
cisco Live!
Let's go

The bridge to possible

Troubleshooting EVPN/VXLAN on Catalyst Catalyst 9000 IOS-XE

Dmytro Vishchuk, Technical Leader
Ivan Shirshin, Technical Leader

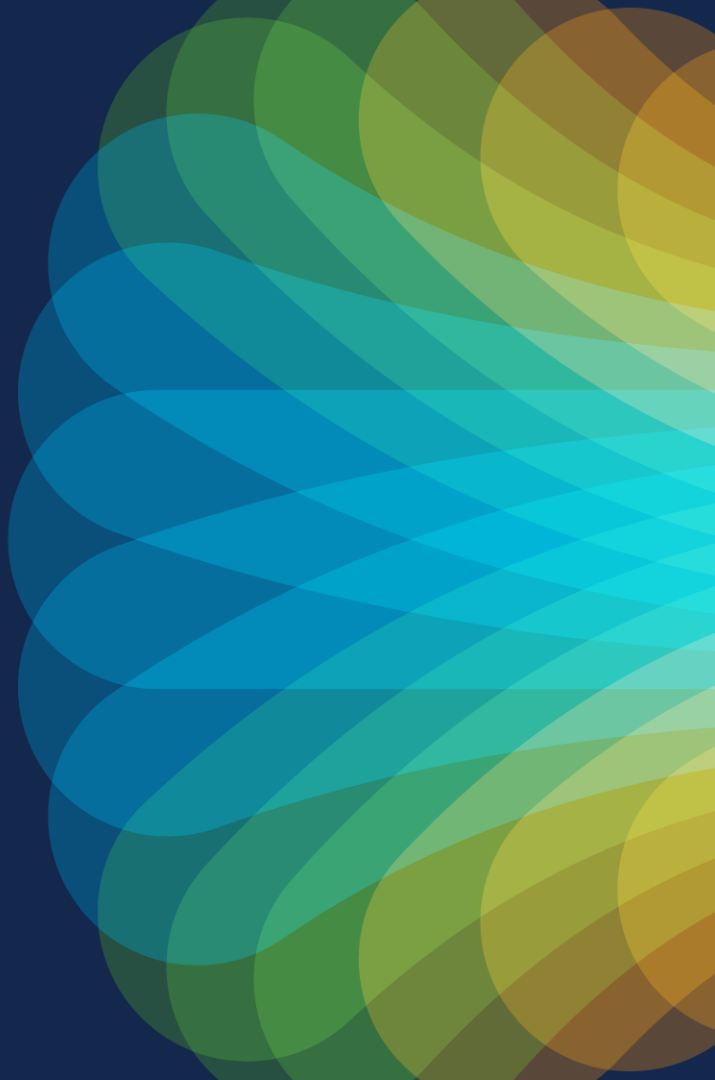
cisco Live!

LTRTRS-2556

Agenda

- Introduction
- EVPN Components and Troubleshooting
- POD Overview
- Troubleshooting Tools in the Lab
- Documentation Overview
- Time Management
- Conclusion

Introduction



Your proctors

Ivan Shirshin

Technical Leader
CX Technical Services (TAC)

Areas of expertise :

- Routing Protocols
- Data Center
- Intent-Based Networking
- EVPN

Dmytro Vishchuk

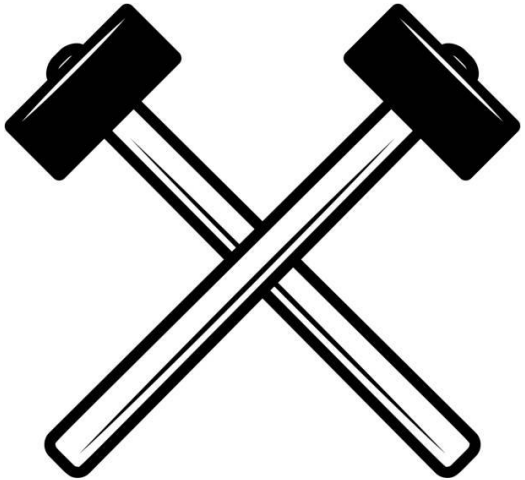
Technical Leader
CX Technical Services (TAC)

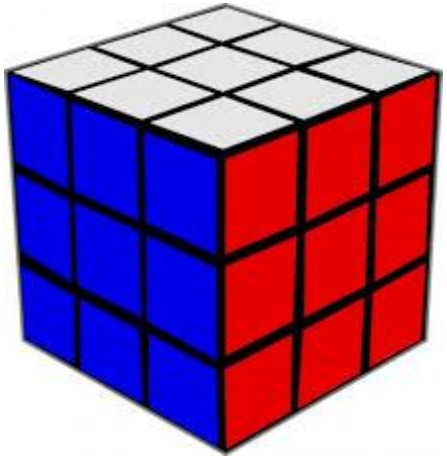
Areas of expertise :

- Routing Protocols
- Service Provider
- Intent-Based Networking
- EVPN

Lab Overview

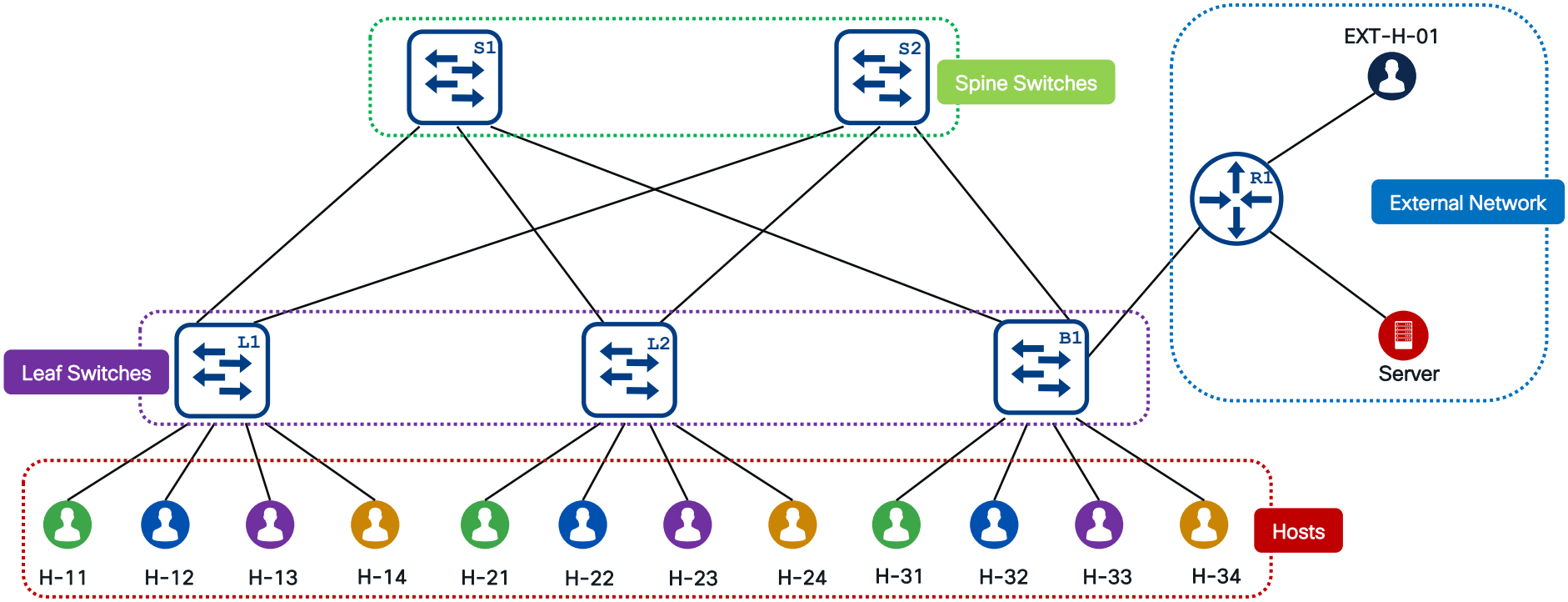
Tasks based on real TAC experience ⟺ 10 challenges (not all are mandatory)



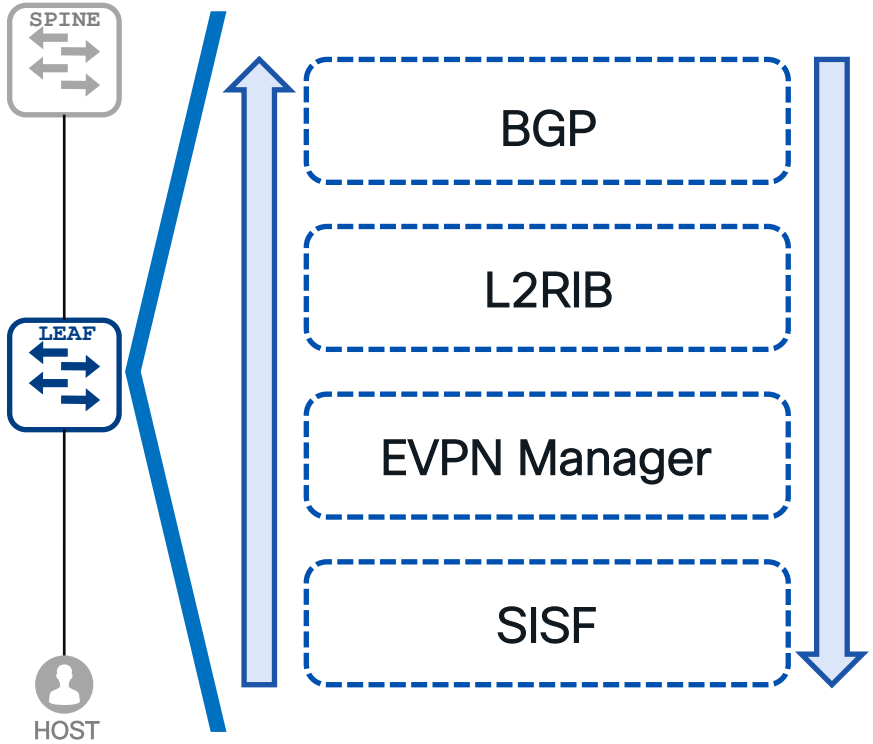


EVPN Components and Troubleshooting

EVPN Fabric and Device Roles



EVPN – IOS-XE Components Interaction



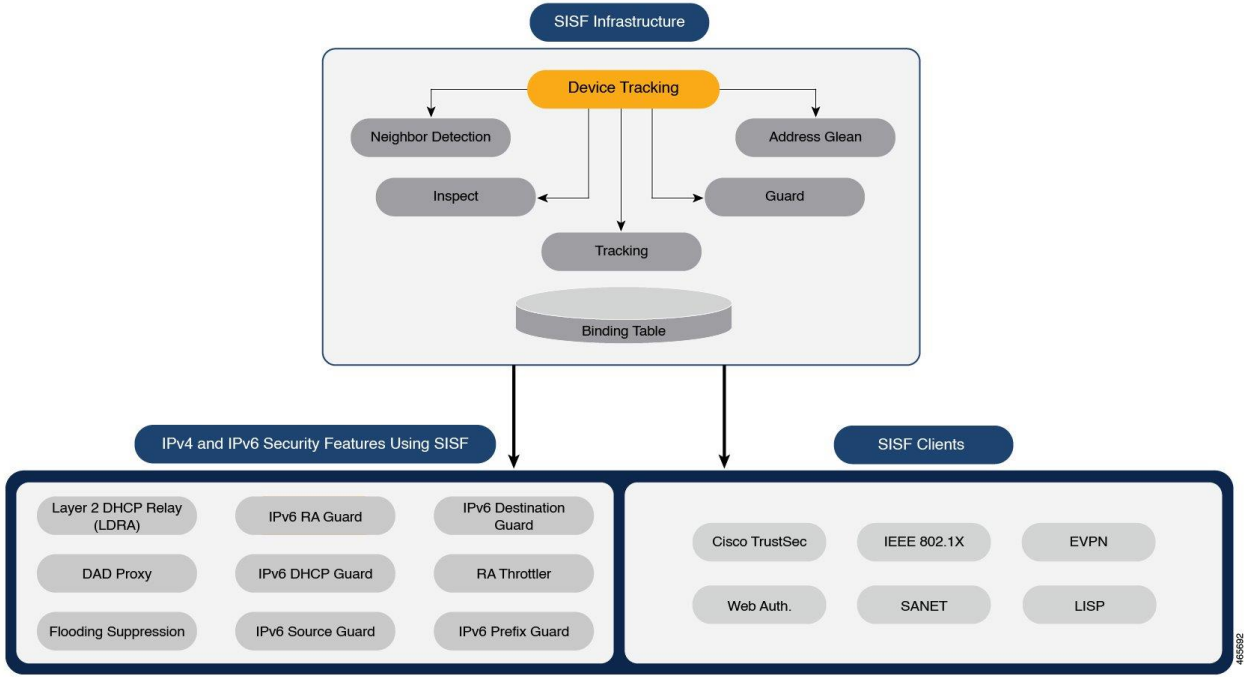
Border Gateway Protocol

SW component L2 Routing Information Base

SW component EVPN Manager

Switch Integrated Security Features (SISF)

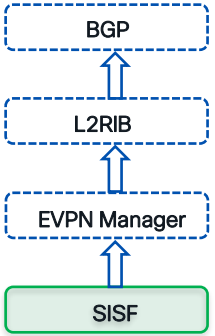
SISF (Switch Integrated Security Features)



SISF (Switch Integrated Security Features)

```
Leaf-01#sh device-tracking database mac 5254.001d.a387 interface gi1/0/7 vlanid 103
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol, DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other
Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


    Network Layer Address                    Link Layer Address     Interface  vlan       prlvl      age        state      Time left
macDB has 1 entries for mac 5254.001d.a387,vlan 103, 1 dynamic
ARP 10.1.103.11                              5254.001d.a387         Gi1/0/7    103        0005       5s         REACHABLE  299 s
```

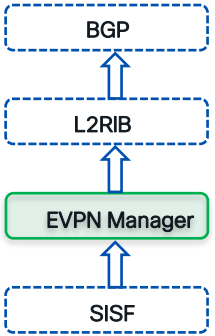
```
# debug device-tracking
# debug device-tracking parser interface gi1/0/1
```



EVPN Manager

```
Leaf-01#show l2vpn evpn mac ip evi 103
IP Address                              EVI    VLAN   MAC Address    Next Hop(s)
--------------------------------------- ------ ------ -------------- -----------
10.1.103.11                             103    103    5254.001d.a387 Gi1/0/7:103
FE80::5054:FF:FE0A:4D5E                 103    103    5254.000a.4d5e 172.16.254.5
FE80::5054:FF:FE1D:B679                 103    103    5254.001d.b679 Gi1/0/7:103
10.1.103.12                             103    103    5254.0018.9e9e 172.16.254.4
```

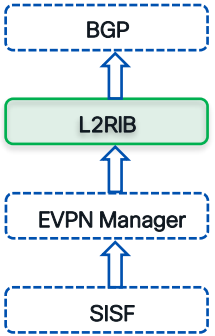
```
# conf t
(config)# monitor event-trace evpn event include detail
(config)# monitor event-trace evpn event size 1000000
(config)# end

# show monitor event-trace evpn event all
# show monitor event-trace evpn error all

# debug l2vpn evpn event
# debug l2vpn evpn event detail
```



L2RIB (L2 Routing Information Base)

```
Leaf-01#show l2route evpn mac ip topology 103
  EVI       ETag  Prod    Mac Address                    Host IP                                 Next Hop(s)
----- ---------- ----- -------------- ------------------------------------- ------------------------------------------
  103          0   BGP 5254.000a.4d5e                FE80::5054:FF:FE0A:4D5E                 V:10103 172.16.254.5
  103          0 L2VPN 5254.001d.a387                          10.1.103.11                             Gi1/0/7:103
  103          0 L2VPN 5254.001d.b679                FE80::5054:FF:FE1D:B679                         Gi1/0/7:103
  103          0   BGP 5254.001b.34f1                          10.1.103.12                 V:10103 172.16.254.4
```

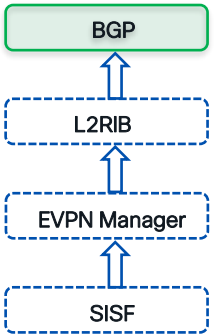
```
# conf t
(config)# monitor event-trace l2rib event include detail
(config)# monitor event-trace l2rib event size 1000000
(config)# end

# show monitor event-trace l2rib event all
# show monitor event-trace l2rib error all

# debug l2rib event
# debug l2rib event detail
```



BGP (Border Gateway Protocol)

```
Leaf-02#show bgp l2vpn evpn 10.1.103.11/32
BGP routing table entry for [2][172.16.254.3:103][0][48][5254001DA387][32][10.1.103.11]/24, version 48874
Paths: (2 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 18
  Local
    172.16.254.3 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10103
      Extended Community: RT:65001:103 ENCAP:8
      Originator: 172.16.255.3, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      Updated on Jan 10 2024 16:26:24 UTC
```

```
# debug bgp l2vpn evpn evi context detail
# debug bgp l2vpn evpn evi event detail
# debug bgp l2vpn evpn nve detail
# debug bgp l2vpn evpn events
# debug bgp l2vpn evpn updates
```



NVE verification

```
Leaf-01#show nve peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Interface  VNI      Type Peer-IP          RMAC/Num_RTs   eVNI     state flags UP time
```

```
Leaf-01#show bgp l2vpn evpn all summ
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 25, main routing table version 25
8 networks peaked at 12:01:00 Aug 28 2023 UTC (00:14:43.566 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.255.1    4        65001      26      20       25    0    0 00:15:42        0
172.16.255.2    4        65001      29      22       25    0    0 00:15:45        0
```

No BGP routes

** NVE peering is built based on BGP information*

NVE verification

```
Leaf-01#show nve peers vni 10102
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Interface  VNI      Type Peer-IP          RMAC/Num_RTs    eVNI      state flags UP time
nve1       10102    L2CP 172.16.254.4     7               10102      UP   N/A   1w1d
```

ARP Suppression

draft-ietf-bess-evpn-proxy-arp-nd-16

As a high level example, the operation of the EVPN Proxy-ARP/ND function in the network of Figure 1 is described below. In this example we assume IP1, IP2 and IP3 are IPv4 addresses:

1. Proxy-ARP/ND is enabled in BD1 of PE1, PE2 and PE3.

2. The PEs start adding dynamic, static and EVPN-learned entries to their Proxy tables:

   A. PE3 adds IP1->M1 and IP2->M2 based on the EVPN routes received from PE1 and PE2. Those entries were previously learned as dynamic entries in PE1 and PE2 respectively, and advertised in BGP EVPN.
   B. PE3 adds IP4->M4 as dynamic. This entry is learned by snooping the corresponding ARP messages sent by CE4.
   C. An operator also provisions the static entry IP3->M3.

3. When CE3 sends an ARP Request asking for the MAC address of IP1, PE3 will:

   A. Intercept the ARP Request and perform a Proxy-ARP lookup for IP1.
   B. If the lookup is successful (as in Figure 1), PE3 will send an ARP Reply with IP1->M1. The ARP Request will not be flooded to the EVPN network or any other local CEs.
   C. If the lookup is not successful, PE3 will flood the ARP Request in the EVPN network and the other local CEs.

3.4. Unicast-forward Sub-Function

As discussed in Section 3.3, in some cases the operator may want to 'unicast-forward' certain ARP-Request and NS messages as opposed to reply to them. The implementation of a 'unicast-forward' function is RECOMMENDED. This option can be enabled with one of the following parameters:

a. unicast-forward always

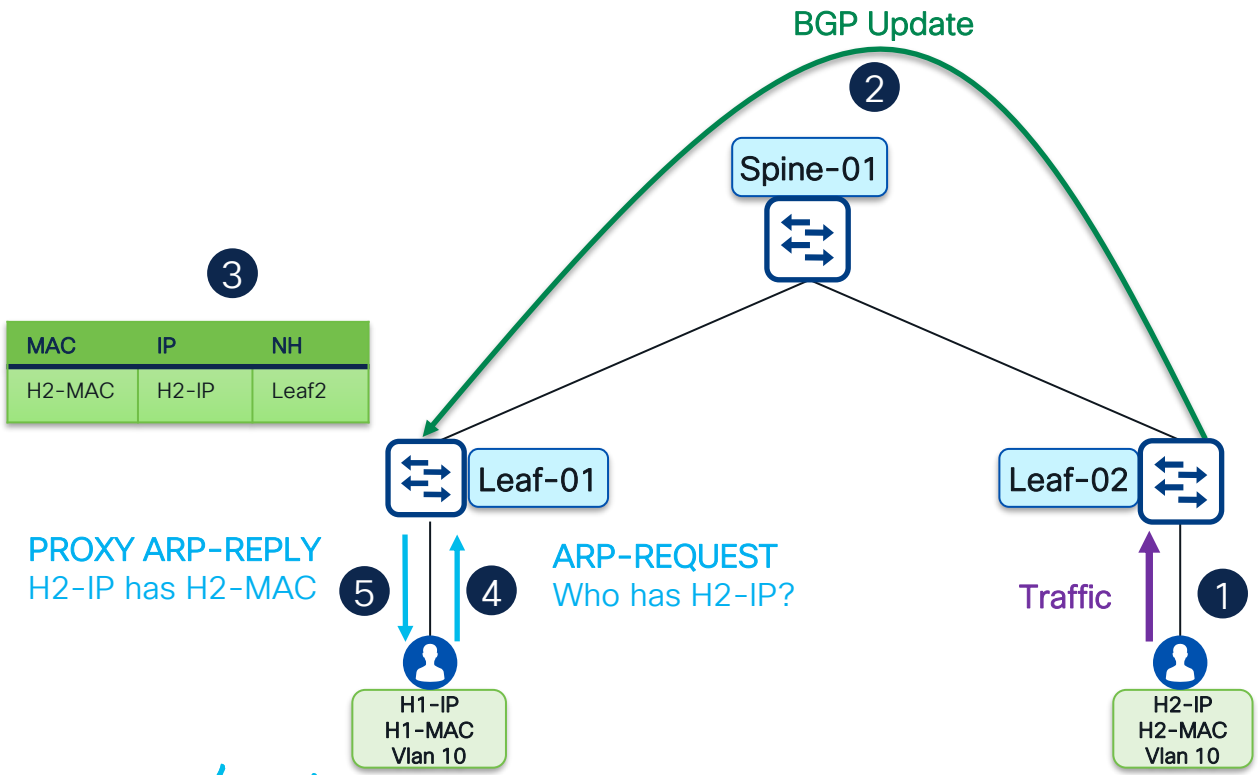
b. unicast-forward unknown-options

If 'unicast-forward always' is enabled, the PE will perform a Proxy-ARP/ND table lookup and in case of a hit, the PE will forward the packet to the owner of the MAC found in the Proxy-ARP/ND table. This is irrespective of the options carried in the ARP/ND packet. This option provides total transparency in the BD and yet reduces the amount of flooding significantly.

If 'unicast-forward unknown-options' is enabled, upon a successful Proxy-ARP/ND lookup, the PE will perform a 'unicast-forward' action only if the ARP-Request or NS messages carry unknown options, as explained in Section 3.3. The 'unicast-forward unknown-options' configuration allows the support of new applications using ARP/ND in the BD while still reducing the flooding.

Irrespective of the enabled option, if there is no successful Proxy-ARP/ND lookup, the unknown ARP-Request/NS will be flooded in the context of the BD, as per Section 3.6.

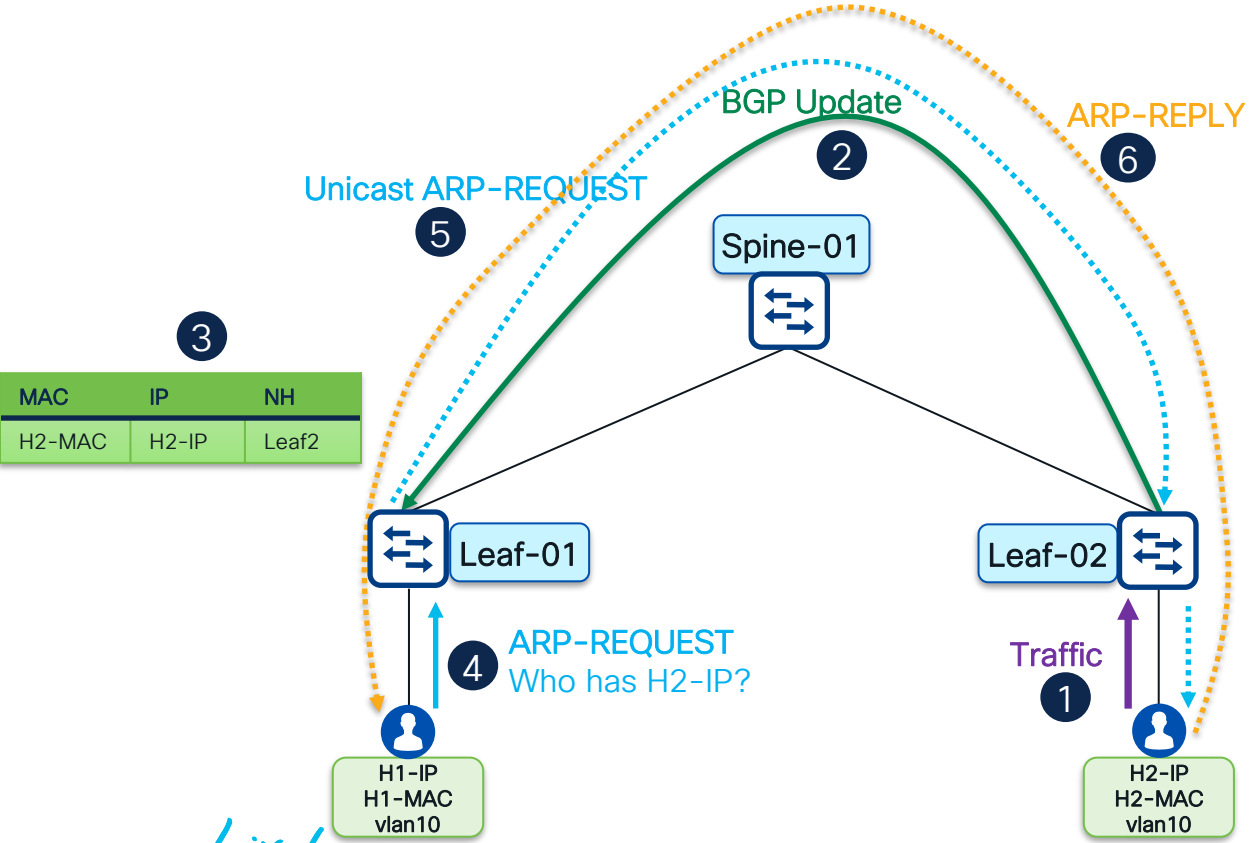
ARP Suppression (proxy-ARP)



1. Traffic from H2 reaches Leaf-02
2. BGP update for H2-IP/H2-MAC propagated over the Fabric
3. H2 entry added to the ARP Proxy table on Leaf-01
4. ARP request received on Leaf-01 from H1
5. Leaf-01 checks the ARP proxy table and sends ARP Reply to H1

- If ARP is known by Leaf, ARP Reply is sent **by Leaf** based on the local cache
- If ARP is unknown by Leaf – **flood** locally and to Fabric

ARP Suppression (Unicast Forwarding)

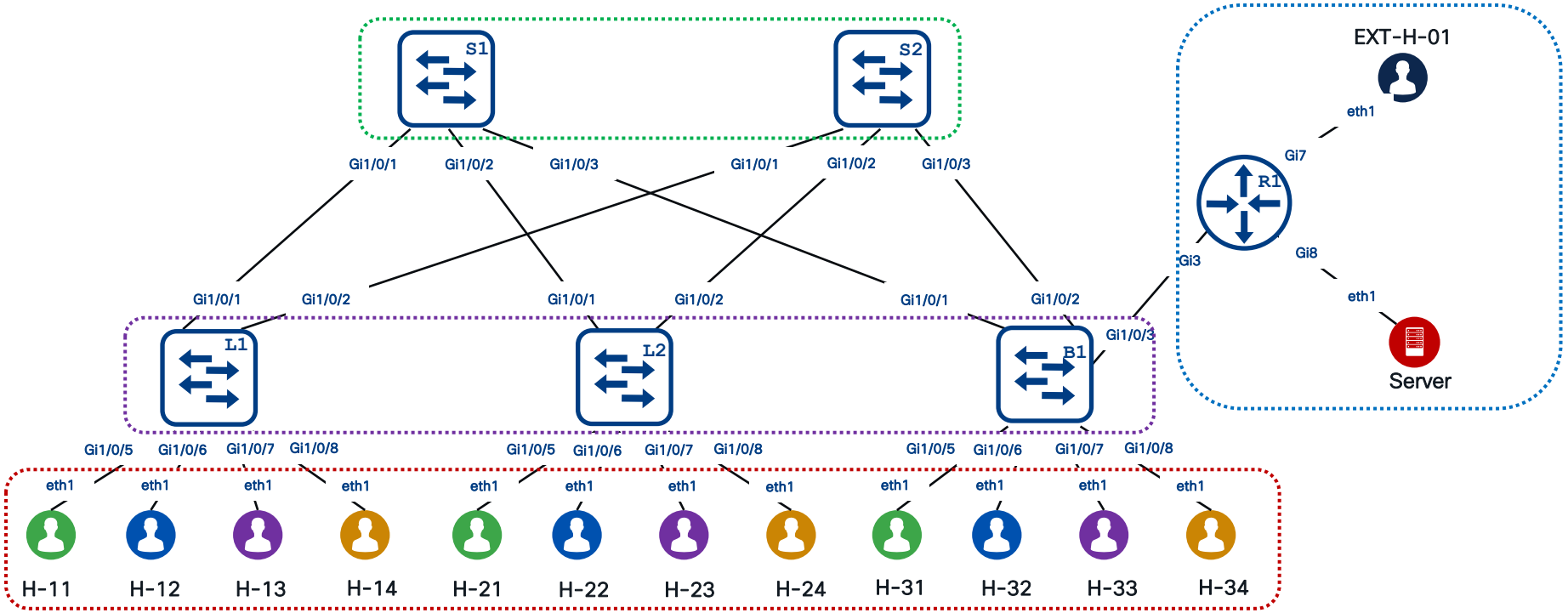


1. Traffic from H2 reaches Leaf-02
2. BGP update for H2-IP/H2-MAC is propagated over the Fabric
3. H2 entry is added to the ARP Proxy table on Leaf-01
4. ARP request received on Leaf-01 from H1
5. Leaf-01 checks the ARP proxy table and sends unicast ARP to H2
6. H2 send ARP reply to H1 directly

- If ARP is known by Leaf, ARP is **unicasted** over the Fabric to the remote host
- ARP cache on the remote host is updated
- If ARP is unknown – **flood** locally and fabric

POD Overview

Physical Topology

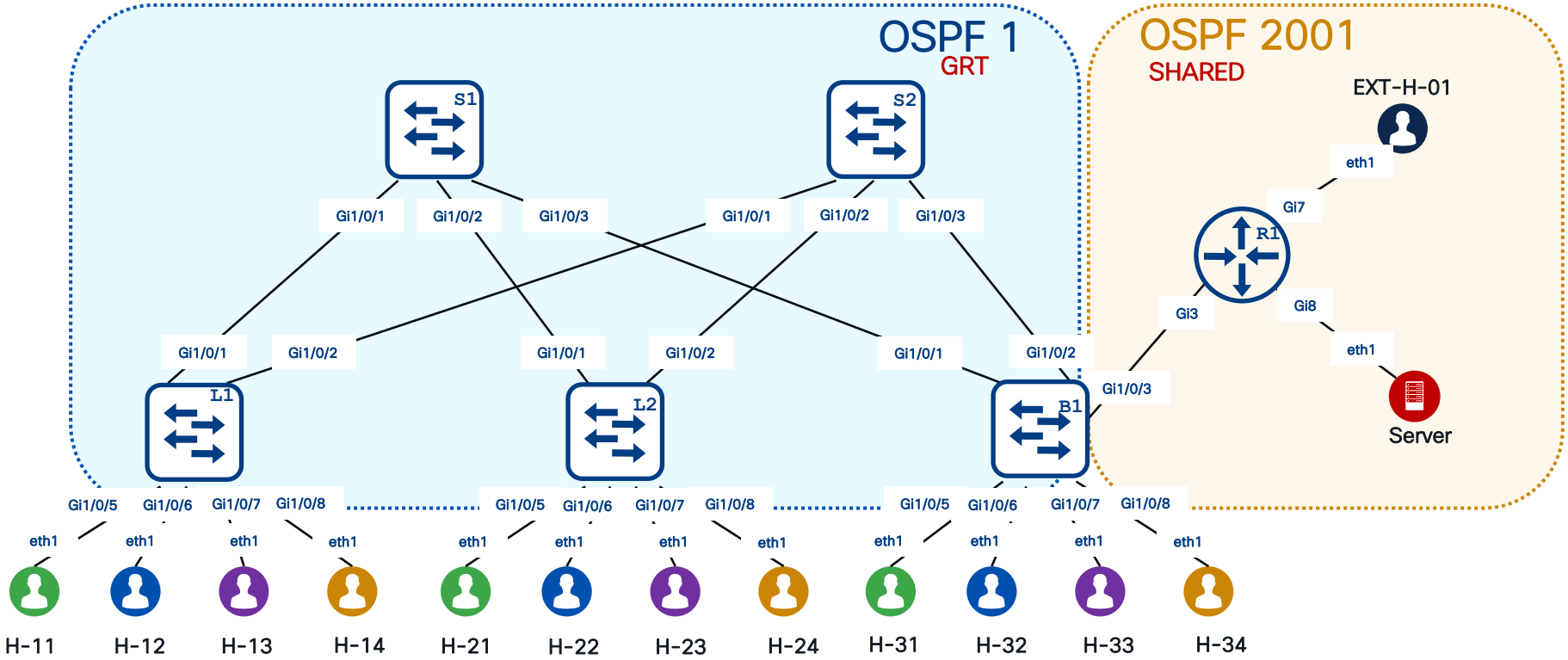


VLANs in the Fabric

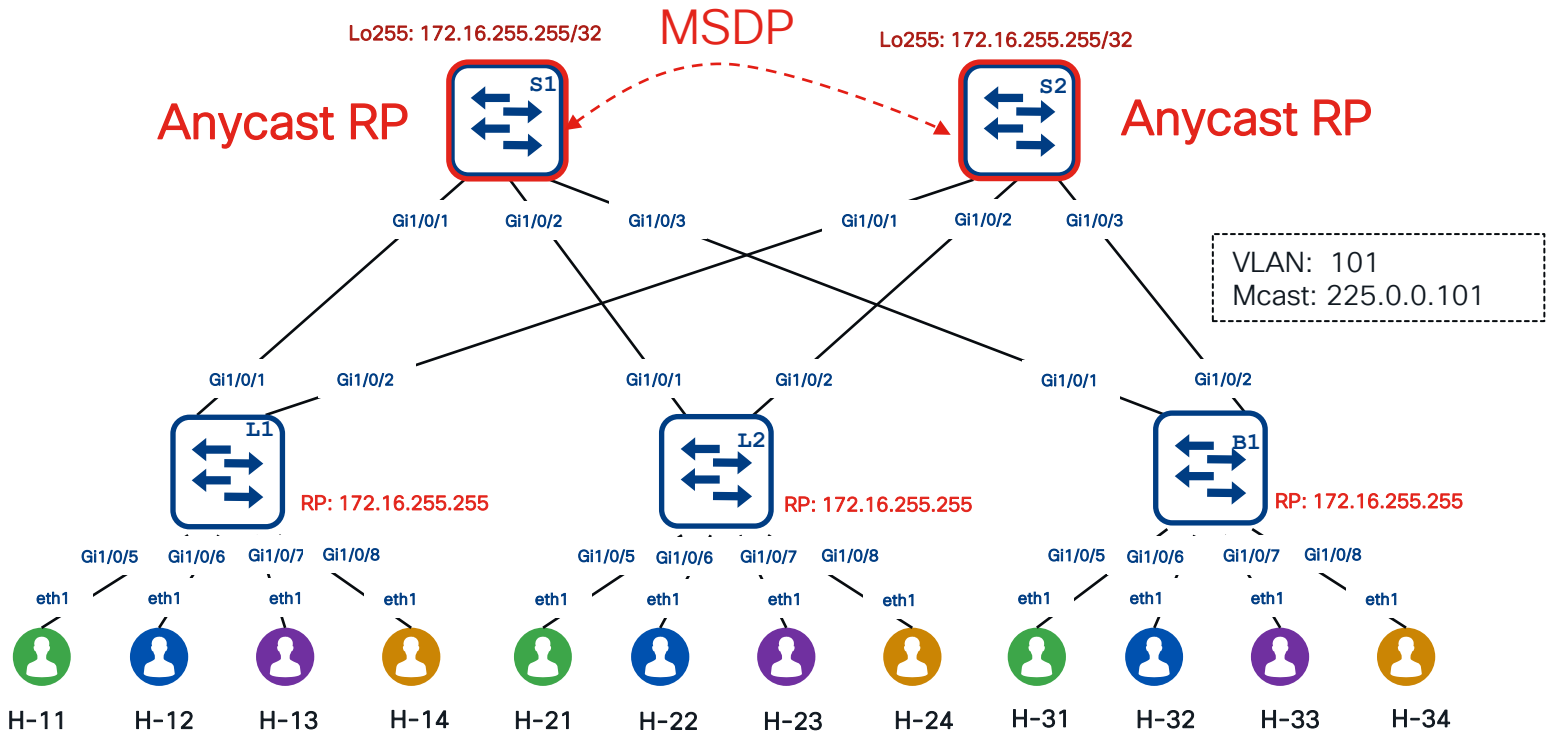



VLAN	101	102	103	104
EVI	101	102	103	104
VNI	10101	10102	10103	10104
BUM	MR	IR	IR	IR
IPv4	10.1.101.0/24	10.1.102.0/24	10.1.103.0/24	10.1.104.0/24
IPv6	2001:10:1:101::/64	2001:10:1:102::/64	2001:10:1:103::/64	2001:10:1:104::/64
Routing	YES	YES	L2 ONLY	YES

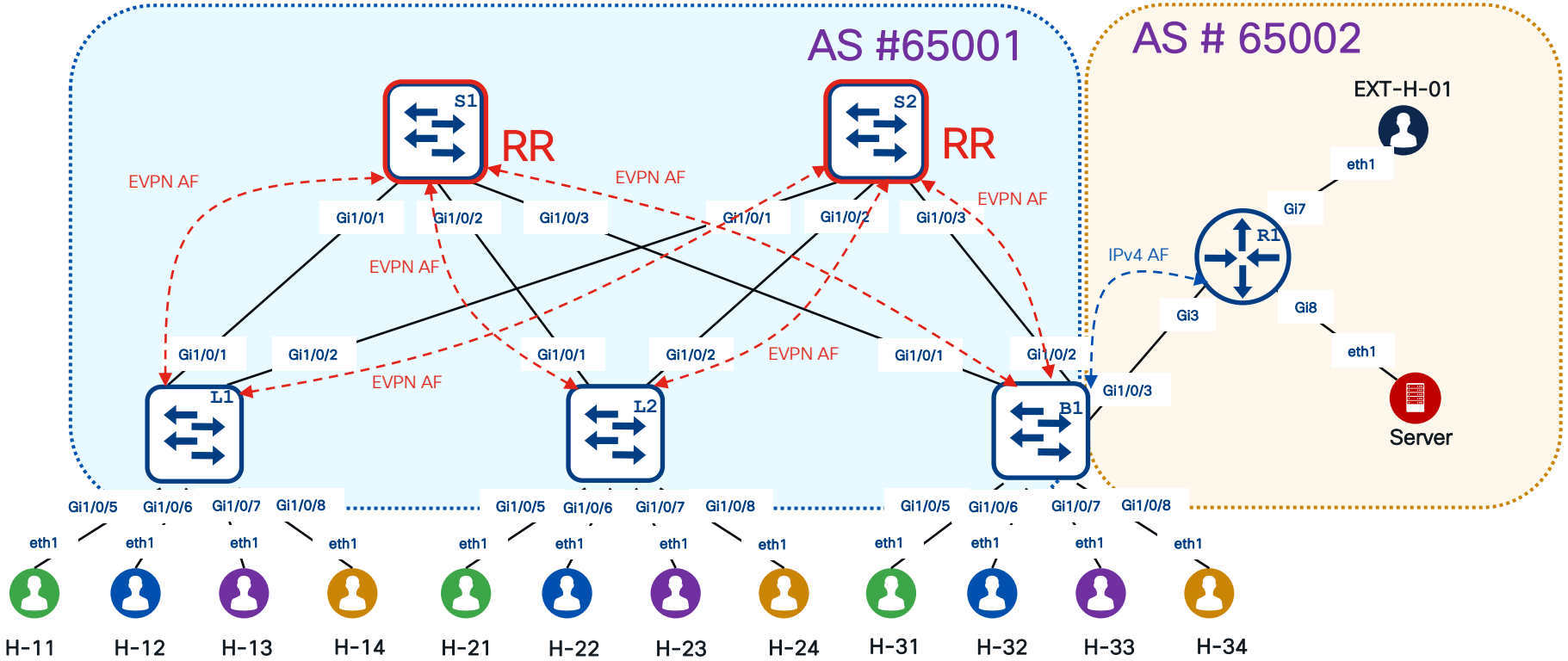
Underlay IGP



Multicast Topology



BGP Topology



VTEP Configuration

```
vrf definition green
rd 1:1
 !
 address-family ipv4
  route-target export 1:1
  route-target import 1:1
  route-target export 1:1 stitching
  route-target import 1:1 stitching
 exit-address-family
!
 address-family ipv6
  route-target export 1:1
  route-target import 1:1
  route-target export 1:1 stitching
  route-target import 1:1 stitching
 exit-address-family
 !
<..skip...>
!
<…continue…>
```

```
l2vpn evpn
 replication-type static
 router-id Loopback1
!
l2vpn evpn instance 101 vlan-based
 encapsulation vxlan
!
l2vpn evpn instance 102 vlan-based
 encapsulation vxlan
 replication-type ingress
!
l2vpn evpn instance 103 vlan-based
 encapsulation vxlan
 replication-type ingress
!
l2vpn evpn instance 104 vlan-based
 encapsulation vxlan
 replication-type ingress
!
vlan configuration 101
 member evpn-instance 101 vni 10101
vlan configuration 102
 member evpn-instance 102 vni 10102
vlan configuration 103
 member evpn-instance 103 vni 10103
vlan configuration 104
 member evpn-instance 104 vni 10104
vlan configuration 901
 member vni 50901
!
<…continue…>
```

```
interface Vlan101
 mac-address 0000.dead.beef
 vrf forwarding green
 ip address 10.1.101.1 255.255.255.0
 ip helper-address global 192.168.20.20
 ipv6 address 2001:10:1:101::1/64
 ipv6 enable
!
interface Vlan102
 mac-address 0000.dead.beef
 vrf forwarding green
 ip address 10.1.102.1 255.255.255.0
 ip helper-address global 192.168.20.20
 ipv6 address 2001:10:1:102::1/64
 ipv6 enable
!
interface Vlan104
 mac-address 0000.dead.beef
 vrf forwarding green
 ip address 10.1.104.1 255.255.255.0
 ipv6 address 2001:10:1:104::1/64
 ipv6 enable
!
interface Vlan901
 vrf forwarding green
 ip unnumbered Loopback1
 no autostate
 ipv6 enable
!
```

VTEP Configuration

```
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp
 member vni 10101 mcast-group 225.0.0.101
 member vni 50901 vrf green
 member vni 10102 ingress-replication
 member vni 10104 ingress-replication
 member vni 10103 ingress-replication
```

```
router bgp 65001
 bgp router-id 172.16.255.3
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 172.16.255.1 remote-as 65001
 neighbor 172.16.255.1 update-source Loopback0
 neighbor 172.16.255.2 remote-as 65001
 neighbor 172.16.255.2 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor 172.16.255.1 activate
  neighbor 172.16.255.1 send-community both
  neighbor 172.16.255.2 activate
  neighbor 172.16.255.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf green
  advertise l2vpn evpn
  redistribute static
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf green
  redistribute connected
  redistribute static
  advertise l2vpn evpn
 exit-address-family
```

Troubleshooting Tools in the Lab

Packets Capture on Catalyst 9000 Switch

```
Leaf-01#mon cap tac int gi1/0/1 in match any start
Started capture point : tac

Leaf-01#mon cap tac stop
Capture statistics collected at software:
        Capture duration - 19 seconds
        Packets received - 17
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared

Stopped capture point : tac

Leaf-01#sh mon cap tac buffer display-filter "arp"
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

 26  11.262827 f4:cf:e2:43:34:c2 -> ff:ff:ff:ff:ff:ff ARP 64 Who has 10.1.102.12? Tell 10.1.102.11
```

Packet Capture on the Router

```
EXT-R-01#mon cap tac int gi8 both match any start
EXT-R-01#mon cap tac stop
Stopped capture point : tac
EXT-R-01#sh mon cap tac buff br
 ----------------------------------------------------------------------------
 #   size   timestamp     source             destination      dscp    protocol
 ----------------------------------------------------------------------------
   0  379    0.000000   172.16.255.5     ->  192.168.20.20    0  BE   UDP
   1  362    0.000992   192.168.20.20    ->  172.16.255.5     0  BE   UDP
   2   70    0.000992   192.168.20.1     ->  192.168.20.20    0  BE   ICMP
<…snip…>
```

```
EXT-R-01#mon cap tac int gi3 efp 2001,2002 both match any start
EXT-R-01#mon cap tac stop
Stopped capture point : tac
EXT-R-01#sh mon cap tac buff br
 ----------------------------------------------------------------------------
 #   size   timestamp     source             destination      dscp    protocol
 ----------------------------------------------------------------------------
   0   68    0.000000  52:54:00:11:A6:BB -> 01:00:0C:CC:CC:CD   --    LLC
   1   68    1.990998  52:54:00:11:A6:BB -> 01:00:0C:CC:CC:CD   --    LLC
   2   68    3.990998  52:54:00:11:A6:BB -> 01:00:0C:CC:CC:CD   --    LLC
<…snip…>
```

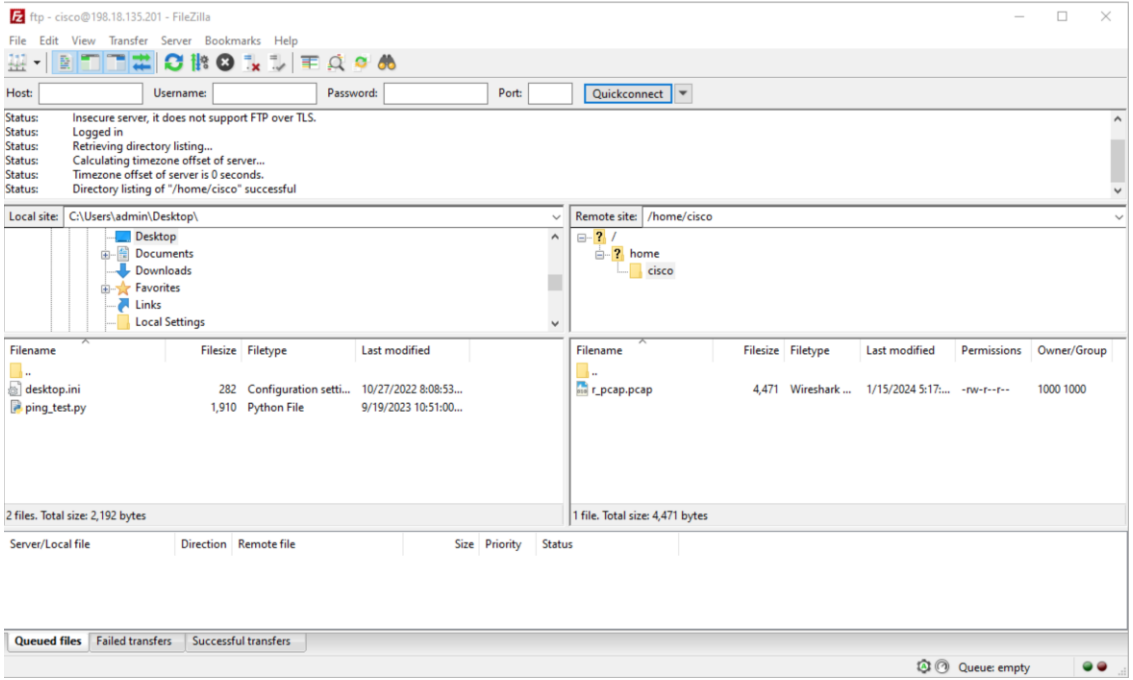
Linux capture

```
H-31:~$ sudo tcpdump -nei eth1 port bootps -v
 <...snip...>
  17:32:18.257703 52:54:00:07:73:c3 > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342: (tos 0x0,
ttl 64, id 0, offset 0, flags [none], proto UDP (17), length 328)
    0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 52:54:00:07:73:c3, length 300, xid
0x1ed9fc73, secs 210, Flags [none]
          Client-Ethernet-Address 52:54:00:07:73:c3
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message (53), length 1: Discover
<…snip…>
```

```
H-31:~$ sudo tcpdump -nei eth1 port bootps

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
12:58:50.757771 52:54:00:07:73:c3 > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342: 0.0.0.0.68 >
255.255.255.255.67: BOOTP/DHCP, Request from 52:54:00:07:73:c3, length 300
12:58:53.761266 52:54:00:07:73:c3 > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342: 0.0.0.0.68 >
255.255.255.255.67: BOOTP/DHCP, Request from 52:54:00:07:73:c3, length 300
12:58:56.764648 52:54:00:07:73:c3 > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342: 0.0.0.0.68 >
255.255.255.255.67: BOOTP/DHCP, Request from 52:54:00:07:73:c3, length 300
<…snip…>
```

FTP server for files exchange

```
EXT-R-01#mon cap tac export ftp://198.18.135.201/r_pcap.pcap
Writing r_pcap.pcap
Exported Successfully
```

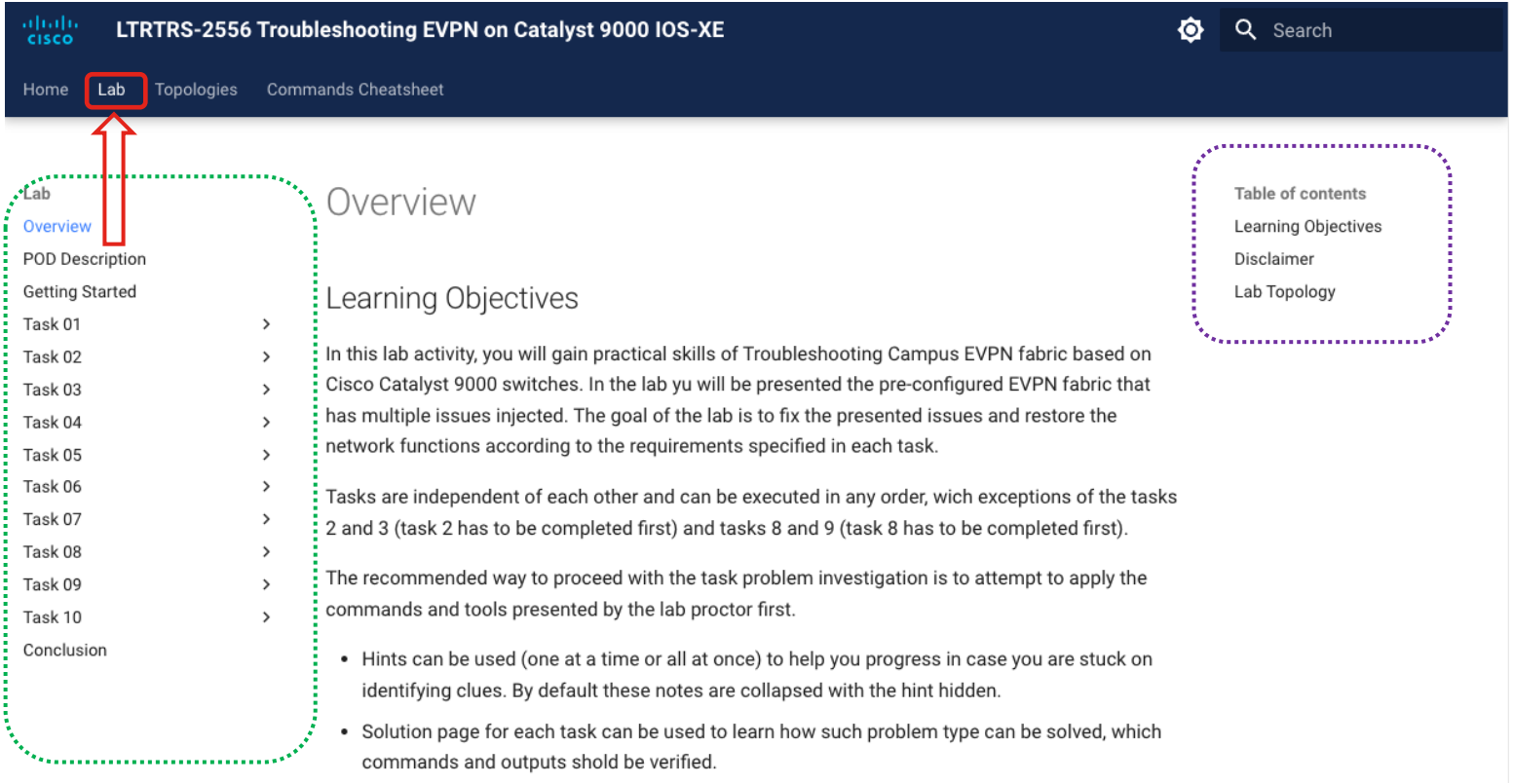


Documentation Overview

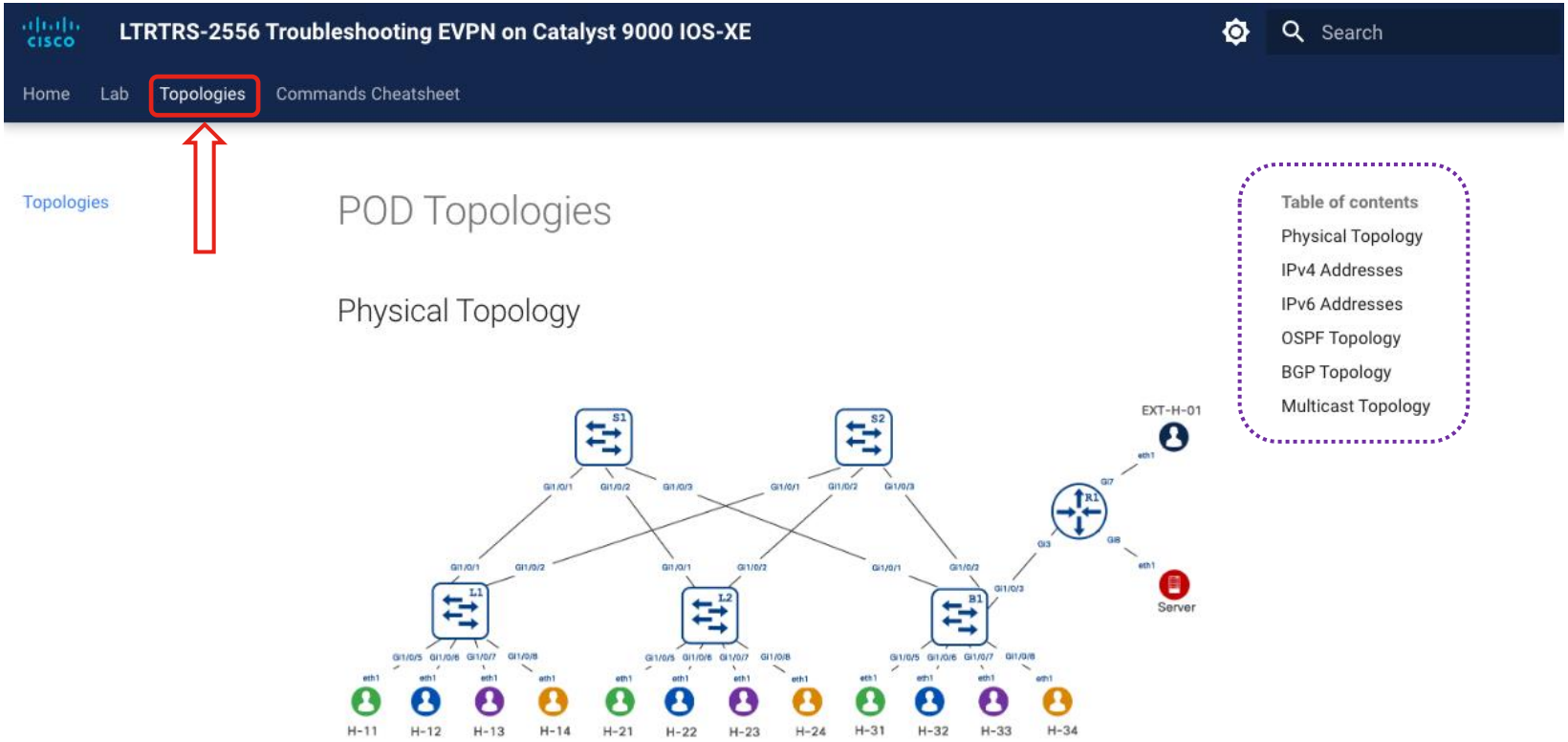
Lab Guide

http://cs.co/9004pZVuA

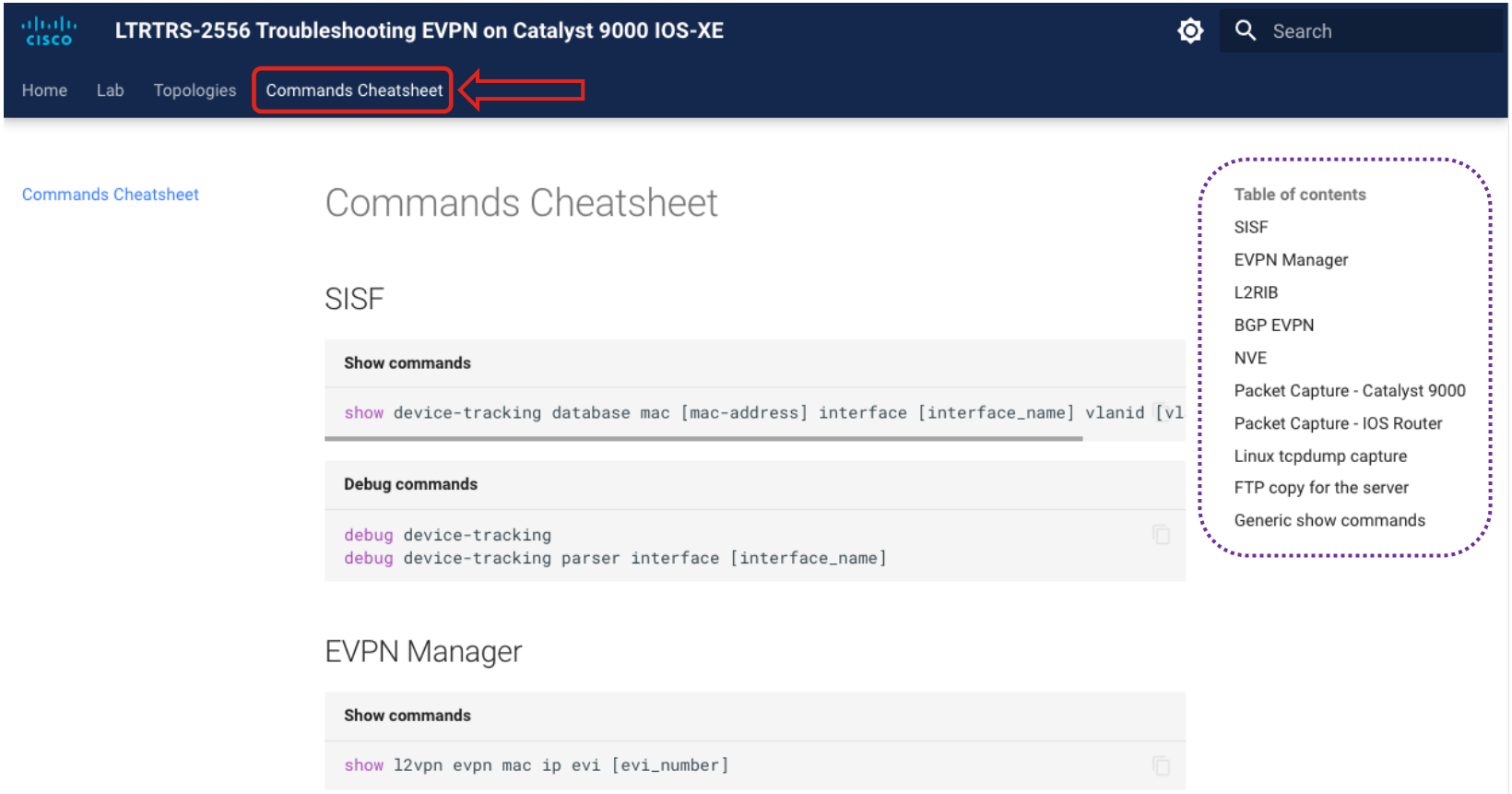
Lab Guide - Structure



Lab Guide - Topologies



Lab Guide – Commands Cheatsheet



Tasks – Problem Description

LTRTRS-2556 Troubleshooting EVPN on Catalyst 9000 IOS-XE

Search

Home | Lab | Topologies | Commands Cheatsheet

- Lab
- Overview
- POD Description
- Getting Started
- Task 01
  - Problem description
  - Explanation
- Task 02
- Task 03
- Task 04
- Task 05
- Task 06
- Task 07
- Task 08
- Task 09
- Task 10
- Conclusion

Task 1: Connectivity issue between H-13 and H-23

Topology

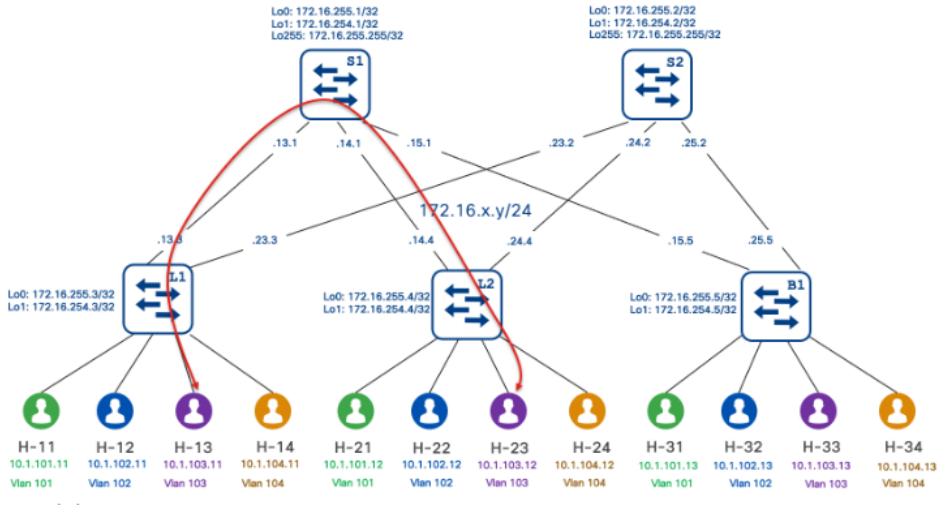
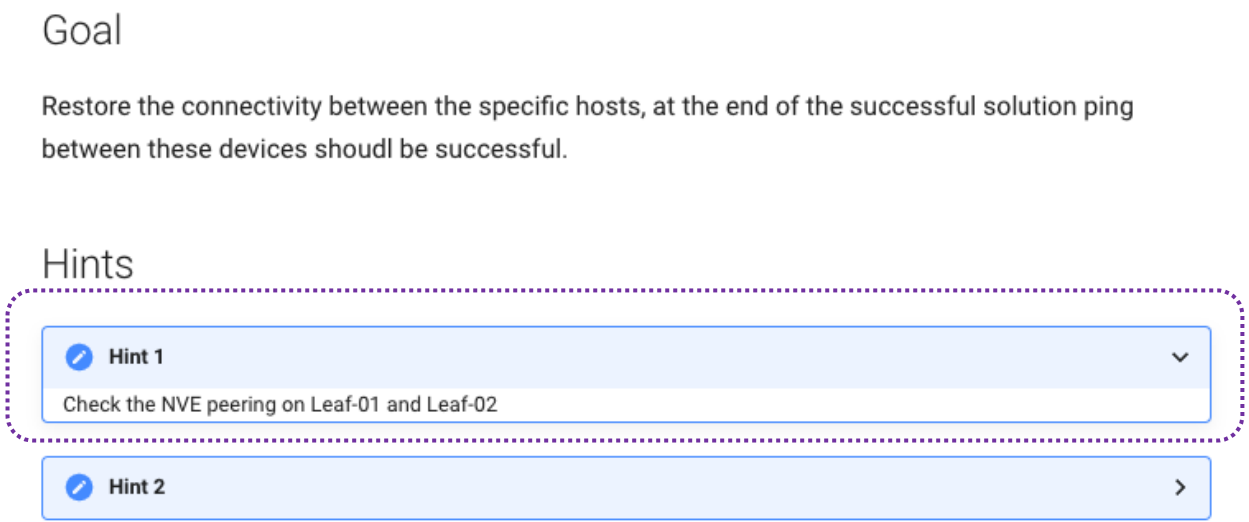


Table of contents

- Topology
- Problem description
- Goal
- Hints

Task Details and Hints

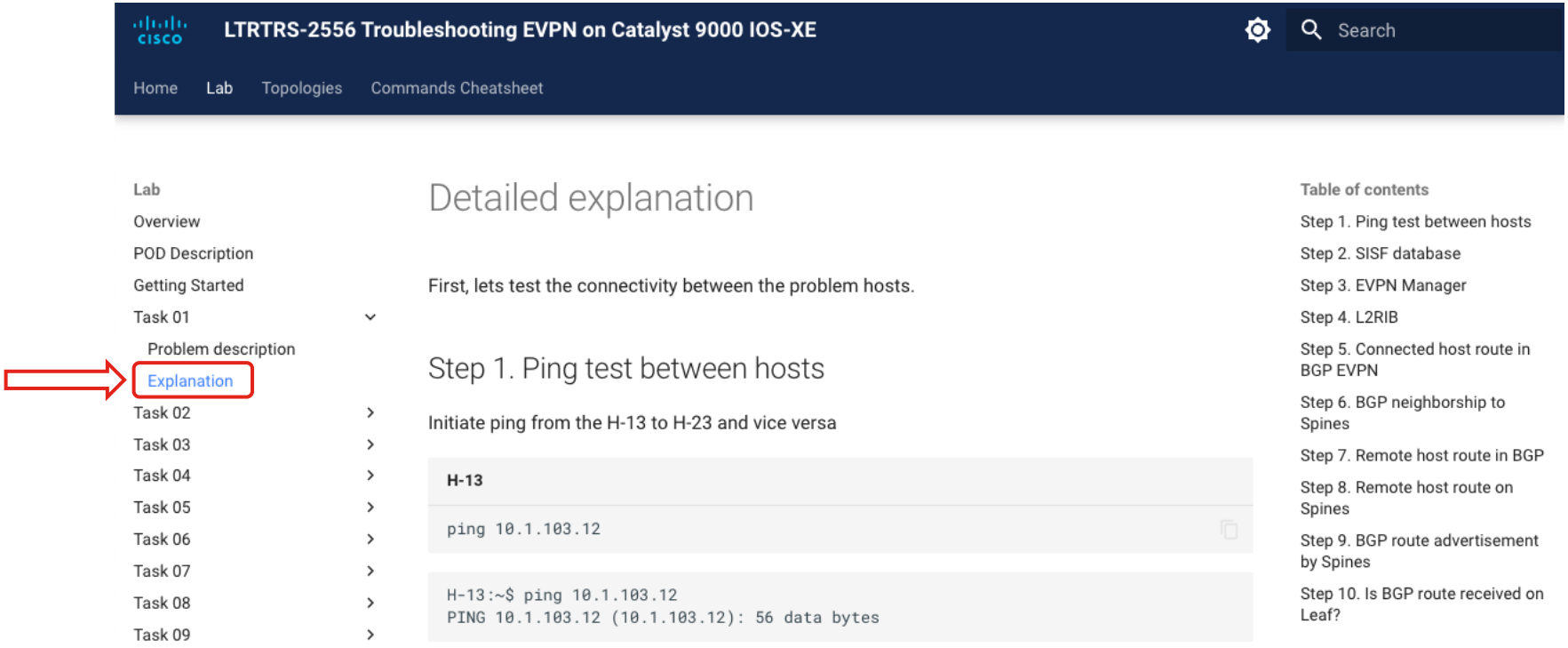
Use Hints if you get stuck in your investigation



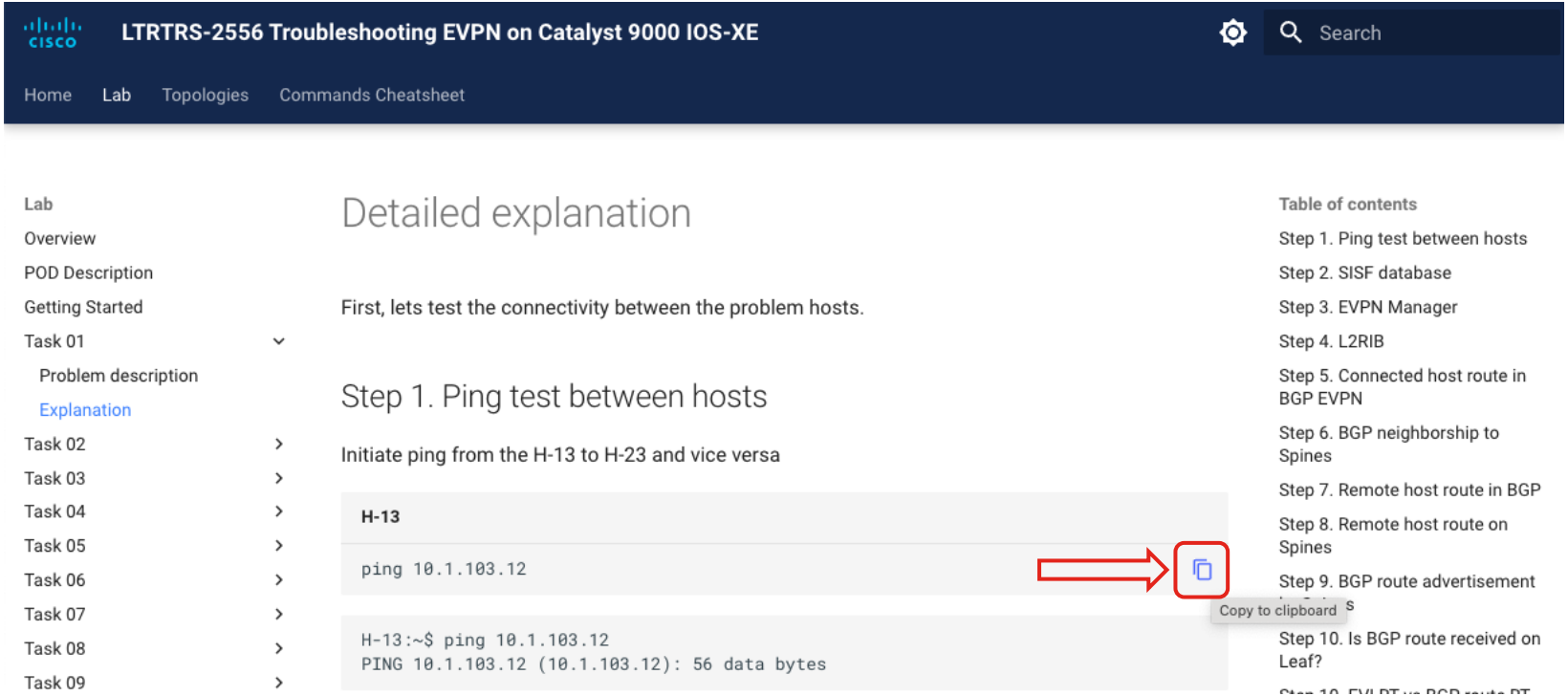
Task Explanation

We recommend to read through explanation at the end of each task to see the full logic of research process for specific issue

Explanation describes step by step research process for the issue

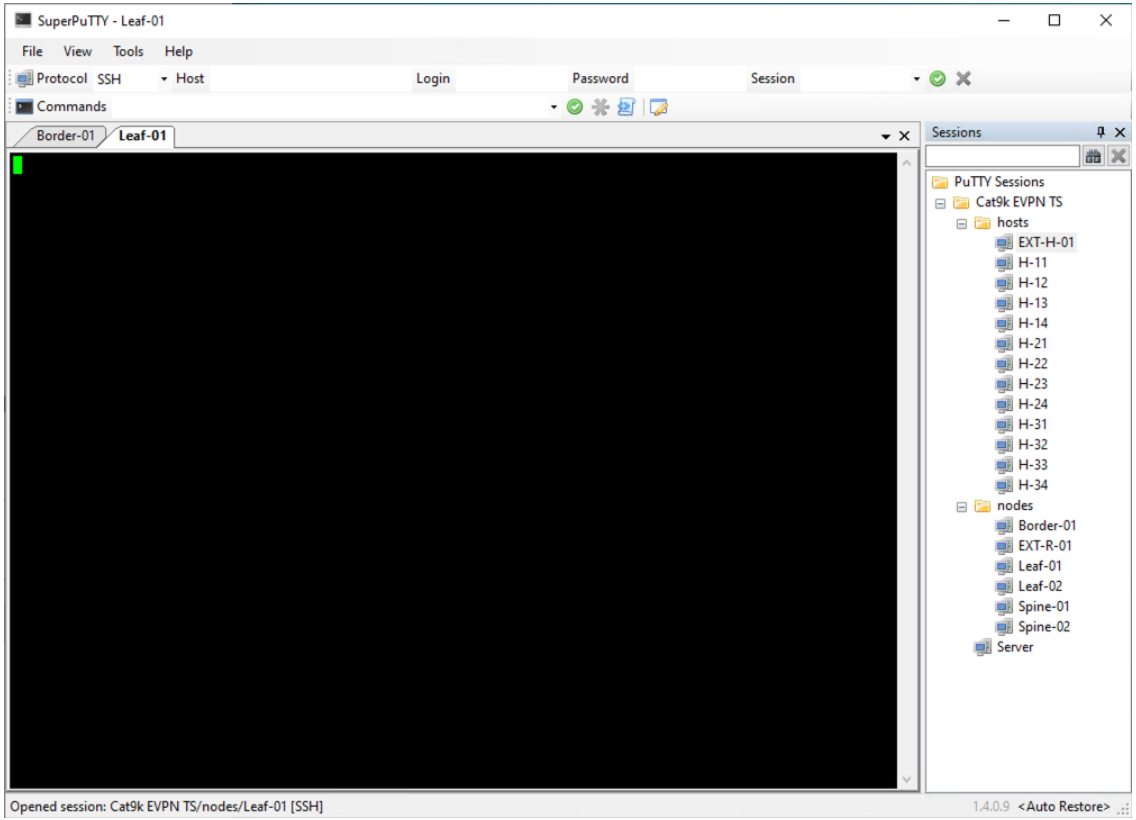


Lab Guide –Copy to Clipboard



Getting Started

Connecting to Devices – SuperPutty



Username: `cisco`
Password: `cisco`

Time Management

Time Management

http://cs.co/9004pZVuA

- Total time of the lab is 4 hours
  - Theory and Lab setup – 45 minutes
  - Labbing time - 3 hours 15 minutes
  - 15-20 minutes per task
  - Dedicate more time for task 1 and its Explanation
- WATCH YOUR TIME!
  - Don’t hesitate to use Hints and Solution if you get stuck
  - Tasks 1-5 : Base
  - Tasks 6-10 : Advanced
- Read the Solution sections
  - recommended for full step by step research details

Conclusion

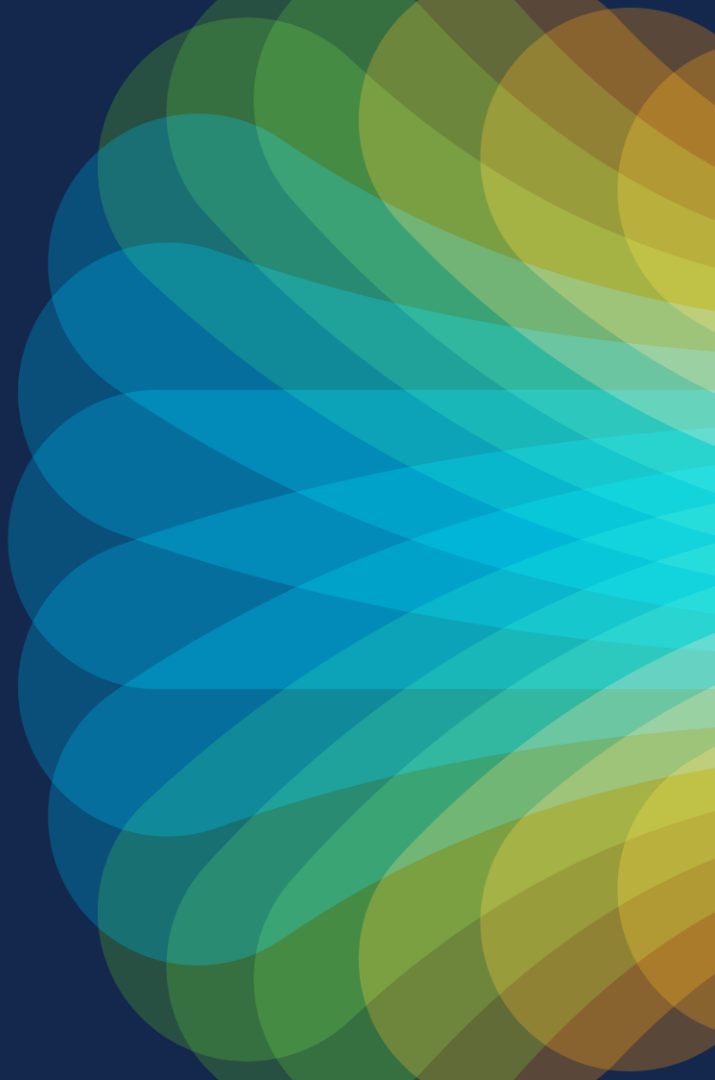
Conclusion

TAC perspective: common EVPN issues, IOS-XE tools and troubleshooting methods

- Campus EVPN Troubleshooting in the Fabric on Catalyst 9000 Switches
- Main EVPN components and relevant troubleshooting methods
- End-to-End verification of the Fabric control plane from Fabric and switch components interaction
- Issue investigation methods used by TAC engineers
- 10 of the most often seen issues in TAC cases

The bridge to possible

Thank you



cisco Live!


cisco Live!
Let's go