
cisco Live!
ALL IN


2
HOLLY SPRINGS FIRE-RESCUE
HOLLY SPRINGS
HOLLY SPRINGS
2
SAFE PLACE

The bridge to possible

Overview of Packet Capturing Tools in Cisco Switches and Routers Part 2

Nick Oliver, Technical Leader
@RTPCCIE

Oscar Silva, Technical Consulting Engineer
@CiscoPerson

BRKTRS-2811

Cisco Webex App

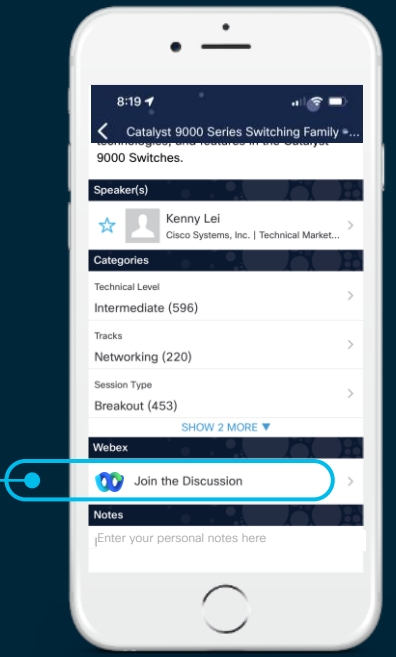
Questions?

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 17, 2022.

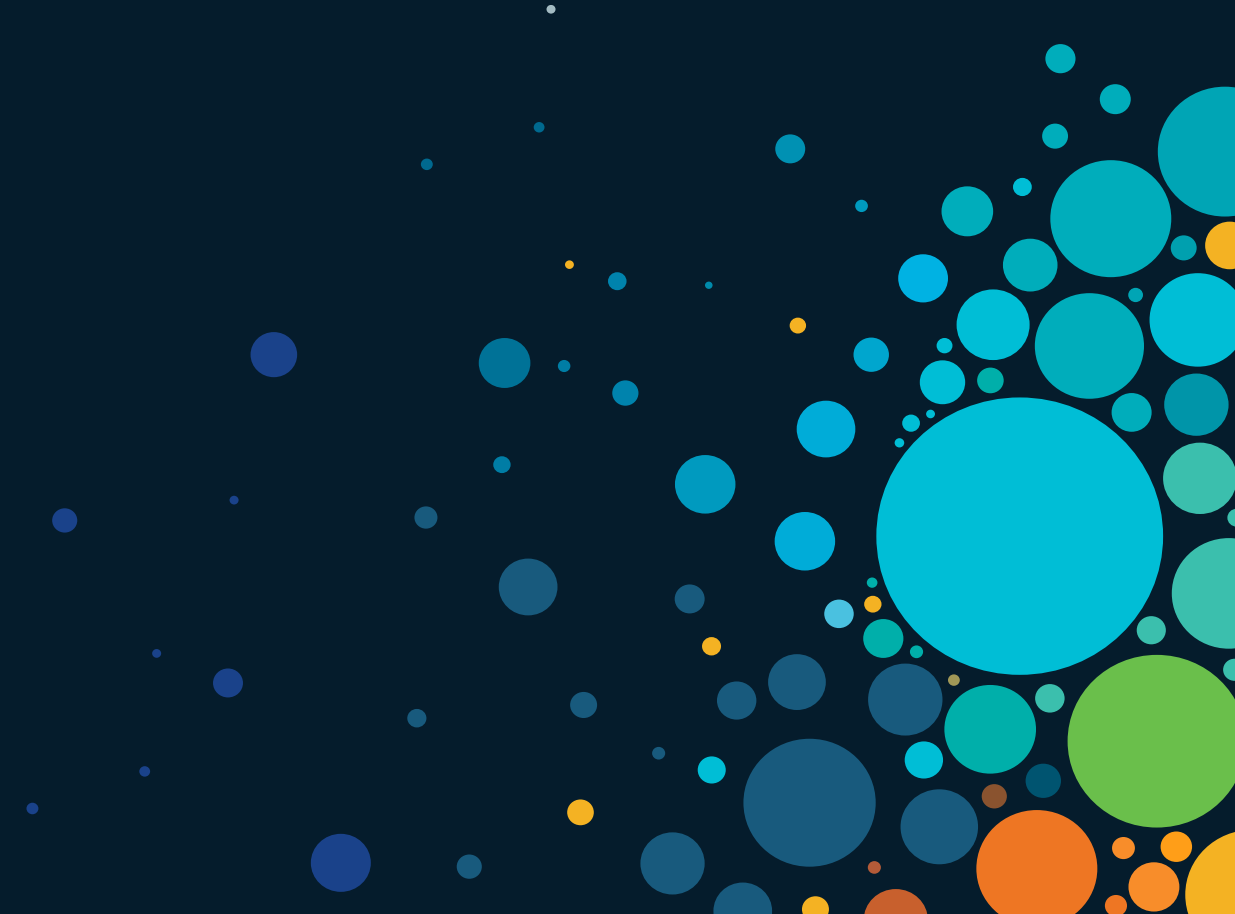


https://ciscolive.ciscoevents.com/ciscolivebot/#BRKTRS-2811

Agenda

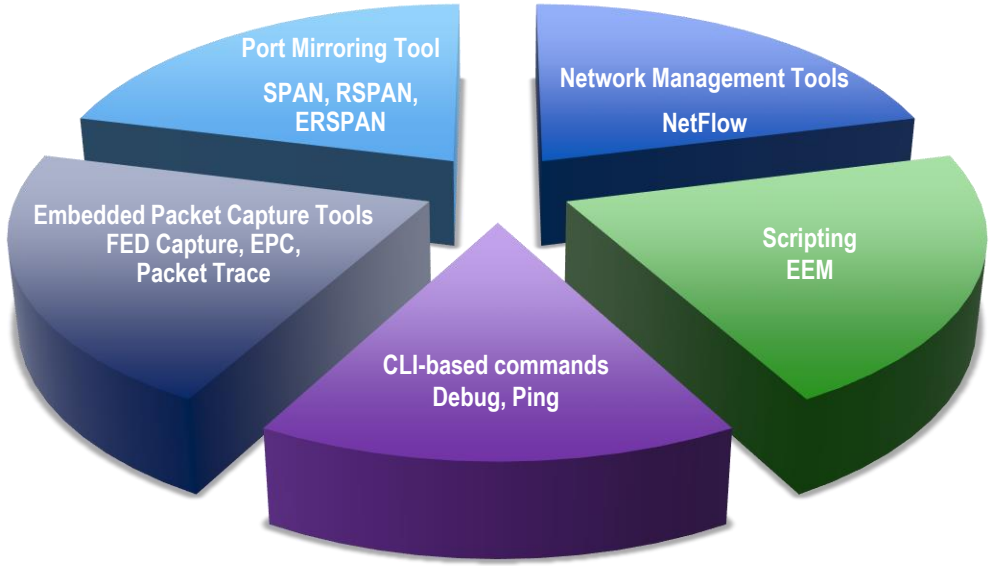
- Introduction
- Port Mirror Tools (RSPAN)
- FED Packet Capture
- Embedded Packet Capture
- Packet Trace
- Putting it All Together

Introduction



Goal of this Session…

- Create an **awareness of the packet capturing** tools that are available
- Learn how to use the tools through **real-world examples**
- Help you understand the **capabilities and features** of each tool

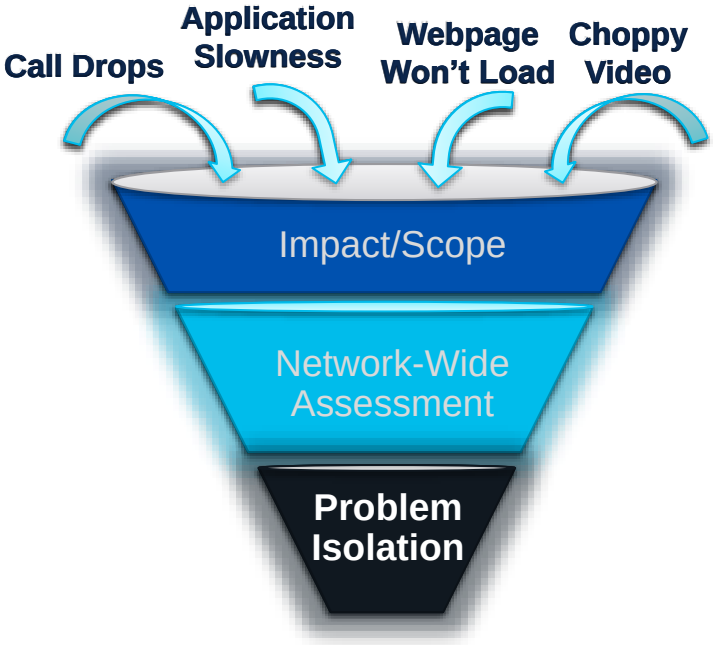


"It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts."

Sherlock Holmes (A Scandal in Bohemia)

A Troubleshooting Methodology

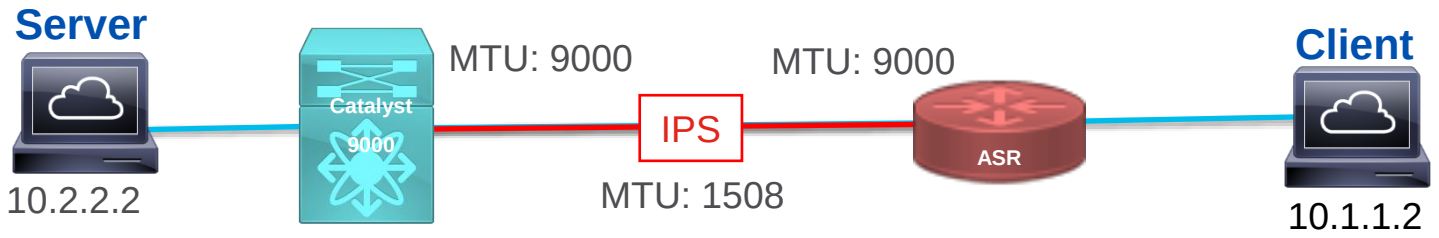
- Define the problem, impact, and scope based on facts, user reports, and considering recent changes
  - E.g., Application slowness after recent migration, When does the issue occur? Are all users impacted?
- Assess what you know from network monitoring
  - E.g., SNMP, Syslogs, NetFlow Data, Real-time performance monitoring, IP SLA. What is outside normal bounds?
- Select the right tool to isolate the problem
  - E.g., Are you on a router or switch? What type of data do you need (flow visibility, pcap, forwarding data, etc)



Having these details helps us make progress towards a resolution.

Data Transfers Are Broken

A Problem with MTU?



Sometimes a cable is more than just a cable.

Additional Breakout Sessions and Labs

For more info…

LABTRS-2456 Packet Capturing Tools in Routing Environments
LABTRS-2391 Packet Capturing Tools in Enterprise Switching Environments
LABTRS-2048 Packet Trace and Conditional Debugger on IOS-XE Routers

LABCRT-2452 CCNP ENCOR – Core Enterprise Network Technologies Practice Lab
LABCRT-2460 CCNP ENARSI – Implementing Cisco Enterprise Advanced Routing and Services Practice Lab
LABCRT-2464 Troubleshoot like a CCNP Practice Lab

BRKTRS-3475 Advanced Troubleshooting of the cat8k, asr1k, ISR and SDWAN Edge Made Easy
BRKXAR-2003 Extending Enterprise Network into Public Cloud with Cisco Catalyst 8000V Edge Software
BRKTRS-3090 Troubleshooting Cisco Catalyst 9000 Series Switches

Acronyms / Definitions

➔ Reference Slide

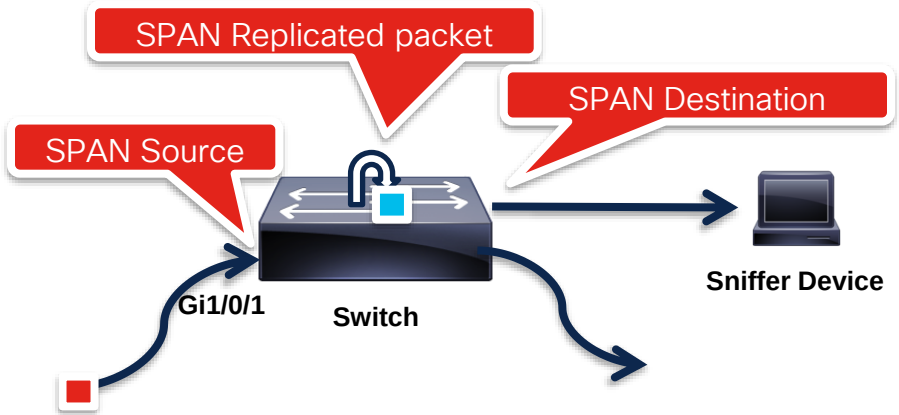
Acronyms	Definitions	Acronyms	Definitions
FNF	Flexible NetFlow	SP	Switch Processor
EPC	Embedded Packet Capture	RP	Route Processor
PSV	Packet State Vector	ASIC	Application Specific Integrated Circuit
SPF	Show Platform Forward	ELAM	Embedded Logic Analyzer Module
SPAN	Switch Port Analyzer	CoPP	Control Plane Policing
RSPAN	Remote SPAN	ACL	Access Control List
ERSPAN	Encapsulated RSPAN	FED	Forwarding Engine Driver
UADP	Unified Access Data Plane	RACL	Router-based ACL
		VACL	VLAN-based ACL

Port Mirroring Tools (RSPAN)

Switch Port Analyzer (SPAN)

Overview

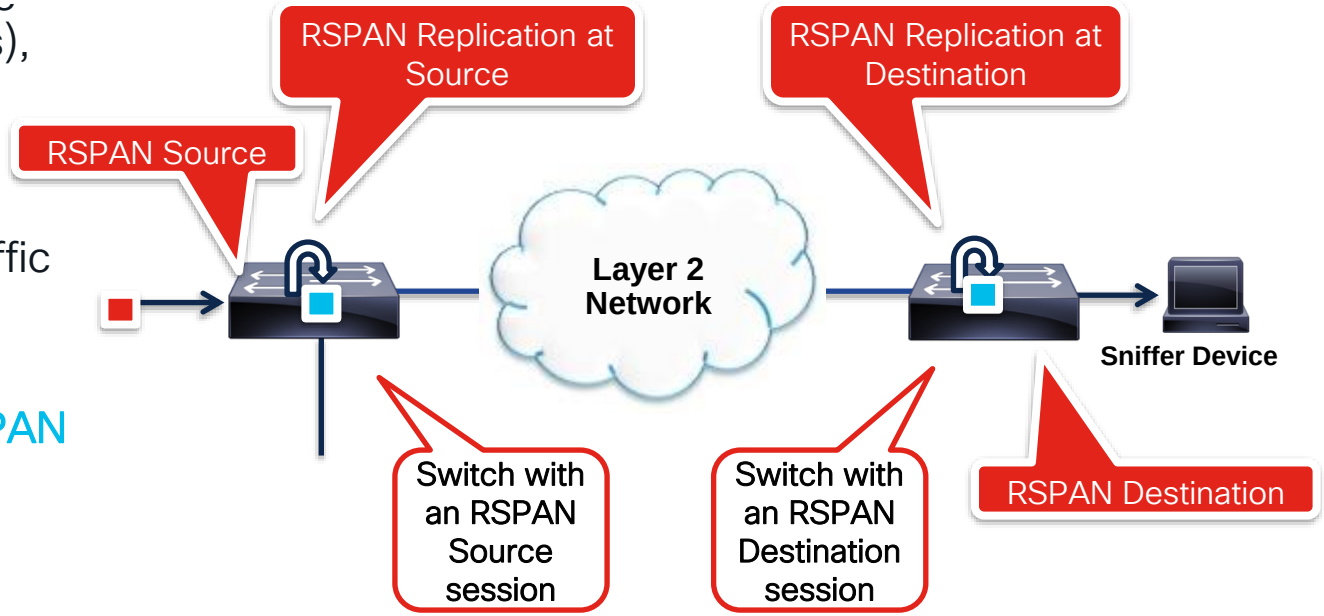
- A SPAN session (also known as port mirroring or monitoring) is an association of source ports/VLANs to one or more destination ports.
- Once the traffic is identified for replication, Cisco switch/router replicates the traffic to the destination port(s).



Remote Switch Port Analyzer (RSPAN)

Overview

- RSPAN supports source ports (or source VLANs), and destinations on different switches
- User-specified **Layer 2 VLAN** carries SPAN traffic between switches
- Consists of an RSPAN source session, an RSPAN VLAN, and an RSPAN destination session



SPAN/RSPAN

Configuration Example

Configuration syntax may differ depending on platform and version of code.

Local SPAN

```
monitor session 1 source <interface/vlan>
monitor session 1 destination interface <interface>
```

```
monitor session 1 type local
 source <interface/vlan>
 destination interface <interface>
 no shutdown
```

RSPAN Source

```
vlan 999
  remote-span
!
monitor session 1 source <interface/vlan>
monitor session 1 destination remote vlan 999
```

```
vlan 999
  remote-span
!
monitor session 1 type rspan-source
 source <interface/vlan>
 destination remote vlan 999
 no shutdown
```

RSPAN Destination

```
vlan 999
  remote-span
!
monitor session 1 source remote vlan 999
monitor session 1 destination interface <interface>
```

```
vlan 999
  remote-span
!
monitor session 1 type rspan-destination
 source remote vlan 999
 destination interface <interface>
 no shutdown
```

Advanced SPAN Filtering

Flow-Based SPAN, VACL and ACL Capture

Overview

- These features allow network administrators to replicate network traffic for monitoring purposes
- Different from traditional SPAN, these techniques provide the ability to selectively monitor traffic of interest via the use of an access-list
- Extremely useful in scenarios where a subset of traffic needs to be monitored on high bandwidth links and it is not practical or possible to capture all traffic

Flow-Based SPAN

Overview

- Apply an IPv4, IPv6, or MAC ACL to filter traffic on a SPAN session

```
ip access-list extended INTERESTING_TRAFFIC
 permit ip host 10.1.100.1 any
 permit ip any host 10.1.100.1
!
monitor session 1 source interface Gi4/0/1
monitor session 1 destination interface Gi4/0/2
monitor session 1 filter ip access-group INTERESTING_TRAFFIC

Switch# show monitor session 1
Session 1
---------
Type                   : Local Session
Source Ports           :
    Both               : Gi4/0/1
Destination Ports      : Gi4/0/2
    Encapsulation      : Native
          Ingress      : Disabled
IP Access-group        : INTERESTING_TRAFFIC
```

Flow-Based SPAN is supported on multiple switching platforms:

- 3560-X, 3750-X >= 12.2(44)SE
- 2960-S, 2960-X
- 3850, 3650
- Catalyst 4000
- Catalyst 9000

VACL Capture

Overview

- VLAN ACLs (VACLs) can provide access control for all packets that are bridged within a VLAN or that are routed into or out of a VLAN. VACL options include:
  - Drop
  - Forward [capture]
  - Redirect
- The **capture** action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets.

VACL Capture is only supported on Catalyst 6500/6800 & Cisco 7600 platforms

VACL Capture

Configuration

```
ip access-list extended INTERESTING_TRAFFIC
 permit ip host 10.1.100.1 any
 permit ip any host 10.1.100.1
ip access-list extended PERMIT_ALL
 permit ip any any
!
vlan access-map VACL_CAPTURE 10
 match ip address INTERESTING_TRAFFIC
 action forward capture
vlan access-map VACL_CAPTURE 20
 match ip address PERMIT_ALL
 action forward
!
vlan filter VACL_CAPTURE vlan-list  10
!
interface GigabitEthernet1/9
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport capture
```

Define an ACL that matches interesting traffic that should be sent to capture port

ACL to allow all remaining traffic

Use the '**forward capture**' keyword to forward the traffic and copy to capture port

Forward all remaining traffic

Apply VACL to VLAN

Configure '**switchport capture**' on capture interface that will receive the copied frames

FED Capture

FED Packet Capture
Catalyst 9000

- Forwarding Engine Driver (FED) is the heart of Cisco Unified Access switching platforms and is responsible for hardware programming and forwarding.
- Capture packets to/from the CPU
- **Non-Intrusive Debug** that can be used for troubleshooting high CPU
- Captures 256-16384 frames (default 4096, wrap with **buffer circular** option)

Direction

From CPU's Perspective

- Punt (Rx)
- Inject (Tx)

Wireshark Based Filters

- Display Filters:
  http://wiki.wireshark.org/DisplayFilters
  "eth.addr==00:00:0c:07:ac:01"
  "ip.src==10.1.1.1 && ip.dst==10.1.1.2"

- Capture Filters:
  http://wiki.wireshark.org/CaptureFilters
  "ether host 00:00:0c:07:ac:01"
  "src host 10.1.1.1 and dst host 10.1.1.2"

FED Packet Capture
Filter Types

Capture Filters:

```
CAT9300-1# debug platform software fed switch active [punt|inject] packet-capture  ?
  buffer         Configure packet capture buffer
  clear-filter   Clear punt PCAP filter
  set-filter     Specify wireshark like filter (Punt PCAP)
  start          Start punt packet capturing
  stop           Stop punt packet capturing
```

To clear the filter, use the 'clear-filter' option.

Capture filters are configured with the 'set-filter' option on the 'debug' command before the capture is started.

Display Filters:

```
CAT9300-1# show platform software fed switch active [punt|inject] packet-capture ?
  brief                Display captured packets
  detailed             Display in detailed format
  display-filter       Specify a wireshark like display filter
  display-filter-help  List all filters supported here
  status               Show packet capturing status
```

Display filters are issued with the 'display-filter' option on the 'show' command after capture.

Using 'display-filter-help' option will provide a long list of platform specific display filters.

FED Packet Capture
Filter Examples

Capture Filters:

Capture Only Spanning-Tree Packets:

```
CAT9300-1# debug platform software fed switch active [punt|inject] packet-capture set-filter "stp"
```

Capture Only Packets to/from 10.1.1.1 or from 192.168.1.1:

```
CAT9300-1# debug platform software fed switch active [punt|inject] packet-capture set-filter
"ip.addr == 10.1.1.1 or ip.src == 192.168.1.1"
```

Display Filters:

Display Only Packets with a Source or Destination of 3c51.0e7c.0182:

```
CAT9300-1# show platform software fed switch active [punt|inject] packet-capture display-filter
"eth.addr==3c51.0e7c.0182" brief
```

Display Only ARP Packets:

```
CAT9300-1# show platform software fed switch active [punt|inject] packet-capture display-filter
"arp" brief
```

Remember: FED Packet Capture is in the control-plane.

FED Packet Capture - Example

```
CAT9300-1# debug platform software fed switch active punt packet-capture start
Punt packet capturing started.

CAT9300-1# show platform software fed switch active punt  packet-capture status
Punt packet capturing: enabled. Buffer wrapping: disabled
Total captured so far: 15 packets. Capture capacity : 4096 packets

CAT9300-1# debug platform software fed switch active punt packet-capture stop
Punt packet capturing stopped. Captured 24 packet(s)

CAT9300-1# show platform software fed switch active punt packet-capture brief
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 24 packets. Capture capacity : 4096 packets

------ Punt Packet Number: 1, Timestamp: 2020/01/02 17:52:50.461 ------
 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000009], pal: GigabitEthernet1/0/1
[if-id: 0x00000009]
 metadata  : cause: 58 [Layer2 bridge domain data packet], sub-cause: 11, q-no: 4, linktype:
MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0100.5e00.000d, src mac: 001b.53bb.bbc5
 ether hdr : ethertype: 0x0800 (IPv4)
 ipv4  hdr : dest ip: 224.0.0.13, src ip: 10.122.164.3
 ipv4  hdr : packet len: 58, ttl: 1, protocol: 103
```

FED Packet Capture - Example

```
CAT9300-1# debug platform software fed switch active [punt|inject] packet-capture [start|stop]
Punt packet capturing started.

CAT9300-1# show platform software fed switch active punt packet-capture status
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets

CAT9300-1# show platform software fed switch active punt packet-capture brief
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets

------ Punt Packet Number: 1, Timestamp: 2020/01/02 18:22:51.757 ------
 interface : physical: GigabitEthernet1/0/2[if-id: 0x0000000a], pal: Vlan101 [i
0x00000042]
 metadata  : cause: 11 [For-us data], sub-cause: 0, q-no: 2, linktyp
 ether hdr : dest mac: 3c51.0e7c.01c1, src mac: 0016.c81c.2f81
 ether hdr : ethertype: 0x0800 (IPv4)
 ipv4  hdr : dest ip: 10.1.1.2, src ip: 10.1.1.1
 ipv4  hdr : packet len: 100, ttl: 255, protocol: 1 (ICMP)
 icmp  hdr : icmp type: 8, code: 0
```

Capture Complete!
If the capture is still running use the 'stop' command.

Ingress on port Gi1/0/2 on VLAN 101

Source and Destination MAC Address

Ethertype is 0x0800 for IPv4

Source and Destination IPv4 Address

FED Packet Capture - Example

```
CAT9300-1# show platform software fed switch active punt packet-capture detailed
<snip>
------ Punt Packet Number: 2, Timestamp: 2020/01/02 18:22:51.757 ------
 interface : physical: GigabitEthernet1/0/2[if-id: 0x0000000a], pal: Vlan101 [if-id:
0x00000042]
 metadata  : cause: 11 [For-us data], sub-cause: 0, q-no: 2, linktype:
MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 3c51.0e7c.01c1, src mac: 0016.c81c.2f81
 ether hdr : ethertype: 0x0800 (IPv4)
 ipv4  hdr : dest ip: 10.1.1.2, src ip: 10.1.1.1
 ipv4  hdr : packet len: 100, ttl: 255, protocol: 1 (ICMP)
 icmp  hdr : icmp type: 8, code: 0

 Packet Data Hex-Dump (length: 118 bytes) :
   3C510E7C01C10016  C81C2F8108004500  0064A2BF0000FF01  02D50A0101010A01
   0102[illegible]   17A8000000000FFD  96A8ABCDABCDABCD  ABCDABCDABCDABCD
   ABCD[illegible]   ABCDABCDABCDABCD  ABCDABCDABCDABCD  ABCDABCDABCDABCD
   ABC[illegible]    ABCDABCDABCDABCD  ABCD1FECA2[illegible]
```

Punt Cause

ICMP Packet

Target MAC

Source MAC

Ethertype 0x0800

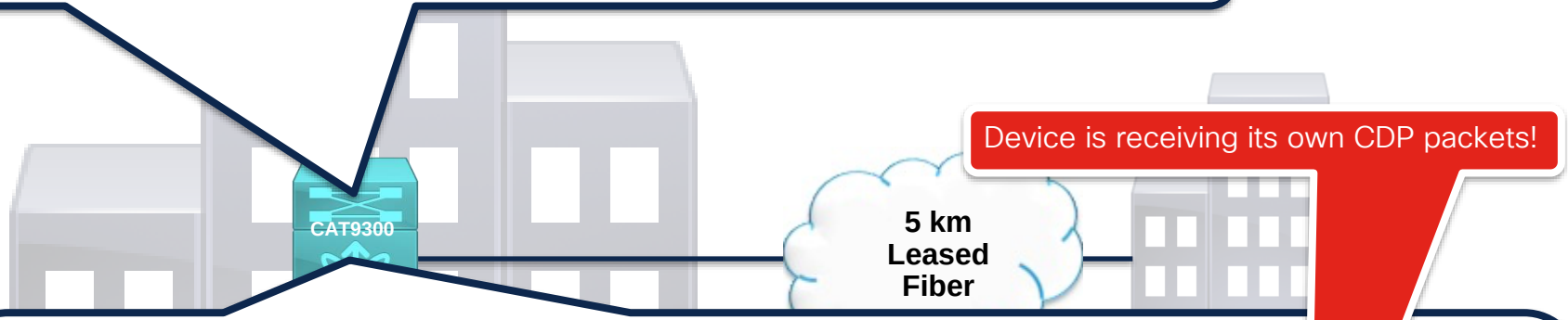
Source IP
10.1.1.1

Target IP
10.1.1.2

Real World Example

High CPU due to CDP Protocol

```
CAT9300-1# show proc cpu sort
CPU utilization for five seconds: 27%/7%; one minute: 28%; five minutes: 20%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 227     1196642     9627692        124 16.87% 17.20% 12.43%   0 CDP Protocol
```




```
CAT9300-1# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
ASR1000.cisco.com   Gig 1/0/3        158             R I   ASR1006   Gig 0
ISR4K-1.cisco.com   Gig 1/0/1        155           R S I   ISR4451-X Gig 0/0/3
ISR4K-2.cisco.com   Gig 1/0/4        136           R S I   ISR4351/K Gig 0/0/0
CAT9300-1.cisco.com Gig 1/0/2        179             S I   C9300L-48 Gig 1/0/2
Total cdp entries displayed : 4
```

Real World Example

High CPU due to CDP Protocol

```
CAT9300-1# debug platform software fed switch active punt packet-capture start
Punt packet capturing started.

CAT9300-1# show platform software fed switch active punt packet-capture status
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets

CAT9300-1# show platform software fed switch active punt packet-capture brief
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets
------ Punt Packet Number: 1, Timestamp: 2020/01/02 19:22:[illegible].001 ------
 interface : physical: GigabitEthernet1/0/2[if-id: 0x0000000a],
   0x0000000a]
 metadata  : cause: 96 [Layer2 control protocols], sub-cause: 0, [illegible]type:
   MCP_LINK_TYPE_LAYER2 [10]
 ether hdr : dest mac: 0100.0ccc.cccc, src mac: 3c51.0e7c.0182
 ether hdr : length: 449

CAT9300-1# debug plat soft fed switch act punt packet-capture set-filter "eth.addr == 0100.0ccc.cccc"
CAT9300-1# debug platform software fed switch active punt packet-capture start

CAT9300-1# sh plat soft fed switch active punt packet-capture brief | count dest mac: 0100.0ccc.cccc
  Number of lines which match regexp = 4096
```

After identifying the flow a short-term solution would be to disable CDP on G1/0/2 until the link issue is resolved:

```
CAT9300-1(config)# int G1/0/2
CAT9300-1(config-if)# no cdp enable
```

Leased Fiber link for the ingress interface

CDP Destination MAC for filter to narrow results

4096 of 4096 CDP Packets came in on G1/0/2

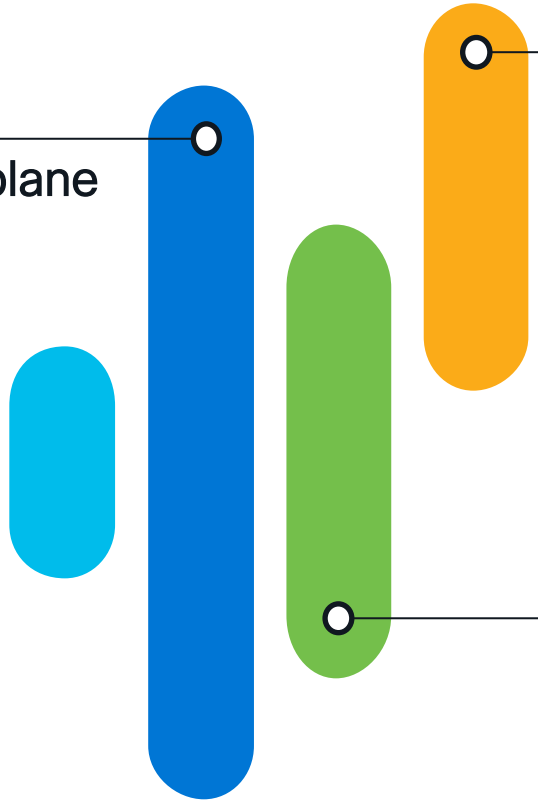
Embedded Packet Capture

Embedded Packet Capture Tools

What are they?

Built-in **control and data plane** capturing

Local buffer for storage



What do they provide?

Export to a **PCAP**

On-box analysis

ACL filtering (**L3/L4 parameters**)

Platforms

IOS Routers

IOS-XE Routers and Switches

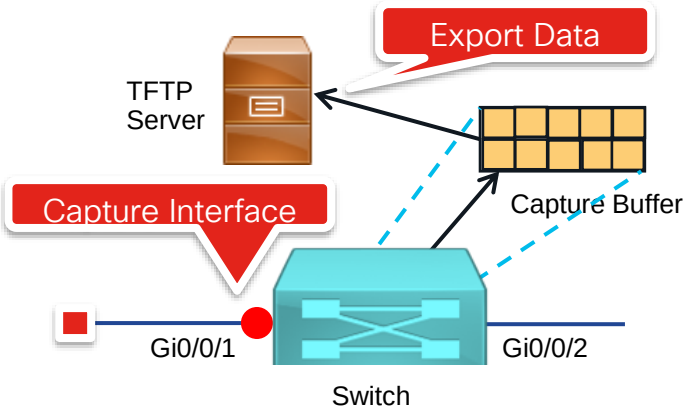
Embedded Packet Capture Tools

Key Advantages and Benefits

- Exec-level commands to start and stop the capture, define buffer size, buffer type (linear or circular) and packet size to capture
- Facility to export the packet capture in PCAP format suitable for analysis
- Useful when it is not possible to tap into the network using a stand-alone packet-sniffing tool, or when need arises to remotely debug and troubleshoot issues
- Capture rate can be throttled using further administrative controls. For example, using an Access Control List (ACL), specify maximum packet capture rate or specify a sampling interval
- Show commands to display packet contents on the device itself

Embedded Packet Capture (EPC)

Workflow



Define Buffer Criteria → Determine interface and filter → Export/display captured data

```
Router# monitor capture MYCAP interface Gig0/0/1 in
Router# monitor capture MYCAP access-list MYACL
Router# monitor capture MYCAP start
```

Embedded Packet Capture

Configuration Steps

Define Buffer Criteria

```
monitor capture MYCAP buffer [circular] [limit packets <1-10000>]
monitor capture MYCAP buffer size <1-10>
```

Determine Interface and Filter

```
monitor capture MYCAP [interface INTERFACE | control-plane] <in|out|both>
monitor capture MYCAP access-list MYACL
monitor capture MYCAP match [ any | mac <MAC> | <ipv4|ipv6> <any|host IP> <any|host IP> ]
monitor capture MYCAP start
```

Export/Display Captured Data

```
show monitor capture MYCAP parameter
show monitor capture MYCAP buffer [brief|detailed|dump]

monitor capture MYCAP export <bootflash:|ftp:|tftp:|sftp:|scp:>
```

Embedded Packet Capture (EPC)

Analyzing the Traffic on the Device

```
Switch# show monitor capture CAP parameter
   monitor capture CAP interface Gig0/0/2 both
   monitor capture CAP access-list MYACL
   monitor capture CAP buffer size 10
   monitor capture CAP limit pps 1000

Switch# show mon cap CAP buffer
 buffer size (KB) : 10240
 buffer used (KB) : 128
 packets in buf   : 5
 packets dropped  : 0
 packets per sec  : 1
```

Indicates total number of packets in the capture buffer

```
Switch# show monitor capture CAP buffer ?
  brief     brief display
  detailed  detailed display
  dump      for dump
  |         Output modifiers
  <cr>
Switch# show monitor capture CAP buffer brief
 ------------------------------------------------------------------------
 #   size   timestamp     source      destination      dscp    protocol
 ------------------------------------------------------------------------
   0   54   0.000000   20.20.24.16      ->  100.100.100.1   4        TCP
   1  114   1.042997   10.1.99.1        ->  224.0.0.5       48 CS6   OSPF
   2   58   2.428979   20.20.24.16      ->  100.100.100.1   4        TCP
   3   74   3.780010   00:00:00:00:00:00 -> 00:00:00:00:00:00   --    LLC
```

"brief" option provides basic information of the traffic like source/destination IP address, protocol type, packet length

Embedded Packet Capture (EPC)

Analyzing the Traffic on the Device

"detail" option provides result of both "brief" and "dump" options

```
Switch# show monitor capture CAP buffer detail
------------------------------------------------------------------------
#   size   timestamp     source       destination      dscp    protocol
------------------------------------------------------------------------
 0   54     0.000000    10.254.0.2       ->  100.100.100.1   4        ICMP
  0000:  0014A8FF A4020008 E3FFFC28 08004500   ...........(..E.
  0010:  00649314 0000FF01 551F0AFE 00026464   .d......U.....dd
  0020:  64010800 DF8F0017 00000000 000029E8   d.............).
  0030:  74C0AB1D ABCDABCD ABCDABCD ABCDABCD   ................
```

Destination MAC | Source MAC | Source IP | Destination IP

```
ASR# monitor capture CAP export bootflash:my_capture.pcap
Exported Successfully
```

Save capture in standard PCAP format

Embedded Packet Capture (Switches)

Configuration Steps and Traffic Analysis

```
C3850# monitor capture TESTCAP vlan 100 both
C3850# monitor capture TESTCAP file location bootflash:MY_CAP.pcap
C3850# monitor capture TESTCAP match ipv4 proto tcp eq 80
C3850# monitor capture TESTCAP start
C3850# monitor capture TESTCAP stop
```

Define a capture point. It could be an interface, VLAN or the control plane

Define a location to build the PCAP file or save to a buffer

Inline-filter matching with protocol type and L4 port number. Could also use an access-list

Start/stop the capture

Display options ….

```
C3850# show monitor capture file bootflash:MY_CAP.pcap
  brief           brief display
  detailed        detailed display
  display-filter  Display filter
  dump            for dump
  |               Output modifiers
```

Just like in EPC/MPA display options, Wireshark supports “brief”, “detail” and “dump” options, with similar results. Can further filter output using Wireshark display-filter syntax

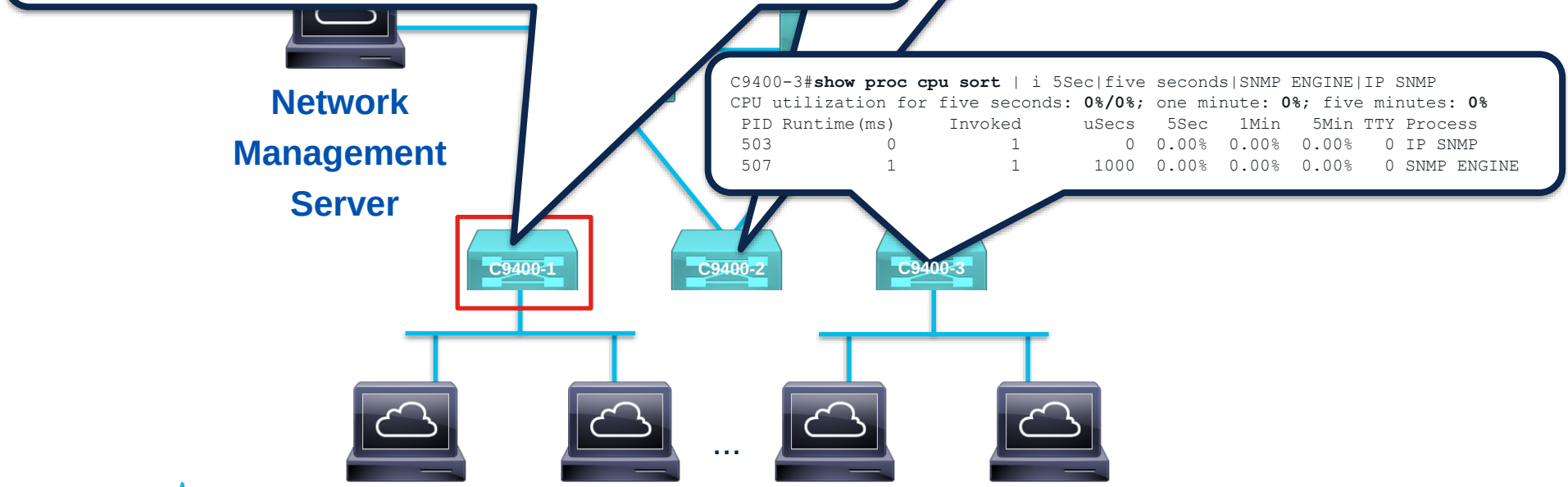
Real World Example

Access Switch with High CPU Utilization in SNMP Engine

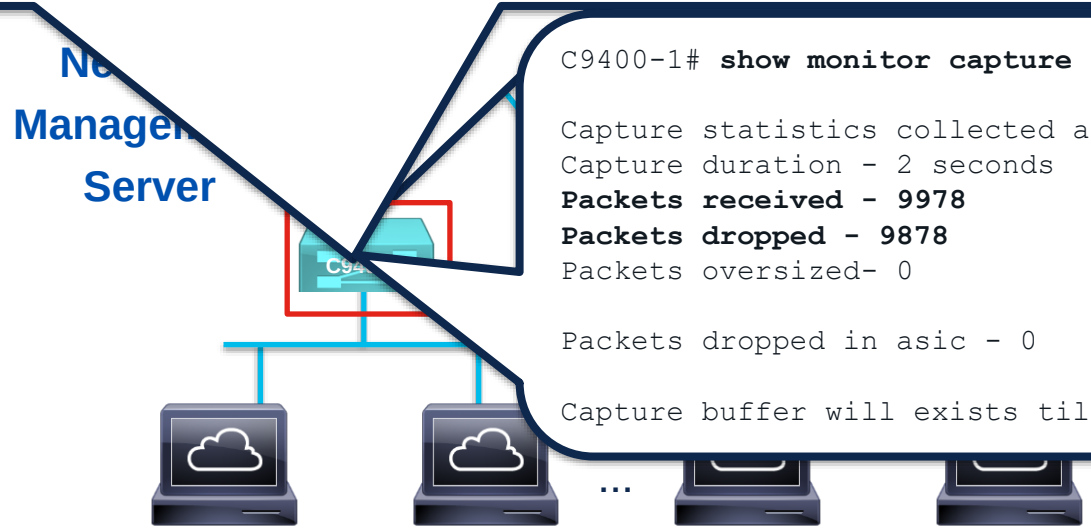
```
C9400-2#show proc cpu sort | i 5Sec|five seconds|SNMP ENGINE|IP SNMP
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 503           0           1          0  0.00%  0.00%  0.00%   0 IP SNMP
 507           1           1       1000  0.00%  0.00%  0.00%   0 SNMP ENGINE
```

```
C9400-1#show proc cpu sort | i 5Sec|five seconds|SNMP ENGINE|IP SNMP
CPU utilization for five seconds: 18%/6%; one minute: 17%; five minutes: 16%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 464      601422    17193042         34  5.19%  5.11%  4.71%   0 SNMP ENGINE
 432      386396    25797588         14  3.27%  3.04%  2.93%   0 IP SNMP
```

```
C9400-3#show proc cpu sort | i 5Sec|five seconds|SNMP ENGINE|IP SNMP
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 503           0           1          0  0.00%  0.00%  0.00%   0 IP SNMP
 507           1           1       1000  0.00%  0.00%  0.00%   0 SNMP ENGINE
```



```
C9400-1# monitor capture snmp_cap control-plane in limit packets 100
C9400-1# monitor capture snmp_cap match ipv4 protocol udp any any eq 161

C9400-1# monitor capture snmp_cap start

Enabling Control plane capture may seriously impact system performance. Do you want to continue?
[yes/no]: yes
Started capture point : snmp_cap

%BUFCAP-6-ENABLE: Capture Point snmp_cap enabled.

%BUFCAP-6-ENABLE: Capture Point snmp_cap disabled.
```




```
C9400-1# show monitor capture snmp_cap capture-statistics

Capture statistics collected at software:
Capture duration - 2 seconds
Packets received - 9978
Packets dropped - 9878
Packets oversized- 0

Packets dropped in asic - 0

Capture buffer will exists till exported or cleared
```

Real World Example

Access Switch with High CPU utilization in SNMP Engine

```
C9400-1# show monitor capture snmp_cap buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1   0.000000   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  2   0.000003   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  3   0.000005   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  4   0.000007   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  5   0.000009   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  6   0.000010   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  7   0.000012   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  8   0.000019   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
  9   0.000021   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0
 10   0.000023   172.16.1.25 -> 172.16.1.11   SNMP 90 get-next-request 1.3.6.1.4.1.9.2.1.1.0

C9400-1# show monitor capture snmp_cap buffer brief | count 1.3.6.1.4.1.9.2.1.1.0
Number of lines which match regexp = 97

C9400-1# monitor capture snmp_cap clear
```

Packet Trace
Catalyst 9000 Series

Capturing Tool Availability

RSPAN

FED Capture

Embedded Packet Capture (EPC)

Packet Trace

Show Platform Forward (SPF)

UADP 2.0/UADP 2.0 mini

Catalyst 9400

Catalyst 9300

Catalyst 9200

Catalyst 9500

Packet State Vector (PSV) + SPF

UADP 3.0

Catalyst 9600

Catalyst 9500H

Packet Trace

Catalyst 9000 Series
Show Platform Forward

Show Platform Forward

Catalyst 9000 Series UADP 2.0, 2.0 mini, 3.0

- **Available starting in 16.3.1**
- **User defined L2/L3/L4** packet data for a specific packet
- **Switch generates 150-200 packets** to determine ingress and egress forwarding decisions
  NOTE: generated **packets do not leave the switch**
- **Import** a specific packet **from a PCAP**
- Determine handling of the packet by the
  - Ingress & Egress decisions
  - Forwarding Interfaces
  - Rewrite Type

Show Platform Forward

Show Platform Forward Syntax in 16.3.1 onwards

- Define the ingress port, vlan, etc
- Provides the L2 Packet Data for the generated traffic
- Specify the L3 Protocol and Addressing for the traffic
- Configure L4 Protocol, Ports, and Optional Flags

```
CAT9300-1# show platform hardware fed switch <#> forward [interface|vlan|control-plane] <SMAC>
<DMAC> <L3 PROTOCOL> <L3 SRC> <L3 DEST> <L4 PROTOCOL> <L4 SRC> <L4 DEST> <OPT FLAGS>
```

```
CAT9300-1# show platform hardware fed switch active forward interface Te1/0/1 0cd0.f852.8042
c014.fe84.cc40 ipv4 192.168.1.37 8.8.8.8 udp 5193 53
```

Show Platform Forward

"Debug Platform Packet-Trace Feature Simulate" Syntax in 17.3.1 onwards

- Provides the L2 Packet Data for the generated traffic
- Specify the L3/L4 Protocol, Addressing, Ports, and Flags for the traffic
- Define the ingress port, vlan, etc

```
mac access-list extended DNS-PACKET-L2
 permit host 0cd0.f852.8042 host c014.f384.cc40

CAT9300-1# debug platform condition feature simulate mac DNS-PACKET-L2

ip access-list extended DNS-PACKET
 10 permit udp host 192.168.1.37 eq 5193 host 8.8.8.8 eq domain

CAT9300-1# debug platform condition feature simulate ipv4 DNS-PACKET

CAT9300-1# debug platform condition feature simulate interface Ten1/0/1

CAT9300-1# debug platform condition start
CAT9300-1# debug platform packet-trace simulation start
```

Show Platform Forward

"Debug Platform Packet-Trace Feature Simulate" Syntax in 17.3.1 onwards

- Define the ingress port, vlan, etc
- Provides the L2 Packet Data for the generated traffic
- Specify the L3 Protocol and Addressing for the traffic
- Configure L4 Protocol, Ports, and Optional Flags

```
CAT9300-1# show platform conditions

Conditional Debug Global State: Start
<SNIP>
Feature Condition        Type                        Value
-----------------------|--------------------------|-----------------------------------------
simulate                 ipv4 acl name               DNS-PACKET
simulate                 mac acl name                DNS-PACKET-L2
simulate                 interface name              TenGigabitEthernet1/0/1
```

Show Platform Forward

Verify the Results in 16.3.1 onward

After 2-5 minutes

%SHFWD-6-PACKET_TRACE_DONE: Switch 1 F0/0: fed: Packet Trace Complete: Execute (show platform hardware fed switch <> forward last summary|detail)
%SHFWD-6-PACKET_TRACE_FLOW_ID: Switch 1 F0/0: fed: Packet Trace Flow id is 65537

```
show platform hardware fed switch active forward last [summary|detail|flowid <#> (summary|detail)]
```

Input Packet

```
Input Packet Details:
###[ Ethernet ]###
  dst       = c0:14:fe:84:cc:40
  src       = 0c:d0:f8:52:80:42
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xa8f3
     src       = 192.168.1.37
     dst       = 8.8.8.8
     options   = ''
###[ UDP ]###
        sport     = 5193
        dport     = domain
        len       = 8
        chksum    = 0x1983
```

Ingress Details

```
Ingress:
   Port                : Ten1/0/1
   Global Port Number  : 1
   Local Port Number   : 1
   Asic Port Number    : 0
   Asic Instance       : 3
   Vlan                : 1
   Mapped Vlan ID      : 4
   STP Instance        : 2
   L3 Interface        : 37
    IPv4 Routing   : enabled
    IPv6 Routing   : enabled
    Vrf Id         : 0
   Adjacency:
    Station Index      : 180
    Destination Index  : 16402
    Rewrite Index      : 2
 Decision:
   Forwarding Mode    : 0 [Bridging]
   Replication Bit Map: ['localData']
   Winner : L2DESTMACVLAN LOOKUP
No exceptions occured.
CMI-2 Catch-all. Do not punt to CPU
```

Egress Details

```
Egress:
   Possible Replication :
       Port              : Ten1/0/2
       Port              : Ten1/0/3
   Output Port Data      :
       Port              : Ten1/0/2
    Global Port Number   : 1536
    Local Port Number    : 2
    Asic Port Number     : 1
    Asic Instance        : 3
    Unique RI            : 0
    Rewrite Type         : 1
                [L2_BRIDGE]
    Mapped Rewrite Type  : 4
                [L2_BRIDGE_INNER_IPv4]
    Vlan                 : 1
    Mapped Vlan ID       : 4
```

Output Packet

```
Output Packet Details:
    Port         : Ten1/0/2
###[ Ethernet ]###
  dst       = c0:14:fe:84:cc:40
  src       = 0c:d0:f8:52:80:42
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xa8f3
     src       = 192.168.1.37
     dst       = 8.8.8.8
     options   = ''
###[ UDP ]###
        sport     = 5193
        dport     = domain
        len       = 8
        chksum    = 0x1983
```

Show Platform Forward

Verify the Results in 17.3.1 onward

After 2-5 minutes

```
%SHFWD-6-PACKET_TRACE_DONE: Switch 1 F0/0: fed: Packet Trace Complete:
     Execute (show platform hardware fed switch <> forward last summary|detail)
%SHFWD-6-PACKET_TRACE_FLOW_ID: Switch 1 F0/0: fed: Packet Trace Flow id is 65537
```

show platform packet-trace simulation [summary|detail|status]

Input Packet

```
Input Packet Details:
###[ Ethernet ]###
  dst       = c0:14:fe:84:cc:40
  src       = 0c:d0:f8:52:80:42
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xa8f3
     src       = 192.168.1.37
     dst       = 8.8.8.8
     options   = ''
###[ UDP ]###
        sport     = 5193
        dport     = domain
        len       = 8
        chksum    = 0x1983
```

Ingress Details

```
Ingress:
   Port               : Ten1/0/1
   Global Port Number : 1
   Local Port Number  : 1
   Asic Port Number   : 0
   Asic Instance      : 3
   Vlan               : 1
   Mapped Vlan ID     : 4
   STP Instance       : 2
   L3 Interface       : 37
    IPv4 Routing   : enabled
    IPv6 Routing   : enabled
    Vrf Id         : 0
   Adjacency:
    Station Index      : 180
    Destination Index  : 16402
    Rewrite Index      : 2
 Decision:
   Forwarding Mode    : 0 [Bridging]
   Replication Bit Map: ['localData']
   Winner : L2DESTMACVLAN LOOKUP
 No exceptions occured.
 CMI-2 Catch-all. Do not punt to CPU
```

Egress Details

```
Egress:
 Possible Replication   :
       Port             : Ten1/0/1
       Port             : Ten1/0/2
       Port             : Ten1/0/3
       Port             : Ten1/0/4
 Output Port Data       :
       Port             : Ten1/0/2
    Global Port Number  : 1536
    Local Port Number   : 2
    Asic Port Number    : 1
    Asic Instance       : 3
    Unique RI           : 0
    Rewrite Type        : 1
                [L2_BRIDGE]
    Mapped Rewrite Type : 4
                [L2_BRIDGE_INNER_IPv4]
    Vlan                : 1
    Mapped Vlan ID      : 4
```

Output Packet

```
Output Packet Details:
    Port         : Ten1/0/2
###[ Ethernet ]###
  dst       = c0:14:fe:84:cc:40
  src       = 0c:d0:f8:52:80:42
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xa8f3
     src       = 192.168.1.37
     dst       = 8.8.8.8
     options   = ''
###[ UDP ]###
        sport     = 5193
        dport     = domain
        len       = 8
        chksum    = 0x1983
```

Show Platform Forward

Verify the Results in 17.3.1 onward

After 2-5 min

```
show platform packet-trace simulation status
========================
Switch 1:
========================
Available Flows in Switch: 1
65537     Progress
```

```
show platform packet-trace simulation summary
```

Input Packet

```
Input Packet Details:
###[ Ethernet ]###
  dst       = c0:14:fe:84:cc:40
  src       = 0c:d0:f8:52:80:42
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xa8f3
     src       = 192.168.1.37
     dst       = 8.8.8.8
     options   = ''
###[ UDP ]###
        sport     = 5193
        dport     = domain
        len       = 8
        chksum    = 0x1983
```

Ingress Details

```
Ingress:
   Port               : Ten1/0/1
   Global Port Number : 1
   Local Port Number  : 1
   Asic Port Number   : 0
   Asic Instance      : 3
   Vlan               : 1
   Mapped Vlan ID     : 4
   STP Instance       : 2
   L3 Interface       : 37
    IPv4 Routing   : enabled
    IPv6 Routing   : enabled
    Vrf Id         : 0
   Adjacency:
    Station Index      : 180
    Destination Index  : 16402
    Rewrite Index      : 2
 Decision:
   Forwarding Mode    : 0 [Bridging]
   Replication Bit Map: ['localData']
   Winner : L2DESTMACVLAN LOOKUP
No exceptions occured.
CMI-2 Catch-all. Do not punt to CPU
```

Egress Details

```
Egress:
 Possible Replication   :
       Port             : Ten1/0/1
       Port             : Ten1/0/2
       Port             : Ten1/0/3
       Port             : Ten1/0/4
Output Port Data      :
       Port             : Ten1/0/2
    Global Port Number  : 1536
    Local Port Number   : 2
    Asic Port Number    : 1
    Asic Instance       : 3
    Unique RI           : 0
    Rewrite Type        : 1
                [L2_BRIDGE]
    Mapped Rewrite Type : 4
                [L2_BRIDGE_INNER_IPv4]
    Vlan                : 1
    Mapped Vlan ID      : 4
```

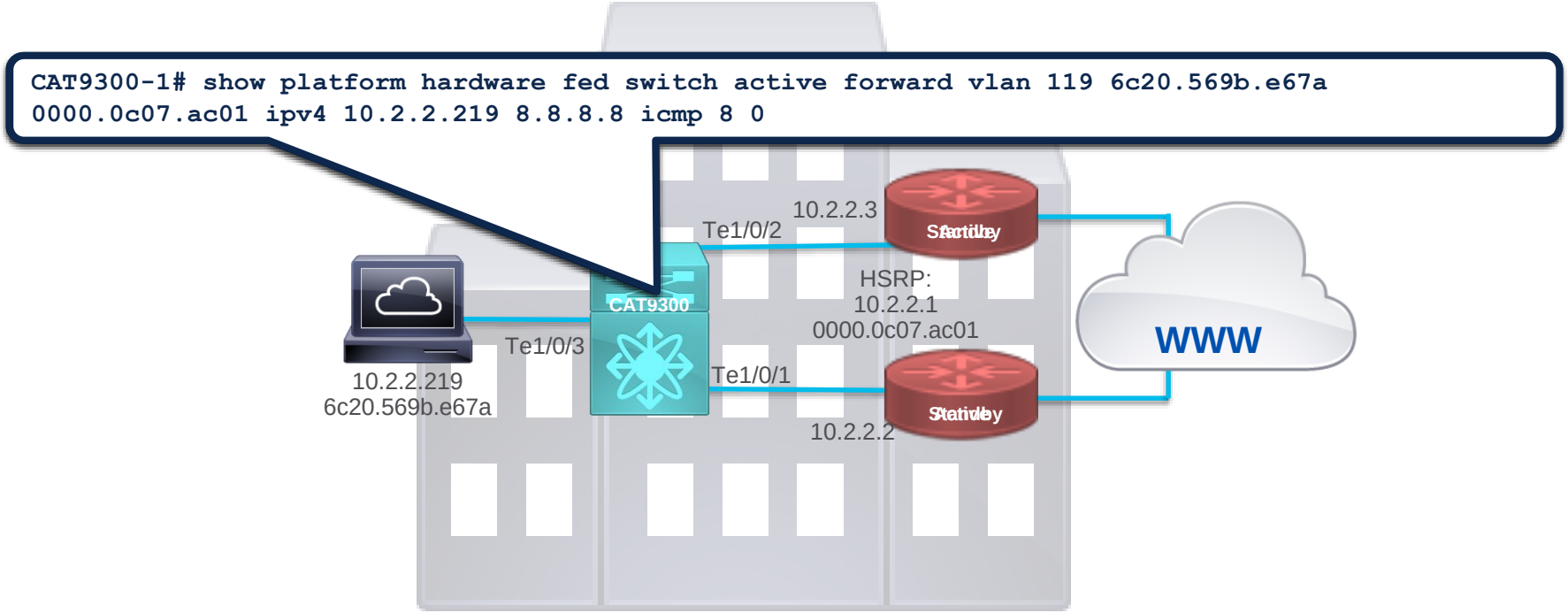
Output Packet

```
Output Packet Details:
    Port         : Ten1/0/2
###[ Ethernet ]###
  dst       = c0:14:fe:84:cc:40
  src       = 0c:d0:f8:52:80:42
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0xa8f3
     src       = 192.168.1.37
     dst       = 8.8.8.8
     options   = ''
###[ UDP ]###
        sport     = 5193
        dport     = domain
        len       = 8
        chksum    = 0x1983
```

Real World Example

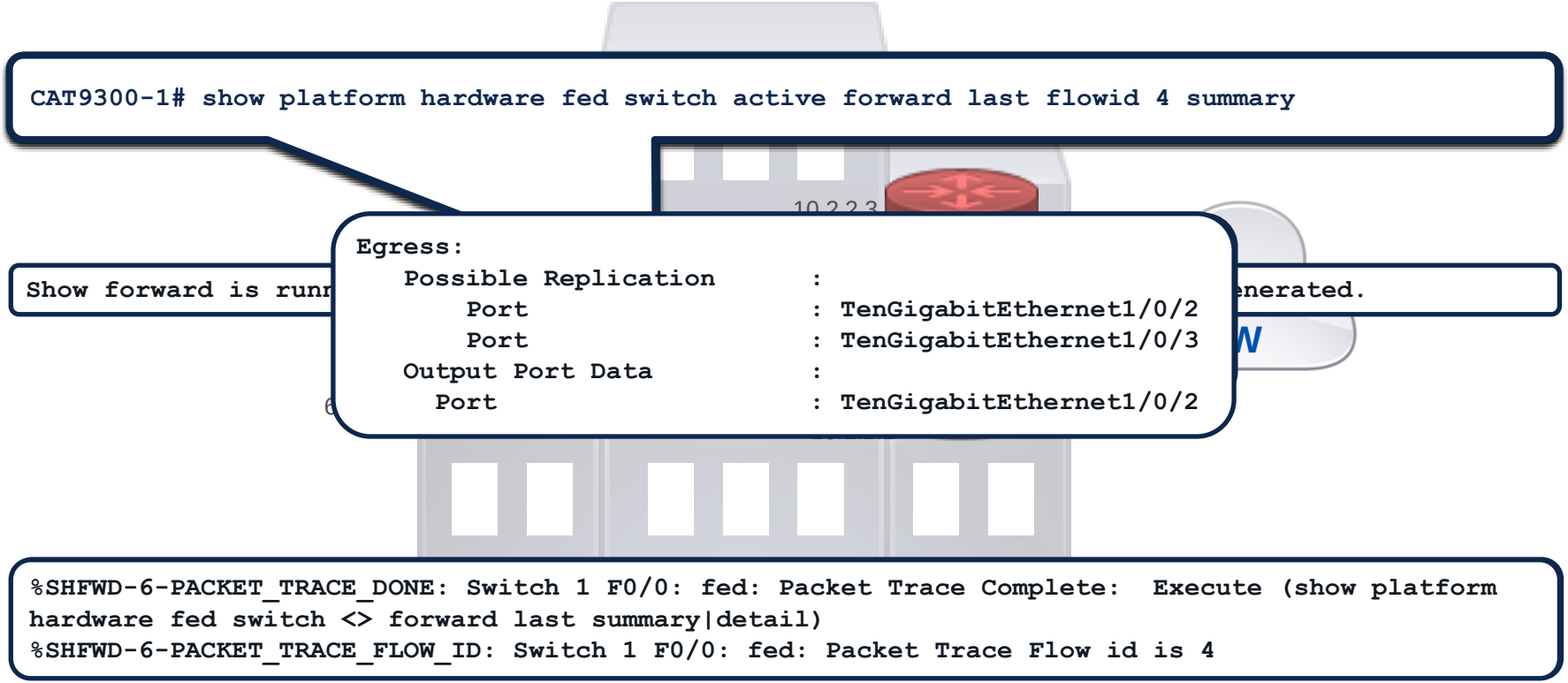
HSRP Forwarding Concerns – 16.3.1 Syntax

```
CAT9300-1# show platform hardware fed switch active forward vlan 119 6c20.569b.e67a
0000.0c07.ac01 ipv4 10.2.2.219 8.8.8.8 icmp 8 0
```



Real World Example

HSRP Forwarding Concerns – 16.3.1 Syntax



```
CAT9300-1# show platform hardware fed switch active forward last flowid 4 summary
```

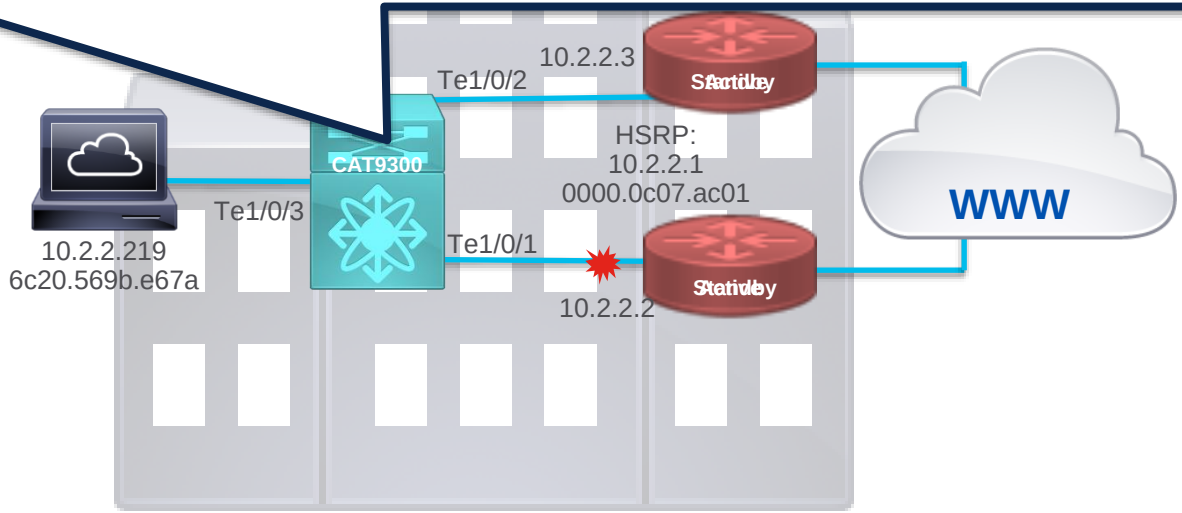
```
Egress:
    Possible Replication        :
        Port                    : TenGigabitEthernet1/0/2
        Port                    : TenGigabitEthernet1/0/3
    Output Port Data            :
      Port                      : TenGigabitEthernet1/0/2
```

```
%SHFWD-6-PACKET_TRACE_DONE: Switch 1 F0/0: fed: Packet Trace Complete:  Execute (show platform
hardware fed switch <> forward last summary|detail)
%SHFWD-6-PACKET_TRACE_FLOW_ID: Switch 1 F0/0: fed: Packet Trace Flow id is 4
```

Real World Example

HSRP Forwarding Concerns

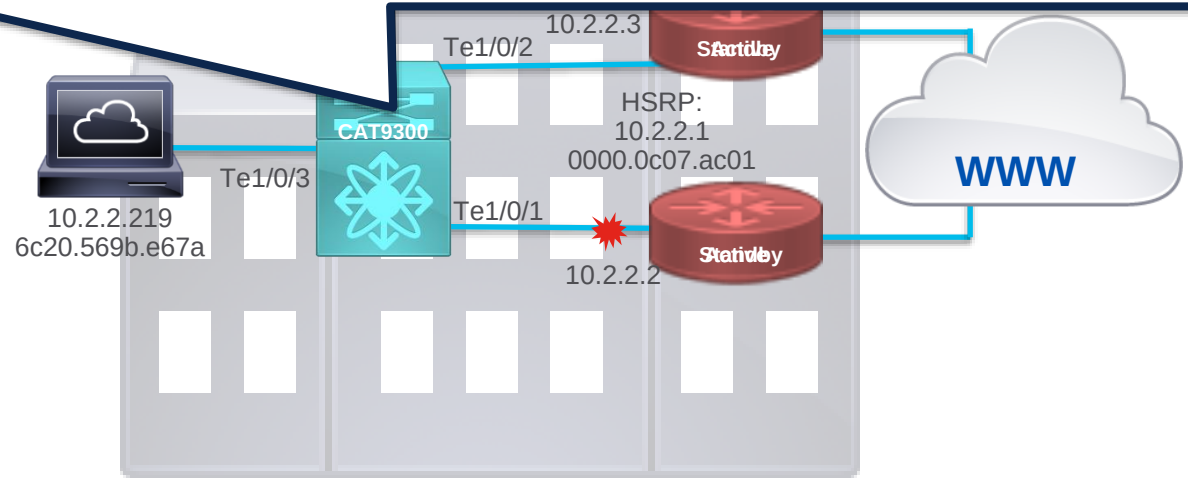
```
CAT9300-1# config t
CAT9300-1(config)# mac access-list extended HSRP-L2
CAT9300-1(config)# permit host 6c20.569b.e67a host 0000.0c07.ac01
CAT9300-1(config)# ip access-list extended HSRP-L3
CAT9300-1(config)# permit icmp host 10.2.2.219 host 8.8.8.8 echo
```



Real World Example

HSRP Forwarding Concerns

```
CAT9300-1# debug platform condition feature simulate interface T1/0/3
CAT9300-1# debug platform condition feature simulate mac HSRP-L2
CAT9300-1# debug platform condition feature simulate ipv4 HSRP-L3

CAT9300-1# debug platform condition start
CAT9300-1# debug platform packet-trace simulation start
```



Real World Example

HSRP Forwarding Concerns

```
CAT9300-1# debug platform condition feature simulate mac HSRP-L2
CAT9300-1# debug platform condition feature simulate ipv4 HSRP-L3
```

```
CAT9300-1# debug platform packet-trace simulation start
```

```
CAT9300-1# debug p
CAT9300-1# debug plat
```

```
CAT9300-1# show platform pac
<SNIP>
Egress:
   Possible Replication
       Port
Output Port Data          :
     Port                 :
```

```
CAT9300-1# show platform packet-trace simulation status
=========================
Switch 1:
=========================
Available Flows in Switch: 1
100728840    Complete
100728841    Complete
=========================
```

```
CAT9300-1# show platform packet-trace simulation status
=========================
Switch 1:
=========================
Available Flows in Switch: 1
100728840    Complete
=========================
```

Packet Trace

Catalyst 9500H & 9600
Packet State Vector

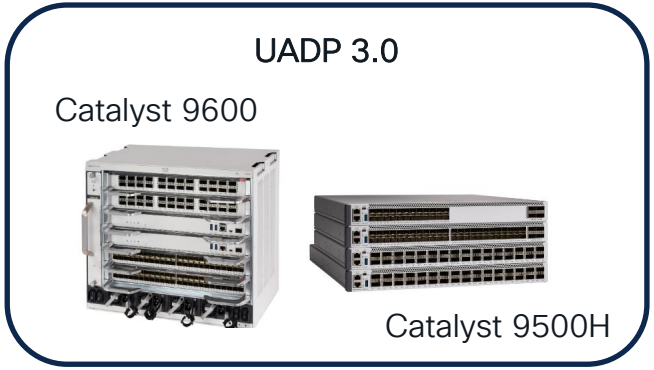
Packet Trace
Catalyst 9500H & 9600- UADP 3.0

- **Live capture** of a single packet
- **Visibility into the ASIC level** forwarding details for the captured packet
- UADP 3.0 feature that is only available on 9600 & 9500H
- “Packet State Vector” first introduced in 16.8.1 (**covered in LABTRS-2391**)
- New **“Packet Trace” syntax starting in 17.3.1**

Catalyst 9600

Catalyst 9500H

Packet Trace



Define Trigger & Start Capture → Review Configuration/Status & Clear → View Packet Forwarding Decision

```
show platform packet-trace summary                 <--- PSV Summary results
show platform packet-trace packet all              <--- Packet details
show platform packet-trace detailed ingress        <--- PSV FIR (Forwarding Ingress Resolution)
show platform packet-trace detailed egress         <--- PSV FER (Forwarding Egress Resolution)
```

Packet Trace

Packet State Vector Configuration Syntax in 16.8.1 onward

Define the Trigger & Start Capture (lots of options, use ? to explore)

```
debug platform hardware fed switch active capture trigger ipv4 192.168.29.132 192.168.32.2 icmp
debug platform hardware fed switch active capture trigger interface twentyFiveGigE2/5/0/5 ingress
debug platform hardware fed switch active capture start
```

Review Configuration/Status & Clear

```
show platform hardware fed switch active capture trigger
show platform hardware fed switch active capture status
clear platform hardware fed switch active capture trigger
```

View Packet Forwarding Decision

```
show platform hardware fed switch active capture summary
show platform hardware fed switch active capture packet
show platform hardware fed switch active capture detailed [ingressFc|egressFc]
```

Packet Trace

Configuration Syntax in 17.3.1 onward

Define the Trigger & Start Capture

```
debug platform condition interface <#> match ipv4 protocol udp host 192.168.1.37 host 8.8.8.8 eq 53 ingress
debug platform condition start
debug platform packet-trace start
```

Review Configuration/Status & Clear

```
show platform conditions
show platform packet-trace status
clear platform condition all
clear platform packet-trace configuration
```

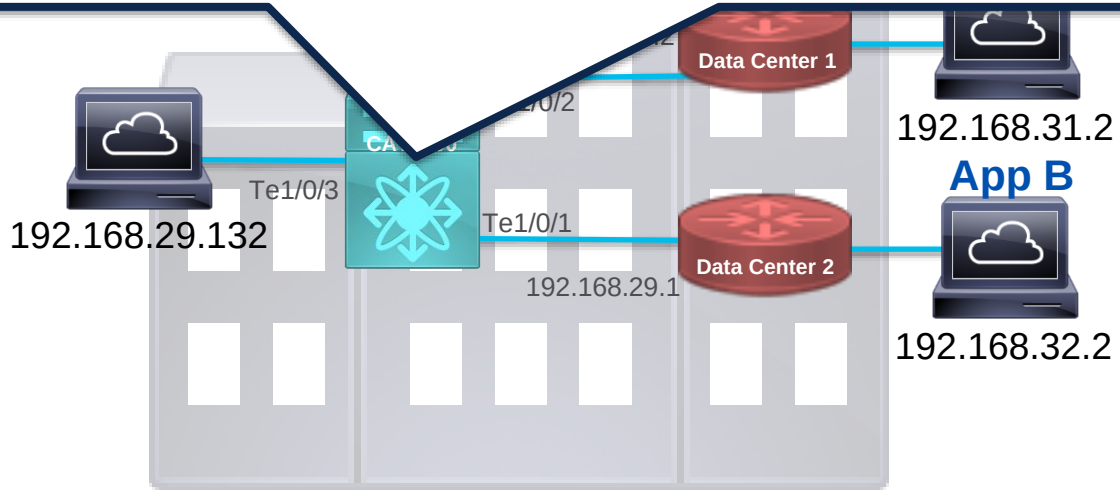
View Packet Forwarding Decision

```
show platform packet-trace summary              <--- PSV Summary results
show platform packet-trace packet all           <--- Packet details
show platform packet-trace detailed ingress     <--- PSV FIR (Forwarding Ingress Resolution)
show platform packet-trace detailed egress      <--- PSV FER (Forwarding Egress Resolution)
```

Packet State Vector Demo

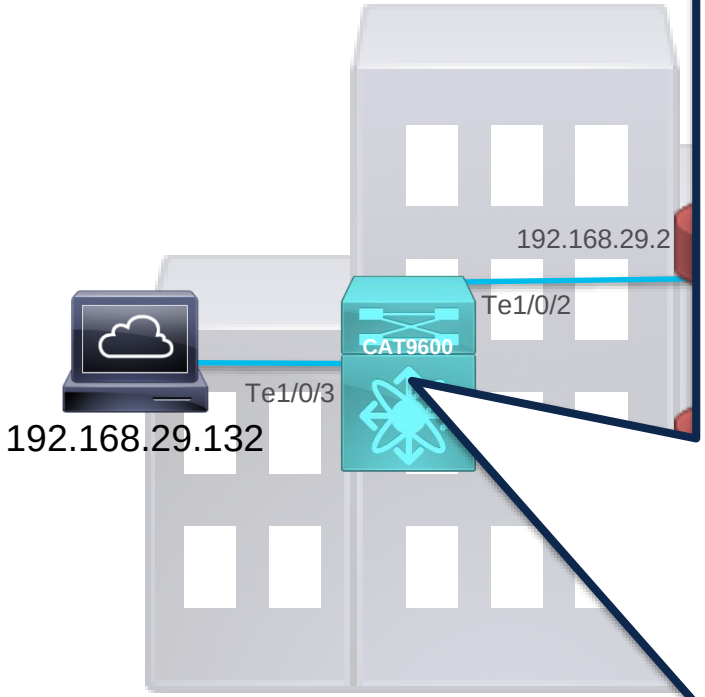
Application Traffic Forwarding Decision

```
CAT9600-1# debug platform condition interface Ten1/0/3 match ipv4 host 192.168.29.132 host 192.168.32.2 ingress
CAT9600-1# debug platform condition start
CAT9600-1# debug platform packet-trace start
```



Packet State Vector D

Application Traffic Forwarding D




```
CAT9600-1# show platform packet-trace packet all
<SNIP>
Trigger: Ingress Interface:TenGigabitEthernet1/0/3
      IP:192.168.32.2 Src IP:192.168.29.132
SUMMARY:
 Input      : Interface Te1/0/3
 Output     : Interface Te1/0/1
 Ingress Port: Te1/0/3
  GPN: 965 ASIC: 0 CORE: 0 SLICE: 1 ContextId: 4
 Ingress Packet:
  DMAC: cc:db:93:a2:18:70 SMAC: 6c:20:56:9b:e6:79 ETYPE:
       0x0800 LEN: 118
  SRC_IP: 192.168.29.132 DST_IP: 192.168.32.2 ToS: 0 PROTO:
       1 TTL: 255 Payload Length: 100
  ICMP TYPE: 0 CODE: 8 SEQ: 0
 Classification & Look up:
  Ingress:
  Replication Bitmap:
   LocalData : 1 LocalCpu  : 0 LocalDpu  : 0
   CoreData  : 0 CoreCpu   : 0 CoreDpu   : 0
   RemoteData: 0 RemoteCpu : 0 RemoteDpu : 0
 Egress Port: Te1/0/1
  GPN: 1344 ASIC: 0 CORE: 0 SLICE: 0 ContextId: 5
 Egress Packet:
  DMAC: cc:db:93:a2:18:70 SMAC: 6c:20:56:9b:e6:79 ETYPE:
       0x0800 LEN: 122
  SRC_IP: 192.168.29.132 DST_IP: 192.168.32.2 ToS: 0 PROTO:
       1 TTL: 255 Payload Length: 100
  ICMP TYPE: 0 CODE: 8 SEQ: 0
 Classification & Look up:
  Egress:
  Replication Bitmap:
   LocalData : 0 LocalCpu  : 0 LocalDpu  : 0
   CoreData  : 0 CoreCpu   : 0 CoreDpu   : 0
   RemoteData: 0 RemoteCpu : 0 RemoteDpu : 0
```

Putting it All Together

Real W

Degraded P

Client A

192.168.29.30/24

GW: 192.168.29.1

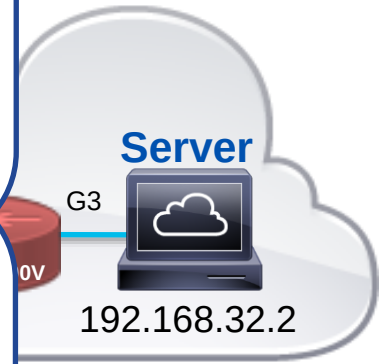
Client B

192.168.29.132

GW: 192.168.29.2

```
cisco@linux1:~$ iperf3 -b 200M -c 192.168.32.2
Connecting to host 192.168.32.2, port 5201
[  5] local 192.168.29.30 port 58538 connected to 192.168.32.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  23.9 MBytes   200 Mbits/sec   12    133 KBytes
[  5]   1.00-2.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   2.00-3.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   3.00-4.00   sec  23.8 MBytes   199 Mbits/sec    0    133 KBytes
[  5]   4.00-5.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   5.00-6.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   6.00-7.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   7.00-8.00   sec  23.8 MBytes   199 Mbits/sec    0    133 KBytes
[  5]   8.00-9.00   sec  23.8 MBytes   199 Mbits/sec    0    147 KBytes
[  5]   9.00-10.00  sec  23.9 MBytes   200 Mbits/sec    0    147 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   312 MBytes   200 Mbits/sec   12             sender
[  5]   0.00-10.00  sec   315 MBytes   200 Mbits/sec                  receiver
iperf Done.
```

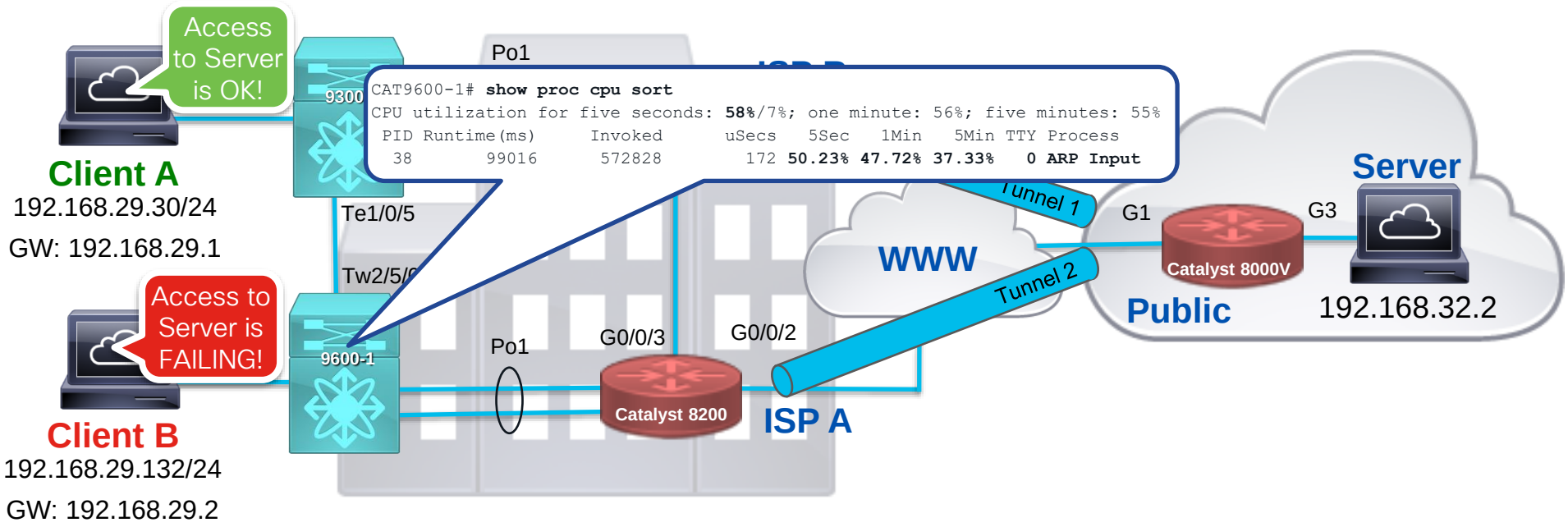
```
cisco@linux2:~$ iperf3 -b 200M -c 192.168.32.2
Connecting to host 192.168.32.2, port 5201
[  5] local 192.168.29.132 port 58554 connected to 192.168.32.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  96.2 KBytes   787 Kbits/sec   11   1.36 KBytes
[  5]   1.00-2.00   sec  0.00 Bytes    0.00 bits/sec    1   1.36 KBytes
[  5]   2.00-3.00   sec  0.00 Bytes    0.00 bits/sec    0   1.36 KBytes
[  5]   3.00-4.00   sec  0.00 Bytes    0.00 bits/sec   11   2.72 KBytes
[  5]   4.00-5.00   sec  31.5 KBytes   258 Kbits/sec    8   1.36 KBytes
[  5]   5.00-6.00   sec  31.3 KBytes   257 Kbits/sec   13   1.36 KBytes
[  5]   6.00-7.00   sec  31.3 KBytes   256 Kbits/sec   13   1.36 KBytes
[  5]   7.00-8.00   sec  0.00 Bytes    0.00 bits/sec    9   1.36 KBytes
[  5]   8.00-9.00   sec  0.00 Bytes    0.00 bits/sec    2   1.36 KBytes
[  5]   9.00-10.00  sec  0.00 Bytes    0.00 bits/sec    1   1.36 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   190 KBytes   156 Kbits/sec   69             sender
[  5]   0.00-10.21  sec   137 KBytes   110 Kbits/sec                  receiver
iperf Done.
cisco@linux2:~$
```



cisco Live!

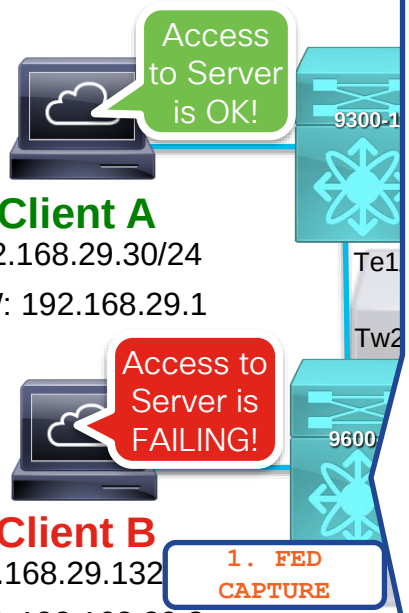
Real World Example

Degraded Performance



Real World

Degraded Perform



Client A
192.168.29.30/24
GW: 192.168.29.1

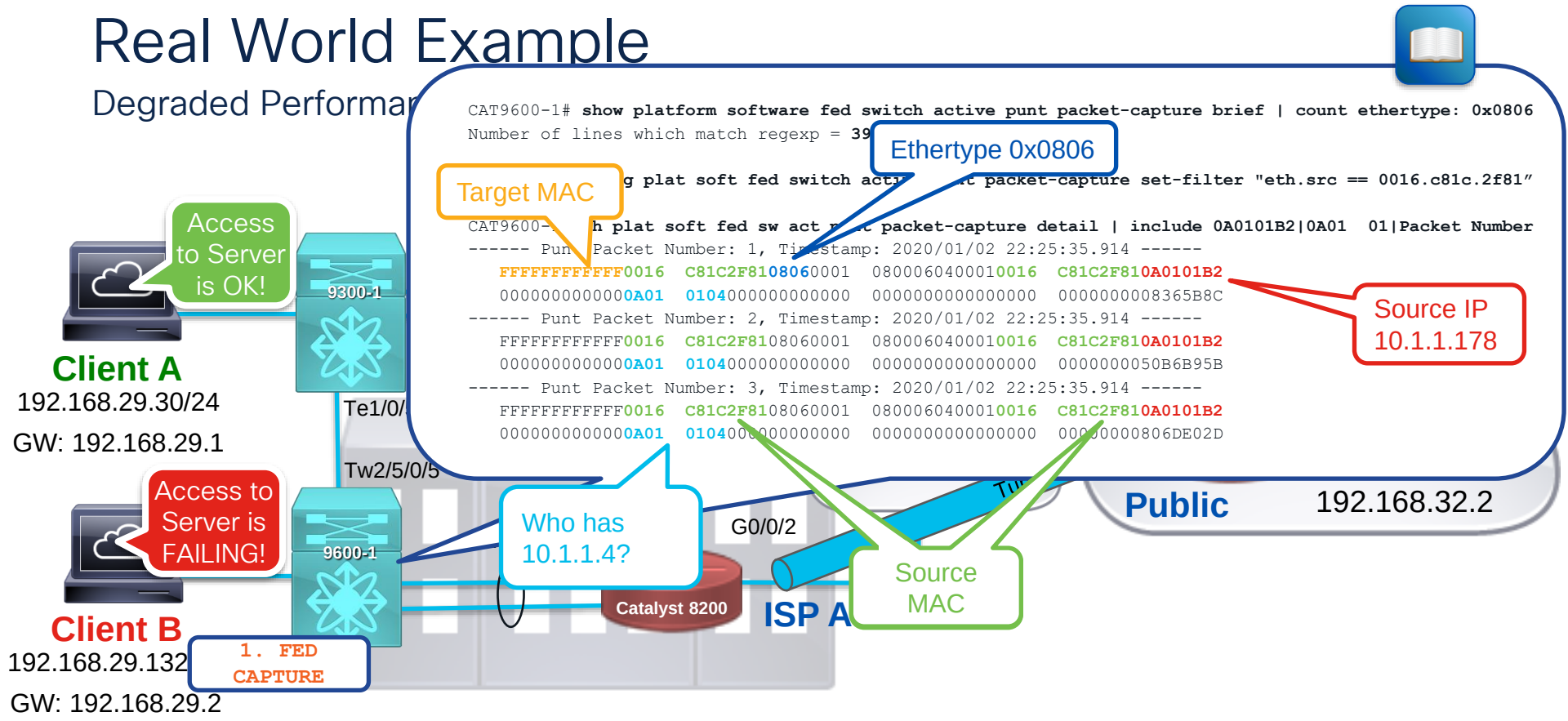
Client B
192.168.29.132
GW: 192.168.29.2

```
CAT9600-1# debug platform software fed switch active punt packet-capture start
Punt packet capturing started.

CAT9600-1# show platform software fed switch active punt packet-capture status
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets

CAT9600-1# show platform software fed switch active punt packet-capture brief
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets

------ Punt Packet Number: 1, Timestamp: 2020/01/02 22:02:32.660 ------
 interface : physical: GigabitEthernet1/0/2[if-id: 0x0000000a], pal: Vlan500 [if-id: 0x00000042]
 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: 0016.c81c.2f81
 ether hdr : ethertype: 0x0806 (ARP)

CAT9600-1# show platform software fed switch active punt packet-capture cpu-top-talker summary
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 4096 packets. Capture capacity : 4096 packets

L2 Top Talkers:
3937    Source mac      00:16:c8:1c:2f:81
3946    Dest mac        ff:ff:ff:ff:ff:ff
3937    Vlan    500

L3 Top Talkers:
3937    Source IPv4     10.1.1.178
15      Dest IPv4       10.1.1.4
131     TTL     255

L4 Top Talkers:
104     Protocol Num    (TCP)
104     L4 Source Port  5427
104     L4 Dest Port    23

Internal Top Talkers:
3937    Interface       Vlan500
3946    CPU Queue       ARP request or response
```

NOTE:
packet-capture cpu-top-talker summary is available in 17.6.1 onwards

Real World Example

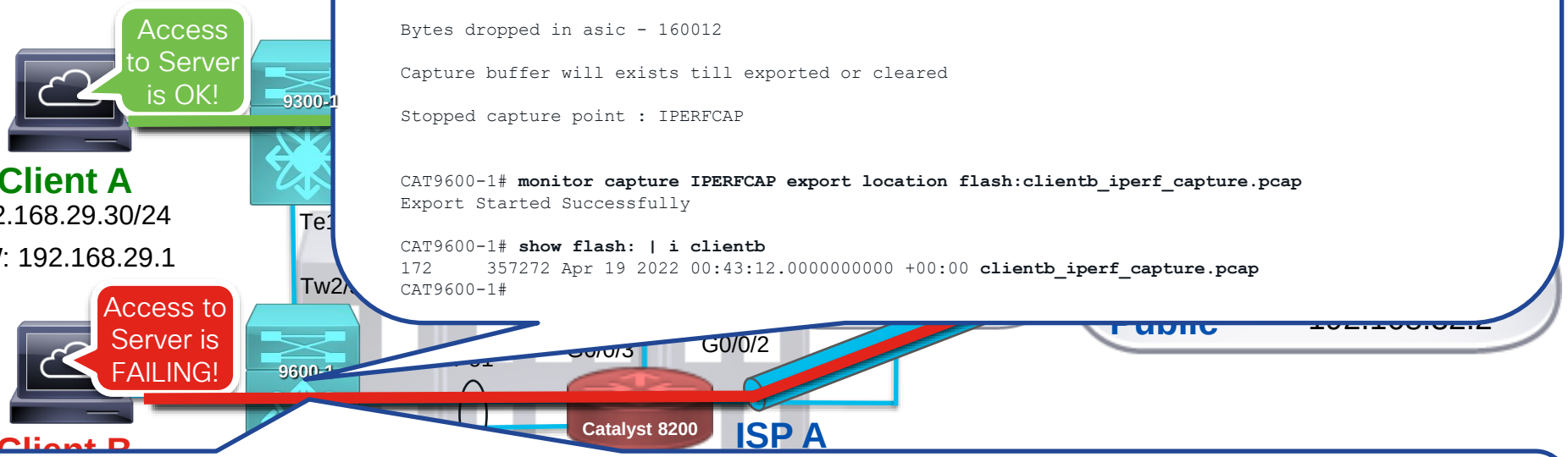
Degraded Performar



Real World

Degraded Perform

```
CAT9600-1# monitor capture IPERFCAP vlan 119 both match ipv4 host 192.168.29.132 host 192.168.32.2
CAT9600-1# monitor capture IPERFCAP start

CAT9600-1# monitor capture IPERFCAP stop
Capture statistics collected at software:
        Capture duration - 22 seconds
        Packets received - 268
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 160012

Capture buffer will exists till exported or cleared

Stopped capture point : IPERFCAP


CAT9600-1# monitor capture IPERFCAP export location flash:clientb_iperf_capture.pcap
Export Started Successfully

CAT9600-1# show flash: | i clientb
172     357272 Apr 19 2022 00:43:12.0000000000 +00:00 clientb_iperf_capture.pcap
CAT9600-1#
```




```
CAT9600-1# show monitor capture file flash:clientb_iperf_capture.pcap brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 192.168.29.132 -> 192.168.32.2 TCP 78 39488 -> 5201 [SYN] Seq=0 Win=64676 Len=0 MSS=1406 SACK_PERM=1 TSval=1112716044 TSecr=0 WS=128
    2   0.001182 192.168.29.132 -> 192.168.32.2 TCP 70 39488 -> 5201 [ACK] Seq=1 Ack=1 Win=64768 Len=0 TSval=1112716045 TSecr=2858875194
    3   0.001217 192.168.29.132 -> 192.168.32.2 TCP 107 39488 -> 5201 [PSH, ACK] Seq=1 Ack=1 Win=64768 Len=37 TSval=1112716045 TSecr=2858875194
    4   0.002761 192.168.29.132 -> 192.168.32.2 TCP 70 39488 -> 5201 [ACK] Seq=38 Ack=2 Win=64768 Len=0 TSval=1112716047 TSecr=2858875196
```

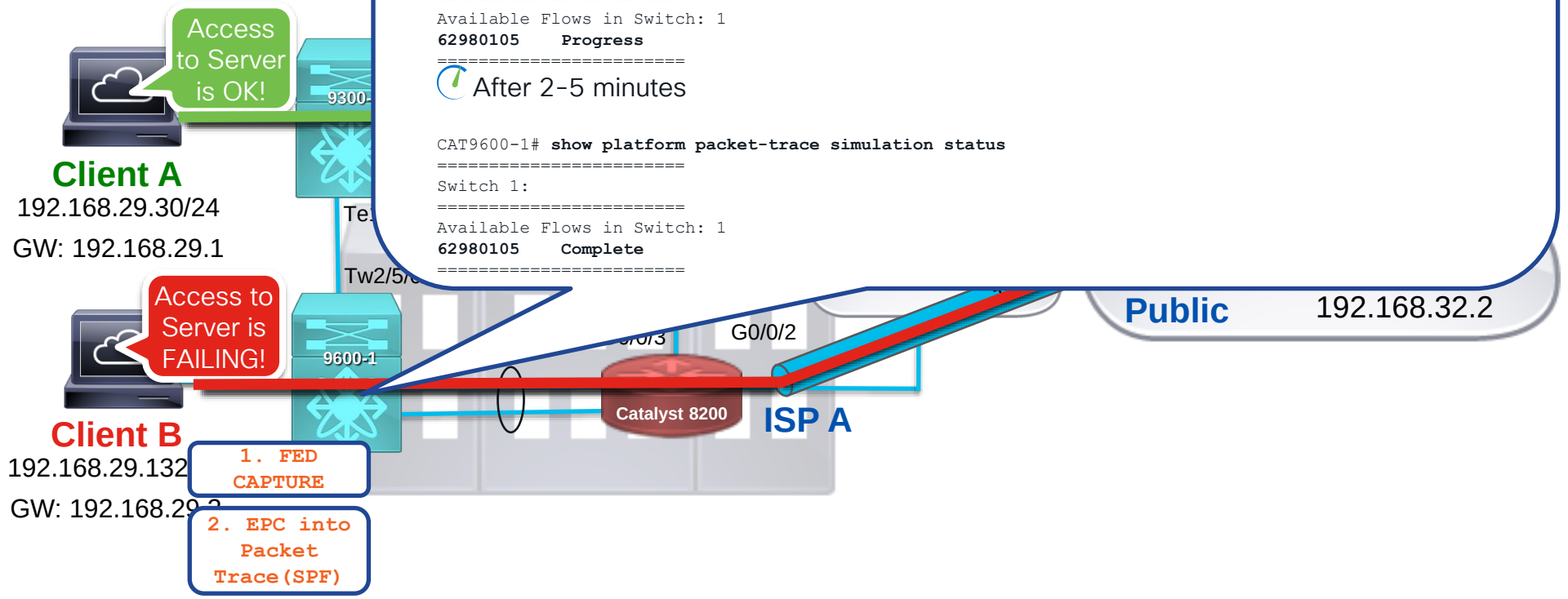
Real World

Degraded Perform

```
CAT9600-1# debug platform condition feature simulate pcap flash:clientb_iperf_capture.pcap number 1
CAT9600-1# debug platform condition feature simulate interface tw2/5/0/1

CAT9600-1# debug platform condition start
CAT9600-1# debug platform packet-trace simulation start

CAT9600-1# show platform packet-trace simulation status
=========================
Switch 1:
=========================
Available Flows in Switch: 1
62980105     Progress
=========================
```

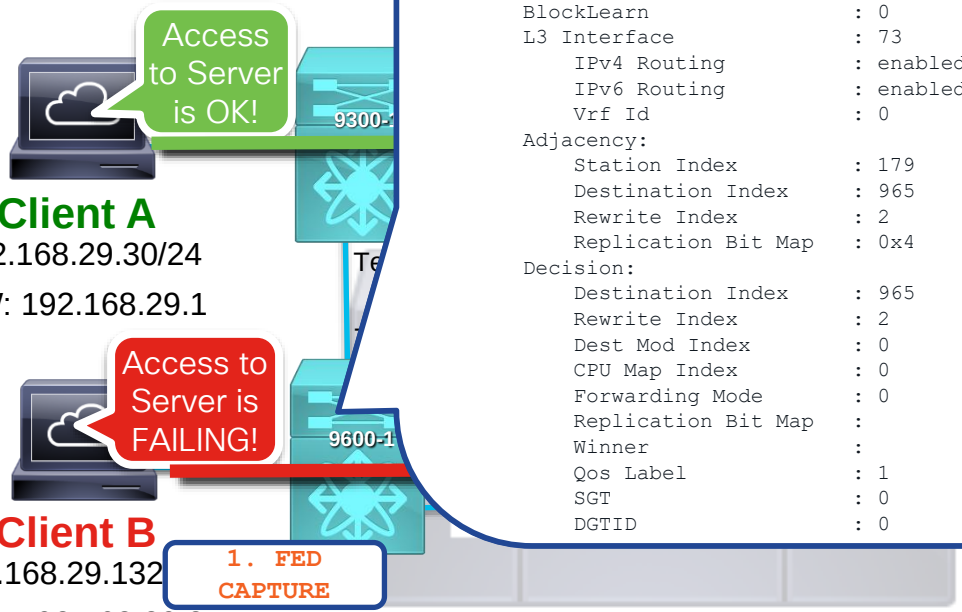
After 2-5 minutes

```
CAT9600-1# show platform packet-trace simulation status
=========================
Switch 1:
=========================
Available Flows in Switch: 1
62980105     Complete
=========================
```



Real World

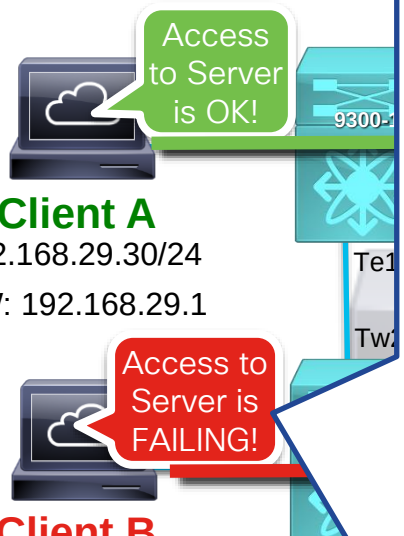
Degraded Perform




```
Ingress:
    Port                        : TwentyFiveGigE2/5/0/1
    Global Port Number          : 961
    Local Port Number           : 1
    Asic Port Number            : 1
    Asic Instance               : 0
    Vlan                        : 119
    Mapped Vlan ID              : 5
    STP Instance                : 4
    BlockForward                : 0
    BlockLearn                  : 0
    L3 Interface                : 73
        IPv4 Routing            : enabled
        IPv6 Routing            : enabled
        Vrf Id                  : 0
    Adjacency:
        Station Index           : 179
        Destination Index       : 965
        Rewrite Index           : 2
        Replication Bit Map     : 0x4     ['localData']
    Decision:
        Destination Index       : 965     [DI_PORT_GPN]
        Rewrite Index           : 2       [RI_L2]
        Dest Mod Index          : 0       [IGR_FIXED_DMI_NULL_VALUE]
        CPU Map Index           : 0       [CMI_NULL]
        Forwarding Mode         : 0       [Bridging]
        Replication Bit Map     :         ['localData']
        Winner                  :         L2DESTMACVLAN LOOKUP
        Qos Label               : 1
        SGT                     : 0
        DGTID                   : 0
```

Real World

Degraded Perform



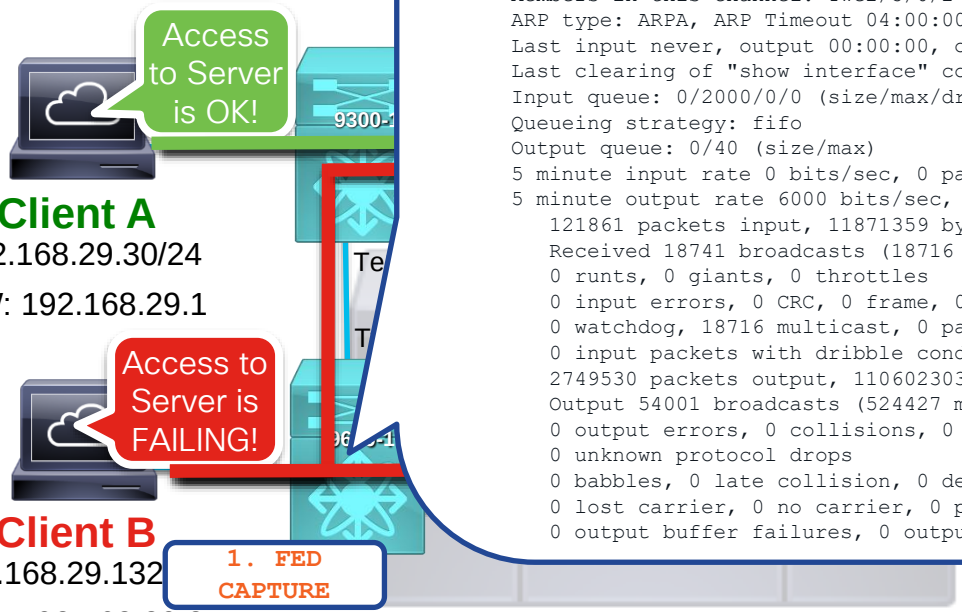
Client A
192.168.29.30/24
GW: 192.168.29.1

Client B
192.168.29.132
GW: 192.168.29.2

```
Egress:
   Possible Replication       :
       Port                   : TwentyFiveGigE2/5/0/5
   Output Port Data           :
     Port                     : TwentyFiveGigE2/5/0/5
       Global Port Number     : 965
       Local Port Number      : 5
       Asic Port Number       : 17
       Asic Instance          : 0
       Unique RI              : 0
       Rewrite Type           : 0       [Unknown]
       Mapped Rewrite Type    : 4       [L2_BRIDGE_INNER_IPv4]
       Vlan                   : 119
       Mapped Vlan ID         : 5

Output Packet Details:
   Port                       : TwentyFiveGigE2/5/0/5
###[ Ethernet ]###
  dst       = 00:00:0c:07:ac:02
  src       = 00:50:56:b4:23:4f
  type      = 0x8100
###[ 802.1Q ]###
     prio      = 0
     id        = 0
     vlan      = 119
     type      = 0x800
###[ IP ]###
        version   = 4
        ihl       = 5
        tos       = 0x0
        len       = 60
        id        = 37744
        flags     = DF
        frag      = 0
        ttl       = 64
        proto     = tcp
        chksum    = 0xe874
        src       = 192.168.29.132
        dst       = 192.168.32.2
        options   = ''
```

Real World

Degraded Perform

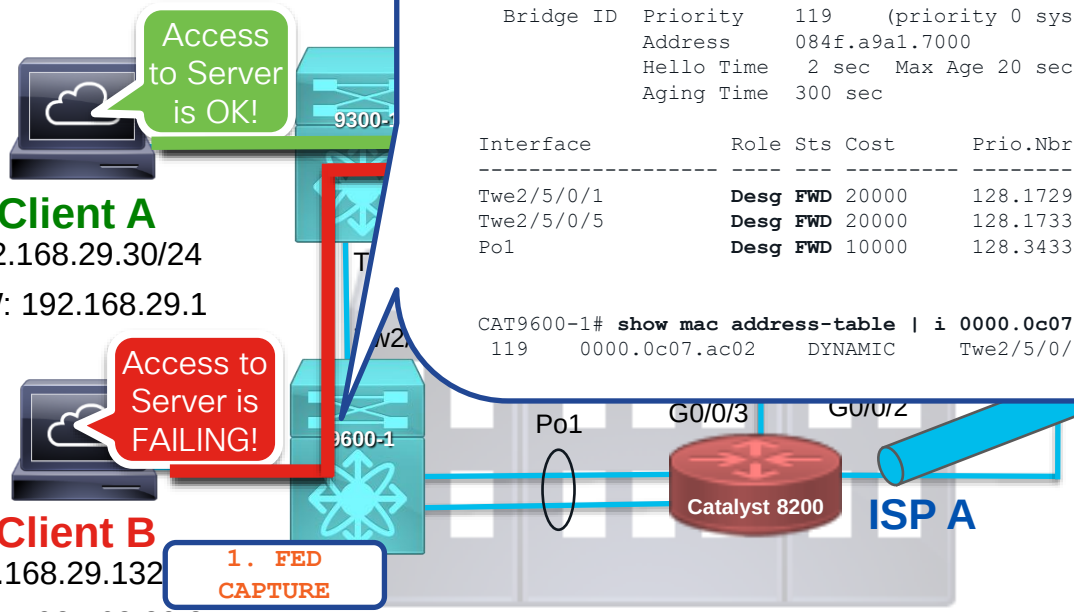



```
CAT9600-1# show int Port-channel 1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is dc8c.3772.fc82 (bia dc8c.3772.fc82)
  MTU 1500 bytes, BW 2000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s, link type is auto, media type is N/A
  input flow-control is on, output flow-control is unsupported
  Members in this channel: Twe2/5/0/2 Twe2/5/0/3
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:00, output hang never
  Last clearing of "show interface" counters 1d19h
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 6000 bits/sec, 11 packets/sec
     121861 packets input, 11871359 bytes, 0 no buffer
     Received 18741 broadcasts (18716 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 18716 multicast, 0 pause input
     0 input packets with dribble condition detected
     2749530 packets output, 1106023032 bytes, 0 underruns
     Output 54001 broadcasts (524427 multicasts)
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

Real World

Degraded Perform

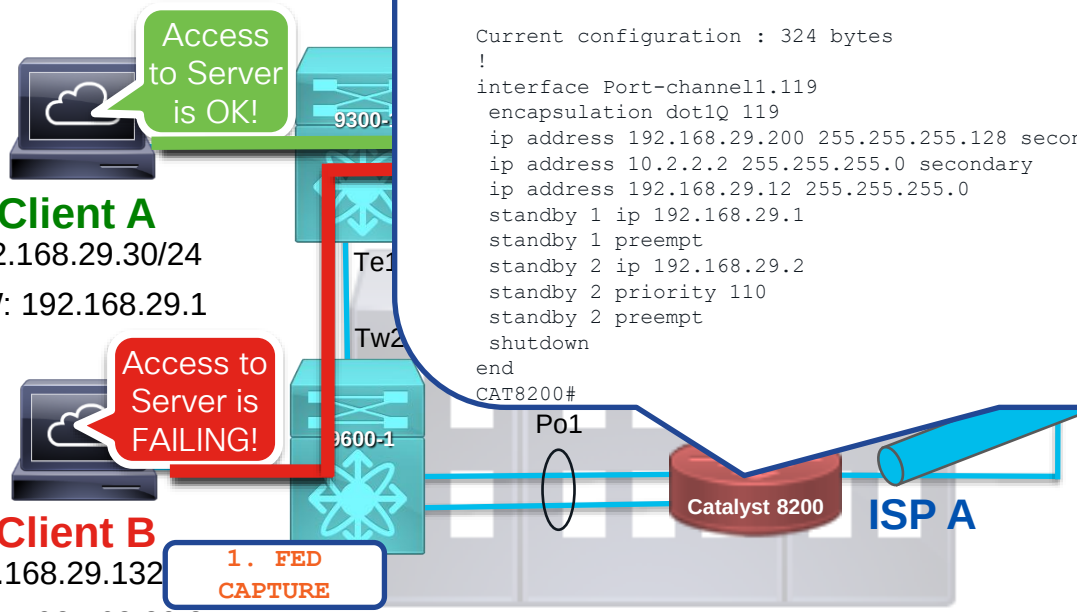
```
CAT9600-1# show spanning-tree vlan 119

VLAN0119
  Spanning tree enabled protocol rstp
  Root ID     Priority    119
              Address     084f.a9a1.7000
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    119    (priority 0 sys-id-ext 119)
              Address     084f.a9a1.7000
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Twe2/5/0/1          Desg FWD 20000     128.1729 P2p
Twe2/5/0/5          Desg FWD 20000     128.1733 P2p
Po1                 Desg FWD 10000     128.3433 P2p


CAT9600-1# show mac address-table | i 0000.0c07.ac02
 119    0000.0c07.ac02    DYNAMIC     Twe2/5/0/5
```



Real World

Degraded Perform

```
CAT8200# show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State   Active          Standby         Virtual IP
Po1.119     1    100 P Init    unknown         unknown         192.168.29.1
Po1.119     2    110 P Init    unknown         unknown         192.168.29.2
CAT8200#

CAT8200# show run int po1.119
Building configuration...

Current configuration : 324 bytes
!
interface Port-channel1.119
 encapsulation dot1Q 119
 ip address 192.168.29.200 255.255.255.128 secondary
 ip address 10.2.2.2 255.255.255.0 secondary
 ip address 192.168.29.12 255.255.255.0
 standby 1 ip 192.168.29.1
 standby 1 preempt
 standby 2 ip 192.168.29.2
 standby 2 priority 110
 standby 2 preempt
 shutdown
end
CAT8200#
```



Real Wor

Degraded Per

```
cisco@linux2:~$ iperf3 -b 200M -c 192.168.32.2
Connecting to host 192.168.32.2, port 5201
[  5] local 192.168.29.132 port 39510 connected to 192.168.32.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  23.9 MBytes   200 Mbits/sec   12    133 KBytes
[  5]   1.00-2.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   2.00-3.00   sec  23.8 MBytes   199 Mbits/sec    0    133 KBytes
[  5]   3.00-4.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   4.00-5.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   5.00-6.00   sec  23.8 MBytes   199 Mbits/sec    0    133 KBytes
[  5]   6.00-7.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   7.00-8.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   8.00-9.00   sec  23.9 MBytes   200 Mbits/sec    0    147 KBytes
[  5]   9.00-10.00  sec  23.9 MBytes   200 Mbits/sec    0    147 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   311 MBytes   200 Mbits/sec   12             sender
[  5]   0.00-10.00  sec   313 MBytes   200 Mbits/sec                  receiver
iperf Done.
cisco@linux2:~$
```

Access to Server is OK!

Client A

192.168.29.30/24

GW: 192.168.29

Access to Server is OK!

Client B

192.168.29.132

GW: 192.168.29.2

1. FED CAPTURE

2. EPC into Packet Trace (SPF)

Po1

G0/0/3

G0/0/2

Catalyst 8200

ISP A

```
CAT8200# config t
CAT8200(config)# int po1.119
CAT8200(config-if)# no shutdown
CAT8200(config-if)# end
CAT8200#
```

G3

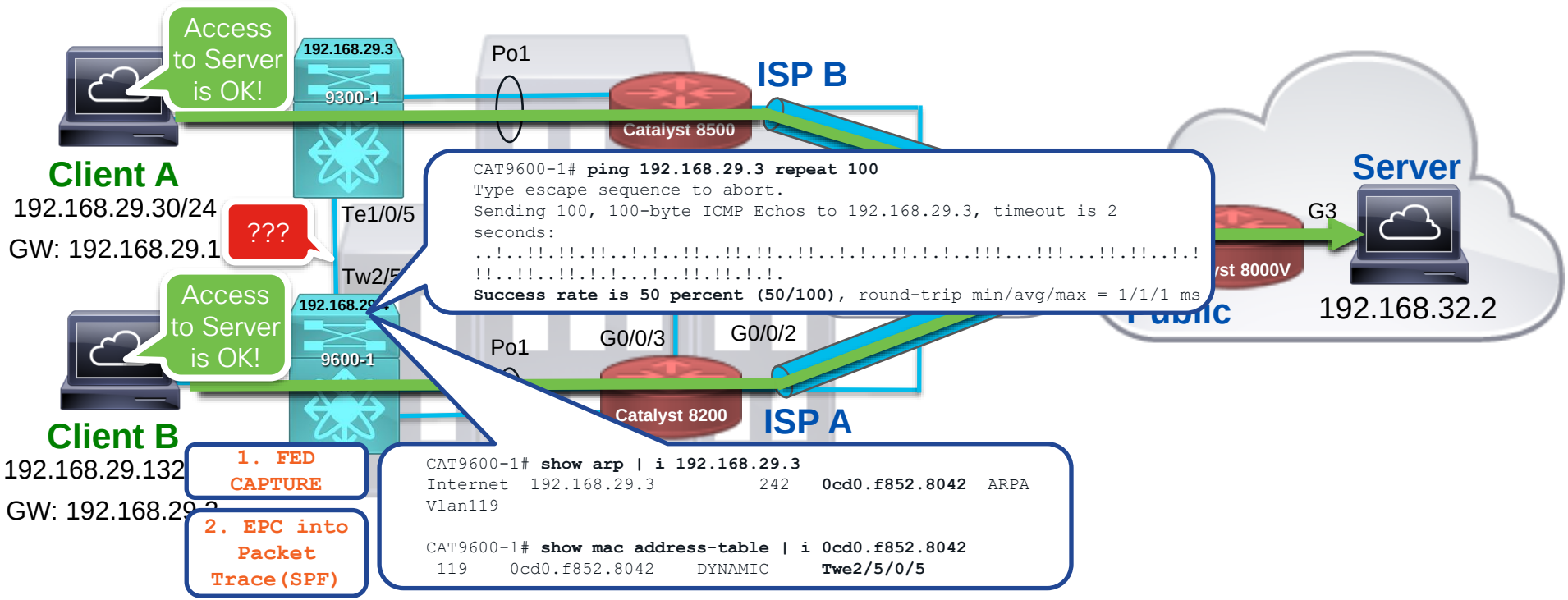
Server

192.168.32.2

Public

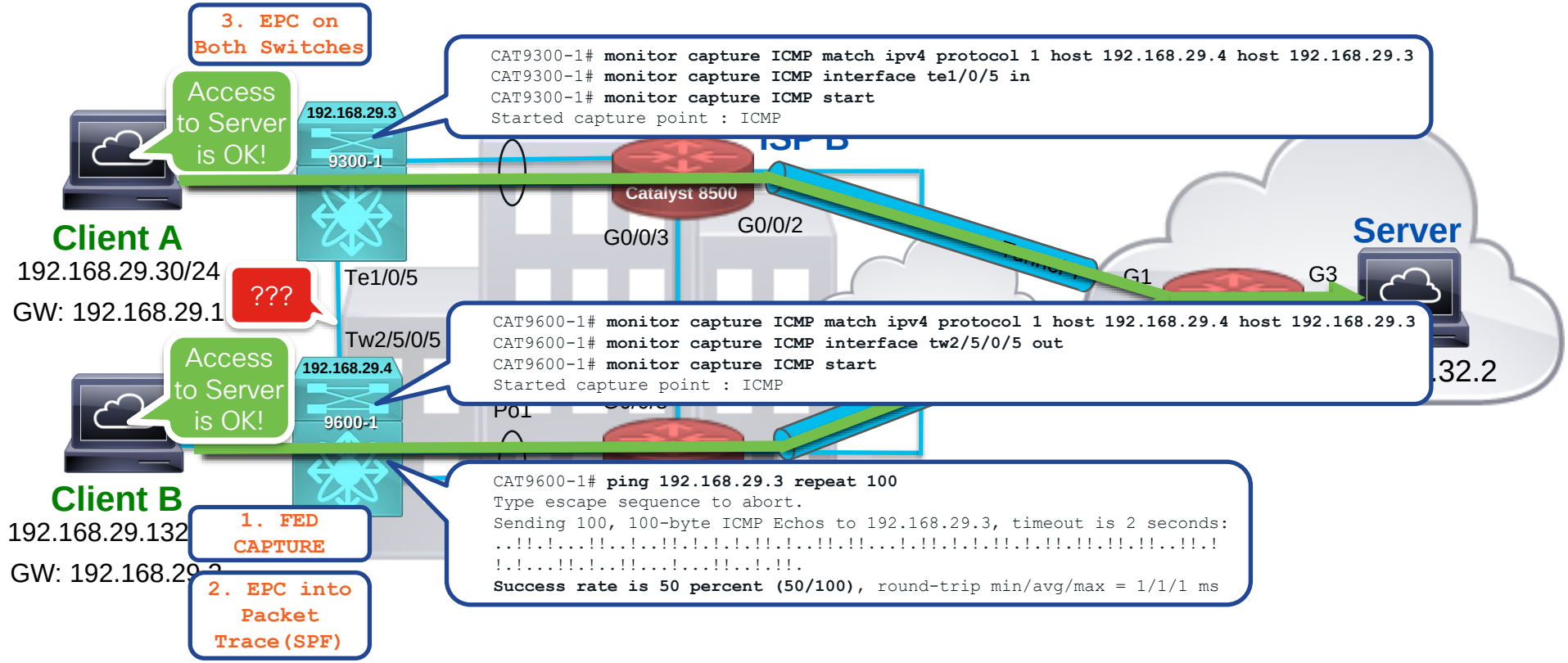
Real World Example

Degraded Performance



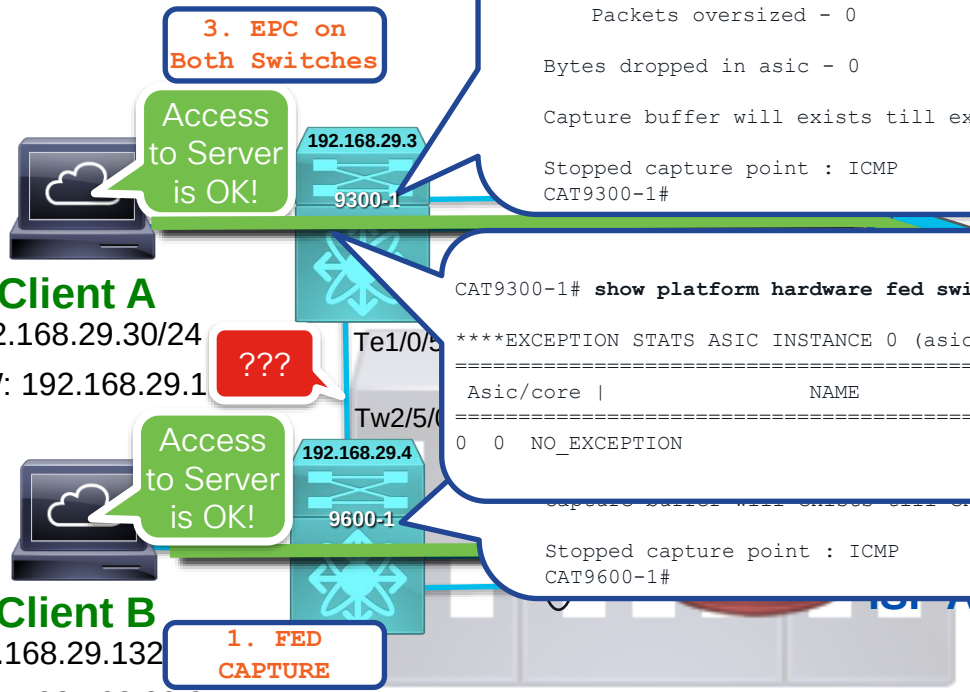
Real World Example

Degraded Performance



Real World E

Degraded Performanc




```
CAT9300-1# monitor capture ICMP stop
Capture statistics collected at software:
    Capture duration - 39 seconds
    Packets received - 50
    Packets dropped - 0
    Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared

Stopped capture point : ICMP
CAT9300-1#
```

```
CAT9300-1# show platform hardware fed switch active fwd-asic drops exceptions | ex 0          0           0

****EXCEPTION STATS ASIC INSTANCE 0 (asic/core 0/0)****
================================================================================
 Asic/core |                NAME                   |   prev   |  current  |  delta
================================================================================
0   0  NO_EXCEPTION                                  50920      51021       101
```

```
Stopped capture point : ICMP
CAT9600-1#
```

Real World Example

Degraded Perfo

```
cisco@linux2:~$ iperf3 -b 200M -c 192.168.32.2
Connecting to host 192.168.32.2, port 5201
[  5] local 192.168.29.132 port 37510 connected to 192.168.32.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  23.8 MBytes   199 Mbits/sec   12    133 KBytes
[  5]   1.00-2.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   2.00-3.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   3.00-4.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   4.00-5.00   sec  23.8 MBytes   199 Mbits/sec    0    133 KBytes
[  5]   5.00-6.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   6.00-7.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   7.00-8.00   sec  23.9 MBytes   200 Mbits/sec    0    133 KBytes
[  5]   8.00-9.00   sec  23.8 MBytes   199 Mbits/sec    0    147 KBytes
[  5]   9.00-10.00  sec  23.9 MBytes   200 Mbits/sec    0    147 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   312 MBytes   200 Mbits/sec   12             sender
[  5]   0.00-10.00  sec   311 MBytes   200 Mbits/sec                  receiver
iperf Done.
cisco@linux2:~$
```

3. EPC or
Both Switch

Access to Server is OK!

Client A
192.168.29.30/
GW: 192.168.2

Access to Server is OK!

Client B
192.168.29.132
GW: 192.168.29.2

1. FED CAPTURE

2. EPC into Packet Trace (SPF)

CAT8200#

Po1

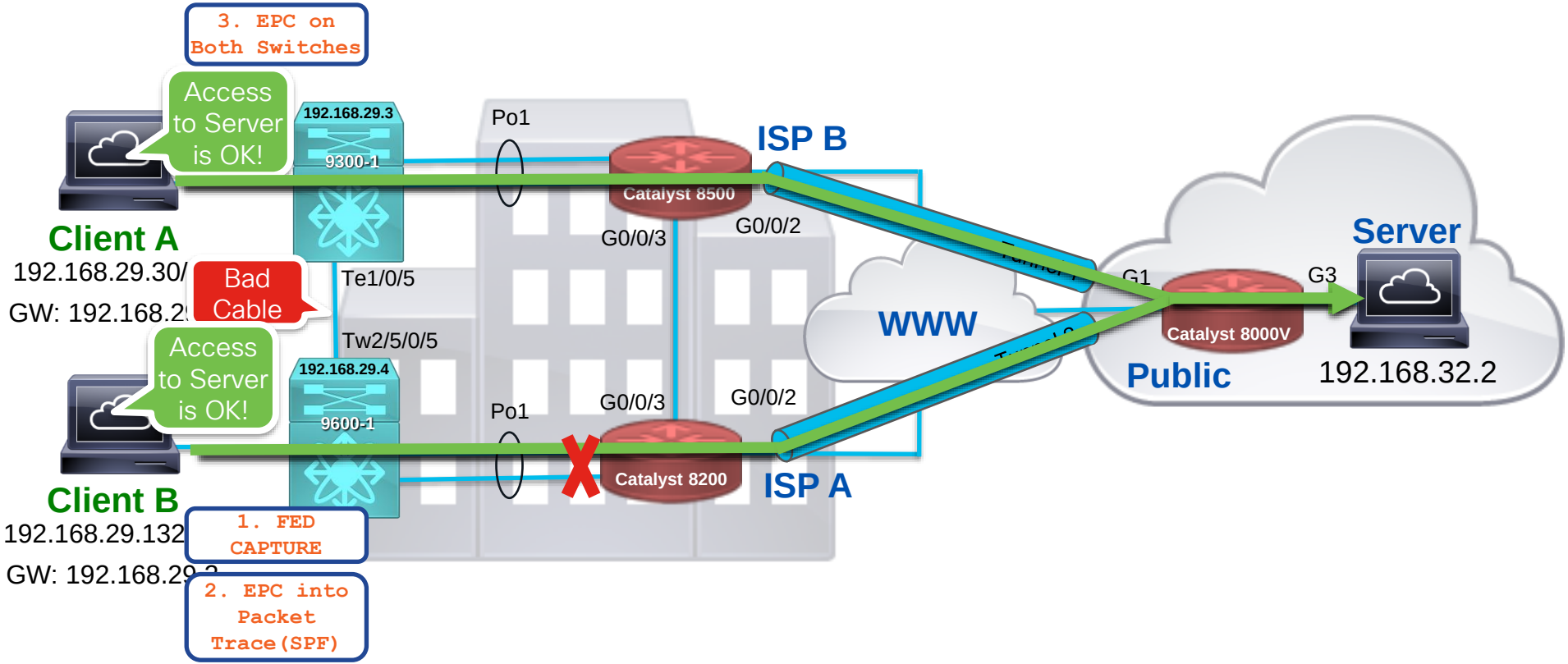
Server
192.168.32.2

Public

```
CAT9600-1# ping 192.168.29.3 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.168.29.3, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/1 ms
```

Real World Example

Degraded Performance



Summary & Take Away

Packet Capturing Tools

Usage Considerations

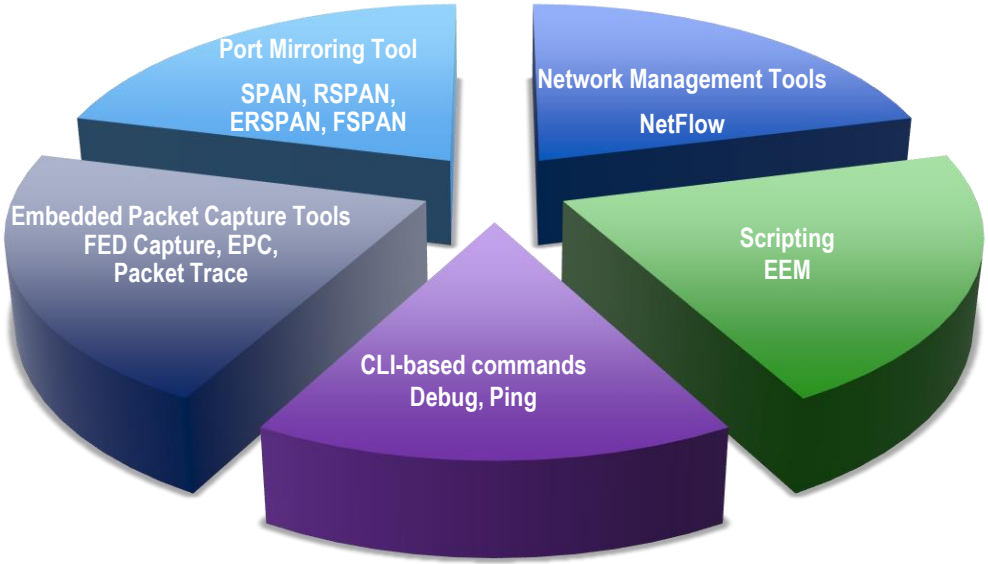
Tool	Impact	Comments
Show commands		This command shows detail of the packets in the system buffer
Catalyst 3650/3850/9000 FED Tracing/Packet Capture		Uses limited CPU/memory resources and can be run during high CPU utilization
Flexible NetFlow		For software-based forwarding platforms, this feature utilizes memory/buffer and CPU cycles. For hardware-based forwarding platforms, number of flows is limited by the hardware capacity.
SPAN / RPSAN / ERSPAN		Packet replication is performed by a specific ASIC. With oversubscription, this could cause adverse effects. With RSPAN, the replicated traffic may get flooded throughout the network. ERSPAN may require CPU cycles for decapsulation.
Embedded Packet Capture		The traffic captured by these tools is saved in the system memory/buffer. It is recommended to fine-tune the capture filters/ACLs to reduce the number of packets captured, size of the packets, etc.
Packet Trace (Switches)		Captures a single packet (PSV) or mimics packet forwarding decision (SPF). Perfectly safe to run with any packet type or conditions.
Packet Trace (Routers)		Similar concerns to Embedded Packet Capture (above)
Debug Commands		Use caution, can increases CPU utilization, filters reduce impact

Packet Capturing Tool	Control Plane	Data Plane	PCAP	Header Info	Full Packet	Local Viewing	Remote Viewing	Filtering	Single Packet	Forwarding Information	CLI Analyzer Support	Platform
Flexible NetFlow	●	●		●		●	●			●		All
FED Tracing/Packet Capture	●			●	●	●		●				3650/3850 & 9000
SPAN/RSPAN/ERSPAN		●	●	●	●		●					Switches & IOS-XE Routers
Embedded Packet Capture	●	●	●	●	●	●	●	●			●	All Routers /IOS-XE Switches
Packet Trace (Routers)		●		●	●	●		●		●		IOS-XE Routers
Packet Trace (Switches-SPF)	●	●		●		●			●	●		Catalyst 9000 Series UADP2.0, 2.0 mini, & 3.0
Packet Trace (Switches-PSV)		●		●		●		●	●	●		Catalyst 9600 & 9500H UADP 3.0

Overview of Troubleshooting Tools

Summary and Take Away ....

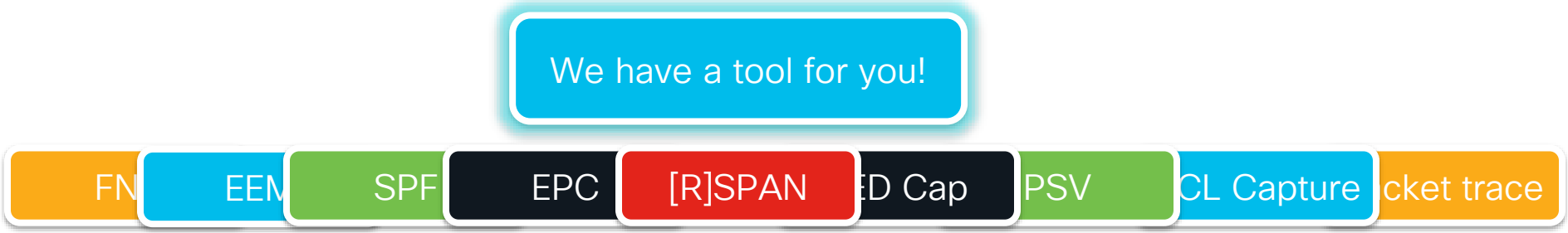
- Capture Control Plane traffic:
  - Show/Debug Commands, FED Tracing,
- Copy Data Plane traffic to external device:
  - SPAN/RSPAN/ERSPAN
- Copy Data Plane traffic to internal buffer:
  - Flexible NetFlow (FNF), Embedded Packet Capture (EPC), and Packet Trace (Routers)
- Capture Data Plane Frame and Forwarding Headers:
  - Packet Trace (Switches)



Overview of Troubleshooting Tools

Summary and Take Away ….

- Cisco Routers and Switches are advanced and feature-rich, built with keeping end-users in mind and also network engineers.
- Cisco provides a rich set of packet capturing tools embedded and supported across the spectrum of our products. These tools give visibility into the products, helping to validate the path-of-the-packet and isolate problems.
- Knowing the tools and capabilities available on each platform will reduce the time to resolution of network issues.



References

- Catalyst 9000 Switching Family
  https://www.cisco.com/c/dam/m/cs_cz/training-events/webinars/tech-club-webinars/catalyst9000-switching-innovations.pdf

References

Flexible NetFlow

- Flexible NetFlow - Data Sheets, Q&A and White Paper
  http://www.cisco.com/en/US/products/ps6965/products_ios_protocol_option_home.html
- Migrating from Traditional to Flexible NetFlow
  http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/ps6965/white_paper_c11-545581.html
- NetFlow Version 9
  http://www.cisco.com/en/US/products/ps6645/products_ios_protocol_option_home.html
- Flexible NetFlow - Data Sheets and Literature
  http://www.cisco.com/en/US/products/ps6601/prod_literature.html

References

Embedded Packet Capture Tools

- Cisco IOS Embedded Packet Capture – Data Sheet: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps9913/datasheet_c78-502727.html
- Utilizing the New Packet Capture Feature – Cisco Support Community: https://supportforums.cisco.com/docs/DOC-5799
- Embedded Wireshark – Configuration Guide: http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/15.1/XE_330SG/configuration/guide/wireshrk.html
- Mini Protocol Analyzer – Configuration Guide for 12.2SX release: http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/mpa.html
- IOS-XE Packet Trace:

  http://www.cisco.com/c/en/us/support/docs/content-networking/adaptive-session-redundancy-asr/117858-technote-asr-00.html

Overview of Troubleshooting Tools

More to Know and Learn…

- Cisco IOS Embedded Event Manager (EEM) provides a real-time network event detection and onboard automation. http://www.cisco.com/en/US/products/ps6815/products_ios_protocol_group_home.html
- Cisco IP Service Level Agreements (SLAs) assures network service levels and proactively monitors network health and performance. http://www.cisco.com/en/US/products/ps6602/products_ios_protocol_group_home.html

Overview of Troubleshooting Tools

More to Know and Learn…

- Cisco ELAM allows capturing of a single packet during the forwarding decision on hardware forwarded platforms.
  http://www.cisco.com/c/en/us/support/docs/switches/nexus-7000-series-switches/116648-technote-product-00.html

  https://www.cisco.com/c/en/us/support/docs/switches/nexus-9000-series-switches/213848-nexus-9000-cloud-scale-asic-tahoe-nx-o.html

- Cisco Packet Tracer on the Nexus 9000 allows for visibility into the forwarding path of traffic through the switch.
  https://www.cisco.com/c/en/us/support/docs/switches/nexus-9000-series-switches/210592-Nexus-9000-Packet-Tracer-tool-explained.html

Continue Your Education

- Walk-in Self-Paced Labs (45-60 minutes, self paced labs)
  - **LABTRS-2456 – Packet Capturing Tools in Routing Environments**
    Hands-on scenarios with Embedded Packet Capture, Packet Trace, and NetFlow
  - **LABTRS-2391 – Packet Capturing Tools in Enterprise Switching Environments**
    Hands-on scenarios focused on Embedded Packet Capture, FED Packet Capture, and Show Platform Forward on the Catalyst 9000 Series Switches
  - **LABTRS-2048 – Packet Trace and Conditional Debugger on IOS-XE Routers**
    Hands-on scenarios with Packet Trace and Conditional Debugger to answer questions about specific traffic flows and the features configured.

Additional Breakout Sessions and Labs

For more info…

LABTRS-2456 Packet Capturing Tools in Routing Environments
LABTRS-2391 Packet Capturing Tools in Enterprise Switching Environments
LABTRS-2048 Packet Trace and Conditional Debugger on IOS-XE Routers

LABCRT-2452 CCNP ENCOR – Core Enterprise Network Technologies Practice Lab
LABCRT-2460 CCNP ENARSI – Implementing Cisco Enterprise Advanced Routing and Services Practice Lab
LABCRT-2464 Troubleshoot like a CCNP Practice Lab

BRKTRS-3475 Advanced Troubleshooting of the cat8k, asr1k, ISR and SDWAN Edge Made Easy
BRKXAR-2003 Extending Enterprise Network into Public Cloud with Cisco Catalyst 8000V Edge Software
BRKTRS-3090 Troubleshooting Cisco Catalyst 9000 Series Switches

Cisco Learning and Certifications

From technology training and team development to Cisco certifications and learning plans, let us help you empower your business and career. www.cisco.com/go/certs

Pay for Learning with Cisco Learning Credits

(CLCs) are prepaid training vouchers redeemed directly with Cisco.

Learn

Cisco U.
IT learning hub that guides teams and learners toward their goals

Cisco Digital Learning
Subscription-based product, technology, and certification training

Cisco Modeling Labs
Network simulation platform for design, testing, and troubleshooting

Cisco Learning Network
Resource community portal for certifications and learning

Train

Cisco Training Bootcamps
Intensive team & individual automation and technology training programs

Cisco Learning Partner Program
Authorized training partners supporting Cisco technology and career certifications

Cisco Instructor-led and Virtual Instructor-led training
Accelerated curriculum of product, technology, and certification courses

Certify

Cisco Certifications and Specialist Certifications
Award-winning certification program empowers students and IT Professionals to advance their technical careers

Cisco Guided Study Groups
180-day certification prep program with learning and support

Cisco Continuing Education Program
Recertification training options for Cisco certified individuals

Here at the event? Visit us at **The Learning and Certifications lounge at the World of Solutions**

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

cisco Live!

Technical Session Surveys

- Attendees who fill out a minimum of four session surveys and the overall event survey will get Cisco Live branded socks!
- Attendees will also earn 100 points in the Cisco Live Game for every survey completed.
- These points help you get on the leaderboard and increase your chances of winning daily and grand prizes.

Thank you


cisco Live!
ALL IN