
cisco
You make possible

Catalyst 9k Troubleshooting

Michel Peters,
Technical Leader Engineering

BRKARC-3190

cisco Live!
Barcelona | January 27-31, 2020

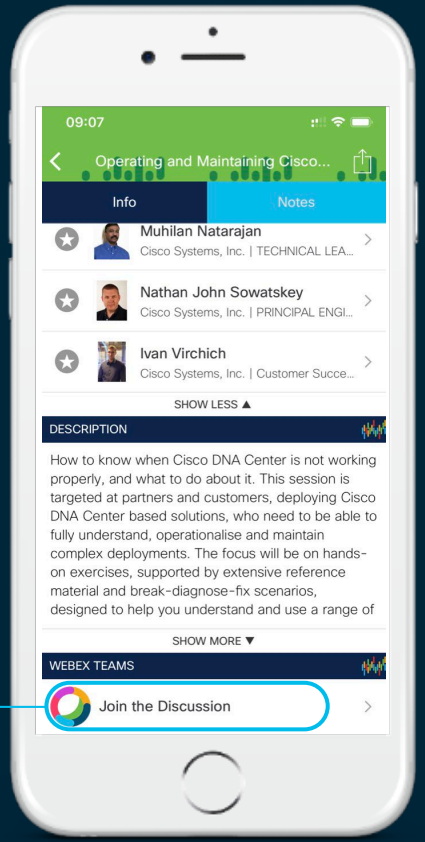
Cisco Webex Teams

Questions?

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

How

1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



Agenda

- Catalyst 9k Platform & Software Architecture
- Troubleshooting tools
- Platform Issues
- Packet Drops
- Forwarding Issues
- Questions

Platform & Software Architecture

Catalyst 9k family

Cisco Catalyst 9200 Series

Cisco Catalyst 9300 Series

Cisco Catalyst 9400 Series

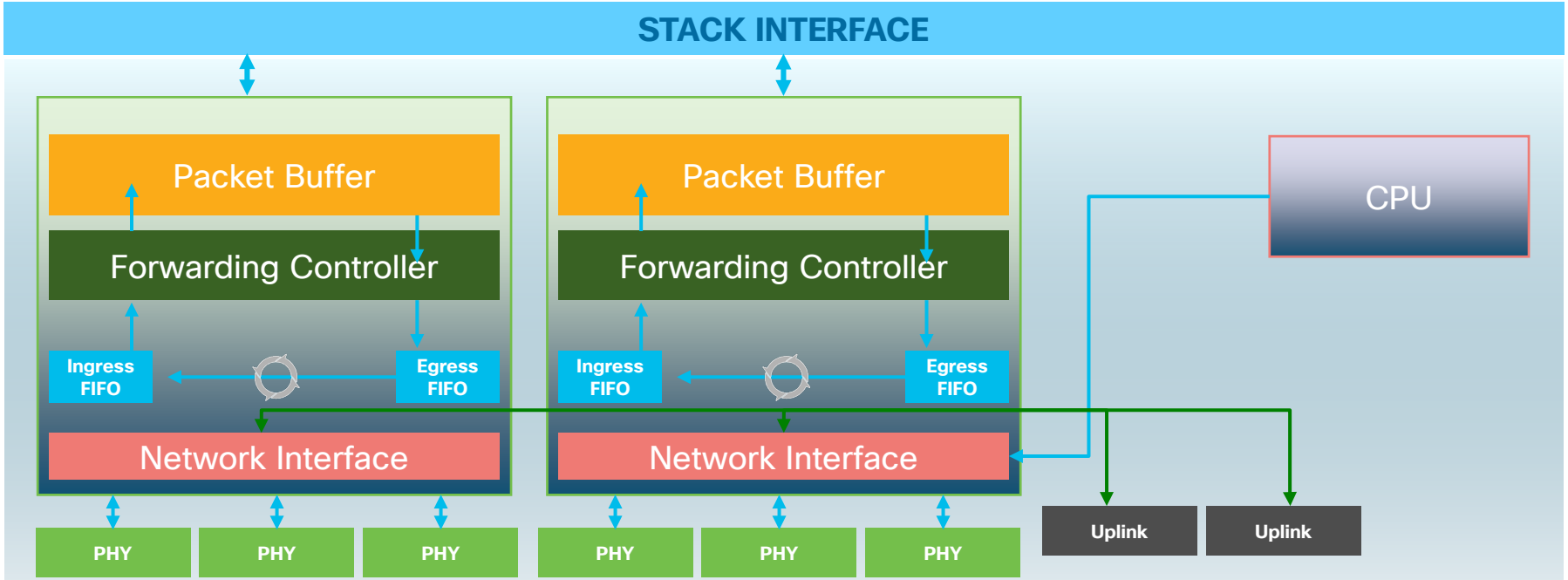
Cisco Catalyst 9500 Series

Cisco Catalyst 9600 Series

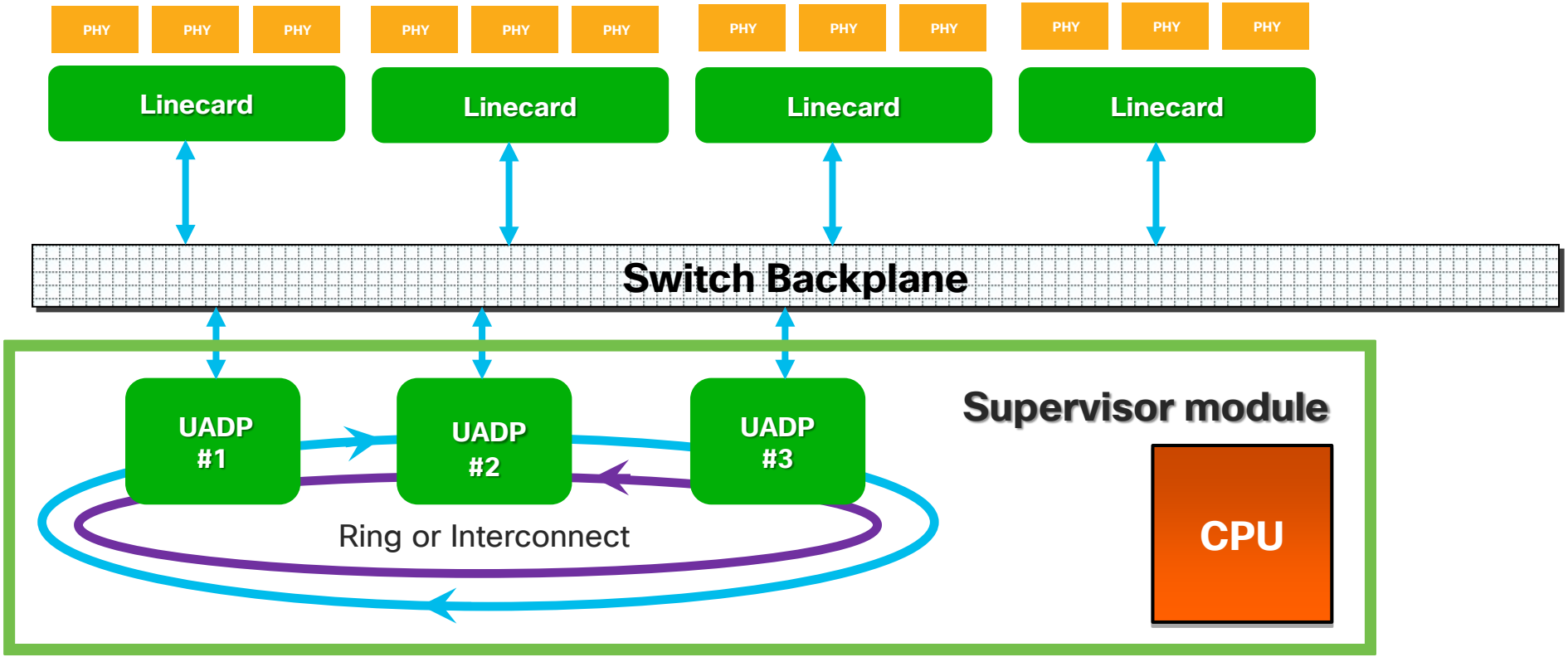
	9200	9300	9400	9500	9600
UADP	2.0 mini	2.0	2.0 XL	2.0 XL & 3.0	3.0
Format	Stackable	Stackale	Chassis	Standalone	Chassis

- UADP Asics are the foundation of all Catalyst 9K switches
- 3850/3650 switches originally launches with UADP 1.x
- Port Asics (UADP) responsible for data plan forwarding
- CPU's handle control plane traffic and forwarding tables

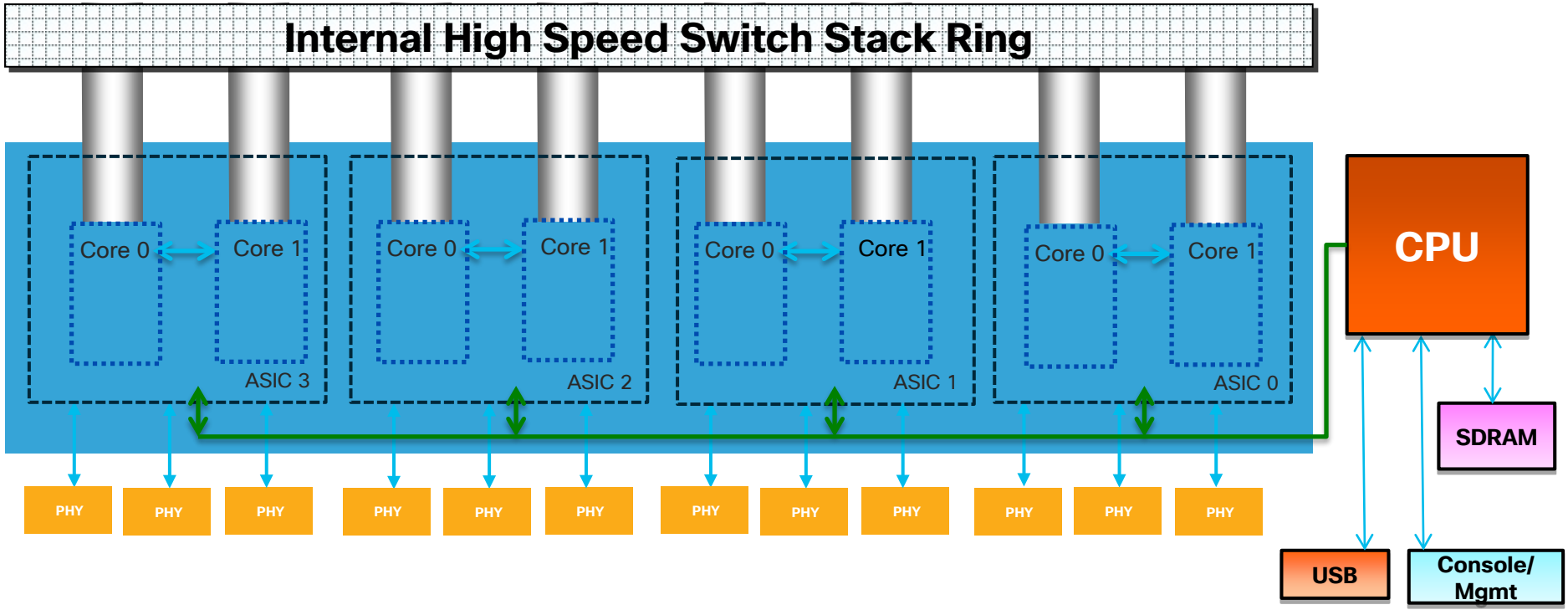
Catalyst 9200/9300 Stackable Switches



Catalyst 9400/9600 Chassis based



Catalyst 9500 Fixed Switches



Interface Internal Mappings

Interface to ASIC mapping important to understand data flows

```
9300_1#show platform software fed switch active ifm mappings
Interface                IF_ID     Inst Asic Core Port SubPort Mac  Cntx LPN  GPN  Type Active
GigabitEthernet1/0/1     0x8         1    0    1    0       0    26   6    1    1    NIF  Y
GigabitEthernet1/0/2     0x4c        1    0    1    1       0    6    7    2    2    NIF  Y
GigabitEthernet1/0/3     0x4d        1    0    1    2       0    28   8    3    3    NIF  Y
```

Internally used interface addressing:

- LPN : Local Port Number
- GPN : Global Port Number
- IF_ID : Interface Identification, used for many fed CLI
- Type : Type of interface, NIF = Network Interface
- Inst : Instance : ASIC + Core
- Port : Asic Ports
- Active : Is Interface Active , multipurpose ports

IFM Mappings 9400/9500/9600

```
9500_1#sh platform software fed switch active ifm mappings | inc 1/./[1] |Int
Interface                IF_ID    Inst Asic Core Port SubPort Mac  Cntx LPN  GPN  Type Active
TenGigabitEthernet1/0/1  0x2c     1    0    1    0    0       11   0    1    1    NIF  Y
TenGigabitEthernet1/1/1  0x18     0    0    0    16   0       0    19   17   17   NIF  N
FortyGigabitEthernet1/1/1 0x3c    0    0    0    24   0       4    4    25   25   NIF  Y

9600_1#sh platform software fed active ifm mappings | inc /0/[12][5] |Int
Interface                IF_ID    Inst Asic Core Port SubPort Mac  Cntx LPN  GPN  Type Active
FortyGigabitEternet1/0/15 0x17    3    1    1    24   0       30
HundredGigE1/0/25        0x21     0    0    0    0    0       0
TwentyFiveGigE2/0/15     0x47     1    0    1    21   0       21
TwentyFiveGigE2/0/25     0x51     3    1    1    7    0       7

C9407R#sh platform software fed active ifm mappings | inc /0/1 |Int
Interface                IF_ID    Inst Asic Core Port SubPort Mac  Cntx LPN  GPN  Type Active
GigabitEthernet2/0/1     0x7      1    0    1    0    0       7    7    1    201  NIF  Y
TenGigabitEthernet3/0/1  0x7f     4    2    0    19   0       19   3    1    301  NIF  Y
TenGigabitEthernet4/0/1  0x7e     5    2    1    3    0       11   3    1    401  NIF  Y
GigabitEthernet5/0/1     0x4b     3    1    1    0    0       4    4    1    501  NIF  Y
```

Different mappings on different platforms

More detail IF_ID mappings

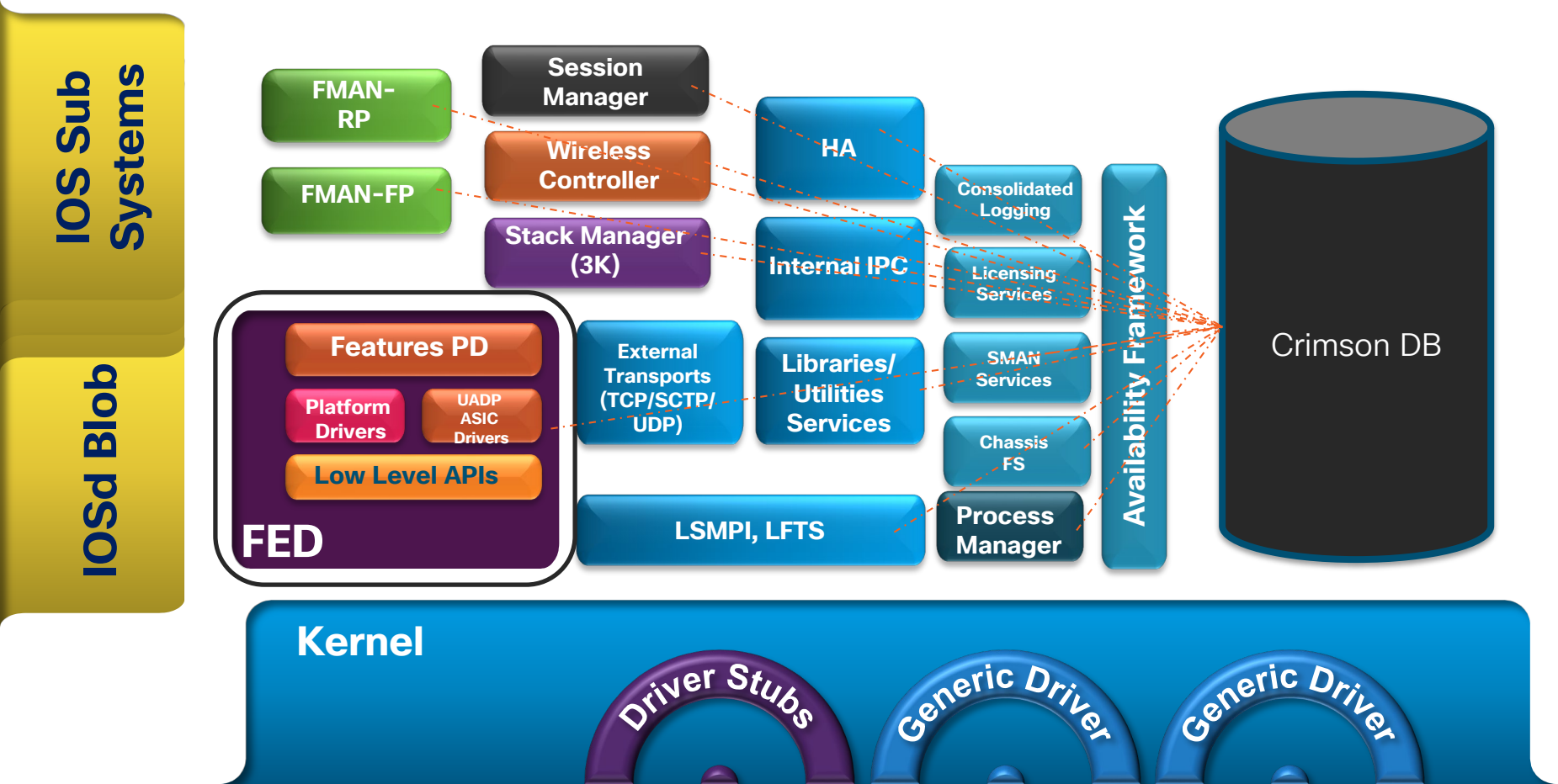
```
9300_1#sh platform software fed switch active ifm interfaces ?
  access-tunnel      Access Tunnel Interface information
  detail             Information for all type interfaces
  efp                EFP Interface Information
  ethernet           Interface ethernet information
  internal           Internal Interface information
  l2-lisp            Layer 2 LISP Interface information
  lisp               LISP Interface information
  loopback           Interface loopback information
  nve                NVE Interface Information
  svi                Interface SVI information
  sw-subif           SW Sub-interface information
  tunnel             Layer 3 Tunnel Interface Information
  vfi-segment        L2VPN VFI Segment Interface Information
  virtualportgroup   Interface vpg information
  vlan               Interface VLAN information
  wired-client       Interface wired client information
```

```
9300_1#sh plat soft fed sw act ifm int svi
Interface    IF_ID    State
-------------------------------------
Vlan1      0x0000004b      READY
Vlan100    0x0000005a      READY
```

All physical and logical Interfaces have an IF_ID

More detail about interfaces can be gathered with
show platform software fed switch <switch> ifm if-id <id>

IOS-XE 16 & IOS-XE 17 graphical overview



Kernel CPU information

```
9300_1#sh proc cpu platform sorted location switch active R0
CPU utilization for five seconds:  1%, one minute:  1%, five minutes:  1%
Core 0: CPU utilization for five seconds:  3%, one minute:  2%, five minutes:  2%
Core 7: CPU utilization for five seconds:  1%, one minute:  1%, five minutes:  0%
                  5Sec     1Min     5Min  Status        Size  Name
-------------------------------------------------------------------------------
 14416   13034       6%       6%       6%  S           223452  fed main event
 10014    9623       1%       1%       1%  S            52212  sif_mgr
  9738    9215       1%       1%       1%  S           818660  linux_iosd-imag
```

- Underlaying kernel shows Multi Core Architecture
- IOSd runs as process on kernel-> linux_iosd-image
- Not all processes run on all switches

```
Switch_1#sh processes cpu platform sorted location switch 1 R0 | in fman
 27777   26990       0%       0%       0%  S         314179584  fman_rp
 19145   17642       0%       0%       0%  S         296591360  fman_fp_image
Switch_1#sh processes cpu platform sorted location switch 3 R0 | in fman
 20643   19400       0%       0%       0%  S         296599552  fman_fp_image
```

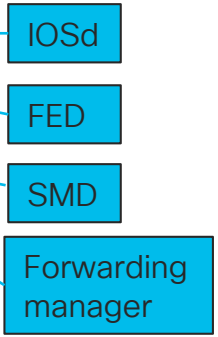
IOSd CPU utilization

```
9300_1#show processes cpu sorted
CPU utilization for five seconds: 1/0% one minute: 0%; five minutes: 0%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 434        8197     1230039          6  0.07%  0.01%  0.00%   0 MMON MENG
 203       10890      614953         17  0.07%  0.03%  0.01%   0 VRRS Main thread
 287           8          46        173  0.07%  0.00%  0.00%   0 Exec
 221       12377     1226864         10  0.07%  0.02%  0.00%   0 IP ARP Retry Age
 113       11806       20043        589  0.07%  0.03%  0.02%   0 Crimson flush tr
 218       12527     1226864         10  0.07%  0.02%  0.01%   0 IPAM Manager
 238       11425      393615         29  0.07%  0.02%  0.01%   0 UDLD
```

- Many process like still running as processes inside IOSd process (BGP, RIP, CEF, ARP, UDLD, CDP , etc)
- CPU utilization 1%/0% showing Utilization for Processes and Utilization for Interrupt
- IOSd does not provide multicore architecture

Platform Memory

```
9300_1#sh processes memory platform  sorted location switch 1 R0
System memory: 7711304K total, 2388036K used, 5323268K free,
Lowest: 5321604K
   Pid    Text      Data    Stack   Dynamic       RSS              Name
-------------------------------------------------------------------------
  9738  151604    817392      136       408    817392   linux_iosd-imag
 14416     152    224856      136     85308    224856    fed main event
 21595     238    168844      136      2780    168844               dbm
 20109     176    128988      136      5404    128988          sessmgrd
 21225    6825    117488      136      3004    117488           fman_rp
 22061     260     93396      136       124     93396         cli_agent
 22864     600     91152      136     16444     91152              smand
```



- Kernel memory utilization is available per switch
- linux_iosd-image process is IOSd
- Resident Set Size(RSS), memory occupied by each Process

IOSd Memory

```
9300_1#sh processes memory sorted
Processor Pool Total: 1445417856 Used:  290878080 Free: 1154539776
 lsmpi_io Pool Total:    6295128 Used:    6294296 Free:        832

  PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
    0   0  288949984   53306504  214634744          0          0 *Init*
    4   0   22511680     100104   22277344          0          0 RF Slave Main Th
   81   0   24107152    2420648   13717584          0          0 IOSD ipc task
  472   0    4133424     105760    4069608     849828          0 EEM ED Syslog
    0   0   62739512   58194512    2978824   23259559     382788 *Dead*
  609   0    6717728    3968088    2803904          0          0 ISIS Upd
  490   0    1719800      90880    1659432          0          0 EEM Server
```

- IOSd runs as a process , doe still provides memory management for all processes running on IOSd
- Processor Pool: Pool for Processes on IOSd
- lsmpi_io : Linux Shared Memory Punt Interface memory , IO buffers

Catalyst IOS-XE Software release schedule

	Everest 16.6	Fuji 16.9	Gibraltar 16.10	Gibraltar 16.11	Gibraltar 16.12	Amsterdam 17.1
Next planned release	16.6.8	16.9.5	None	None	16.12.3	17.1.2
9200	No	No	Yes	Yes	Yes	Yes
9300/9400/9500	Yes	Yes	Yes	Yes	Yes	Yes
9600	No	No	No	Yes	Yes	Yes
Maintenance Throttle	Yes	Yes	No	No	Yes	No

Maintenance throttles receive more rebuilds and thus would be recommended over feature releases

Not all SKU support all IOS-XE versions, above table indicative

Recommended releases: https://www.cisco.com/c/en/us/support/docs/switches/catalyst-9300-series-switches/214814-recommended-releases-for-catalyst-9200-9.html

Troubleshooting Tools

Platform Specific Show commands

- Platform Independent Show commands are similar like on any IOS platform.
  Ex. Show cdp, show bgp, show udld
- IOS-XE specific troubleshooting commands are under show platform

```
Switch#show platform software ip switch active R0 cef
ASR_1k#show platform software ip rp active cef
```

- *Show platform [hardware|software] fed* contain Platform Dependent troubleshooting for the Catalyst 3k and 9k (FED layer)

```
9300#show platform software fed switch active ifm mappings
9300#show platform software fed switch 5 ifm mappings
9400#show platform software fed active ifm mappings
```

- Outputs might vary depending on if it executed on active/standy and or/on Forwarding Processor or Route Processon

Debugging IOSd processes

- Features running inside IOSd can be debugged in traditional methods/debugs

```
9300_1#debug arp
ARP packet debugging is on
9300_1#debug ip icmp
ICMP packet debugging is on
```

- Some processes on IOSd support additional event-tracing inside IOSd

```
9300_1#sh monitor event-trace arp all
*Jan 20 03:52:48.977: RCV PKT: S: 10.100.10.100 T: 10.100.10.1 IF: Vlan100 OP: rep
*Jan 20 03:52:52.740: RCV PKT: S: 172.17.11.2 T: 172.17.11.1 IF: Vlan1 OP: req
9300_1#sh monitor event-trace  bgp all
Tracing currently disabled, from exec command
```

Global Config:
monitor event-trace ..

Debugging, non-IOSd Processes

- IOSd cannot do real time debugging on processes outside IOSd
- To facilitate debugging/logging trace logs are available per process
- Tracing levels set with granularity (default notice). Tracing always on
- Common processes: smd , fed, forwarding-manager

```
Switch#set platform software trace smd switch 1 R0 dot1x-all debug
Switch#show platform software trace level smd switch 1 R0 | inc dot1x
dot1x                    Notice
dot1x-all                Debug
dot1x-redun              Notice
```

- Processes *can* run on any switch in the stack, not just on active

Always on Tracing usage

- Tracelog files are stored in crashinfo:/logs in binary format.
- Traces can be displayed using show platform software trace command
- Archive of traces can be created using the command
  "**request platform software trace archive**"
- Archives contain binary files, not readable with text viewer

Example fed logs from switch 1.
'| inc dot1x' filters out dot1x logs

```
Switch#show platform software trace message fed switch 1 | include dot1x
2020/01/29 21:49:20.540 {smd_R0-0}{1}: [dot1x] [19244]: UUID: 0, ra: 0, TID: 0 (debug): [0000.0000.0000:Gi1/0/1]
Received pkt saddr = 70e4.22e5.c4f7 , daddr = 0180.c200.0003, pae-ether-type = 0x888E
2020/01/29 21:49:20.540 {smd_R0-0}{1}: [dot1x] [19244]: UUID: 0, ra: 0, TID: 0 (debug): [0000.0000.0000:Gi1/0/1]
Dequeued pkt: CODE = 2,TYPE = 4,LEN = 22
2020/01/29 21:49:20.540 {smd_R0-0}{1}: [dot1x] [19244]: UUID: 0, ra: 0, TID: 0 (debug): [0000.0000.0000:Gi1/0/1]
Queuing an EAPOL pkt on Authenticator Q
2020/01/29 21:49:20.540 {smd_R0-0}{1}: [dot1x] [19244]: UUID: 0, ra: 0, TID: 0 (debug): [0000.0000.0000:unknown]
Pkt body: 02 a3 00 16 04 10 12 ee 6c c4 9e 43 29 ef dd f4 0e 9c ae 75 34 0a
2020/01/29 21:49:20.540 {smd_R0-0}{1}: [dot1x] [19244]: UUID: 0, ra: 0, TID: 0 (info): [70e4.22e5.c4f7:Gi1/0/1]
EAP Packet - RESPONSE, ID : 0xa3
2020/01/29 21:49:20.540 {smd_R0-0}{1}: [dot1x] [19244]: UUID: 0, ra: 0, TID: 0 (info): [70e4.22e5.c4f7:Gi1/0/1]
Received EAPOL packet - Version : 1,EAPOL Type : EAP, Payload Length : 22, EAP-Type = MD5-Challenge
2020/01/29 21:49:20.540 {smd_R0-0}{1}: [dot1x] [19244]: UUID: 0, ra: 0, TID: 0 (debug): [0000.0000.0000:unknown]
Received EAPOL pkt (size=92u) on l2 socket
```

Embedded Packet Capture (EPC)

- EPC provides insight into both Data Plane and Control Plane traffic
- Captures can be done on Interfaces or Control Plane
- Data can be buffered and exported or stored directly in flash
- Data capture implemented on Port Asics, traffic copied to EPC process.
- EPC process provides deeper packet capture and display filtering
- Analysis can be done off-box or on box using included packet dissectors

Running Embedded Packet Capture

More granular filtering of traffic after send to wireshark process

Where to store and how long to capture traffic

What traffic gets send to wireshark process

```
Switch#monitor capture CL
  access-list     access-list to be attached
  buffer          Buffer options
  class-map       class name to attached
  clear           Clear Buffer
  control-plane   Control Plane
  export          Export Buffer
  file
  interface       Interface
  limit           Limit Packets
  match           Describe filters inline
  start           Enable Capture
  stop            Disable Capture
```

Up to 8 capture points can be defined. *One* can be active

Running Embedded Packet Capture

```
Switch#monitor capture CL interface GigabitEthernet 1/0/2 both
Switch#monitor capture CL match ipv4 any any
Switch#monitor capture CL limit duration 60
Switch#monitor capture CL file location flash:cl.cap
Switch#monitor capture CL start display brief

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


  1   0.000000 10.254.111.100 -> 10.254.254.1 TCP 74 734 b^F^R 2049 [SYN]
Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=445826583 TSecr=0 WS=128
  2   0.000501 10.254.254.1 -> 10.254.111.100 TCP 60 2049 b^F^R
Seq=1 Ack=1 Win=0 Len=0
  3   1.001299 10.254.111.100 -> 10.254.254.1 TCP 74 711 b^F^R
Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=445826833 TSecr=0 WS=128
  4   1.001582 10.254.254.1 -> 10.254.111.100 TCP 60 2049 b^F^R 711 [RST, ACK]
Seq=1 Ack=1 Win=0 Len=0

Capture CLUS stopped - Capture duration limit reached
```

Where and what to capture

Storage location and duration

Starts the capture and enables live capture

Displaying packet captures

```
Switch#show monitor capture file flash:cl.cap brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
  1   0.000000 10.200.10.100 -> 10.200.10.200 TCP 66 59498 b^F^R 80 [ACK] Seq=1 Ack=1
Switch#show moni capture file flash:cl.cap packet-number 1 detailed | be Transmission
Transmission Control Protocol, Src Port: 59498 (59498), Dst Port: 80 (80), Seq: 1, Ack:
1,  Source Port: 59498
    Destination Port: 80
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    Header Length: 32 bytes
    Flags: 0x010 (ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
        [TCP Flags: *******A****]
    Window size value: 24464
```

On Box Analysis of saved captures

Details packet decodes

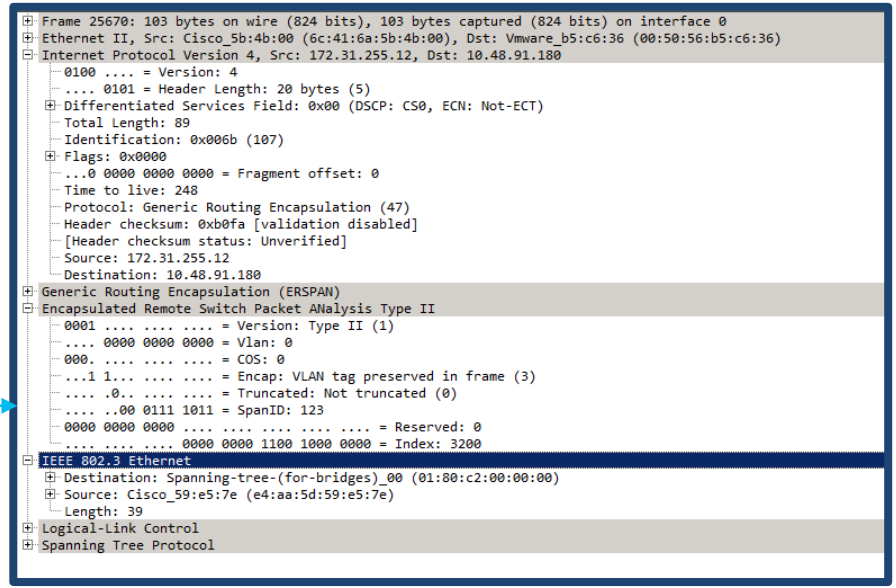
Off Box Analysis using Wireshark also possible

SPAN

- Local SPAN, local traffic copied to interface on switch (Local)
- Remote SPAN , traffic copied to/from special rspan VLAN (Layer 2 domain)
- Encapsulated RSPAN, traffic encap/decap'd using GRE (Layer 3 domain)

```
monitor session 1 type erspan-source
 source interface Gi2/0/3
 destination
  erspan-id 123
  ip address 10.48.91.180
  origin ip address 172.31.255.12
```

Using ERSPAN to send traffic from Gi2/0/3 to PC



Embedded Event Manager, reacting to events

```
event manager applet RouteChange authorization bypass

event routing network 10.48.91.0/24 type all vrf Bru

  action 0.1 cli command "enable"
  action 1.0 cli command "show ip bgp vpnv4 all | append flash:rclog.txt"
  action 1.1 cli command "show ip route vrf Bru | append flash:rclog.txt"
  action 2 syslog msg "Route change detected, flash:rclog.txt appended"
```

Definition

Trigger

Actions

- EEM framework allows actions to be taken based on a number of triggers
- Assists with information gathering and possibly corrective actions
- Automated capture allows information to be gather close to an event

Show Tech Enhancements

- Show tech contains lot of generic information, not feature specific
- For more focused information gathering show tech <keyword>
- Scripted command generation based on provided parameters
- Examples:
  - show tech-support cts
  - show tech-support port
  - show tech-support acl
  - show tech-support identy
  - show tech-support fabric

Show tech can be large, redirect to flash

```
9300_1#sh tech identity mac 0001.0001.0001 interface Gi 1/0/1 | redirect flash:shtech.log
9300_1#dir flash:shtech.log
Directory of flash:/shtech.log
671754  -rw-           1504931  Jun 10 2019 00:07:47 +00:00  shtech.log
11353194496 bytes total (9337597952 bytes free)
```

Show tech enhancement example

- Show tech enhancements go further then just show commands
- Results of outputs are used to generate and execute follow up commands

```
9300_1#sh tech-support cef ipv4 10.48.91.128/25 | re flash:shtechcef.log
9300_1#more flash:shtechcef.log | inc ---- show
------------------ show clock ------------------
------------------ show version ------------------
------------------ show running-config ------------------
------------------ show cef state ------------------
------------------ show cef state capabilities ------------------
------------------ show cef table internal ------------------
------------------ show ip cef vrf Default 10.48.91.128/25 internal ------------------
------------------ show ip route 10.48.91.128 255.255.255.128 ------------------
------------------ show interface TenGigabitEthernet1/1/6 ------------------
------------------ show cef interface TenGigabitEthernet1/1/6 internal ------------------
------------------ show adj link IP TenGigabit1/1/6 172.31.250.30 connectionid 0 internal
------------------ show arp 172.31.250.30 TenGigabitEthernet1/1/6 detail ---
------------------ show monitor event-trace cef ipv4 vrf Default 10.48.91.128 all ----
```

Programmability & Troubleshooting

- IOSXE Programmability helps gathering information and detecting failures
- Ability to execute CLI on IOSd, parse and possibly react to events
- Python (Guest shell)
  Python on guestshell allows interaction with IOSd, config, cli etc

```
9300_1#guestshell enable
9300_1#guestshell run python3
Python 3.6.3 (default, Nov  1 2018, 15:47:26)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-36)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cli
>>> print (cli.execute("show run | inc app-|iox" ))
iox
app-hosting appid guestshell
 app-vnic management guest-interface 0
```

Programmability & Troubleshooting

- TCL included in IOS-XE.
- Does not require guestshell to be enabled
- TCL scripts can be sourced from filesystem or inputted

```
9300_1#tclsh
9300_1(tcl)#puts [ exec "show ver" ]
Cisco IOS XE Software, Version 17.01.01
Cisco IOS Software [Amsterdam], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 17.1.1, RELEASE
SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
```

```
9300_1#tclsh tftp://10.48.91.151/cl.tcl
Loading cl.tcl from 10.48.91.151 (via TenGigabitEthernet1/1/6): !
[OK - 418 bytes]
Number of Up CLNS neigbours 4 Number of up BFD sessions 4
 Number of sessions match
```

Platform Issues

Platform Health overview

Overview of Memory & CPU usages

```
9300_1#show platform resources
**State Acronym: H - Healthy, W - Warning, C - Critical
Resource                Usage                Max            Warning        Critical       State
-------------------------------------------------------------------------------------------------
 Control Processor      1.39%                100%           5%             10%            H
  DRAM                  2336MB(31%)          7530MB         90%            95%            H
```

Yellow threshold and Fan Fail action configurable.

```
9300_1#show environment all
Switch 1 FAN 1 is OK
Switch 1 FAN 2 is OK
Switch 1 FAN 3 is OK
FAN PS-1 is OK
FAN PS-2 is NOT PRESENT
Switch 1: SYSTEM TEMPERATURE is OK
Inlet Temperature Value: 37 Degree Celsius
Temperature State: GREEN
Yellow Threshold : 46 Degree Celsius          Red Threshold    : 56 Degree Celsius
Hotspot Temperature Value: 56 Degree Celsius
Temperature State: GREEN
Yellow Threshold : 105 Degree Celsius.       Red Threshold    : 125 Degree Celsius
SW   PID                 Serial#     Status           Sys Pwr  PoE Pwr  Watts
--   ------------------  ----------  ---------------  -------  -------  -----
1A   PWR-C1-715WAC       DCA2120G1D4  OK              Good     Good     715
1B   Not Present
```

CPU Inband Traffic

- Dataplane forwarding occurring on UADP without CPU interaction
- CPU Destined traffic sent to corresponding CPU queue by UADP aisc
- CPU queues:
  - Routing Protocol, ex EIGRP, OSPF, PIM, etc
  - Layer 2 Control , ex Spanning Tree, UDLD , etc
  - Software Forwarding, Traffic hitting exception, TCAM overload
  - Host Traffic (For us), ex SNMP, SSH, NETCONF, ICMP
  - ICMP, traffic needing ICMP generation
  - ..
- Control Plane policing default *enabled* to protect CPU resources

Inband Packet Statistics

```
9300_1#sh controllers cpu-interface
queue                    retrieved    dropped      invalid      hol-block
-------------------------------------------------------------------------
Routing Protocol         42856        0            0            0
L2 Protocol              52577        0            0            0
sw forwarding            1            0            0            0
broadcast                11360        0            0            0
icmp gen                 0            0            0            0
icmp redirect            0            0            0            0
logging                  0            0            0            0
rpf-fail                 0            0            0            0
DOT1X authentication     0            0            0            0
Forus Traffic            120          0            0            0
Forus Resolution         46750        0            0            0
Inter FED                0            0            0            0
L2 LVX control           0            0            0            0
EWLC control             0            0            0            0
EWLC data                0            0            0            0
L2 LVX data              0            0            0            0
Openflow                 0            0            0            0
Topology control         85166        0            0            0
Proto snooping           0            0            0            0
DHCP snooping            0            0            0            0
```

- Counts packets received at CPU
- Different Queues to handle different classes of traffic for CPU

Control Plane Policing Statistics

```
9300_1#show policy-map control-plane
 Control Plane
  Service-policy input: system-cpp-policy
    Class-map: system-cpp-police-ios-routing (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: none
      police:
          rate 13000 pps, burst 3173 packets
        conformed 379638519 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          drop
    Class-map: system-cpp-police-ios-feature (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: none
      police:
          rate 6000 pps, burst 1464 packets
        conformed 20422413 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          drop
```

- Multiple Queues might map to one class map
- COPP policy map configurable (not recommended)
- Drops in critical queues can lead to instability

Control Plane Policing HW stats

```
9300_1#show plat hardware fed switch 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
                       CPU Queue Policer Statistics
Policer     Policer Accept   Policer Accept  Policer Drop  Policer Drop
  Index          Bytes            Frames         Bytes           Frames
-----------------------------------------------------------------------
0           17261371         11408           0             0
1           17682901         52775           0             0
2           357304765        43037           0             0
                          CPP Classes to queue map
=================================================================================
PlcIdx CPP Class                                  :  Queues
---------------------------------------------------------------------------------
0      system-cpp-police-data                     :  ICMP GEN/ BROADCAST/ ICMP Redirect/
10     system-cpp-police-sys-data                 :  Openflow/ Exception/ EGR Exception/ NFL SAMPLED DATA/ RPF
13     system-cpp-police-sw-forward               :  Sw forwarding/ LOGGING/ L2 LVX Data Pack/ Transit Traffic
```

Per queue stats

Policer stats

Mappings

Determining where CPU traffic originates from

```
9500_1#show plat software fed switch active punt rates interfaces
Punt Rate on Interfaces Statistics
Packets per second averaged over 10 seconds, 1 min and 5 mins
```

Active Interfaces sending to CPU

Interface Name	IF_ID	Recv 10s	Recv 1min	Recv 5min	Drop 10s	Drop 1min	Drop 5min
TenGigabitEthernet1/0/1	**0x0000002e**	**1**	**1**	**1**	**0**	**0**	**0**
FortyGigabitEthernet1/1/1	**0x00000032**	**1**	**1**	**18**	**0**	**0**	**0**

```
9500_1#show platform software fed switch active punt cpuq rates
Punt Rate CPU Q Statistics
Packets per second averaged over 10 seconds, 1 min and 5 mins
```

Per Queue Statistics

Q no	Queue Name	Rx 10s	Rx 1min	Rx 5min	Drop 10s	Drop 1min	Drop 5min
2	**CPU_Q_FORUS_TRAFFIC**	**0**	**0**	**17**	**0**	**0**	**0**
4	**CPU_Q_ROUTING_CONTROL**	**3**	**3**	**3**	**0**	**0**	**0**

Ternary Content-Addressable Memory

- UADP asics use TCAM memory for storing a majority of the forwarding related tables
- Normal Memory has 0 or 1. TCAM has 0, 1 and “don’t care”
- Provides ability to do fast lookups needed for forwarding
- TCAM stores tables with Values , Mask and Result (VMR)
- TCAM memory shared by various features. Allocation of TCAM space done using Switching Database Manager (SDM)
- Different SDM templates available depending on platform
- SDM changes require reload
- Exceeding TCAM space impacts ability to do Hardware forwarding. Depending on feature SW forwarding, drops or flooding

```
9500#show sdm prefer ?
  core            Core Template
  distribution    Distribution Template
  nat             NAT Template
  sda             SDA Template
```

Monitoring TCAM Utilization

```
9300_1#show platform hardware fed switch active fwd-asic resource tcam utilization
CAM Utilization for ASIC  [0]
 Table                                              Max Values          Used Values
 ------------------------------------------------------------------------------------
 Unicast MAC addresses                              32768/1024           81/21
 L3 Multicast entries                               8192/512             0/7
 L2 Multicast entries                               8192/512             0/9
 Directly or indirectly connected routes            24576/8192           24/37
 QoS Access Control Entries                         5120                  40
 Security Access Control Entries                    5120                 125
 Ingress Netflow ACEs                                256                   8
 Policy Based Routing ACEs                          1024                  20
 Egress Netflow ACEs                                 768                   8
  ..
```

- Not all Port Asics get programmed equally.
- Exceeding TCAM resources could lead to performance issues, traffic might be software forwarded, flooded or dropped

Dumping TCAM entries

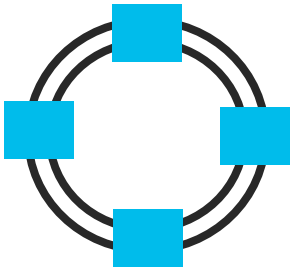
- TCAM information can be displayed per switch and per ASIC
- For forwarding check on both ingress Asic and egress Asic
- TCAM dumps show raw information, content might not be easily readable

```
9300_1#sh plat hard fed sw act fwd-asic resource tcam table ?
  acl               L3 IPv4 ACL Table
  client_le         Client LE Table
  cpp               CPP Table
  fspan             FSPAN Table
  l2                L2 Lookup Table
  l2_multicast      L2 IPv4 Multicast Forwarding Table
  l3_multicast      L3 IPv4 Multicast Forwarding Table
  qos               L3 IPv4 QoS Table
  sghash            SGT/DGT Lookup Table
  spd_lookup        SPD Lookup Table
  tunnel            Tunnel Termination Table
  unicast           L3 Unicast Table
```

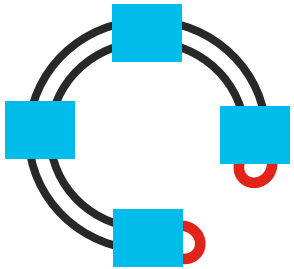
```
9300_1#sh platform hardware fed switch active fwd-asic resource tcam table l2
Printing entries for region L2_EXACT_MATCH (0) type 1 asic 0
=========================================================
HashTable 0 Left bucket 593 Index 0 (A:0,C:0) Valid
KEY - vlan:53 mac:0xa0f849104867 l3_if:1 gpn:19 epoch:0 static:0 flood_en:0 vlan_lead_wless_flood_en: 0
client_home_asic: 0 learning_peerid 0, learning_peerid_valid 0
MASK - vlan:0 mac:0x0 l3_if:1 gpn:0 epoch:0 static:0 flood_en:0 vlan_lead_wless_flood_en: 0 client_home_asic: 0
learning_peerid 0, learning_peerid_valid 0
SRC_AD - need_to_learn:0 lrn_v:0 catchall:0 static_mac:0 chain_ptr_v:0 chain_ptr: 0 static_entry_v:0 auth_state:0
auth_mode:0 auth_behavior_tag:0 traf_m:0 is_src_ce:0
DST_AD - si:0x2d bridge:0 replicate:0 blk_fwd_o:0 v4_rmac:1 v6_rmac:1 catchall:0 ign_src_lrn:0 port_mask_o:0
afd_cli_f:0 afd_lbl:0 prio:3 dest_mod_idx:0 destined_to_us:1 pv_trunk:0 smr:0
```

Catalyst 9200, 9300 Stacking

- Stack connections are crucial to system stability
- 9200/9300 stackable switches deploy ring topology to form a stack
- One failure in stack recoverable. Dual Failure fatal
- Ensure proper fitting and tightening of stack connectors.



Dual Rings



Single Ring

Stack Port Overview

```
Switch#show switch stack-ports summary
Sw#/Port#   Port Status   Neighbor   Cable Length   Link OK   Link Active   Sync OK   #Changes to LinkOK   In Loopback
-----------------------------------------------------------------------------------------------------------------
1/1         OK            3          50cm           Yes       Yes           Yes       1                    No
1/2         OK            2          50cm           Yes       Yes           Yes       1                    No
2/1         OK            1          50cm           Yes       Yes           Yes       1                    No
2/2         OK            3          50cm           Yes       Yes           Yes       1                    No
3/1         OK            2          50cm           Yes       Yes           Yes       1                    No
3/2         OK            1          50cm           Yes       Yes           Yes       1                    No
```

Neighbor switch on port

Cable length should be showing

All stack links should be Active and in sync. Changes to LinkOK should be low

Stack numbers are NOT allocated based on physical presense. Renumber switches or bring up sequentially

Switch Stacks

```
Switch#show switch detail
Switch/Stack Mac Address : 58bf.eab6.7f80 - Local Mac Address
Mac persistency wait time: Indefinite
                                                     H/W   Current
Switch#   Role    Mac Address     Priority Version  State
-----------------------------------------------------------
 1       Member   58bf.eab6.4200     1       V01     Ready
*2       Active   58bf.eab6.7f80     4       V01     Ready
 3       Standby  58bf.eab7.5800     3       V01     Ready
          Stack Port Status             Neighbors
Switch#  Port 1     Port 2             Port 1   Port 2
-----------------------------------------------------------
  1        OK          OK                3        2
  2        OK          OK                1        3
  3        OK          OK                2        1
```

Every stack has 1 Active switch. Standby and members are optional

Set Priority to designate Active/Standby Switches with switch x priority x

SVI interfaces take the mac address of stack.
Layer 2 interfaces of the switch

Show tech fabric command prints out stack related information as well as stack related registers and counters

Consoles

- On switch stacks all console ports are active, redirection occurs to active switch
- Redirected session is a vty(telnet), console on Active is con (console)
- * indicates current active session in use, internal IPs are not in routing tabls

```
FE2050#show users
    Line       User       Host(s)              Idle       Location
*   0 con 0    cisco      idle                 00:00:00
    3 vty 2    cisco      idle                 00:01:30 192.168.1.5
FE2050#show tcp brief
TCB        Local Address               Foreign Address             (state)
7FF9CC3DEE20  192.168.1.6.23             192.168.1.5.43170           ESTAB
```

- Chassis based switches do not use console redirection, use “active” Supervisor
- Standby console can be enabled
- Functionality limited on standby

```
9407R#sh run | sec redundancy
redundancy
 mode sso
 main-cpu
  standby console enable
```

Unexpected Reloads

- When Switch (active/standby/member) unexpectedly reloads a system_report is generated in local crashinfo:
- Crashinfo file contains core file , traces, crashinfo files
- Crashinfo of member switches reachable via dir crashinfo-x:

```
guest@estg-bru-tftp:/tftpboot/cl$ tar -xzvf system-report_1_20190802-112928-UTC.tar.gz
/bootflash/.prst_sync/reload_info
/tmp/FE2050_1_RP_0-bootuplog-20190802-112928-UTC.log
/flash/core/FE2050_1_RP_0_x86_64_crb_linux_iosd_ngwc-universalk9-ms_9699_20190802-UTC.core.gz
/crashinfo/crashinfo_RP_00_00_20190802-112859-UTC
guest@estg-bru-tftp:/tftpboot/cl$ cat bootflash/.prst_sync/reload_info
ReloadReason=Critical software exception, check crashinfo:crashinfo_RP_00_00_20190802-112859-
UTC RET_2_RCALTS=1564745340 RET_2_RTS=11:29:00 UTC Fri Aug 2 2019
```

Crashinfo File

- Part of Crashinfo file contains:
  - Last interactions on CLI
  - Syslog messages

```
guest@estg-bru-tftp:/tftpboot/cl/crashinfo$ more crashinfo_RP_00_00_20190802-112859-UTC

CMD: 'sh device-tracking counters ' 11:28:53 UTC Fri Aug 2 2019
CMD: 'sh device-tracking counters vlan 1021' 11:28:57 UTC Fri Aug 2 2019
*Aug  2 11:28:57.445: %HA_EM-6-LOG: catchall: show device-tracking counters vlan 1021
CMD: 'sh device-tracking counters vlan 1022' 11:28:58 UTC Fri Aug 2 2019
*Aug  2 11:28:57.445: %HA_EM-6-LOG: catchall: show device-tracking counters vlan 1021
CMD: 'sh device-tracking counters vlan 1022' 11:28:58 UTC Fri Aug 2 2019
Exception to IOS Thread:
Frame pointer 0x7FD1C728CA28, PC = 0x562DFBC20D36
```

CSCvp72220 crash at sisf_show_counters after entering show device-tracking counters command

  - Memory Utilization
  - Buffer Pool Utilization
  - Uptime and IOS-XE version

- Looking through Crashinfo file can give clues about the state of the system prior to the unexpected reload

Packet Drops

Ethernet controller Statistics

```
9500_1#show controllers ethernet-controller te 1/0/1
Transmit                  TenGigabitEthernet1/0/1    Receive
  38633194582 Total bytes                130046767680 Total bytes
     92351261 Unicast frames                100227375 Unicast frames
  34062623315 Unicast bytes              116307995084 Unicast bytes
       689338 Multicast frames                2043062 Multicast frames
   4570570691 Multicast bytes             13738772084 Multicast bytes
            9 Broadcast frames                      8 Broadcast frames
          576 Broadcast bytes                     512 Broadcast bytes
            0 System FCS error frames               0 IpgViolation frames
            0 MacUnderrun frames                    0 MacOverrun frames

..
       2241744 65 to 127 byte frames          11667872 65 to 127 byte frames
      2838336 128 to 255 byte frames          3177806 128 to 255 byte frames
     80722296 256 to 511 byte frames          1593500 256 to 511 byte frames
```

Ethernet Statistics also in show tech

Error statistics from interface

```
Switch#show  interfaces gigabitEthernet 1/0/1 counters errors
Port        Align-Err     FCS-Err     Xmit-Err     Rcv-Err  UnderSize  OutDiscards
Gi1/0/1             0           0            0           0          0            0
Port      Single-Col   Multi-Col    Late-Col  Excess-Col  Carri-Sen        Runts
Gi1/0/1             0           0            0           0          0            0
```

Ethernet Interface Utilization

```
Switch#show interfaces | inc line|rate
Vlan1 is up, line protocol is up , Autostate Enabled
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet0/0 is administratively down, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet1/0/1 is up, line protocol is up (connected)
  5 minute input rate 103000 bits/sec, 174 packets/sec
  5 minute output rate 3879000 bits/sec, 324 packets/sec
GigabitEthernet1/0/2 is down, line protocol is down (notconnect)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bi
GigabitEthernet1/0/3 is down
  5 minute input rate 0 bits
  5 minute output rate 0 bi
```

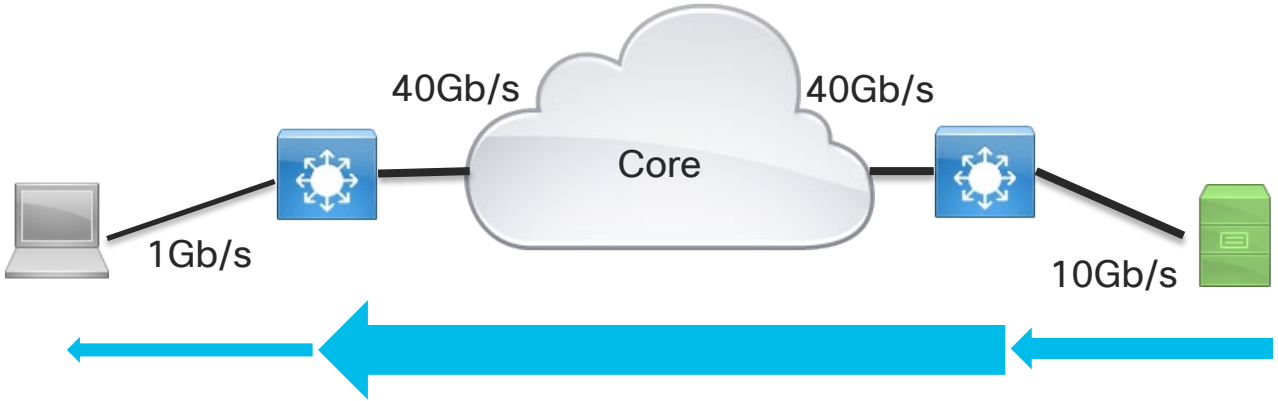
Average rate over interval set with load interval
Does not always reflect current load

```
9500 show controllers utilization
Port          Receive Utilization  Transmit Utilization
Te1/0/1                    3                  2
Te1/0/2                    2                  3
.
Te1/0/16                   0                  0
Fo1/1/1                    0                  0
Fo1/1/2                    0                  0
Total Ports : 18
Total Ports Receive Bandwidth Percentage Utilization  : 0
Total Ports Transmit Bandwidth Percentage Utilization : 0
Average Switch  Percentage Utilization : 0
```

Bandwidth in %
Current load

Drops due to egress buffering

- WTD drops due to buffering most common cause of drops
- Data center servers typically high bandwith 10Gb/s+
- Increasing number of buffers could lead to Jitter and Latency
- Protocols like TCP increase window size till congestion experienced



Tail Drops

- Tail drops occur when exceeding buffer thresholds.

```
9300_1#show interfaces gigabitEthernet 1/0/1 | inc output drops
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 1277
9300_1#show controllers ethernet-controller gig 1/0/1 | inc Excess Def
         1277 Excess Defer frames                    0 Collision fragments
SNMP:
SNMPv2-SMI::enterprises.9.2.2.1.1.27.8 = INTEGER: 1277
```

- Buffer allocation per class can be changed inside service-policy
- Global multiplier to increase buffers up to 1200%
  qos softmax-queue-multiplier <percentage>
- Increasing buffers increases maximum, buffers allocated based on availability
- 17.1.1 introduces CLI to monitor high watermark utilization on ports

QoS Hardware configuration

```
9300_1#sh plat hard fed switch active qos queue config interface gi 1/0/1
Asic:0 Core:1 DATA Port:0 GPN:1 LinkSpeed:0x1
-------------------------------------------------------------
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd
  ----- --------  --------  --------  --------  ---------
 0   1  5   200       3200   5   500   0     0   6  9600
 1   1  4     0  13  4800   5   750   2   300   6  9600
 Priority   Shaped/shared   weight  shaping_step  sharpedWeight
 --------   -------------   ------  ------------  -------------
 0       0     Shared            50             0            0
 1       0     Shared            75             0            0
 Port       Port            Port    Port
 Priority   Shaped/shared   weight  shaping_step
 --------   -------------   ------  ------------
        2     Shaped           254           255
   Weight0 Max_Th0 Min_Th0 Weigth1 Max_Th1 Min_Th1  Weight2 Max_Th2 Min_Th2
   ------- ------- ------- ------- ------- -------  ------- ------- ------
 0       0    2709                    3028                    3400       0
 1       0    3825       0       0    4275       0       0    4800       0
```

- Hardmax.
  Reserved buffers
- Softmax.
  From Global Pool

- Queue mode
- Queue limit:
  Step/weight * speed

- Drop thresholds
  per queue in buffers

QoS hardware statistics

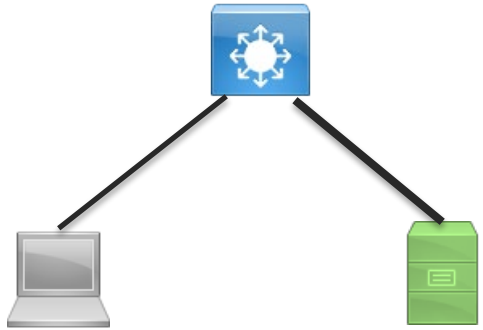
```
9300_1#sh platform hardware fed switch active qos queue stats interface gigabitEthernet 1/0/1
-------------------------------------------------------------------------------------------------
AQM Global counters
GlobalHardLimit:  7976   |   GlobalHardBufCount: 0
GlobalSoftLimit: 11872   |   GlobalSoftBufCount: 0
-------------------------------------------------------------------------------------------------
High Watermark Soft Buffers:      0 <--- clear on read
-------------------------------------------------------------------------------------------------
Asic:0 Core:1 DATA Port:0 Hardware Enqueue Counters
-------------------------------------------------------------------------------------------------
 Q Buffers          Enqueue-TH0          Enqueue-TH1          Enqueue-TH2              Qpolicer
   (Count)              (Bytes)              (Bytes)              (Bytes)              (Bytes)
-- ------- -------------------- -------------------- -------------------- --------------------
 0       0                    0               385820             46085690                    0
 1       0                    0                    0                    0                    0
Asic:0 Core:1 DATA Port:0 Hardware Drop Counters
-------------------------------------------------------------------------------------------------------------------------------------
 Q              Drop-TH0             Drop-TH1             Drop-TH2               SBufDrop              QebDrop         QpolicerDrop
                 (Bytes)              (Bytes)              (Bytes)                (Bytes)              (Bytes)              (Bytes)
-- -------------------- -------------------- -------------------- -------------------- -------------------- --------------------
 0                    0                    0                    0                    0                    0                    0
 1                    0                    0                    0                    0                    0                    0
```

set plat hard fed active qos port-monitor interface <if>

- At UADP level there are 8 Queues/3 Thresholds
- Enqueue/Drop Counters available per queue/per threshold
- Buffers (count) show currently assigned buffers to Queue (256 bytes)
- High water mark counter (17.x) shows highest watermark since last output

Buffer tuning , lab setup

- 10Gb/s Server, Debian 8.5
- 1Gb/s Client, Debian 8.5
- Interfaces on 9300 default
- Client and server use iperf3
- Both OS & iperf3 using default settings
- Achieved bandwidth , drops, high watermark & max Congestion window noted and



```
michelpe@DUT1:~$ iperf3 -v
iperf 3.1.3
Linux debian1 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2 (2016-04-08) x86_64
```

Buffer tuning , example

Cwnd = Congestion window

```
michelpe@DUT1:~$ iperf3 -c 10.200.10.100
Connecting to host 10.200.10.100, port 5201
[  4] local 10.200.10.200 port 37127 connected to 10.200.10.100 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   113 MBytes   951 Mbits/sec    6    167 KBytes
[  4]   1.00-2.00   sec   112 MBytes   941 Mbits/sec    0    198 KBytes
[  4]   2.00-3.00   sec   112 MBytes   941 Mbits/sec    0    225 KBytes
[  4]   3.00-4.00   sec   112 MBytes   941 Mbits/sec    0    247 KBytes
[  4]   4.00-5.00   sec   112 MBytes   941 Mbits/sec    1    195 KBytes
[  4]   5.00-6.00   sec   112 MBytes   942 Mbits/sec    0    229 KBytes
[  4]   6.00-7.00   sec   112 MBytes   941 Mbits/sec    0    250 KBytes
[  4]   7.00-8.00   sec   112 MBytes   941 Mbits/sec    0    262 KBytes
[  4]   8.00-9.00   sec   112 MBytes   941 Mbits/sec    1    208 KBytes
[  4]   9.00-10.00  sec   112 MBytes   941 Mbits/sec    0    233 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  1.10 GBytes   942 Mbits/sec    8             sender
[  4]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec                  receiver
```

Cwnd increases till drops occur

Test Results

	No Service Policy					AutoQos Policy				
	Default	200	400	800	1200	Default	200	400	800	1200
Cwnd	273	421	990	1840	2003	117	233	421	762	1200
Drops	18	22	19	50	47	23	20	12	20	38
Speed	942	944	945	945	945	933	938	938	940	941
Buffers	1200	2400	4800	9600	14400	500	1000	2000	4000	6000
Water mark	1197	2394	4795	9499	10171	497	994	1195	3997	5999

- Increasing number of buffers increases max achieved congestion window
- Drops happen in burst, most drops happen at initial ramp up of Cwnd

Packet Parting packet drops

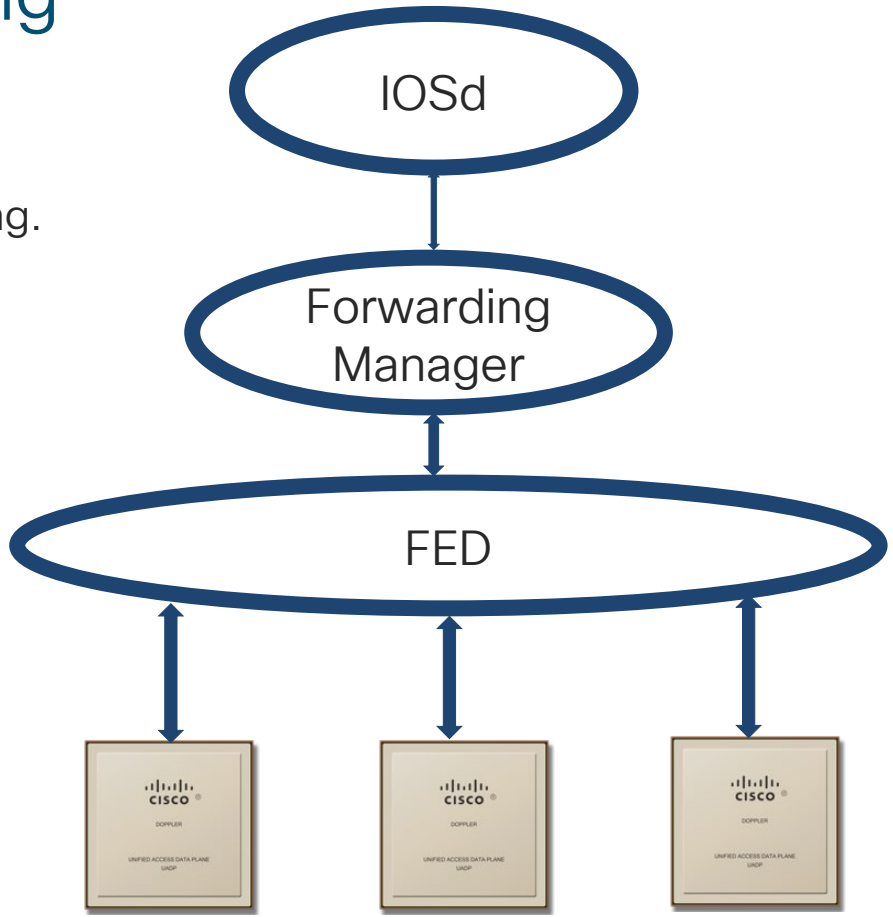
```
9300_1#sh platform hardware fed switch active fwd-asic drops exceptions
****EXCEPTION STATS ASIC INSTANCE 0 (asic/core 0/0)****
=================================================================================
 Asic/core |                  NAME                    |   prev   |  current  |  delta
=================================================================================
0   0   NO_EXCEPTION                                   35364016   35364108    92
0   0   IPV4_CHECKSUM_ERROR                           0          0           0
0   0   ROUTED_AND_IP_OPTIONS_EXCEPTION               2          2           0
0   0   CTS_FILTERED_EXCEPTION                        0          0           0
0   0   SIA_TTL_ZERO                                  0          0           0
0   0   ALLOW_NATIVE_EXCEPTION_COUNT                  0          0           0
0   0   ALLOW_DOT1Q_EXCEPTION_COUNT                   0          0           0
0   0   ALLOW_PRIORITY_TAGGED_EXCEPTION_COUNT         0          0           0
0   0   ALLOW_UNKNOWN_ETHER_TYPE_EXCEPTION            0          0           0
0   0   IP_SOURCE_GUARD_VIOLATION                     0          0           0
```

- Every packet passing through Port Asic gets parsed,
  Layer 2 switched packets also get fully parsed
- Exception drops are counted per Asic , no per port statistics

Forwarding Issues

Troubleshooting Forwarding

- UADP responsible for all hardware forwarding. CPU is not directly involved in forwarding.
- IOS-XE 16+ uses Forwarding manager, Forwarding manager process manages forwarding related information but only on a Platform Independent level. Forwarding manager is on all platforms running IOS-XE16+
- FED (Forwarding Engine Driver) process is Platform Dependant layer, interfaces with Forwarding manager and responsible for all tables on UADP asics



Forwarding troubleshooting, the easy way

- Show forward supported since 2900/3500XL switches , upto 3750 family only software emulation of forwarding results were used
- UADP introduced HW captures of lookup results during various stages of packet forwarding
- CLI: "*Show platform hardware fed switch <ingress switch> forward … "*
- Supports Input using packet capture file or packet parameters
- Readability of results greatly improved since 16.9 with summary option
- Packets introduced for captures inserted at Ingress and removed at Egress before sending out on the wire.

Running Show platform hardware fed forward

```
9300_1#sh monitor capture file flash:icmp.pcap packet-number 11
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
  11   5.006009 10.100.10.100 b^F^R 10.200.10.200 ICMP 98 Echo (ping) request  id=0x262f
```

Verify capture

- Using the packet capture the show forward can be executed
- Always execute it on the switch where the packet ingresses

```
9300_1#sh plat hard fed 1 forward int gi 1/0/1 pcap flash:icmp.pcap num 11 data
Show forward is running in the background. After completion, syslog will be generated.
```

- Once completed a syslog gets generated and results will be available
- Can only run one show forward at a time

```
*Jan 27 10:07:35.009: %SHFWD-6-PACKET_TRACE_DONE: Switch 1 R0/0: fed: Packet Trace
Complete:  Execute (show platform hardware fed switch <> forward last summary|detail)
```

Show platform forward results

```
9300_1#sh platform hardware fed switch active forward last summary
Input Packet Details:
###[ Ethernet ]###
  dst       = a0:f8:49:10:48:51
  src       = 00:0c:29:4d:9e:16
  type      = 0x800
###[ IP ]###
     version   = 4L
     ihl       = 5L
     tos       = 0x0
     len       = 84
     id        = 46165
     flags     = DF
     frag      = 0L
     ttl       = 64
     proto     = icmp
     chksum    = 0x5bfc
     src       = 10.100.10.100
     dst       = 10.200.10.200
     options   = ''
```

First section shows the exact packet being send for analysis

Show platform forward results

```
Ingress:
   Port                        : GigabitEthernet1/0/1
   Global Port Number          : 1
   Local Port Number           : 1
   Asic Port Number            : 0
   Asic Instance               : 1
   Vlan                        : 100
   Mapped Vlan ID              : 6
   STP Instance                : 4
   BlockForward                : 0
   BlockLearn                  : 0
   L3 Interface                : 50
       IPv4 Routing            : enabled
       IPv6 Routing            : enabled
       Vrf Id                  : 0
   Adjacency:
       Station Index           : 185
       Destination Index       : 21358
       Rewrite Index           : 34
       Replication Bit Map     : 0x10    ['coreData']
```

Next section shows associated data

Ingress port

Vlan and STP info

IPv4/v6 setting.
Vrf-id 0 = GRT

Show platform forward results

```
Adjacency:
      Station Index          : 185
      Destination Index      : 21358
      Rewrite Index          : 34
      Replication Bit Map    : 0x10    ['coreData']
  Decision:
      Destination Index      : 21358
      Rewrite Index          : 34
      Dest Mod Index         : 0       [IGR_FIXED_DMI_NULL_VALUE]
      CPU Map Index          : 0       [CMI_NULL]
      Forwarding Mode        : 3       [Other or Tunnel]
      Replication Bit Map    :         ['coreData']
      Winner                 :         L3FWDIPV4 LOOKUP
      Qos Label              : 1
      SGT                    : 0
      DGTID                  : 0
```

Forwarding Decision

Destination Index, where to send

Rewrite Index, how to rewrite

Forwarding mode, how to treat the packet

Show platform forward results

```
Egress:
   Possible Replication      :
       Port                  : TenGigabitEthernet1/1/7
   Output Port Data          :
     Port                    : TenGigabitEthernet1/1/7
       Global Port Number    : 59
       Local Port Number     : 59
       Asic Port Number      : 58
       Asic Instance         : 0
       Unique RI             : 34
       Rewrite Type          : 9       [L3_UNICAST_IPV4]
       Mapped Rewrite Type   : 9       [L3_UNICAST_IPV4]
       Vlan                  : 0
       Mapped Vlan ID        : 0
```

Last section showing Egress Packet

```
Output Packet Details:
   Port                         : TenGigabitEthernet1/1/7
###[ Ethernet ]###
  dst       = 00:50:56:92:48:d8
  src       = a0:f8:49:10:48:66
  type      = 0x800
###[ IP ]###
     version   = 4L
     ihl       = 5L
 tos       = 0x0
     len       = 84
     id        = 46165
     flags     = DF
     frag      = 0L
     ttl       = 63
     proto     = icmp
     chksum    = 0x5cfc
     src       = 10.100.10.100
     dst       = 10.200.10.200
     options   = ''
```

Packet removed before sending out to the wire.

Show platform hardware fed forward detail results

- Flash contains shfwd<>.log with detailed information
- Same information gathered with
  show platform hardware fed switch <switch> forward last detail
- Detail info containing raw information regarding lookups

```
9300_1#sh platform hardware fed switch 1 forward last detail
-----------------------------------------------
Starting IPP capture
============================
IppDefaultClientTable[4]
----------------------------------
  defaultClientLeIndex          1
==================================
IppClientLeAd[1]
------------------------------------------------------------
                     LEAD_CLIENT_ANCHORED          0
            LEAD_CLIENT_AUTH_BEHAVIOR_TAG          0
                 LEAD_CLIENT_CLIENT_GROUP          0
```

Layer 2 Forwarding. Verifying STP state

- Show spanning tree gives IOSd view of Spanning Tree
- Hardware forwarding states can be checked *per switch* on FED layer
- Outputs will show what interface are in forwarding state and if traffic will be tagged or untagged
- Flood list indicates what Ports will receive flooded traffic on this switch

```
9300_1#show platform hardware fed switch 1 vlan 100 egress
VLAN STP State in hardware
vlan id is:: 101
Interfaces in forwarding state: : Te1/1/7(Untagged), Gi1/0/1(Untagged)
       show platform hardware fed switch 1 vlan 100 ingress
VLAN STP State in hardware
vlan id is:: 101
Interfaces in forwarding state: : Te1/1/7(Untagged), Gi1/0/1(Untagged)
flood list: : Te1/1/7, Gi1/0/1
```

Layer 2 Forwarding, IOSd mac address tables

```
9300_1#sh mac address-table vlan 100
          Mac Address Table
-------------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 100    000c.294d.9e16    DYNAMIC     Gi1/0/1
 100    0050.5692.adb3    DYNAMIC     Gi1/0/1
 100    0050.5692.e9aa    DYNAMIC     Gi1/0/1
 100    a0f8.4910.4851    STATIC      Vl100
Total Mac Addresses for this criterion: 4
```

- Show mac address table contains a system wide mac table on IOSd
- Types can be static, dynamic, drop
- Mac Address of SVI interfaces also showing in mac address table

FED MATM Mac Address Table

```
9300_1#sh platform software fed switch 1 matm macTable vlan 100
VLAN   MAC                  Type  Seq#   EC_Bi  Flags  machandle          siHandle        diHandle        *a_time  *e_time  ports
-------------------------------------------------------------------------------------------------------------------------------
100    a0f8.4910.4851      0x8002     0  99817     64  0x7f91986dfcd8     0x7f9198dadb78    0x0              0        0    Vlan100
100    0050.5692.e9aa         0x1   347      0      0  0x7f9199054668     0x7f9199020798    0x7f91986e4a58   300      234  Gi1/0/1
100    0050.5692.adb3         0x1   352      0      0  0x7f91990144a8     0x7f9199020798    0x7f91986e4a58   300      71   Gi1/0/1
100    000c.294d.9e16         0x1   364      0      0  0x7f919900e9d8     0x7f9199070018    0x7f91986e4a58   300      290  Gi1/0/1

*a_time=aging_time(secs)  *e_time=total_elapsed_time(secs)
Type:
MAT_DYNAMIC_ADDR           0x1  MAT_STATIC_ADDR            0x2  MAT_CPU_ADDR              0x4  MAT_DISCARD_ADDR          0x8
MAT_ALL_VLANS             0x10  MAT_NO_FORWARD            0x20  MAT_IPMULT_ADDR           0x40  MAT_RESYNC               0x80
MAT_DO_NOT_AGE           0x100  MAT_SECURE_ADDR          0x200  MAT_NO_PORT              0x400  MAT_DROP_ADDR            0x800
MAT_DUP_ADDR            0x1000  MAT_NULL_DESTINATION    0x2000  MAT_DOT1X_ADDR          0x4000  MAT_ROUTER_ADDR         0x8000
MAT_WIRELESS_ADDR      0x10000  MAT_SECURE_CFG_ADDR     0x20000  MAT_OPQ_DATA_PRESENT   0x40000  MAT_WIRED_TUNNEL_ADDR  0x80000
MAT_DLR_ADDR          0x100000  MAT_MRP_ADDR           0x200000  MAT_MSRP_ADDR         0x400000  MAT_LISP_LOCAL_ADDR    0x800000
MAT_LISP_REMOTE_ADDR 0x1000000  MAT_VPLS_ADDR         0x2000000
```

- Every FED has its own Mac address table.
- Type Field indicates the type of mac address using a bitmap
- Sequence number of an entry changing would indicated relearning

Layer 3 Forwarding. Routing protocols

```
9300_1#ping 10.48.91.151
Sending 5, 100-byte ICMP Echos to 10.48.91.151, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
9300_1#sh ip route 10.48.91.151
Routing entry for 10.48.91.128/25
  Known via "isis", distance 115, metric 40, type level-2
  Redistributing via isis
  Last update from 172.31.250.30 on TenGigabitEthernet1/1/6, 6d22h ago
  Routing Descriptor Blocks:
  * 172.31.250.30, from 172.31.255.21, 6d22h ago, via TenGigabitEthernet1/1/6
      Route metric is 40, traffic share count is 1
9300_1#sh ip arp 172.31.250.30
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.31.250.30           56  2c4f.523b.c142  ARPA   TenGigabitEthernet1/1/6
```

- Check Routing Table for correct next hop (Routing Information Base)
- ARP table display rewrite information for next hop (destination mac)

Cisco Express Forwarding (the FIB)

```
9300_1#sh ip cef 10.48.91.128/25 internal
10.48.91.128/25, epoch 7, RIB[I], refcnt 6, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 4th priority
  ifnums:
    TenGigabitEthernet1/1/6(85): 172.31.250.30
  path list 7F3B3265DE78, 139 locks, per-destination, flags 0x4D [shble, hvsh, rif, hwcn]
    path 7F3B32181A60, share 1/1, type attached nexthop, for IPv4
      nexthop 172.31.250.30 TenGigabitEthernet1/1/6, IP adj out of TenGigabitEthernet1/1/6,
      addr 172.31.250.30 7F3B33B02738
  output chain:
    IP adj out of TenGigabitEthernet1/1/6, addr 172.31.250.30 7F3B33B02738
9300_1#sh adjacency 172.31.250.30 detail
Protocol Interface                 Address
IP       TenGigabitEthernet1/1/6   172.31.250.30(89)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 11
                                   Encap length 14
                                   2C4F523BC142A0F8491048500800
                                   L2 destination address byte offset 0
                                   L2 destination address byte length 6
                                   Link-type after encap: ip
                                   ARP
```

Source of route, RIB

Next hop

Adjacency -> rewrite info

Platform CEF tables (RP)

```
9300_1#sh platform software ip switch ac R0 cef prefix 10.48.91.128/25
Forwarding Table
Prefix/Len                          Next Object      Index
-------------------------------------------------------------------
10.48.91.128/25                     OBJ_ADJACENCY    0x24
9300_1#sh platform software adjacency switch active R0 index 0x24
Number of adjacency objects: 17
Adjacency id: 0x24 (36)
  Interface: TenGigabitEthernet1/1/6, IF index: 85, Link Type: MCP_LINK_IP
  Encap: 2c:4f:52:3b:c1:42:a0:f8:49:10:48:50:8:0
  Encap Length: 14, Encap Type: MCP_ET_ARPA, MTU: 9100
  Flags: no-l3-inject
  Incomplete behavior type: None
  Fixup: unknown
  Fixup_Flags_2: unknown
  Nexthop addr: 172.31.250.30
  IP FRR MCP_ADJ_IPFRR_NONE 0
  OM handle: 0x348066bc48
```

Prefix points to Adjacency 0x24

Similar output should be present on standby RP and the FP processes

FED Routing tables

```
9300_1#sh platform software fed switch 1 ip route 10.48.91.128/25
vrf    dest              htm              flags    SGT    DGID MPLS Last-modified
---    ----              ---              -----    ---    ---- ---- ----------------------
0      10.48.91.128/25   0x7f9199010f78 0x0      0      0          2020/01/14 13:49:42.054
  FIB: prefix_hdl:0x5a00026d, mpls_ecr_prefix_hdl:0
  ========== OCE chain =====
  ADJ:objid:36 {link_type:IP ifnum:0x55, si:0x9b00003d, IPv4:   172.31.250.30 }
  ===============
  MPLS info: mpls_ecr_scale_prefix_adj:0, mpls_lspa_hdl:0
9300_1#sh platform software fed switch 1 ip adj | inc dest|--|172.31.250.30
dest             if_name     dst_mac         si_hdl   ri_hdl   pd_flags adj_id  Last-modified
----             -------     -------         ------   ------   -------- -----   ----------------------
172.31.250.30    Te1/1/6     2c4f.523b.c142  0x7f9198 0x7f9198 0x0       0x24    14:09:12.058
```

- FED layer has its own copy of the IP routing table and rewrite information
- In a stacked enviroment every switch has its own FED process. Important to check Ingress and Egress switch
- FED programs TCAM to facilitate forwarding
- Every VRF has its own unique number, 0 is Global Routing Table

Agenda

- Catalyst 3k/9k Platform & Software Architecture
- Troubleshooting tools
- Platform Issues
- Packet Drops
- Forwarding Issues
- Questions

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

cisco Live!

Continue your education

Demos in the Cisco campus

Walk-in self-paced labs

Meet the engineer 1:1 meetings

Related sessions

Thank you

You make **possible**