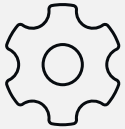
Packet journey inside Catalyst 9000 switches

Ivan Shirshin - Technical Leader, Cisco TAC
Nathan Pan - Technical Leader, Cisco TAC
BRKARC-3090

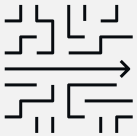
Session goals:

Level **Advanced**

Architecture Details
Catalyst 9000 platform architecture



ASICs insights
UADP vs S1
Forwarding Engines



Troubleshooting & Verification
How to verify forwarding decision ?

Knowledge & Understanding

Toolbox

Webex App

Questions?

Use the Webex app to chat with the speaker after the session

How

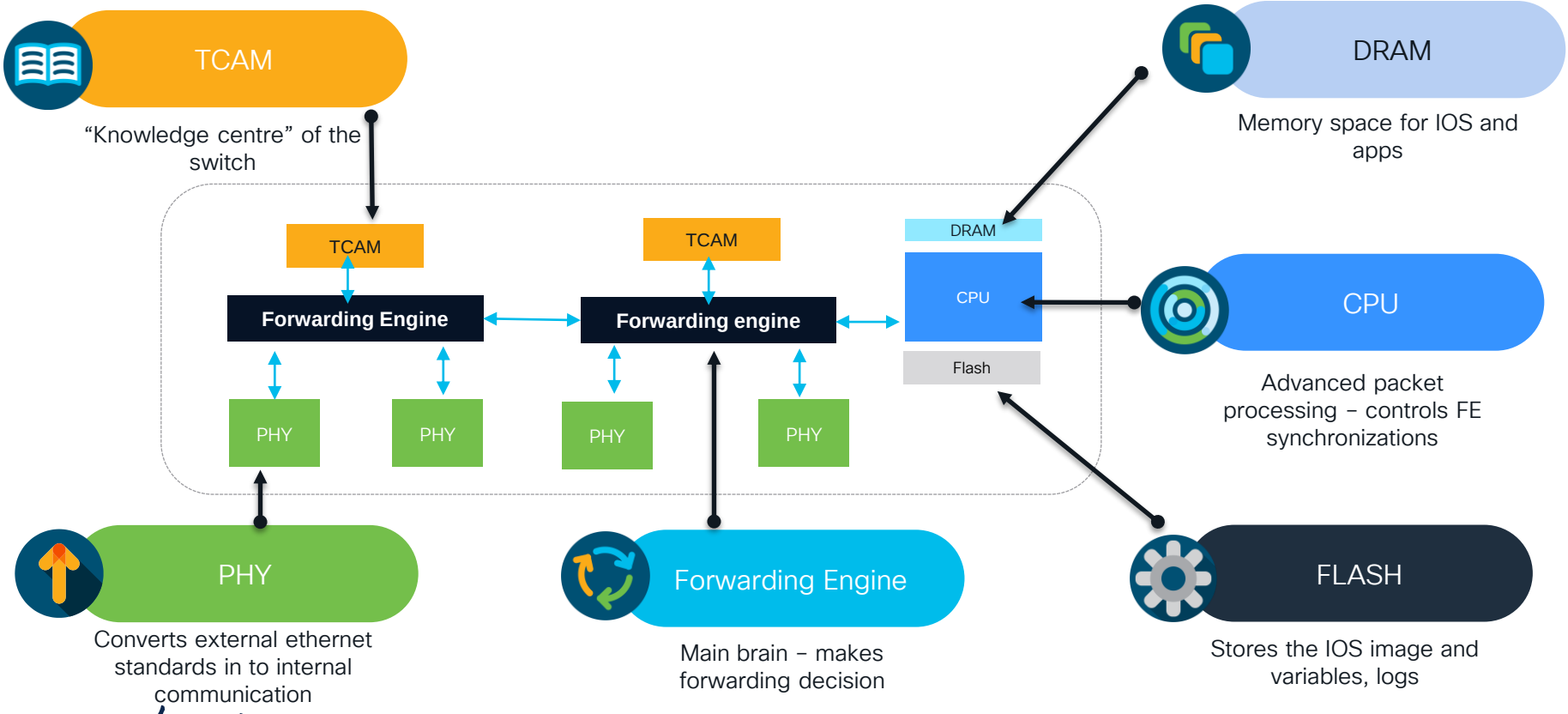
1. Find this session in the Cisco Events mobile app
2. Click “Join the Discussion”
3. Install the Webex app or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 28, 2025.

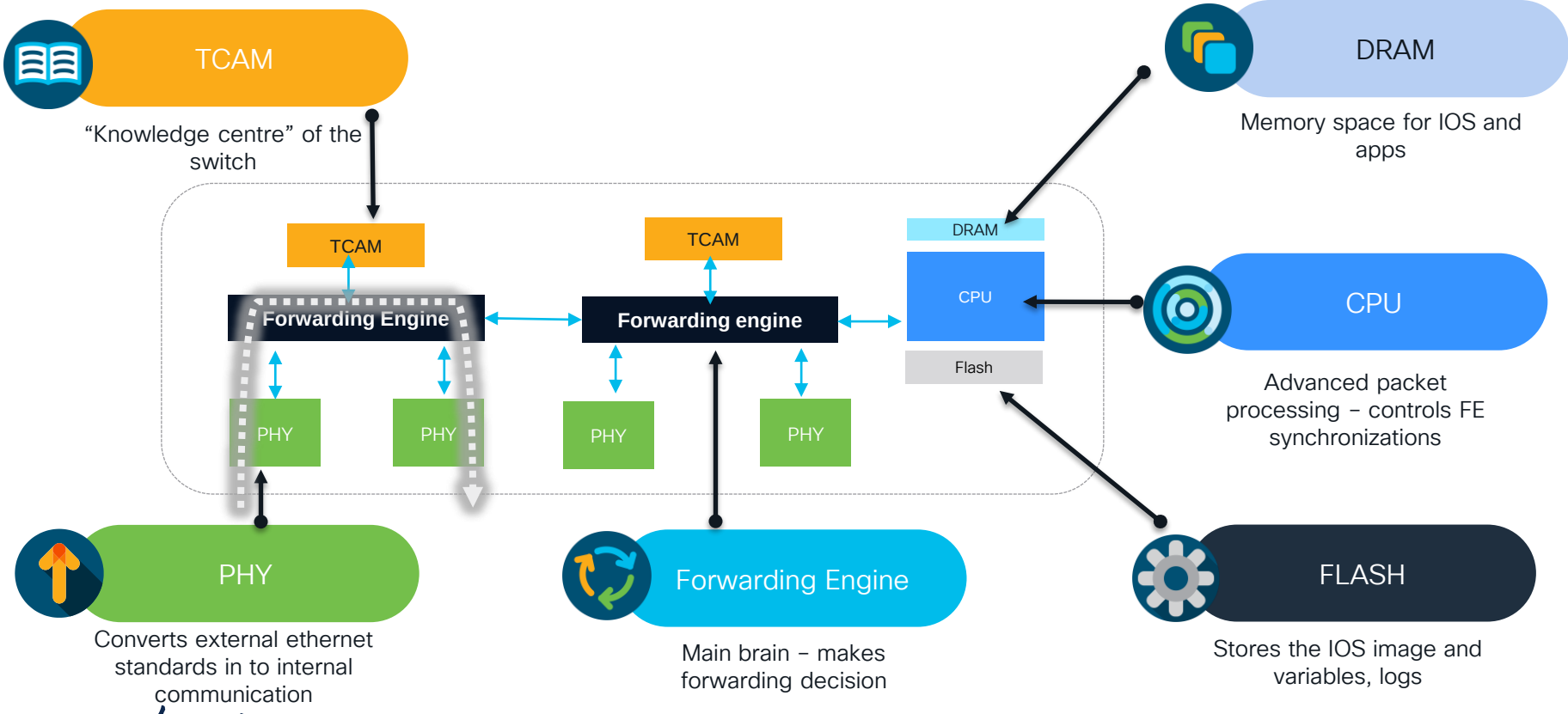
Agenda

- Introduction & Catalyst 9000 Family Overview
- ASICs at the Core: UADP vs SiliconOne
- Data Plane Forwarding Scenarios
- Control Plane Packet Path
- Forwarding Decision Validation
- Wrap-Up

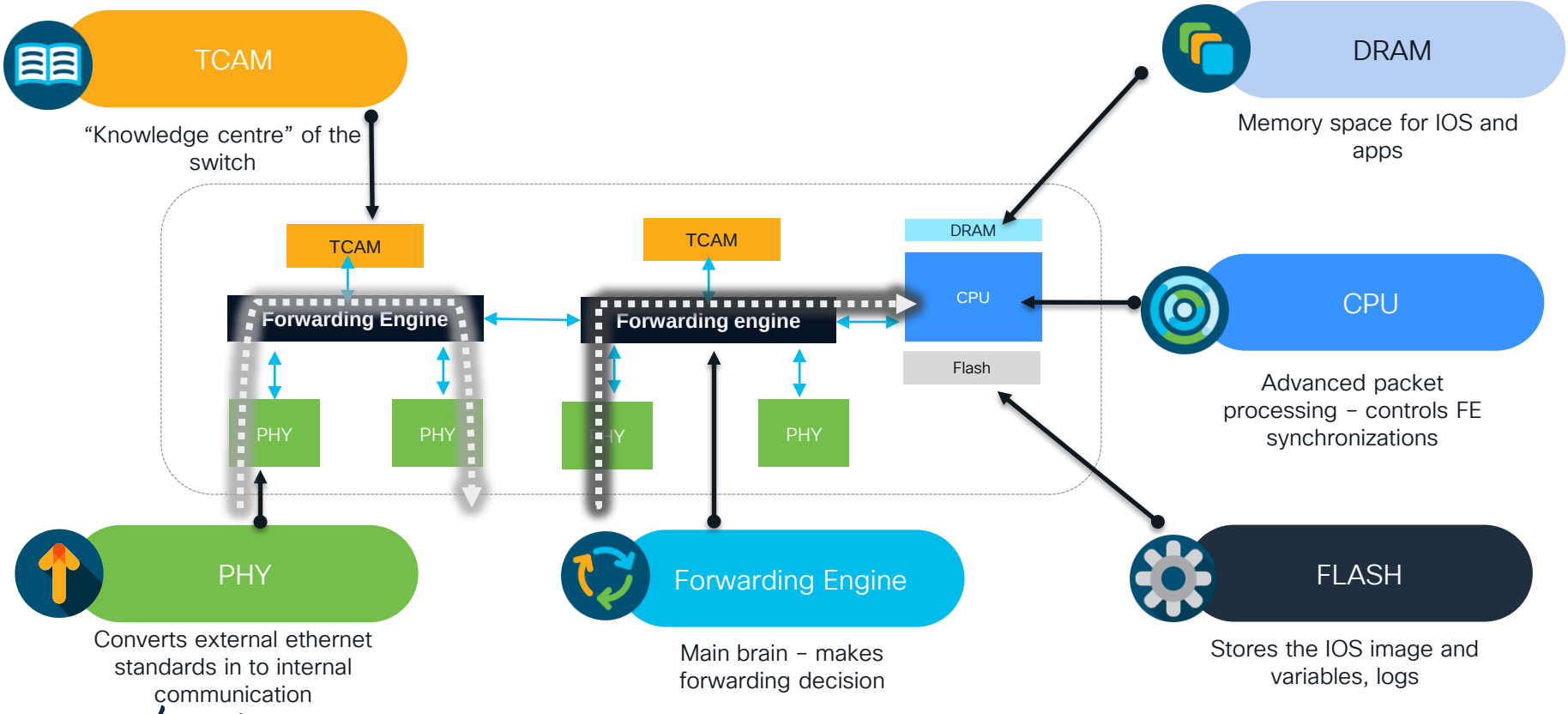
General Switch Architecture – Moon view



General Switch Architecture – Moon view

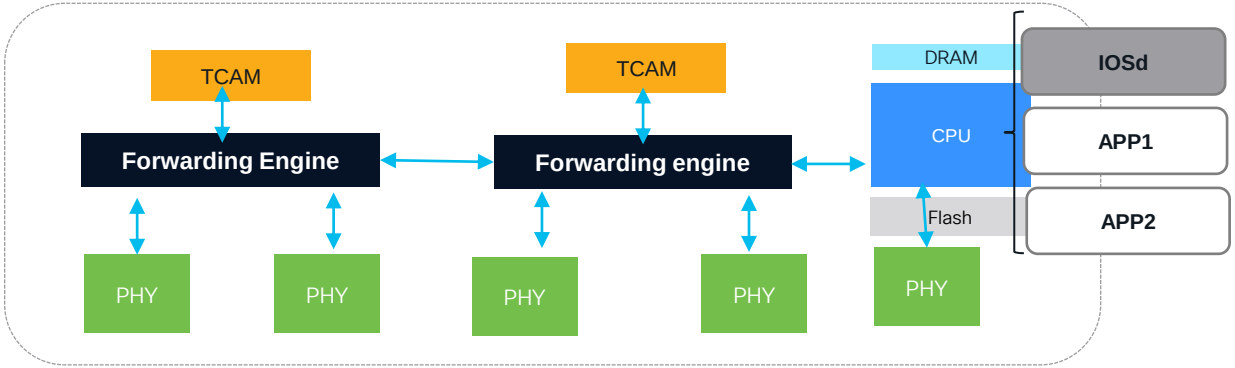


General Switch Architecture – Moon view



Data plane & control plane – is that all ?

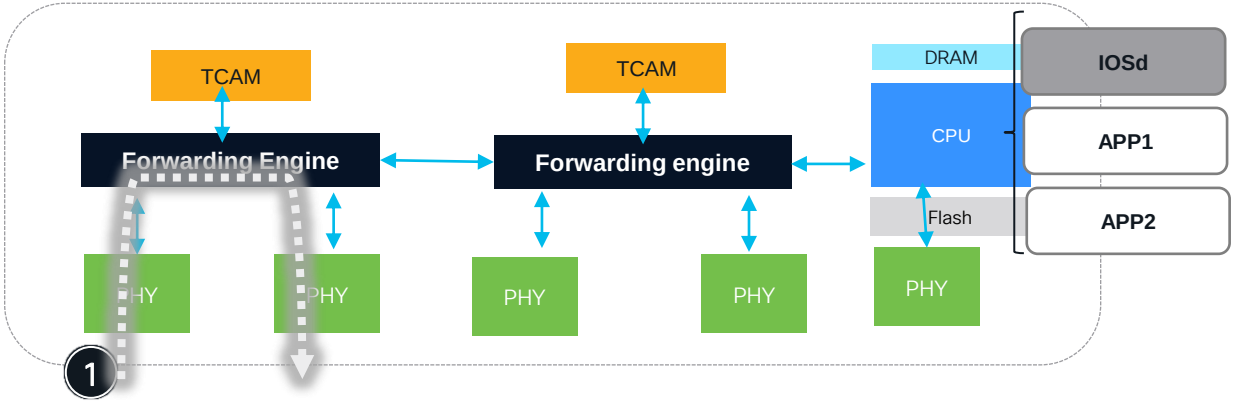
Forwarding Playbook



Data plane & control plane – is that all ?

1 **Inner single FE/single core forwarding**

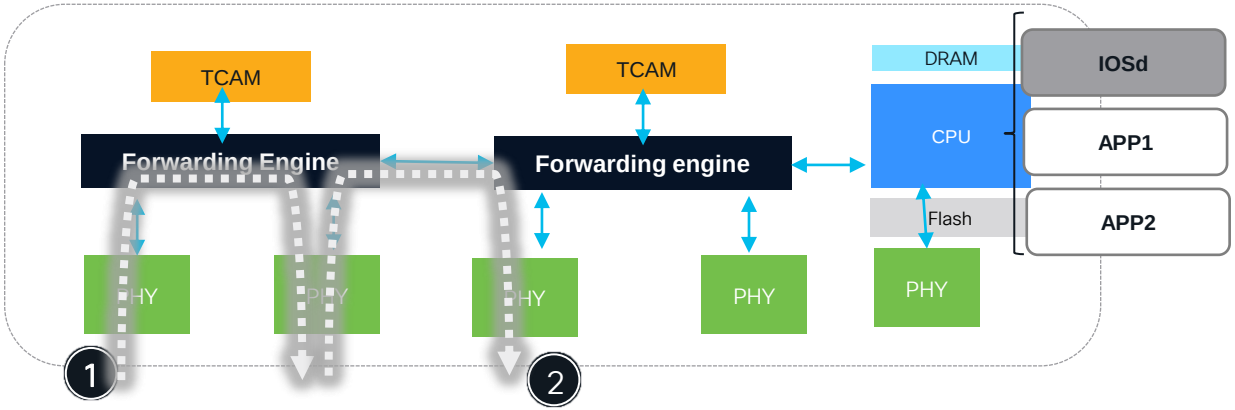
Forwarding Playbook



Data plane & control plane – is that all ?

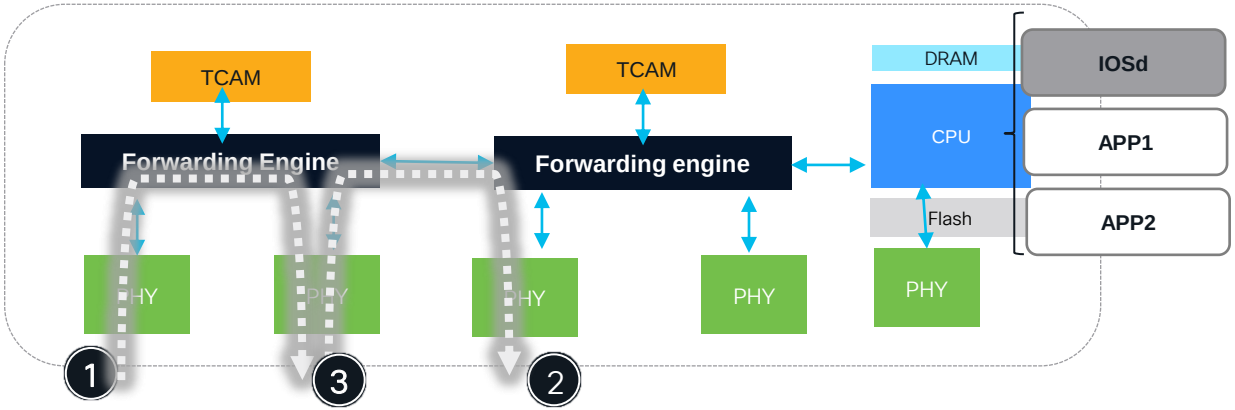
Forwarding Playbook

1. **Inner single FE/single core forwarding**
2. **Intra FE/ multiple core forwarding**



Data plane & control plane – is that all ?

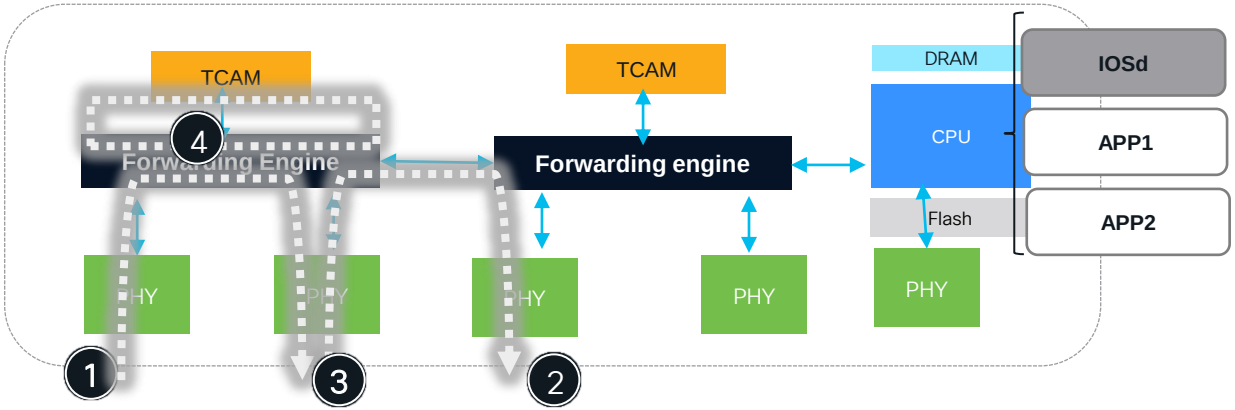
Forwarding Playbook



1. **Inner single FE/single core forwarding**
2. **Intra FE/ multiple core forwarding**
3. **Intra FE across different stack members**

Data plane & control plane – is that all ?

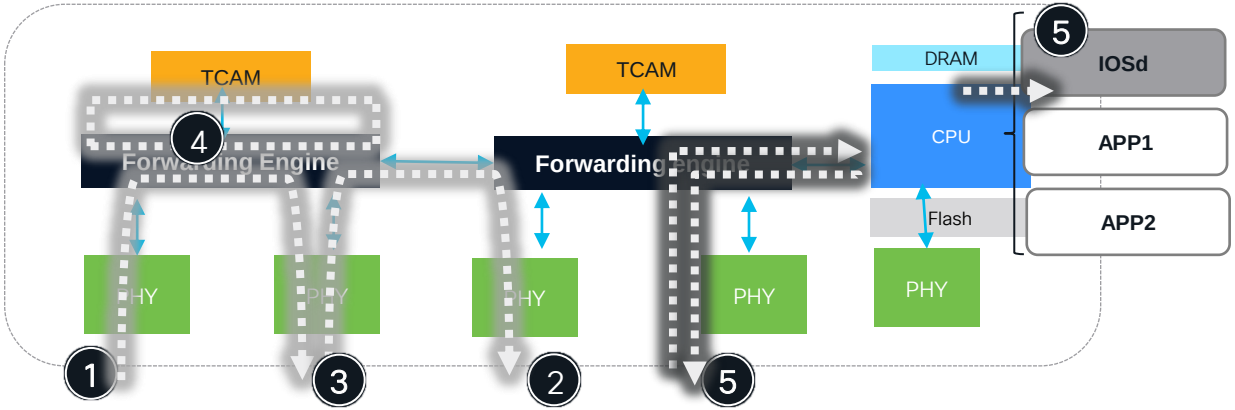
Forwarding Playbook



1. Inner single FE/single core forwarding
2. Intra FE/ multiple core forwarding
3. Intra FE across different stack members
4. Recirculation within FE

Data plane & control plane – is that all ?

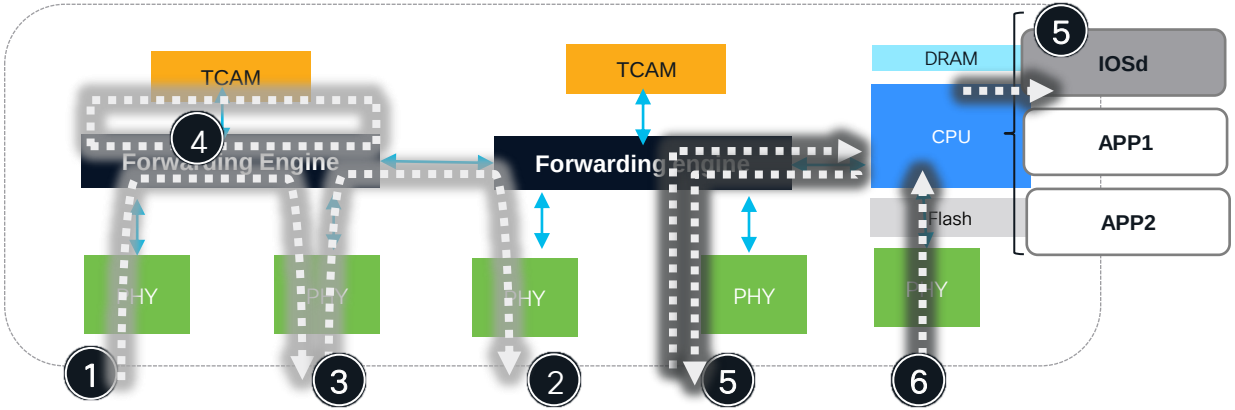
Forwarding Playbook



1. Inner single FE/single core forwarding
2. Intra FE/ multiple core forwarding
3. Intra FE across different stack members
4. Recirculation within FE
5. IOS Control-plane forwarding

Data plane & control plane – is that all ?

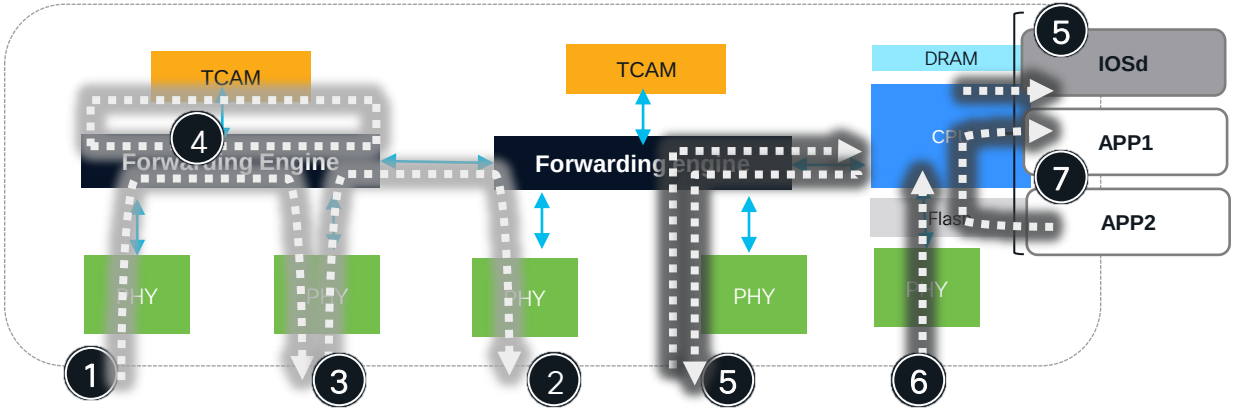
Forwarding Playbook



1. Inner single FE/single core forwarding
2. Intra FE/ multiple core forwarding
3. Intra FE across different stack members
4. Recirculation within FE
5. IOS Control-plane forwarding
6. Mgmt port forwarding

Data plane & control plane – is that all ?

Forwarding Playbook



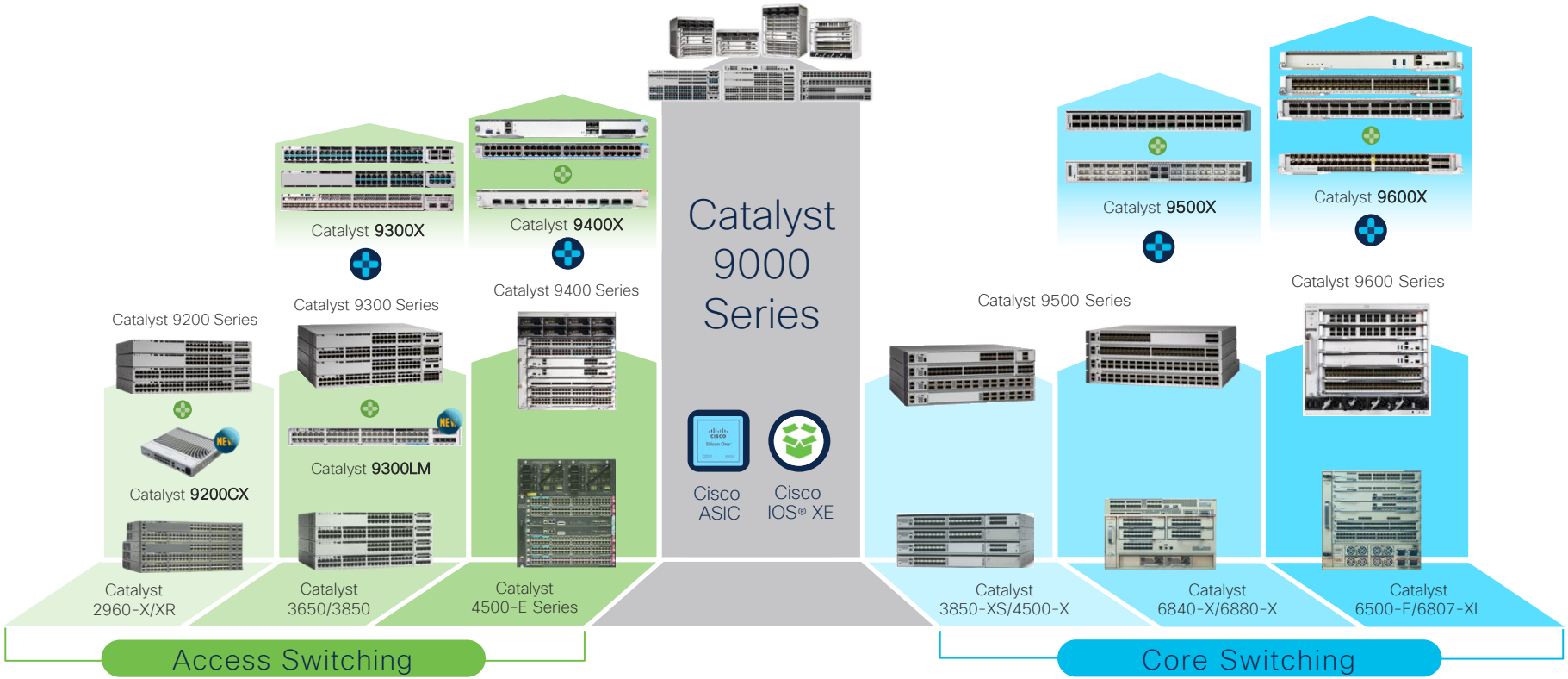
1. Inner single FE/single core forwarding
2. Intra FE/ multiple core forwarding
3. Intra FE across different stack members
4. Recirculation within FE
5. IOS Control-plane forwarding
6. Mgmt port forwarding
7. App-hosting forwarding

Intro & Catalyst 9000 Family Overview

cisco Live!

Cisco Catalyst 9000 Switching Portfolio

One Family from Access to Core – Common Hardware & Software



Catalyst 9000 – Forwarding Engines (aka ASICs)

Cisco **Unified Access Data-Plane** (UADP)

Cisco **Silicon One**™

Flexible Pipelines
Investment Protection

Adaptable Tables
Universal Deployment

Scalable Resources
Enhanced Scale and Buffering

Flexible & Programmable ASICs – Adapt to New Technologies

Catalyst 9000 with UADP

Flexible ASIC Evolution

UADP

Catalyst 9200/L/C
Catalyst 9300/X/L
Catalyst 9400/X
Catalyst 9500/H
Catalyst 9600 Sup1

UADP 2.0 m

Catalyst 9200CX

UADP 2.0 sec

Catalyst 9300X/LM

UADP 3.0 sec

Catalyst 9400X-SUP2

UADP 2.0 m

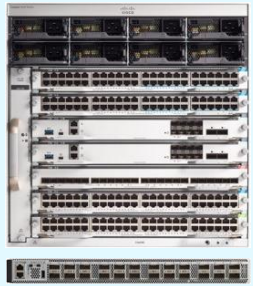
Catalyst 9200/L

UADP 2.0

Catalyst 9300/L

UADP 2.0XL

Catalyst 9400-SUP1 / 9500



UADP 3.0

Catalyst 9500(H) / 9600-SUP1

Catalyst 9000 with S1

Introducing the next-generation of ASICs

Silicon One

Catalyst **9500X**

Catalyst **9600X (Sup2)**

Q200

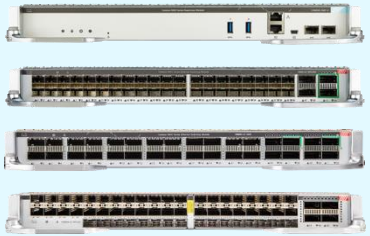
Catalyst 9500X-28C8D

Q200

Catalyst 9500X-60L4D

Q200

Catalyst 9600X-SUP2



ASICs at the Core: UADP vs SiliconOne

cisco Live!

UADP – one architecture multiple cores

Catalyst 9200CX

UADP 2.0mini

Catalyst 9300L-48

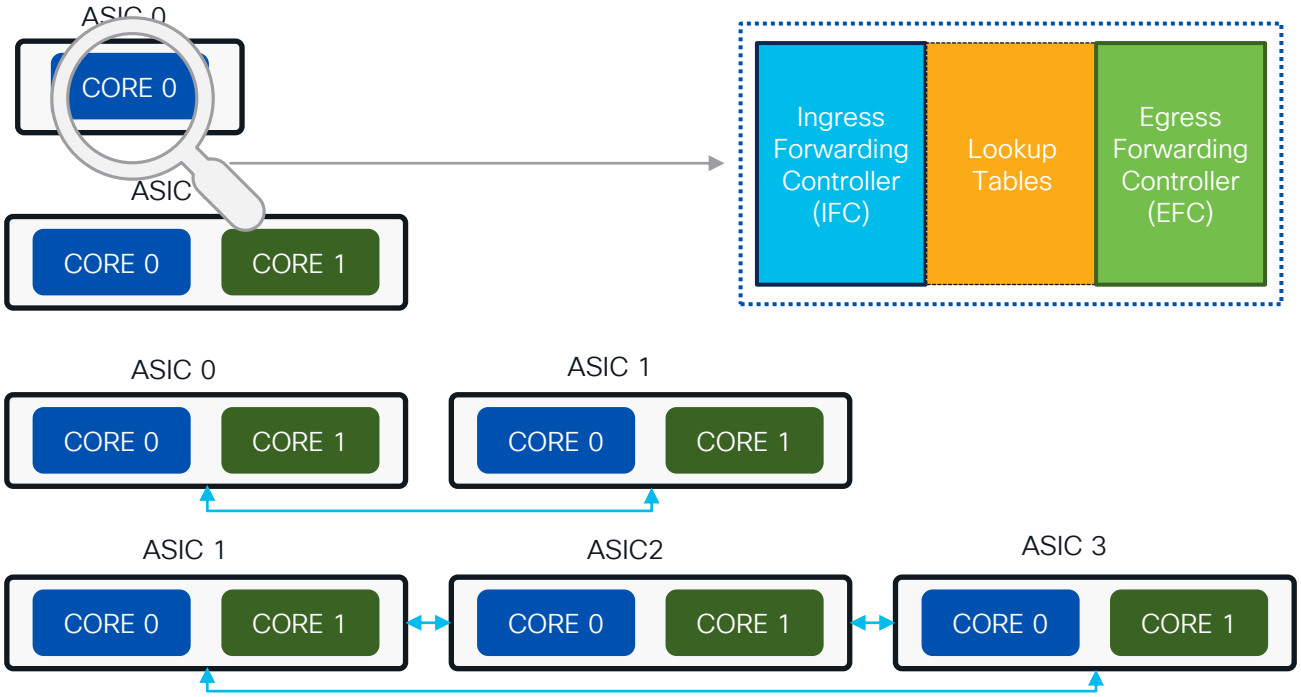
UADP 2.0

Catalyst 9300X

UADP 2.0 sec

Catalyst 9600-Sup1

UADP 3.0

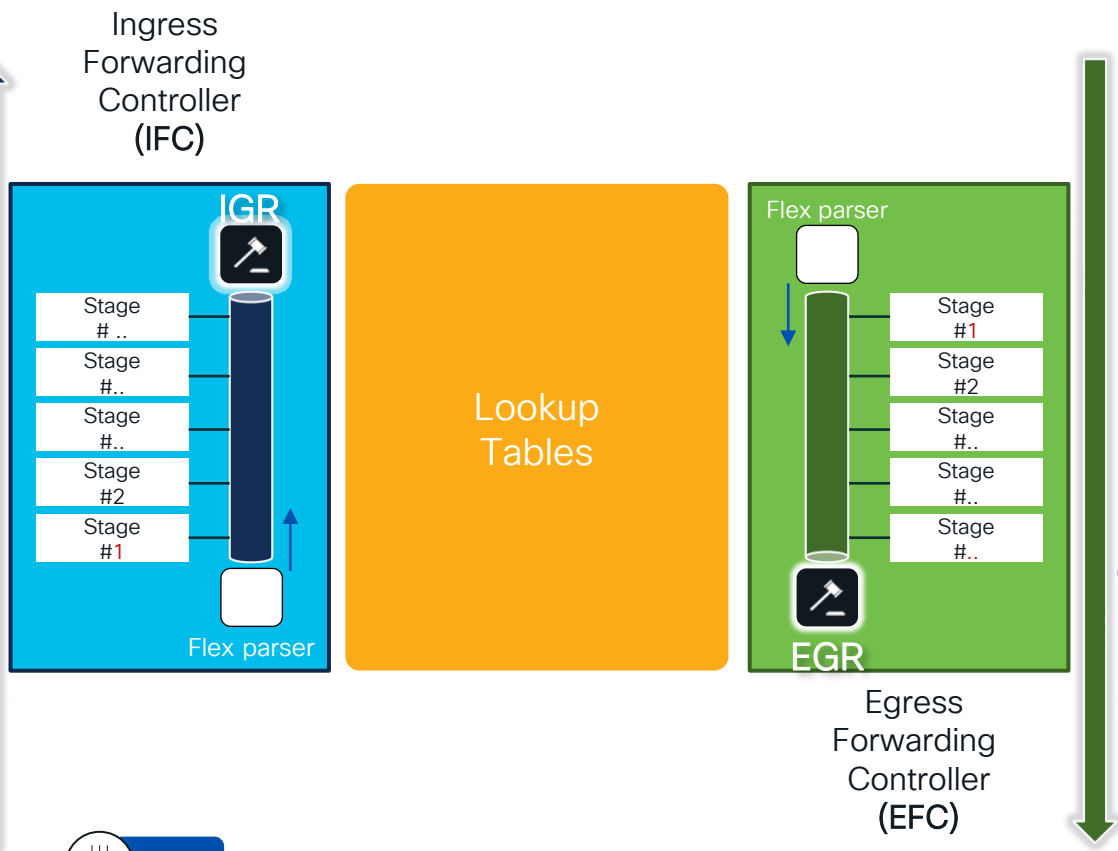


Multiplying ASICs on a switch and multiplying cores with in an UADP is a method to boost processing power

Packet lookup – Programmable Pipelines

Ingress decision on packet's future

- Tunnel Termination Lookup
- Punt to CPU
- PBR Lookup
- Security Association Lookup
- Ingress Policer
- Ingress NetFlow Lookups
- Ingress NetFlow Control
- Ingress ACL, QoS
- Ingress SPAN Lookup
- L3 Destination, (S,G) Lookup
- L2 Destination, (*,G) Lookup
- L2, L3 Source Lookup



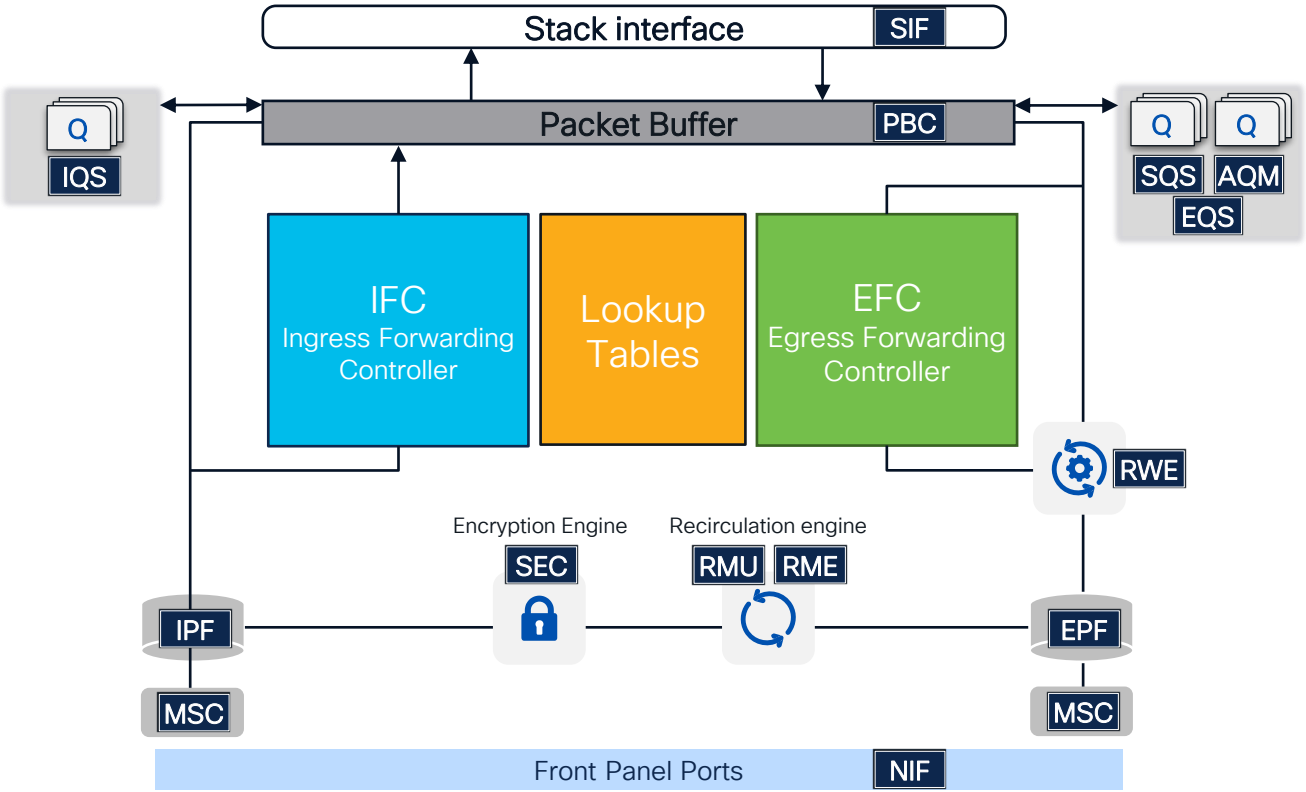
- Egress ACL, QoS Lookup
- Egress NetFlow Lookup
- Egress NetFlow Actions
- Egress Policer
- Security Group Tag Lookup
- Egress SPAN, SA Lookup

Egress decision on packet's future

cisco Live!

Applicable to: UADP

UADP Core Architecture



IQS
Ingress Queuing and Scheduling

IPF
Ingress FIFO

MSC
MACsec

NIF
Network Interfaces

SQS
Stack Queuing and Scheduling

EQS
Egress Scheduler

AQM
Active Queue Management

RWE
Rewrite and Fragmentation Engine

EPF
Egress FIFO

MSC
MACsec

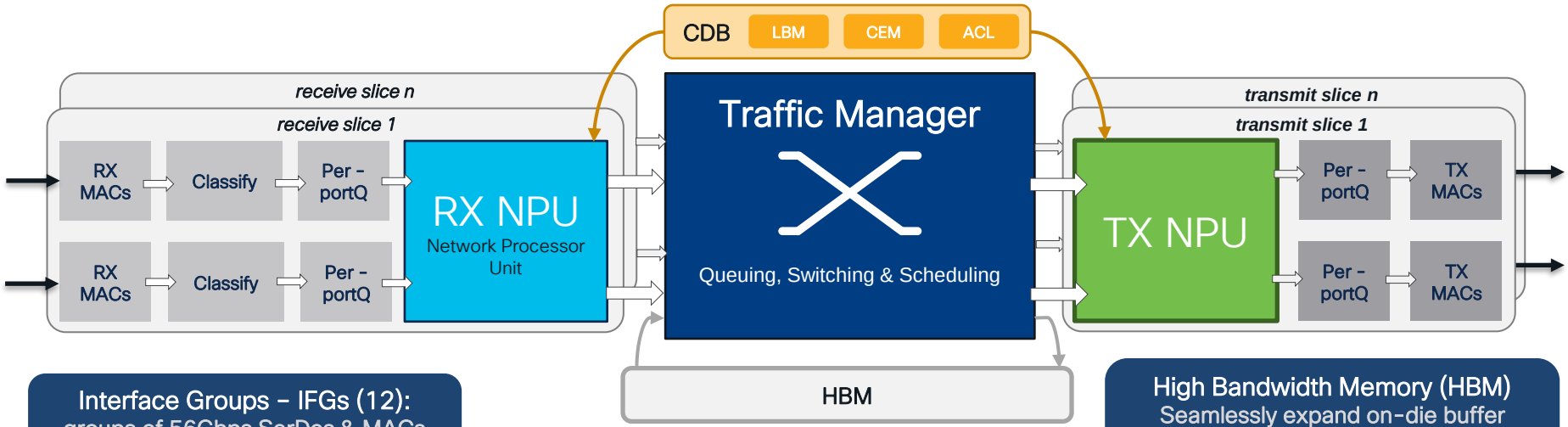
UADP

Cisco Silicon One – Q200

Packet Processing Slices (6):
1 packet per clock (@ 1.35GHz)
Slice = 2x IFGs + 1 RX & TX NPU

RX and TX NPU (per slice):
P4 programmable Run-to-Complete
Large Central Database (CDB) Tables
Expandable LPM in external HBM

Traffic Manager (TM)
Large fully-shared memory switch
Congestion Management
Pool of queues & flexible scheduling



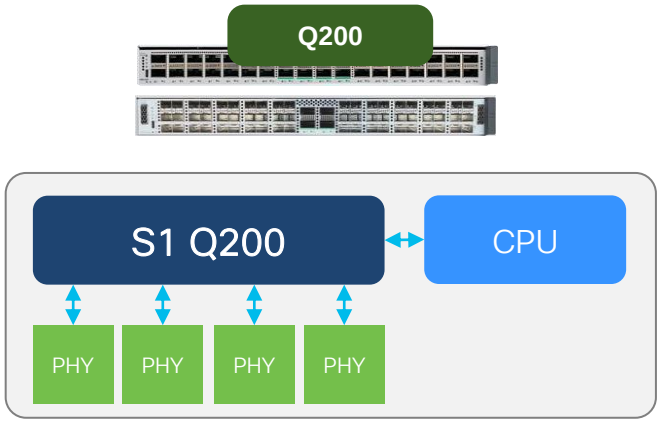
Interface Groups – IFGs (12):
groups of 56Gbps SerDes & MACs
10/25/50GE & 40/100/200/400GE

High Bandwidth Memory (HBM)
Seamlessly expand on-die buffer
expansion of CDB-LPM database
4-8GB of fully shared memory

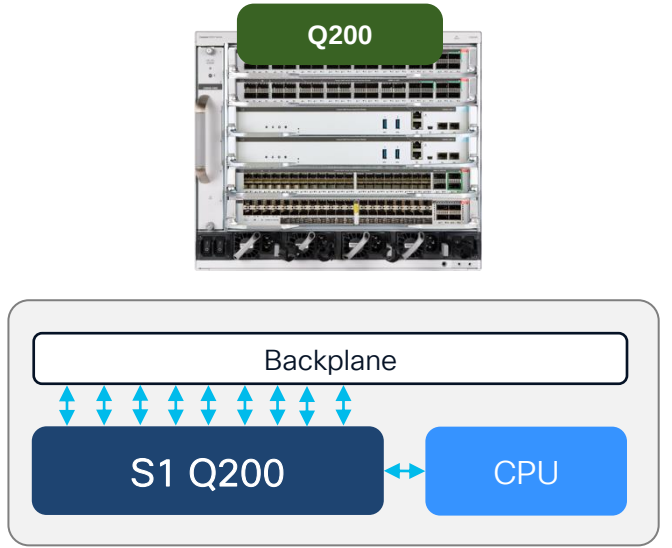
Cisco Silicon One – Q200

Combining multiple ASICs in One SOC

C9500X-28C8D & 60L4D



C9600X-SUP-2

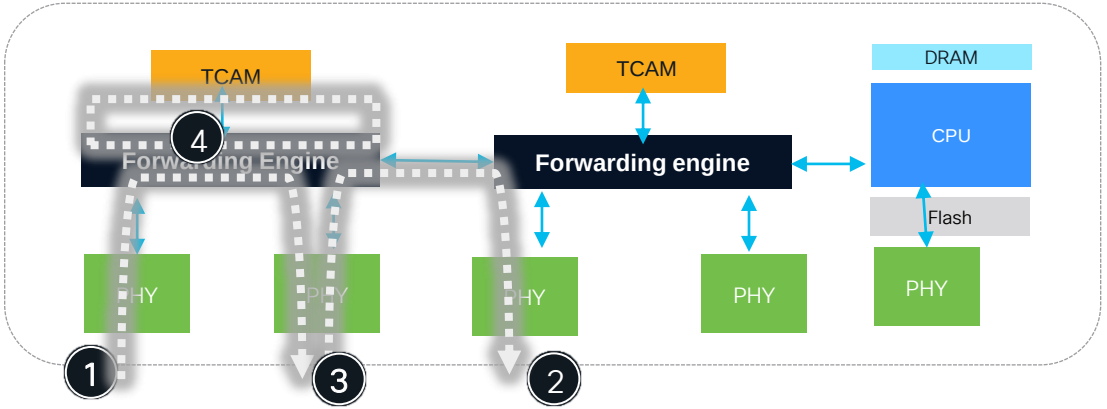


Data Plane Forwarding Scenarios

cisco Live!

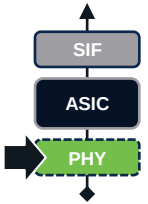
Data plane forwarding

Forwarding Playbook



1. Inner single FE/single core forwarding
2. Intra FE/ multiple core forwarding
3. Intra FE across different stack members
4. Recirculation within FE

PHY counters check




```
Switch# show interface GigabitEthernet 1/0/1
--snip--
     1034 packets input, 124552 bytes, 0 no buffer
     Received 14 broadcasts (13 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
```

IOS counters

```
Switch# show controller ethernet-controller GigabitEthernet 1/0/1
Transmit                    GigabitEthernet1/0/1        Receive
       132970 Total bytes                          125384 Total bytes
         1027 Unicast frames                         1027 Unicast frames
       119944 Unicast bytes                        119944 Unicast bytes
          128 Multicast frames                        14 Multicast frames
        12962 Multicast bytes                       5376 Multicast bytes
            1 Broadcast frames                         1 Broadcast frames
           64 Broadcast bytes                         64 Broadcast bytes
            0 System FCS error frames                  0 IpgViolation frames
--snip--
            0 Late collision frames                    0 SymbolErr frames
            0 Excess Defer frames                      0 Collision fragments
            0 Good (1 coll) frames                     0 ValidUnderSize frames
            0 Good (>1 coll) frames                    0 InvalidOverSize frames
            0 Deferred frames                          0 ValidOverSize frames
            0 Gold frames dropped                      0 FcsErr frames
```

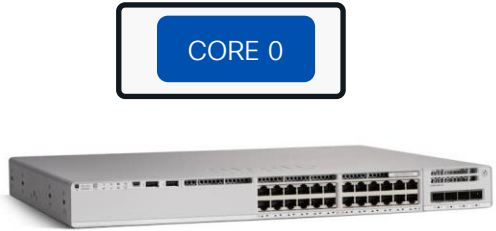
PHY Counters

```
Switch# show controller ethernet-controller GigabitEthernet 1/0/1 phy
```

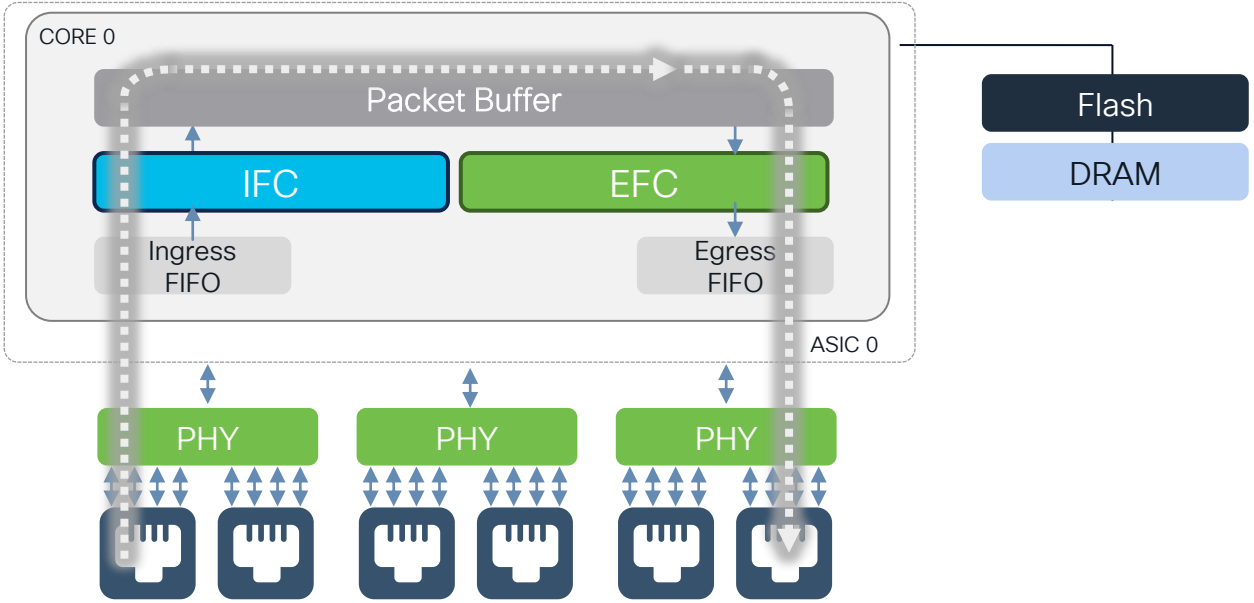
L1 Auto neg status

Single ASIC switch

1 **Inner single FE/single core forwarding**



Example: C9200-24P



Multiplying ASICs on a switch and multiplying cores with in an UADP UADP is a method to boost processing power

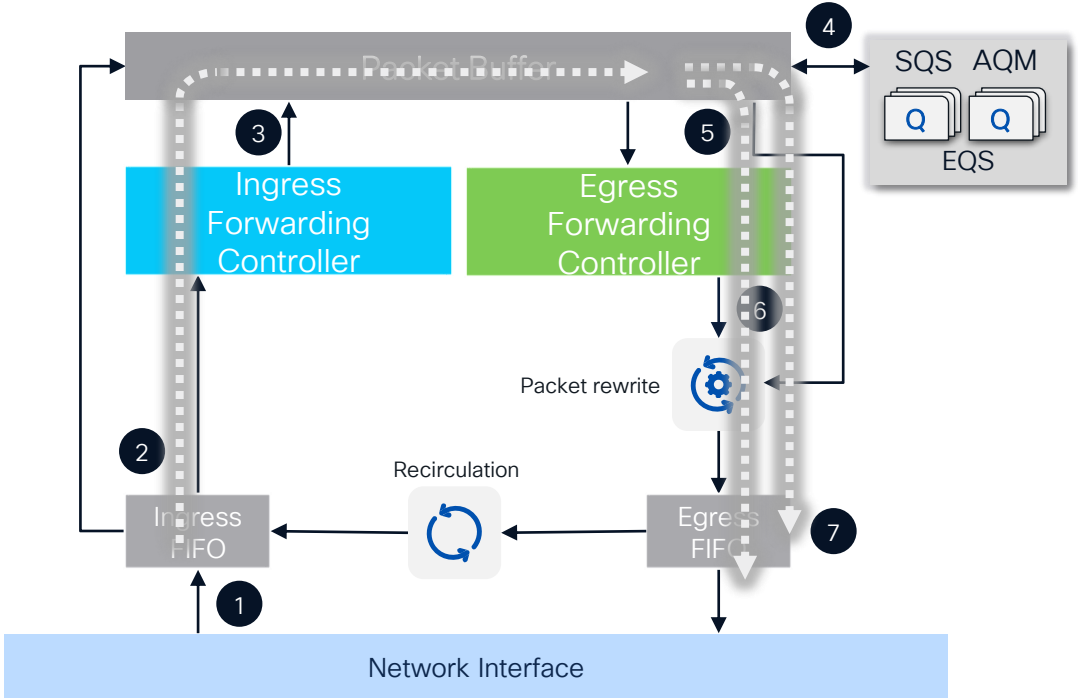
Local Switching: Within the ASIC

4 Descriptor has local destination, PBC sends the info to EQS

3 Goes through IFC, result descriptor send to PBC

2 A copy to buffer, a copy to IFC

1 Received, processed by MACSEC and into FIFO



5 EQS schedule PBC to send a copy to EFC and a copy to ReWrite (includes descriptor)

5a EQS schedule PBC to send another copy to EFC in case of BUM

6 EFC sends results to ReWrite

7 Rewrite the packet and send out through the egress FIFO

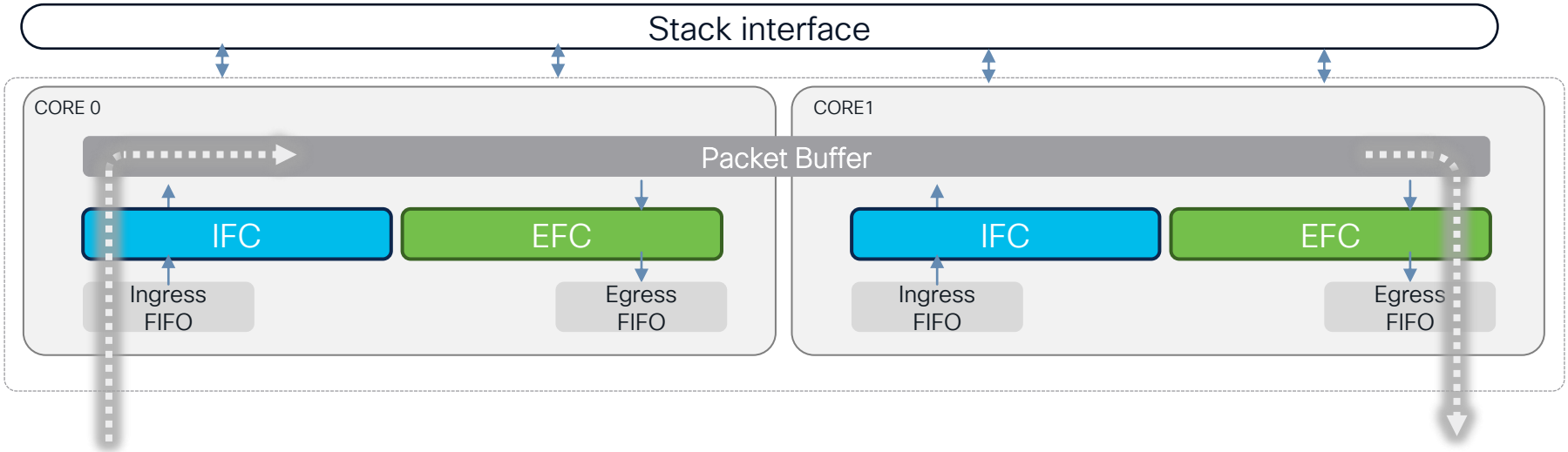
One double-core ASIC

2 **Intra FE/ multiple core forwarding**

ASIC 0

CORE 0 | CORE 1

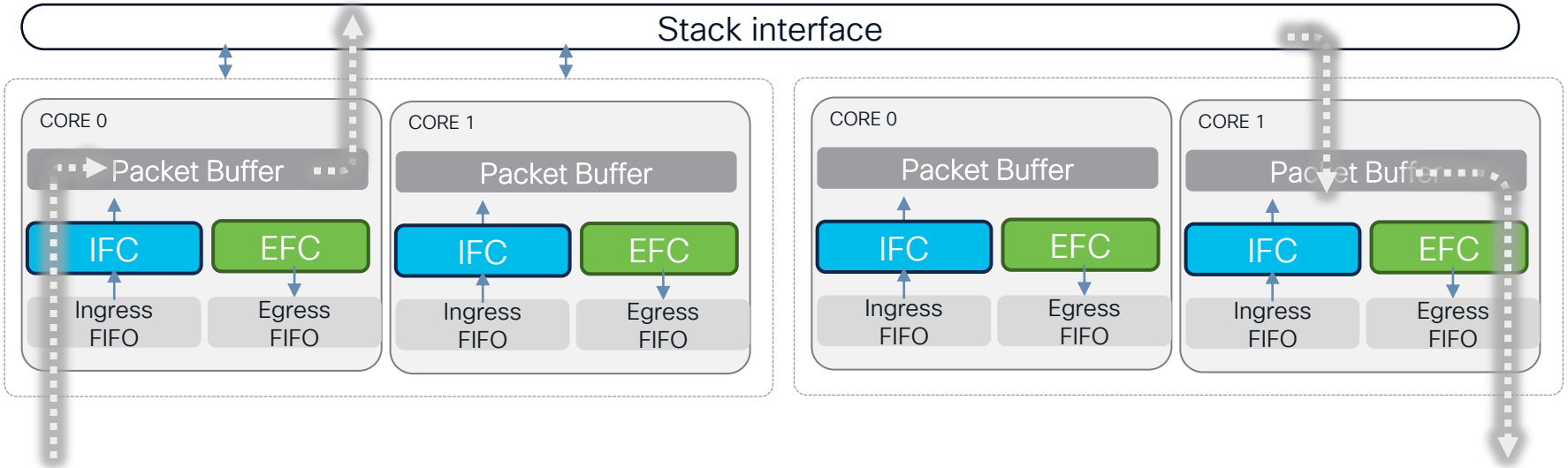
Example: C9300-48T



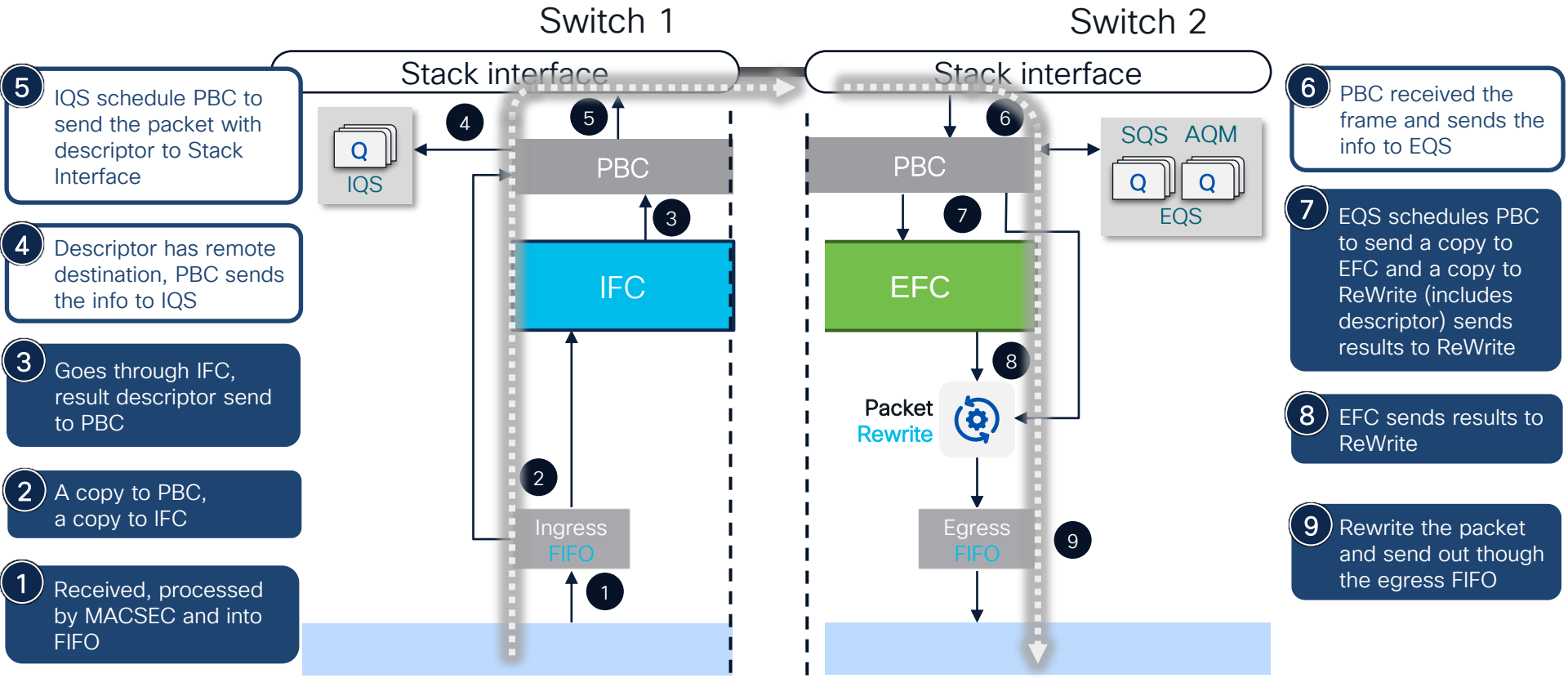
Double ASIC/Across Stack

2 **Intra FE/ multiple core forwarding**

Example: C9300-24UX



Forwarding across ASICs/Stack members



5 IQS schedule PBC to send the packet with descriptor to Stack Interface

4 Descriptor has remote destination, PBC sends the info to IQS

3 Goes through IFC, result descriptor send to PBC

2 A copy to PBC, a copy to IFC

1 Received, processed by MACSEC and into FIFO

6 PBC received the frame and sends the info to EQS

7 EQS schedules PBC to send a copy to EFC and a copy to ReWrite (includes descriptor) sends results to ReWrite

8 EFC sends results to ReWrite

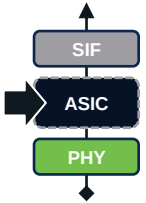
9 Rewrite the packet and send out though the egress FIFO

UADP

ASIC to port Mapping


```
Switch# show platform software fed switch [active|1|2] 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     0x9         1     0     1     1       0      6     7     2     2     NIF   Y
GigabitEthernet1/0/3     0xa         1     0     1     2       0      28    8     3     3     NIF   Y
GigabitEthernet1/0/4     0xb         1     0     1     3       0      27    9     4     4     NIF   Y
GigabitEthernet1/0/5     0xc         1     0     1     4       0      30    10    5     5     NIF   Y
...
GigabitEthernet1/0/17    0x18        0     0     0     16      0      26    6     17    17    NIF   Y
GigabitEthernet1/0/18    0x19        0     0     0     17      0      6     7     18    18    NIF   Y
GigabitEthernet1/0/19    0x1a        0     0     0     18      0      28    8     19    19    NIF   Y
GigabitEthernet1/0/20    0x1b        0     0     0     19      0      27    9     20    20    NIF   Y
...
```

ASIC Exceptions (aka drops)




```
Switch# show platform hardware fed [active|1|2] fwd-asic drops exceptions


****EXCEPTION STATS ASIC INSTANCE 0 (asic/core 0/0)****
================================================================================
 Asic/core |                NAME                |   prev   |  current  |  delta
================================================================================
0  0  NO_EXCEPTION                                1653       1857        202
0  0  IPV4_CHECKSUM_ERROR                         54         61          7
0  0  ROUTED_AND_IP_OPTIONS_EXCEPTION             0          0           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
--snip--
```

Identify correct asic instance

delta column helps to identify the name of the exception

Exception name

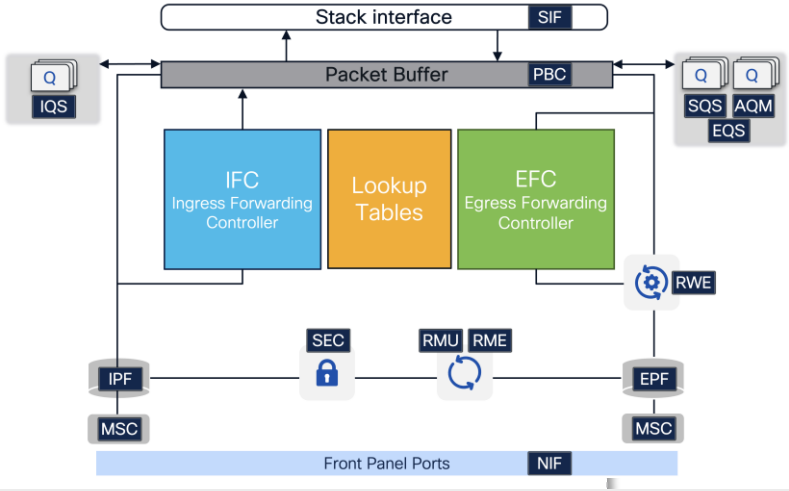
Is my ASIC/Core dropping any traffic ?

ASIC Data Path

```
Switch# show platform hardware fed switch [active|1|2] fwd-asic data-path asic_port 1 asic_no 0 asic_core 1

data path:
    +--------------+
    |      Sif     |
    |  IQS     SQS |
    |      PBC     |
    |          AQM |
    |          EQC |
    |          ESM |
    |          RWE |
    |    ASIC 0    |
    |    Core 1    |
    |              |
    |    (Mac  2)  |
    |Nif_Rx   NifTx|
    +--------------+
        ^        |
        |        |
        |        |
        |        V
-- snip ---
```

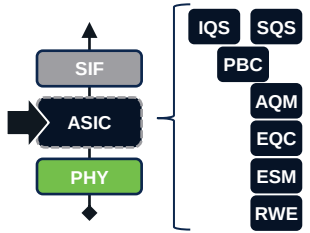
Direct correlation of ASIC functional blocks

```
show platform software fed switch ... ifm mappings
Interface                IF_ID     Inst  Asic  Core  Port
GigabitEthernet1/0/2     0x9         1     0     1     1
```



Allows to correlate FE counters with a front panel port

ASIC Data path cont



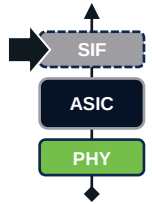

```
Switch# show platform hardware fed switch [active|1|2] fwd-asic data-path
asic_port 1 asic_no 0 asic_core 1

-- snip ---
For local/core 0 Switching:
                                        SqsCumulativeStatistics
                                         totalEnqStat                29940959
                                         totalDeqStat                29940959
                                         totalDropStat               0
                                        SqsCumulativeStatisticsB
                                         totalEnqStat                9624718091
                                         totalDeqStat                9624718091
                                         totalDropStat               0
================================================
For local/core 1 Switching:
                                        SqsCumulativeStatistics
                                         totalEnqStat                92354313
                                         totalDeqStat                92354313
                                         totalDropStat               0
                                        SqsCumulativeStatisticsB
                                         totalEnqStat                20153637926
                                         totalDeqStat                20153637926
                                         totalDropStat               0
================================================
-- snip ---
```

ASIC functional block

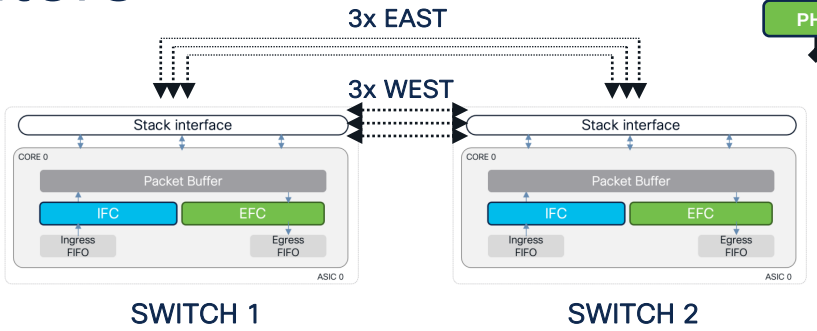
Not only stats but also errors and drops

Stack interfaces CRC counters

Registers check



3 **Intra FE across different stack members**




```
Switch# show platform hardware fed switch [] fwd-asic register read register-name SifRacDataCrcErrorCnt
For asic 0

module 0
SifRacDataCrcErrorCnt on  Asic 0
[0]             count 0x00000002
[1]             count 0x00000060
[2]             count 0x00000058
[3]             count 0x00000071
[4]             count 0x00000054
[5]             count 0x0000005f
```

6 rings :

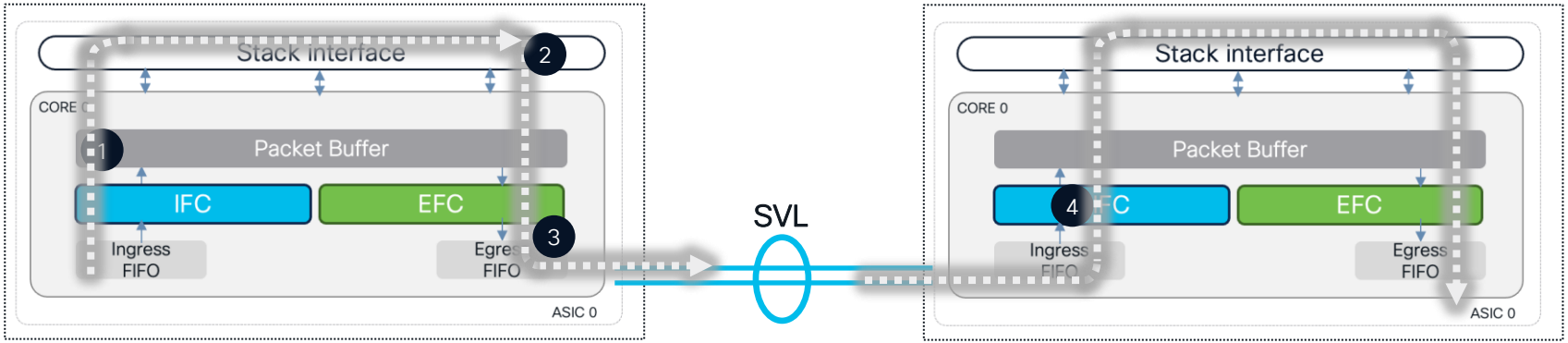
- 3x WEST
- 3x EAST

```
SifRacDataCrcErrorCnt
SifRacRwCrcErrorCnt
SifRacPcsCodeWordErrorCnt
SifRacInvalidRingWordCnt
```

Additional stack registers

Stackwise Virtual

Packet Forwarding to remote switch



1. Ingress - same as before
2. DI (destination Index) points to the SVL
3. Packets going over SVL skip egress processing
4. Packets from SVL skip all the ingress processing and go to egress
5. Egress – same as before

S1 ASIC switch

1 **Inner single FE/single core forwarding**

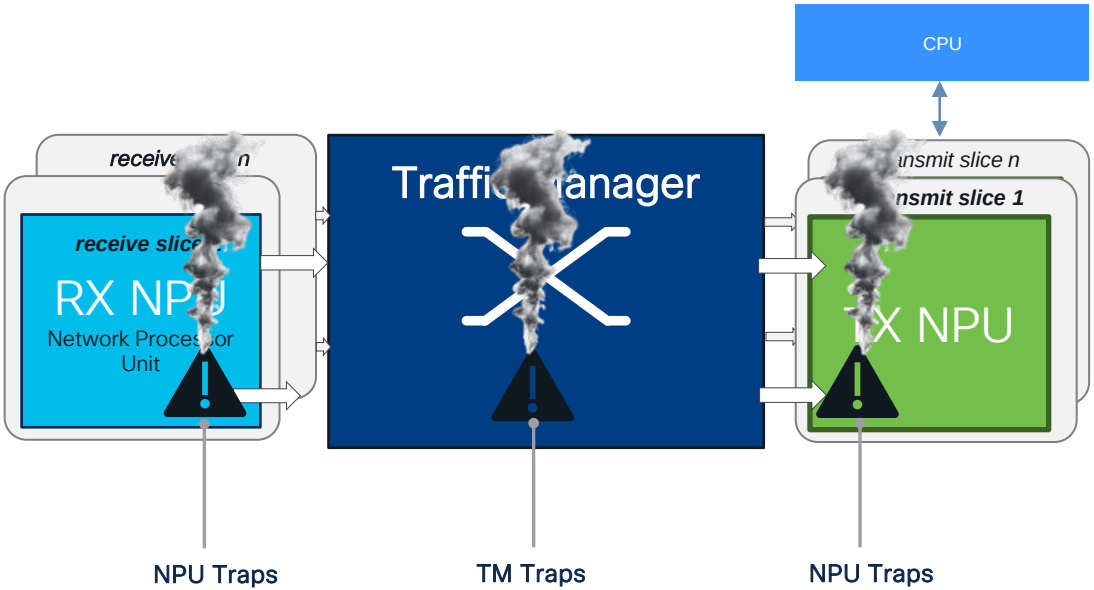
S1 TRAP Mechanism

- Traffic based
- Triggered based on predefined set of events / type of packets (Expected as well as non expected behaviours)
- Available for NPUs and TM components
- Allows to redirect traffic to CPU instead silently dropping it

Ability to understand dropped traffic

S1 ASIC switch

1 **Inner single FE/single core forwarding**



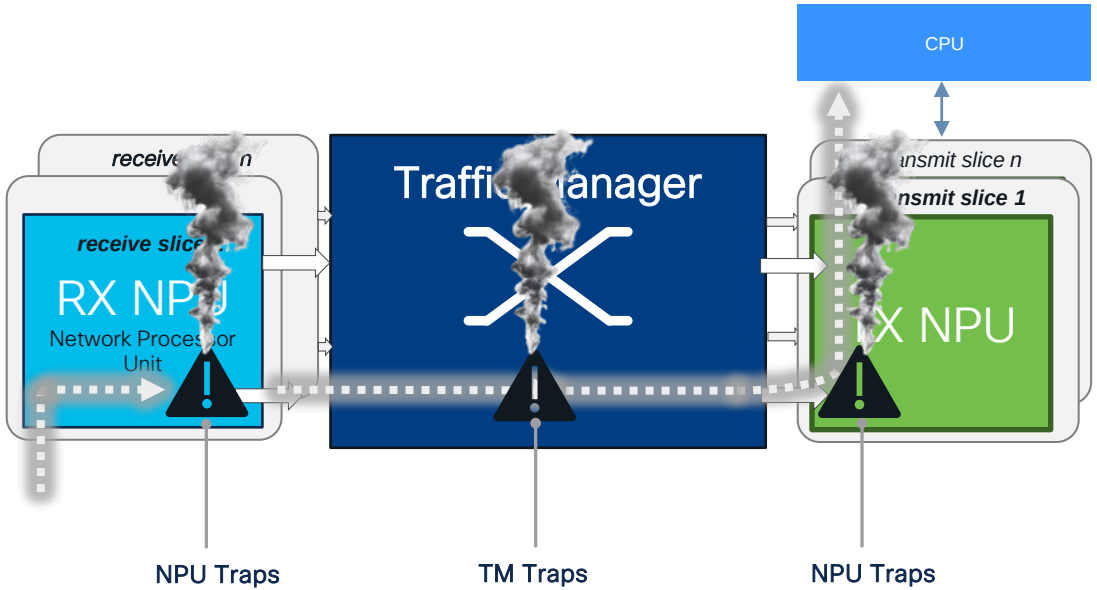
S1 TRAP Mechanism

- Traffic based
- Triggered based on predefined set of events / type of packets (Expected as well as non expected behaviours)
- Available for NPUs and TM components
- Allows to redirect traffic to CPU instead silently dropping it

Ability to understand dropped traffic

S1 ASIC switch

1 **Inner single FE/single core forwarding**



S1 TRAP Mechanism

- Traffic based
- Triggered based on predefined set of events / type of packets (Expected as well as non expected behaviours)
- Available for NPUs and TM components
- Allows to redirect traffic to CPU instead silently dropping it

Ability to understand dropped traffic

EPDA – Enhanced Packet Drop Analyzer

1 Identify traffic

```
Switch# show platform hardware fed active fwd-asic traps npu-traps asic all
-------------------------------------------------------------------------------------------------------------
Trap ID |     Asic  |                    NPU Trap Name                    |     Prev     |   Current    |    Delta
-------------------------------------------------------------------------------------------------------------
      1 |         0 |                        la_event_e_ETHERNET_ACL_DROP |            0 |            0 |            0
      2 |         0 |                   la_event_e_ETHERNET_ACL_FORCE_PUNT |            0 |            0 |            0
      4 |         0 |       la_event_e_ETHERNET_NO_TERMINATION_ON_L3_PORT |            0 |            0 |            0
      5 |         0 |                  la_event_e_ETHERNET_CISCO_PROTOCOLS |            2 |            2 |            0
      6 |         0 |                        la_event_e_ETHERNET_DA_ERROR |            0 |            0 |            0
      7 |         0 |                  la_event_e_ETHERNET_DHCPV4_CLIENT |            0 |            0 |            0
<snip>
    149 |         0 |                             la_event_e_L3_NULL_ADJ |        30255 |        34850 |         4595
```

2 Enable selected trap

```
Switch# debug platform software fed active drop-capture set-trap npu-traps l3 l3-null-adj
```

```
Switch# debug platform  software fed active drop-capture start
```

3 Capture traffic

```
Switch# debug platform  software fed active drop-capture stop
```

EPDA Continued

4 Display packets

```
Switch# show platform software fed active drop packet-capture brief
DropPackets packet capturing: disabled. Buffer wrapping: disabled
Total captured so far :  2313 packet(s)
Capture capacity      :  4096 packet(s)
Max. Meta header size :    88 byte(s)
Max. Packet data size :   128 byte(s)

------ DropPackets Packet Number: 1, Timestamp: 2024/03/25 15:04:46.823 ------
 interface : phy:  [if-id: 0x00000000], pal:  [if-id: 0x00000000]
 misc info : cause: 0 [Reserved ], sub-cause: 0, linktype: UNKNOWN [0]
 CE    hdr : dest mac: 4e41.5000.0111, src mac: 4e41.5000.0111, ethertype: 0x7106
 meta  hdr : Nxt. Hdr: 0x1, Fwd. Hdr: 0x2, SSP: 0x19
 meta  hdr : DSP: 0xffff, SLP: 0xe, DLP: 0x95
 ether hdr : dest mac: 341b.2d76.fd02, src mac: 6c29.d29d.36c3
 ether hdr : vlan: 3012, ethertype: 0x8100
 ipv4  hdr : dest ip: 172.16.10.11, src ip: 192.168.100.18
 ipv4  hdr : packet len: 100, ttl: 254, protocol: 1 (ICMP)
 icmp  hdr : icmp type: 8, code: 0
```

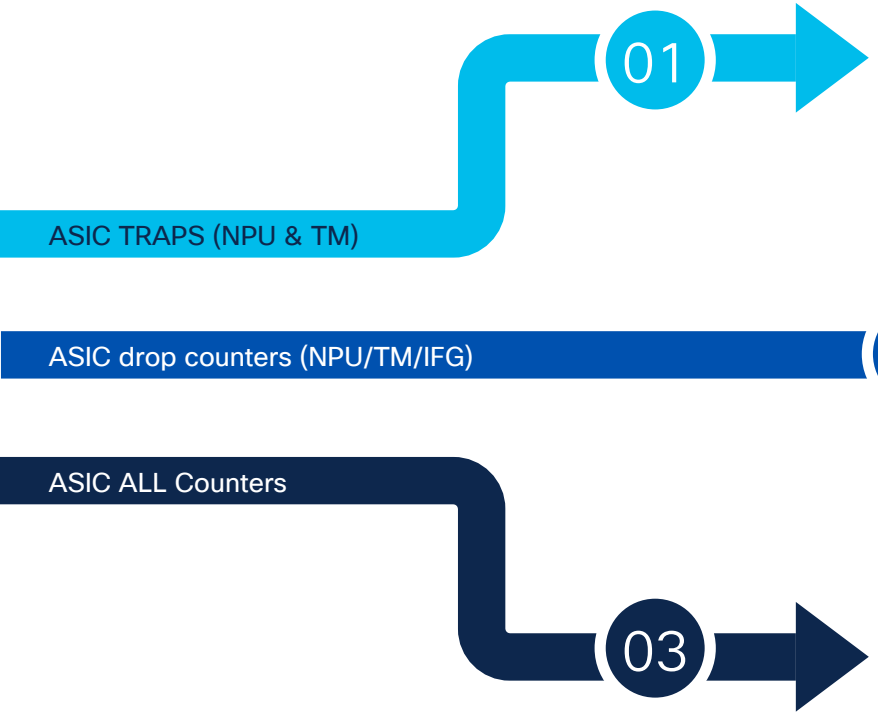
Default buffer of 4k (can be changed to 16k)

Details of dropped packet

5 Clear trap

```
Switch# debug platform software fed active drop-capture clear-trap npu-traps l3 l3-null-adj
```

S1 forwarding drops troubleshooting



97%

- TRAPS counters and the EDD are the first steps in the troubleshooting process.
- Drops captured by TRAPS are the most common and allow the collection of extra insights regarding the dropped packets.

```
#show platform hardware fed active fwd-asic traps ?
  npu-traps  View NPU Traps
  tm-traps   View TM Traps
```

2%

- Drop counters do not provide the ability to capture extra insights and can be collected at ASIC-level granularity.

```
#show platform hardware fed active fwd-asic drops ?
  asic  asic
  ifg   View drop ifg
  npu   View drop npu
  tm    View drop tm
```

<1%

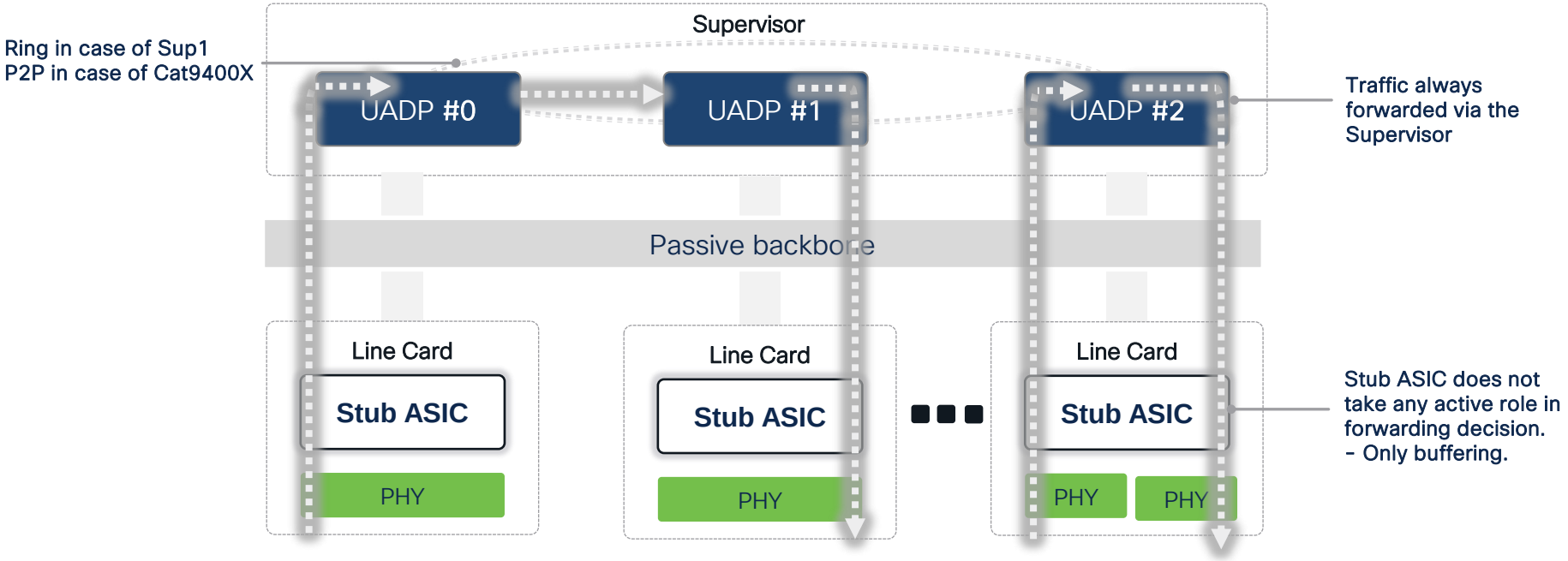
- ALL counters represent both the normal counters measuring forwarded traffic as well as counters that might represent drops or errors.

```
#show platform hardware fed active fwd-asic counters 0
counters_all
```

Catalyst 9400 Centralized Forwarding

UADP 2.0XL — Cat9400

UADP 3.0 — Cat9400X

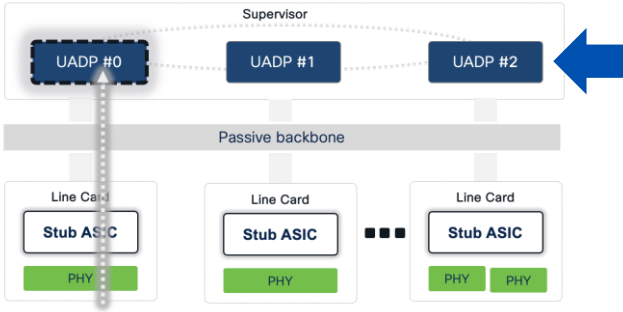


Data path: Supervisor statistics

```
Switch# show platform hardware cman fp active data-path 1 1 detail
--snip–
data path:

        slot 5
    +- ACTIVE_SUP --+
    |   Sif 1       |
    | IQS     SQS   |
    |     PBC       |
    |         AQM   |
    |         EQC   |
    |         ESM   |
    |         RWE   |
    |   ASIC 0      |
    |   Core 0      |
    | Asic Port  0  |
    |               |
    |   (Mac  4)    |
    |Nif_Rx   NifTx |
    +---------------+
        ^       |
        |       |
        |       |
        |       V
--snip–
```

Linecard 1 | Port 1

ASIC 0, Core 0 on slot 5 Active Supervisor is associated with Gig1/0/1



Associates forwarding counters with the TLA from the UADP ASIC

NIF represents the connection in between the LC and Supervisor

Data path: Supervisor statistics, cont.

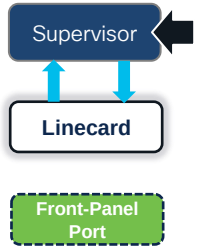
```
Switch# show platform hardware cman fp active data-path 1 1 detail
```

NIF Rx stats

```
NifRxByteGroupStats:
 rxBytes                    23015204

NifRxPortStatusGroupStats:
 rxCollisionFragments       0
 rxFcsErrorFrames           0
 rxInvalidOversizeFrames    0
 rxMacOverrunFrames         0
 rxIpgViolationFrames       0
 rxOamDroppedFrames         0
 rxSymbolErrorFrames        0
 rxValidOversizeFrames      0
 rxValidUndersizeFrames     0
--snip--
```

NIF Tx stats

```
NifTxByteGroupStats:
 txBytes                    7155446474775

NifTxPortStatusGroupStats:
 txLateCollisionFrames      0
 txsystemFcsErrorFrames     0
 txOversizeFrames           0
 txMacUnderrunFrames        0
 txDeferredFrames           0
 txExcessiveDeferralFrames  0
 txOkMultipleCollisionFrames  0
 txOkSingleCollisionFrames  0
 goldFramesTruncated        0
--snip--
```



Data path: Stub ASIC Line card statistics

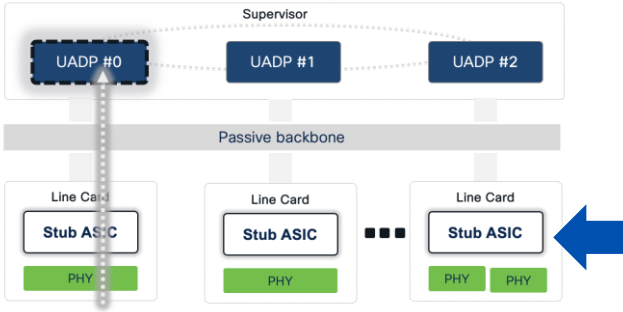
```
Switch# show platform hardware iomd 1/0 data-path 1 detail
--snip--
data path:
```

Linecard 1 Port 1

```
                slot 5
           +--ACTIVE SUP--+
           |              |
           |   ASIC 0     |
           |   Core 0     |
           | Asic Port  0 |
           |              |
           |   (Mac  4)   |
           |Nif_Rx   NifTx|
           +--------------+
                  |      |
 SLI MAC  8       |      |
           +---------------+
           | SLI_Tx  SLI_Rx|
           |               |
           |    ASIC 0     |
           | Asic  Port  0 |
           |               |
           |   (Mac 12)    |
           | NIF_Rx  NIF_Tx|
           +---------------+
             Front Port 1/0/1
               ^       |
               |       V
```

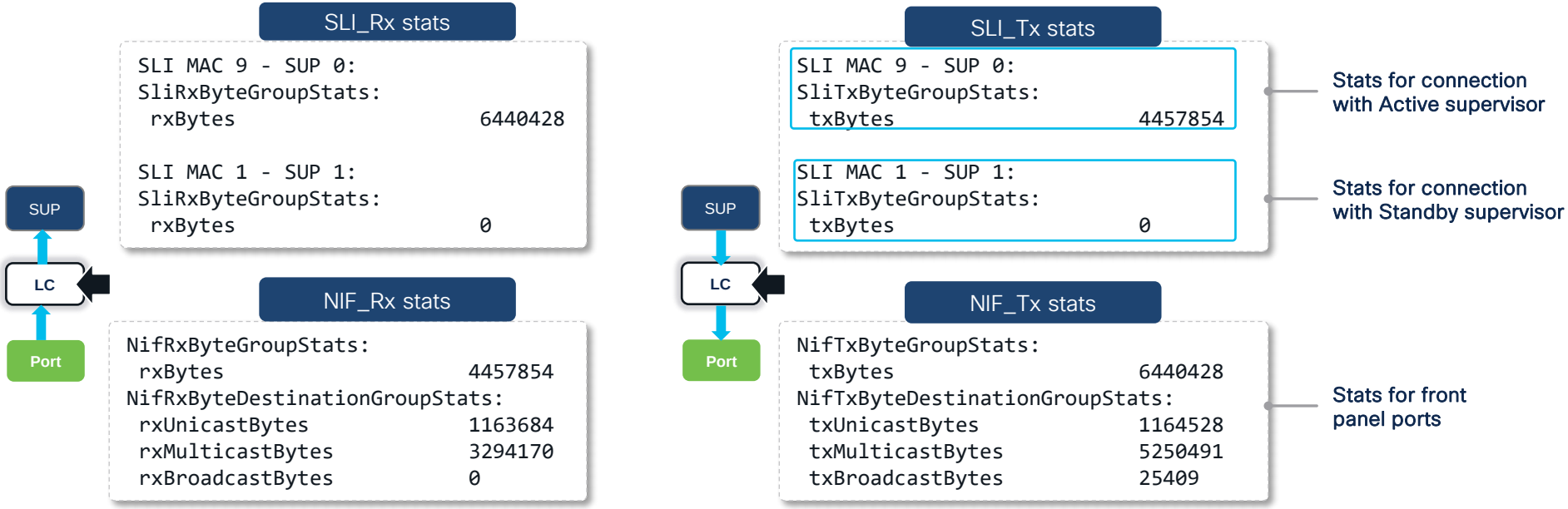
ASIC 0, Core 0 on slot 5 Active Supervisor is associated with Gig1/0/1

Line card 1. Stats in this command are only for line card ASIC



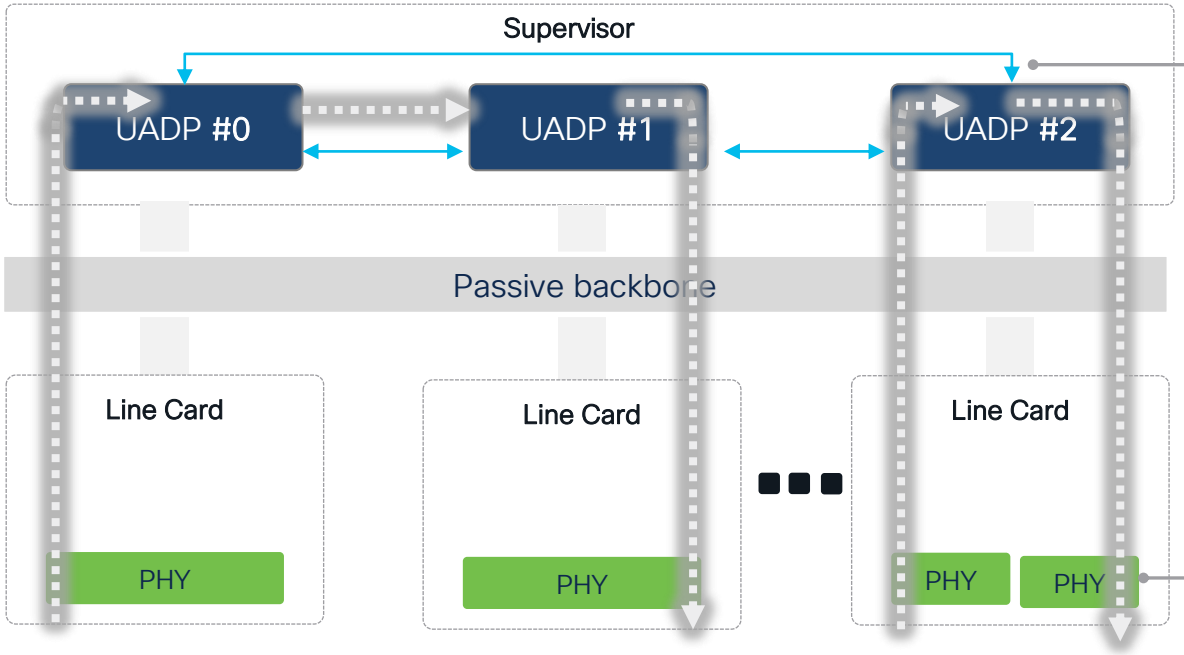
Data path: Line card statistics, cont.

```
Switch# show platform hardware iomd 1/0 data-path 1 detail
```



Catalyst 9600 Centralized Forwarding

UADP 3.0

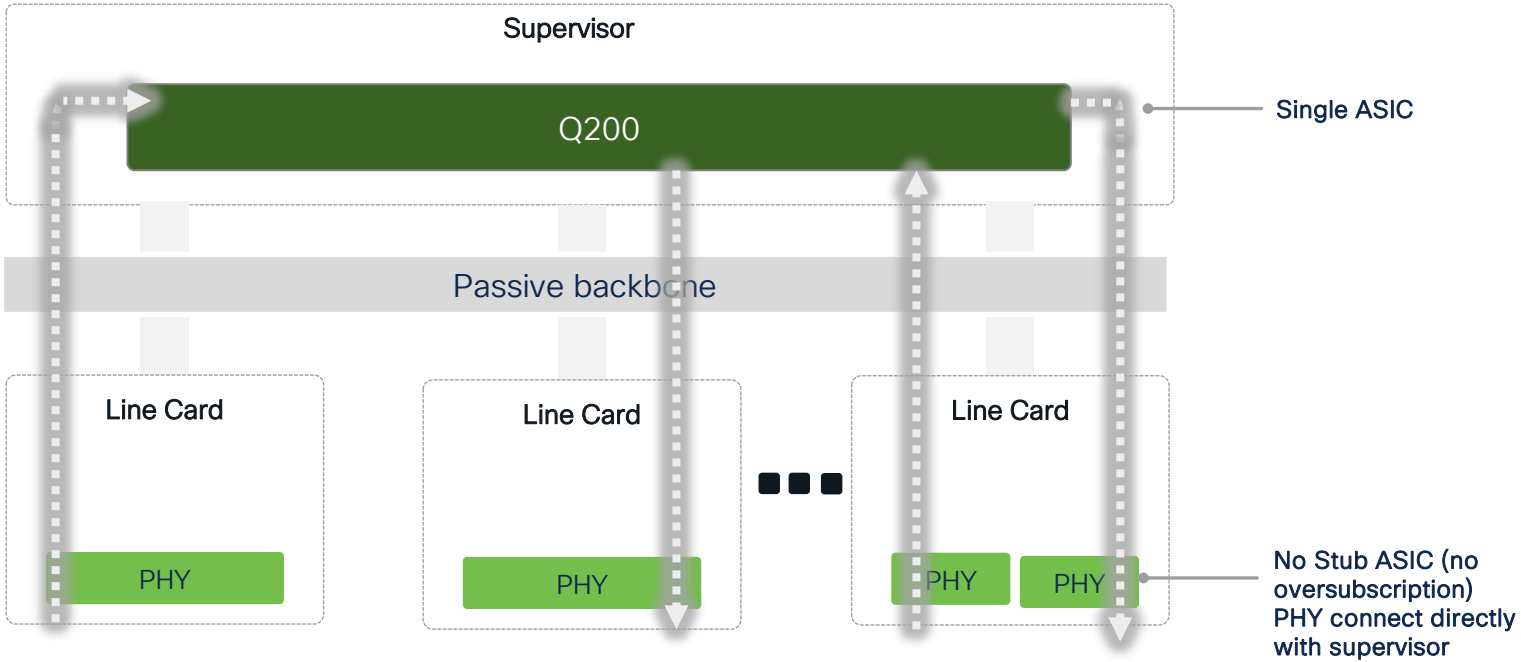


No ring – Direct P2P connection in between UADP 3.0

No Stub ASIC (no oversubscription) PHY connect directly with supervisor

Catalyst 9600 Centralized Forwarding

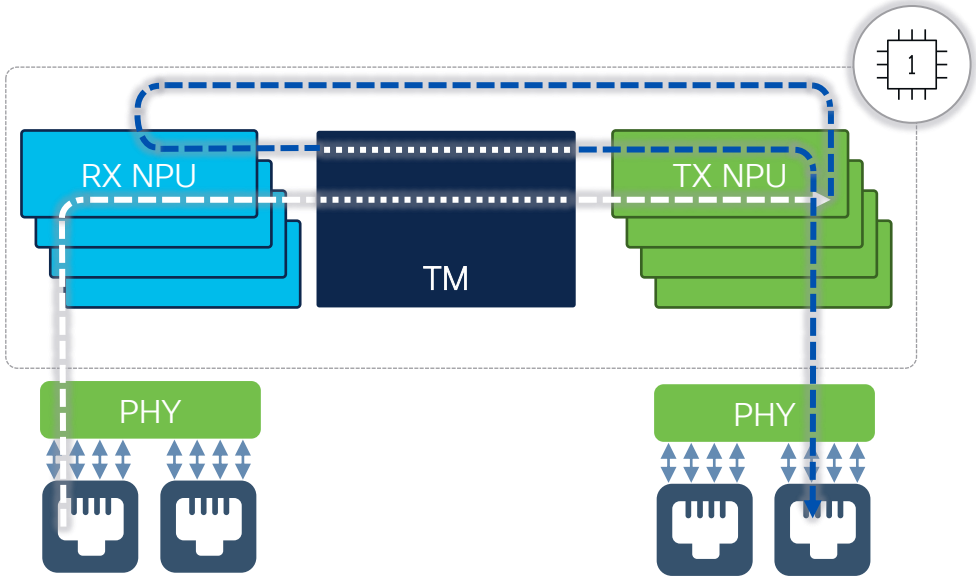
Q200

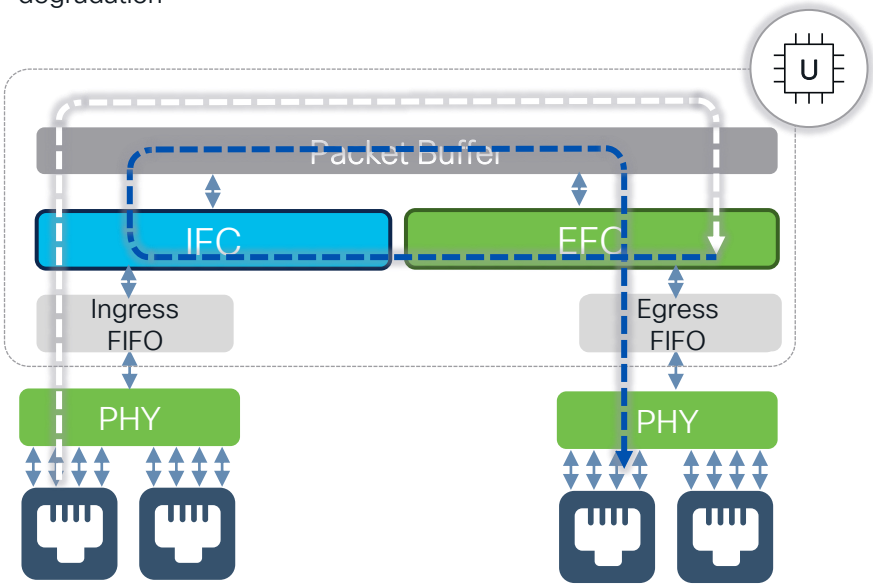


Advanced Forwarding

Recirculation

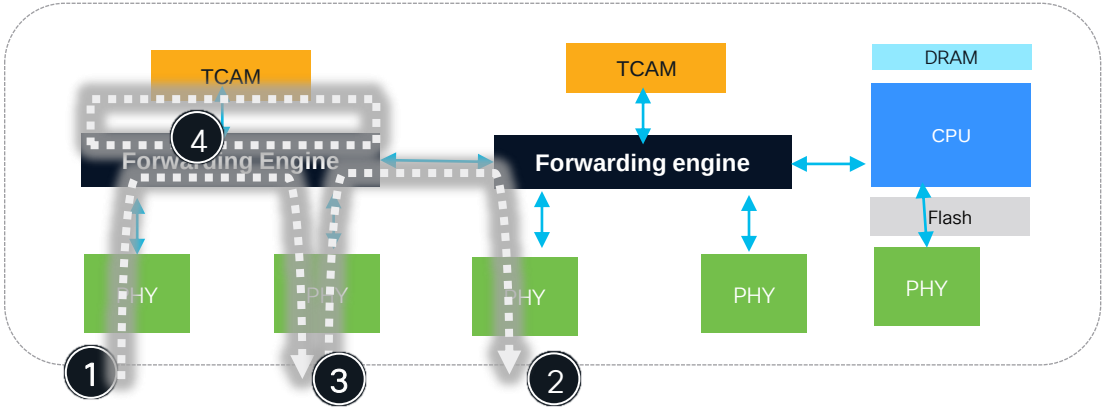
- Advanced scenario typically involved in double IP packet encapsulations (eg. VXLAN SDA/EVPN)
- Leveraging a concept of Internal RCP (Recirculation port)
- 1st Pass outer IP lookups + removal of outer IP
- 2nd Pass inner IP lookups + final packet rewrite
- Recirculation does not cause observable performance degradation





Data plane forwarding

Forwarding Playbook

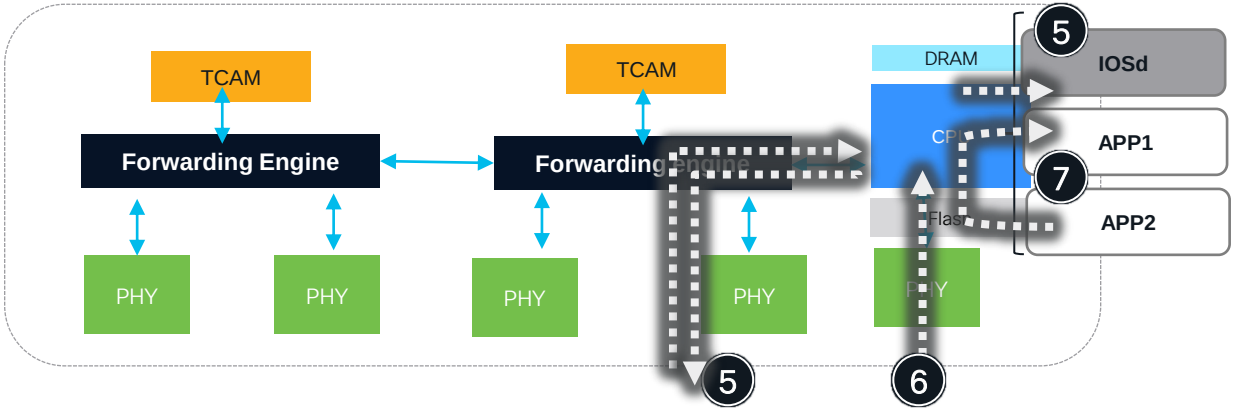


1. Inner single FE/single core forwarding
2. Intra FE/ multiple core forwarding
3. Intra FE across different stack members
4. Recirculation within FE

Control Plane Packet Path

Data plane & control plane – is that all?

Forwarding Playbook

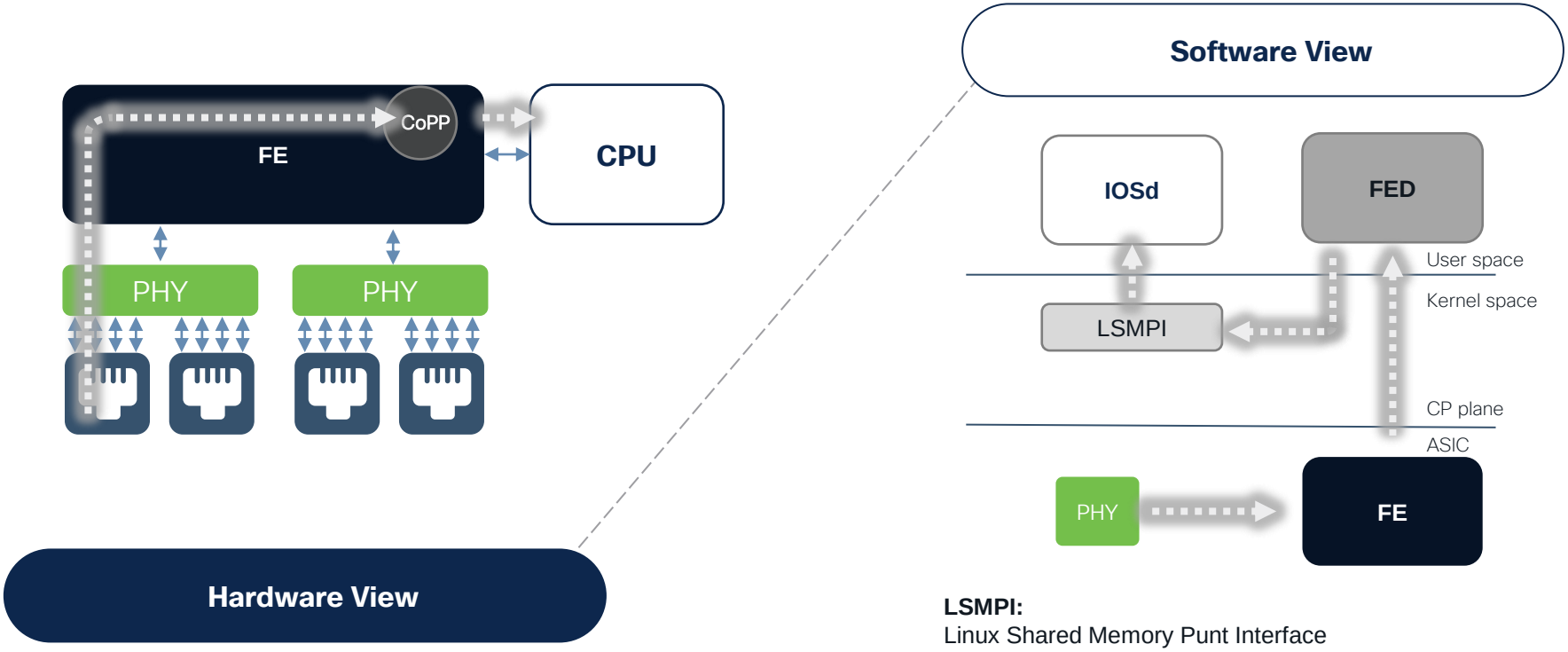


5 **IOS Control-plane forwarding**

6 **Mgmt port forwarding**

7 **App-hosting forwarding**

5: Control Plane – HW vs SW Moon view

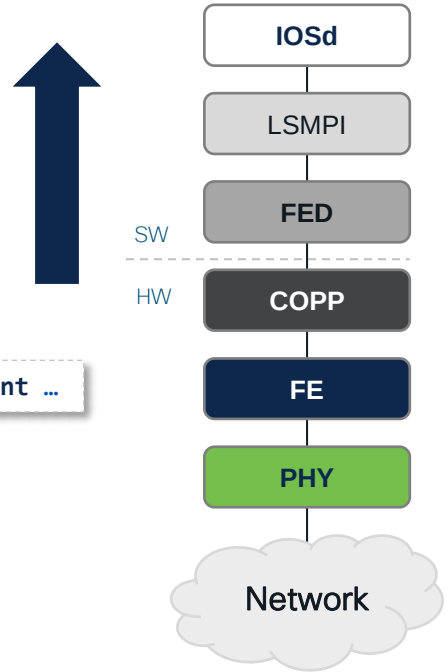


LSMPI:
Linux Shared Memory Punt Interface

CP Architecture

Punt

- Network traffic travels from hardware to software for CPU Processing



Inject

- Local CPU generated network traffic travels from software to hardware into the network

```
show platform software fed switch <> punt …
```

```
show platform software fed switch <> inject …
```

CoPP Overview

CoPP is ON by default

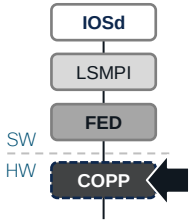
```
Switch(config)# control-plane
Switch(config-cp)# no service-policy in system-cpp-policy
Switch(config-cp)# service-policy in custom-CoPP
Policy map system-cpp-policy is already attached
```

```
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# no class system-cpp-police-sys-data
 It is not allowed to delete system cpp classes
Remove class system-cpp-police-sys-data from policy
system-cpp-policy is not allowed.
```

```
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-police-sys-data
Switch(config-pmap-c)# no police rate 600 pps
Switch(config-pmap-c)# end
```

- Only the **system-cpp-policy** policy-map can be installed on the control plane interface
- The **system-cpp-policy** policy-map and the system-defined classes cannot be modified or deleted.
- Only the police action is allowed under the **system-cpp-policy policy-map**. Further, the police rate can be configured only in packets per second (pps). (UADP Only)
- On S1 we can only disable/enable individual pre configured classes

CoPP Overview UADP




```
Switch# show platform 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      1222          314
2    14     Forus traffic                 Yes      4000     4000      0             0
3    0      ICMP GEN                      Yes      750      750       0             0
4    2      Routing Control               Yes      5500     5500      0             0
<snip>
11   13     L2 LVX Data Pack              Yes      1000     1000      0             0
12   0      BROADCAST                     Yes      750      750       0             0
13   10     Openflow                      Yes      250      250       0             0
14   18     Sw forwarding                 Yes      1000     1000      0             0
<snip>
```

Non-zero counters indicate a loss of punted traffic

Policer Index

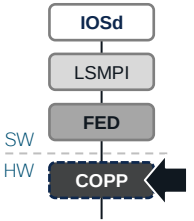
Hardware Q IDs

Policing values

Drops

UADP

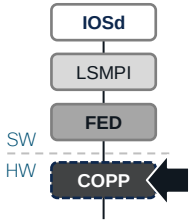
CoPP Overview UADP cont.




```
Switch# show platform hardware fed [switch] active qos queue stats internal cpu policer
<snip>
 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 Failed/
13     system-cpp-police-sw-forward              :  Sw forwarding/ LOGGING/ L2 LVX Data Pack/ Transit Traffic/
9      system-cpp-police-multicast               :  MCAST Data/
15     system-cpp-police-multicast-end-station   :  MCAST END STATION /
7      system-cpp-police-punt-webauth            :  Punt Webauth/
1      system-cpp-police-l2-control              :  L2 Control/
2      system-cpp-police-routing-control         :  Routing Control/ Low Latency/
3      system-cpp-police-system-critical         :  System Critical/ Gold Pkt/
4      system-cpp-police-l2lvx-control           :  L2 LVX Cont Pack/
8      system-cpp-police-topology-control        :  Topology Control/
11     system-cpp-police-dot1x-auth              :  DOT1X Auth/
12     system-cpp-police-protocol-snooping       :  Proto Snooping/
<snip>
```

UADP

CoPP Policer S1




```
Switch# show platform software fed active punt entries
Punject Punt Entries

Source    Name               Pri  TC  Policy                        CIR-SW  CIR-HW Pkts(A) Bytes(A) Pkts(D) Bytes(D)
TRAP      ACL Drop(ETH)      1    0   system-cpp-default             2000    1931   0       0        0       0
MIRROR    ARP                4    4   system-cpp-police-arp          1000    965    0       0        0       0
TRAP      CISCO Protocols    3    5   system-cpp-police-l2-control   16000   15449  2       636      0       0
TRAP      DHCP Client(v4)    3    4   system-cpp-police-dhcp-v4      6000    5793   3245    2332525  245     3532
TRAP      DHCP Server(v4)    3    4   system-cpp-police-dhcp-v4      6000    5793   0       0        0       0
TRAP      DHCP Client(v6)    3    4   system-cpp-police-dhcp-v6      6000    5793   0       0        0       0
TRAP      DHCP Server(v6)    3    4   system-cpp-police-dhcp-v6      6000    5793   0       0        0       0
TRAP      ETH HOP-OPT        88   3   system-cpp-police-sw-forward   2000    1931   0       0        0       0
MIRROR    ISIS(L2)           3    5   system-cpp-police-isis         1000    965    4       6284     0       0
TRAP      LLDP               4    5   system-cpp-police-l2-control   16000   15449  2       636      0       0
<snip>
```

Configuration Class

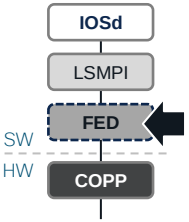
Policing Values

Accepted

Dropped

Forwarding Engine Driver

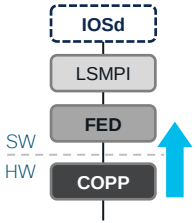
```
Switch# show platform software fed switch active punt cpuq all
Punt CPU Q Statistics
=============================================
<snip>
CPU Q Id                        : 1
CPU Q Name                      : CPU_Q_L2_CONTROL
Packets received from ASIC      : 794283
Send to IOSd total attempts     : 794283
Send to IOSd failed count       : 0
RX suspend count                : 0
RX unsuspend count              : 0
RX unsuspend send count         : 0
RX unsuspend send failed count  : 0
RX consumed count               : 0
RX dropped count                : 0
RX non-active dropped count     : 0
RX conversion failure dropped   : 0
RX INTACK count                 : 794280
RX packets dq'd after intack    : 13
Active RxQ event                : 794280
RX spurious interrupt           : 29
RX phy_idb fetch failed           : 0
<snip>
```

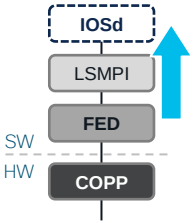



```
CPU Queue Statistics
============================================
QId PlcIdx  Queue Name                 Enabled
--------------------------------------------
0    11     DOT1X Auth                  Yes
1    1      L2 Control                  Yes
<snip>
```

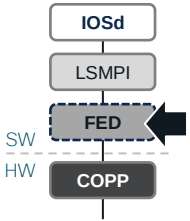
Sent by FED to IOSd

Received by FED from HW





Forwarding Engine Driver



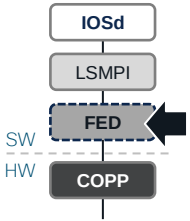

```
Switch# show platform software fed active punt asic-cause brief
ASIC Cause Statistics Brief
+-------------------------------------------------------------------------------------+
| Source |         Cause          |           Rx           |           Drop           |
|        |                        |     cur    |    delta  |     cur    |    delta    |
+-------------------------------------------------------------------------------------+
   UKNWN   UNKNOWN                          918          918          918          918
   INMIR   ARP MIRROR                      1108         1108            0            0
   INMIR   ISIS-L2                       501736       501736            0            0
    LPTS   ICMP IPv4                        137          137            0            0
    LPTS   PIM IPv4                      576381       576381            0            0
    LPTS   UDP SRC PORT LISP I                1            1            0            0
    LPTS   IPv4 IGMP                        56729        56729            0            0
    LPTS   TCP default IPv4              12776015     12776015            0            0
```

Received by FED from HW

```
Switch# show platform software fed active punt ios-cause brief
Statistics for all causes

Cause  Cause Info                      Rcvd                 Dropped
-------------------------------------------------------------------------------
0      Reserved                        1                    919
5      CLNS IS-IS Control              2007896              0
7      ARP request or response         2947                 0
11     For-us data                     16154496             0
24     Glean adjacency                 1                    0
55     For-us control                  402749               0
58     Layer2 bridge domain data pack  230577               0
96     Layer2 control protocols        542180               0
```

Sent by FED to IOSd

Forwarding Engine Driver

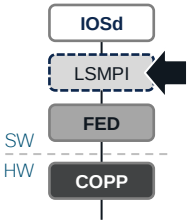



```
Switch# debug platform software fed active punt packet-capture start
Punt packet capturing started.
```

```
Switch# debug platform software fed active punt packet-capture stop
Punt packet capturing stopped. Captured 30 packet(s)
```

```
Switch# show platform software fed active punt packet-capture brief
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far :    30 packet(s)
Capture capacity      :  4096 packet(s)
Max. Meta header size :    88 byte(s)
Max. Packet data size :   128 byte(s)

------ Punt Packet Number: 1, Timestamp: 2024/12/16 10:43:38.526 ------
 interface : phy: HundredGigE1/0/3 [if-id: 0x000004a7], pal: HundredGigE1/0/3 [if-id: 0x000004a7]
 misc info : cause: 5 [CLNS IS-IS Control], sub-cause: 0, linktype: LAYER2 [10]
 CE    hdr : dest mac: 4e41.5000.0111, src mac: 4e41.5000.0111, ethertype: 0x7106
 meta  hdr : Nxt. Hdr: 0x1, Fwd. Hdr: 0, SSP: 0x1b
 meta  hdr : DSP: 0xffff, SLP: 0x4, DLP: 0x84
 ether hdr : dest mac: 0900.2b00.0005, src mac: 6c29.d2b2.59c6
 ether hdr : length: 1494
<snip>
```

Linux Shared Memory Punt Interface

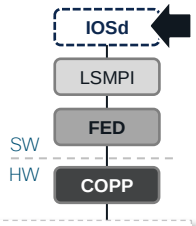



```
Switch# show platform software infrastructure lsmpi driver

LSMPI Driver stat ver: 3
Packets:
        In: 1096817
       Out: 1097135

Rings:
        RX: 4095 free    0    in-use    4096 total
        TX: 2047 free    0    in-use    2048 total
    RXDONE: 4094 free    1    in-use    4096 total
    TXDONE: 2046 free    1    in-use    2048 total

Buffers:
        RX: 8193 free    1    in-use    8194 total

Transmit fail retry: Disabled
<snip>
```

```
Reason for RX drops :
     Ring full         : 0
     Ring put failed   : 0
     No free buffer    : 0
     Receive failed    : 0
     Packet too large  : 0
     Other inst buf    : 0
     Consecutive SOPs  : 0
     No SOP or EOP     : 0
     EOP but no SOP    : 0
     Particle overrun  : 0
     Bad particle ins  : 0
     Bad buf cond      : 0
     DS rd req failed  : 0
     HT rd req failed  : 0
```

```
Reason for TX drops :
     Bad packet len    : 0
     Bad buf len       : 0
     Bad ifindex       : 0
     No device         : 0
     No skbuff         : 0
     Device xmit fail  : 0
     Device xmit rtry  : 0
     Tx Done ringfull  : 0
     Bad u->k xlation  : 0
     No extra skbuff   : 0
     Consecutive SOPs  : 0
     No SOP or EOP     : 0
     EOP but no SOP    : 0
     Particle overrun  : 0
     Other inst buf    : 0
```

IOSd (App level)




```
Switch# debug ip packet detail
Switch# debug arp
Switch# debug bgp …
```

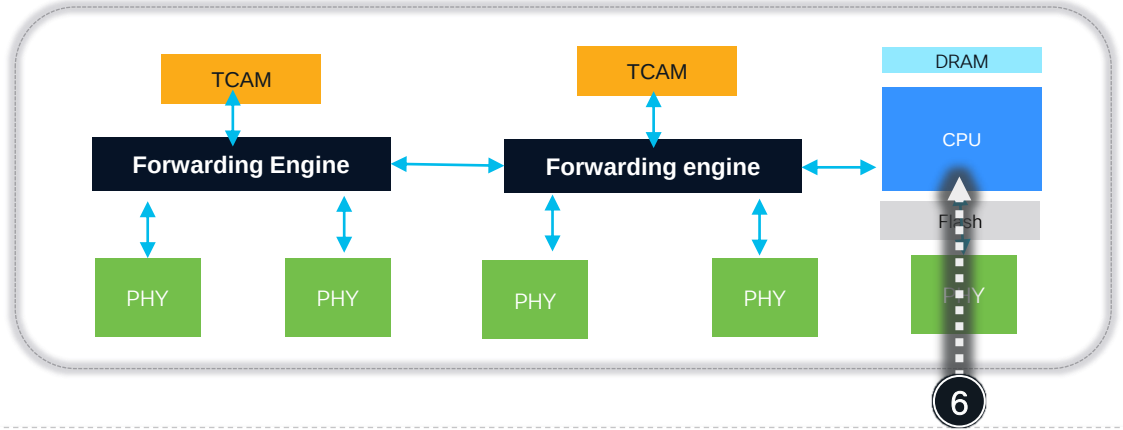
```
Switch# monitor capture CPU control-plane both match any start
```

```
Switch# monitor capture CPU stop
```

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

    1   0.000000 6c:29:d2:b2:59:c6 -> 09:00:2b:00:00:05 ISIS HELLO 1508 P2P HELLO, System-ID: 0100.9800.4001
    2   0.508553 6c:29:d2:93:6d:46 -> 09:00:2b:00:00:05 ISIS HELLO 1508 P2P HELLO, System-ID: 0100.9800.4010
    3   0.617116 192.168.40.197 -> 224.0.0.13   PIMv2 72 Hello
    4   0.685620 192.168.40.3 -> 100.64.0.1   Syslog 120 LOCAL7.INFO: 65477: *Dec 16 10:53:56.213: %BUFCAP-6-ENABLE: Capture
Point CPU enabled.
    5   1.010079 34:1b:2d:76:fc:01 -> 01:00:0c:cc:cc:cd STP 64 RST. Root = 32768/1/34:1b:2d:76:fc:00  Cost = 0  Port =
0x8001
```

```
Switch# monitor capture CPU export location flash:capture_cpu.pcap
Export Started Successfully
```

6: Management forwarding

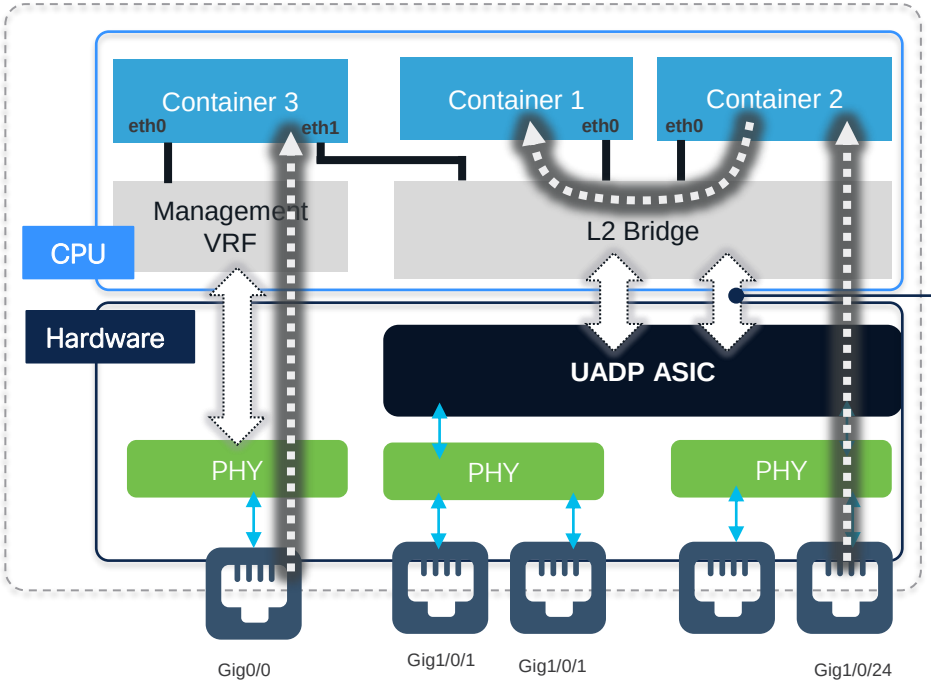


- Traffic from GigabitEthernet0/0 is handled directly by CPU bypassing the regular FE/FED Path
- Previous capture tools will not apply

```
Switch# debug ip packet detail
```

```
*Dec 16 11:02:15.938: FIBipv4-packet-proc: route packet from (local) src 10.62.97.92 dst 10.209.0.209
*Dec 16 11:02:15.938: FIBfwd-proc: packet routed by adj to GigabitEthernet0/0 10.62.97.1
*Dec 16 11:02:15.938: FIBipv4-packet-proc: packet routing succeeded
*Dec 16 11:02:15.938: IP: tableid=1, s=10.62.97.92 (local), d=10.209.0.209 (GigabitEthernet0/0) nexthop=10.62.97.1, routed
via FIB
*Dec 16 11:02:15.938: IP: s=10.62.97.92 (local), d=10.209.0.209 (GigabitEthernet0/0), len 600, sending
*Dec 16 11:02:15.938:     TCP src=22, dst=59184, seq=1270792680, ack=2671593600, win=32726 ACK
*Dec 16 11:02:15.938: IP: s=10.62.97.92 (local), d=10.209.0.209 (GigabitEthernet0/0), len 600, sending full packet
*Dec 16 11:02:15.938:     TCP src=22, dst=59184, seq=1270792680, ack=2671593600, win=32726 ACK
```

7: App-hosting forwarding



3 sub cases:

a) Container ⟷ Front panel port
b) Container ⟷ Management port
c) Container ⟷ Container

```
Switch# show int Ap1/0/1
AppGigabitEthernet1/0/1 is Up, line protocol is UP
  Hardware is App-hosting Gigabit Ethernet, address is
6c29.d2b2.59a9
  MTU 9198 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
(snip>
```

Note: C9300X, C9400X-Sup-2/2XL and C9500X models have 2 x 10G of AppGigabitEthernet ports.

7a: Container <-> Front panel port

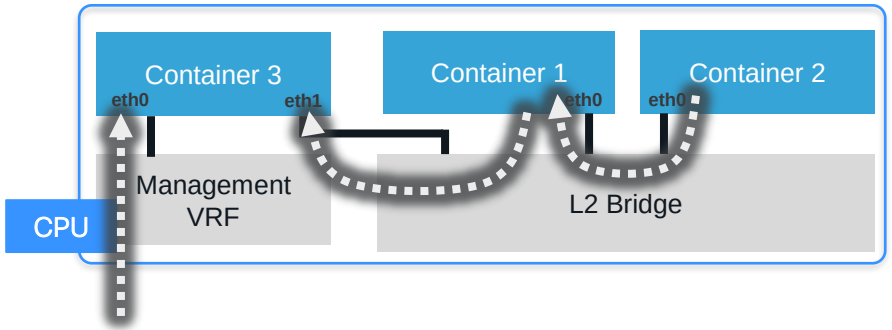
```
Switch# monitor capture APP interface AppGigabitEthernet 1/0/1 both match any start
```

```
Switch# app-hosting connect appid TE_Agent session bash
Switch:/# ping 100.1.1.200
PING 100.1.1.200 (100.1.1.200): 56 data bytes
64 bytes from 100.1.1.200: seq=0 ttl=254 time=0.581 ms
64 bytes from 100.1.1.200: seq=1 ttl=254 time=0.659 ms
64 bytes from 100.1.1.200: seq=2 ttl=254 time=0.618 ms
^C
Switch:/# exit
```

```
Switch# monitor capture APP stop
Capture statistics collected at software:
        Capture duration - 42 seconds
        Packets received - 182
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0
Capture buffer will exists till exported or cleared
Stopped capture point : APP
```

```
Switch# show monitor capture APP buffer | i ICMP
    9   2.282212  100.1.1.200 -> 100.1.1.150  ICMP 118 Echo (ping) request  id=0x0006, seq=0/0, ttl=254
   10   2.282255  100.1.1.150 -> 100.1.1.200  ICMP 118 Echo (ping) reply    id=0x0006, seq=0/0, ttl=64 (request in 9)
   11   2.282724  100.1.1.200 -> 100.1.1.150  ICMP 118 Echo (ping) request  id=0x0006, seq=1/256, ttl=254
   12   2.282758  100.1.1.150 -> 100.1.1.200  ICMP 118 Echo (ping) reply    id=0x0006, seq=1/256, ttl=64 (request in 11)
   13   2.283134  100.1.1.200 -> 100.1.1.150  ICMP 118 Echo (ping) request  id=0x0006, seq=2/512, ttl=254
   14   2.283150  100.1.1.150 -> 100.1.1.200  ICMP 118 Echo (ping) reply    id=0x0006, seq=2/512, ttl=64 (request in 13)
   15   2.283511  100.1.1.200 -> 100.1.1.150  ICMP 118 Echo (ping) request  id=0x0006, seq=3/768, ttl=254
```

7b&c: Container <-> Management port/Container

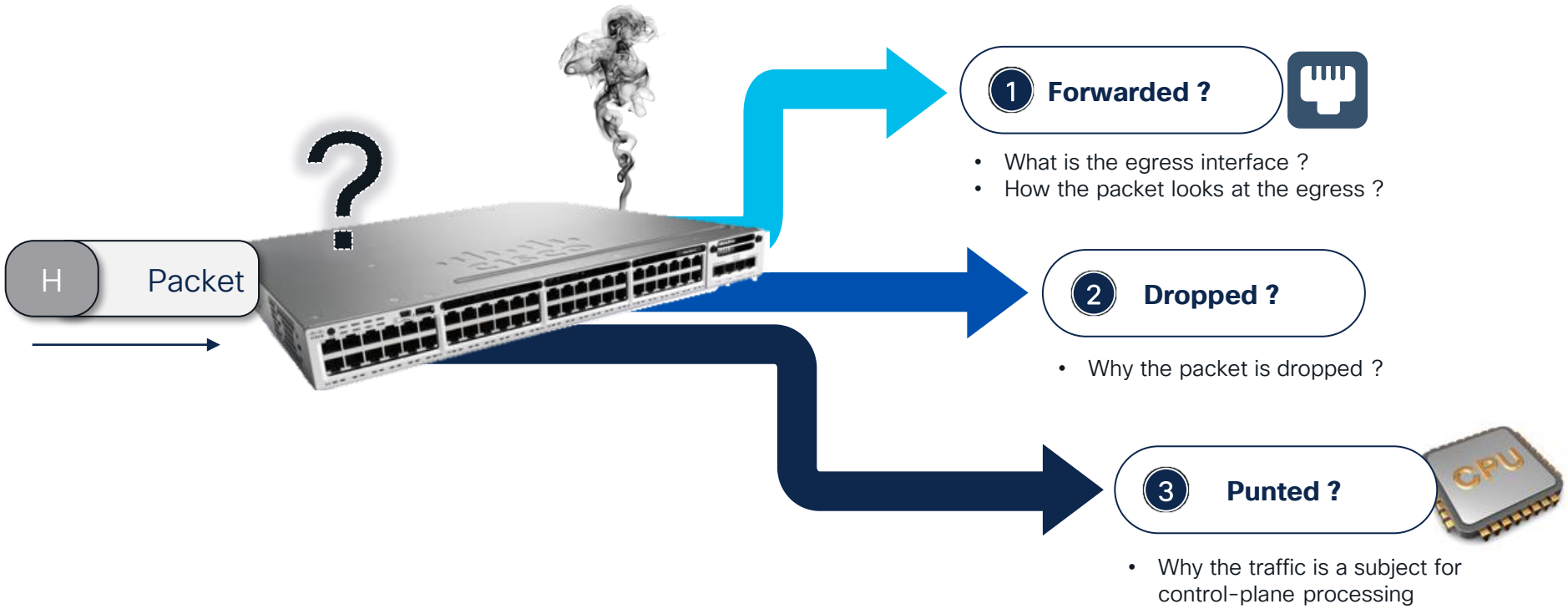


Only inner container tools available
- typically **tcpdump**

```
Switch# app-hosting connect appid TE_Agent session bash
Switch :/# tcpdump  icmp -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
15:24:05.891595 IP 100.1.1.200 > 100.1.1.151: ICMP echo request, id 34, seq 0, length 80
15:24:05.891614 IP 100.1.1.151 > 100.1.1.200: ICMP echo reply, id 34, seq 0, length 80
15:24:05.892001 IP 100.1.1.200 > 100.1.1.151: ICMP echo request, id 34, seq 1, length 80
15:24:05.892005 IP 100.1.1.151 > 100.1.1.200: ICMP echo reply, id 34, seq 1, length 80
15:24:05.892348 IP 100.1.1.200 > 100.1.1.151: ICMP echo request, id 34, seq 2, length 80
15:24:05.892351 IP 100.1.1.151 > 100.1.1.200: ICMP echo reply, id 34, seq 2, length 80
15:24:05.892667 IP 100.1.1.200 > 100.1.1.151: ICMP echo request, id 34, seq 3, length 80
15:24:05.892670 IP 100.1.1.151 > 100.1.1.200: ICMP echo reply, id 34, seq 3, length 80
```

Forwarding verification tools

How to track forwarding decisions ?



My packet is addressed to 192.168.40.1

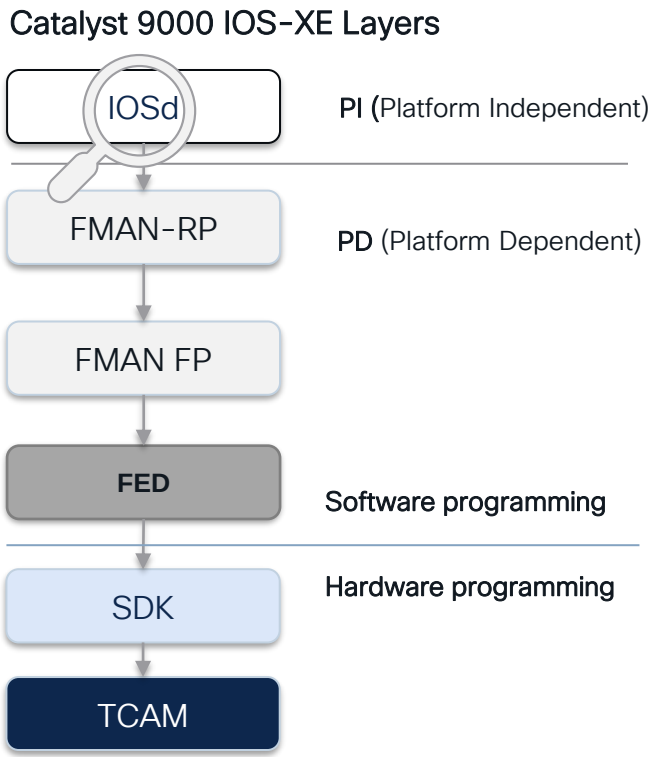
How will it egress my switch ?

```
# show ip route 192.168.40.1
Routing entry for 192.168.40.1/32
  Known via "isis", distance 115, metric 30, type level-2
  Redistributing via isis
  Last update from 192.168.40.222 on Vlan412, 4d19h ago
  Routing Descriptor Blocks:
  * 192.168.40.222, from 192.168.40.1, 4d19h ago, via Vlan412
      Route metric is 30, traffic share count is 1
```

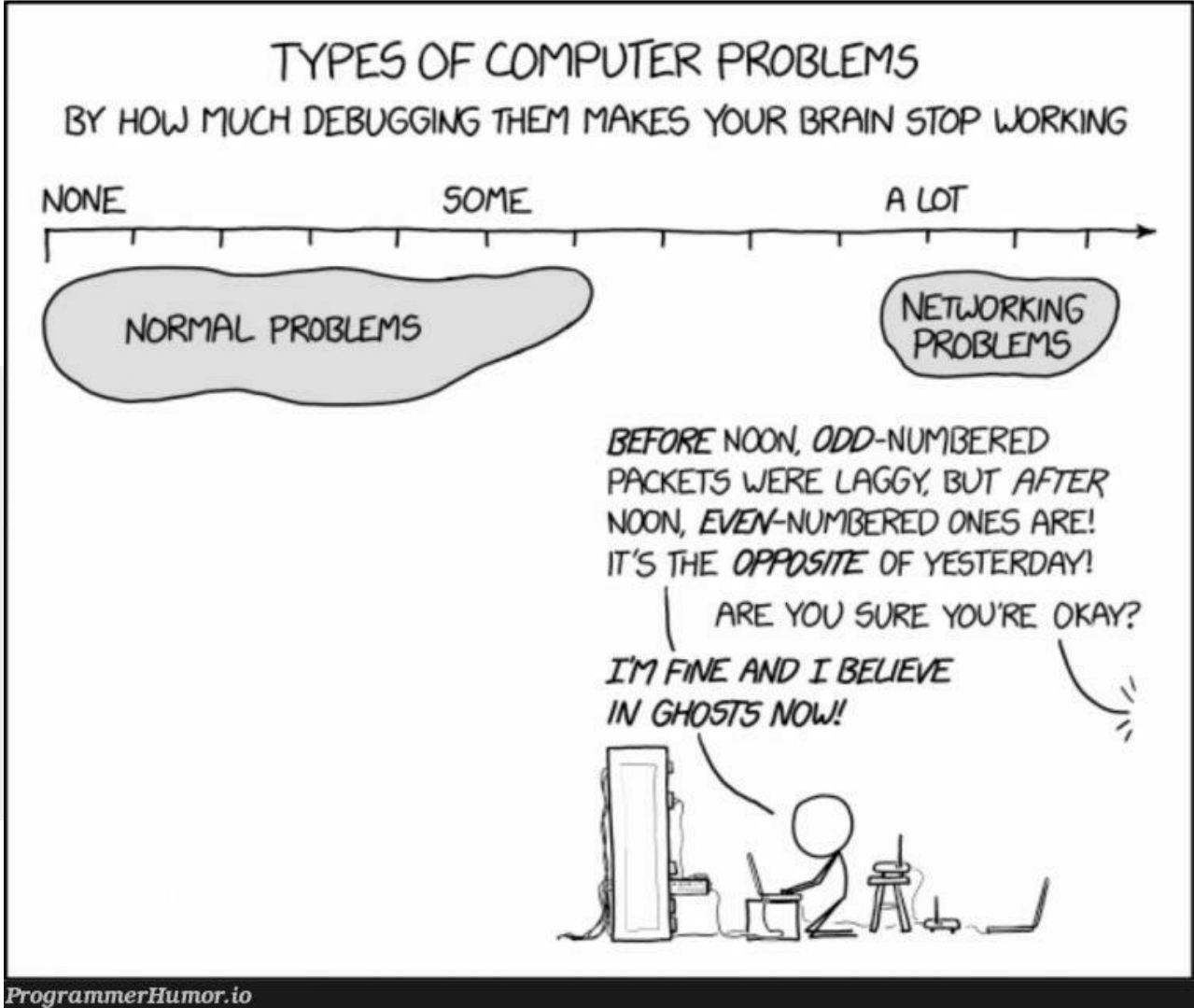
```
# show ip arp 192.168.40.222
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.40.222         174  6c29.d289.a15b  ARPA   Vlan412
```

```
# show mac address-table address 6c29.d289.a15b
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 412    6c29.d289.a15b    DYNAMIC     Hu1/0/1
```

Will my packet egress out via the Hu1/0/1 ?



When do we need to use forwarding decision tracking tools?



Forwarding Tracking Tools

Catalyst 9000 Family

Simulates the arrival of a requested packet at a specified interface by the CPU. Based on the current state of the TCAM, it determines how the switch would process the traffic if it were received on the simulated interface. This is a software-based solution

UADP 2.0
UADP 3.0
Q200

CPU Simulation

SPF
Show Platform Forward

```
# show platform hardware fed active forward …
```

Triggered by the first packet that meets the specified conditions, it collects forwarding data from the lookup stages without impacting live traffic. This solution relies on ASIC support and is hardware-based

Live Packet Tracing

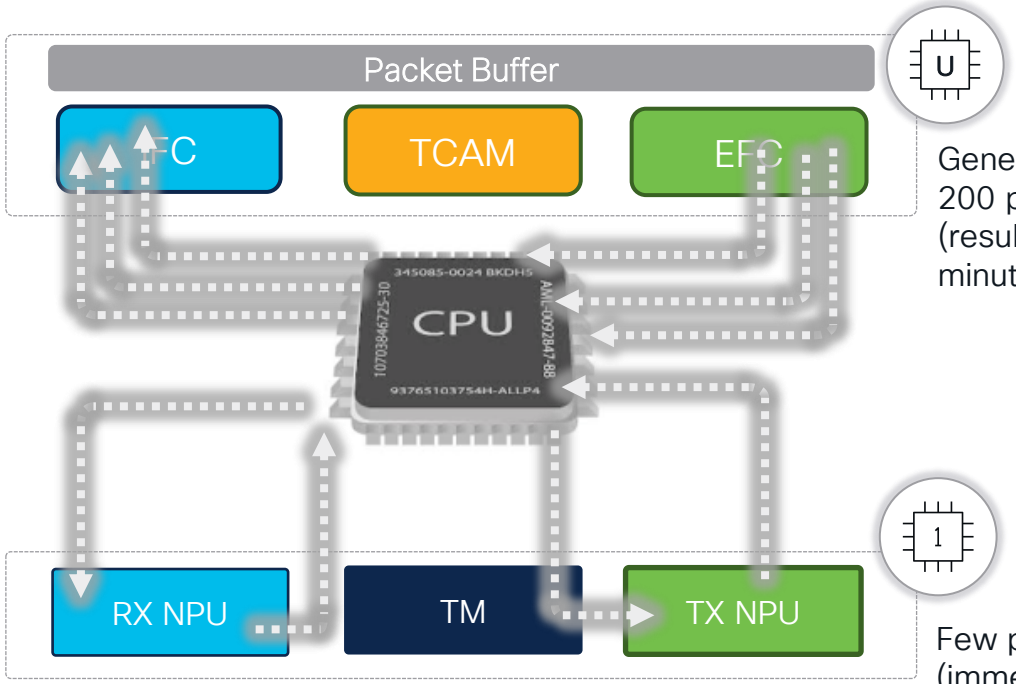
UADP 3.0

PSV
Packet State Vector

```
# debug platform hardware fed active capture …
```

Forwarding tracing tools depend on the actual content of TCAM memory not only on the PI layer

Show Platform Forward



Generates ~150-200 packets (results in minutes)

Few packets (immediate results)

1. **Works on all UADP and S1 based platforms**
2. **Offline mode**
3. **Typically generates multiple packets to capture each stage**
4. **The packets generated in SPF don't get forwarded**
5. **Show forward cannot indicate QOS/Policer drops**
6. **Input manual or in pcap format**
7. **Adjusted to advanced forwarding scenarios**

Show Platform Forward

Manual Header Definition

1. Define packet header. Specify L2/L3/L3 headers

UADP & Silicon One

Incoming interface

```
show platform hardware fed switch [1|2|..] forward interface GigabitEthernet 1/0/22
0011.9267.b370 58bf.eab6.7fe2 ipv4 10.200.1.100 10.201.1.100  tcp 65000 80 0
```

L2 SRC | L2 DST | L3 SRC | L3 DST | L4

~ 2min

2. Wait for results. Time depends on the type of used encapsulation

UADP

```
*Jun 12 10:50:49.075: %SHFWD-6-PACKET_TRACE_DONE:Switch 1 R0/0: fed:
Show fwd is completed. The capture file can be found at /flash/shfwd+timestamp.log
(ie. shfwdxxxxxx-xxxxxx.log) .
```

3. Verify forwarding results

UADP & Silicon One

```
show platform hardware  fed switch [1|2|..] forward last summary
show platform hardware  fed switch [1|2|..] forward last detail
```

Show Platform Forward

Leveraging EPC

1. Define EPC filters and start capture

```
monitor capture TAC interface Gig 1/0/2 in match any
monitor capture TAC start
monitor capture TAC stop
Capture statistics collected at software:
        Capture duration - 8 seconds
        Packets received - 11
        Packets dropped - 0
        Packets oversized - 0
Capture buffer will exists till exported or cleared
Stopped capture point : TAC
```

2. Verify Content of EPC buffer and find the interesting frame

```
show monitor capture TAC buffer
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

1 0.000000 192.168.100.100 -> 192.168.100.1 ICMP 114 Echo (ping) request
2 0.000011 192.168.100.1 -> 192.168.100.100 ICMP 114 Echo (ping) reply
```

3. Export EPC buffer to a file

```
monitor capture TAC export location flash:capture2.pcap
```

4. Use the pcap as the input for SPF

```
show platform hardware  fed switch 1 forward interface GigabitEthernet 1/0/2
pcap flash:capture2.pcap number 2 data
```

Show Platform Forward (UADP)

UADP

Last Summary results

```
show platform hardware fed switch [1|2|..] forward last summary
```

1 H Packet

Input packet

```
###[ Ethernet ]###
  dst             = a0:f8:49:0e:5a:83
  src             = 00:17:94:61:7d:50
  type            = 0x800
###[ IP ]###
     version       = 4L
     ihl           = 5L
     tos           = 0x0
     len           = 28
     id            = 1
     flags         =
     frag          = 0L
     ttl           = 64
     proto         = icmp
     chksum        = 0xa5a9
     src           = 10.10.1.100
     dst           = 200.200.1.1
     options   = ''
###[ ICMP ]###
        type       = echo-reply
        code       = 0
        chksum = 0xffff
        id         = 0x0
        seq        = 0x0
```

2 IFC

Ingress resolution

```
Port                        : GigabitEthernet2/0/1
   Global Port Number       : 97
   Local Port Number                : 1
   Asic Port Number                 : 0
   Asic Instance                    : 1
  STP Instance                      : 2
   BlockForward                     : 0
   BlockLearn                       : 0
   L3 Interface                     : 36
       IPv4 Routing                 : enabled
       IPv6 Routing                 : enabled
       Vrf Id                       : 0
   Adjacency:
       Station Index                : 174
       Destination Index            : 21083
       Rewrite Index                : 2
       Replication Bit Map          : 0x4
['localData']
   Decision:
       Destination Index            : 21083
       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
```

3 EFC

Egress resolution and rewrite

```
Egress:
   Possible Replication             :
       Port                         :
GigabitEthernet2/0/3
   Output Port Data                 :
     Port                           :
GigabitEthernet2/0/3
       Global Port Number           : 99
       Local Port Number            : 3
       Asic Port Number             : 2
       Asic Instance                : 1
       Unique RI                    : 2
       Rewrite Type                 : 1
                                    [L2_BRIDGE]
       Mapped Rewrite Type          : 4
                                [L2_BRIDGE_INNER_IPv4]
       Vlan                         : 1
       Mapped Vlan ID               : 4
```

L2DESTMACVLAN

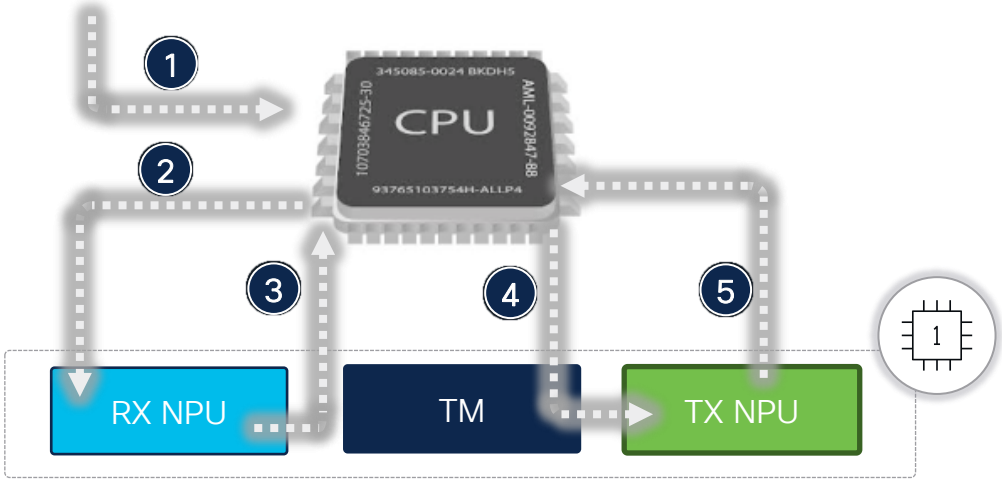
L3FWDIPV4

PBRIPV4

```
 Port          : GigabitEthernet2/0/3
###[ Ethernet ]###
  dst             = a0:f8:49:0e:5a:83
  src             = 00:17:94:61:7d:50
  type            = 0x800
###[ IP ]###
     version       = 4L
     ihl           = 5L
     tos           = 0x0
     len           = 28
     id            = 1
     flags         =
     frag          = 0L
     ttl           = 64
     proto         = icmp
     chksum         = 0xa5a9
     src           = 10.10.1.100
     dst           = 200.200.1.1
     options   = ''
###[ ICMP ]###
        type       = echo-reply
        code       = 0
        chksum     = 0xffff
        id         = 0x0
        seq        = 0x0
```

Show Platform Forward

SiliconOne Platforms



1. Simulated packet definition
2. Info on how the packet is injected in NPU RX
3. NPU RX Forwarding results (Ingress resolution)
4. Info on how the packet is injected in NPU TX
5. NPU TX Forwarding results (Egress resolution + Rewrite Info)

By default only stages 1,3,5 presented

```
sh plat hardw fed act forward last summary
```

In detail mode all 5 stages presented

```
sh plat hardw fed act forward last detail
```

Results available immediately - no need to wait

Last 16 results stored

```
sh plat hardw fed act forward last list
```

Available since 17.15.1

Understanding the outputs

By default, the outputs present the most relevant information to help understand the forwarding decision:

1 H Packet

Input packet

```
Packet-trace Id: spf1284569044
   Input packet:
      ###[ Ethernet ]###
        dst       = 98:a2:c0:7e:35:02
        src       = 20:20:30:30:40:40
        type      = IPv4
      ###[ IP ]###
           version   = 4
           ihl       = 5
           tos       = 0x0
           len       = 28
           id        = 1
           flags     =
           frag      = 0
           ttl       = 64
           proto     = icmp
           chksum    = 0xa831
           src       = 192.168.40.141
           dst       = 192.168.40.209
           \options   \
      ###[ ICMP ]###
              type      = echo-reply
              code      = 0
              chksum    = 0xffff
              id        = 0x0
              seq       = 0x0
              unused    = ''
```

3 RX NPU

Ingress resolution

```
RX (Ingress):
   Ingress Interface : HundredGigE1/0/4
   traces:
      - # 1

    Decision:
       #
       # dsp: HundredGigE1/0/2
       # fwd_hdr_type: ipv4
       # fwd_relay_id_or_pwe_id: '0x0'
       # l3_dlp_id: 0x8 (Vlan411)
       # rx_nw_app_or_lb_key: '0xe746'
       # tm_hdr_type: '0x1'
       #
```

5 TX NPU

Egress resolution and rewrite

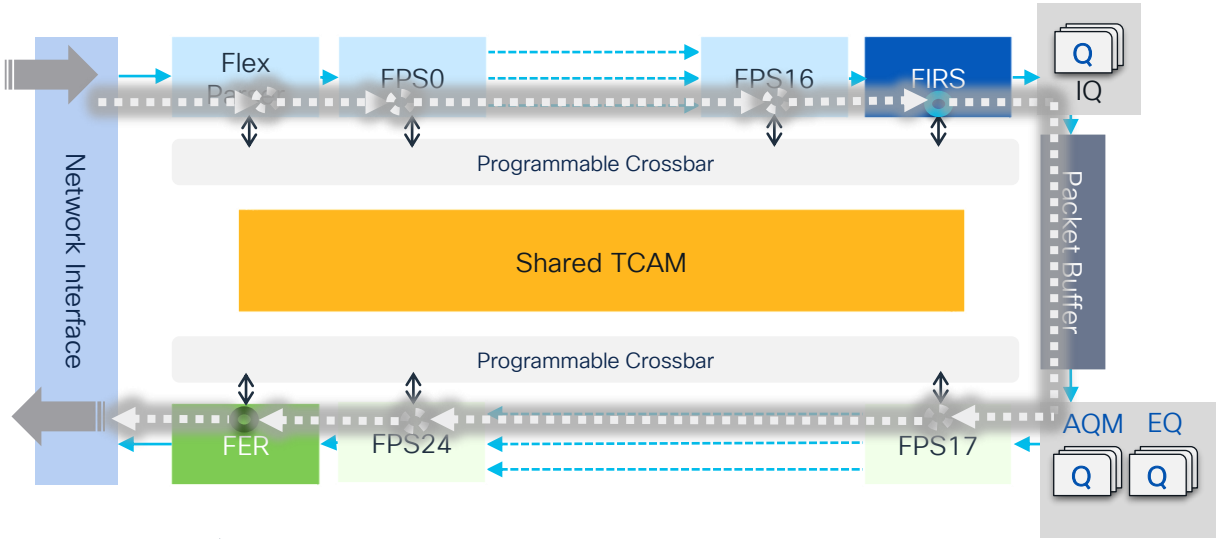
```
TX (Egress):
   traces:
      - # 1

       hierarchical view:
       #
       # code: ethernet_acl_force_punt
       # destination_sp: HundredGigE1/0/2
       # source: outbound_mirror
       #

           ###[ Ethernet ]###
             dst  = 6c:29:d2:9d:36:ee
             src  = 98:a2:c0:7e:35:02
             type      = VLAN
           ###[ 802.1Q ]###
                prio      = 0
                id        = 0
                vlan      = 411
                type      = IPv4
           ###[ IP ]###
               version   = 4
               ihl       = 5
               tos       = 0x0
                len       = 28
                id        = 1
                flags     =
                frag      = 0
                ttl       = 63
                proto     = icmp
                chksum    = 0xa931
                src       = 192.168.40.141
                dst       = 192.168.40.209
                \options   \
           ###[ ICMP ]###
                   type      = echo-reply
                   code      = 0
                   chksum    = 0xffff
                   id        = 0x0
                   seq       = 0x0
                   unused    = ''
           ###[ Padding ]###
                   load      =
0000000000000000000000000000
```

Packet State Vector

UADP 3.0

Catalyst 9500 High Performance

Catalyst 9600



- Allows for tracing packet flow in UADP 3.0 ASIC
- Uses the live traffic received by the switch
- Captures first packet which matches the capture criteria
- Independent of any feature interaction
- No effect on the switch functionality

Configuring PSV Debug Capture Trace

1. Define trigger. Enable mode required

```
debug platform hardware fed active capture trigger
[ipv4 | ipv6 <src><dst> [l3 protocol | icmp | igmp | sctp | tcp | tos |
udp<src_port><dst_port>]]
[layer2 [ethertype | src_mac | dst_mac]]
[if-id <if_id>ingress | egress ] [interface <ifname>ingress | egress]
[vlan <vlan-id>ingress | egress]
```

2. Start the capture

```
debug platform hardware fed active capture start
```

3. Verify status. Waiting for the first matching packet

```
show platform hardware fed active capture status
Asic: 0 Status: Running

<..packet arrives..>

show platform hardware fed active capture status
Asic: 0 Status: completed
```

4. Verify status. Waiting for the first matching packet

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

PSV example

1

```
switch# enable
switch# debug platform hardware fed active capture trigger ipv4 10.0.0.1 20.0.0.2 icmp
Capture trigger set successful.
```

```
switch# show platform hardware fed active capture trigger
Trigger Set:
Ether Type: 0x0800
Dest IP: 20.0.0.2
Src IP: 10.0.0.1
Protocol: 0x1
```

2

```
switch# debug platform hardware fed active capture start
Packet Capturing Started.
```

3

```
switch# show platform hardware fed active capture status
Asic: 0 Status: Running
```

SPF vs PSV

Show Platform Forward

PROS

- Available on all platforms
- Support for advanced forwarding scenarios
- Unified syntax on all platforms
- Ability to use PCAP as input

CONS

- CPU simulation
- Awaiting time (for UADP platforms)

Packet State Vector

PROS

- Live packet tracing
- High reliability
- Instant results

CONS

- Only manual trigger
- Supported only by UADP3.0
- No support for advanced triggers

Packet tracing tools are your best friends

Wrap-Up

Take away:

Importance of Packet Journey Analysis

- Helps evaluate performance and identify bottlenecks.
- Facilitates investigation of forwarding and drop issues.

Data Plane vs. Control Plane

- Distinction between the **Data Plane (ASIC)** and **Control Plane (CPU)**, each with multiple sub-paths.
- Data Plane based on **UADP** or **S1** Asics

Catalyst 9000 as a campus networking platform

- Forms the foundation for both legacy and modern campus designs.
- Understanding the **packet journey** is essential for supporting diverse applications and use cases.

Built-In Tools for Packet Path Analysis

- **Statistics, counters and drop exceptions** to analyze packet paths and drop scenarios.
- **Packet Tracing Tools** are essential for hardware forwarding decision verification.

"What we understand, we control. What we explore, we master."

–Vernor Vinge

Fill Out Your Session Surveys

Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)

All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the ‘Participant Dashboard’

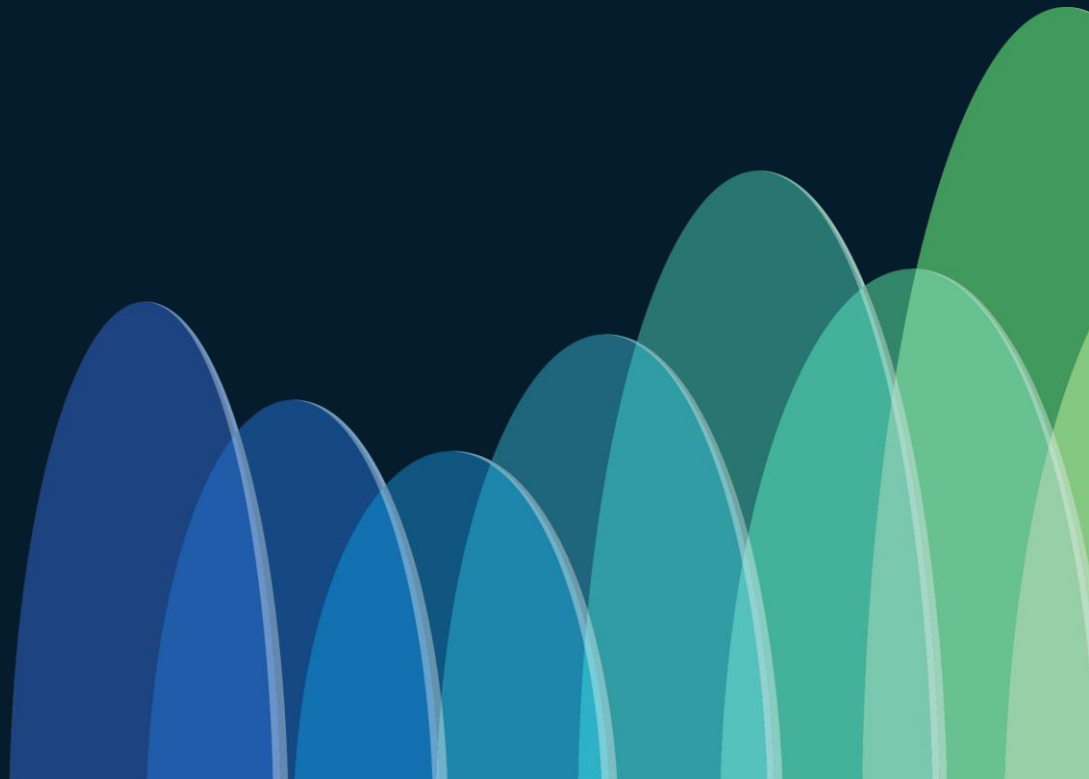
Content Catalog

Continue your education

- Visit the On-Demand Library for more sessions at ciscolive.com/on-demand. Sessions from this event will be available from March 3.

Contact us at: ishirshi@cisco.com, natpan@cisco.com

Q&A




cisco
Thank you
cisco Live!