cisco Live!

Let’s go

#CiscoLive

The bridge to possible

Leverage Innovations in PoE, Troubleshooting Tools, and Logging on Catalyst 9000

Jason Babb, Escalation Engineer
Nathan Pan, Technical Leader

BRKTRS-2002

Cisco Webex App

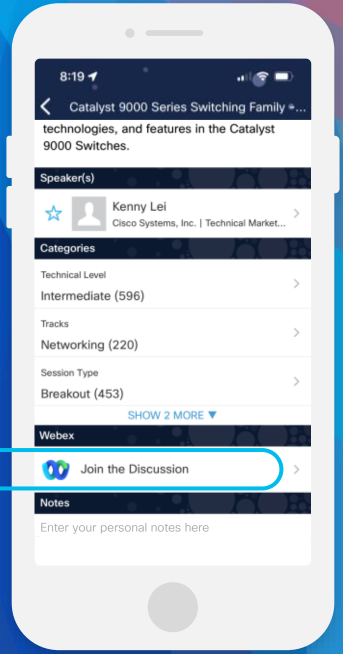
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click "Join the Discussion"
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated
by the speaker until June 9, 2023.



8:19

Catalyst 9000 Series Switching Family •...

technologies, and features in the Catalyst 9000 Switches.

Speaker(s)

Kenny Lei
Cisco Systems, Inc. | Technical Market...

Categories

Technical Level
Intermediate (596)

Tracks
Networking (220)

Session Type
Breakout (453)

SHOW 2 MORE ▼

Webex

Join the Discussion

Notes

Enter your personal notes here

https://ciscolive.ciscoevents.com/ciscolivebot/#BRKTRS-2002

Agenda

- Introduction
- Troubleshooting Tools
- Logging
- PoE Innovations
- Conclusion

Introduction

Jason Babb

Quick facts!

- TAC Team Lead – Enterprise Routing and Switching
- Formerly Army
- Loves animals (including Charlie- a very good boy)
- Enjoys playing music, gardening and being out in nature

Nathan Pan

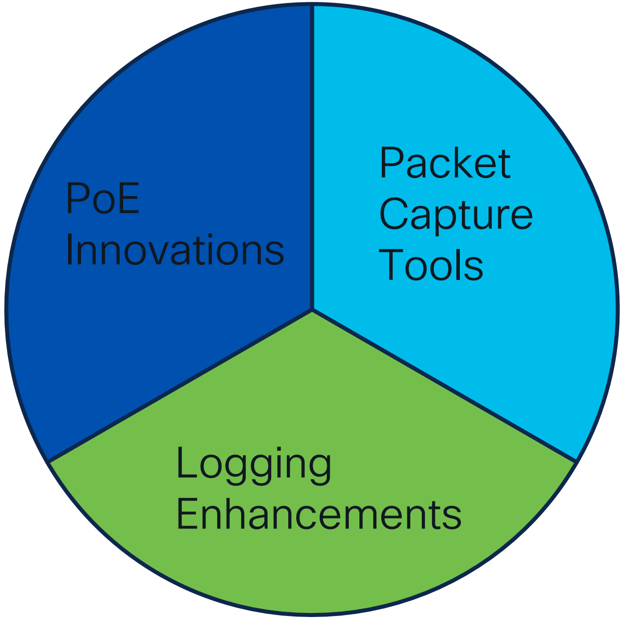
A Little Bit About Myself

- Technical Leader in SD-Access and Enterprise Switching
- 29 years old, 7 years experience in Cisco TAC
- Problem solver at heart, passionate about mentoring and teaching others
- Enjoy spending time playing piano, working out, and catching soccer games

Our Focus

What am I going to get out of this session?

- Leverage logging enhancements already present on Cat9000 switches
- Learn about new and existing PoE innovations on Cat9000 switches.
- Ability to choose the right troubleshooting tool for problems you can encounter



Embedded Packet Capture

Embedded Wireshark

Overview

What is it?

- Combines Wireshark + Embedded Packet Capture (EPC) to provide control and data plane capture capabilities
- Leverages local buffer to store packet data

What Does it Provide?

- Ability to export data in PCAP format that can be viewed in Wireshark
- View packet capture onboard with varying levels of granularity
- Flexible filter options to capture and display relevant data

Supported Platforms: Catalyst 9300, 9400, 9500, 9600

- Requires DNA Advantage Licensing
- Catalyst 9200 supports EPC only

Embedded Packet Capture

Advantages and Benefits

- Onboard capture with priv-exec commands to start/stop the capture, define buffer and capture parameters
- Ability to export packet capture in PCAP format to view on Wireshark
- Shines when onsite access is unavailable, remote debug and troubleshooting only available
- Capture can be manipulated (ACL filter, maximum packet capture rate, duration, or sample interval)
- Ability to view packet capture on switch itself via show commands

Embedded Packet Capture

Limitations and Restrictions

- Best effort feature (uses memory and CPU resources) may result in inaccurate packet captures if there are resource constraints
- 1000 packet per second (pps) rate-limiter set to protect CPU can introduce artificial loss in the packet capture
- Multicast packets will be captured ingress, but not replicated packets on egress
- Packets captured in an egress interface capture may not properly reflect packet rewrite (TTL, MAC address, DSCP/CoS, VLAN tag)

Embedded Packet Capture

Configuration Steps

Configuration Steps:

1. Define the capture buffer parameters (circular vs linear, buffer size)
2. Define the capture point (interface or control-plane)
3. Define any capture filters (match any, match ipv4, ipv6, mac, access-list)
4. Capture data
5. Export/display captured data

```
Catalyst-9300X-24HX#monitor capture CAP buffer circular size 5

Catalyst-9300X-24HX#monitor capture CAP interface Ten1/0/1 both

Catalyst-9300X-24HX#monitor capture CAP access-list ACL

Catalyst-9300X-24HX#monitor capture CAP start

Catalyst-9300X-24HX#monitor capture CAP stop

Catalyst-9300X-24HX#monitor capture CAP export location bootflash:capture1.pcap
```

Embedded Packet Capture

Validation and Setup

```
Catalyst-9300X-24HX#show monitor capture CAP parameter

   monitor capture CAP interface TenGigabitEthernet1/0/1 BOTH

   monitor capture CAP access-list ACL

   monitor capture CAP buffer size 2
```

```
Catalyst-9300X-24HX#monitor capture CAP start
Catalyst-9300X-24HX#monitor capture CAP stop
Capture statistics collected at software:
     Capture duration - 7 seconds
     Packets received - 38
     Packets dropped - 0
     Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared
```

Embedded Packet Capture

Viewing Packet Capture Onboard the Switch

```
Catalyst-9300X-24HX#show monitor capture CAP buffer ?
  brief             brief display
  detailed          detailed display
  display-filter    Display filter
  dump              for dump
  |                 Output modifiers
  <cr>              <cr>
```

```
Catalyst-9300X-24HX#show monitor capture CAP buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

   11   5.590941  172.19.13.1 -> 255.255.255.255 DHCP 361 DHCP Offer    - Transaction ID 0x111d
   12   5.590969  172.19.13.1 -> 255.255.255.255 DHCP 361 DHCP Offer    - Transaction ID 0x111d
```

Embedded Packet Capture

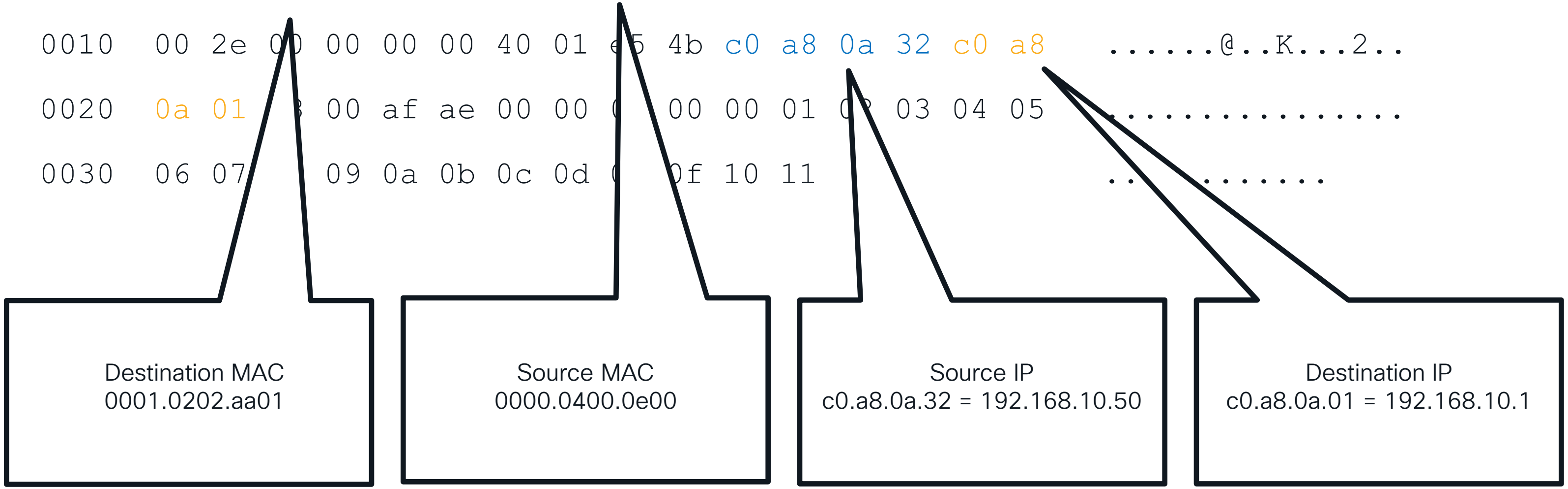
Viewing Packet Capture Onboard the Switch

```
Catalyst-9300X-24HX#show monitor capture CAP buffer detailed

Starting the packet display ........ Press Ctrl + Shift + 6 to exit

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface /tmp/epc_ws/wif_to_ts_pipe, id 0

Ethernet II, Src: 00:00:04:00:0e:00 (00:00:04:00:0e:00), Dst: ff:ff:ff:ff:ff:ff(ff:ff:ff:f:ff:f)

Internet Protocol Version 4, Src: 172.19.13.1, Dst: 255.255.255.255

User Datagram Protocol, Src Port: 68, Dst Port: 67

Dynamic Host Configuration Protocol (Offer)
```

Embedded Packet Capture

Viewing Packet Capture Onboard the Switch

```
Catalyst-9300X-24HX#show monitor capture CAP buffer dump

Starting the packet display ........ Press Ctrl + Shift + 6 to exit


0000   00 01 02 02 aa 01 00 00 04 00 0e 00 08 00 45 00   ..............E.

0010   00 2e 00 00 00 00 40 01 e5 4b c0 a8 0a 32 c0 a8   ......@..K...2..

0020   0a 01 08 00 af ae 00 00 00 00 00 01 02 03 04 05   ................

0030   06 07 08 09 0a 0b 0c 0d 0e 0f 10 11               ............
```



Destination MAC
0001.0202.aa01

Source MAC
0000.0400.0e00

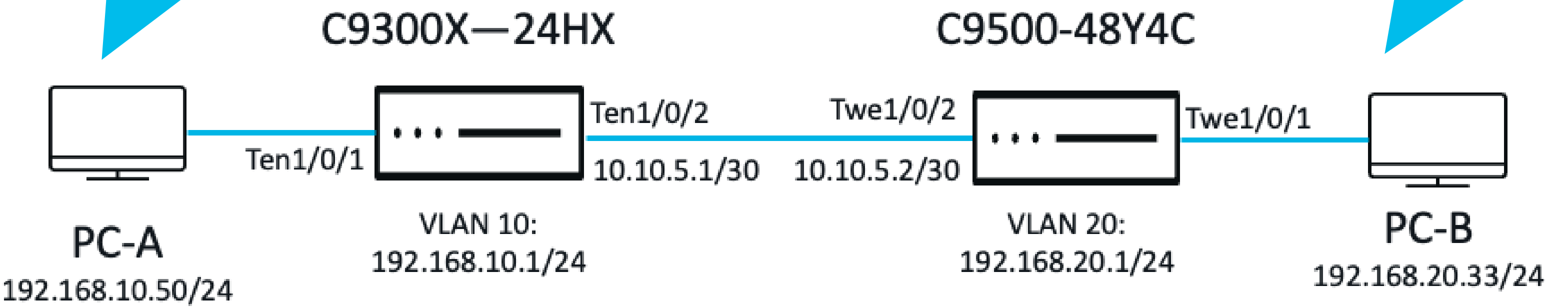
Source IP
c0.a8.0a.32 = 192.168.10.50

Destination IP
c0.a8.0a.01 = 192.168.10.1

Real World Example
Isolate the Packet Loss

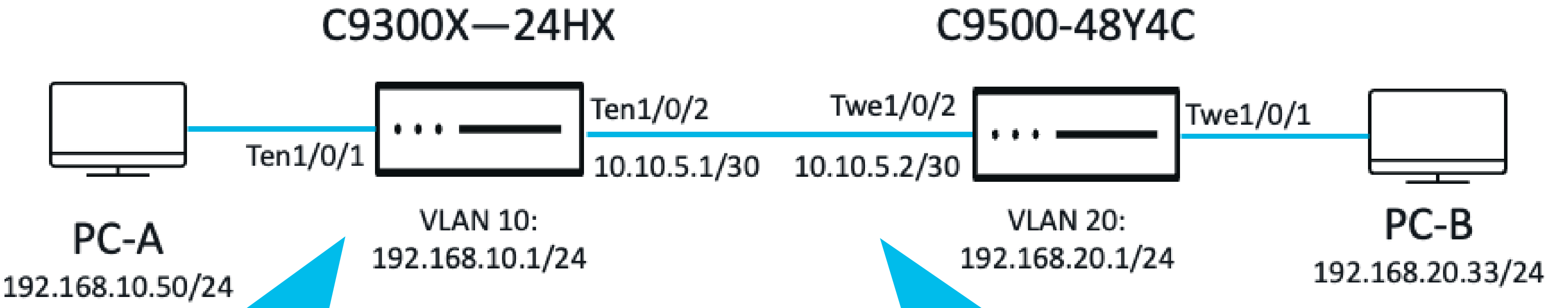
I am unable to SSH into PC-B, but can ping PC-B properly

Wireshark capture shows no TCP SYNs coming from PC-A



Let’s leverage Embedded Wireshark

Real World Example
Isolate the Packet Loss



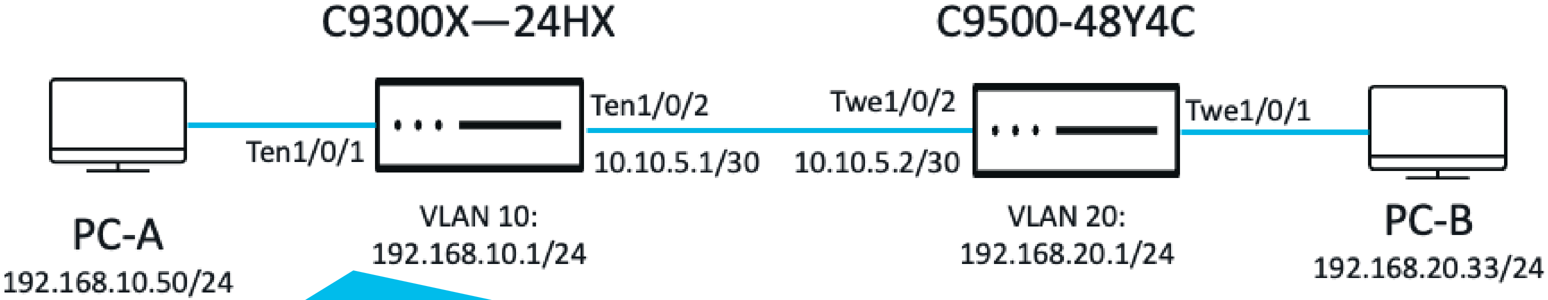
```
ip access-list extended MYACL
permit ip host 192.168.10.50 host 192.168.20.33
monitor capture CAP interface Ten1/0/1 in access-list MYACL
monitor capture CAP start
monitor capture CAP stop
```

```
ip access-list extended MYACL
permit ip host 192.168.10.50 host 192.168.20.33
monitor capture CAP interface Twe1/0/2 in access-list MYACL
monitor capture CAP start
monitor capture CAP stop
```

Real World Example
Isolate the Packet Loss

Only ICMP Requests seen on Twe1/0/2 EPC

```
Catalyst-9500-48Y4C#show monitor capture CAP buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 192.168.10.50 b^F^R 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=63
    2   0.199968 192.168.10.50 b^F^R 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=63
    3   0.399982 192.168.10.50 b^F^R 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=63
    4   0.599966 192.168.10.50 b^F^R 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=63
```



```
Catalyst-9300X-24HX#show monitor capture CAP buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 192.168.10.50 -> 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=64
    2   0.000026 192.168.10.50 -> 192.168.20.33 SSH 60 Server: [TCP SYN] , Encrypted packet (len=6)
    3   0.200037 192.168.10.50 -> 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=64
    4   0.200074 192.168.10.50 -> 192.168.20.33 TCP 60 [TCP SYN] [TCP Retransmission] 22 -> 0 [<None>] Seq=1 Win=0 Len=6
```

Did the TCP SYN packets leave the C9300? Let's find out.

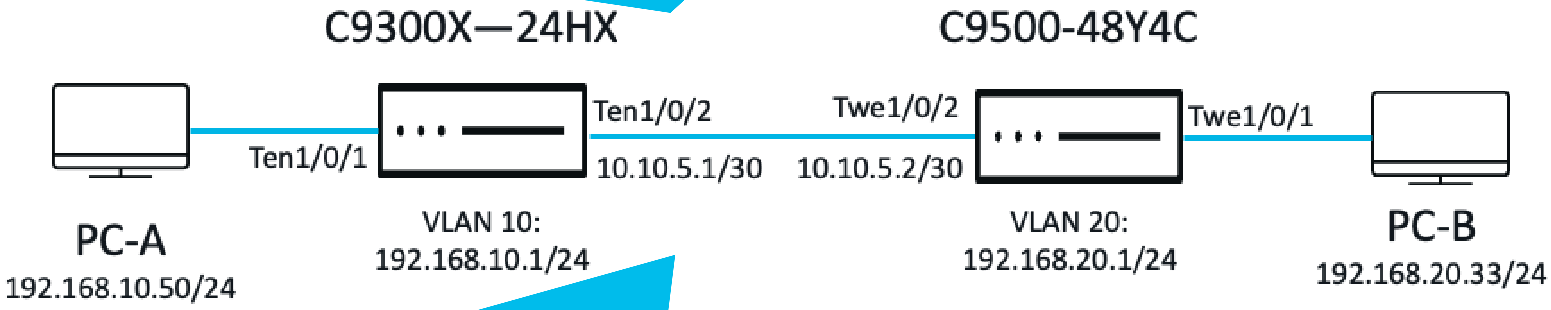
Real World Example

Isolate the Packet Loss

```
ip access-list extended MYACL
permit ip host 192.168.10.50 host 192.168.20.33
monitor capture CAP interface Ten1/0/2 out access-list MYACL
monitor capture CAP start
monitor capture CAP stop
```

Only ICMP Requests seen on Ten1/0/2 EPC

C9300X is dropping the TCP SYN packets



```
Catalyst-9300X-24HX#show monitor capture CAP buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 192.168.10.50 -> 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=64
    2   0.199974 192.168.10.50 -> 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=64
    3   0.399986 192.168.10.50 -> 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=64
    4   0.599973 192.168.10.50 -> 192.168.20.33 ICMP 60 Echo (ping) request  id=0x0000, seq=0/0, ttl=64
```

Show Platform Forward (SPF)

Show Platform Forward (SPF)

Overview

What is it?

- Uses 150-200 CPU generated dummy packets to identify switch forwarding decision
- Manual packet creation or Embedded Wireshark PCAP as trigger

What Does it Provide?

- Ability to see forwarding decision for a given frame/packet
- View forwarding decision with vary level of detail
- Later IOS XE versions support complex forwarding scenarios (MPLS, VxLAN)

Supported Platforms: Catalyst 9200, 9300, 9400, 9500 (except 9500H)
Starting in 17.2.X, 9500H and 9600 support this feature

Show Platform Forward (SPF)

Advantages and Benefits

- Onboard capture with priv-exec commands to identify switch forwarding decision based on manual or Embedded Wireshark PCAP trigger
- Ability to identify what the switch would do with a frame/packet (drop/forward/punt)
- Shines when onsite access is unavailable, remote debug and troubleshooting only available
- Packet trigger can accept manual parameters or PCAP data
- Ability to view forwarding decision on switch itself via show commands

Show Platform Forward (SPF)

Limitations and Restrictions

- Uses CPU resources (high CPU or large number of packets at the CPU may result in control-plane instability)
- Does not prove actual receipt of a packet, simulates what would happen if that packet was received
- Will not demonstrate packet drop due to QoS/Policer drops

Show Platform Forward (SPF)

Configuration Steps

Configuration Steps:

1. Identify the switch and interface the frame/packet should ingress
2. Manually define the packet parameters or utilize packet from PCAP
3. Execute show platform forward
4. View summary result

```
Catalyst-9400#show platform hardware fed active forward interface Gig1/0/1 0000.0400.0e00 ffff.ffff.ffff ipv4 0.0.0.0 255.255.255.255 udp 68 67

Catalyst-9400#show platform hardware fed active forward interface Gig1/0/1 pcap flash:DHCP_DISCOVER.pcap number 1 data

Catalyst-9400#show platform hardware fed active forward last summary
```

Show Platform Forward (SPF)

View Forwarding Decision

```
Catalyst-9400#show platform hardware fed active forward interface Gig1/0/1 flash:DHCP_DISCOVER.pcap packet 1 data
Show forward is running in the background. After completion, syslog will be generated.
```

```
Catalyst-9400#show platform hardware fed active forward last summary
Input Packet Details:
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 00:00:04:00:0e:00
  type      = 0x8100
###[ 802.1Q ]###
     vlan      = 10
     type      = 0x800
###[ IP ]###
        version   = 4
        frag      = 0
        ttl       = 64
        proto     = udp
        chksum    = 0x7ad1
        src       = 0.0.0.0
        dst       = 255.255.255.255
        options   = ''
###[ UDP ]###
           sport     = bootpc
           dport     = bootps
           len       = 8
           chksum    = 0xff57
```

Show Platform Forward (SPF)

View Forwarding Decision

```
Ingress:
   Port                         : GigabitEthernet1/0/1
Vlan                         : 10
   Mapped Vlan ID               : 7
L3 Interface                 : 0
       IPv4 Routing             : enabled
       IPv6 Routing             : enabled
       Vrf Id                   : 0
Decision:
       Destination Index        : 25      [DI_DIET_L2]
       Rewrite Index            : 2       [RI_L2]
       Dest Mod Index           : 24
       CPU Map Index            : 0       [CMI_NULL]
       Forwarding Mode          : 0       [Bridging]
       Replication Bit Map      :         ['localData', 'remoteData', 'coreData']
       Winner                   :         L2DESTMACVLAN LOOKUP
       Qos Label                : 1
       SGT                      : 0
       DGTID                    : 0
```

Show Platform Forward (SPF)

View Forwarding Decision

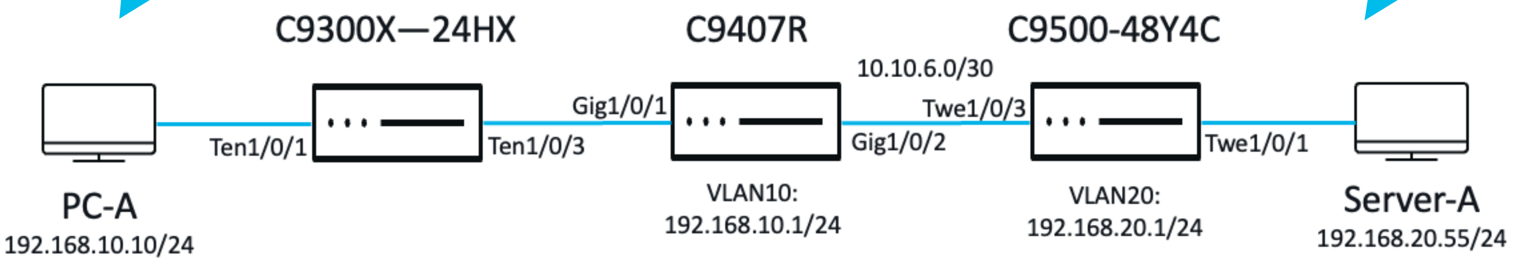
```
Egress:
   Possible Replication        :
       Port                    : GigabitEthernet1/0/1
       Port                    : GigabitEthernet1/0/2
   Output Port Data            :
     Port                      : GigabitEthernet1/0/2
       Rewrite Type            : 1       [L2_BRIDGE]
       Mapped Rewrite Type     : 4       [L2_BRIDGE_INNER_IPv4]
       Vlan                    : 10
       Mapped Vlan ID          : 7
```

Real World Example

Multicast Packet Loss

I am interested in multicast traffic 239.1.1.5 but not receiving anything!

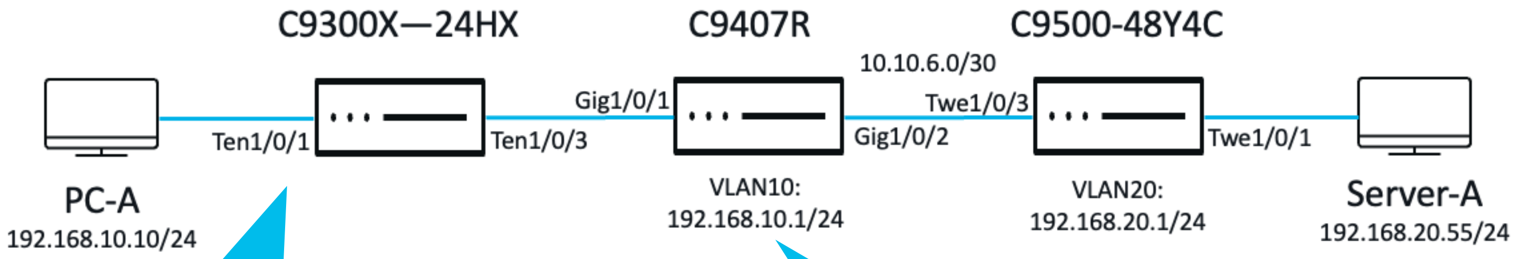
TCPDump shows multicast UDP packets destined to 239.1.1.5 leaving!



Let’s leverage Show Platform Forward (SPF)

Real World Example

Multicast Packet Loss



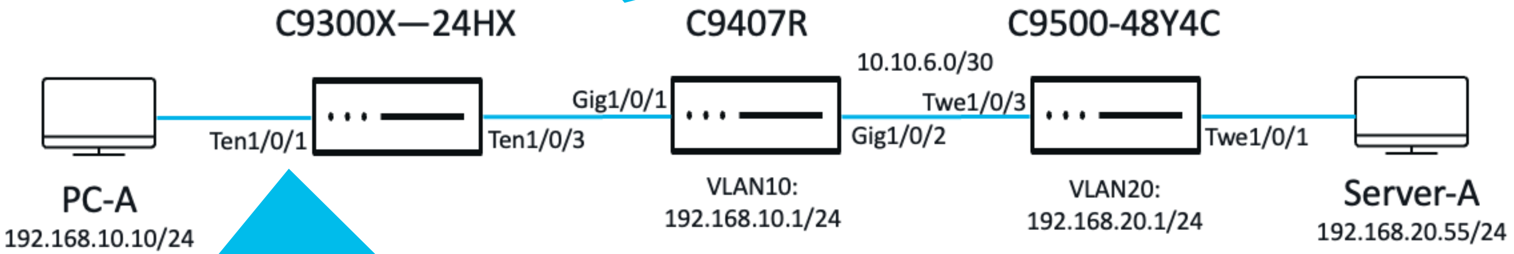
```
ip access-list extended IGMP
permit ip host 192.168.10.10 host 239.1.1.5
monitor capture IGMP interface Ten1/0/1 in access-list IGMP
monitor capture IGMP start
monitor capture IGMP stop
```

```
ip access-list extended IGMP
permit ip host 192.168.10.10 host 239.1.1.5
monitor capture IGMP interface Gig1/0/1 in access-list IGMP
monitor capture IGMP start
monitor capture IGMP stop
```

Real World Example

Multicast Packet Loss

```
Catalyst-9400#show monitor capture IGMP buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 192.168.10.10 b^F^R 239.1.1.5    IGMPv2 64 Membership Report group 239.1.1.5
    2   9.999945 192.168.10.10 b^F^R 239.1.1.5    IGMPv2 64 Membership Report group 239.1.1.5
    3  19.999791 192.168.10.10 b^F^R 239.1.1.5    IGMPv2 64 Membership Report group 239.1.1.5
```

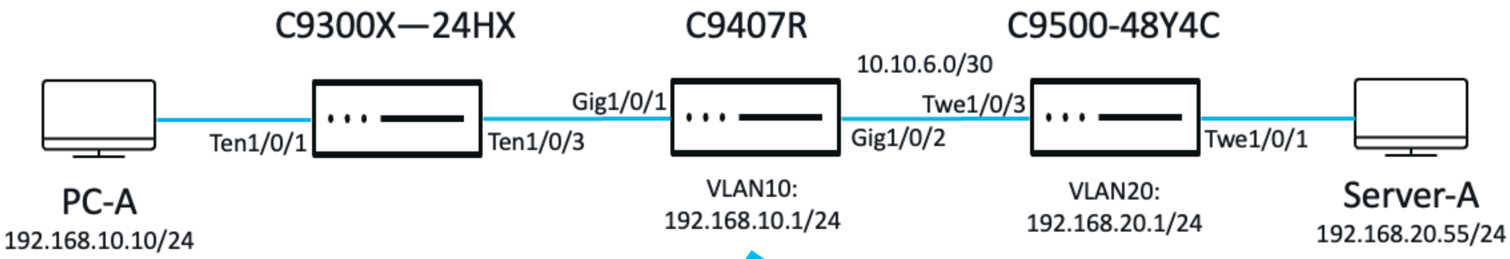


```
Catalyst-9300X-24HX#show monitor capture IGMP buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 192.168.10.10 -> 239.1.1.5    IGMPv2 60 Membership Report group 239.1.1.5
    2   0.199882 192.168.10.10 -> 239.1.1.5    IGMPv2 60 Membership Report group 239.1.1.5
    3   0.399955 192.168.10.10 -> 239.1.1.5    IGMPv2 60 Membership Report group 239.1.1.5
    4   0.599865 192.168.10.10 -> 239.1.1.5    IGMPv2 60 Membership Report group 239.1.1.5
```

Real World Example

Multicast Packet Loss

```
Catalyst-9400#show ip igmp snooping groups
Vlan      Group                    Type       Version     Port List
-----------------------------------------------------------------------

Catalyst-9400#
```

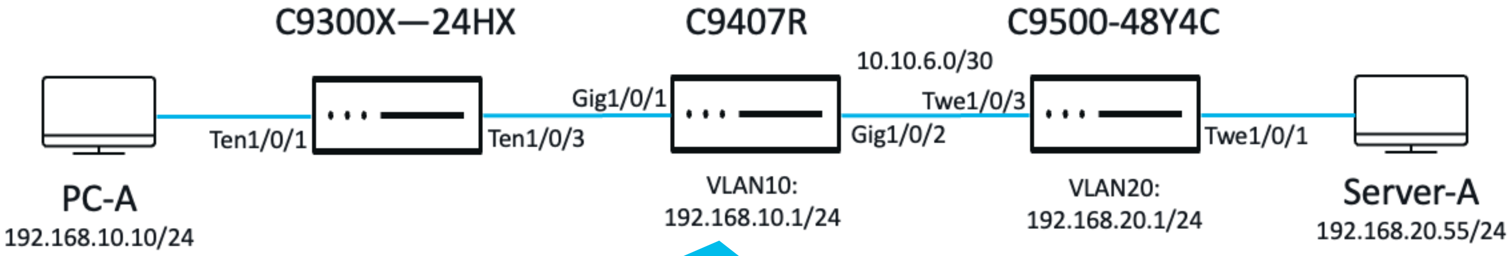


```
Catalyst-9400#show ip mroute 239.1.1.5
Group 239.1.1.5 not found
```

Real World Example

Multicast Packet Loss

```
Catalyst-9400#show platform hardware fed active forward interface gig1/0/1 pcap flash:IGMP.pcap number 1 data
Show forward is running in the background. After completion, syslog will be generated.
```



```
Catalyst-9400#monitor capture IGMP export location bootflash:IGMP.pcap
Export Started Successfully
```

Real World Example

Multicast Packet Loss

```
Catalyst-9400#show platform hardware fed active forward last summary
Input Packet Details:
###[ Ethernet ]###
  dst          = 01:00:5e:01:01:05
  src          = 00:00:04:00:0e:00
  type         = 0x8100
###[ 802.1Q ]###
     prio         = 0
     id           = 0
     vlan         = 10
     type         = 0x800
###[ IP ]###
        version    = 4
        tos        = 0x0
        len        = 32
        frag       = 0
        ttl        = 64
        proto      = igmp
        chksum     = 0x2b1f
        src        = 192.168.10.10
        dst        = 239.1.1.5
        options    = '\\x94\x04\x00\x00'
###[ IP Options ]###
           optcopy    = 0
           optcls     = 0
           opttype    = 22
```

Real World Example

Multicast Packet Loss

C9400 is dropping the packet due to STP!

```
Ingress:
   Port                          : GigabitEthernet1/0/1
   Vlan                          : 10
   Mapped Vlan ID                : 7
   STP Instance                  : 5
   BlockForward                  : 1
   BlockLearn                    : 1
   L3 Interface                  : 39
       IPv4 Routing              : enabled
       IPv6 Routing              : enabled
       Vrf Id                    : 0
Decision:
       Destination Index         : 0       [DI_NULL]
       Rewrite Index             : 1       [RI_CPU]
       Dest Mod Index            : 1
[IGR_FIXED_DMI_DROP_FORWARDING_CONTEXT]
       CPU Map Index             : 0       [CMI_NULL]
       Forwarding Mode           : 0       [Bridging]
       Replication Bit Map       :         []
       Winner                    :         CPPIPV4 LOOKUP1
       Qos Label                 : 1
       SGT                       : 0
       DGTID                     : 0
Packet DROPPED
 Drop due to STP block forward.
```

Packet State Vector (PSV)

Packet State Vector (PSV)

Overview

What is it?

- Single shot capture mechanism that captures the very first packet
- ELAM-like capture that captures a live packet based on capture criteria defined by the administrator
- No effect on switch functionality and is independent on any feature interaction(s)

What Does It Provide?

- Provides confirmation of packet receipt and subsequent forwarding decision
- Flexible capture criteria to capture various types of frames/packets

Supported Platforms: UADP 3.0 ASICs (C9500H models and C9600-SUP-1)

Packet State Vector (PSV)

Advantages and Benefits

- Onboard capture with priv-exec commands to start/stop the capture, define capture criteria
- Ability to combine capture criteria to identify switch forwarding decision
- Shines when onsite access is unavailable, remote debug and troubleshooting only available
- Distinguishes itself by being able to truly confirm packet receipt and subsequent forwarding decision
- Triggers can be as specific or generic as needed

Packet State Vector (PSV)

Limitations and Restrictions

- Tool will only pick up the first packet that matches the capture criteria defined. Administrators must re-enable PSV to capture subsequent packets
- Packets that require recirculation (VXLAN, MPLS, VPLS) will require PSV multiple captures to see final forwarding decision

Packet State Vector (PSV)

Configuration Steps

Configuration Steps:

1. Identify the switch and interface the frame/packet should ingress/egress
2. Define PSV capture criteria/trigger
3. Start PSV capture
4. View PSV capture status
5. View PSV capture data

```
Catalyst-9500-48Y4C#debug platform hardware fed active capture trigger ipv4 10.10.6.1 10.10.6.2 icmp

Capture trigger set successful.

Catalyst-9500-48Y4C#debug platform hardware fed active capture trigger interface Twe1/0/3 ingress

Capture trigger set successful.

Catalyst-9500-48Y4C#debug platform hardware fed active capture start

Packet Capturing Started.
```

Packet State Vector (PSV)

View Trigger and Status

```
Catalyst-9500-48Y4C#show platform hardware fed active capture trigger


Trigger Set:

Ingress Interface: TwentyFiveGigE1/0/3

Dest IP: 10.10.6.2

Src IP: 10.10.6.1

Protocol: 0x1


Catalyst-9500-48Y4C#show platform hardware fed active capture status

Asic: 0  Status: Running
```

Packet State Vector (PSV)

View Status and Results

```
Catalyst-9500-48Y4C#show platform hardware fed active capture status

Asic: 0  Status: Completed


Catalyst-9500-48Y4C#show platform hardware fed active capture summary


Trigger: Ingress Interface:TwentyFiveGigE1/0/3 Dest IP:10.10.6.2 Src IP:10.10.6.1 Protocol:0x1


Input        Output        State      Reason

Tw1/0/3      cpuQ 2        PUNT       Bridged
```

Packet State Vector (PSV)

View Packet

```
Catalyst-9500-48Y4C#show platform hardware fed active capture packet

Trigger: Ingress Interface:TwentyFiveGigE1/0/3 Dest IP:10.10.6.2 Src IP:10.10.6.1 Protocol:0x1

Ingress Packet Data:

Error:0

DataValid:1

PakLen:118

Interface:Tw1/0/3
```

Packet State Vector (PSV)

View Packet

```
Packet:
###[ Ethernet ]###
  dst        = 5c:5a:c7:61:4c:5f
  src        = 5c:71:0d:4b:1c:26
  type       = 0x800
###[ IP ]###
     version   = 4
     len       = 100
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0x9c73
     src       = 10.10.6.1
     dst       = 10.10.6.2
     options   = ''
```

Packet State Vector (PSV)

View Packet

```
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0x12a1
        id        = 0x3
        seq       = 0x0
###[ Raw ]###
           load      = '00 00 00 00 00 B1 6A F5 AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB
CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD
AB CD AB CD AB CD AB CD AB CD AB CD AB CD'
```

Packet State Vector (PSV)

View Packet

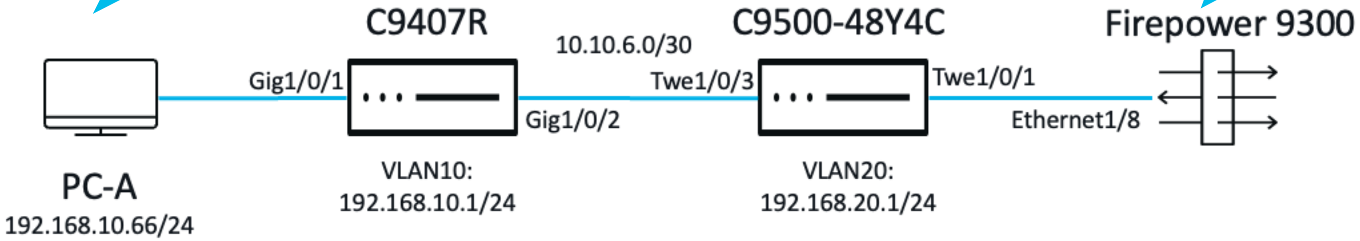
```
Egress Packet Data:


Error:0

DataValid:1

PakLen:118

Interface:CpuQ 2 [CPU_Q_FORUS_TRAFFIC]
```

Real World Example

Web Browsing Slowness

I am trying to access software.cisco.com to download an image but the webpage takes upwards of 30s to load

LINA and Snort level captures do not see any drops or policy violations

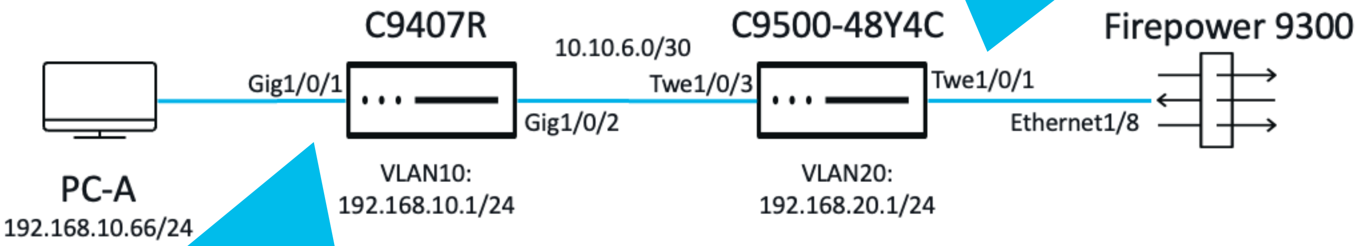


Let’s Leverage Packet State Vector (PSV)

Real World Example

Web Browsing Slowness

```
ip access-list extended WEB
permit ip host 192.168.10.66 host 173.36.127.57
permit ip host 173.36.127.57 host 192.168.10.66
monitor capture WEB start
monitor capture WEB stop
```

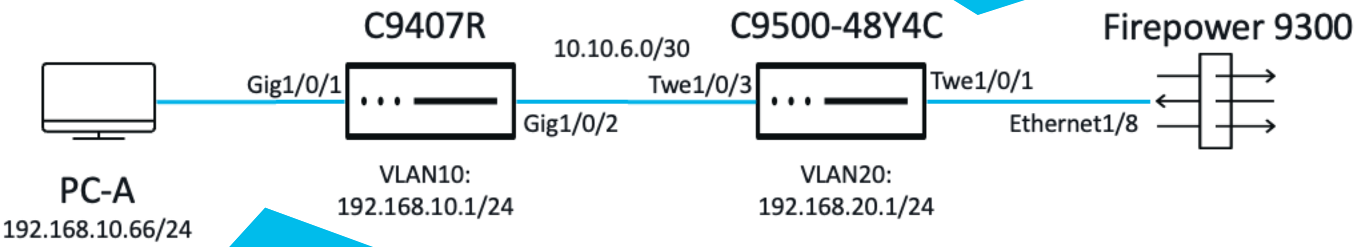


```
ip access-list extended WEB
permit ip host 192.168.10.66 host 173.36.127.57
permit ip host 173.36.127.57 host 192.168.10.66
monitor capture WEB start
monitor capture WEB stop
```

Real World Example

Web Browsing Slowness

```
Catalyst-9500-48Y4C#show monitor capture WEB buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit


Catalyst-9500-48Y4C#
```

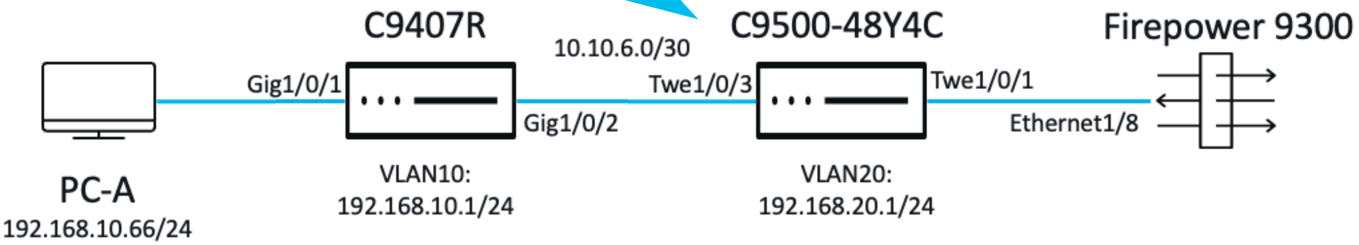


```
Catalyst-9400#show monitor capture WEB buff brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

1   1.346555 192.168.10.66 b^F^R 173.36.127.57 TCP 60 34306 b^F^R 443 [SYN] Seq=0 Win=4128 Len=0 MSS=536
2   3.346918 192.168.10.66 b^F^R 173.36.127.57 TCP 60 [TCP Retransmission] 34306 b^F^R 443 [SYN] Seq=0 Win=4128 Len=0 MSS=536
```

Real World Example

Web Browsing Slowness

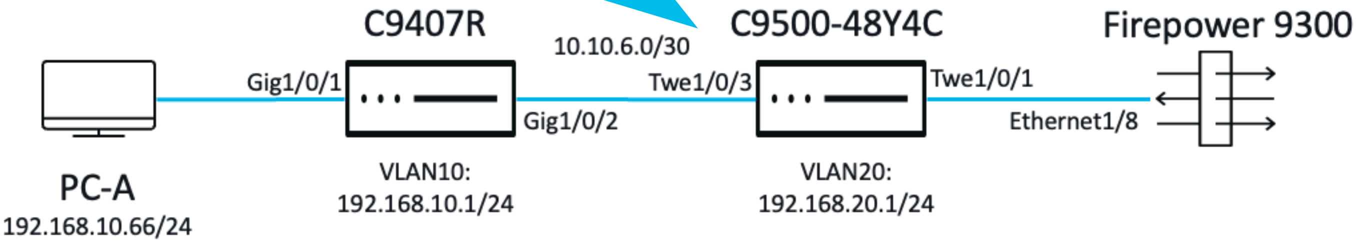
```
Catalyst-9500-48Y4C#debug platform hardware fed active capture trigger ipv4 192.168.10.66 173.36.127.57 tcp
Capture trigger set successful.

Catalyst-9500-48Y4C#debug platform hardware fed active capture trigger interface twe1/0/3 ingress
Capture trigger set successful.

Catalyst-9500-48Y4C#debug platform hardware fed active capture start
Packet Capturing Started.
```



Real World Example

Web Browsing Slowness

```
Catalyst-9500-48Y4C#show platform hardware fed active capture status
Asic: 0  Status: Completed

Catalyst-9500-48Y4C#show platform hardware fed active capture summary

Trigger: Ingress Interface:TwentyFiveGigE1/0/3 Dest IP:173.36.127.57 Src IP:192.168.10.66 Protocol:0x6

Input         Output        State      Reason
Tw1/0/3       cpuQ 14       PUNT       Bridged
```



Real World Example

Web Browsing Slowness

```
Catalyst-9500-48Y4C#show platform hardware fed active capture packet

Trigger: Ingress Interface:TwentyFiveGigE1/0/3 Dest IP:173.36.127.57 Src IP:192.168.10.66 Protocol:0x6

Ingress Packet Data:

Error:0

DataValid:1

PakLen:64

Interface:Tw1/0/3
```

Real World Example

Web Browsing Slowness

```
Packet:
###[ Ethernet ]###
  dst        = 5c:5a:c7:61:4c:5f
  src        = 5c:71:0d:4b:1c:26
  type       = 0x800
###[ IP ]###
     version   = 4
     len       = 44
     frag      = 0
     ttl       = 254
     proto     = tcp
     chksum    = 0x66f2
     src       = 192.168.10.66
     dst       = 173.36.127.57
     options   = ''
```

Real World Example

Web Browsing Slowness

```
Egress Packet Data:

Error:0

DataValid:1

PakLen:64

Interface:CpuQ 14 [CPU_Q_SW_FORWARDING]
```

C9500-48Y4C is punting the TCP SYN up to the CPU!

FED PUNJECT (Punt/Inject)

FED Punject

Overview

What is it?

- Onboard capture tool that aids in identification of traffic that is punted or injected at the CPU

What Does it Provide?

- Ability to see frames/packet(s) punted (from ASIC to CPU) and injected (from CPU to ASIC) in varying degrees of detail
- Supports various display and capture filters
- 17.6.X supports the ability to sort by top talker

Supported Platforms

- Cat9000 series switches 16.X and above

FED Punject

Advantages and Benefits

- Onboard capture with priv-exec commands to start/stop the capture, define buffer and capture parameters
- Dedicated packet capture tool for frames/packets destined or coming from the CPU
- Capture can be manipulated (buffer limit, capture limit, and display filters)
- Ability to view packet capture on switch itself via show commands

FED Punject

Limitations and Restrictions

- Capture is solely focused on CPU punted/injected traffic, not for hardware-forwarded traffic
- Caution during high CPU situations, may resulted in control-plane instability

FED Punject

Configuration Steps

Configuration Steps:

1. Define packet capture parameters (punt/inject, circular/packet limit)
2. Start the capture
3. Stop capture
4. View packet capture with any display-filters

```
Cat9k#debug platform software fed switch active punt packet-capture start
Punt packet capturing started.

Cat9k#debug platform software fed switch active punt packet-capture stop

Punt packet capturing stopped. Captured 3 packet(s)
```

FED Punject

Viewing Capture

```
Cat9k#show platform software fed switch active punt packet-capture brief

Punt packet capturing: disabled. Buffer wrapping: disabled

Total captured so far: 3 packets. Capture capacity : 4096 packets

------ Punt Packet Number: 1, Timestamp: 2000/01/28 20:18:46.797 ------

 interface : physical: TenGigabitEthernet1/0/48[if-id: 0x00000037], pal: Vlan1 [if-id:
0x00000070]

 metadata  : cause: 55 [For-us control], sub-cause: 0, q-no: 4, linktype: MCP_LINK_TYPE_IP [1]

 ether hdr : dest mac: 0100.5e00.0002, src mac: 0000.0c07.acca

 ether hdr : ethertype: 0x0800 (IPv4)

 ipv4  hdr : dest ip: 224.0.0.2, src ip: 10.122.162.131

 ipv4  hdr : packet len: 78, ttl: 1, protocol: 17 (UDP)

 udp   hdr : dest port: 1985, src port: 1985
```

FED Punject

Viewing Capture

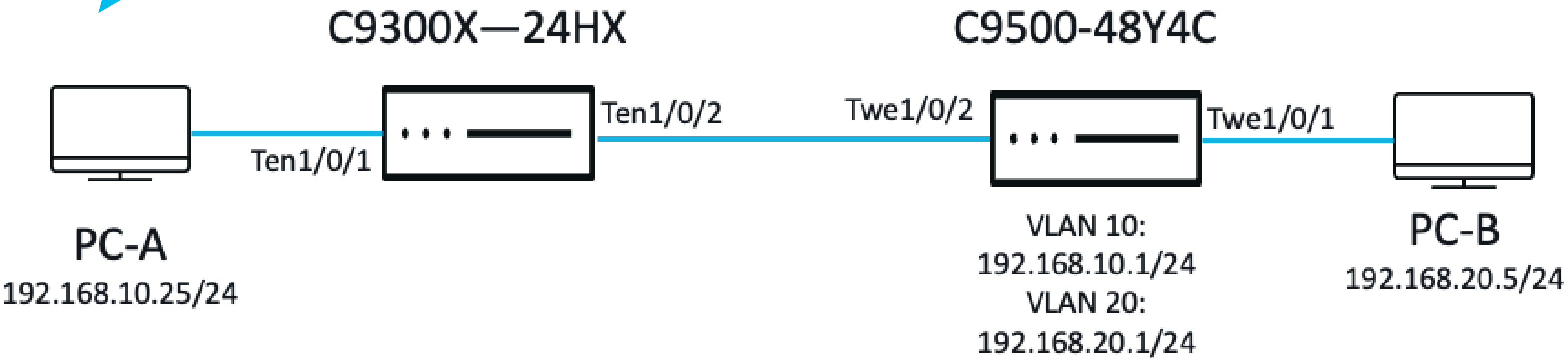
```
Cat9K#show platform software fed switch active punt packet-capture detailed

Punt packet capturing: disabled. Buffer wrapping: disabled

Total captured so far: 3 packets. Capture capacity : 4096 packets

------ Punt Packet Number: 1, Timestamp: 2000/01/28 20:18:46.797 ------

 interface : physical: TenGigabitEthernet1/0/48[if-id: 0x00000037], pal: Vlan1 [if-id: 0x00000070]

 metadata  : cause: 55 [For-us control], sub-cause: 0, q-no: 4, linktype: MCP_LINK_TYPE_IP [1]

 ether hdr : dest mac: 0100.5e00.0002, src mac: 0000.0c07.acca

 ether hdr : ethertype: 0x0800 (IPv4)

 ipv4  hdr : dest ip: 224.0.0.2, src ip: 10.122.162.131

 ipv4  hdr : packet len: 78, ttl: 1, protocol: 17 (UDP)

 udp   hdr : dest port: 1985, src port: 1985

 Packet Data Hex-Dump (length: 96 bytes) :

   01005E0000020000  0C07ACCA080045C0  004E000000000111  2BE00A7AA283E000

   000207C107C1003A  85E5000010030A64  CA00000000000000  00000A7AA281041C

   010000000A7AA283  00000000E37C6591  5DE8A9F3B420C4F7  AA912501857BCDB2
```

Real World Example

Packet Loss

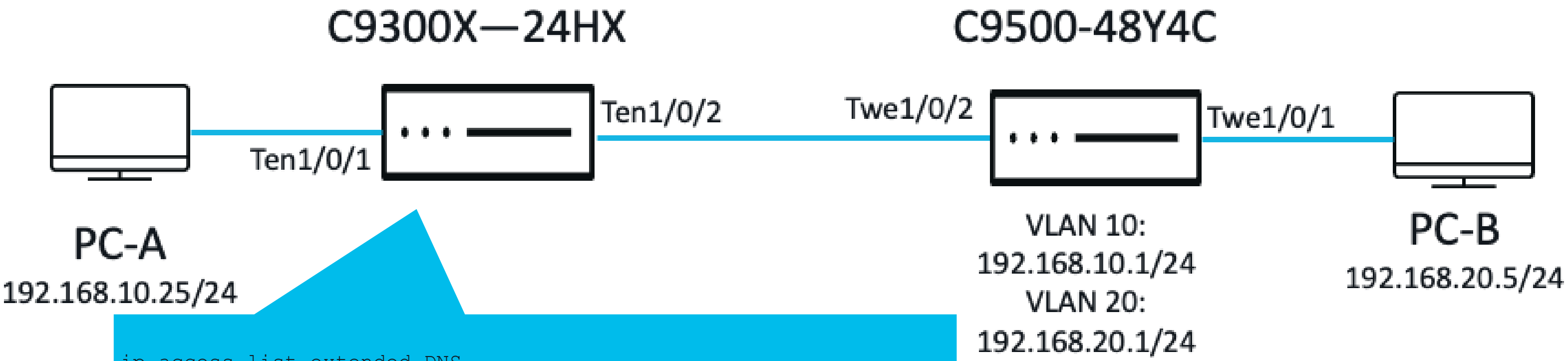
I am unable to ping 8.8.8.8 at all.

Embedded Wireshark does not show ICMP Requests coming into Twe1/0/2, other users in VLAN 10 report similar symptoms with reachability to various IP addresses.



Real World Example

Packet Loss

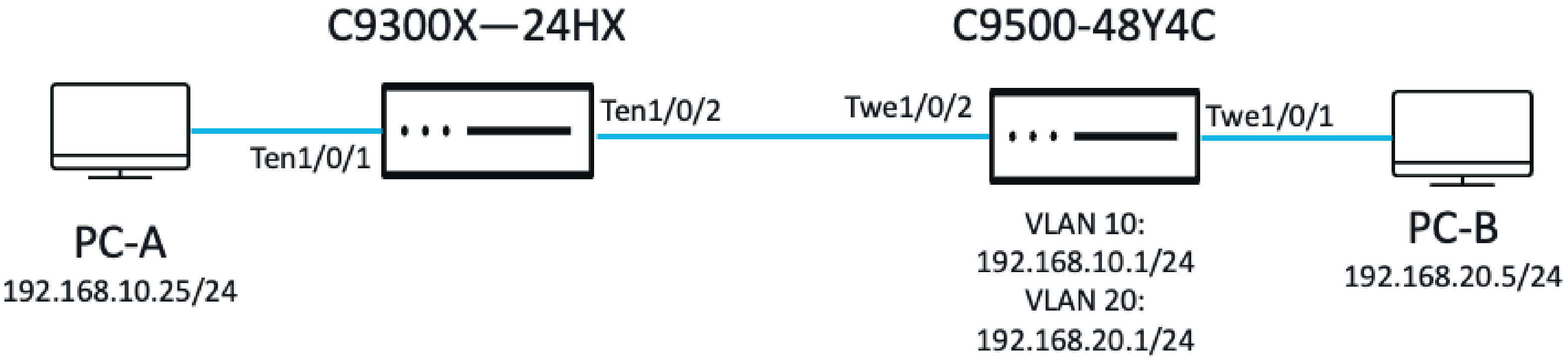


```
ip access-list extended DNS
permit ip host 192.168.10.25 host 8.8.8.8
permit ip host 8.8.8.8 host 192.16.10.25
monitor capture DNS interface Ten1/0/1 both access-list DNS start
monitor capture DNS stop
```

Real World Example

Packet Loss

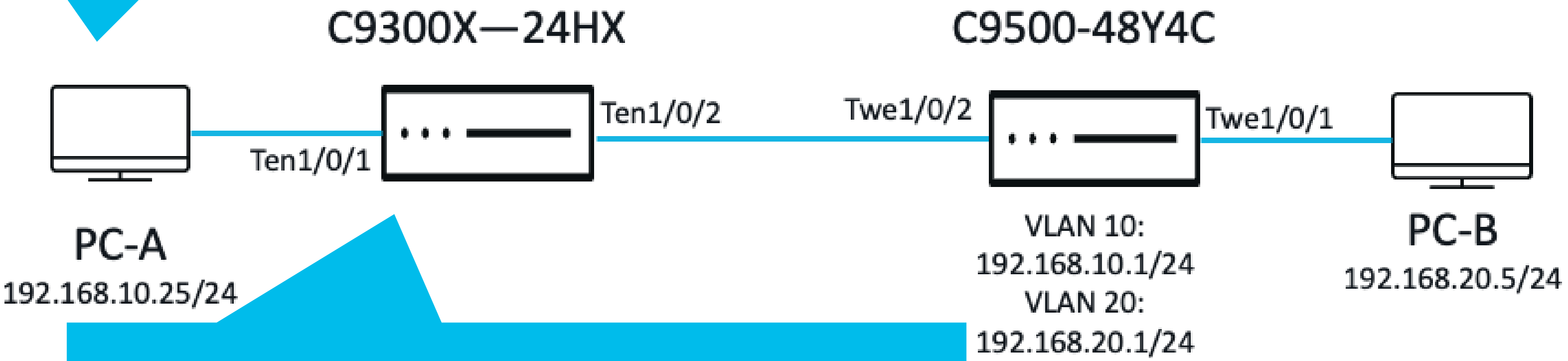
```
C9300#monitor capture DNS stop
Capture statistics collected at software:
        Capture duration - 3 seconds
        Packets received - 0
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared

Stopped capture point : DNS
```



Real World Example

Packet Loss

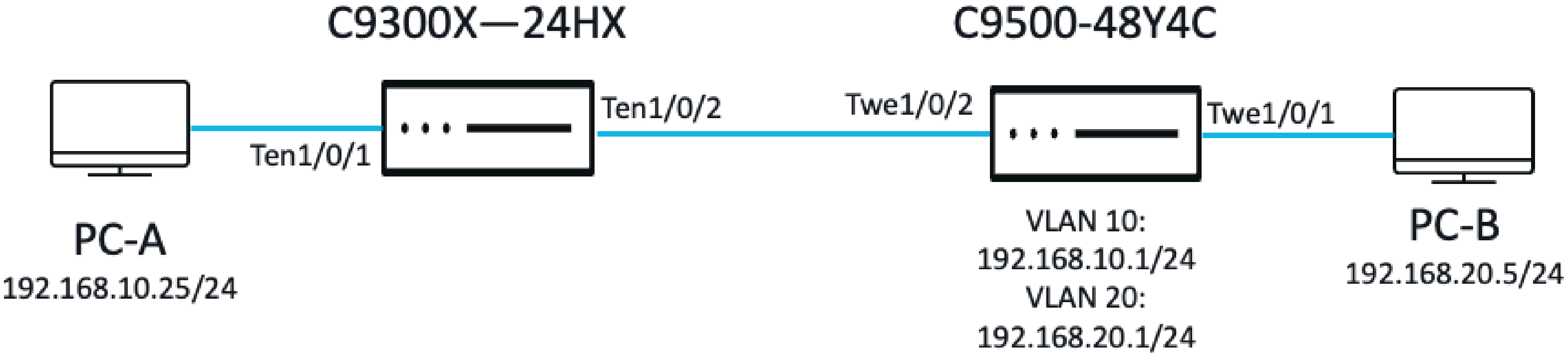
I cannot even ping 192.168.10.1!



```
monitor capture DNS interface Ten1/0/1 both match any
monitor capture DNS start
monitor capture DNS stop
```

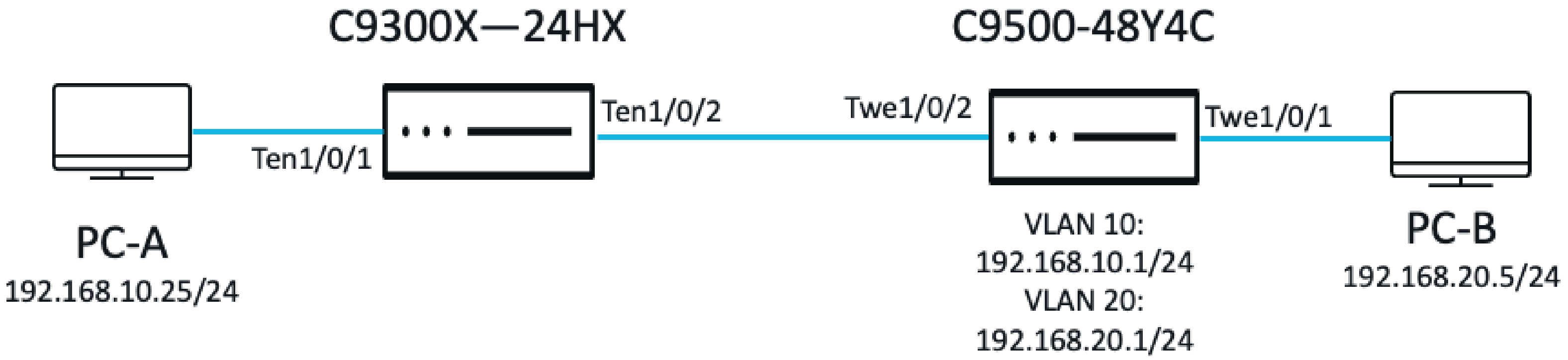
Real World Example
Packet Loss

```
C9300#monitor capture DNS stop
Capture statistics collected at software:
        Capture duration - 10 seconds
        Packets received - 110
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared
```



Real World Example
Packet Loss

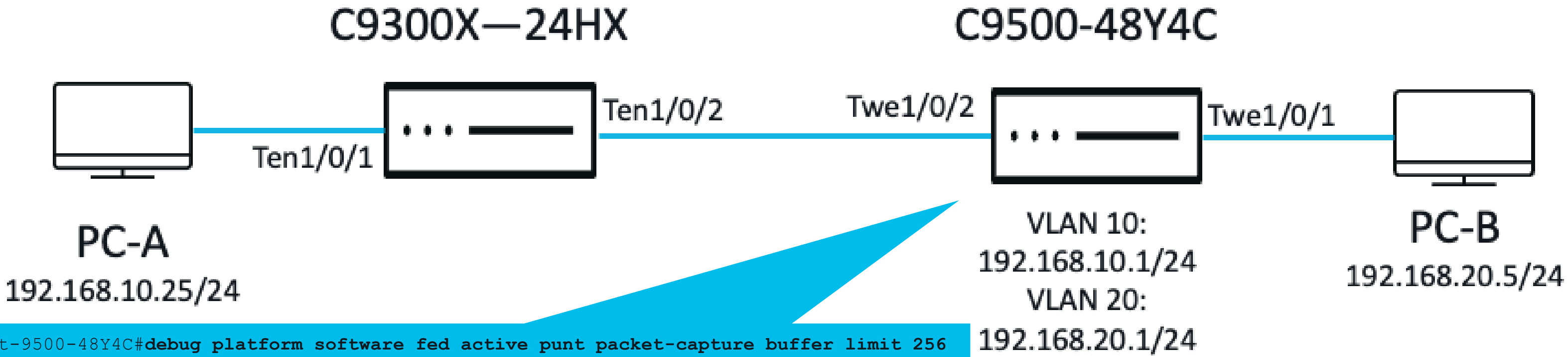
```
C9300#show monitor capture DNS buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 00:00:04:00:0e:00 -> ff:ff:ff:ff:ff:ff ARP 60 Who has 192.168.10.1? Tell 192.168.10.25
    2   2.000000 00:00:04:00:0e:00 -> ff:ff:ff:ff:ff:ff ARP 60 Who has 192.168.10.1? Tell 192.168.10.25
    3   3.999998 00:00:04:00:0e:00 -> ff:ff:ff:ff:ff:ff ARP 60 Who has 192.168.10.1? Tell 192.168.10.25
```



Real World Example

Packet Loss

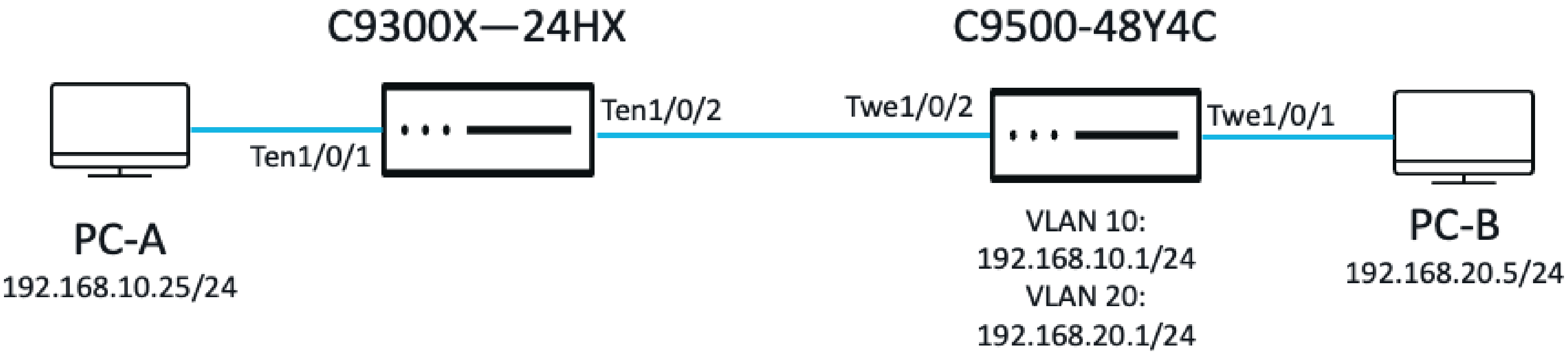
```
Catalyst-9500-48Y4C#show process cpu sorted | exclude 0.00
CPU utilization for five seconds: 12%/4%; one minute: 5%; five minutes: 1%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
  39       33273      523082         63  7.91%  2.86%  0.68%   0 ARP Input
 254      505463    23218634         21  0.07%  0.10%  0.08%   0 Spanning Tree
```



```
Catalyst-9500-48Y4C#debug platform software fed active punt packet-capture buffer limit 256
Punt PCAP buffer configure: one-time with buffer size 256...done

Catalyst-9500-48Y4C#debug platform software fed active punt packet-capture start
Punt packet capturing started.

Catalyst-9500-48Y4C#debug platform software fed active punt packet-capture stop
Punt packet capturing stopped. Captured 256 packet(s)
```

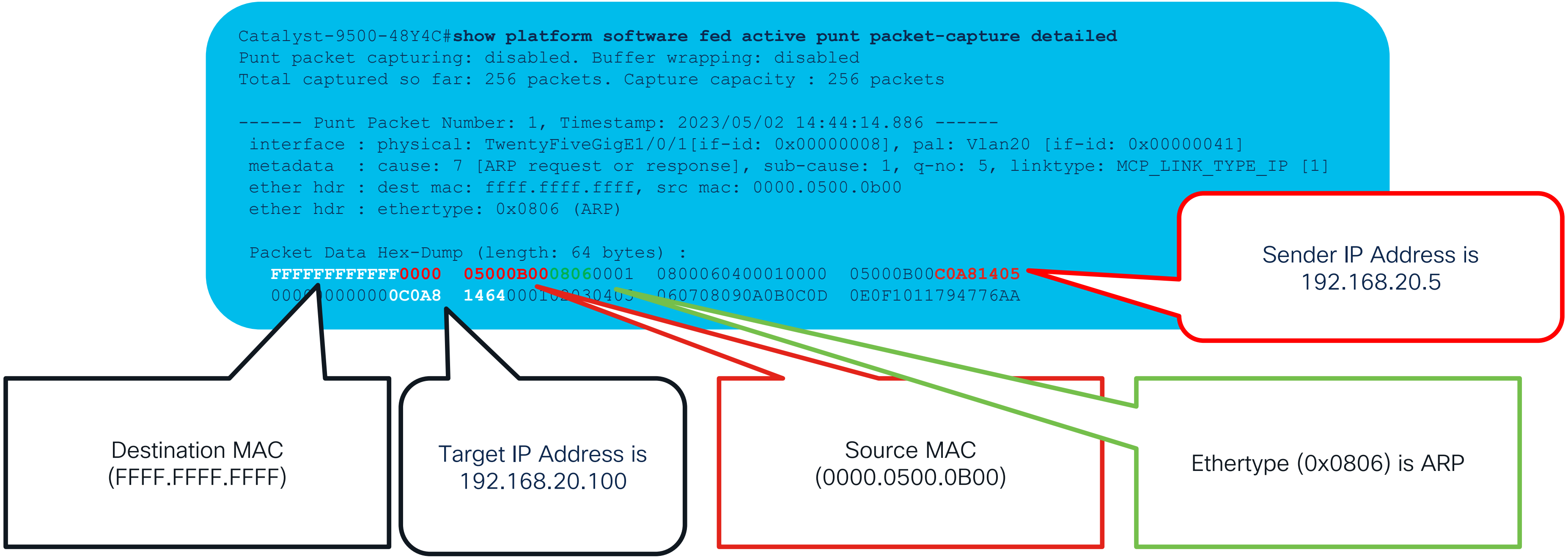
Real World Example
Packet Loss

```
Catalyst-9500-48Y4C#show platform software fed active punt packet-capture brief
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 256 packets. Capture capacity : 256 packets

------ Punt Packet Number: 1, Timestamp: 2023/05/02 14:44:14.886 ------
 interface : physical: TwentyFiveGigE1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000041]
 metadata  : cause: 7 [ARP request or response], sub-cause: 1, q-no: 5, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: ffff.ffff.ffff, src mac: 0000.0500.0b00
 ether hdr : ethertype: 0x0806 (ARP)

------ Punt Packet Number: 2, Timestamp: 2023/05/02 14:44:14.887 ------
 interface : physical: TwentyFiveGigE1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000041]
 metadata  : cause: 7 [ARP request or response], sub-cause: 1, q-no: 5, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: ffff.ffff.ffff, src mac: 0000.0500.0b00
 ether hdr : ethertype: 0x0806 (ARP)
```



Real World Example

Packet Loss



```
Catalyst-9500-48Y4C#show platform software fed active punt packet-capture detailed
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 256 packets. Capture capacity : 256 packets

------ Punt Packet Number: 1, Timestamp: 2023/05/02 14:44:14.886 ------
 interface : physical: TwentyFiveGigE1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000041]
 metadata  : cause: 7 [ARP request or response], sub-cause: 1, q-no: 5, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: ffff.ffff.ffff, src mac: 0000.0500.0b00
 ether hdr : ethertype: 0x0806 (ARP)

 Packet Data Hex-Dump (length: 64 bytes) :
   FFFFFFFFFFFF0000  05000B0008060001  0800060400010000  05000B00C0A81405
   000000000000C0A8  1464000102030405  060708090A0B0C0D  0E0F1011794776AA
```

Sender IP Address is 192.168.20.5

Destination MAC (FFFF.FFFF.FFFF)

Target IP Address is 192.168.20.100

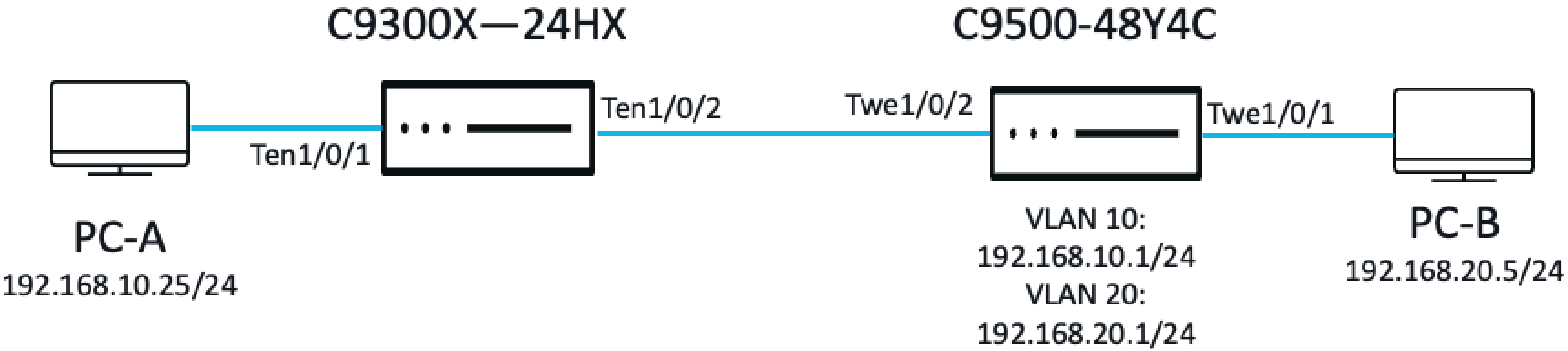
Source MAC (0000.0500.0B00)

Ethertype (0x0806) is ARP

Real World Example

Packet Loss

```
Catalyst-9500-48Y4C#show platform hardware fed active qos queue stats internal cpu policer

                         CPU Queue Statistics
============================================================================================
                                                  (default)  (set)     Queue        Queue
QId PlcIdx  Queue Name                   Enabled   Rate      Rate      Drop(Bytes)  Drop(Frames)
--------------------------------------------------------------------------------------------
0    11     DOT1X Auth                    Yes     1000      1000      0            0
1    1      L2 Control                    Yes     2000      2000      0            0
2    14     Forus traffic                 Yes     4000      4000      0            0
3    0      ICMP GEN                      Yes     750       750       0            0
4    2      Routing Control               Yes     5500      5500      0            0
5    14     Forus Address resolution      Yes     4000      4000      70179268     1096471
<snip>
```



C9300X—24HX

C9500-48Y4C

Ten1/0/1

Ten1/0/2

Twe1/0/2

Twe1/0/1

PC-A
192.168.10.25/24

VLAN 10:
192.168.10.1/24
VLAN 20:
192.168.20.1/24

PC-B
192.168.20.5/24

Switched Port Analyzer (SPAN) Tools

SPAN

Overview

- Switched Port Analyzer
- Provides the ability to mirror traffic from one port, group of ports, vlan, etc. to a local or remote destination
- Universally supported on the Catalyst 9000-family of switches*

*C9200-models do not support ERSPAN

SPAN
Advantages and Benefits

- On-board port-mirroring capability
- No impact to network traffic
- Destination port can inject traffic from network security devices
- Not subject to internal rate-limiter

SPAN
Variants

Local SPAN

- Most trustworthy
- Often used by TAC

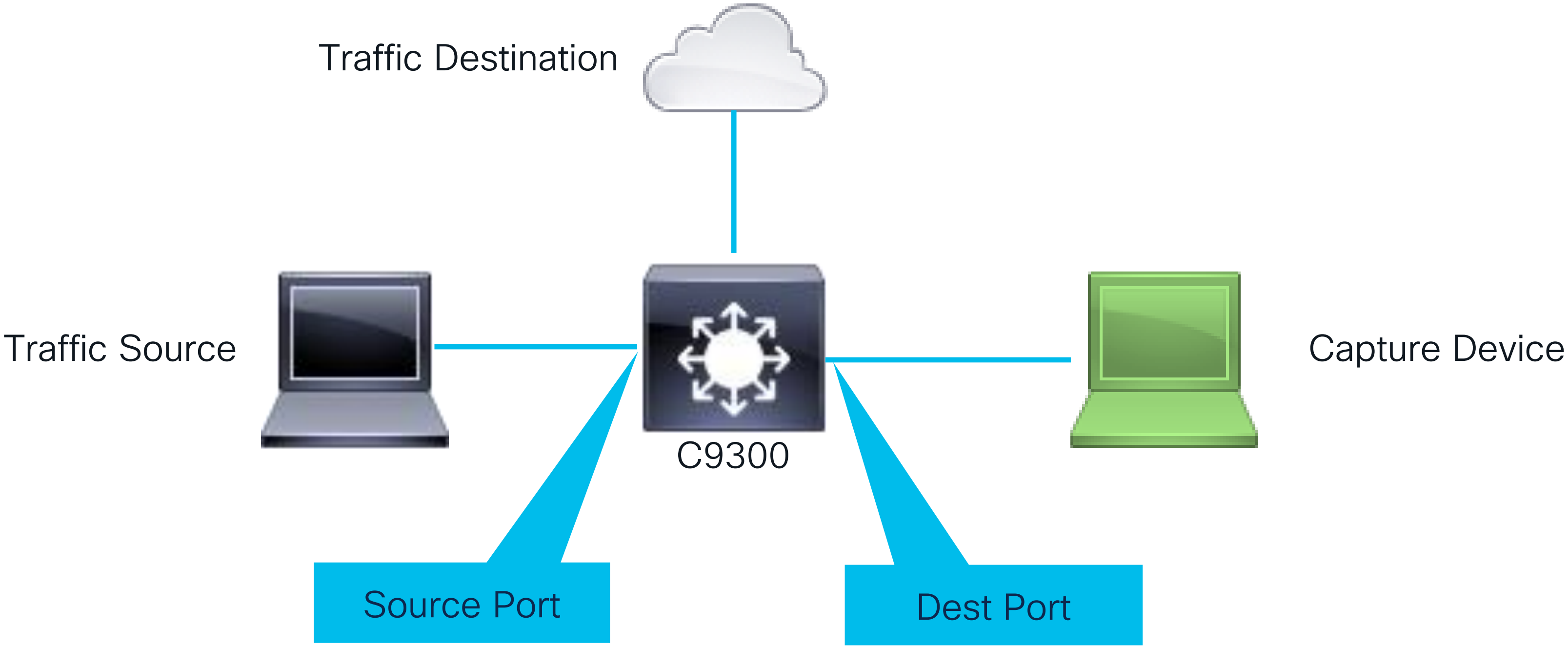
Remote SPAN (RSPAN)

- Allows forwarding of monitored traffic to a distant, L2-adjacent destination
- Remote-span VLAN must be carried on all trunks between source and destination

ER-SPAN (ERSPAN)

- Uses GRE to encapsulate monitored traffic
- Allows forwarding of monitored traffic over L3 boundaries

SPAN
Local SPAN



SPAN

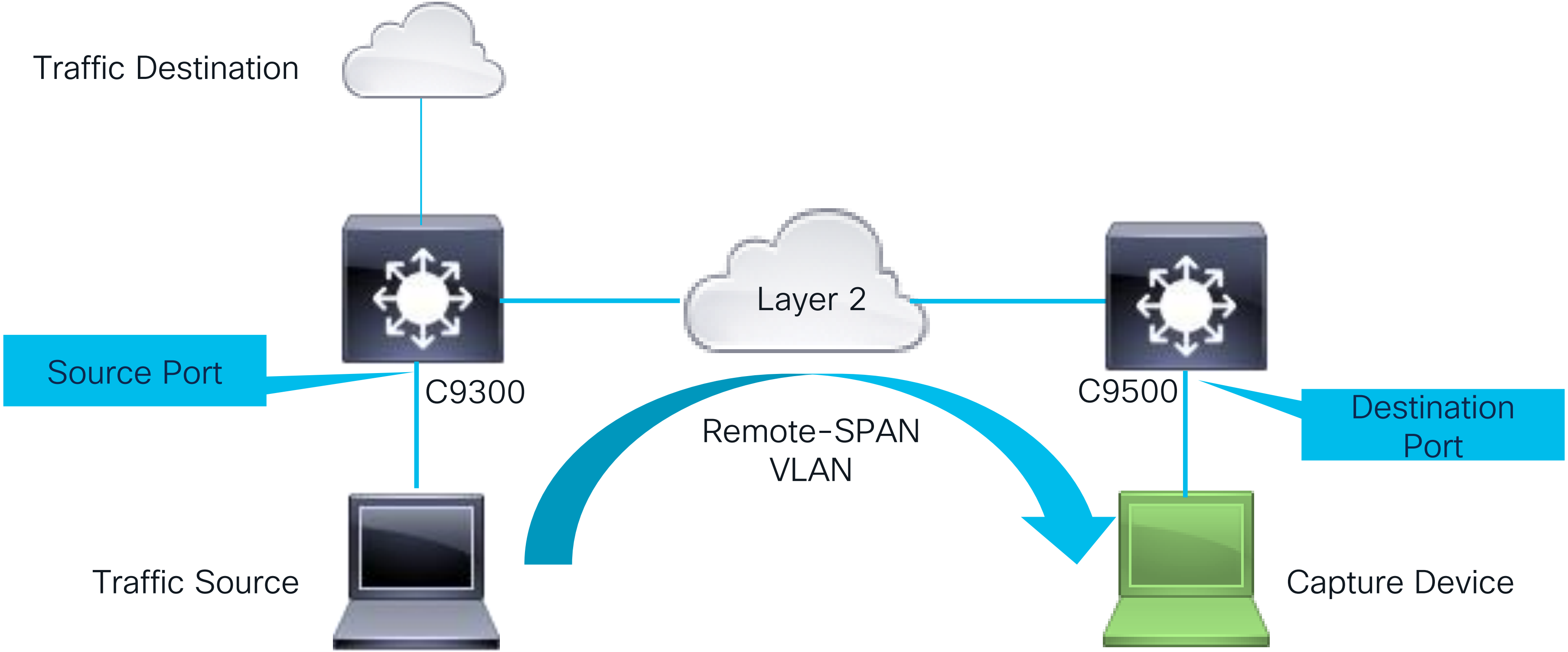
Configuration Steps

Local SPAN Configuration Example

```
C9300(config)#monitor session 1 source interface tenGigabitEthernet 1/0/1 both
C9300(config)#monitor session 1 destination interface te1/0/3
```

SPAN

Remote SPAN (RSPAN)



SPAN

Configuration Steps

Remote SPAN Configuration Example:

Source Session -

```
C9300(config)#vlan 33
C9300(config-vlan)#remote-span
```

Destination Session -

```
C9300(config)#vlan 33
C9300(config-vlan)#remote-span
```

*remote-span VLAN must exist on source and destination switches, as well as on any switch(es) in between

SPAN

Configuration Steps

Remote SPAN Configuration Example:

Source Session -

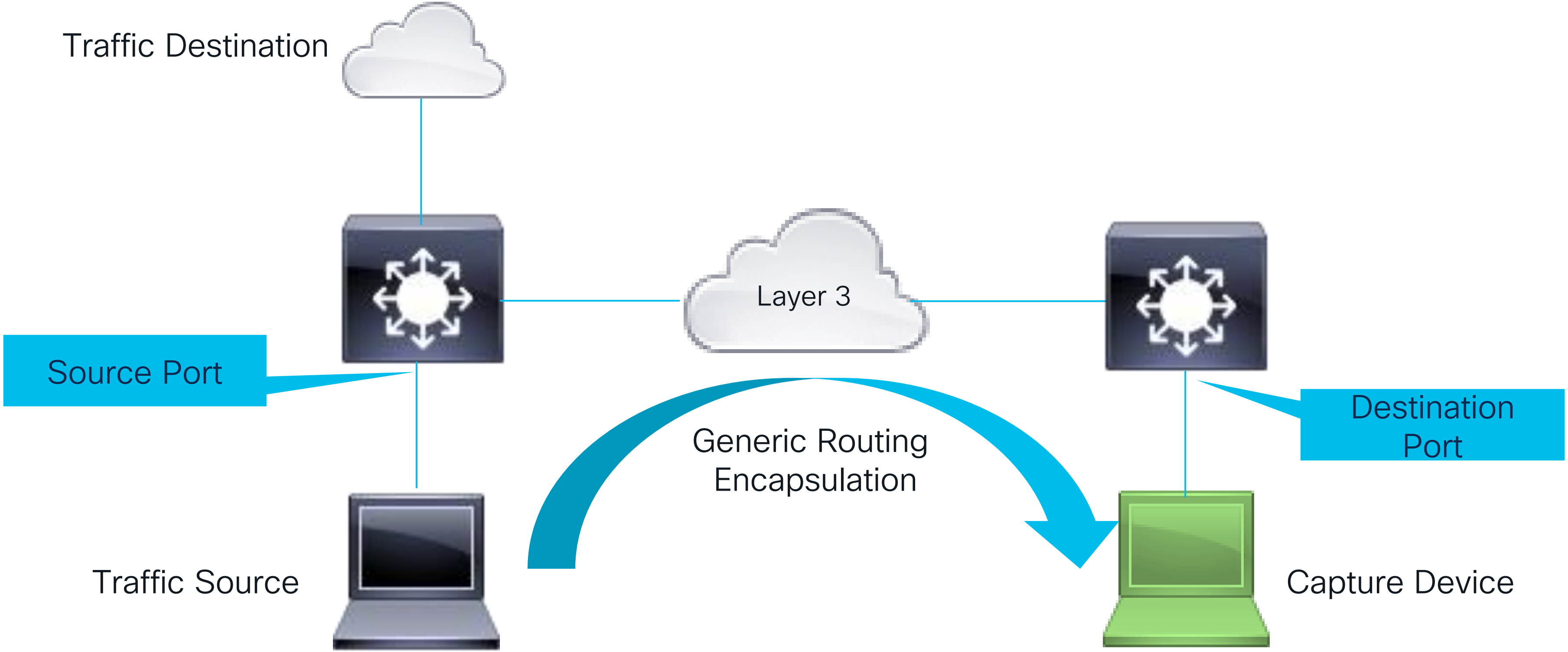
```
C9500(config)#monitor session 1 source interface twentyFiveGigE 1/0/3 tx
C9500(config)#monitor session 1 destination remote vlan 33
```

Destination Session -

```
C9300(config)#monitor session 1 source remote vlan 33
C9300(config)#monitor session 1 destination interface tenGigabitEthernet 1/0/3
```

SPAN

Encapsulated Remote SPAN



SPAN

Configuration Steps

Encapsulated Remote SPAN Configuration Example:

Source Session:

```
C9300(config)#monitor session 33 type erspan-source
C9300(config-mon-erspan-src)#source interface te1/0/2 rx
C9300(config-mon-erspan-src)#destination
C9300(config-mon-erspan-src-dst)#erspan-id 33
C9300(config-mon-erspan-src-dst)#ip address 10.10.10.94
C9300(config-mon-erspan-src-dst)#origin ip address 10.10.10.93
C9300(config-mon-erspan-src-dst)#exit
C9300(config-mon-erspan-src)#no shutdown
```

SPAN
Configuration Steps

Encapsulated Remote SPAN Configuration Example:

Destination Session:

```
Catalyst-9400(config)#monitor session 33 type erspan-destination
Catalyst-9400(config-mon-erspan-dst)#destination interface Gi1/0/24
Catalyst-9400(config-mon-erspan-dst)#source
Catalyst-9400(config-mon-erspan-dst-src)#erspan-id 33
Catalyst-9400(config-mon-erspan-dst-src)#ip address 10.10.10.94
Catalyst-9400(config-mon-erspan-dst-src)#exit
Catalyst-9400(config-mon-erspan-dst)#no shutdown
```

SPAN

Filtering – Local SPAN

```
Catalyst-9300X-24HX#show monitor session 2
Session 2
---------
Type                     : Local Session
Source VLANs             :
    Both                 : 2
Destination Ports        : Te1/0/24
    Encapsulation        : Native
          Ingress        : Disabled
```

```
C9300(config)#monitor session 2 filter ?
  ip    Specify IP Access control rules
  ipv6  Specify IPv6 Access control rules
  mac   Specify MAC Access control rules
  vlan  SPAN filter VLAN
```

```
C9300(config)#monitor session 2 filter ip access-group MY_ACL
```

SPAN

Filtering - RSPAN

```
C9500#show monitor session 1
Session 1
---------
Type                     : Remote Source Session
Source Ports             :
    Both                 : Twe1/0/3
Dest RSPAN VLAN          : 33
```

```
C9500(config)#monitor session 1 filter ?
  ip     Specify IP Access control rules
  ipv6   Specify IPv6 Access control rules
  mac    Specify MAC Access control rules
  vlan   SPAN filter VLAN
```

```
C9500(config)#monitor session 1 filter mac access-group MY_MACL
```

SPAN

Filtering - ERSPAN

```
Catalyst-9300X-24HX#show monitor session 33
Session 33
----------
Type                        : ERSPAN Source Session
Status                      : Admin Enabled
Description                 : TO-9400
Source Ports                :
    Both                    : Te1/0/2
Destination IP Address      : 10.10.10.94
MTU                         : 9000
Destination ERSPAN ID       : 33
Origin IP Address           : 10.10.10.93
```

SPAN

Filtering - ERSPAN

```
Catalyst-9300X-24HX(config)#monitor session 33 filter ip access-group MY_ACL
% Please use sub-mode form of CLI to configure this session
```

```
Catalyst-9300X-24HX(config)#monitor session 33 type erspan-source
Catalyst-9300X-24HX(config-mon-erspan-src)#?
Monitor sess type erspan source config commands:
  description  Properties for this session
  destination  Specify Destination and their properties
  exit         Exit from monitor erspan source session mode
  filter       SPAN filter VLAN
  header-type  ERSPAN header-type for encapsulation. Default is type 2
  no           Negate a command or set its defaults
  shutdown     Shutdown this session
  source       SPAN source Interface/VLAN

Catalyst-9300X-24HX(config-mon-erspan-src)#filter ip access-group MY_ACL
```

SPAN

Validate

```
C9300#show monitor session all | include Session
Session 1
Type                     : Remote Destination Session
Session 2
Type                     : Local Session
Session 33
Type                     : ERSPAN Source Session
```

```
C9300#show monitor session 1
Session 1
---------
Type                     : Remote Destination Session
Source RSPAN VLAN        : 33
Destination Ports        : Te1/0/7
    Encapsulation        : Native
          Ingress        : Disabled
```

SPAN

Validate

```
Catalyst-9300X-24HX#show monitor session 33 detail
Session 33
----------
Type                        : ERSPAN Source Session
Status                      : Admin Enabled
<snip>
Destination IP Address      : 10.10.10.94
Destination IPv6 Address : None
Destination IP VRF          : None
MTU                         : 9000
Destination ERSPAN ID       : 33
Origin IP Address           : 10.10.10.93
Origin IPv6 Address         : None
IP QOS PREC                 : 0
IPv6 Flow Label             : None
IP TTL                      : 255
IPV6 TTL                    : 255
ERSPAN header-type          : None
```

SPAN

Validate

```
Catalyst-9300X-24HX#show monitor session ?
  <1-66>                SPAN session number
  all                   Show all SPAN sessions
  erspan-destination    Show only Destination ERSPAN sessions
  erspan-source         Show only Source ERSPAN sessions
  local                 Show only Local SPAN sessions
  range                 Show a range of SPAN sessions in the box
  remote                Show only Remote SPAN sessions
```

```
Catalyst-9300X-24HX#show monitor session remote detail
Session 1
---------
Type                     : Remote Destination Session
Description              : -
Source Ports             :
    RX Only              : None
    TX Only              : None
    Both                 : None
<snip>
```

Event Trace, Binary Trace, TLS Syslog

Event Trace

Event-Trace

Basics

- Event-Trace allows for persistent logging of processes within IOSd
  - Human-readable (unlike archived binary traces)
  - Survives reload (unlike common Syslogging)
  - Augments existing logging to help provide a more complete picture
  - Little/no danger of resource drain

- Processes supporting event-trace include:
  - Spanning-Tree
  - Routing Protocols (EIGRP, BGP, ISIS, etc.)
  - UDLD
  - L2VPN, L3VPN
  - CEF

Event-Trace

Configure process monitoring

```
C9300(config)#monitor event-trace ?
  ac                AC traces
  acl               ACL Traces
  adjacency         Adjacency Events
  <snip>
  stacktrace        Display stack trace stored with event trace entries
  stp               STP Traces
  timestamps        Format of event trace timestamps
  tracking          Tracking traces
  tunnel            tunnel event trace
  udld              UDLD Traces
  vlan              VLAN Traces
  xconnect          old alias for l2vpn traces
  xdr               XDR traces
```

*As IOS-XE is platform independent, not all processes listed are supported on Catalyst switches. See the relevant configuration guide for details.

Event-Trace

Configure Parameters

```
C9300(config)#monitor event-trace stp ?
  bpdu        STP Bpdu traces
  critical    STP Critical traces
  errors      STP Error traces
  events      STP Event traces

C9300(config)#monitor event-trace stp critical ?
  dump-file   Set name of trace dump file
  size        Set size of trace
  stacktrace  Trace call stack at tracepoints; clear the trace buffer first
  <cr>        <cr>

C9300(config)#monitor event-trace stp critical size ?
  <1-1000000>  Number of entries in trace

C9300(config)#monitor event-trace stp critical size 1000000
```

Event-Trace

Show Results

```
C9300#show monitor event-trace stp critical ?
  all           Show all the traces in current buffer
  back          Show trace from this far back in the past
  clock         Show trace from a specific clock time/date
  from-boot     Show trace from this many seconds after booting
  instance      Filter traces based on the vlan/instance
  latest        Show latest trace events since last display
  parameters    Parameters of the trace
```

```
C9300#show monitor event-trace stp critical all
*May 24 13:08:45.136: STP root bridge changed to 5c71.0d4b.1c00 root path cost
                      20000
*May 24 13:08:45.136: STP port role changed to root for  Te1/0/3
*May 24 13:08:45.136: STP port role changed to designated for Te1/0/2
*May 24 13:08:45.136: Superior bpdu received on :Te1/0/3
```

Event-Trace

Show Results

```
C9300#show monitor event-trace stp critical instance ?
  <0-4094>  VLAN ID /Instance id of stp to  be filtered

C9300#show monitor event-trace stp critical instance 2 ?
  all          Show all the traces in current buffer
  back         Show trace from this far back in the past
  clock        Show trace from a specific clock time/date
  from-boot    Show trace from this many seconds after booting
  latest       Show latest trace events since last display
```

Binary Tracing

Binary Tracing

Always-on persistent logging

- Survives reload - think of it as a blackbox recorder on a plane
- Active traces occupy 1MB of volatile memory before rotating to persistent filesystem

Archived within “crashinfo:/”

- Archive can be created with “**request platform software trace archive**” utility
- Archives are in binary (.bin) format. Not readable w/ text viewer
- TAC will often ask for an archive – use system-generated filename

```
C9300#request platform software trace archive
Creating archive file [flash:C9300_1_RP_0_trace_archive-20230504-143034.tar.gz]
Done with creation of the archive file:
[flash:C9300_1_RP_0_trace_archive-20230504-143034.tar.gz]
C9300#
```

Binary Tracing

Readable traces can be displayed via CLI

- **“show platform software trace message <process>”** (scheduled for deprecation – IOS XE 17.9.x)
- **“show logging process <process>”** – current syntax

```
C9300#show logging process iosrp
Logging display requested on 2023/05/04 15:11:48 (UTC) for Hostname: [C9300],
Model: [C9300X-24HX], Version: [17.09.01], SN: [FOC263569FP], MD_SN: [FOC2641Y2MK]

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis 1 ...
Unified Decoder Library Init .. DONE
Found 1 UTF Streams

2023/05/04 15:02:52.151852847 {iosrp_R0-0}{1}: [parser_cmd] [5160]:
(note): id= console@console:jason= cmd: 'show ip route' SUCCESS 2023/05/04 15:01:16.438 UTC
(note): id= console@console:jason= cmd: 'show ip ospf neighbors' SUCCESS 2023/05/04 15:01:18.438 UTC
<snip>
```

Binary Tracing

- Traces can be written to file for offline analysis
  - For best results, run “**request platform software trace rotate all**” first- Moves in-memory traces to crashinfo:
  - Use the “to-file” argument to export output

```
C9300#show logging process iosrp to-file flash:iosrp_traces.txt
Logging display requested on 2023/05/04 15:31:57 (UTC) for
Hostname: [C9300], Model: [C9300X-24HX], Version: [17.09.01]

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis 1 ...
Files being merged in the background, please check [/flash/iosrp_traces.txt] output file
```

Binary Tracing
Processes of Interest

- Example processes of interest
  - **IOSRP** – IOS Route Processor. “Brain” of the operating system. Most anything creating a syslog will be found here
  - **FED** – Forwarding Engine Driver. Manages hardware programming and forwarding.
  - **SMD** – Session Manager Daemon. Controls front-panel authentication and AAA functions such as dot1x, MAB and RADIUS
  - **SIF** – Stack interface. Significant with stacked systems. Manages the connection between the stacking hardware and ASIC
  - **Chassis-Manager** – PoE, Fan Tray, PSUs.

Binary Tracing

Setting Trace Level

- Many logging ‘levels’ are supported
  - Emergency, Error, Warning, Notice, Info, Debug, Verbose, Noise
  - Use “**set platform software trace switch active r0 <process> <level>**” to enable more (or less) granular tracing
  - “Classic” IOS debugging can also be used to enable related traces in some cases
    - “debug dot1x all”, “debug aaa authentication”, “debug radius”
    - Processes subordinate to IOS components often do not appear in syslog output

Binary Tracing

Setting trace level

```
C9300#debug dot1x all

All Dot1x debugging is on
```

```
C9300#set platform software trace smd switch active r0 dot1x ?
  debug        Debug messages
  emergency    Emergency possible message
  error        Error messages
  info         Informational messages
  noise        Maximum possible message
  notice       Notice messages
  verbose      Verbose debug messages
  warning      Warning messages
```

Binary Tracing

Example – Determining source of configuration change

- “Show history” provides a limited view (last 10 commands normally)
- The “**parser_cmd**” subcomponent of “iosrp” tracks all commands entered by any user
- Use “**show logging process iosrp | include parser_cmd**” to view entries
  - Note that only messaging stored in volatile memory are viewable (1MB)
  - Tracelog archive will provide extended history- check with TAC

Binary Tracing - Example

Who Deleted VLAN 97? Was it Nathan or Jason?

```
C9300#show logging process iosrp reverse | in parser_cmd
<snip>: [parser_cmd] [5160]: (note): id= console@console:user=
cmd: 'show logging process iosrp | in parser_cmd' SUCCESS 2023/05/04 16:07:46.498 UTC
<snip>:[parser_cmd] [5160]: (note): id= console@console:user=
cmd: 'end' SUCCESS 2023/05/04 16:07:40.049 UTC
<snip>:[parser_cmd] [5160]: (note): id= 10.202.17.186@vty0:user=jason
cmd: 'show inv' SUCCESS 2023/05/04 16:07:32.499 UTC
<snip>: [parser_cmd] [5160]: (note): id= 10.202.17.186@vty0:user=jason
cmd: 'sh ver' SUCCESS 2023/05/04 16:07:29.989 UTC

<snip>:[parser_cmd] [5160]: (note): id= 10.202.17.182@vty0:user=nathan
cmd: 'no int vlan 97' SUCCESS 2023/05/04 16:07:17.688 UTC
user=nathan cmd: 'exit' SUCCESS 2023/05/04 16:06:48.205 UTC
```

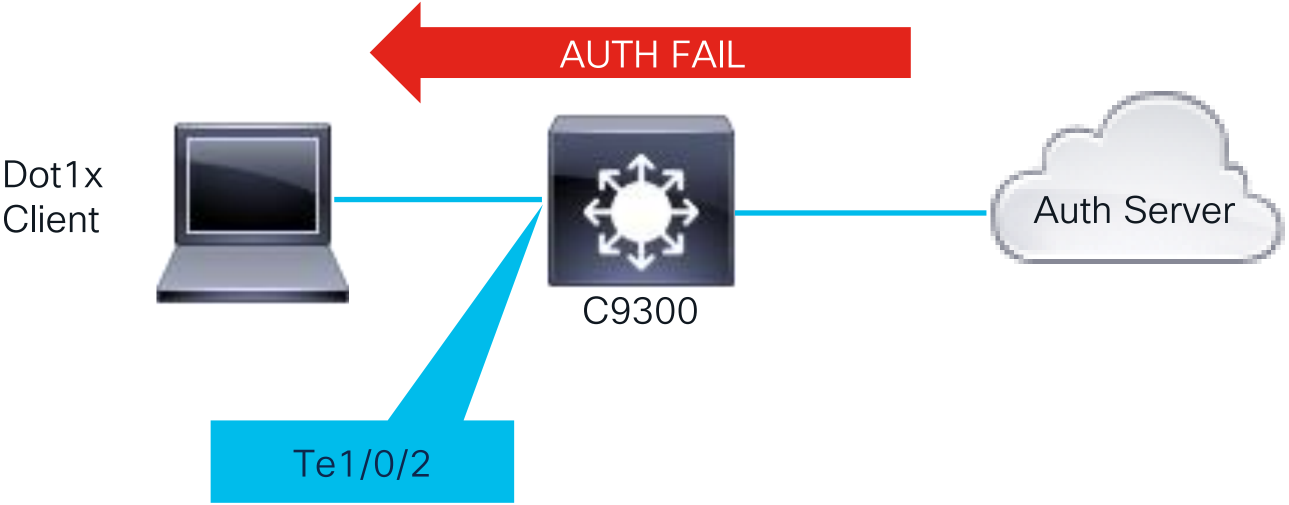
GOT ‘EM

Binary Tracing

Example

Authentication problems

- MAB and dot1x are managed by SMD
- Output will be chatty – write to file for ease of analysis



Binary Tracing

Example

Best practices when leveraging traces:

- Set the specific process and component to the desired level (debug or noisier)
- Rotate traces prior to your recreating your AAA failure
- Perform the test then collect the logs

```
C9300#set platform software trace smd switch active r0 all-modules debug
C9300#request platform software trace rotate all
<AAA test complete>
C9300#show logging process smd to-file flash:smd_tracelogs.txt
Logging display requested on 2023/05/04 16:50:56 (UTC) for Hostname:
[C9300], Model: [C9300X-24HX], Version: [17.09.01], SN: [FOC263569FP], MD_SN: [FOC2641Y2MK]

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis 1 ...
Files being merged in the background, please check [/flash/smd_tracelogs.txt] output file

Logging display requested on 2023/05/24 16:54:15 (UTC) for Hostname: [C9300],
Model: [C9300X-24HX], Version: [17.09.01], SN: [FOC263569FP], MD_SN: [FOC2641Y2MK]
```

Binary Tracing

Example

```
C9300#more flash:smd_tracelogs.txt | in 1/0/2
<snip> [dot1x] [22624]: (info): [5c5a.c761.4bc2:Te1/0/2] EAPOL packet sent to client
<snip> [dot1x] [22624]: (info): [5c5a.c761.4bc2:Te1/0/2] Received an EAP Timeout
<snip> [dot1x] [22624]: (info): [5c5a.c761.4bc2:Te1/0/2] Entering idle state
<snip> [dot1x] [22624]: (info): [5c5a.c761.4bc2:Te1/0/2] Posting AUTH_TIMEOUT on Client
<snip> [errmsg] [22624]: (note): %DOT1X-5-FAIL: R0/0: sessmgrd:
Authentication failed for client (5c5a.c761.4bc2) with reason (No Response from Client)
on Interface Te1/0/2 AuditSessionID 13A37A0A00000123E7707A1F
<snip> [auth-mgr] [22624]: (info): [5c5a.c761.4bc2:Te1/0/2]
Authc failure from Dot1X, Auth event no-response
<snip> [auth-mgr] [22624]: (info): [5c5a.c761.4bc2:Te1/0/2]
Method dot1x changing state from 'Running' to 'Authc Failed'
```

*Since we can confirm the switch sends EAPoL to client,
yet we generate an EAP timeout, ensure the client is transmitting EAPoL

Binary Tracing

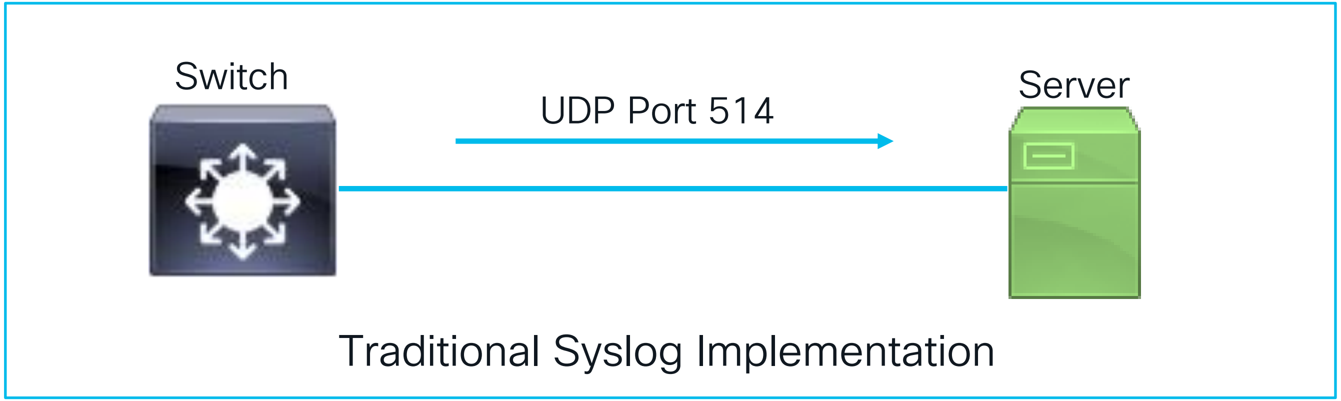
Return trace level to default

```
C9300#set platform software trace smd switch active r0 all-modules ?
  debug      Debug messages
  emergency  Emergency possible message
  error      Error messages
  info       Informational messages
  noise      Maximum possible message
  notice     Notice messages
  verbose    Verbose debug messages
  warning    Warning messages
C9300#set platform software trace smd switch active r0 all-modules notice
C9300#show platform software trace level smd switch active r0
Module Name                     Trace Level
-----------------------------------------------
aaa                             Notice
aaa-acct                        Notice
aaa-admin                       Notice
<snip>
```

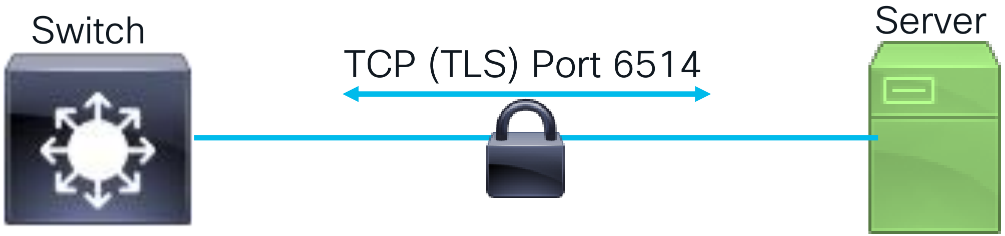
* “undebug all” may also be used

Transport Layer Security (TLS) Syslog

Transport Layer Security (TLS) Syslog



Traditional Syslog Implementation



TLS Syslog Implementation

Transport Layer Security (TLS) Syslog

Basics

- Defined in RFC 5425
- Catalyst 9K support in IOS XE Amsterdam (17.2.x) and beyond
- Provides a method for secure sending of syslogs from switch to server
- Allows for confidentiality, integrity of messages and mutual authentication

TLS Syslog

Configuration Steps

- Install Certificate on the Catalyst Switch
  - Process is the same for other utilities requiring certs
  - Refer to relevant platform/code configuration guide for details
- Install Certificate on the Syslog Server
  - Follow guidelines specific to server
- Configure the Switch for Syslog TLS

TLS Syslog

Configure the Switch for TLS Syslogging

Configure logging profile on the Catalyst Switch:

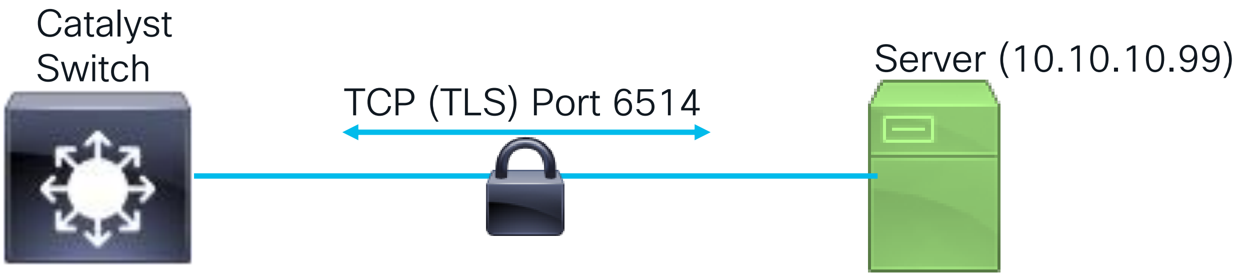
```
Catalyst-9400(config)#logging tls-profile SYSLOG-TLS

Catalyst-9400(config-tls-profile)#?
TLS configurations for secure syslog connection:
  ciphersuite             Secure ciphersuite for syslog connection
  client-id-trustpoint    Trustpoint for syslog client ID certificate
  default                 Set a command to its defaults
  exit                    Exit from TLS profile configuration sub mode
  no                      Negate a command or set its defaults
  tls-version             TLS version for syslog connection
Catalyst-9400(config-tls-profile)#tls-version TLSv1.2
Catalyst-9400(config-tls-profile)#client-id-trustpoint TLS-SYSLOG-TRUSTPOINT
Catalyst-9400(config-tls-profile)#end
```

TLS Syslog

Configuration Steps

Configure Logging to the Syslog TLS Server

```
C9400(config)#logging host 10.10.10.99 transport tls profile SYSLOG-TLS
```



*Certificate Installed
*TLS-Profile Configured
*Logging method defined

*Certificate installed
* Supports TLS syslogging

TLS Syslog

Validation

```
Catalyst-9400#show logging
Syslog logging: enabled
<snip>

    Trap logging: level informational, 141 message lines logged
        Logging to 10.10.10.99  (tls port 6514, audit disabled,
              link down),
              0 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number disabled
              filtering disabled
              tls-profile: SYSLOG-TLS
        Logging Source-Interface:       VRF Name:
    TLS Profiles:
        Profile Name: SYSLOG-TLS
              Ciphersuites: Default
              Trustpoint: TLS-SYSLOG-TRUSTPOINT
              TLS version: TLSv1.2
```

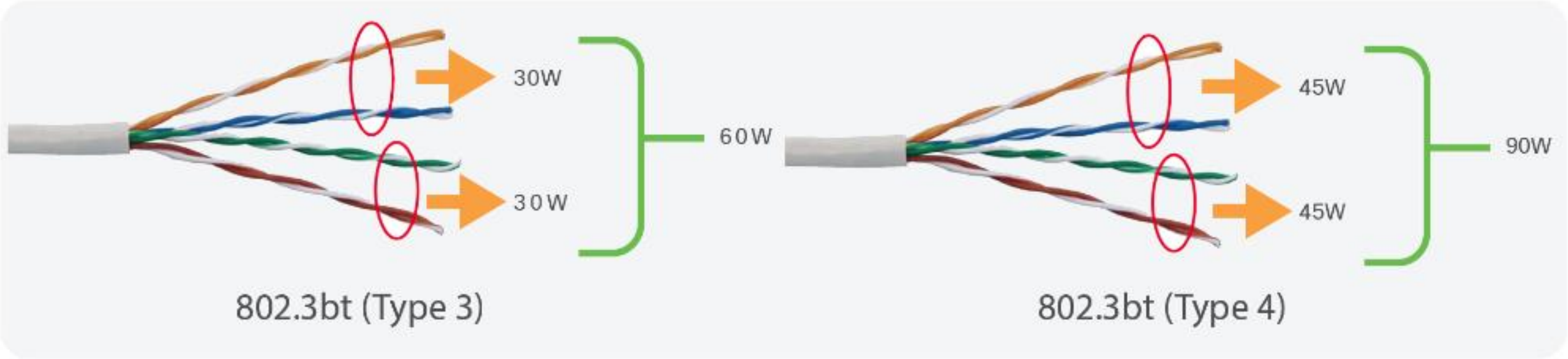
Power Over Ethernet (PoE)

Power over Ethernet (PoE)

Evolution of PoE Standards

- IEEE 802.3af
  - Original IEEE standard, adopted in 2003
  - Power Sourcing Equipment (PSE) provides up to 15.4W (12.95W available)
- IEEE 802.3at
  - Established in 2009. Also known at PoE+
  - 30W of power provided by switch; 25.5W delivered to Powered Device (PD)
- IEEE 802.3bt
  - Established in 2018. 90W at the PSE
  - Supports type 3 (51W available) and type 4 (71.3W available) PDs

IEEE 802.3bt - Type 3 and 4



- Cisco Universal Power over Ethernet (UPOE) supports 60W at the PSE
- UPOE+ Supports both type 3 and type 4

Catalyst 9000 UPOE+ Support

UPOE+ is supported on the following platforms:

C9300 Series Switches:

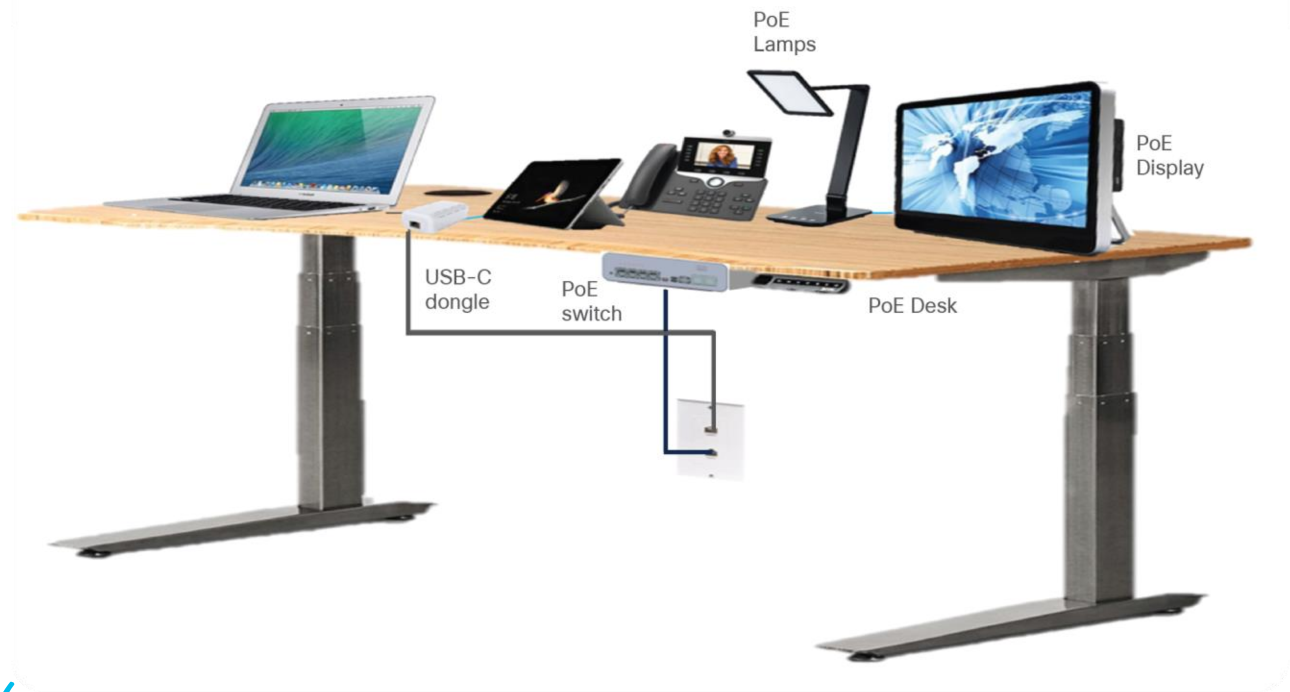
- C9300X-48HX
- C9300-48HXN
- C9300X-24HX
- C9300-48H
- C9300-24H

C9400 Series Line Cards:

- C9400-LC-48HX
- C9400-LC-48HN
- C9400-LC-48H

UPOE+

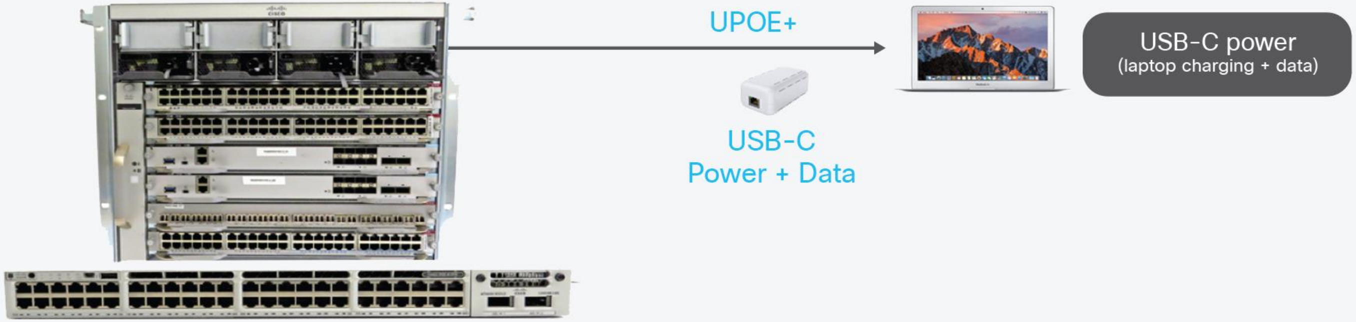
UPOE+ combines the IEEE 802.3bt standard and Cisco UPOE



90W Use Cases

USB-C Charging

Catalyst 9400 1G UPOE© + 90W line card (C9400-LC-48H)



UPOE+

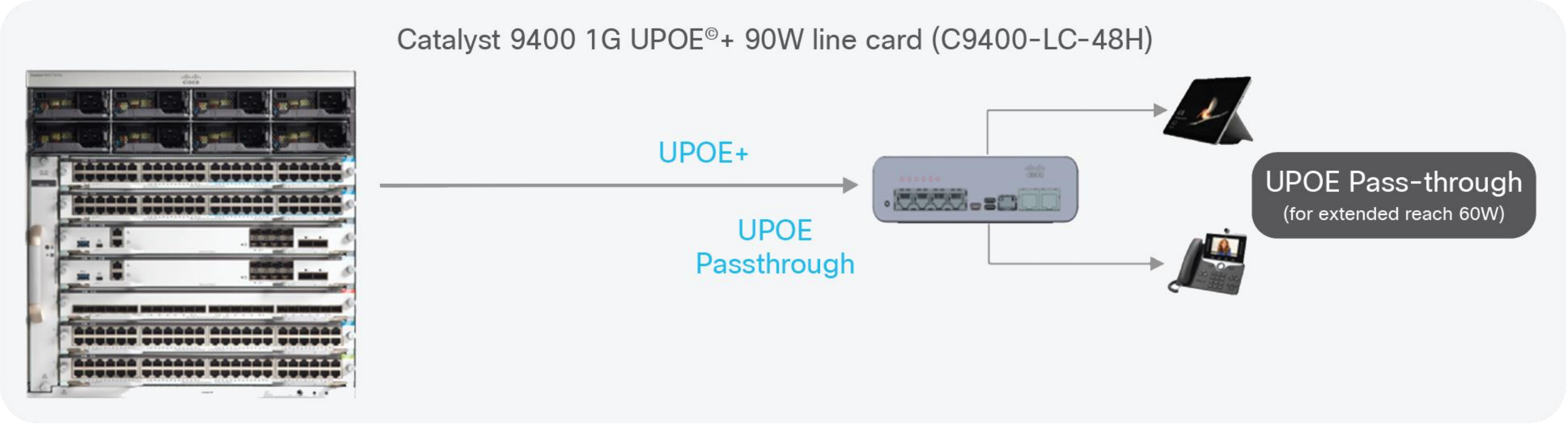
USB-C
Power + Data

USB-C power
(laptop charging + data)

IEEE802.3bt compliant platforms
Catalyst 9400 and 9300 Series*

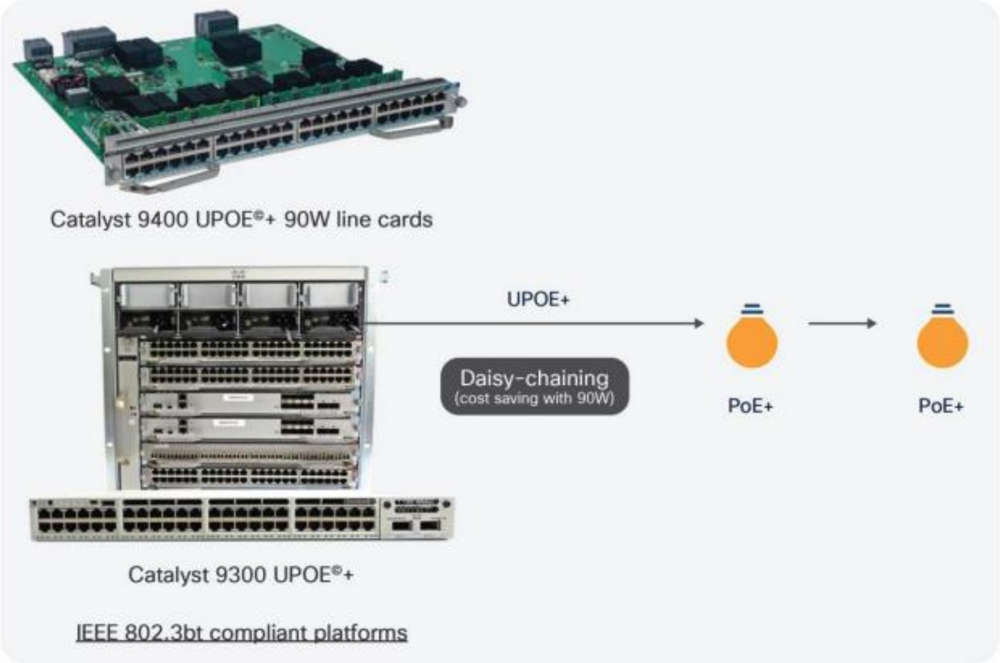
90W Use Cases

Pass-through PoE



90W Use Cases

Daisy-Chaining



Catalyst 9400 UPOE®+ 90W line cards

Catalyst 9300 UPOE®+

IEEE 802.3bt compliant platforms

PoE Features

Fast and Perpetual PoE (available starting in IOS XE 16.5.1)

- Perpetual PoE provides uninterrupted power to connected PDs while PSE reloads
- Fast PoE allows a switch to provide power before operating system loads
- Features are most often configured together- offers fast recovery after power failure and continuous power during reloads

PoE Features

Fast PoE

```
interface TenGigabitEthernet1/0/24
 description Building Lighting
 switchport access vlan 101
 switchport mode access
 power inline port perpetual-poe-ha
 power inline port poe-ha
end
```

PoE Features

Perpetual PoE

```
interface TenGigabitEthernet1/0/24
 description Building Lighting
 switchport access vlan 101
 switchport mode access
 power inline port perpetual-poe-ha
 power inline port poe-ha
end
```

PoE Features

PoE Port Priority

- PoE Power Management is available on the C9K family
  - By default, PoE interfaces are all given “low” priority
  - During power scarcity, PDs are powered down based on the system’s power-management algorithm
- Assigning priority to critical devices ensures these devices are prioritized

PoE Features

PoE Port Priority

```
interface TenGigabitEthernet1/0/22
 description CEO Phone Port
 switchport access vlan 101
 switchport mode access
 power inline port priority high
end
```

PoE Features

2-Event Classification

- Allows Class 4 PD to receive 30W without any CDP or LLDP negotiation
- Enables Class 4 PD to detect PSE capability to provide 30W
    - PD can move up to PoE+ without negotiation
- Otherwise, PD would be allocated 15.4W and rely on negotiation to upscale to PoE+

PoE Features

2-Event Classification

```
interface TenGigabitEthernet1/0/24
 description Phone Port
 switchport access vlan 101
 switchport mode access
 power inline port 2-event
end
```

Validation Commands

```
C9300#show post
Stored system POST messages:

Switch 1
---------

POST: MBIST Tests : Begin
POST: MBIST Tests : End, Status Passed
<snip>

POST: Inline Power Controller Tests : Begin
POST: Inline Power Controller Tests : End, Status Passed

POST: Thermal, Temperature Tests : Begin
POST: Thermal, Temperature Tests : End, Status Passed

POST: Thermal, Fan Tests : Begin
POST: Thermal, Fan Tests : End, Status Passed
```

Validation Commands

Classic validation CLI

```
C9300#show power inline | exclude off

Module    Available      Used      Remaining
           (Watts)     (Watts)      (Watts)
------    ---------   --------     ---------
1            525.0       46.2        478.8
Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------- ----- ----
Te1/0/41  auto   on         15.4    Ieee PD             4     60.0
Te1/0/42  auto   on         15.4    Ieee PD             4     60.0
Te1/0/43  auto   on         15.4    Ieee PD             4     60.0
--------- ------ ---------- ---------- ---------- ------ -----
Totals:          3     on   46.2
```

Validation Commands

UPOE-PLUS validation CLI

```
C9300#show power inline upoe-plus te1/0/24

Device IEEE Mode - BT

Codes: DS - Dual Signature device, SS - Single Signature device
       SP - Single Pairset device

Interface   Admin  Type Oper-State         Power(Watts)      Class   Device Name
            State       Alt-A,B        Allocated Utilized  Alt-A,B
----------- ------ ---- -------------- --------- --------- ------- -----------

Te1/0/24    auto   SS   on,off         7.0       3.7       2       IP Phone 8845
```

Summary and Conclusion

Packet Capture Tools

Usage Guidelines

Tool	Impact	Comments
Embedded Wireshark	🟡	Utilizes CPU and memory resources. Leverage capture filters/ACLs to reduce the possibility of inaccurate captures
Show Platform Forward (SPF)	🟡	Injects dummy packets from CPU to simulate forwarding decision, use PCAP for simple trigger creation
Packet State Vector (PSV)	🟢	Captures one packet a time, with no effect on switch functionality, triggers can be as generic or specific as needed
Fed Punject	🟡	Dedicated CPU capture tool focused on punted/injected packets, not advised during high CPU situations
SPAN	🟢	Provides the ability to mirror traffic locally, across L2 or L3 domain(s). Local SPAN may result in oversubscription, RSPAN may result in traffic flooding, ERSPAN requires packet de/encapsulation

Packet Capture Tool	Control Plane	Data Plane	PCAP	Header Info	Full Packet	Local Viewing	Remote Viewing	Filtering	Single Packet	Forwarding Decision	Platform (Only UADP ASIC)
Embedded Wireshark	●	●	●	●	●	●	●	●			All Cat9000* (C9200 supports EPC only)
Show Platform Forward (SPF)	●	●		●		●				●	All Cat9000* (C9500H and C9600 on later codes)
Packet State Vector (PSV)	●	●		●		●			●	●	C9500H and C9600 only
FED Punject	●			●		●		●			All Cat9000
SPAN/RSPAN/ERSPAN		●	●	●			●	●			All Cat9000

Overview of Troubleshooting Tools

Summary and Conclusion

Control Plane Traffic:

- Embedded Wireshark, FED Punject

Data Plane Traffic to internal buffer:

- Embedded Wireshark

Data Plane Traffic to external device:

- SPAN/RSPAN/ERSPAN

Forwarding Decision:

- Show Platform Forward (SPF), Packet State Vector (PSV)

Logging

Logging Tools Comparison

Tool	Impact	Comments
Event Trace	🟢	Per-process logging. Logs to 'notice' level by default. Survives reload and is human-readable.
Binary Trace	🟢	Per-process logging. Also set to 'notice' by default. Traces in volatile memory are readable and exportable to text file. Traces are archived in binary format to crashinfo directory. Archives are not human readable.
TLS Syslog	🟢	Secure implementation of classic syslogging. Encrypts syslog messages between switch and server.

PoE

PoE Key Points:

Cisco UPOE+ brings 90W PoE to the Catalyst product line

- USB-C charging
- Pass-through
- Daisy-chaining

UPOE+ is backwards-compatible across all IEEE standards

Catalyst 9000 supports high-available PoE features

- Fast PoE
- Perpetual PoE

Questions?

Fill out your session surveys!

Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!

Attendees will also earn 100 points in the **Cisco Live Challenge** for every survey completed.

These points help you get on the leaderboard and increase your chances of winning daily and grand prizes

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

The bridge to possible

Thank you

#CiscoLive

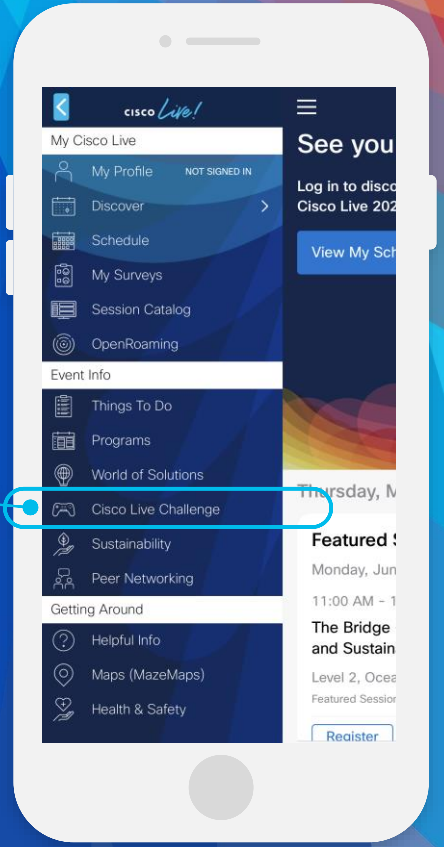
Cisco Live **Challenge**

Gamify your Cisco Live experience!

Get points for attending this session!

How:

1. Open the Cisco Events App.
2. Click on 'Cisco Live Challenge' in the side menu.
3. Click on View Your Badges at the top.
4. Click the + at the bottom of the screen and scan the QR code:



cisco Live!

Let's go

#CiscoLive