14:23 2020 (0 seconds)
Next recovery window start: Thu May 14 15:54:23 2020 (126 seconds)
```

What to do next

Configure a QoS policy on the spine. (For configuration example, see Configuration Examples for NGOAM Southbound Loop Detection and Mitigation, on page 23).

Configuring NGOAM Southbound Loop Detection on Layer-3 Interfaces

Follow these steps to enable NGOAM Southbound Loop Detection on Ethernet and L3 port-channel interfaces.

Before you begin

Enable the NGOAM feature.

Use the following command to create space for the TCAM ing-sup region:

```
hardware access-list tcam region ing-sup 768
```

Note

- Ensure that additional TCAM entries are freed up before increasing the allocation for the ing-sup region.
- Configuring the TCAM region requires the node to be rebooted.

Procedure

Step 1 **configure terminal**

Example:

```
switch# config terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Run the [**no**] **ngoam loop-detection** command in global configuration mode, to enable NGOAM Southbound loop detection and mitigation for all VLANs or ports.

Example:

```
switch# config terminal
switch(config)# ngoam loop-detection
switch(config-ng-oam-loop-detection)#
```

This feature is disabled by default.

Step 3 Run the [**no**] **l3 ethernet port** *port-range*command to enable the L3 loop-detection on ethernet interfaces.

Example:

```
switch(config-ng-oam-loop-detection)# l3 ethernet port Eth1/49
```

Use the **no** form of this command to disable the L3 loop-detection on ethernet interfaces.

Step 4 Run the [**no**] **l3 port-channel port** *port-range* command to enable the L3 loop-detection on port-channel interfaces.

Example:

```
switch(config-ng-oam-loop-detection)# l3 port-channel port port-channel1
```

Use the **no** form of this command to disable the L3 loop-detection on port-channel interfaces.

Step 5 (Optional) Run the **show ngoam loop-detection status l3** command to verify the loops detected on L3 interfaces.

Example:

```
switch# show ngoam loop-detection status l3
Port       Status       NumLoops      DetectionTime             ClearedTime
==========================================================================
 Eth1/2     BLOCKED      2             Tue Jun 25 02:38:54 2024  Tue Jun 25 02:38:52 2024
```

Step 6 (Optional) Run the **show run ngoam** command to verify the loop-detection configuration and current loop summary.

Example:

```
switch# show run ngoam
ngoam loop-detection
  periodic-probe-interval 60
  port-recovery-interval 600
  l3 ethernet port Ethernet1/1-3
!
2024 Jun 25 02:37:50 switch %ETHPORT-5-IF_DOWN_NONE: Interface Ethernet1/2 is down (None)
2024 Jun 25 02:37:50 switch %ETHPORT-5-IF_DOWN_ERROR_DISABLED: Interface Ethernet1/2 is down (Error
 disabled. Reason:error)
2024 Jun 25 02:38:52 switch %ETHPORT-5-IF_ERRDIS_RECOVERY: Interface Ethernet1/2 is being recovered
 from error disabled state (Last Reason:error)
2024 Jun 25 02:38:54 switch %ETHPORT-5-IF_DOWN_ERROR_DISABLED: Interface Ethernet1/2 is down (Error
 disabled. Reason:error)
```

Detecting Loops and Bringing Up Ports On Demand

Follow the steps in this section to detect loops or bring up blocked ports on demand.

Procedure

Step 1 (Optional) Run the **ngoam loop-detection probe** {**vlan** *vlan-range*} [**port** *port-range*] command to sends a loop-detection probe on the specified VLAN or port.

Example:

```
switch# ngoam loop-detection probe vlan 1200 port ethernet 1/1
```

This command also send a notification to check whether the probe was successfully sent.

Step 2 (Optional) Run the **ngoam loop-detection bringup** {**vlan** *vlan-range*} [**port** *port-range*] command to bring UP the VLANs or ports that were blocked earlier.

Example:

```
switch# ngoam loop-detection bringup vlan 1200 port ethernet 1/1
```

This command also clears any entries stuck in the NGOAM.

Note
It can take up to two port-recovery intervals for the ports to come up after a loop is cleared. You can speed up the recovery by manually overriding the timer with the **ngoam loop-detection bringup vlan** {**vlan** *vlan-range*} [**port** *port-range*] command.

Step 3 (Optional) Run the **show ngoam loop-detection status** [**history**] [**vlan** *vlan-range*] [**port** *port-range*] command to verify the loop-detection status for the VLAN or port with and without the **history** option..

Example:

Without **history** option

```
switch# show ngoam loop-detection status
VlanId Port   Status     NumLoops  Detection Time                 ClearedTime
====== ====== ========== ========= ============================== ================
100    Eth1/3 BLOCKED     1        Tue Apr 14 20:07:50.313 2020  Never
```

With **history** option

```
switch# show ngoam loop-detection status history
VlanId Port   Status     NumLoops  Detection Time                 ClearedTime
====== ====== ========== ========= ============================== ================
100    Eth1/3 BLOCKED     1        Tue Apr 14 20:07:50.313 2020  Never
200    Eth1/2 FORWARDING  1        Tue Apr 14 21:19:52.215 2020  May 11 21:30:54.830 2020
```

The status can be one of the following:

- **BLOCKED**: The VLAN or port is shut down because a loop has been detected.
- **FORWARDING**: A loop has not been detected, and the VLAN or port is operational.
- **RECOVERING**: Recovery probes are being sent to determine if a previously detected loop still exists.

The **history** option displays blocked, forwarding, and recovering ports. Without the **history** option, the command displays only blocked and recovering ports.

Configuration Examples for NGOAM Southbound Loop Detection and Mitigation

The following example hows to configure a QoS policy on the spine and apply it to all of the spine interfaces to which the loop-detection-enabled leaf is connected:

```
class-map type qos match-any Spine-DSCP56
match dscp 56
policy-map type qos Spine-DSCP56
class Spine-DSCP56
set qos-group 7

interface Ethernet1/31
mtu 9216
```

```
no link dfe adaptive-tuning
service-policy type qos input Spine-DSCP5663
no ip redirects
ip address 27.4.1.2/24
ip router ospf 200 area 0.0.0.0
ip pim sparse-mode
no shutdown
```

The following sample output shows the loop-detection configuration and current loop summary:

```
switch# show ngoam loop-detection summary
Loop detection:enabled
Periodic probe interval: 200
Port recovery interval: 300
Number of vlans: 1
Number of ports: 1
Number of loops: 1
Number of ports blocked: 1
Number of vlans disabled: 0
Number of ports disabled: 0
Total number of probes sent: 214
Total number of probes received: 102
Next probe window start: Thu May 14 15:14:23 2020 (0 seconds)
Next recovery window start: Thu May 14 15:54:23 2020 (126 seconds)
```

The following sample output shows the loop-detection status for the specified VLANs or ports with and without the **history** option:

```
switch# show ngoam loop-detection status
VlanId Port   Status     NumLoops  Detection Time                ClearedTime
====== ====== ========== ========= ============================= ===============
100    Eth1/3 BLOCKED     1        Tue Apr 14 20:07:50.313 2020  Never

switch# show ngoam loop-detection status history
VlanId Port   Status     NumLoops  Detection Time                ClearedTime
====== ====== ========== ========= ============================= ===============
100    Eth1/3 BLOCKED     1        Tue Apr 14 20:07:50.313 2020  Never
200    Eth1/2 FORWARDING  1        Tue Apr 14 21:19:52.215 2020  May 11 21:30:54.830 2020
```