TOMORROW starts here.

ASR-9000/IOS-XR hardware Architecture, QOS, EVC, IOS-XR Configuration and Troubleshooting

Session ID BRKSPG-2904

Xander Thuijs CCIE #6775 – Principal Engineer

High-End Routing and Optical group ASR9000

Cisco live!

Agenda

- Introduction

1. ASR9000 operation and capabilities
2. Packet flow and punt path
3. Differences between Trident and Typhoon (NPU)
4. Multicast architecture and verification/troubleshooting
5. QOS architecture
6. Troubleshooting techniques (punt path troubleshooting/architecture)
7. IOS-XR differences to legacy IOS
8. Mapping IOS to XR configurations (eg EVC infrastructure)

- Summary

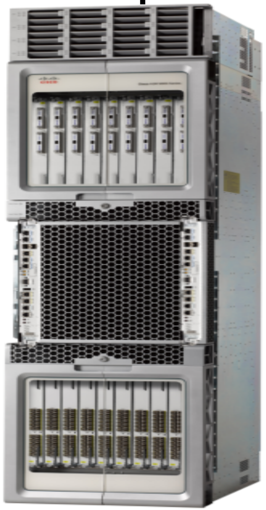
ASR 9K Chassis Overview

240 Gbps

3.5 Tbps

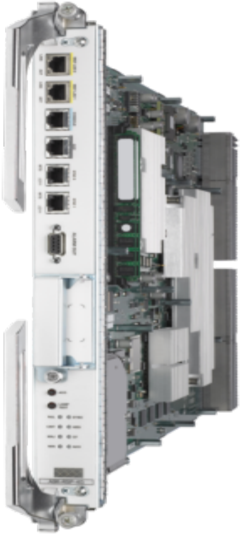
7 Tbps

48 Tbps



	ASR 9001 (Ironman)	ASR 9006	ASR 9010	ASR 9922 (Megatron)
Max Capacity (bi-directional)	120Gbps	440G/slot 4 I/O slots	440G/slot 8 I/O slots	1.2T/slot 20 I/O slot
Size	2RU	10RU	21RU	44RU
Max Power	750W	6KW	9KW	24KW
Air Flow	Side to side	Side to back	Front to back	Front to back
FCS	**4.2.1 release**	Shipping	Shipping	**4.2.2 release**

ASR 9K RSP (Route/Switch Processors)



	Current RSP2	RSP440
Processors	2 x 1.5GHz Freescale 8641D CPU	Intel x86 Jasper Forest 4 Core 2.27 GHz
RAM (user expandable)	4GB @133MHz SDR 8GB	**6GB (RSP440-TR) and 12GB (RSP440-SE)** version @1066MHz DDR3
Cache	L1: 32KB L2: 1MB	L1: 32KB per Core L2: 8MB shared
Primary persistent storage	4GB disk0/1, primary boot, mirror can be disabled	16GB - SDD
Secondary persistent storage (HD/SSD)	30GB – HDD Logging and crash dumps	16GB - SDD
USB 2.0 port HW assisted CPU queues	No No	Yes, can boot from rommon mediaboot usb:/file Yes
nV Cluster – EOBC ports	No	Yes, **2 x 1G**/10G SFP+
Switch fabric bandwidth	184G/slot (with dual RSP)	440G/slot (with dual RSP)

RSP440

RSP440 – Front Ports

1G/10G SFP+
EOBC ports for nV Cluster

Note, red color is the new front ports, which is supported on RSP440 only, not RSP2

USB Type A

LEDs
Status, Alarm

Aux

IEEE 1588, GPS
SyncE, IEEE1588 master and slave
10/100M Copper Ethernet

Console

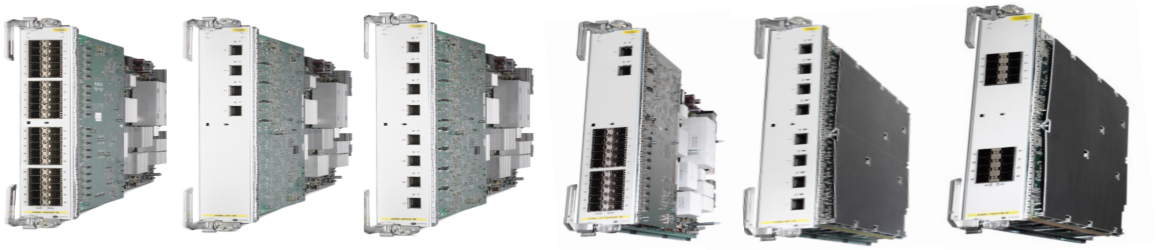
BITS/J.211
Sync 0, Sync 1
RJ45

Management Ethernet

ASR 9K Ethernet Line Card Overview

First-generation LC
(Trident NP)

-L, -B, -E

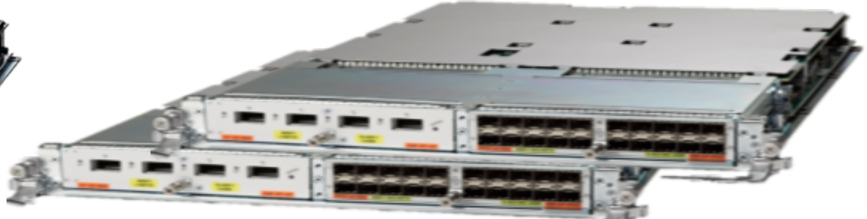


A9K-40G | A9K-4T | A9K-8T/4 | A9K-2T20G | A9K-8T | A9K-16T/8

Second-generation LC (Typhoon NP)

-TR, -SE

A9K-24x10GE
A9K-2x100GE



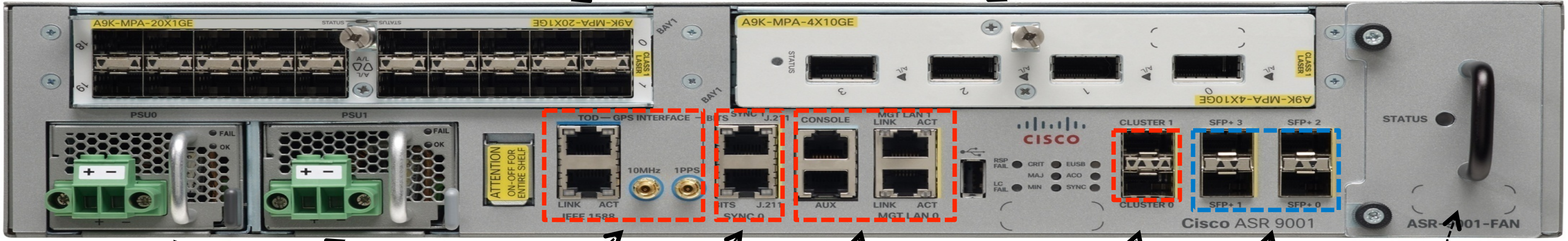
A9K-MOD80
A9K-MOD160

MPAs
20x1GE
2x10GE
4x10GE
1x40GE
2x40GE

A9K-36x10GE

ASR 9001 “Iron Man” Overview

Two Modular bays
Supported MPA: 20xGE, 2/4x10GE, 1x40GE (4.3.0)



Redundant (AC or DC) Power Supplies Field Replaceable

GPS, 1588

BITS

Console, Aux, Management

EOBC ports for nV Cluster (2xSFP+)

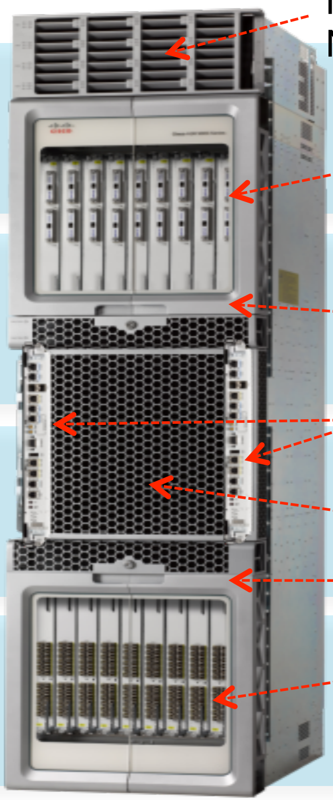
Fixed 4x10G SFP+ ports

Fan Tray Field Replaceable

Note, 2x40GE MPA is not supported on Iron man system

New ASR 9922 "Megatron" System

Slots	• 20 Line Card Slots • 2 dedicated RP slots • multi-plane, multi-stage fabric • N:1 Switch Fabric Redundancy
Dimensions	• Height : 44 RU (AC & DC) • Depth : 30.0" (800mm) • Width : 17.75" (fits 19" rack)
Power	• AC & DC power supplies • Pay As You Grow Modular Power • 24KW max power, ~30W per 10GE
Bandwidth	• efficient, scalable fabric silicon • 550G w/ 4+1 fabric @ FCS • 770G w/ 6+1 fabric post-FCS • higher BW fabrics in development



New HW PID and Target Release

Part Number	Target Release
ASR 9001	4.2.1
ASR 9000v	4.2.1
ASR 9922	4.2.2
A9K-24x10GE-SE	4.2.0
A9K-24x10GE-TR	4.2.0
A9K-2x100GE-SE	4.2.0
A9K-2x100GE-TR	4.2.0
A9K-36x10GE-SE	4.2.2
A9K-36x10GE-TR	4.2.2

Part Number	Target Release
A9K-RSP440-SE	4.2.0
A9K-RSP440-TR	4.2.0
A9K-MOD80-SE	4.2.0
A9K-MOD80-TR	4.2.0
A9K-MOD160-SE	4.2.1
A9K-MOD160-TR	4.2.1
A9K-MPA-2x10GE	4.2.1
A9K-MPA-4x10GE	4.2.0
A9K-MPA-20x1GE	4.2.0
A9K-MPA-1x40GE	4.3.0
A9K-MPA-2x40GE	4.2.1

HW Ready Typhoon "Only" Features

Feature	Trident	Typhoon
nV Cluster (also requires RSP440)	N	Y
nV Satellite (Fabric Port) (also requires RSP440)	N	Y
BNG (Subscriber Awareness)	N	Y
SP WiFi	N	Y
MPLS-TP	N	Y
1588v2 (PTP)	N	Y
Advanced Vidmon (MDI, RTP metric)	N	Y
PBB-VPLS	N	Y
IPv6 Enhancement (ABF, LI, SLA, oGRE)	N	Y
PW-HE	N	Y
E-VPN/ PBB-EVPN	N	Y
Scale ACL	N	Y

Typhoon Scale v/s Trident

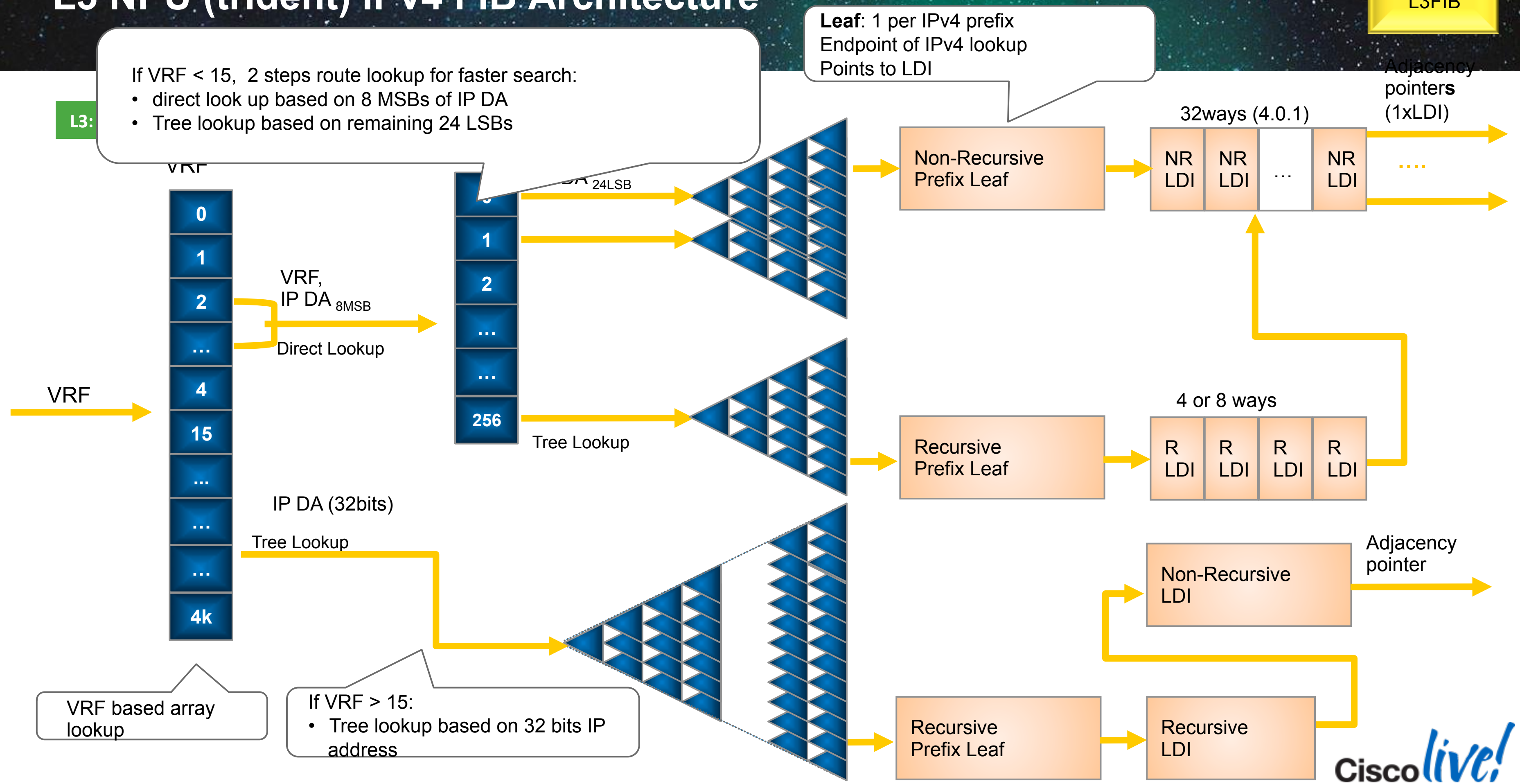
Metric	Trident	Typhoon (TR/SE)
FIB Routes (v4/v6)	1.3M/650K	4M/2M
Multicast FIB	32K	128K
MAC Addresses	512K	2M
L3 VRFs	4K	8K
Bridge Domains / VFI	8K	64K
PW	64K	128K
L3 Subif / LC	4K	8K (TR) 20K (SE)
L2 Interfaces (EFPs) / LC	4K (-L) 32K (-E)	16K (TR) 64K (SE)
MPLS labels	256K	1M
IGP Routes	20K	40K
BGP Load balancing	8-way	32-way

Route scale shared by v4 and v6:

Formula 2xIPv6 + IPv4 = credits

See via google the asr9000 route scale architecture (trident has subtrees that impose some limits)

L3 NPU (trident) IPv4 FIB Architecture



Typhoon QoS Scale Vs. Trident

Feature	Trident	Typhoon
Queue Scale	32K egress + 32K ingress for 10GE line cards 64K egress + 32K ingress for 40x1GE line cards	192K egress + 64K ingress
Policer scale	64K per NP (-E cards)	256K per NP (-SE cards)
Buffer size per 10G Port (SE or E card)	150 ms	~ 226msec per port "IF" eachNP is mapped to 3x10Gports ~ 339msec per port "IF" each NP is mapped to 2x10Gports
Buffer size per 10G Port (TR or L card)	~50 ms	~ 113msec per port "IF" eachNP is mapped to 3x10Gports ~ 170msec per port "IF" each NP is mapped to 2x10Gports
Minimal queue/police bandwidth	64 Kbps Granularity 64k	64 Kbps Granularity 8k

Google: asr9000 quality of service architecture

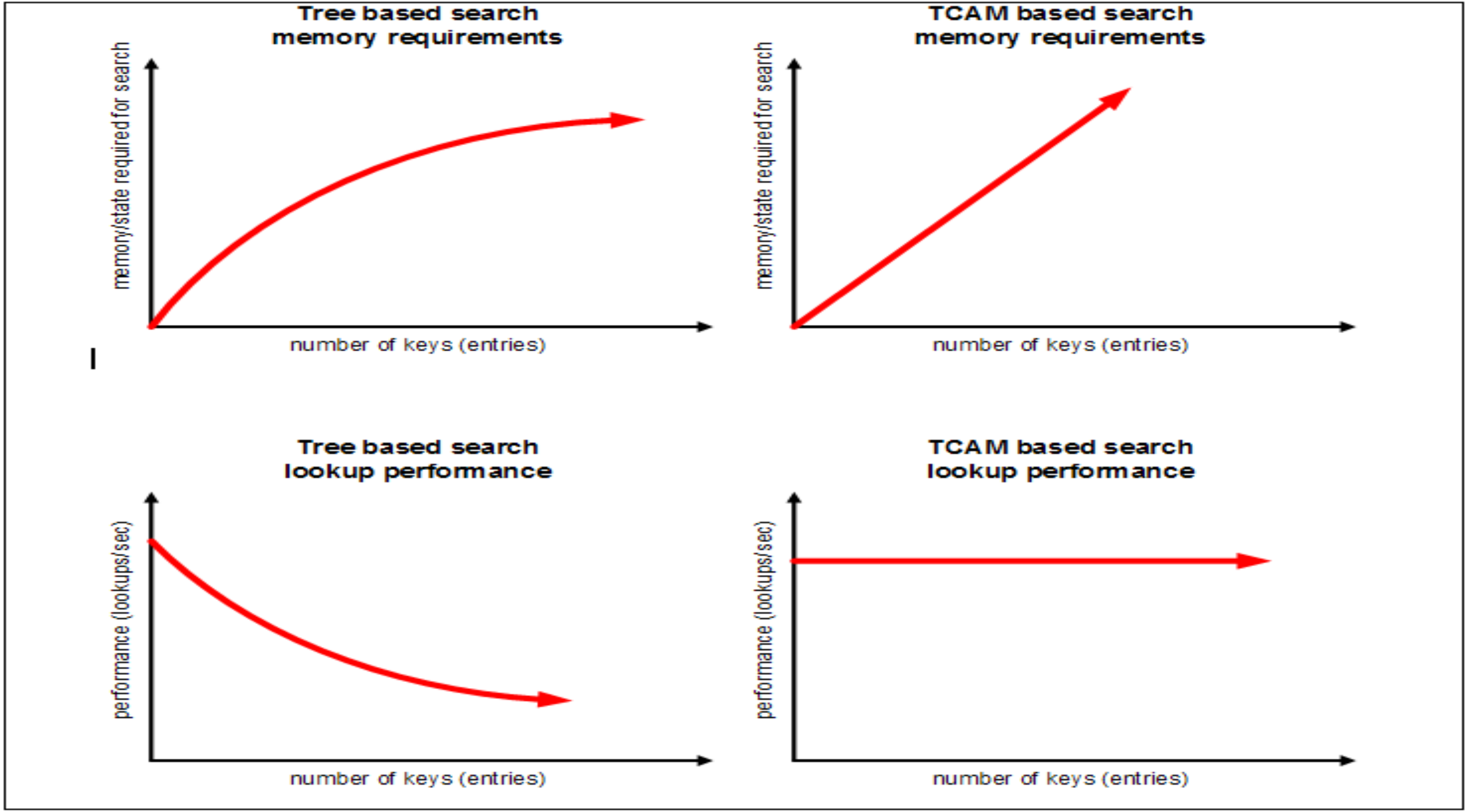
What's the Difference Between "-SE" and "-TR"?

Feature	-TR	-SE	Comments	
FIB (V4+V6)	4M		V4 and V6 share the same table V6 uses two FIB entries Support per-VRF FIB table download per LC	System wide scale
Multicast FIB	128K			
MAC	2M		Support per-LC MAC learning in the future	
L3 VRF	4K		8K in 4.2.1	
BD/VFI	64K			
PW	128K			
L3 interface	8K/LC	20K/LC		Per-LC scale
L2 interface	16K/LC	64K/LC		
QoS	8 queues/port (I and O) 8K policers/NP 1G frame memory/ NP	256K queues (I+O) / NP 256K policers/NP 2G frame memory/NP		
ACL*	24k ACE	96k ACE	10k ACL, compression supported XR4.3.1 ACL max 64k ACE (to be changed!)	

Scaled ACL problem statement:

- Provide a solution that can do ACL filtering for exceptionally large rulesets at high packet rates, within hardware (cost/power/space) constraints that makes it affordable/ deployable, with low variability in performance.
- ***Hot Tip: This is really #^(&ing hard. But we did it anyway.***
- Two part solution:

1. how do you configure really large rulesets in the management plane
2. how do you process them in the data plane?


Tree based search
memory requirements
memory/state required for search
number of keys (entries)
TCAM based search
memory requirements
memory/state required for search
number of keys (entries)
Tree based search
lookup performance
performance (lookups/sec)
number of keys (entries)
TCAM based search
lookup performance
performance (lookups/sec)
number of keys (entries)

Configuration improvements:

```
object-group network ipv4 SRC_1
 10.10.1.0/24
 host 4.5.6.7
!
object-group network ipv4 SRC_2
 20.20.1.0/24
 host 7.8.9.10
!
object-group network ipv4 DEST_1
 30.30.0.0/16
 host 3.4.5.6
```

```
object-group network ipv4 DEST_2
 40.40.40.32/30
 host 2.3.4.5
!
object-group port PORT_1
 eq domain
 range 1024 65535
!
object-group port PORT_2
 eq 80
 range 0 1023
```

```
ipv4 access-list example
  10 permit tcp net-group SRC_1 net-group DEST_1 port-group PORTS_1
  20 permit tcp net-group SRC_2 net-group DEST_2 port-group PORTS_1
  30 permit tcp net-group SRC_1 net-group DEST_1 port-group PORTS_1
  40 permit tcp net-group SRC_2 net-group DEST_2 port-group PORTS_1
```

Data structure selection:

- Trees (tries): provide efficient memory usage, but non-deterministic (highly variable) performance.
- The number of lookups can vary a lot depending on exactly where you find the match.
- The Juniper MX solution builds the ACL rulesets into trees, which are then stored in very fast (but very small) lookup memory, and used for forwarding.

TCAMs:

- Essentially "reverse" memory that takes a lookup key and mask, and returns a result. (TCAM "rule" or "**V**alue**M**ask**R**esult")
- Always returns the result in a single memory access (i.e. "order one" lookup) – so it's really fast and very determinstic.
- BUT, TCAMs are large, dedicated hardware devices. High power, high cost, and limited to (practically) tens of thousands of rules.

test notes/observations

- security only ACL’s in 4.3.1
  - no QoS or other applications
- all ACLs on a given NPU must have same compression level
- for *very* large ACLs, it takes 10-15 seconds to commit the changes. for “normal” sized ACLs it’s not more than a couple of seconds.
- PPS performance decreases as compression level increases
- We’ve taken very large infra ACL’s from real use cases and able to fit 2.7M ACE’s into 62k TCAM entries

Compression levels

- There are three available compression levels for a scaled ACL. ("level 2" is not used/implemented at present on the asr9k...)
- level 0 simply expands the object groups and dumps into TCAM (cross product)
  – identical performance to legacy ACL
  – Benefit: more convenient configuration
- ***level 1*** compresses only the source prefix object-groups
  – smallest performance hit, but still very high scale
- ***level 3*** compresses both SRC/DEST, pfx and port groups
  – higher performance reduction, but wicked-crazy-massive scale improvements
- ***General recommendation***: use least compression that fits.
  – "more flexibility" to trade performance vs. scale vs. cost
  – do NOT forget that –SE cards have much larger TCAMs than –TR cards!!!

Scaled ACL : counters

- In the hardware, each TCAM entry points at a counter.
- Regardless of legacy vs. object-group config, each configured ACE will have one counter associated.
- Scaled ACL allows you to combine lots and lots of rules into a single ACE, which also becomes a single counter.
- IF you need more granularity in your counters, break out a separate rule (just like before, but with more flexibility)
- Still order-dependent, so use sequence numbers...

scaled ACL commands

- *show pfilter-ea fea ipv4-acl <ACL> loc <loc>*
  – shows you how many ACEs, how many TCAM entries, and TCAM entries per ACE (must be applied to see)
- *show pfilter-ea fea summary loc <loc>*
  – shows how many total ACEs/TCAM entries/stats counters are used on the linecard (per NP, where NP="chan#")
- *show access-lists ipv4 <acl> hardw ing resource-usage LOC*
  – shows compiled ACL hardware stats (TCAM, compression, etc)
- *show controller np struct SACL-PREFIX summary loc 0/0/cPU0*
  – shows prefix usage for compressed tables

Side note: use new apply-groups to manage config

```
group MY_ACL_INTF
 interface 'TenGigE0/[02]/0/[0-2]'
  ipv4 access-group example1-compressed ingress compress level 1
 !
end-group
group ospf-defaults
 router ospf '1'
  area '0'
   interface 'TenGigE.*'
    network point-to-point
    dead-interval 8
    hello-interval 2
end-group
```


Performance PPS impact of using scaled ACL

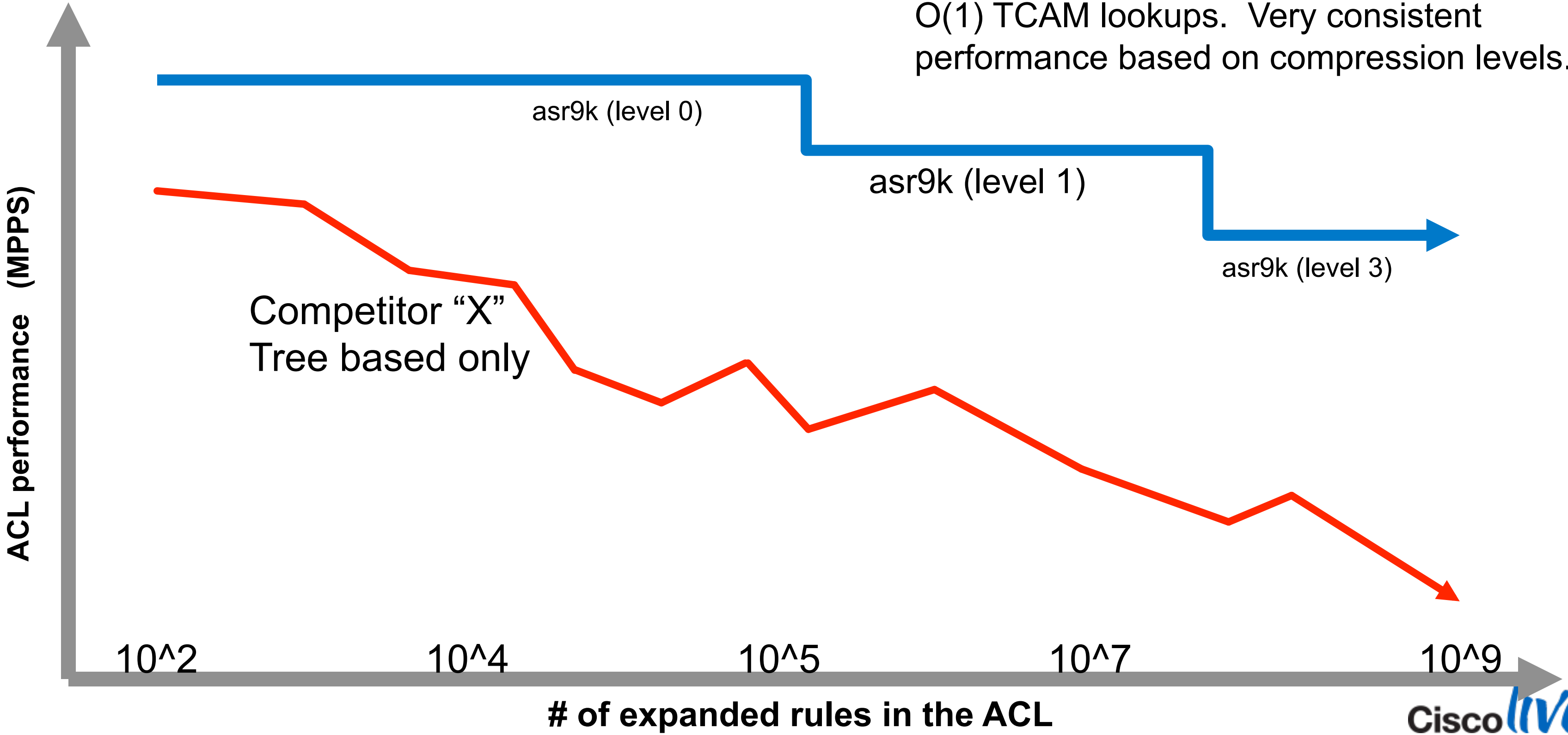
- No ACL no features: 44Mpps
- Uncompressed ACL: Input or Output ACL cost about ~10%
- Level 1 compression: Input ACL or Output ACL only cost about ~20%
- Level 3 compression: Input ACL or Output ACL cost about ~50%

- Performance degradation is because of tree lookup in search memory
- Remember that deny ACE cost less performance (packet is gone from pipeline)
  – *We'll talk more about that later*
- Non hybrid, tree based ACL differ in performance where you match in the ACL, ASR9000 does NOT suffer from that (TCAM!)

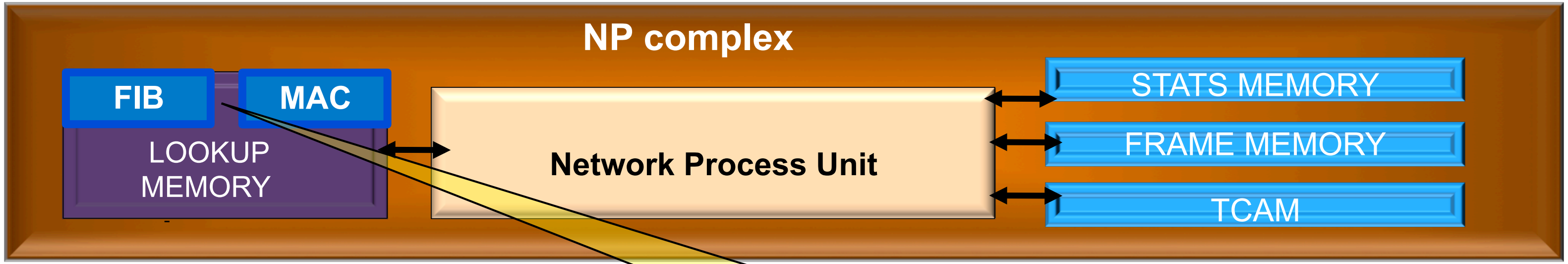
Disclaimer: These are indicational numbers from benchmarking only, specific to release and subject to variation

Performance overview

ASR9k: combination of short prefix trees and O(1) TCAM lookups. Very consistent performance based on compression levels.



L/B/E (Trident) SE/TR (Typhoon) Line Cards
What's the Difference?



- Each NPU has Four Main memories:
  - **Lookup/Search Memory (RLDRAM)**: stores MAC, FIB, and Adjacencies Tables
  - **TCAM**: classification (Vlan Tag (EVCs), QoS and Security ACL
  - **Stats QDR memory**: interface and forwarding statistics, policers data, etc
  - **Frame memory**: buffer memory for Queues
- 3 LC versions – low, base and extended - differ for size of memories
  - TCAM, QDR and Frame memory sizes depend on LC version

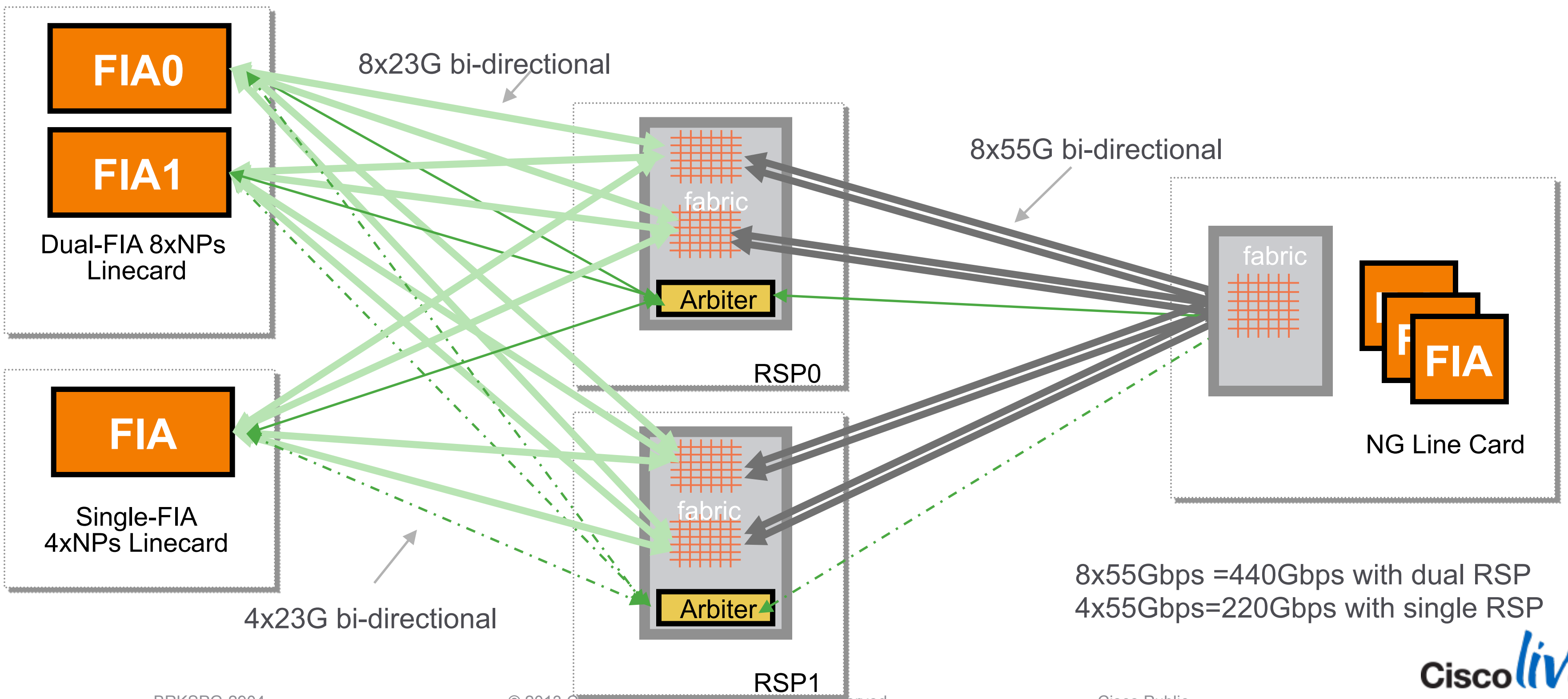
    Affects number of QoS queues and L2 sub-interfaces supported
  - Search Memory is same

    System level scale (unicast, multicast, MPLS label) adjacency and MAC address) not affected by a mix of LCs

Trident:
Shared search Mem for L2 and L3 (that is why there are scale profiles for Trident to shift boundary between L2 and L3)

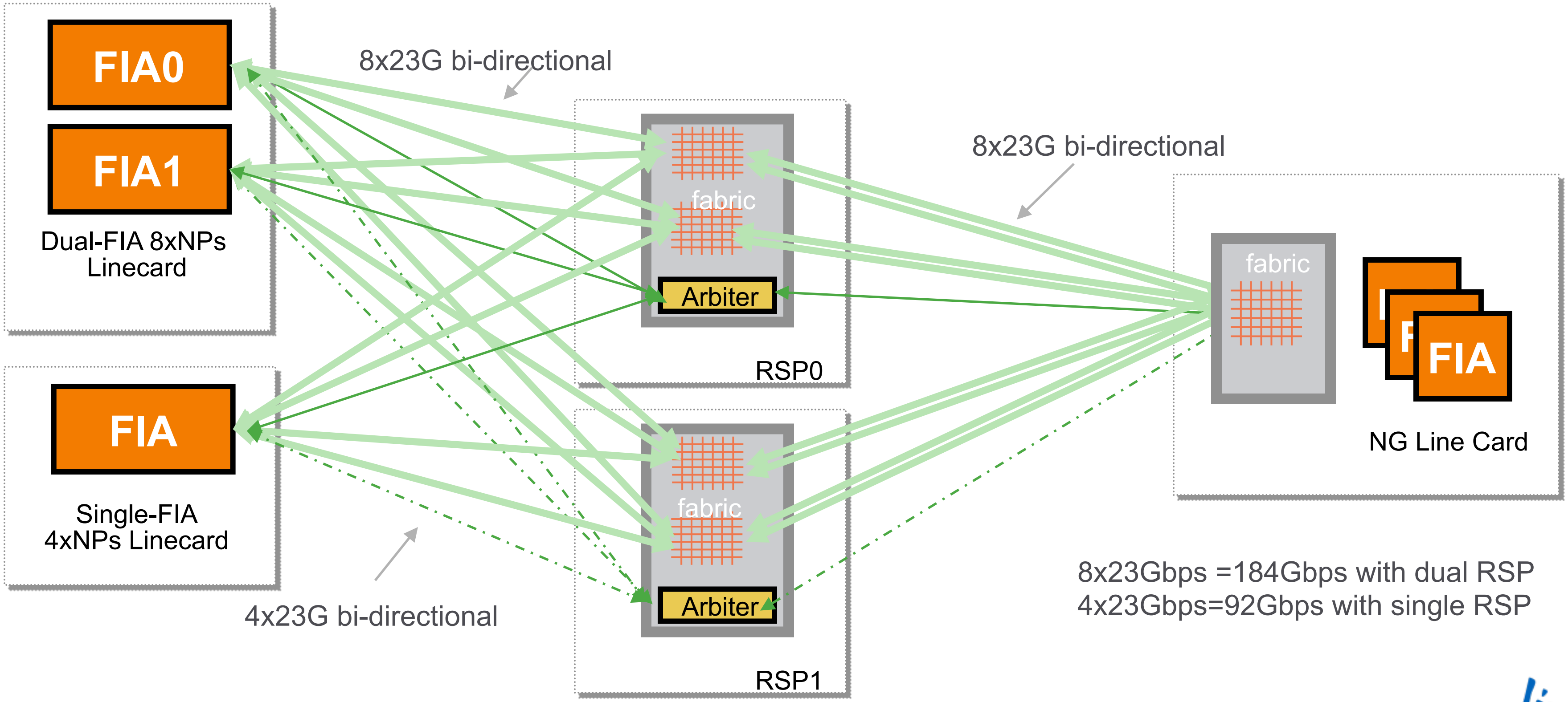
Typhoon:
Dedicated L2 and L3 separated search mem

Back-compatible: NG Switch Fabric
Mixed New Linecard and Existing Linecard



8x55Gbps =440Gbps with dual RSP
4x55Gbps=220Gbps with single RSP

Forward-compatible: Existing Switch Fabric Mixed NG Linecard and Existing Linecard

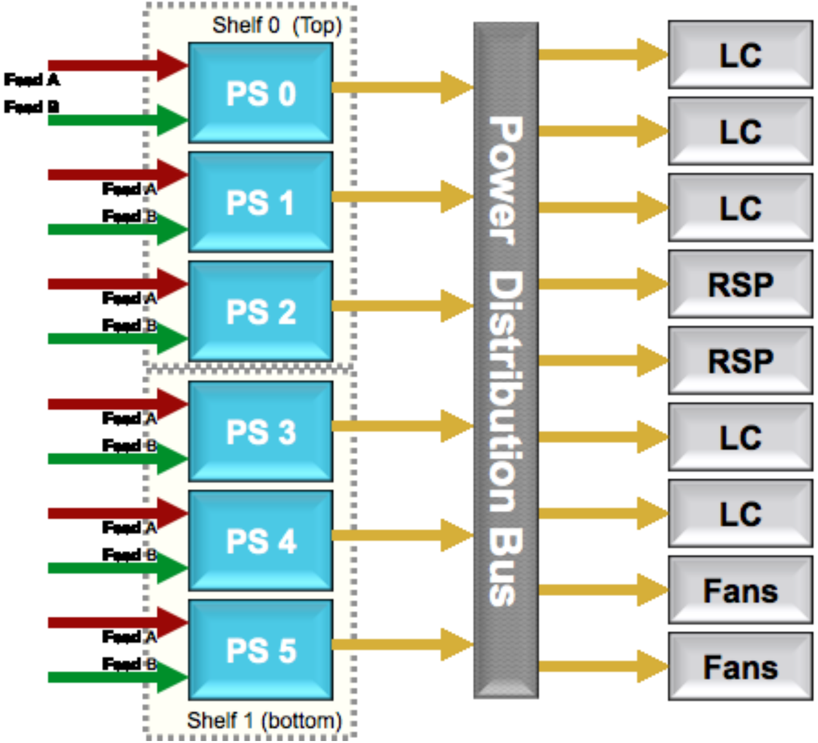


8x23Gbps =184Gbps with dual RSP
4x23Gbps=92Gbps with single RSP

Few words about power

Power Distribution (DC N:1 protection)

10 slots chassis



- Single power zone, one distribution bus
- All modules load share
- 2kW and 1.5kW supplies
- Each power supply is wired to both 'A' and 'B' feed
- Feed failure doubles draw on remaining feed
- supply failure increases draw on remaining supplies

•For DC Feed A & B loadshare

•You should see ~50% distribution

•Under "high load" conditions, all modules should provide almost equal power to the bus

•In Low load conditions this may be slightly off

•Picture shows "v1" power trays (3 per shelf). "v2" has 4 modules per shelf, same hardware, different formfactor.

•Each DC feed needs breaker for max amp (that is 2.1K/48V)

•Efficiency near 98%

•All modules feed the bus, RSPs booted first with Fans, LC's next starting slot 0 until avail power is gone

•Split 4 modules 2 on 2 (i2c bus on the shelf)

•Command "admin show env power"

Cisco live!

ic

Example output

```
P/0/RSP0/CPU0:A9K-BNG#admin show env power
Wed Jun 26 09:53:24.867 EDT
R/S/I    Modules          Capacity        Status
0/PM0/* host     PM       3000            Ok
0/PM1/* host     PM       0               Failed
R/S/I           Power Supply    Voltage         Current
                (W)             (V)             (A)
 0/PM0/*        1514.8          54.1            28.0
 0/PM1/*           0.0           0.0             0.0
--------------
Slot                                                      Max Watts
 ----                                                      ---------
 0/RSP0/CPU0                                                     350
 0/RSP1/CPU0                                                     350   (default)
 0/0/CPU0                                                        590
 0/2/CPU0                                                        850
 0/FT0/SP                                                        275
 0/FT1/SP                                                        275`
```

Module status

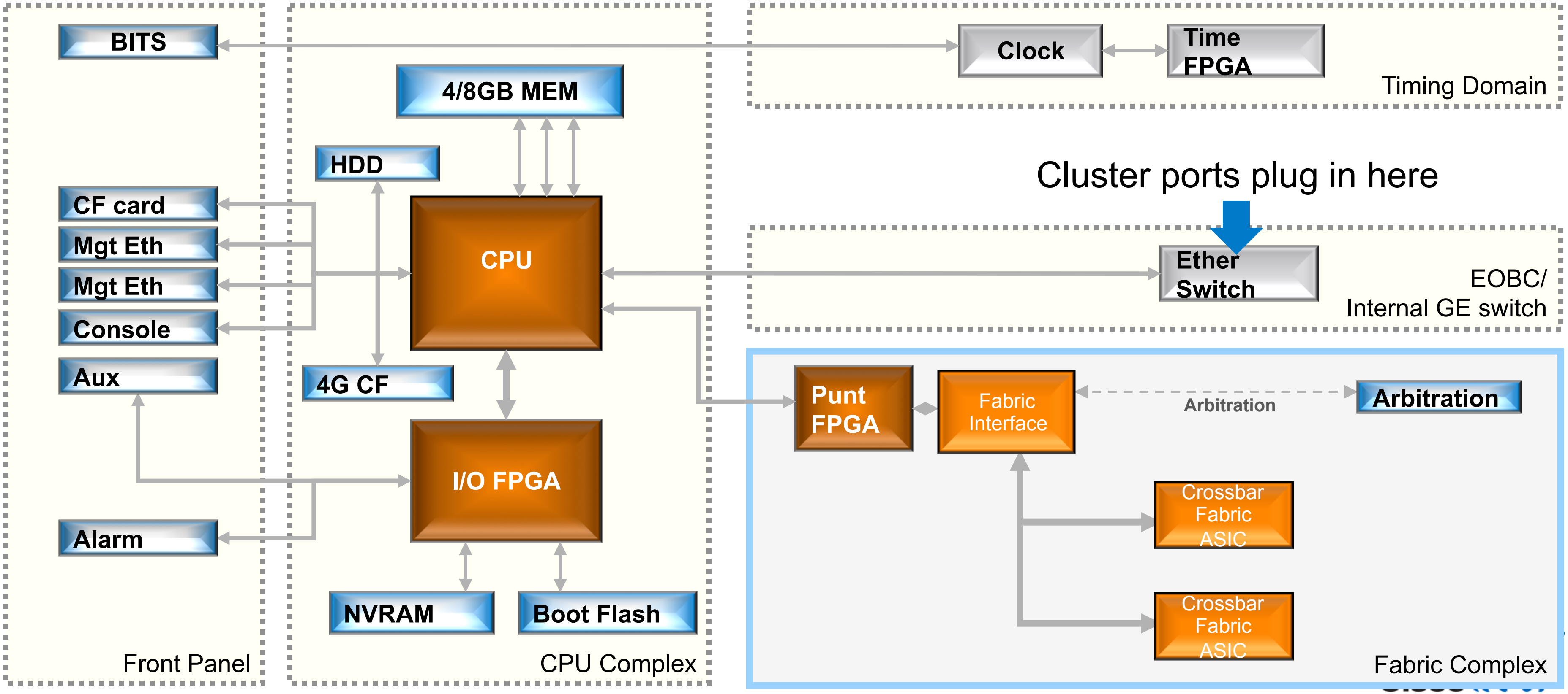
Actual draw

Software budget table (based on defined temperature profile)

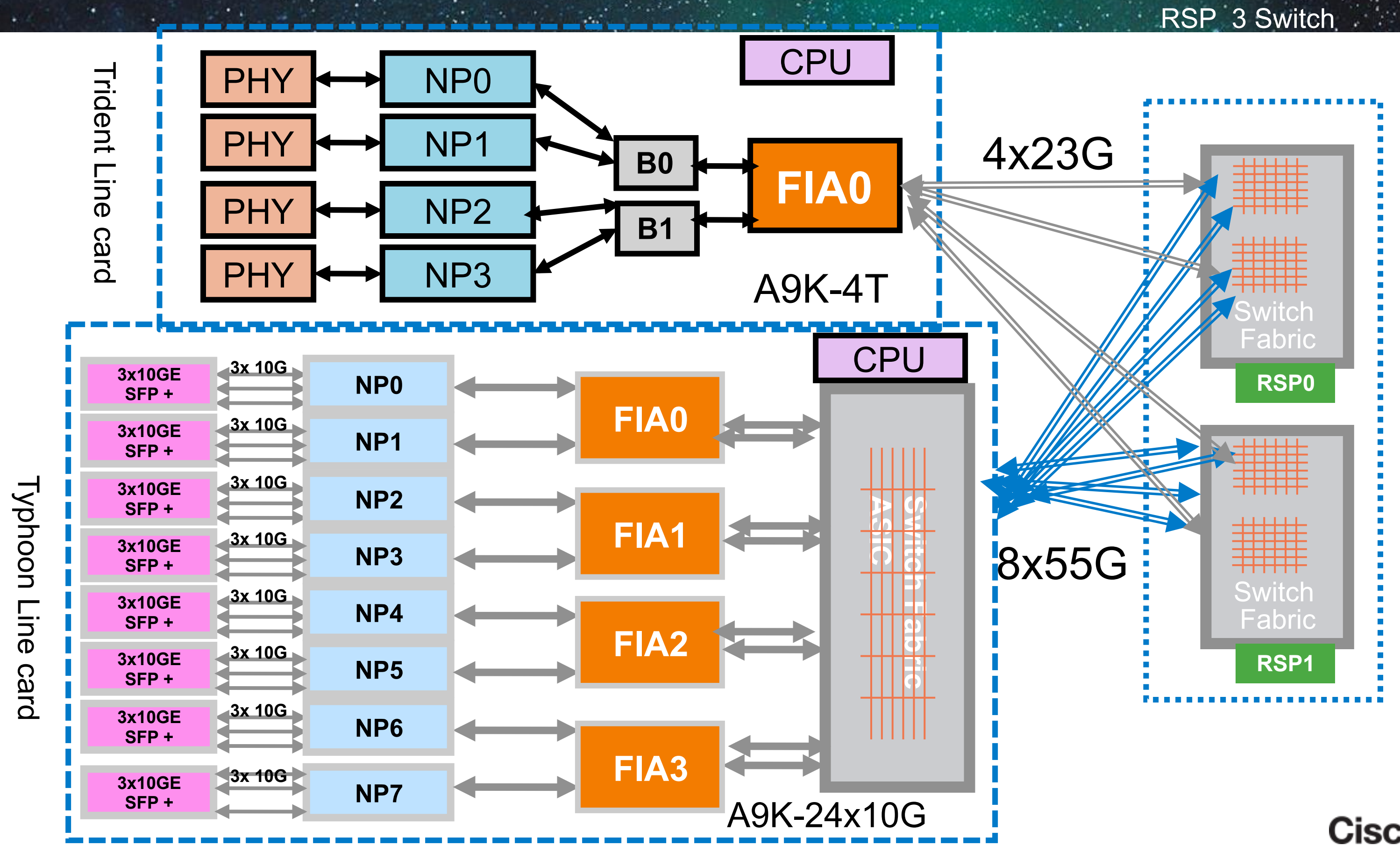
Hard coded, used by power manager to determine cards to boot

Standby RSP is allocated for by default

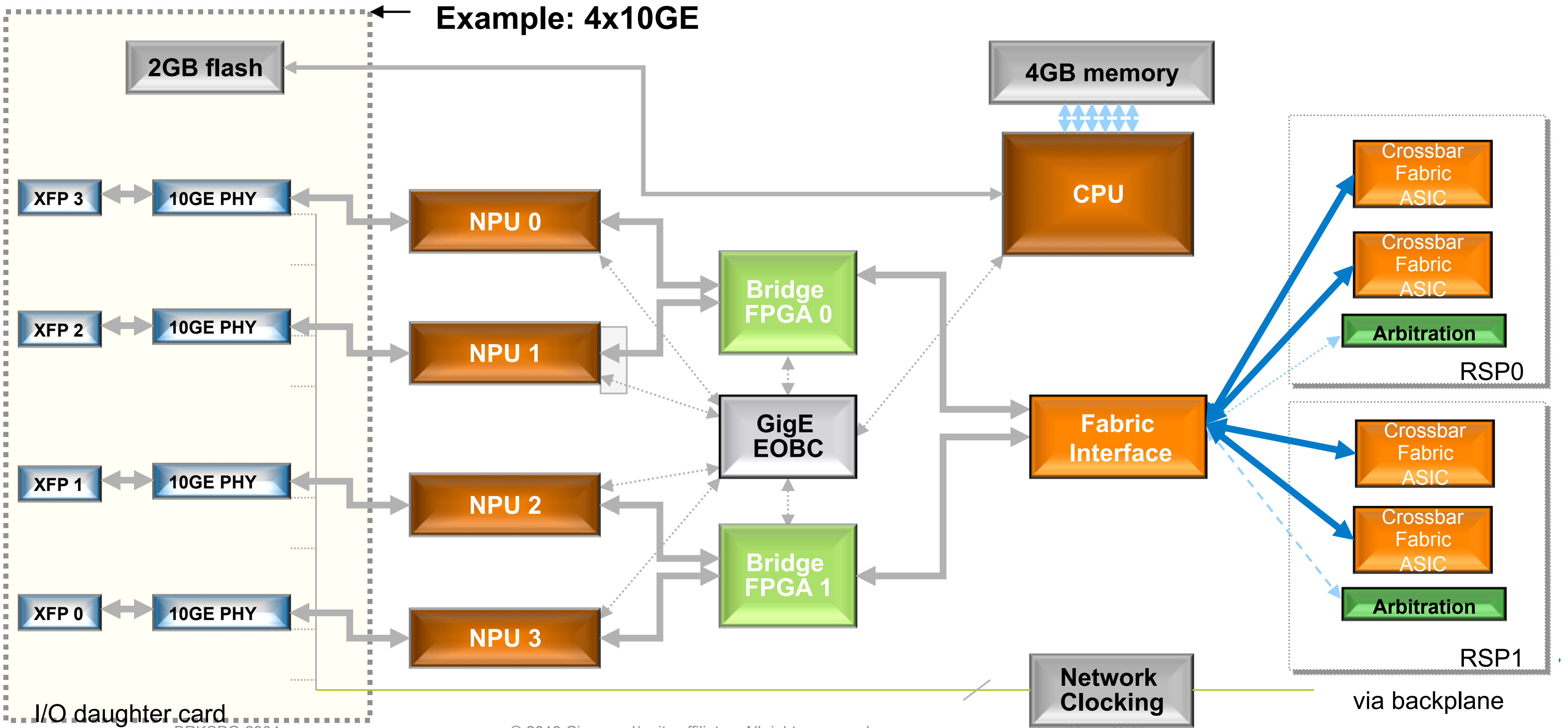
RSP Engine Architecture



Line Card Architecture Overview



40G Line Card Hardware Architecture



Generic LC Architecture (1) – Components

PHY

Pluggable physical interfaces

- speeds: GE, 10GE, 40GE, 100GE
- form factors: SFP, SFP+, XFP, QSFP, CFP
- media/reach: T, SR, LR, ZR, LR4, SR10
- colors: gray, CWDM, DWDM, Tunable

CPU

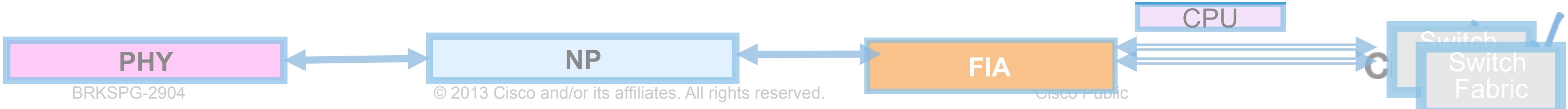
Distributed Control planes
SW switched packets
Inline Netflow
Program HW forwarding tables

Typhoon NP

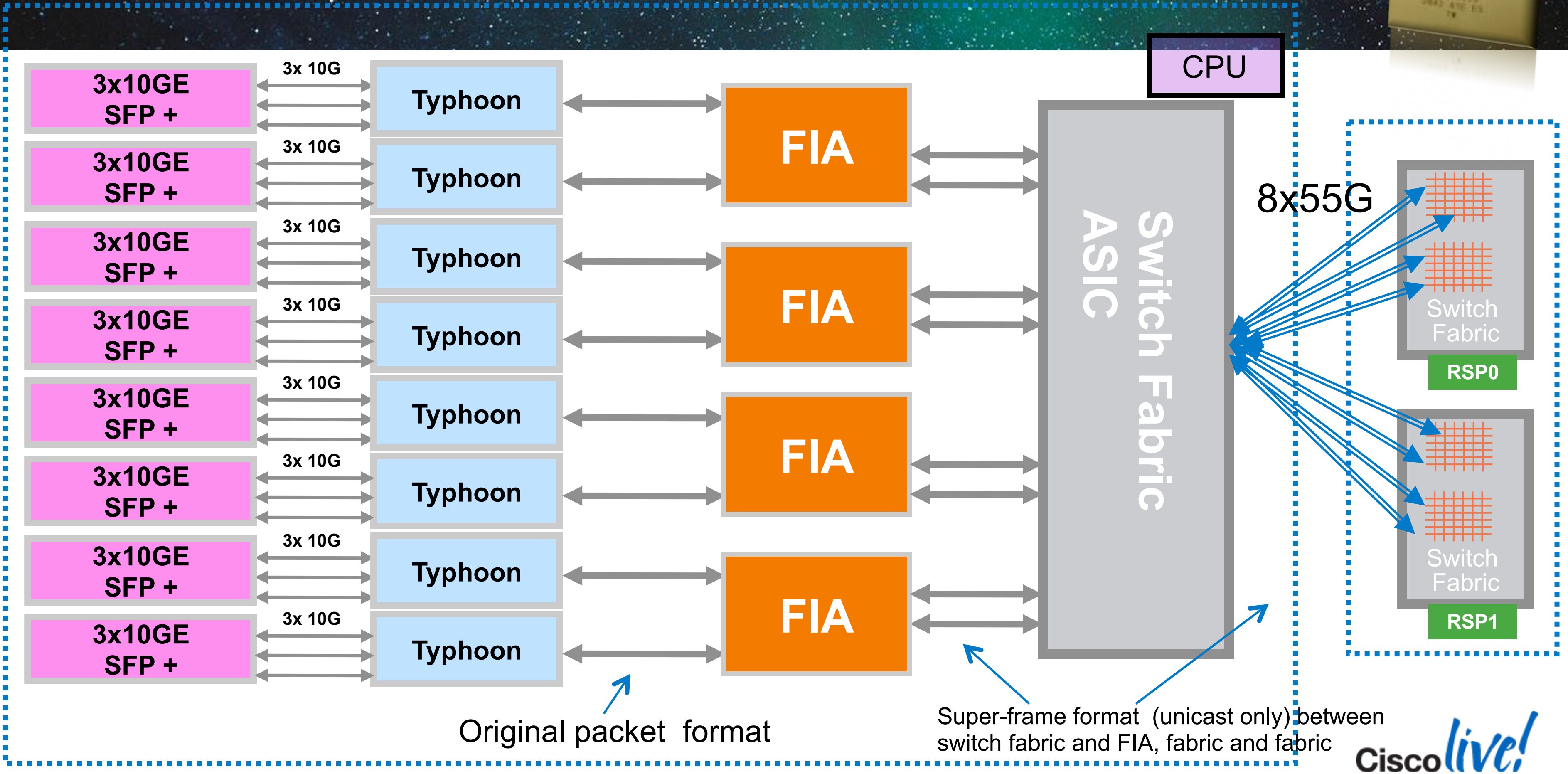
- forwarding and feature engine for the LC
- scales bandwidth via multiple NPs
  - up to 8 NPs/LC for performance vs. density options
- highly integrated silicon as opposed to multiple discrete components
  - shorter connections, faster communication channels
  - higher performance, density with lower power draw
  - simplified software development model

FIA

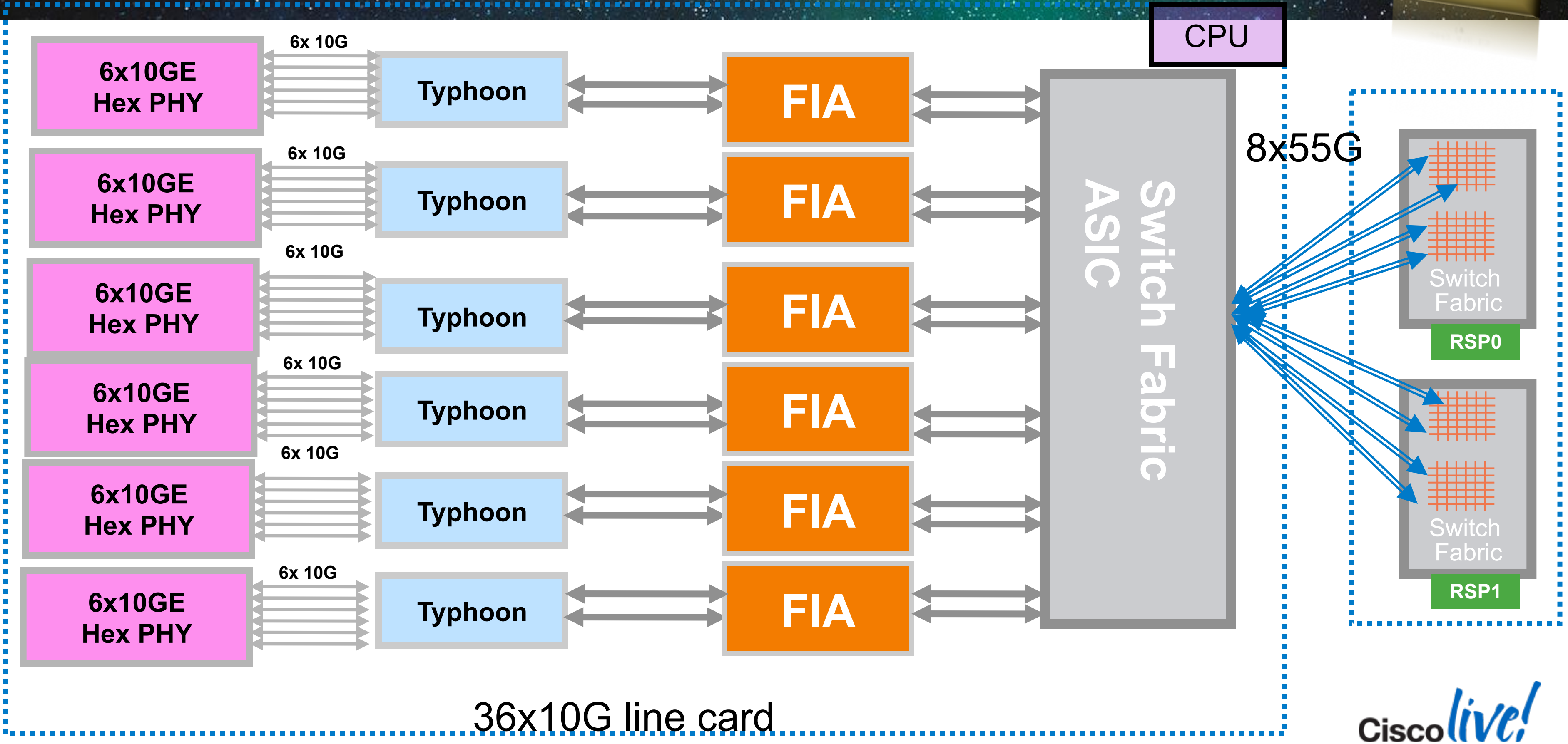
- interface between forwarding processor and system switch fabric
- arbitration, framing, accounting in HW
- provides buffering and virtual output queueing for the switch
  - passive backplane & switch itself has minimal buffering
- QoS awareness for Hi/Lo and ucast/mcast
  - total flexibility regarding relative priority of unicast vs. multicast



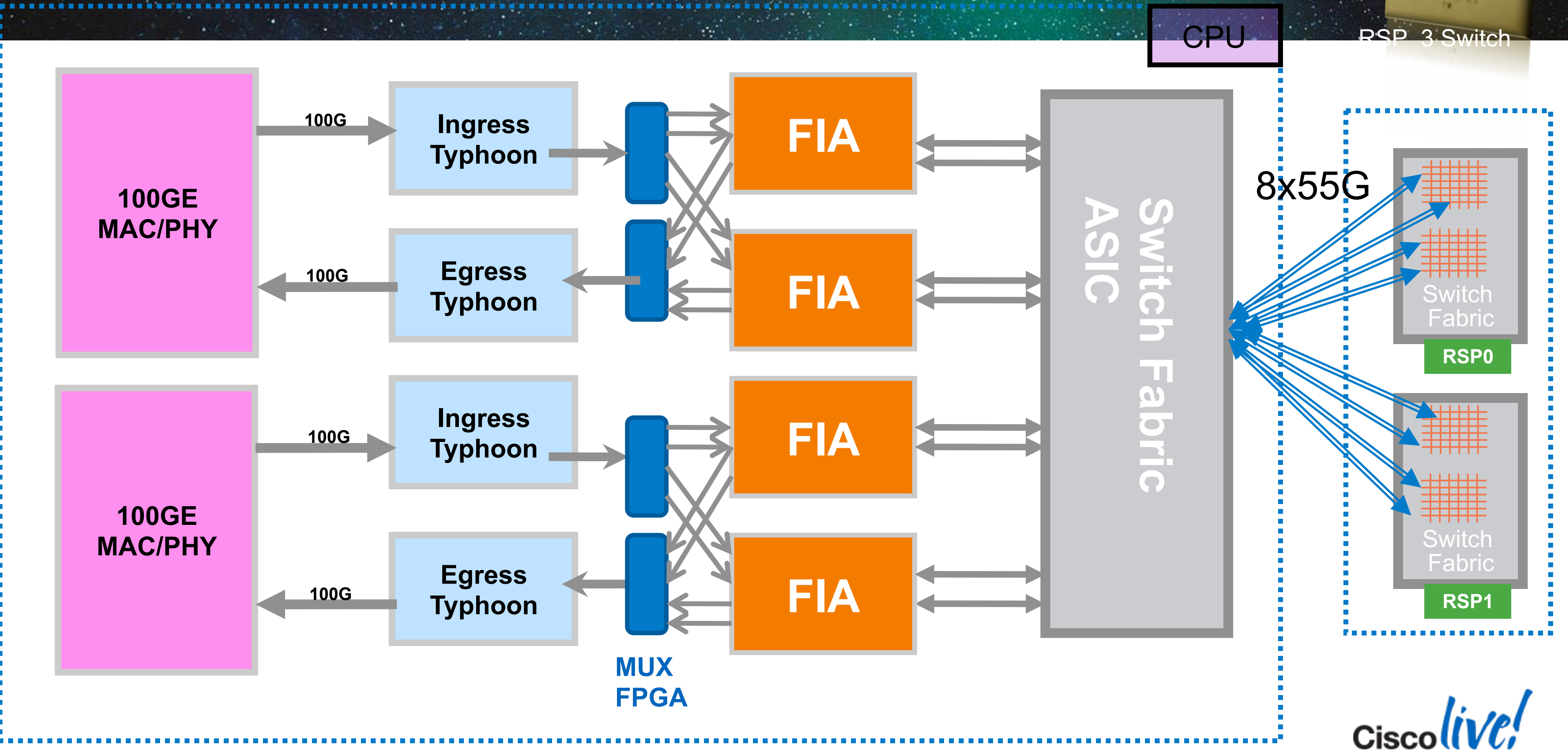
LC Architecture – 24x10G



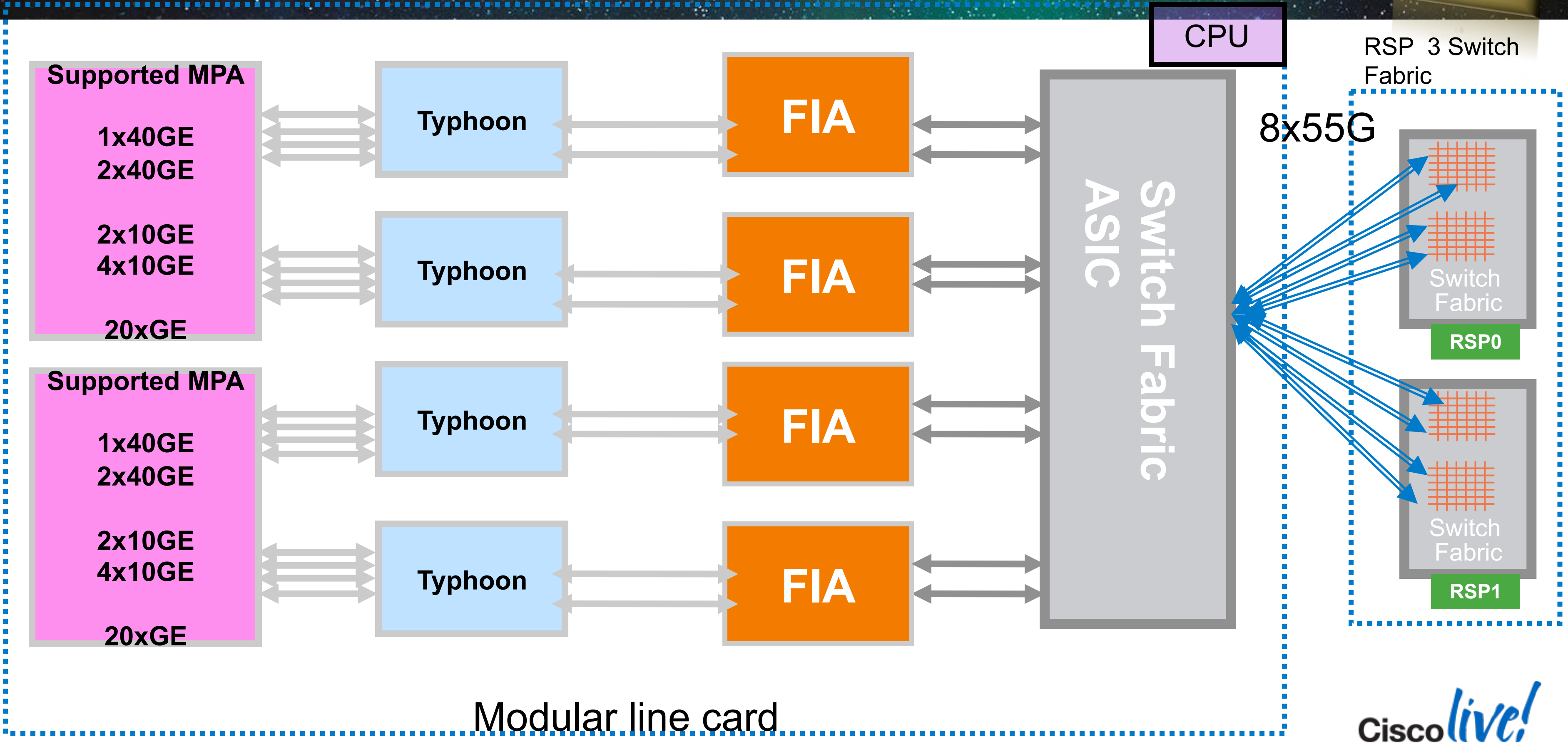
LC Architecture – 36x10G



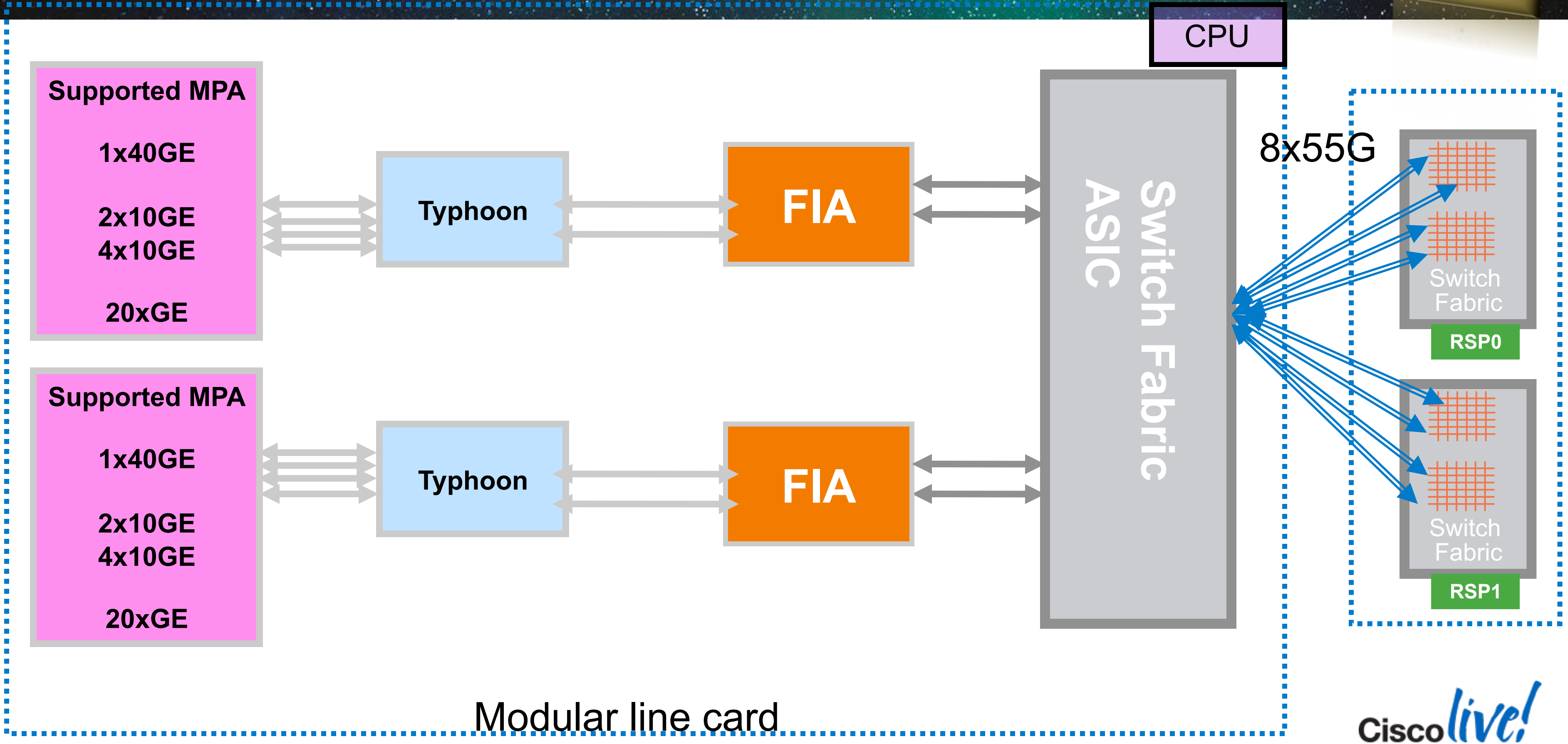
LC Architecture – 2x100G



LC Architecture – Modular Ethernet MOD160

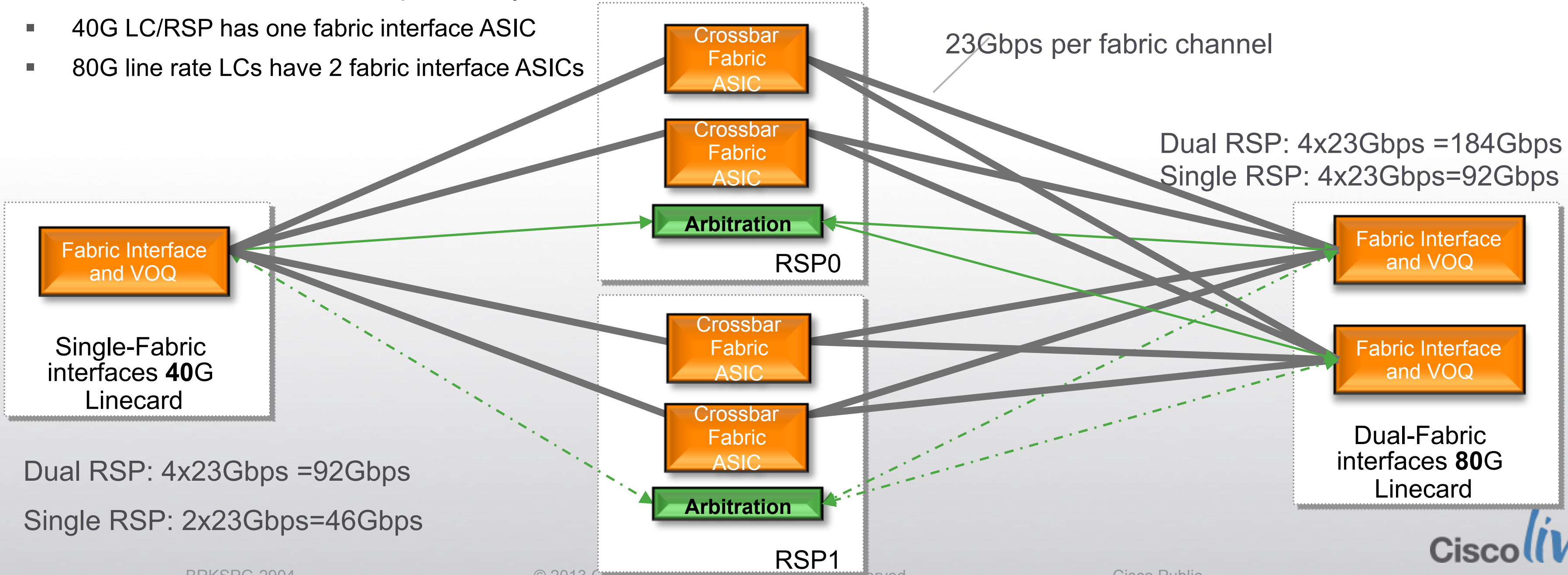


LC Architecture – Modular Ethernet MOD80



Fabric Overview

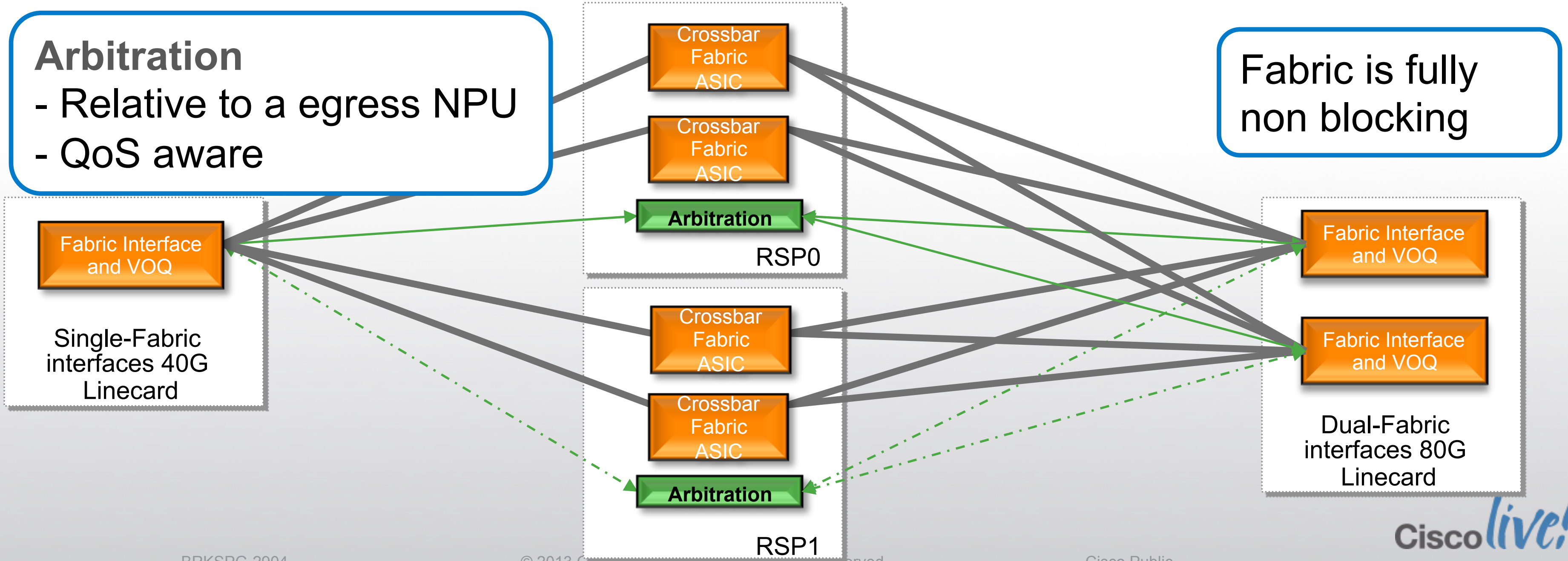
- Physically separated from LC. Resides on RSP
- Logically separated from LC and RSP
  - All fabric ASICs run in active mode regardless of RSP Redundancy status
  - Extra fabric bandwidth and instant fabric switch over
  - If the FAB has been previously initiated then even with RP in rommon FABRIC IS ACTIVE!
- 40G LC/RSP has one fabric interface ASIC
- 80G line rate LCs have 2 fabric interface ASICs



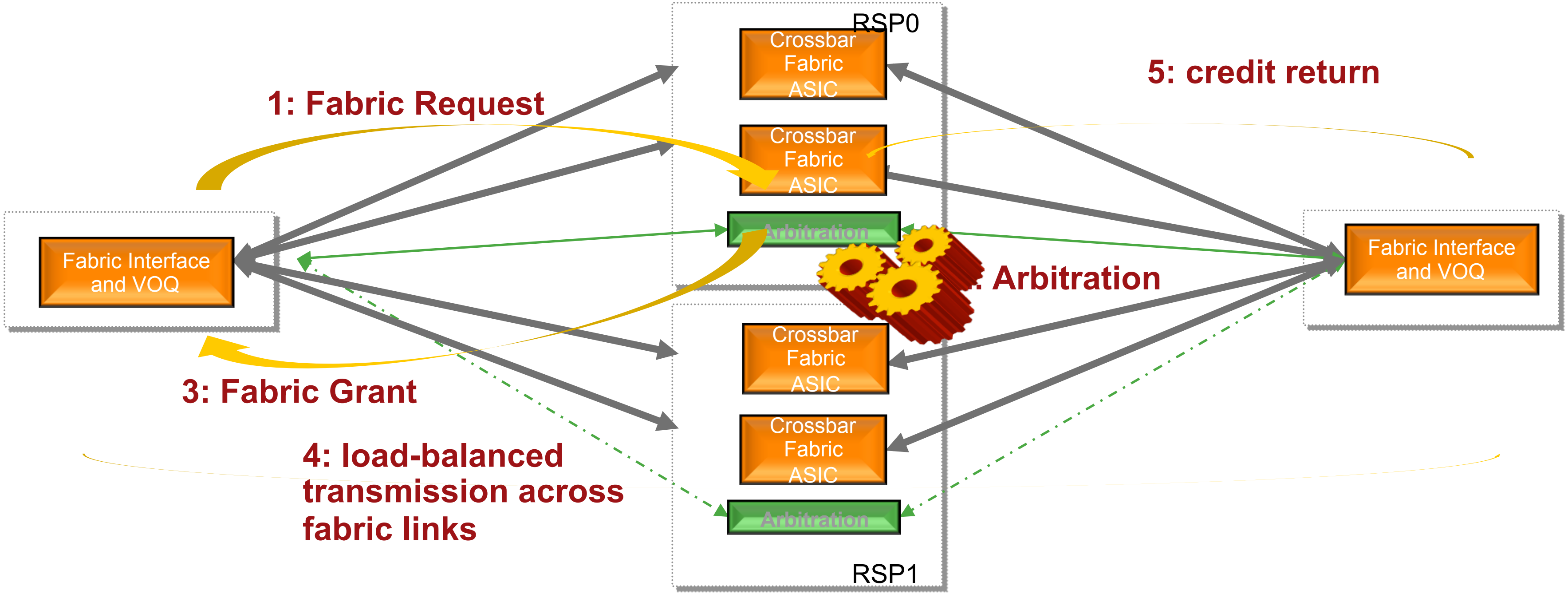
Fabric Arbitration and Redundancy

"0" packet loss guarantee during RSP failover and OIR

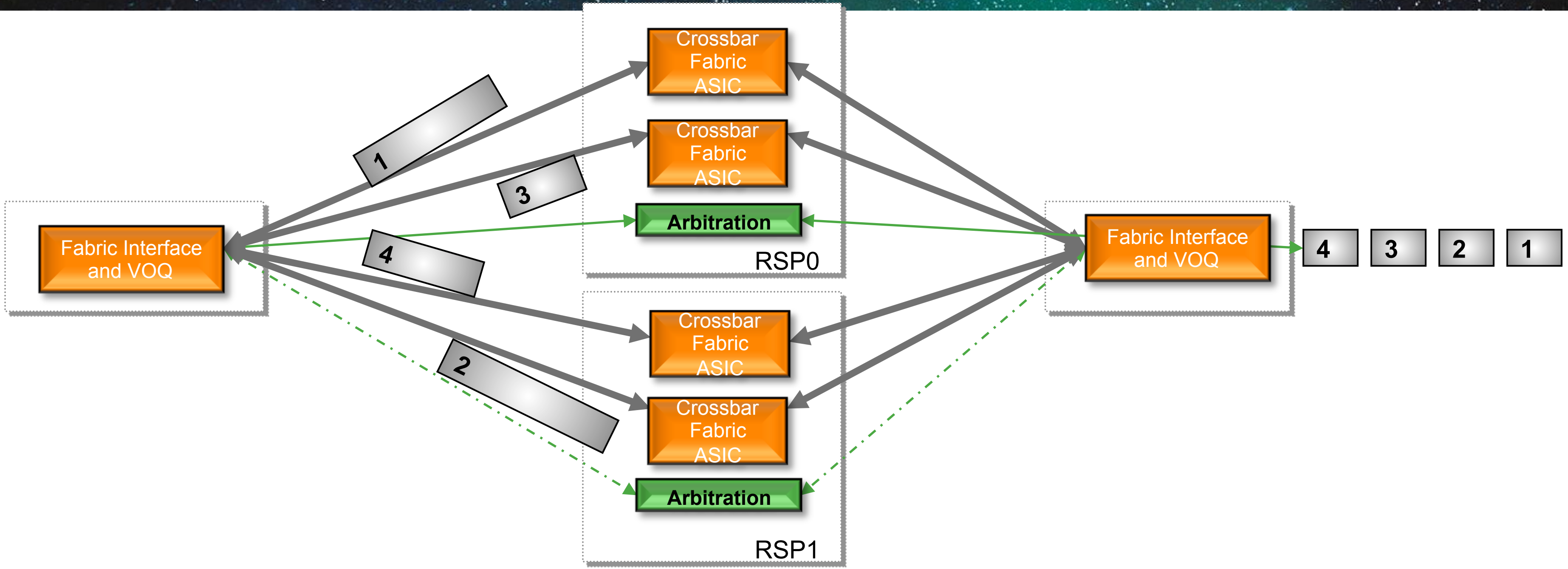
- Access to fabric controlled using central arbitration.
  - One Arbitration ASIC (Arbiter) per RSP
  - Both Arbiters work in parallel – both answer to requests to transmit
  - FIAs follow active Arbiter, and switch to backup if needed
  - Arbiter switchover controlled by low level hardware signalling



Fabric Arbitration

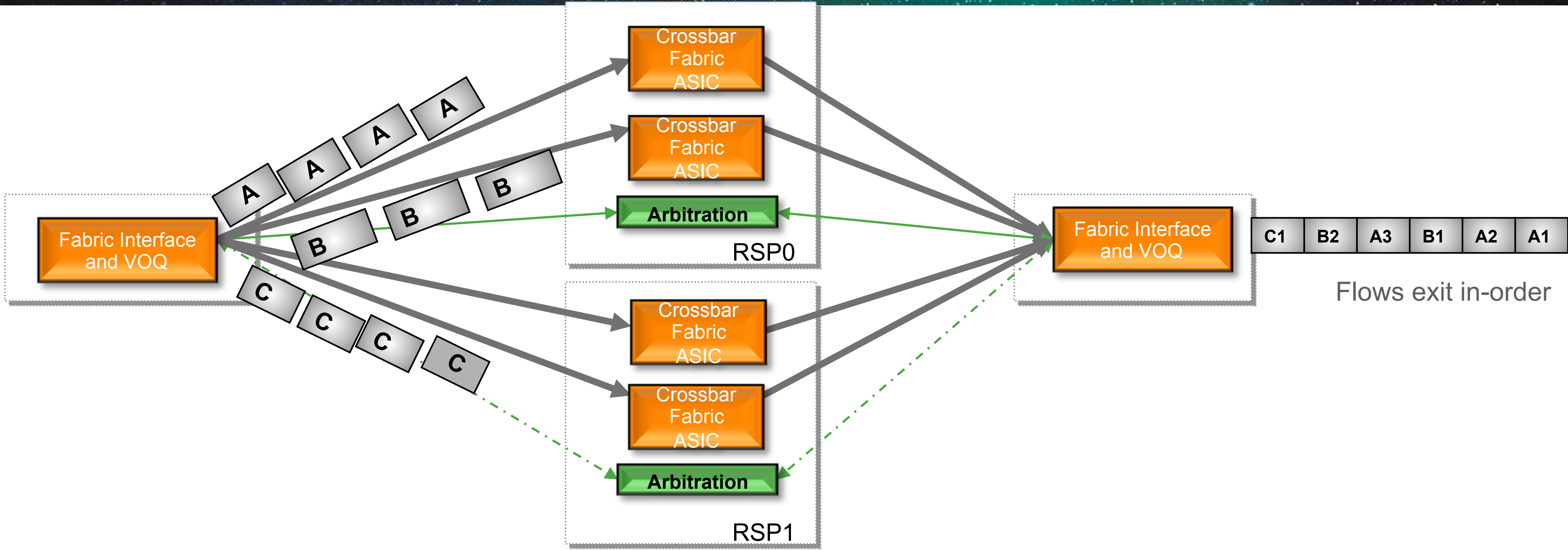


Fabric Load Sharing – Unicast



- Unicast traffic sent across first available fabric link to destination (maximizes efficiency)
- Each frame (or superframe) contains sequencing information
- All destination fabric interface ASIC have re-sequencing logic
- Additional re-sequencing latency is measured in nanoseconds

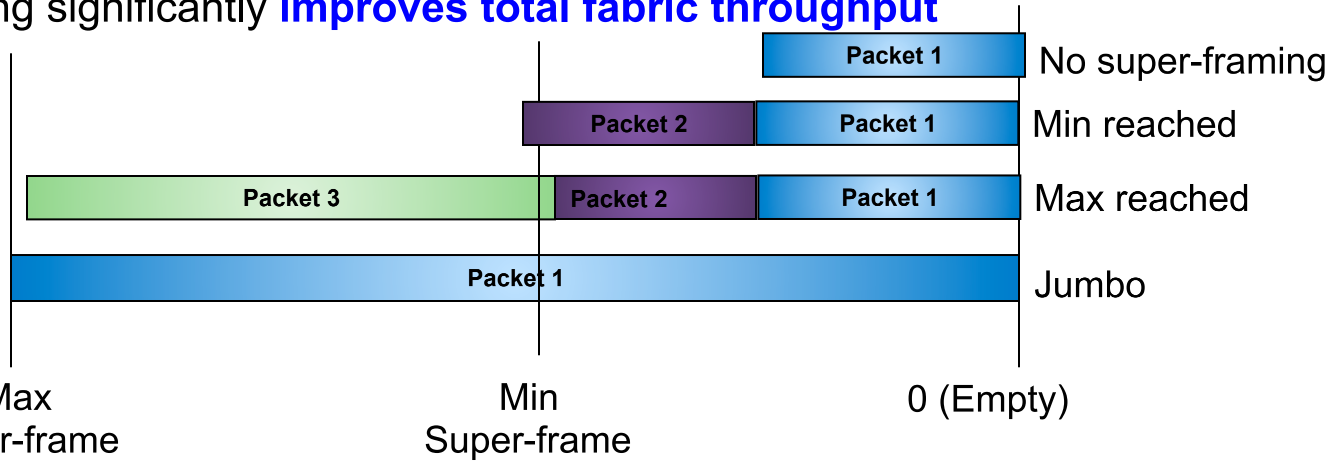
Fabric Load Sharing – Multicast



- Multicast traffic hashed based on (S,G) info to maintain flow integrity
- Very large set of multicast destinations preclude re-sequencing
- Multicast traffic is non arbitrated – sent across a different fabric plane

Fabric Super-framing Mechanism

- **Multiple unicast frames** from/to same destinations aggregated into **one super frame**
- Super frame is created if there are frames waiting in the queue, up to 32 frames or when min threshold met, can be aggregated into one super frame
- Super frame only apply to unicast, **not multicast**
- Super-framing significantly **improves total fabric throughput**



- Note that fabric counters are showing super frames not individual packets!!
  - (show controller fabric fia loc 0/X/CPU0)

Meaning of hard drop -x reason in *sh controllers fabric fia drops [ingress|egress]*

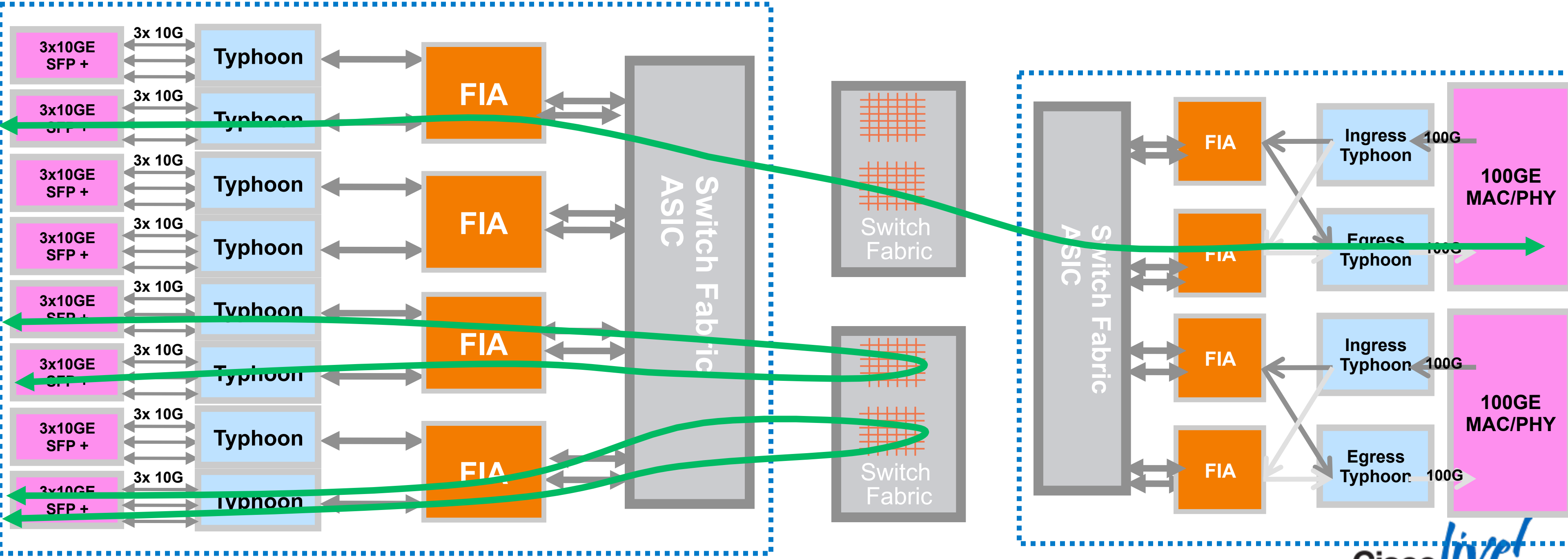
There are four priority levels and four physical XBAR links. Now the confusion is that, fia egress drop stats are per priority, while fia ingress drop stats are per XBAR link.
The fia egress drop stats, Tail, Hard, WRED, (offsets 0-3) represent fabric priority stats and correspond as...
0 - high priority level 1
1 - high priority level 2
2 - low priority
3 - not used (asr9k)

The fia ingress drop stats offsets (0-3) represent XBAR link stats and correspond as...

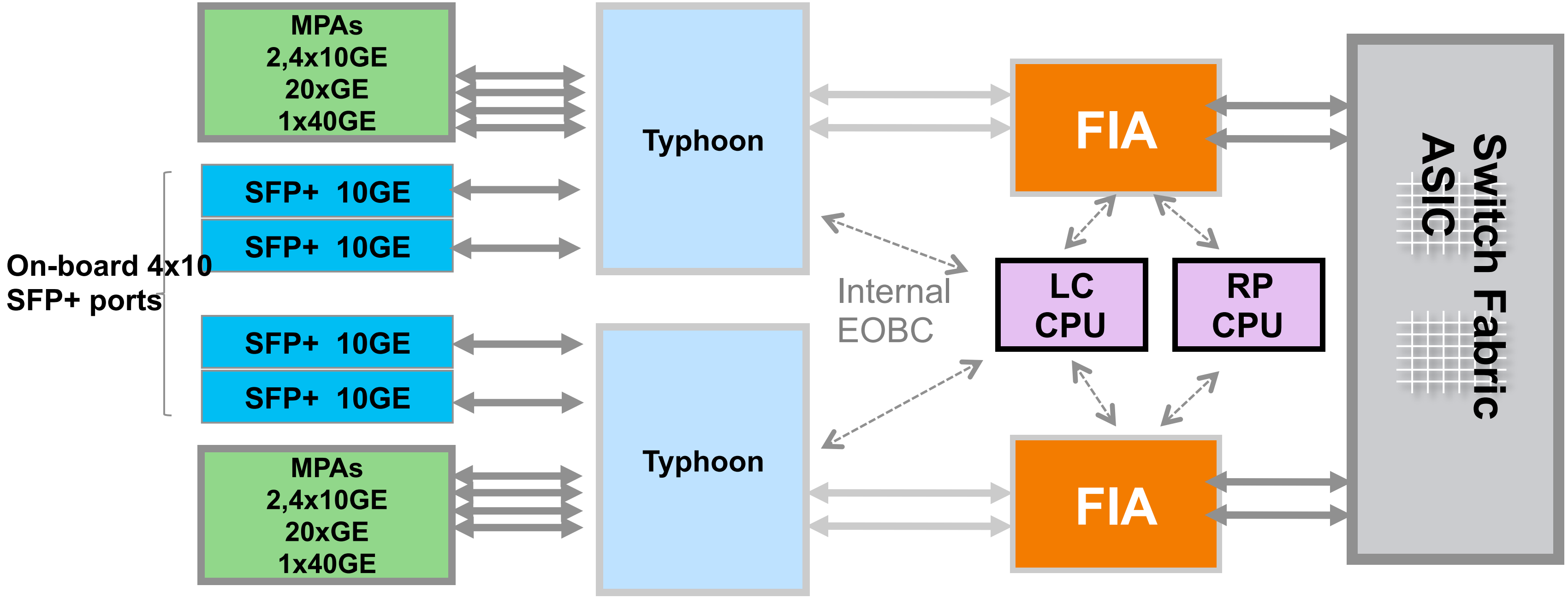
0-1 XBAR links to RSP0 (Trident+RSP2)
2-3 XBAR links to RSP1 (Trident+RSP2)
On Typhoon cards the FIA links with 2 links to the local fabric.
The local fabric connects with 8x55G links to the RSP fabric

Packet Flow Overview

Same as existing system: Two-stage IOS-XR packet forwarding
Uniform packet flow: All packet go through central fabric on the RP



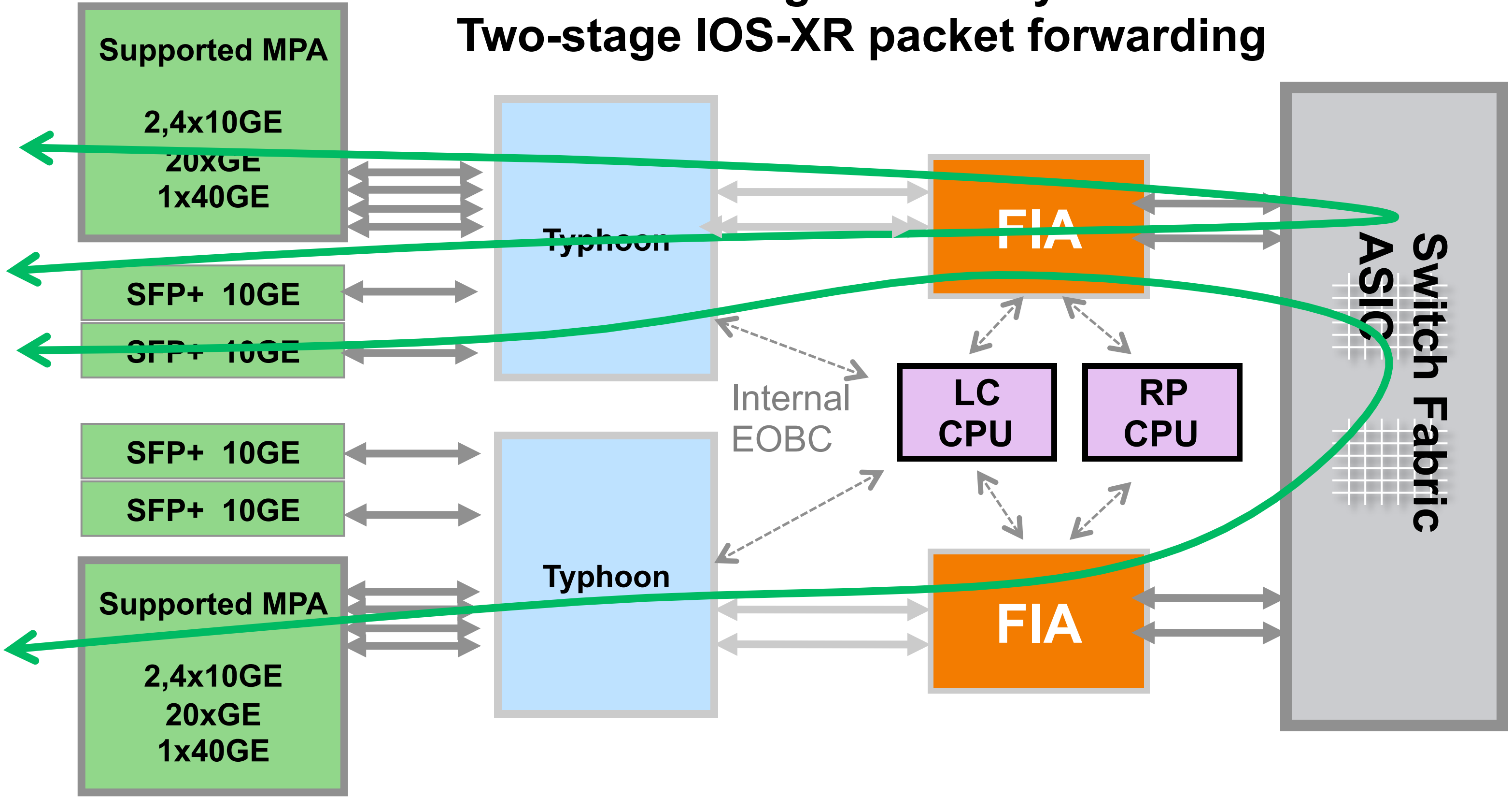
ASR 9001 System Architecture Overview



It has both central RP and LC CPU like big chassis
But it only have central switch fabric, no LC fabric
Maximum 120Gbps bi-directional system.
9001-S, a 60G version is available with only 1 Bay enabled, can upgrade to 120G via license

ASR 9001 Packet Flow Overview

Same as big chassis system:
Two-stage IOS-XR packet forwarding



Port to NPU mapping

RP/0/RSP0/CPU0:A9K-BNG#**show controller np ports all loc 0/0/cpU0**

```
                Node: 0/0/CPU0:
----------------------------------------------------------------

NP Bridge Fia                       Ports
-- ------ --- ---------------------------------------------------
0  --     0   GigabitEthernet0/0/0/0 - GigabitEthernet0/0/0/9
1  --     1   GigabitEthernet0/0/0/10 - GigabitEthernet0/0/0/19
2  --     2   TenGigE0/0/1/0
3  --     3   TenGigE0/0/1/1
```




Troubleshooting ASR9000 Forwarding

NPU Packet Processing - Ingress

5 Stages:

All packets go through the TM regardless of whether QOS is enabled

Parse → Search → Resolve → Modify → Queueing Scheduling

Parse
- L2/L3 header packet parsing in TCAM
- Builds keys for ingress ACL, QoS and forwarding lookups (uCode)

Search
- Performs QoS and ACL lookups in TCAM tables
- Performs L2 and L3 lookups in RLDRAM

Resolve
- Processes Search results:
  - ACL filtering
  - Ingress QoS classification and policing
  - Forwarding (egress SFP determined)
- Performs L2 MAC learning

Modify
- Adds internal system headers
- Egress Control Header (ECH)
- Switch Fabric Header (SFH)

Queueing Scheduling
- Queuing, Shaping and Scheduling functions

Where to start when there are forwarding issues

- First identify interface in question with problem
- Identify the mapping from interface to NPU
  – Show controller np ports all location 0/X/CPU0 (where x is the slot)
- Show the controller NPU counters
  – Show controller np count npY location 0/X/CPU0 (where y is the NPU for IF)
- Look for rate counters that match lost traffic rate
- Lookup description for counter (see next slide)
- Check FIA counters
- Check fabric counters
- Move to egress interface and repeat steps 2 and 3.

Example

```
RP/0/RSP0/CPU0:A9K-BNG#show controller np counters np0 loc 0/0/CPU0
                Node: 0/0/CPU0:
----------------------------------------------------------------
Show global stats counters for NP0, revision v2
Read 57 non-zero NP counters:
Offset  Counter                              FrameValue   Rate (pps)
-------------------------------------------------------------------------------
 16  MDF_TX_LC_CPU                            22755787         6
 17  MDF_TX_WIRE                              1614696          0
 21  MDF_TX_FABRIC                            1530106          0
 29  PARSE_FAB_RECEIVE_CNT                    1555034          0
 33  PARSE_INTR_RECEIVE_CNT                   22026578         6
 37  PARSE_INJ_RECEIVE_CNT                    335774           0
 41  PARSE_ENET_RECEIVE_CNT                   2115361          1
 45  PARSE_TM_LOOP_RECEIVE_CNT                17539300         5
```

MDF=Modify TX transmit WIRE to the wire = egress

Packets received from the fabric

Delta between received from Fab to TX-wire should almost be 0, if not, we dropped packets, could be ACL, QOS, or for other reasons (eg PUNT)

Note

- Some counters have an index to a port.
- For instance, there is an aggregate count per NPU showing the misses from vlan to subinterface mapping:
  - UIDB_TCAM_MISS_AGG_DROP
- There is also a specific counter from which port index these drops came from:
  - UIDB_TCAM_MISS_DROP_1
- This means that the second port (starting count from zero) on that NPU experienced that drop.
- So if your show controller np ports tells us that ports X Y and Z are connected to this NPU, and the drop index is _1, then port Y is the culprit.

Capturing lost packets in the NPU

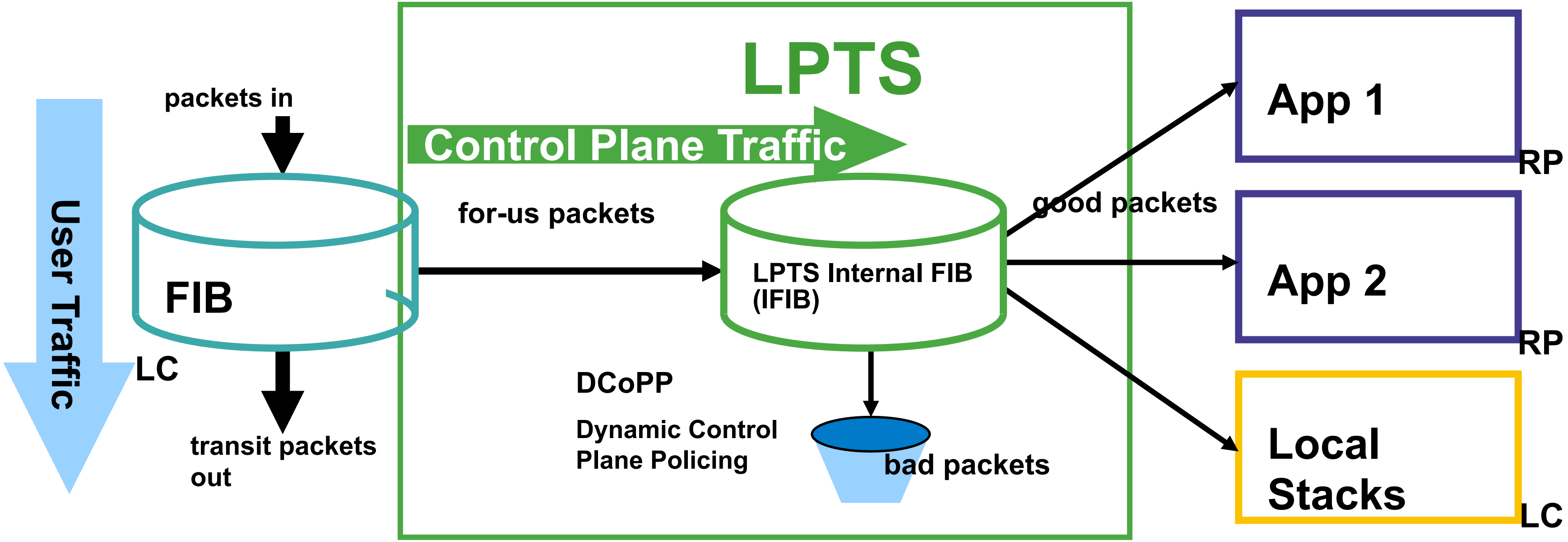
- CLI:
  - monitor np counter <COUNTER_NAME> <NPU> count <N>

- You can monitor any counter in the NPU
- For an X number of packets when it exits automatically
- It will reset the NPU (3 second forwarding stop) when completed or exited
  - This will be enhanced later
- Packets subject to punt cant be captured by this methodology
- Captured packets are always dropped
- Use with care




Troubleshooting ASR9000 Forwarding Punt/Inject verification (LPTS)

IOS XR Control Plane
Local Packet Transport Service



- LPTS enables applications to reside on any or all RPs, DRPs, or LCs
  - Active/Standby, Distributed Applications, Local processing
- IFIB forwarding is based on matching control plane flows
  - DCoPP is built in firewall for control plane traffic.
- LPTS is transparent and automatic

IOS XR LPTS in action

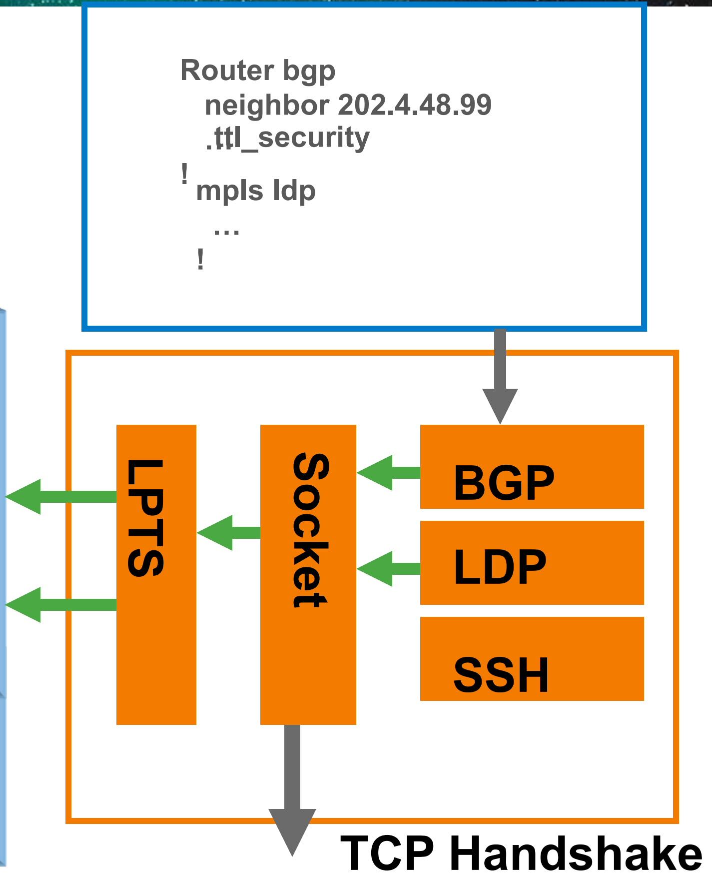
- LPTS is an automatic, built in firewall for control plane traffic.
- Every Control and Management packet from the line card is rate limited in hardware to protect RP and LC CPU from attacks

LC 1 IFIB TCAM HW Entries

Local	port	Remote	port	Rate	Priority
Any	ICMP	ANY	ANY	1000	low
any	179	any	any	100	medium
any	179	202.4.48.99	any	1000	medium
202.4.48.1	179	202.4.48.99	2223	10000	medium
200.200.0.2	13232	200.200.0.1	646	100	medium

ttl 255

LC 2 IFIB TCAM HW Entries …



Verifying LPTS policer values

```
RP/0/RP0/CPU0:CRS1-4#show lpts pifib hardware police location 0/7/CPU0
-------------------------------------------------------------
                Node 0/7/CPU0:
-------------------------------------------------------------
 Burst = 100ms for all flow types
-------------------------------------------------------------
FlowType                Policer Type    Cur. Rate  Def. Rate  Accepted   Dropped
----------------------- ------- ------- ---------- ---------- ---------- ----------
unconfigured-default    100     Static  500        500        0          0
Fragment                106     Global  0          1000       0          0
OSPF-mc-known           107     Static  20000      20000      0          0
OSPF-mc-default         111     Static  5000       5000       0          0
OSPF-uc-known           161     Static  5000       5000       0          0
OSPF-uc-default         162     Static  1000       1000       0          0
BGP-known               113     Static  25000      25000      18263      0
BGP-cfg-peer            114     Static  10000      10000      6          0
BGP-default             115     Global  0          10000      0          2
PIM-mcast               116     Static  23000      23000      19186      0
PIM-ucast               117     Static  10000      10000      0          0
IGMP                    118     Static  3500       3500       9441       0
ICMP-local              119     Static  2500       2500       1020       0
ICMP-app                120     Static  2500       2500       0          0
na                      164     Static  2500       2500       72         0
LDP-TCP-cfg-peer        152     Static  10000      10000      0          0
LDP-TCP-default         154     Static  10000      10000      0          0
……cut……
```

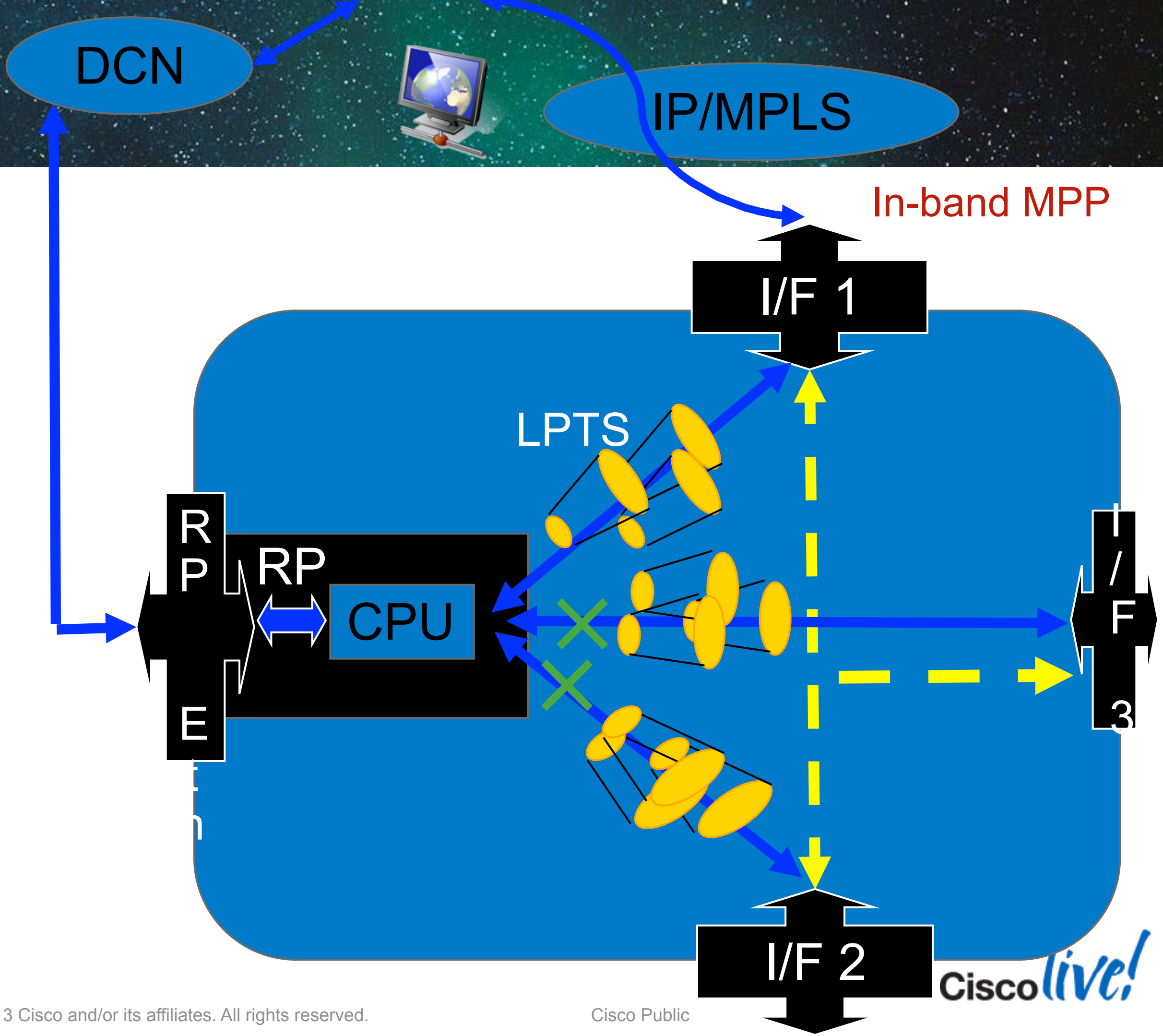
```
lpts pifib hardware police
  flow fragment rate 0
  flow bgp default rate 0
```

Tightening LPTS

- If you can use only p2p OSPF network type
  - `flow ospf-uc-known rate 0`
  - `flow ospf-uc-default rate 0`
- Note that OSPF p2p network type is the recommended setting even on Ethernet interfaces unless you have multiple routers on the same segment.
- Do we really need BGP, LDP-TCP, MSDP, default – for unconfigured sessions
  - `flow bgp-default rate 0`
  - `flow ldp-tcp-default rate 0`
  - `flow msdp-default rate 0`
- Further investigation needed for the following
  - `flow udp-default rate 0`
  - `flow tcp-default rate 0`
  - `flow raw-default rate 0`

MPP

- I/F 1 is configured as MPP in-band interface. I/F 1 is also part of global routing/ forwarding.
- Management traffic to RP from all non-MPP interfaces is dropped (I/F 2 and I/F 3).
- RP Eth/Console/Aux continues to operate as dedicated out-of-band.
- LPTS still continues to provide rate limiting irrespective of MPP.



Troubleshooting MPP -- LPTS

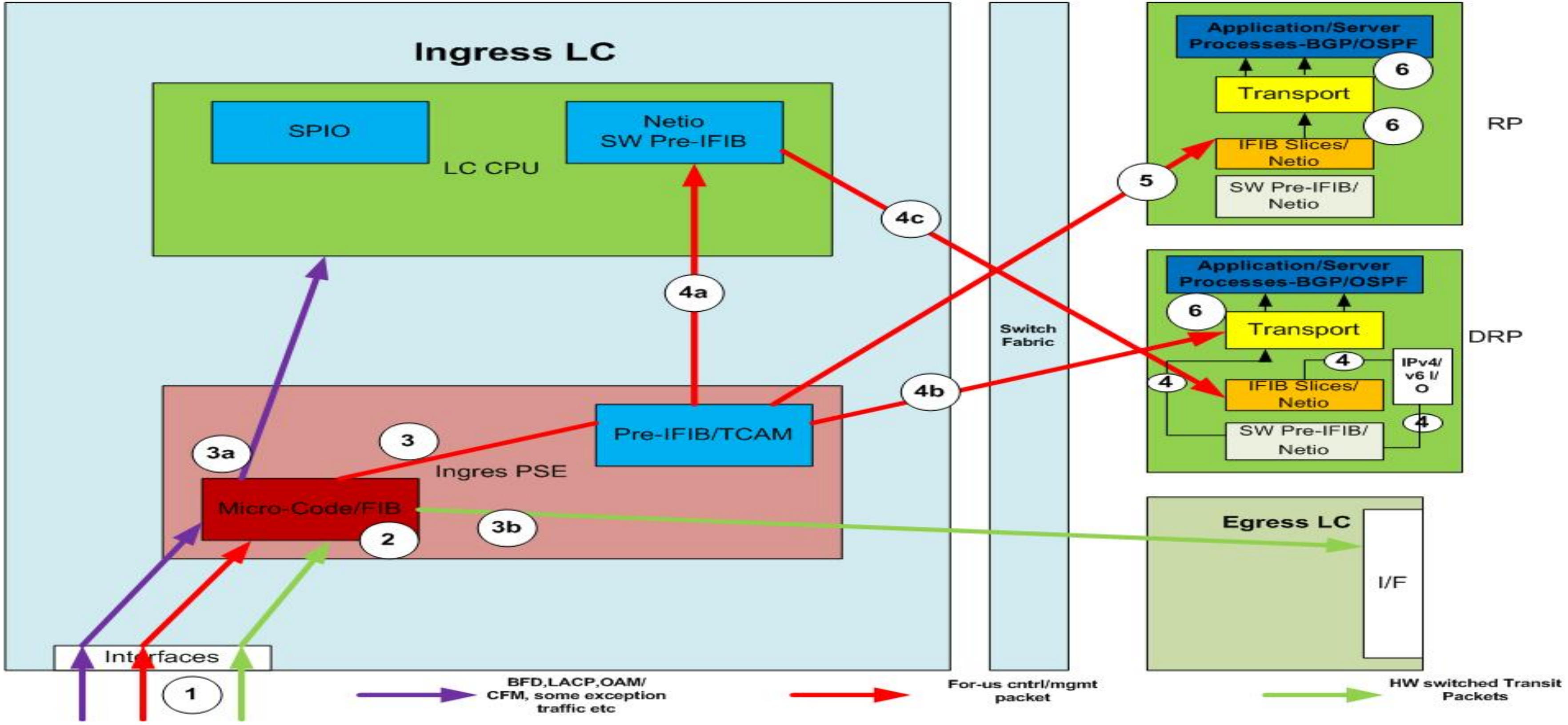
```
control-plane
 management-plane
  inband
   interface Loopback87
    allow SNMP
   !
   interface
GigabitEthernet0/7/1/0
    allow SSH
    allow Telnet
   !
   interface
GigabitEthernet0/7/1/3
    allow Telnet peer
     address ipv4 3.3.3.3
     address ipv4 5.5.5.0/2
    !
   !
  !
 !
!
```

```
RP/0/RP0/CPU0:CRS1-4#show lpts bindings brief | i (any.23 )
 0/RP0/CPU0 TCP  LR IPV4 TCP   default  any.23 any             Mg0/RP0/CPU0/0
 0/RP0/CPU0 TCP  LR IPV4 TCP   default  any.23 any             Gi0/7/1/0
 0/RP0/CPU0 TCP  LR IPV4 TCP   default  any.23 3.3.3.3         Gi0/7/1/3
 0/RP0/CPU0 TCP  LR IPV4 TCP   default  any.23 5.5.5.0/28      Gi0/7/1/3
RP/0/RP0/CPU0:CRS1-4#
RP/0/RP0/CPU0:CRS1-4#show lpts bindings brief | i (any.161 )
 0/RP0/CPU0 UDP  LR IPV4 UDP   default  any.161 any            Mg0/RP0/CPU0/0
 0/RP0/CPU0 UDP  LR IPV4 UDP   default  any.161 any            Lo87
RP/0/RP0/CPU0:CRS1-4#
RP/0/RP0/CPU0:CRS1-4#show lpts bindings brief | i (any.22 )
```

```
RP/0/RP0/CPU0:CRS1-4#show lpts pifib hardware entry bri location 0/7/cpu0 | i (.23 )
 (def).23 3.3.3.3                 TCP   GigabitEthernet0/7/1/3 0/RP0/CPU0   24     7
 (def).23 5.5.5.0/28              TCP   GigabitEthernet0/7/1/3 0/RP0/CPU0   24     7
 (def).23 any                     TCP   GigabitEthernet0/7/1/0 0/RP0/CPU0   24     7
 (def).23 10.10.20.100.33732      TCP   any         0/RP0/CPU0   24     6
 (def).23 10.10.20.100.53964      TCP   any         0/RP0/CPU0   24     6
RP/0/RP0/CPU0:CRS1-4#
RP/0/RP0/CPU0:CRS1-4#
RP/0/RP0/CPU0:CRS1-4#show lpts pifib hardware entry bri location 0/0/cpu0 | i (.23 )
 (def).23 10.10.20.100.33732      TCP   any         0/RP0/CPU0   24     6
 (def).23 10.10.20.100.53964      TCP   any         0/RP0/CPU0   24     6
RP/0/RP0/CPU0:CRS1-4#
```

Packet flow

"for-us" control/management plane traffic entering LC interfaces



Legend to previous slide

2. Ingress NPU in the LC will perform packet lookup using the HW FIB to determine how to switch the packet.

3. If FIB lookup determines that this is a “for-us” control/management plane packet, then further lookup has to be performed on the pre-IFIB table in the HW/TCAM to match it against a flow entry, perform policing on the packet stream, and ascertain the node/element and application to deliver

3a. If the incoming packet is of L2 type such as CDP, ARP, LACP PDU, BFD, CFM/OAM etc FIB will punt them to LC CPU for further processing. Also transit traffic to be forwarded, but frag required Packets with DF bit set packets, IP options packet, packets with RA, transit traffic dropped by ACL etc will be punted to LC CPU

3b. If the incoming packet is part of transit traffic, they will be switched by the LC HW and sent to the egress LC through the fabric

4a. For some of the “for-us” control packets, which needs to be delivered locally, requiring special handling such as ICMP echo, TTL expired packets, , HW Pre-IFIB look-up will punt the packets to LC CPU

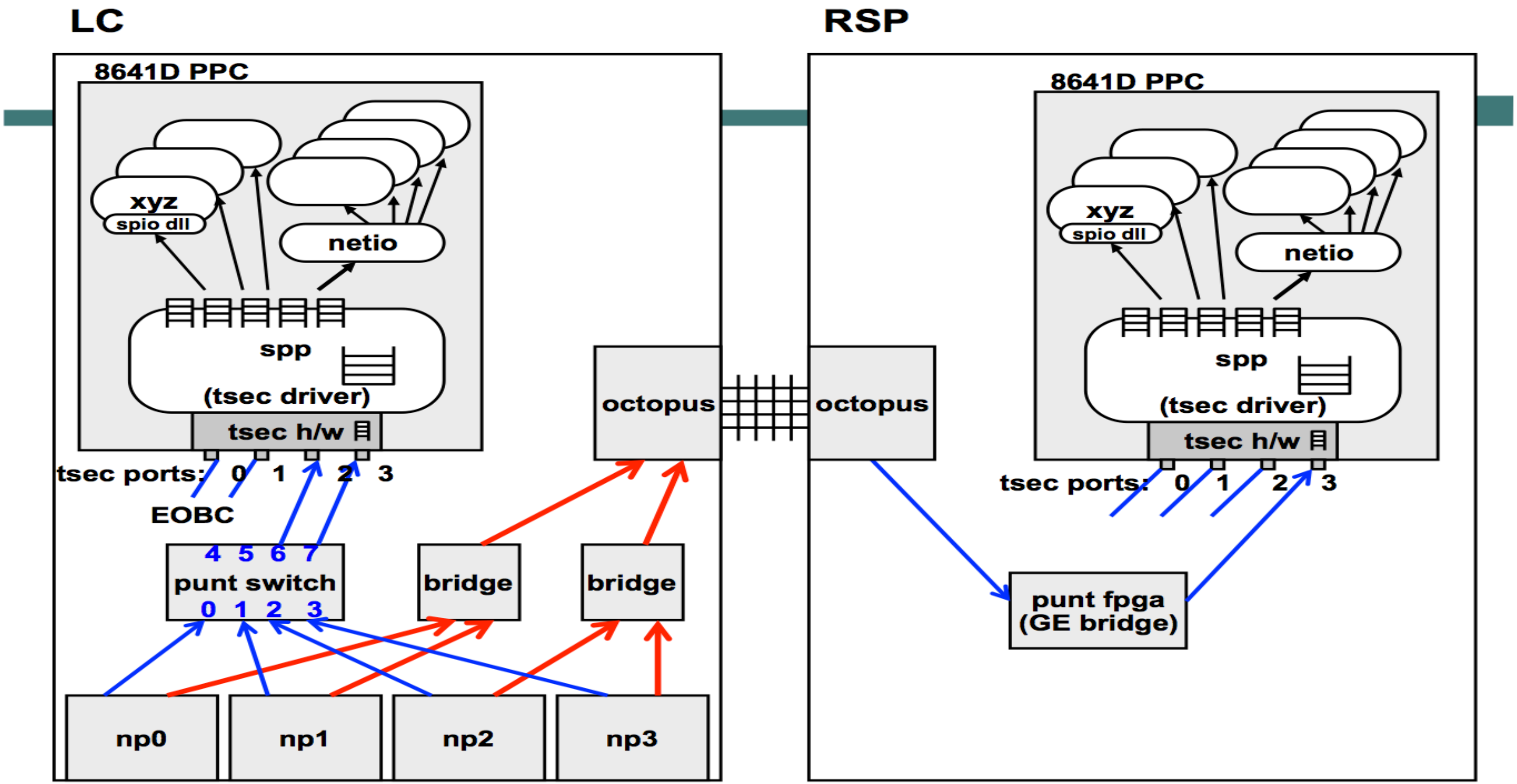
4b. LC HW Pre-IFIB look up may be a trivial one, meaning it will have all the information to deliver the “for-us” packets to the right application in the right node/element.

4c. Fragmented “for-us” control/management plane packets will be punted to LC CPU/SW pre-ifib lookup, they have to be re-assembled first only after that pre-IFIB lookup can be performed. LC SW pre-ifib will pick a re-assembly servers (RP/DRP netio), which in turn will sent to appropriate I/O (Ipv4 _io or v6_io). Reassembled packets will be sent to pre-ifib for further look-up and will be delivered accordingly to the right node/element (be it local or remote node accordingly)

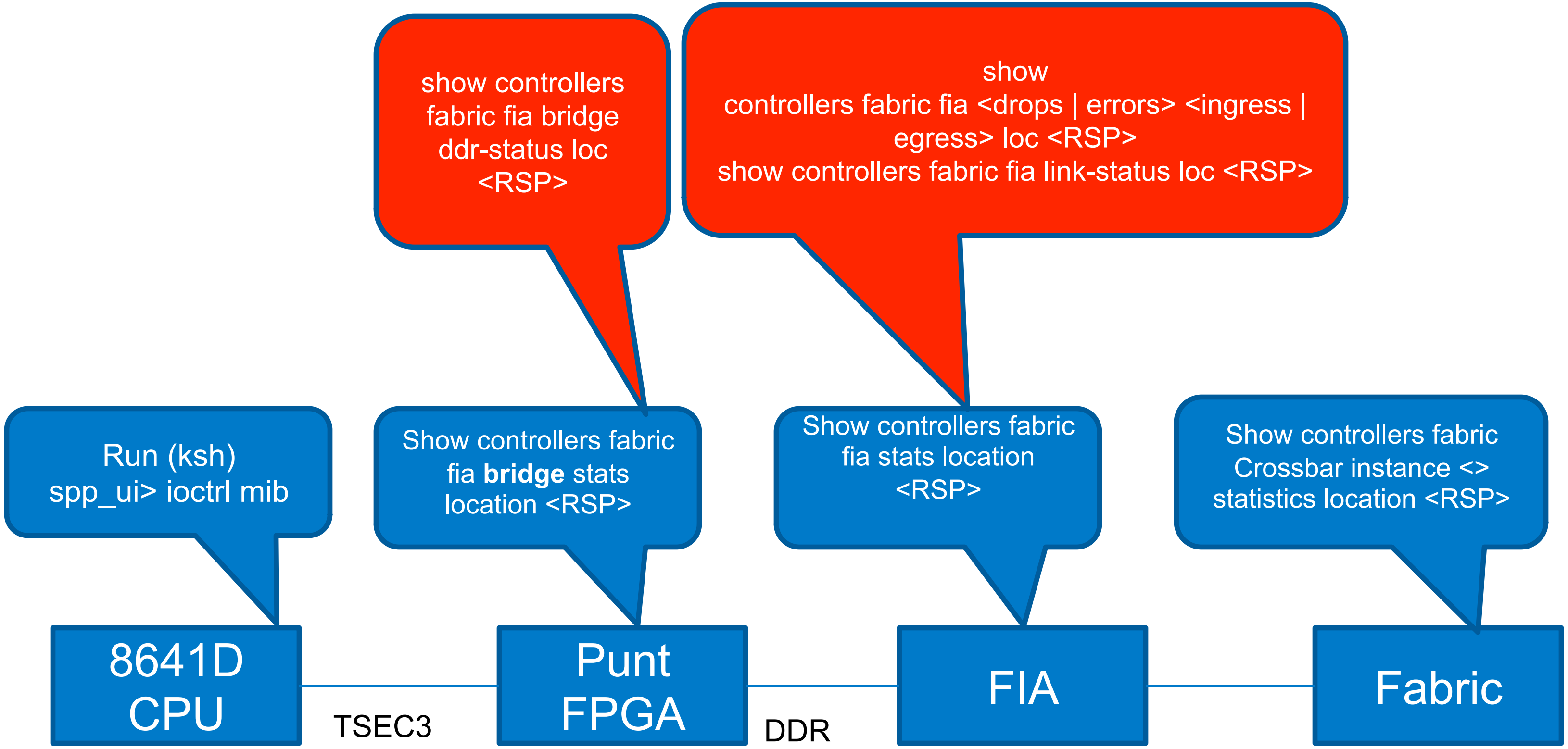
5. For some of the “for-us” packets, which needs complex, flow match, HW Pre-IFIB will send the packets for IFIB slice lookup in flow manager process running in RP/DRP.

6. IFIB slice lookup on a local node will provide transport and the associated application/server processes the packet needs to be delivered

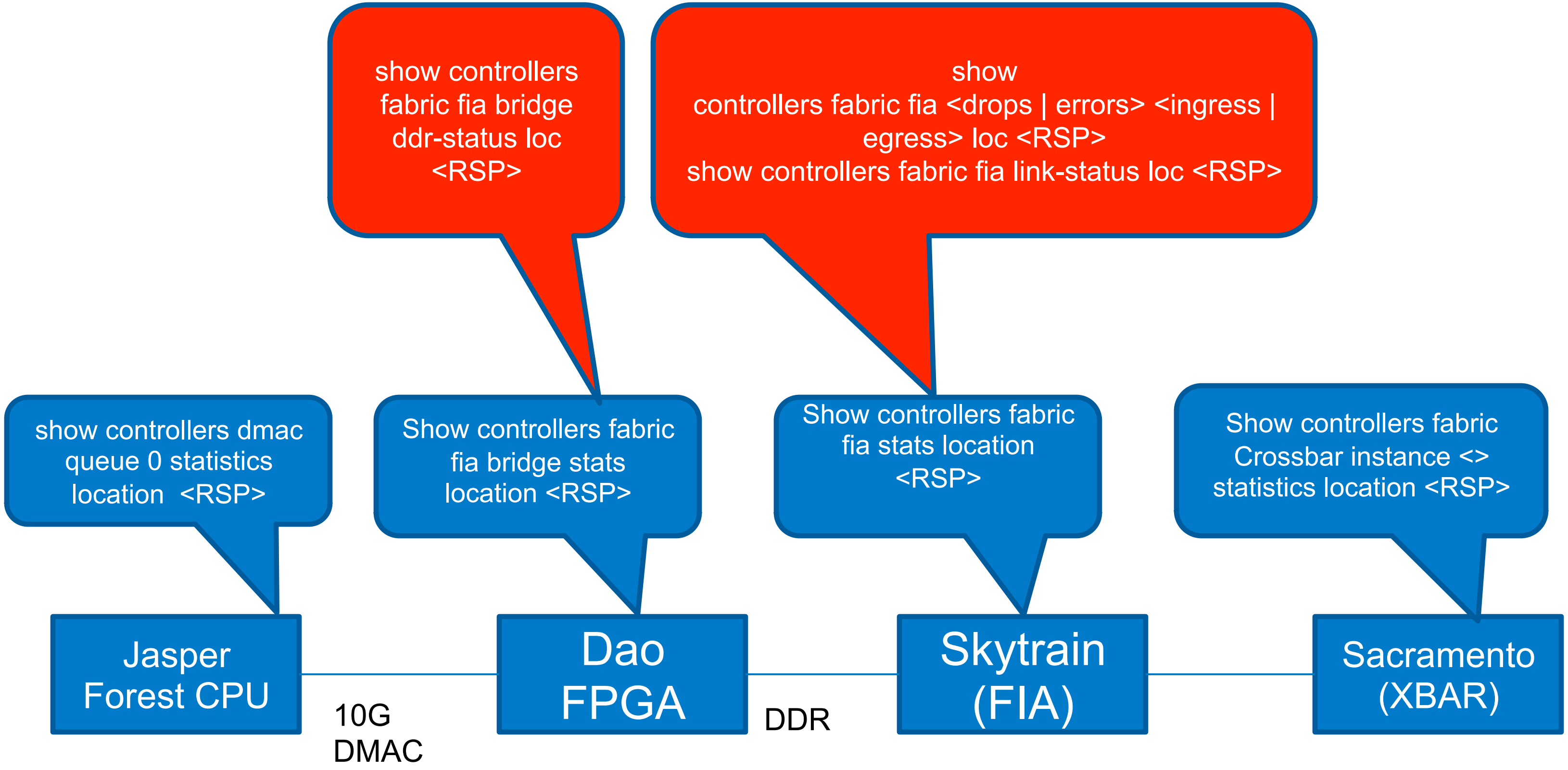
Detailed packet path of for-us packets



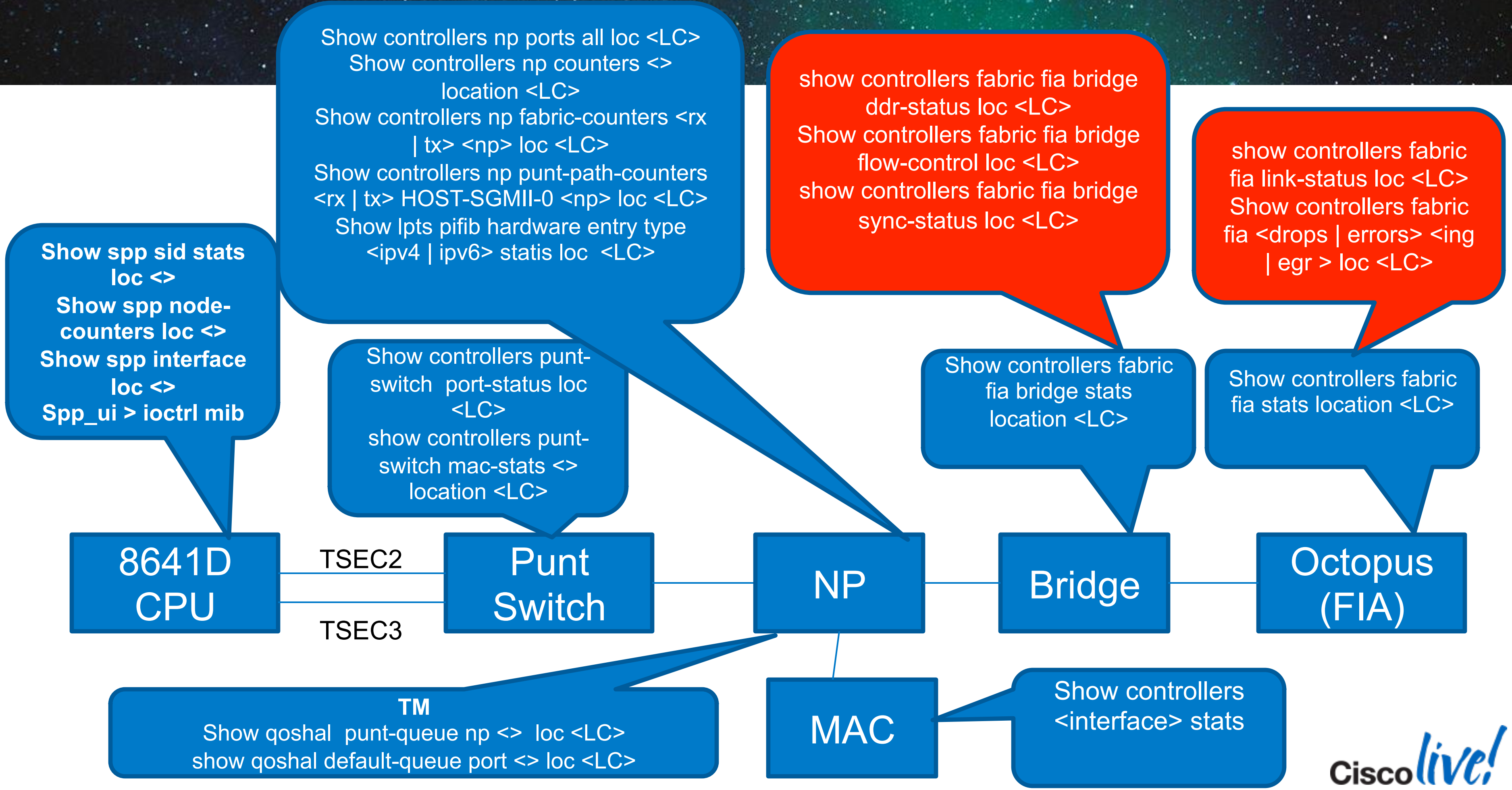
RSP2


show controllers fabric fia bridge ddr-status loc <RSP>
show controllers fabric fia <drops | errors> <ingress | egress> loc <RSP>
show controllers fabric fia link-status loc <RSP>
Run (ksh) spp_ui> ioctrl mib
Show controllers fabric fia **bridge** stats location <RSP>
Show controllers fabric fia stats location <RSP>
Show controllers fabric Crossbar instance <> statistics location <RSP>
8641D CPU
TSEC3
Punt FPGA
DDR
FIA
Fabric

RSP440

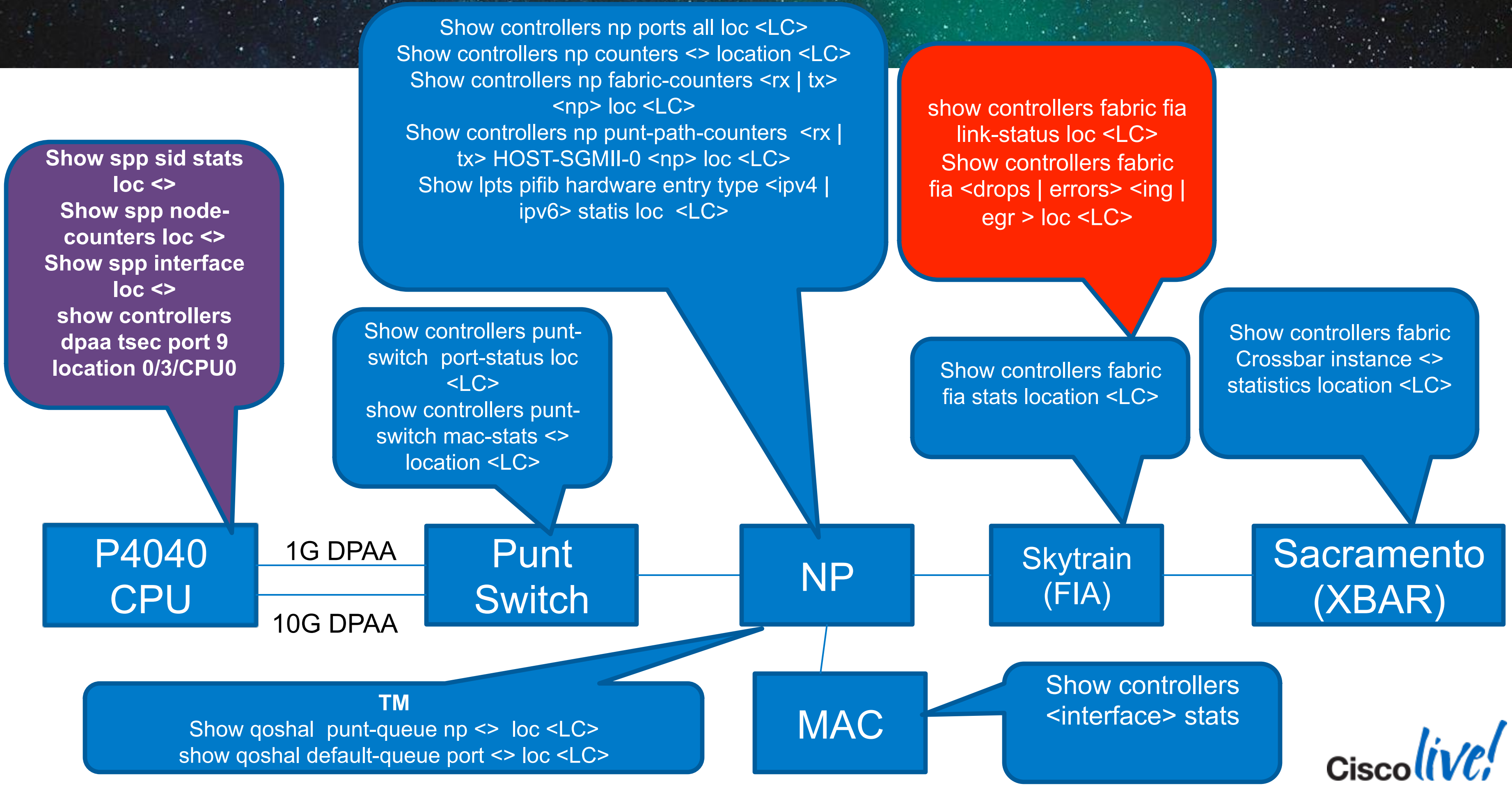

show controllers fabric fia bridge ddr-status loc <RSP>
show controllers fabric fia <drops | errors> <ingress | egress> loc <RSP>
show controllers fabric fia link-status loc <RSP>
show controllers dmac queue 0 statistics location <RSP>
Show controllers fabric fia bridge stats location <RSP>
Show controllers fabric fia stats location <RSP>
Show controllers fabric Crossbar instance <> statistics location <RSP>
Jasper Forest CPU
10G DMAC
Dao FPGA
DDR
Skytrain (FIA)
Sacramento (XBAR)

Trident LC



Show controllers np ports all loc <LC>
Show controllers np counters <> location <LC>
Show controllers np fabric-counters <rx | tx> <np> loc <LC>
Show controllers np punt-path-counters <rx | tx> HOST-SGMII-0 <np> loc <LC>
Show lpts pifib hardware entry type <ipv4 | ipv6> statis loc <LC>
show controllers fabric fia bridge ddr-status loc <LC>
Show controllers fabric fia bridge flow-control loc <LC>
show controllers fabric fia bridge sync-status loc <LC>
show controllers fabric fia link-status loc <LC>
Show controllers fabric fia <drops | errors> <ing | egr > loc <LC>
Show spp sid stats loc <>
Show spp node-counters loc <>
Show spp interface loc <>
Spp_ui > ioctrl mib
Show controllers punt-switch port-status loc <LC>
show controllers punt-switch mac-stats <> location <LC>
Show controllers fabric fia bridge stats location <LC>
Show controllers fabric fia stats location <LC>
8641D CPU
TSEC2
TSEC3
Punt Switch
NP
Bridge
Octopus (FIA)
TM
Show qoshal punt-queue np <> loc <LC>
show qoshal default-queue port <> loc <LC>
MAC
Show controllers <interface> stats

Typhoon LC



Show controllers np ports all loc <LC>
Show controllers np counters <> location <LC>
Show controllers np fabric-counters <rx | tx> <np> loc <LC>
Show controllers np punt-path-counters <rx | tx> HOST-SGMII-0 <np> loc <LC>
Show lpts pifib hardware entry type <ipv4 | ipv6> statis loc <LC>
show controllers fabric fia link-status loc <LC>
Show controllers fabric fia <drops | errors> <ing | egr > loc <LC>
Show spp sid stats loc <>
Show spp node-counters loc <>
Show spp interface loc <>
show controllers dpaa tsec port 9 location 0/3/CPU0
Show controllers punt-switch port-status loc <LC>
show controllers punt-switch mac-stats <> location <LC>
Show controllers fabric fia stats location <LC>
Show controllers fabric Crossbar instance <> statistics location <LC>
P4040 CPU
1G DPAA
10G DPAA
Punt Switch
NP
Skytrain (FIA)
Sacramento (XBAR)
TM
Show qoshal punt-queue np <> loc <LC>
show qoshal default-queue port <> loc <LC>
MAC
Show controllers <interface> stats

RO (Trident) vs XMEN (Typhoon) LC

Item	RO LC	XMEN LC	
CPU Port	TSEC (2x1G): TSEC2 / TSEC3	DPAA (1x10G)	RO LC: spp_ui> ioctrl mib (clear on Read) XMEN LC: show controllers dpaa tsec port 9 location <>
Punt Switch	10 port / 16 port (1G) Port7: TSEC2 Port8: TSEC3 Port[0..(N-1)]: NP [0… (N-1)] (exception 8 NP LC)	24x1G + 2x10G Port24: 10G DPAA Port10: 1G DPAA Port [0… (N-1)]: NP [0… (N-1)]	Show controllers punt-switch mac-stats <> location <>
NP	Trident	Typhoon	Show controllers np ports all location <> Show controllers np fabric-counters <rx | tx> <np> location <> Show controllers np counters <np> location <>
FIA	Octopus	Skytrain	Show controllers fabric fia statistics location <>
Bridge	Punt	N.A (integrated into Skytrain	Show controllers fabric bridge stats loc <>
Fabric (XBAR)	N.A	Sacramento	Show controllers fabric Crossbar instance <> statistics location <>

LPTS

- Local Packet Transport System
  - Pre-IFIB packet processing (for-us packets)
  - Control plane for Control packets
- L3 applications on RSP responsible for triggering / installation of the LPTS entries
- LPTS entries are installed in software (on the local CPU) and in hardware (TCAM)
- 3 categories
  - Default entries (TCAM) : L3
  - Dynamic entries (TCAM) : L3
  - Static entries (NP SRAM) : L2 / internal interest
- "show lpts pifib hardware entry type <ipv4 | ipv6> brief location <LC>
- "show lpts pifib hardware entry type <ipv4 | ipv6> statistics location <LC>"
- "show prm server tcam …."
- show lpts pifib hardware static-police location <LC>
  - Displays the Static punt table stats

(PRM is platform resource manager, the entity that controls the hw programming between CPU nad NPU+its attached asics/memory)

Netio Tx on RSP (process switching)

- "show netio idb fint location RSP" (4.1.0 onwards)
- "show netio idb all brief location RSP" (prior to 4.1.0 to identify the interface in question)
- "show netio idb ifhandle <> location RSP" (prior to 4.1.0 based on the ifhandle in question)
- "show netio drops location RSP"
- "run"
  - "fwd_netio_debug" [stats counters / error counters / last 64 dropped packets (PD headers + initial part of payload) logged]
- "debug netio drivers location RSP" [filter packets going to fabric]

SPP Tx on RSP (software packet path IntX switching)

- Look in the following order
  - "show spp client location RSP"
    - Look for the very 1st queue which belongs to SPP and is used by clients to place messages to SPP.
      - Messages have super-frames in case of Packet Inject case.
- "show spp graph location RSP"
- "show spp sid stats location RSP"
  - Not useful for non-SPIO injects in Tx direction.
  - Typically used by all clients in the Rx direction.
- "show spp node-counters location RSP" and "show spp node location RSP"
- "show spp interface location RSP"
- "run"
  - "spp_ui"
    - "ioctrl mib" [RFC1213 MIB counters]; Clear on Read; Look for Tx stats
- "run"
  - "spp_ui"
    - "help"
    - "help trace" [gateway to tracing packets]
    - "help <>"
    - "trace filter node <>" ➔ Use the appropriate Tx node (inject or tx)
    - "trace filter set …."
    - "trace start 100"
    - "trace stop"
    - "trace ascii save" [ASCII]
    - "trace save" [PCAP]
    - "trace filter clear"
    - "trace filter show"
    - trace filter node all ➔ match on all SPP nodes
    - Mainly look for correct VQI / Fabric mcast bit for sent packets to ensure that they land on the correction destination card.
- "clear spp client location RSP"
- "clear spp node-counters location RSP"
- "clear spp interface location RSP"
- "show spp buffer location RSP"
- "show spp clientlib trace location RSP"
- "show spp trace [error | event] location RSP"

SPP Rx on RSP

- “run”
  - “spp_ui”
    - “ioctrl mib” [RFC1213 MIB counters]; Clear on Read; Look for Rx stats
- “show spp interface location RSP”
- “show spp node-counters location RSP”
- “show spp node location RSP”
- “show spp sid stats location RSP”
  - Updated by the classification node based on SID lookup
- “show spp client location RSP”
- “show spp buffer location RSP”
- “run”
  - “spp_ui”
    - “buffer allocs” ➔ Look for leaked buffers.
- “show spp graph location RSP”
- “run”
  - “spp_ui”
    - “trace….” ➔ Look for “classify” or “punt” or “drop” nodes
    - Note that “trace filter node “tsec3/rx” is not allowed as Packet capture at this node is not possible currently; “tsec3/classify” is the very 1st trace-able node in the Rx direction

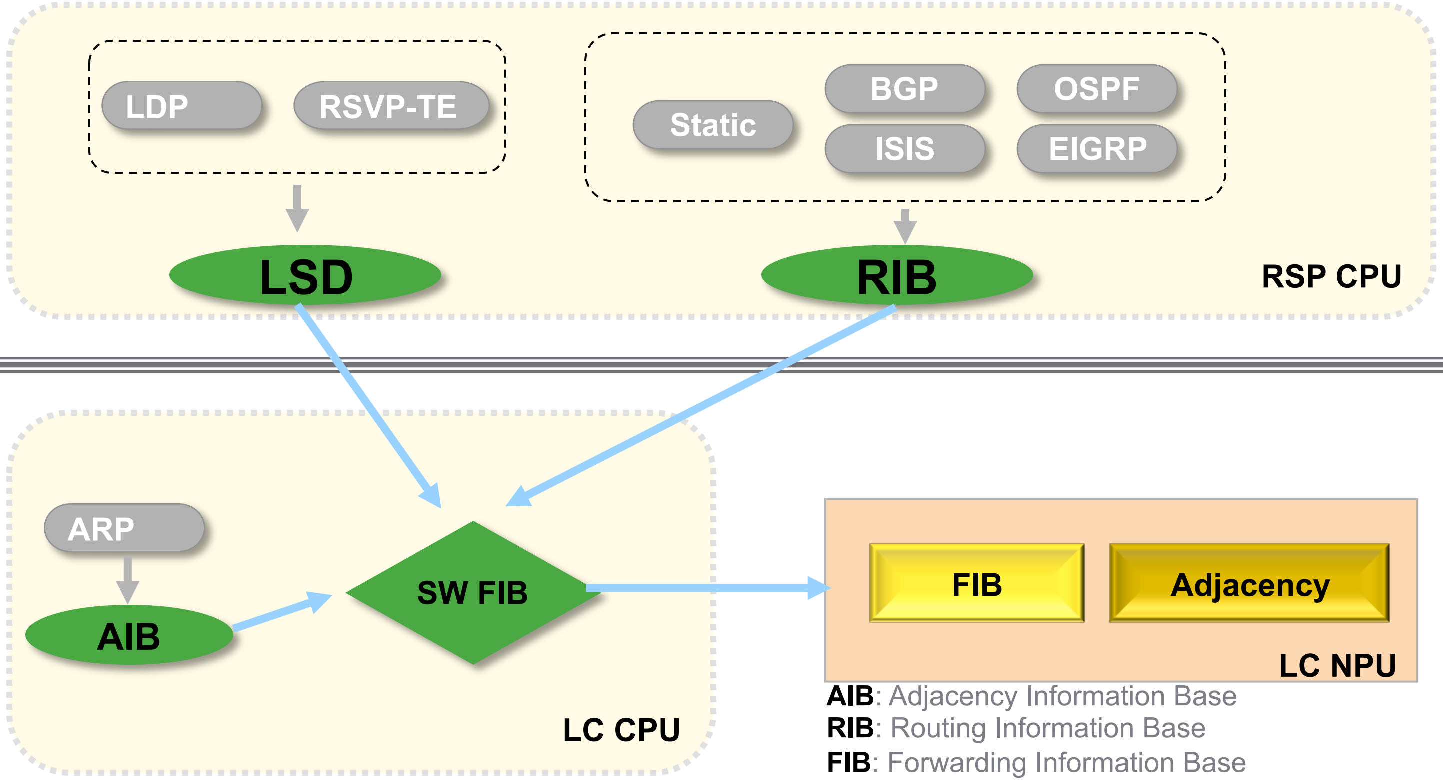
Netio Rx on RSP

- "show netio idb fint location RSP" (4.1.0 onwards)
- "show netio idb all brief location RSP" (prior to 4.1.0 to identify the interface in question)
- "show netio idb ifhandle <> location RSP" (prior to 4.1.0 based on the ifhandle in question)
- "show netio drops location RSP"
- "run"
  - "fwd_netio_debug" [stats counters / error counters / last 64 dropped packets (PD headers + initial part of payload) logged]
- "debug netio drivers location RSP" [filter packets coming in from fabric]
- "debug lpts packet…" [for debugging packets of type PKT_LPTS]; use "drops", "detail", "errors", etc.

Punt FPGA (on RSP)

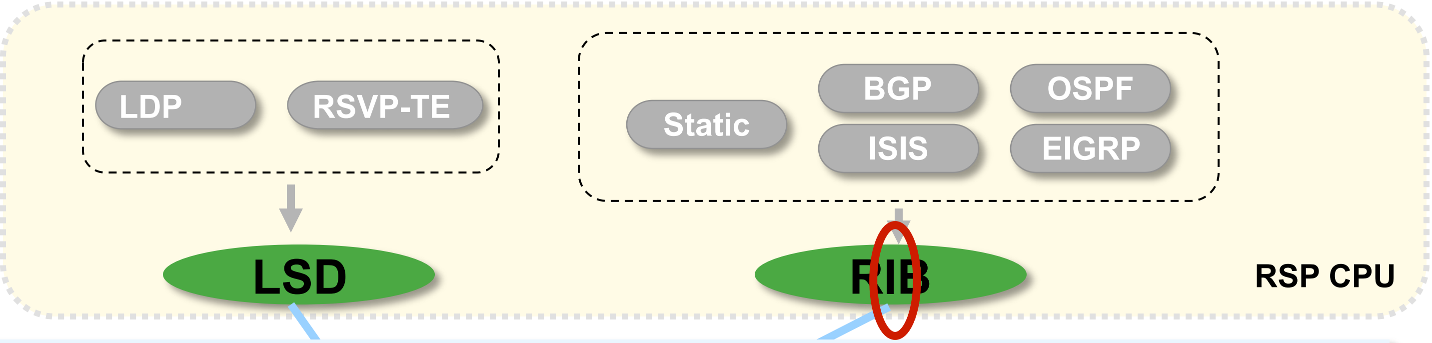
- “show controllers fabric fia bridge..” on RSP
  - Not all CLI sub-options applicable to RSP
- Use the following sub-options
  - “ddr-status” [look for SYNC status]
  - “stats”
  - “flow-control”
- “clear controller fabric fia loc RSP”
  - Clears all of Punt FPGA, FIA counters on RSP
- “admin” mode: “show hw-module fpd location RSP”
  - Look for any mismatches and need for up-grade/down-grade.
  - Most likely issue of drops in hardware is due to FPD change requirements.

L3 IPv4 Control Plane Architecture



AIB: Adjacency Information Base
RIB: Routing Information Base
FIB: Forwarding Information Base
LSD: Label Switch Database

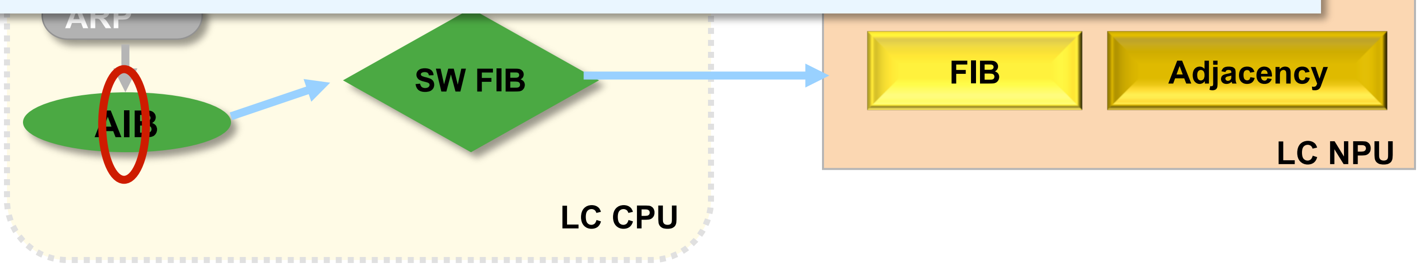
L3 IPv4 Control Plane Architecture
Show commands




```
RP/0/RSP0/CPU0:asr#sh route 222.0.0.6/31

Routing entry for 222.0.0.6/31
  Known via "isis isis1", distance 115, metric 20, type level-1
  Installed Mar  2 17:58:12.251 for 00:00:47
  Routing Descriptor Blocks
    222.0.0.2, from 222.2.2.1, via TenGigE0/1/0/3
      Route metric is 20
  No advertising protos.
```

L3 IPv4 Control Plane Architecture
Show commands

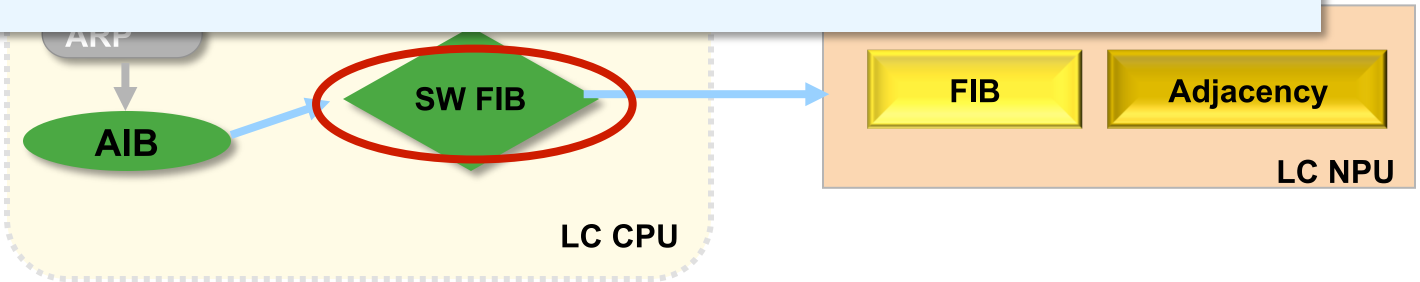
```
RP/0/RSP0/CPU0:asr#show adjacency summary location 0/1/CPU0

Adjacency table (version 26) has 19 adjacencies:
     11 complete adjacencies
      8 incomplete adjacencies
      0 deleted adjacencies in quarantine list
      8 adjacencies of type IPv4
          8 complete adjacencies of type IPv4
          0 incomplete adjacencies of type IPv4
          0 deleted adjacencies of type IPv4 in quarantine list
          0 interface adjacencies of type IPv4
          4 multicast adjacencies of type IPv4
```



L3 IPv4 Control Plane Architecture

Show commands

```
RP/0/RSP0/CPU0:viking-1#sh cef 222.0.0.6 location 0/1/CPU0
222.0.0.6/31, version 1, internal 0x40000001
Updated Mar  2 17:58:11.987
 local adjacency 222.0.0.2
 Prefix Len 31, traffic index 0, precedence routine (0)
   via 222.0.0.2, TenGigE0/1/0/3, 5 dependencies, weight 0, class 0
    next hop 222.0.0.2
    local adjacency
```

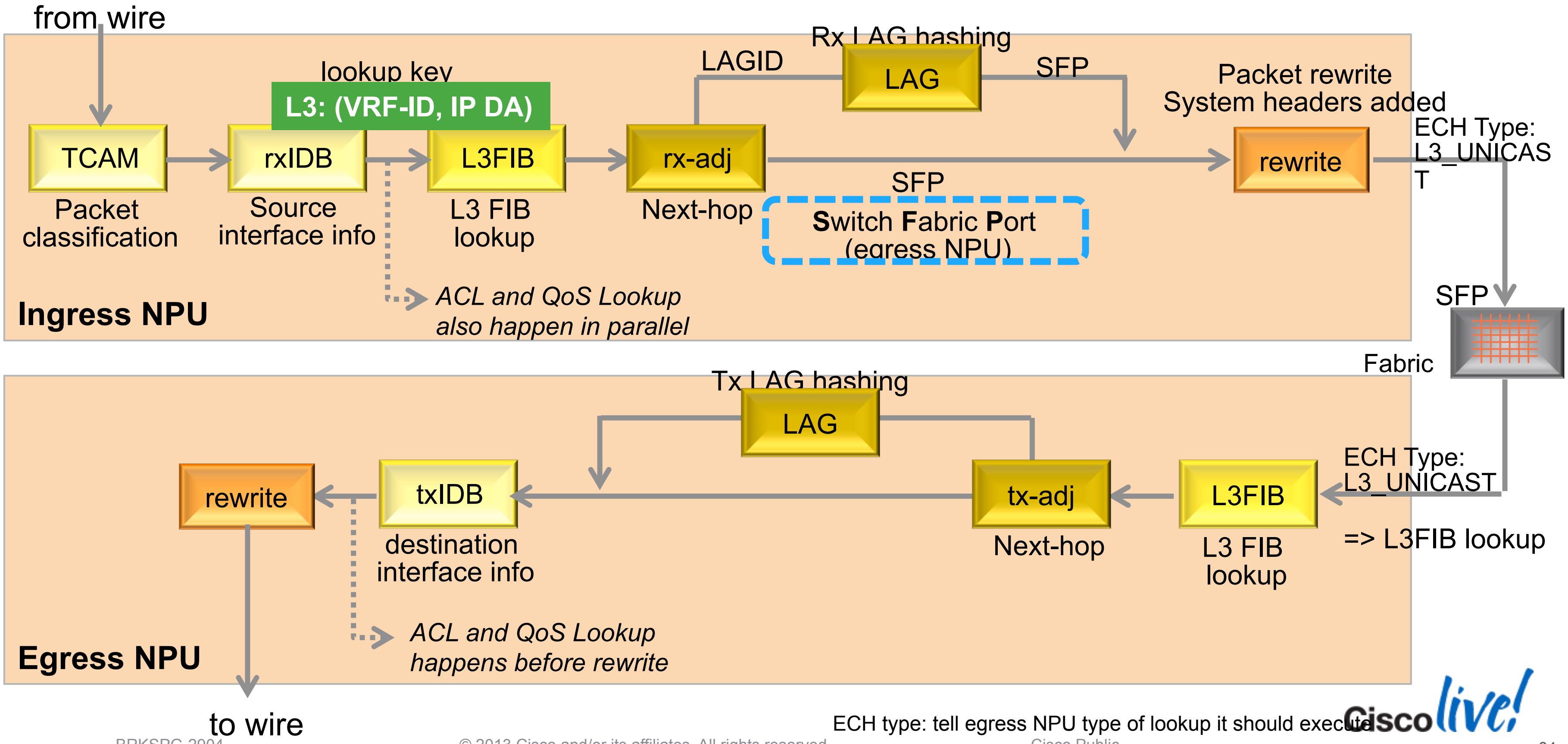


L3 IPv4 Control Plane Architecture
Show commands

```
RP/0/RSP0/CPU0:asr#sh cef 222.0.0.6 hardware ingress lo 0/1/CPU0
222.0.0.6/31, version 1, internal 0x40000001 (0xb1d66c6c) [1], 0x0 (0xb1b4f758), 0x0 (0x0)
 Updated Mar  2 17:58:11.987
 local adjacency 222.0.0.2
 Prefix Len 31, traffic index 0, precedence routine (0)
   via 222.0.0.2, TenGigE0/1/0/3, 5 dependencies, weight 0, class 0
    next hop 222.0.0.2
    local adjacency

        EZ:0 Leaf
        ============
  Search ctrl-byte0:    0x3  ctrl-byte1:    0x8  ctrl-byte2:0x5

  Leaf Action :     FORWARD

prefix length :     31

Search Control Flags :

   match          :   1       valid:   1
   done           :   0       ifib_lookup:   0
   ext_lsp_array :   0       match_all_bit:   0
   recursive      :   0       nonrecursive  :   1
   default_action:   1

  Non Recursive Leaf:
  -------------------

   ldi ptr   : 10936 (0x2ab8)       igp statsptr:0
  rpf ptr :    0x0000
  BGP policy a/c :     0       AS number :     0
```

AIB: Adjacency Information Base
RIB: Routing Information Base
FIB: Forwarding Information Base
LSD: Label Switch Database

L3 Unicast Forwarding
Packet Flow (Simplified)



ECH type: tell egress NPU type of lookup it should execute

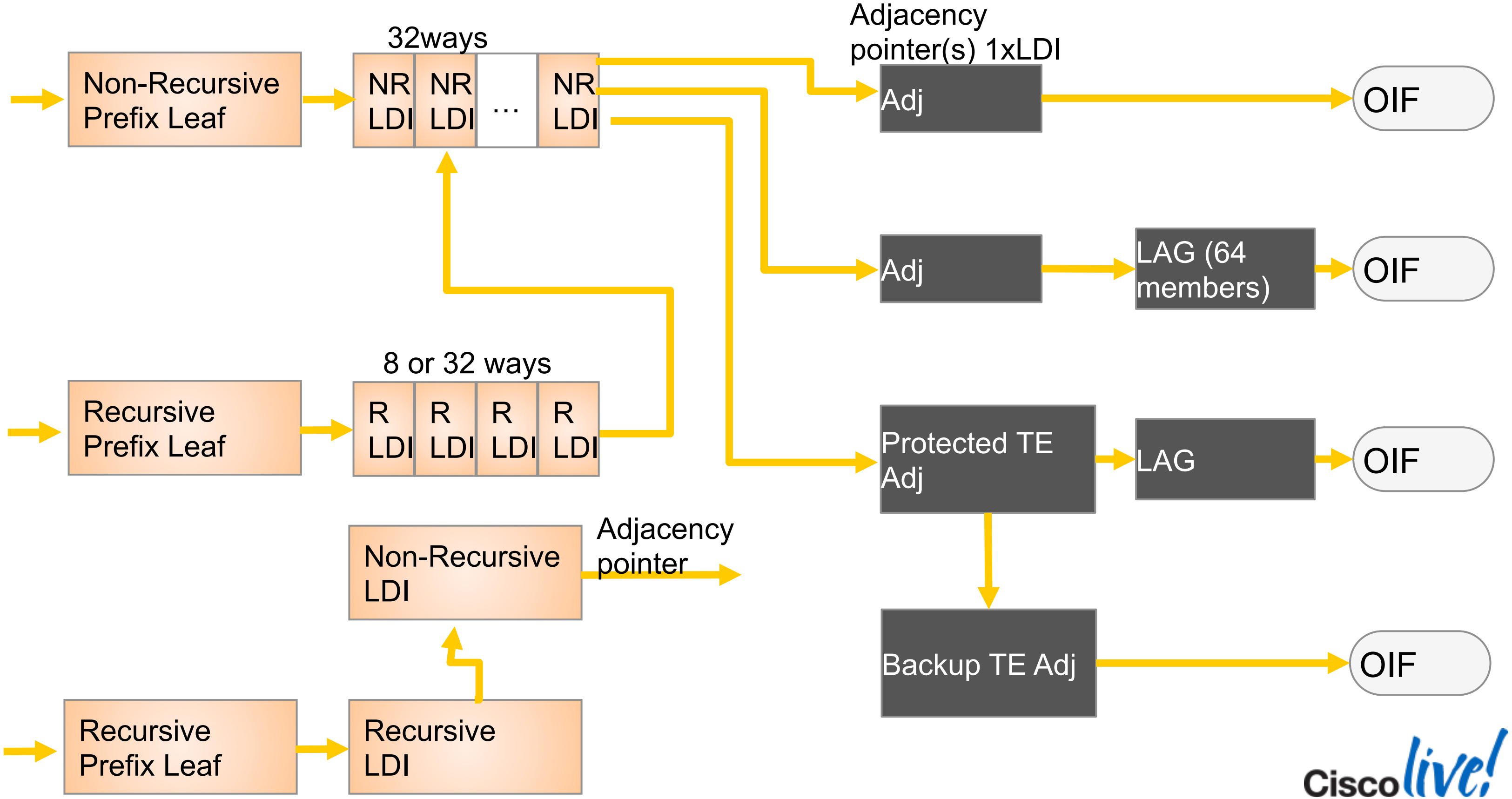
Switch Fabric Port

```
RP/0/RSP1/CPU0:asr#sh controllers pm interface gig 0/0/0/1 loc 0/0/CPU0
Ifname(1): GigabitEthernet0_0_0_1, ifh: 0x40000c0 :
iftype             0xf
egress_uidb_index  0x3
ingress_uidb_index 0x3
port_num           0x1
phy_port_num       0x1
channel_id         0x3
lag_id             0x0
virtual_port_id    0x0
switch_fabric_port 0x3
```

Ports connected to the same NPU share the same SFP value

L3 NPU IPv4 FIB Architecture

L3FIB



ECMP Load balancing

IPv6 uses first 64 bits in 4.0 releases, full 128 in 42 releases

A: IPv4 Unicast or IPv4 to MPLS (3)

– No or unknown Layer 4 protocol: IP SA, DA and Router ID

– UDP or TCP: IP SA, DA, Src Port, Dst Port and Router ID

B: IPv4 Multicast

– For (S,G): Source IP, Group IP, next-hop of RPF

– For (*,G): RP address, Group IP address, next-hop of RPF

C: MPLS to MPLS or MPLS to IPv4

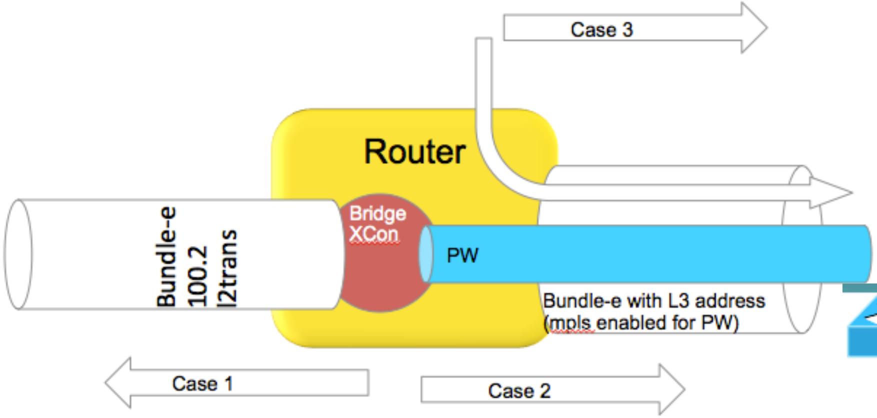
– # of labels <= 4 : same as IPv4 unicast (if inner is IP based, EoMPLS, etherheader will follow: 4th label+RID)

– # of labels > 4 : 4th label and Router ID

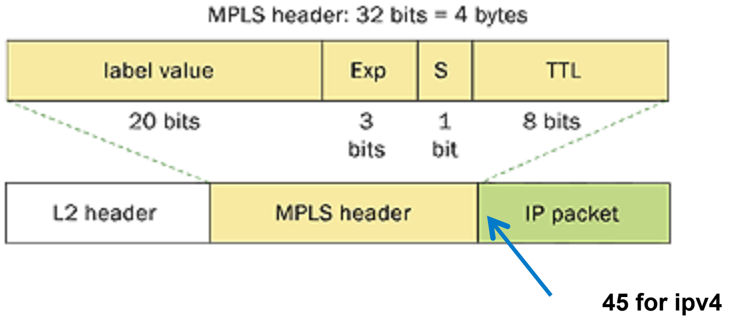
Bundle Load balancing

- (3) L3 bundle uses 5 tuple as "1" (eg IP enabled routed bundle interface)
- (3) MPLS enabled bundle follows "C"
- (1) L2 access bundle uses access S/D-MAC + RID, OR L3 if configured (under l2vpn)
- (2) L2 access AC to PW over mpls enabled core facing bundle uses PW label (not FAT-PW label even if configured)
    - FAT PW label only useful for P/core routers

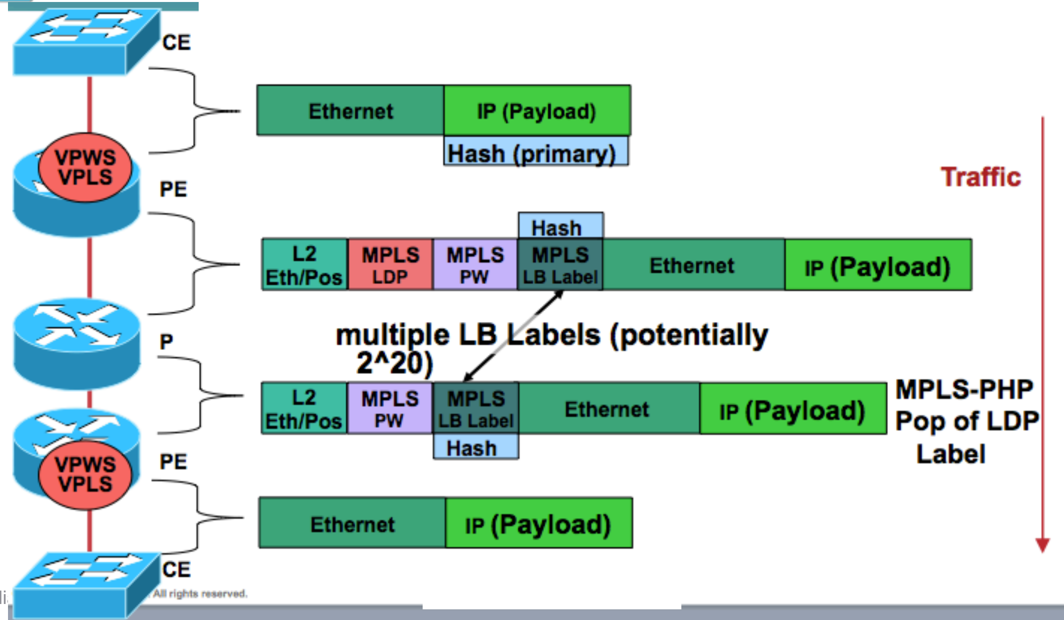
Load-balancing scenarios



MPLS/IP protocol stack



EoMPLS protocol stack

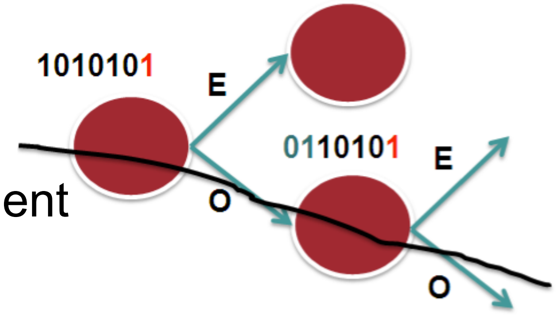


MPLS vs IP Based loadbalancing

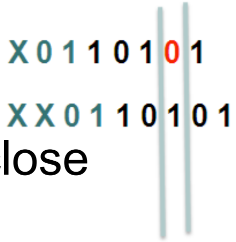
- When a labeled packet arrives on the interface.
- The ASR9000 advances a pointer for at max 4 labels.
- If the number of labels <=4 and the next nibble seen right after that label is
  - 4: default to IPv4 based balancing
  - 6: default to IPv6 based balancing
- This means that if you have a P router that has no knowledge about the MPLS service of the packet, that nibble can either mean the IP version (in MPLS/IP) or it can be the DMAC (in EoMPLS).
- RULE: If you have EoMPLS services AND macs are starting with a 4 or 6. You HAVE to use Control-Word

- Control Word inserts additional zeros after the inner label showing the P nodes to go for label based balancing.
- In EoMPLS, the inner label is VC label. So LB per VC then. More granular spread for EoMPLS can be achieved with FAT PW (label based on FLOW inserted by the PE device who owns the service

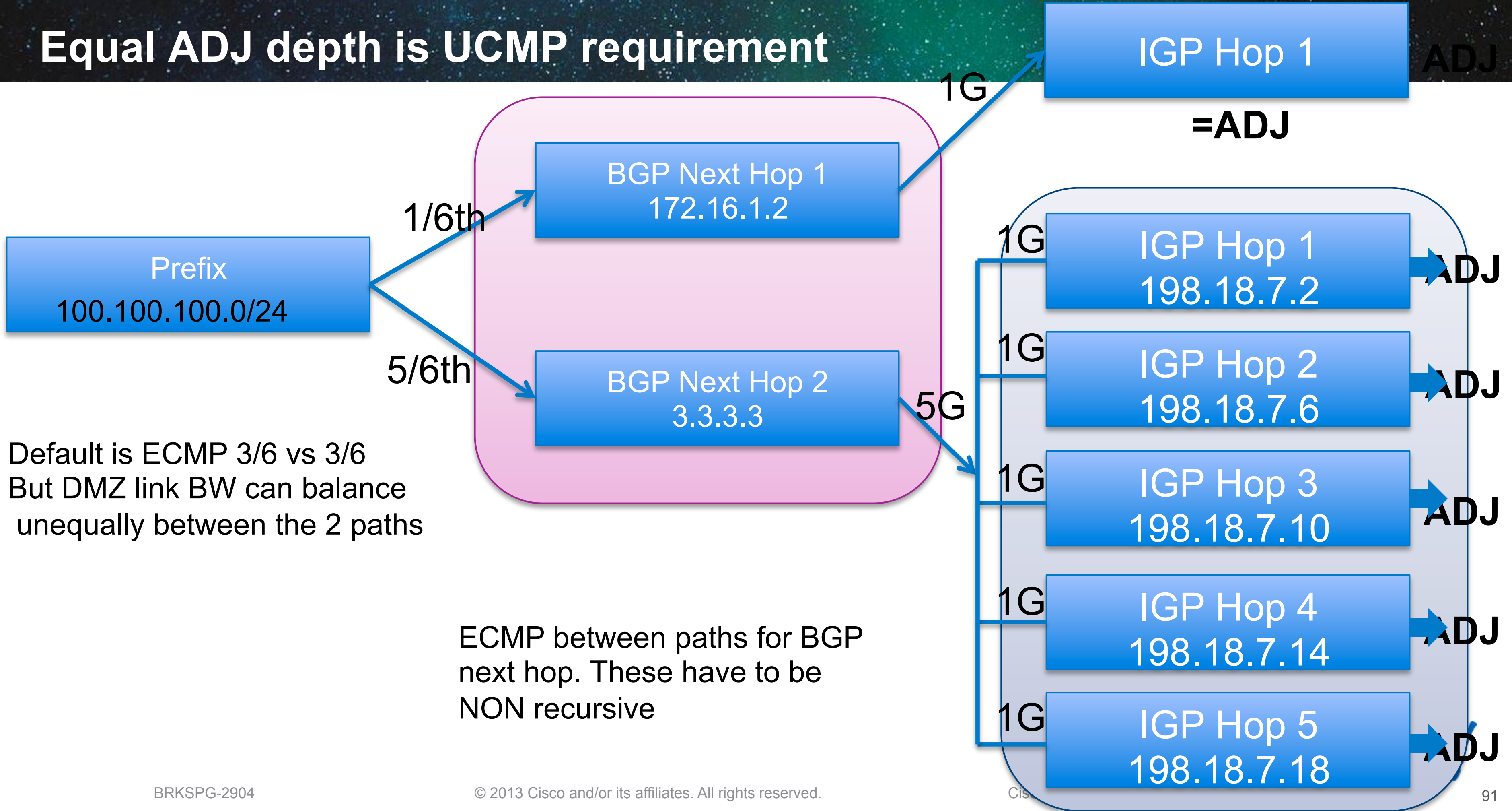
Loadbalancing ECMP vs UCMP and polarization



- Support for Equal cost and Unequal cost
- 32 ways for IGP paths
- 32 ways (Typhoon) for BGP (recursive paths) 8-way Trident
- 64 members per LAG
- Make sure you reduce recursiveness of routes as much as possible (static route misconfigurations…)
- All loadbalancing uses the same hash computation but looks at different bits from that hash.
- Use the hash shift knob to prevent polarization.
- Adj nodes compute the same hash, with little variety if the RID is close
  - This can result in north bound or south bound routing.
  - Hash shift makes the nodes look at complete different bits and provide more spread.
  - Trial and error… (4 way shift trident, 32 way typhoon, values of >5 on trident result in modulo)



Equal ADJ depth is UCMP requirement



Default is ECMP 3/6 vs 3/6
But DMZ link BW can balance
unequally between the 2 paths

ECMP between paths for BGP
next hop. These have to be
NON recursive

Show CEF output for loadbalancing

Unequal adj depth breaks loadbalancing capabilities

```
RP/0/RSP0/CPU0:PR-ASR9K-3#show cef 3.3.3.3/32 det
Tue Apr 23 08:27:41.826 UTC
3.3.3.3/32, version 611, internal 0x4000001 (ptr 0x7178e220) [4], 0x0 (0x0), 0x0 (0x0)
Updated Apr 23 08:27:23.875
Prefix Len 32, traffic index 0, precedence routine (0), priority 3
  gateway array (0x70f2524c) reference count 1, flags 0x8020, source rib (5), 0 backups
              [1 type 3 flags 0x90111 (0x7105025c) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]

  Level 1 - Load distribution: 0 1 2 3 4
  [0] via 198.18.7.2, recursive
  [1] via 198.18.7.6, recursive
  [2] via 198.18.7.10, recursive
  [3] via 198.18.7.14, recursive
  [4] via 198.18.7.18, recursive
```

Buckets for LB distribution and path index

```
router static
address-family ipv4 unicast
 3.3.3.3/32 198.18.7.2
 3.3.3.3/32 198.18.7.6
 3.3.3.3/32 198.18.7.10
 3.3.3.3/32 198.18.7.14
 3.3.3.3/32 198.18.7.18
```

Static routes missing a next hop interface are perceived recursive!!

Non recursive static routes

```
RP/0/RSP0/CPU0:PR-ASR9K-3#show cef 3.3.3.3 detail loc 0/0/cpu0
3.3.3.3/32, version 4471, internal 0x4000001 (ptr 0x8850f79c) [4], 0x0 (0x0), 0x
………………..

 Level 1 - Load distribution: 0 1 2 3 4
 [0] via 198.18.7.2, recursive
 [1] via 198.18.7.6, recursive
 [2] via 198.18.7.10, recursive
 [3] via 198.18.7.14, recursive
 [4] via 198.18.7.18, recursive
```

```
router static
address-family ipv4 unicast
  3.3.3.3/32 198.18.7.2
  3.3.3.3/32 198.18.7.6
  3.3.3.3/32 198.18.7.10
  3.3.3.3/32 198.18.7.14
  3.3.3.3/32 198.18.7.18
```

```
router static
address-family ipv4 unicast
  3.3.3.3/32 GigabitEthernet0/0/0/5.10 198.18.7.2
  3.3.3.3/32 GigabitEthernet0/0/0/5.20 198.18.7.6
  3.3.3.3/32 GigabitEthernet0/0/0/5.30 198.18.7.10
  3.3.3.3/32 GigabitEthernet0/0/0/5.40 198.18.7.14
  3.3.3.3/32 GigabitEthernet0/0/0/5.50 198.18.7.18
```

```
RP/0/RSP0/CPU0:PR-ASR9K-3#show cef 3.3.3.3/32 det

3.3.3.3/32, version 695, internal 0x4000001 (ptr 0x7178e220) [7], 0x0
……..
  via 198.18.7.2, GigabitEthernet0/0/0/5.10, 4 dependencies, weight 0, class 0
[flags 0x0]
   path-idx 0 [0x7213a560 0x0]
   next hop 198.18.7.2
   remote adjacency
  via 198.18.7.6, GigabitEthernet0/0/0/5.20, 4 dependencies, weight 0, class 0
   path-idx 1 [0x7213a5bc 0x0]
   next hop 198.18.7.6
   remote adjacency
 ……….

   Load distribution: 0 1 2 3 4 (refcount 2)

   Hash  OK  Interface                 Address
   0     Y   GigabitEthernet0/0/0/5.10 remote
   1     Y   GigabitEthernet0/0/0/5.20 remote
   2     Y   GigabitEthernet0/0/0/5.30 remote
   3     Y   GigabitEthernet0/0/0/5.40 remote
   4     Y   GigabitEthernet0/0/0/5.50 remote
```

Show cef for recursive prefix (non fixed)

Weight distribution:

slot 0, weight 9, normalized_weight 5

slot 1, weight 9, normalized_weight 5

Weight is 5 (5 next hops for 1 prefix)

Level 1 - Load distribution: 0 1 0 1 0 1 0 1 0 1

[0] via 3.3.3.3, recursive

[1] via 172.16.1.2, recursive

10 indexes, because weight is 5 and 2 paths

via 3.3.3.3, 4 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]

path-idx 0 [0x7178e220 0x0]

next hop 3.3.3.3 via 3.3.3.3/32

Load distribution: _ _ _ _ _ _ _ _ _ _ _ (refcount 1)

Hash OK Interface Address

- Y **GigabitEthernet0/0/0/5.50 remote**
- Y **GigabitEthernet0/0/0/5.10 remote**
- Y **GigabitEthernet0/0/0/5.20 remote**
- Y **GigabitEthernet0/0/0/5.30 remote**
- Y **GigabitEthernet0/0/0/5.40 remote**

via 172.16.1.2, 15 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]
path-idx 1 [0x7178f078 0x0]
next hop 172.16.1.2 via 172.16.1.2/32

- Y **GigabitEthernet0/0/0/0 remote**
- Y **GigabitEthernet0/0/0/0 remote**
- Y **GigabitEthernet0/0/0/0 remote**
- Y **GigabitEthernet0/0/0/0 remote**
- Y **GigabitEthernet0/0/0/0 remote**

Adj is remote because
Show command not done with location 0/0/CPU0

50/50 split over 2 paths

Show cef for the recursive prefix (fixed)

Weight distribution:

slot 0, **weight 9, normalized_weight 9**

slot 1, **weight 1, normalized_weight 1**

This weight is set as part of the dmz link BW (not auto computed!!)

Level 1 - Load distribution: 0 1 0 0 0 0 0 0 0 0

[0] via 3.3.3.3, recursive

[1] via 172.16.1.2, recursive

via 3.3.3.3, 7 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]

path-idx 0 [0x7178e220 0x0]

next hop 3.3.3.3 via 3.3.3.3/32

Load distribution: 0 1 2 3 4 (refcount 1)

Hash	OK	Interface	Address
0	Y	GigabitEthernet0/0/0/5.10	remote
1	Y	GigabitEthernet0/0/0/5.20	remote
2	Y	GigabitEthernet0/0/0/5.30	remote
3	Y	GigabitEthernet0/0/0/5.40	remote
4	Y	GigabitEthernet0/0/0/5.50	remote

via 172.16.1.2, 7 dependencies, recursive, bgp-ext, bgp-multipath [flags 0x60a0]

path-idx 1 [0x7178f078 0x0]

next hop 172.16.1.2 via 172.16.1.2/32

Load distribution: 0 (refcount 1)

Hash	OK	Interface	Address
5	Y	GigabitEthernet0/0/0/0	remote

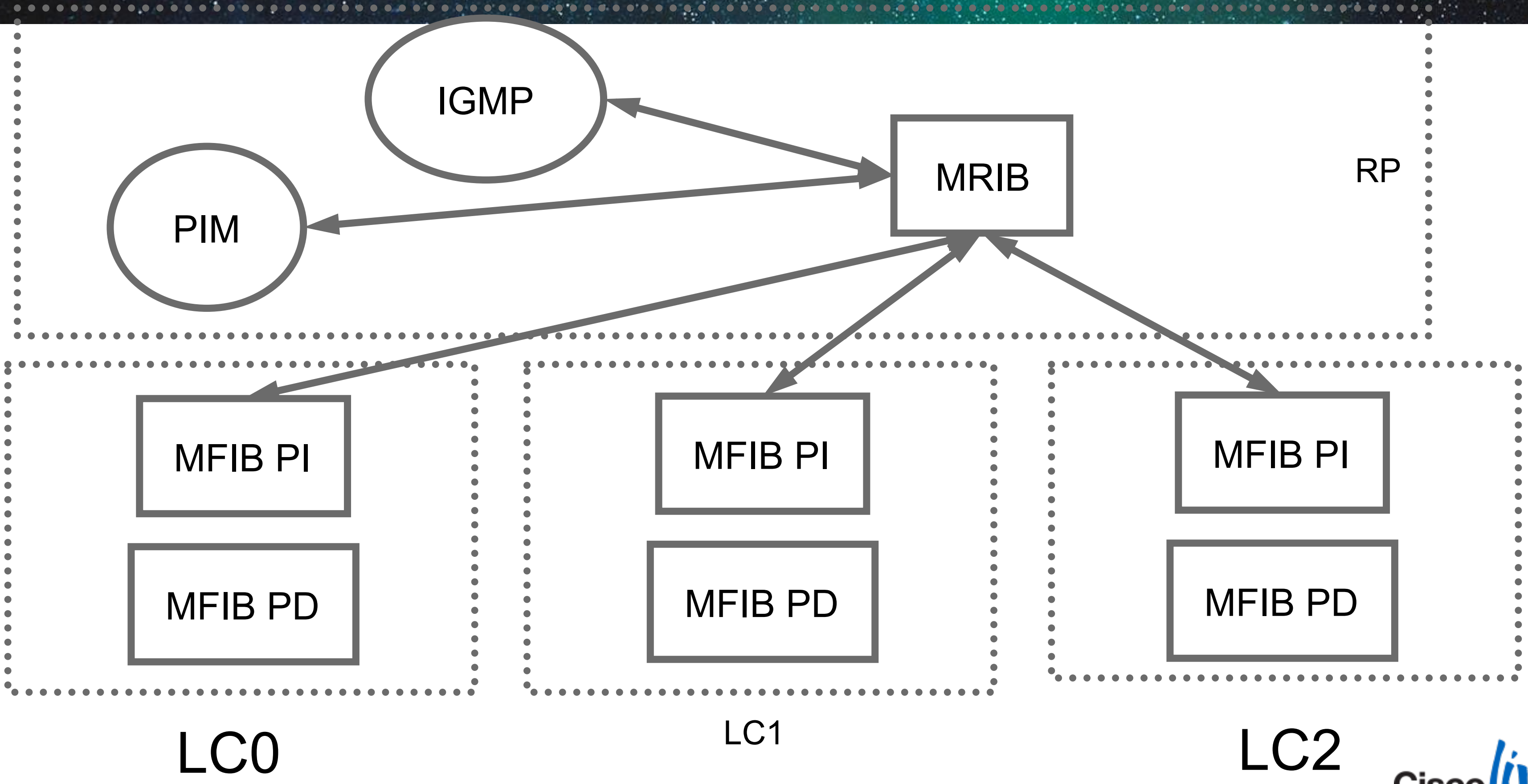
Great references

- Understanding NP counters
  - *https://supportforums.cisco.com/docs/DOC-15552*
- Capturing packets in the ASR9000 forwarding path
  - https://supportforums.cisco.com/docs/DOC-29010
- Loadbalancing Architecture for the ASR9000
  - https://supportforums.cisco.com/docs/DOC-26687
- Understanding UCMP and ECMP
  - https://supportforums.cisco.com/docs/DOC-32365

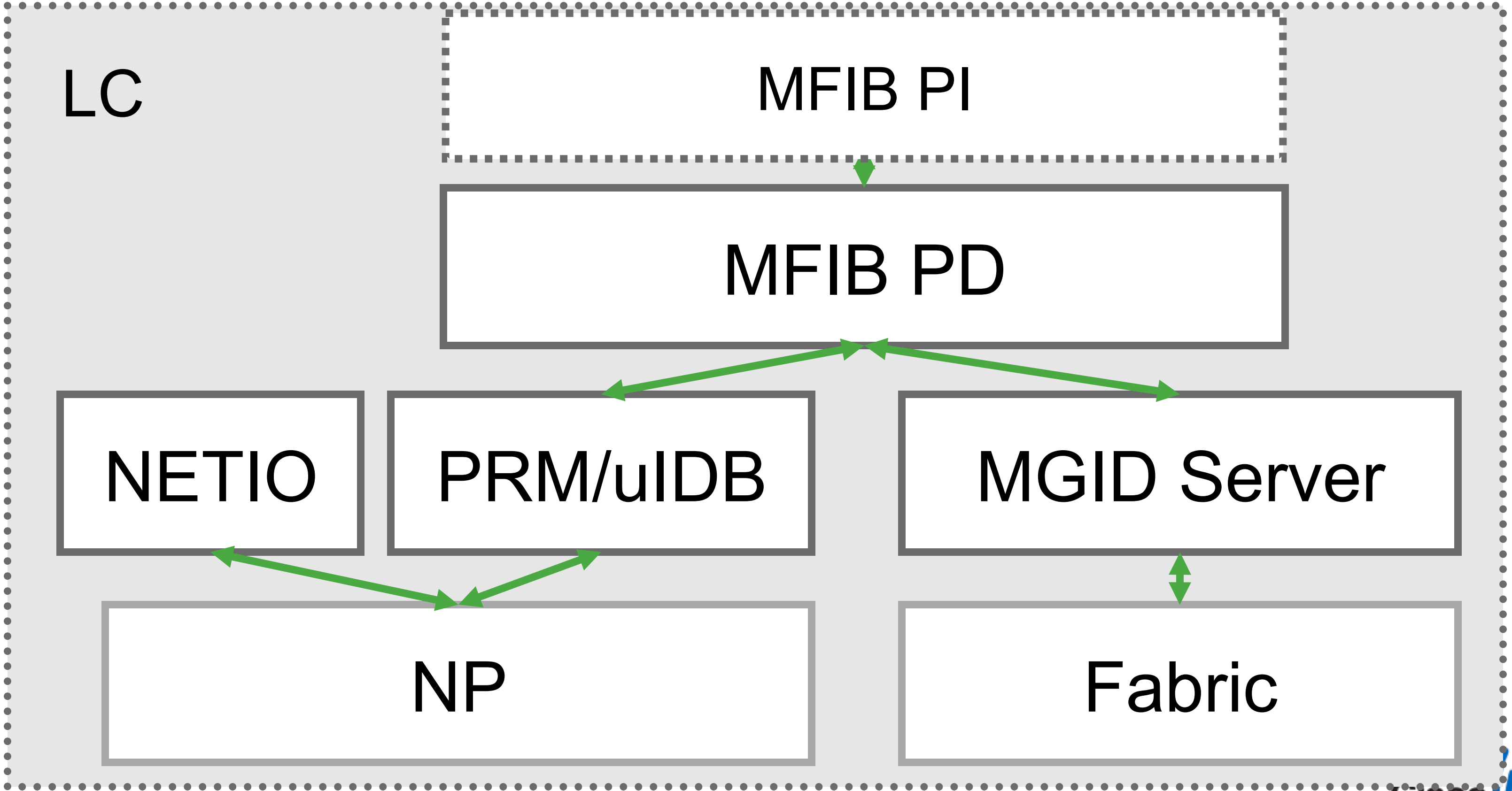
Multicast troubleshooting

- MRIB and MFIB
- MFIB and LC components

Software Architecture – MRIB/MFIB



Software Architecture – MFIB on LC

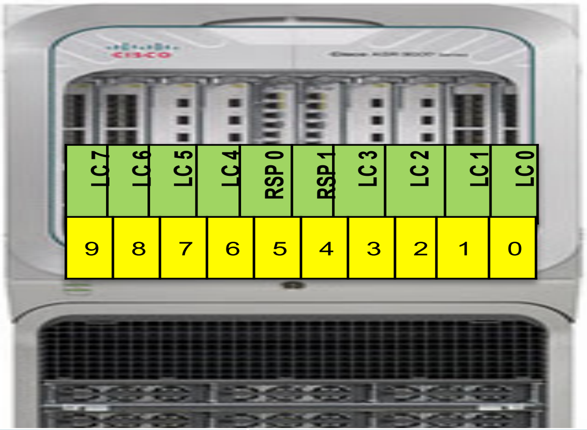


MGIDs and FGIDs

- MGID - Multicast Group Identifier
  - Unique ID assigned to a multicast group
  - Used by FIA/Bridge to determine replication requirements per multicast group
- FGID - Fabric Group Identifier
  - Slotmask used by switch fabric to determine replication to line card/RSP slots
  - Assigned to each group by multicast PD control plane

FGID (Slotmask)

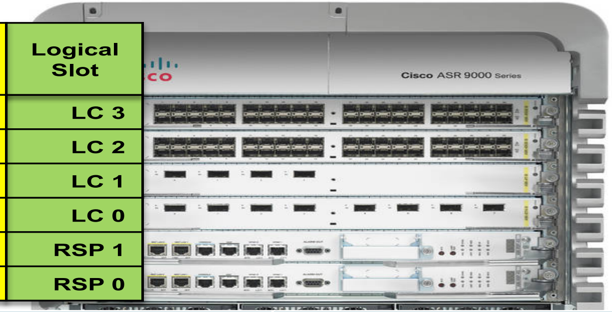
FGIDs: 10 Slot Chassis



Slot		Slot Mask	
Logical	Physical	Binary	Hex
LC7	9	1000000000	0x0200
LC6	8	0100000000	0x0100
LC5	7	0010000000	0x0080
LC4	6	0001000000	0x0040
RSP0	5	0000100000	0x0020
RSP1	4	0000010000	0x0010
LC3	3	0000001000	0x0008
LC2	2	0000000100	0x0004
LC1	1	0000000010	0x0002
LC0	0	0000000001	0x0001

FGIDs: 6 Slot Chassis

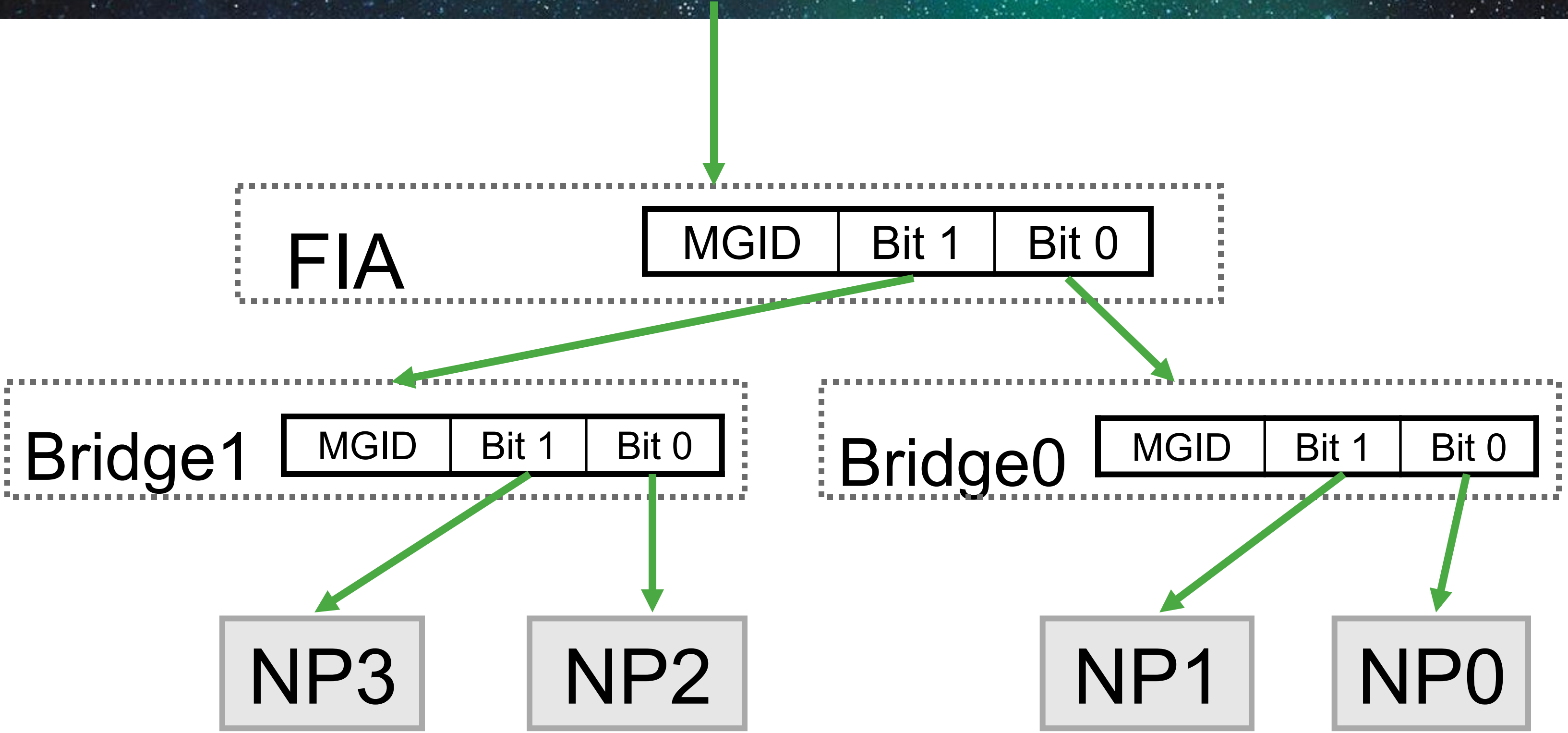
Phy Slot Number	Logical Slot
5	LC 3
4	LC 2
3	LC 1
2	LC 0
1	RSP 1
0	RSP 0



Slot		Slot Mask	
Logical	Physical	Binary	Hex
LC3	5	0000100000	0x0020
LC2	4	0000010000	0x0010
LC1	3	0000001000	0x0008
LC0	2	0000000100	0x0004
RSP1	1	0000000010	0x0002
RSP0	0	0000000001	0x0001

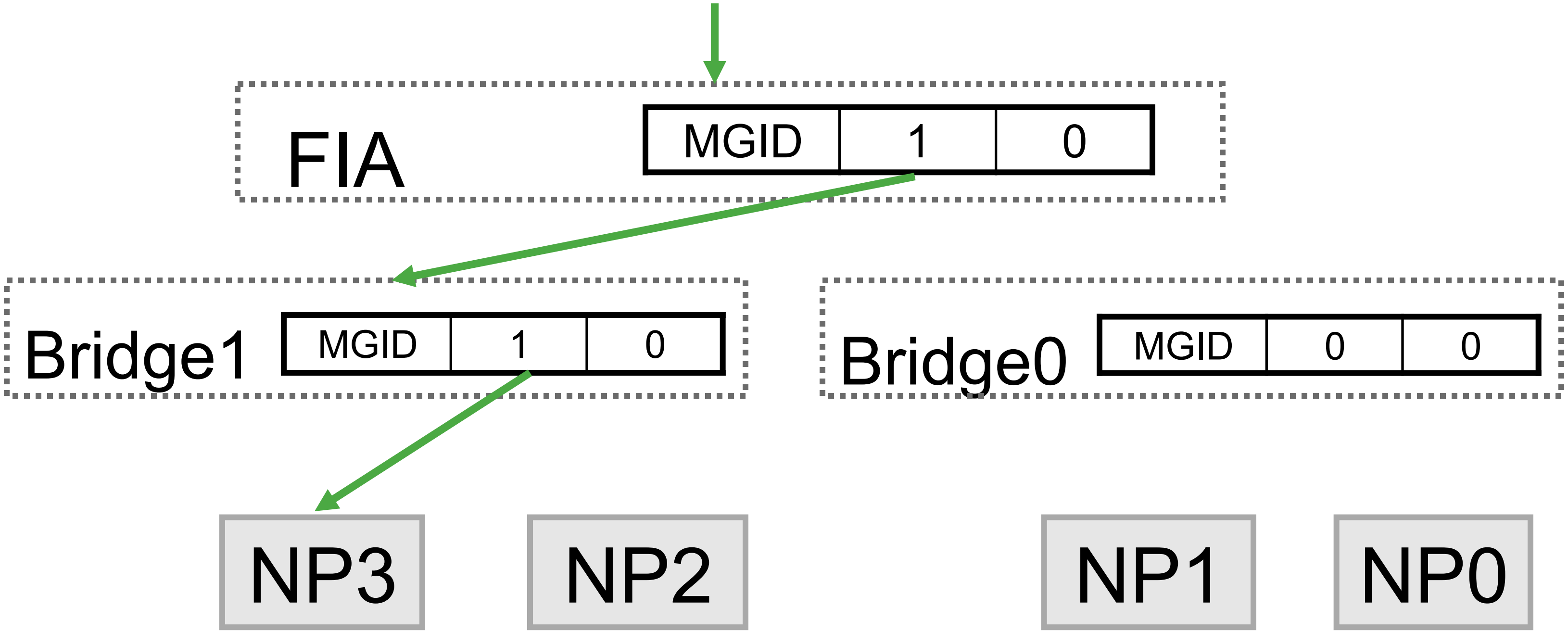
Target Linecards	FGID Value (10 Slot Chassis)
LC6	0x0100
LC1 + LC5	0x0002 | 0x0080 = 0x0082
LC0 + LC3 + LC7	0x0001 | 0x0008 | 0x0200 = 0x0209

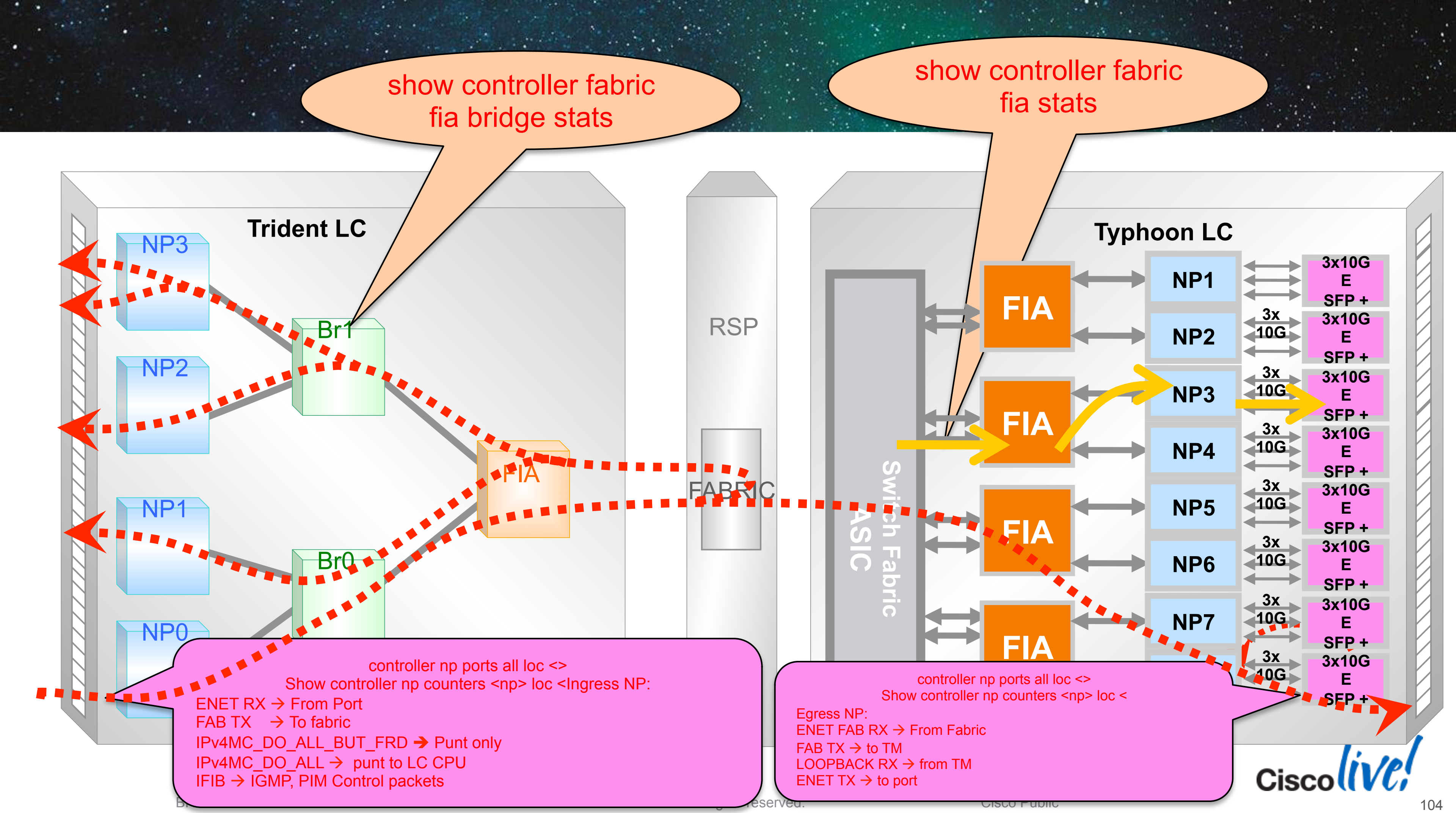
MGID Tables   MGID Bitmasks



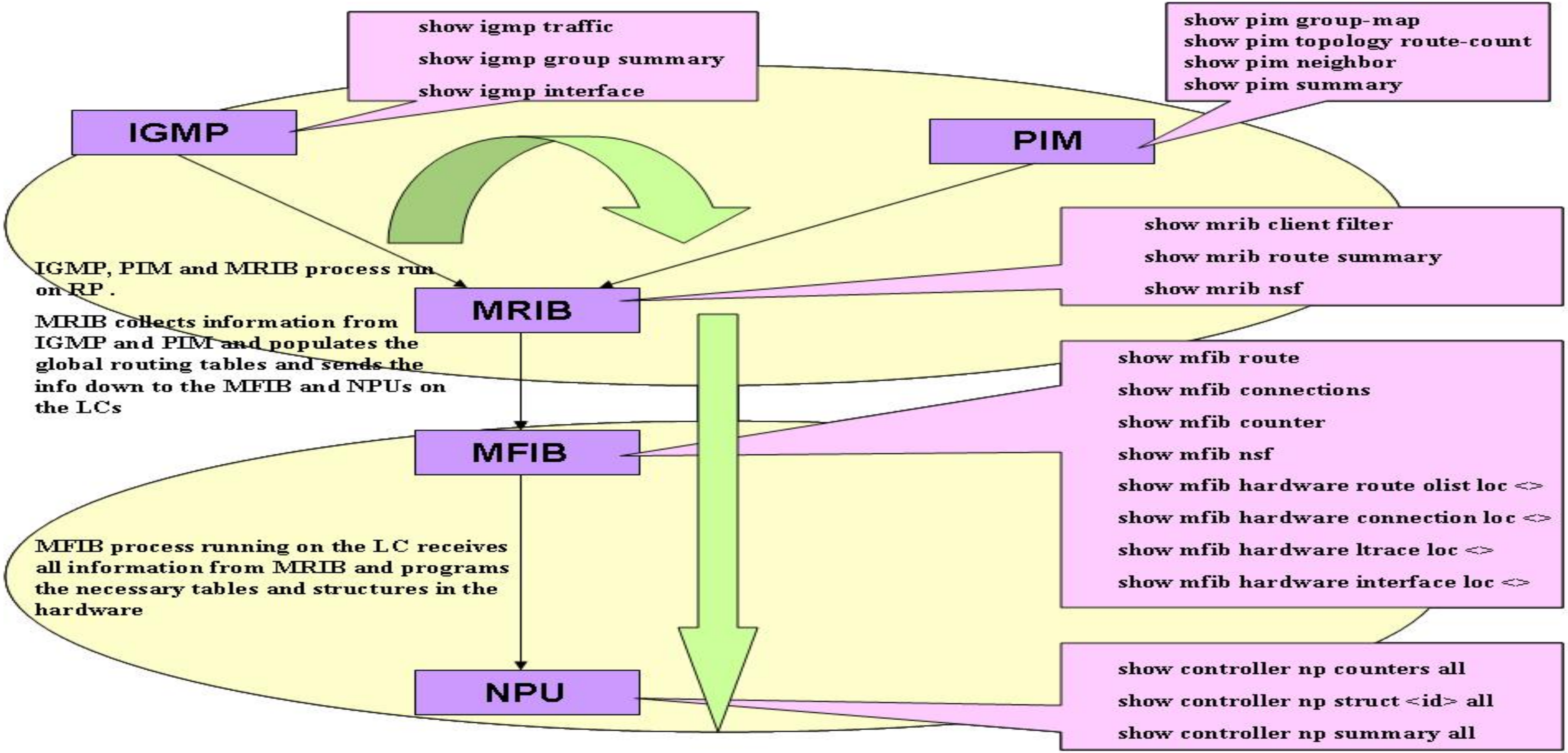
MGID Tables

Mcast traffic replication based on mgid




show controller fabric fia bridge stats
show controller fabric fia stats
Trident LC
NP3
NP2
NP1
NP0
Br1
Br0
FIA
RSP
FABRIC
Typhoon LC
Switch Fabric ASIC
FIA
FIA
FIA
FIA
NP1
NP2
NP3
NP4
NP5
NP6
NP7
3x 10G
3x10G E SFP +
controller np ports all loc <>
Show controller np counters <np> loc <Ingress NP:
ENET RX → From Port
FAB TX → To fabric
IPv4MC_DO_ALL_BUT_FRD → Punt only
IPv4MC_DO_ALL → punt to LC CPU
IFIB → IGMP, PIM Control packets
controller np ports all loc <>
Show controller np counters <np> loc <
Egress NP:
ENET FAB RX → From Fabric
FAB TX → to TM
LOOPBACK RX → from TM
ENET TX → to port
Cisco live!

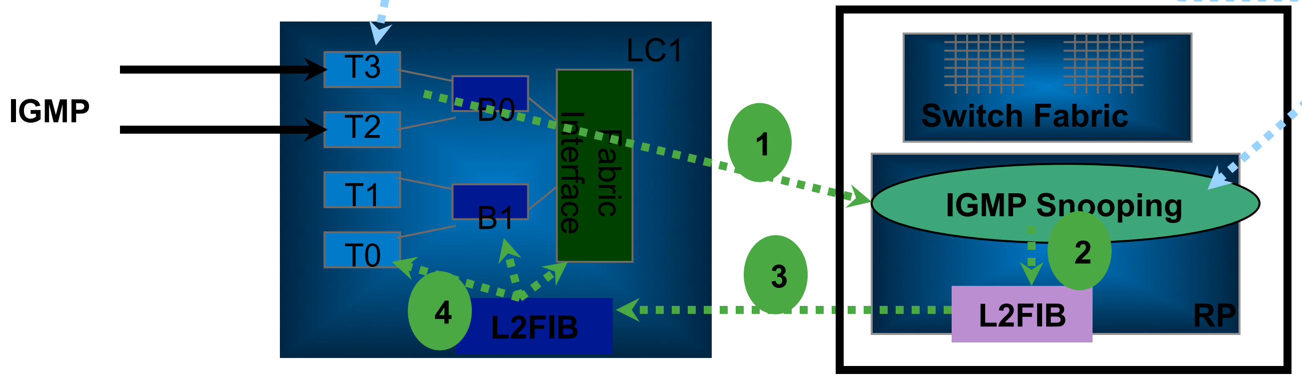
L3 Multicast Show CLIs

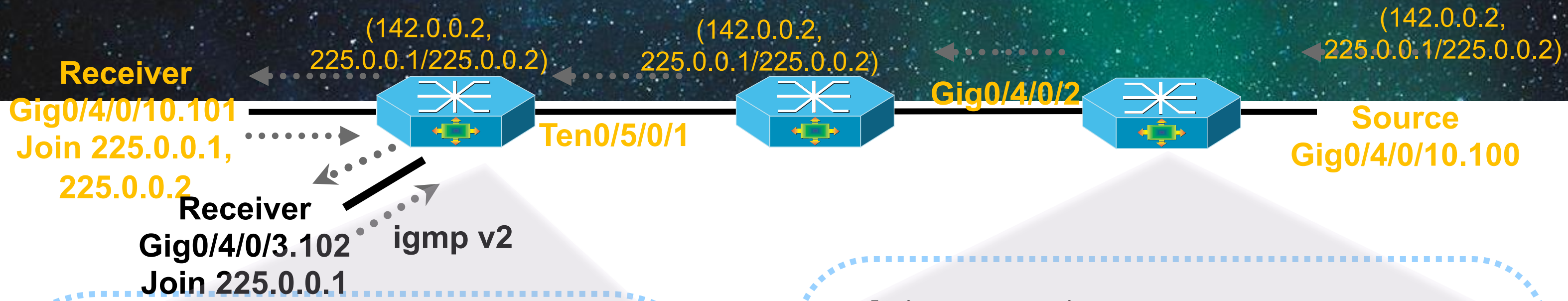


L2 Multicast Show CLIs

show l2vpn forward mroute ipv4 hardware

sh igmp snoop sum
sh igmp snoop sum stat
sh igmp snoop group
sh igmp snoop bridge






```
interface GigabitEthernet0/4/0/10.101
 ipv4 address 33.0.2.1 255.255.255.0
 encapsulation dot1q 101
interface GigabitEthernet0/4/0/3.102
 ipv4 address 42.0.1.2 255.255.255.0
 encapsulation dot1q 102
interface TenGigE0/5/0/1
 ipv4 address 40.0.75.2 255.255.255.0
!
multicast-routing
 address-family ipv4
interface all enable
router pim
 address-family ipv4
  rp-address 110.0.0.24
interface TenGigE0/5/0/1
   enable
  interface GigabitEthernet0/4/0/3.102
   enable
interface GigabitEthernet0/4/0/10.101
   enable
RP/0/RSP0/CPU0:ASR9K-3#
```

```
multicast-routing
 address-family ipv4
interface all enable
router pim
 address-family ipv4
  rp-address 110.0.0.24
interface GigabitEthernet0/4/0/2
   enable
interface GigabitEthernet0/4/0/10.100
   enable
RP/0/RSP0/CPU0:ASR9K-2#
```

Example 1 – L3 Multicast PIM SSM
Show CLI – Validate the mrib and mfib entry

```
RP/0/RSP1/CPU0:asr9k-2#show mrib route 225.0.0.1
== snip ==
(142.0.0.2,225.0.0.1) RPF nbr: 142.0.0.2 Flags: L
  Up: 4d05h
  Incoming Interface List
    GigabitEthernet0/4/0/10.100 Flags: A, Up: 4d03h
  Outgoing Interface List
    GigabitEthernet0/4/0/2 Flags: F NS, Up: 2d22h
RP/0/RSP0/CPU0:asr9k-3#show mrib route 225.0.0.2 detail
 === snip ===
(142.0.0.2,225.0.0.2) Ver: 0x2fba RPF nbr: 40.0.75.1 Flags:,
 PD: Slotmask: 0x40      ← Same slot mask as 225.0.0.1. Because egress LC is same.
     MGID: 19921         ← Different MGID. Packets replicated to only one NP.
  Up: 2d23h
  Incoming Interface List
    TenGigE0/5/0/1 Flags: A, Up: 2d23h
  Outgoing Interface List
    GigabitEthernet0/4/0/10.101 Flags: F NS, Up: 2d23h
RP/0/RSP0/CPU0:asr9k-3#
```

MGID tables

Getting MGID and Displaying MGID table

```
RP/0/RSP0/CPU0:asr9k-3#show mrib route 225.0.0.1 detail
 (*,225.0.0.1) Ver: 0x429a RPF nbr: 40.0.75.1 Flags: C,
 PD: Slotmask: 0x40
     MGID: 19919
  Up: 2d21h
  Incoming Interface List
    TenGigE0/5/0/1 Flags: A NS, Up: 2d21h
  Outgoing Interface List
    GigabitEthernet0/4/0/3.102 Flags: F NS LI, Up: 14:20:00
    GigabitEthernet0/4/0/10.101 Flags: F NS LI, Up: 2d21h
(142.0.0.2,225.0.0.1) Ver: 0x7163 RPF nbr: 40.0.75.1 Flags:,
 PD: Slotmask: 0x40    ← FGID Used for Fabric Replication 0x40 == 0001000000 (slot 4)
     MGID: 19918       ← MGID Used by egress LC's FIA and Bridge ASIC for replication
  Up: 3d00h
  Incoming Interface List
    TenGigE0/5/0/1 Flags: A, Up: 3d00h ← Interface towards source (RPF to source)
  Outgoing Interface List
    GigabitEthernet0/4/0/3.102 Flags: F NS, Up: 14:20:00 ← interface towards receivers
    GigabitEthernet0/4/0/10.101 Flags: F NS, Up: 2d21h   ← interface towards receivers
RP/0/RSP0/CPU0:asr9k-3#
```

```
RP/0/RSP0/CPU0:asr9k-3#show controllers mgidprgm mgidindex 19918 location 0/4/CPU0

Device          MGID-Bits        Client-Last-Modified
=============================================================

FIA             10          MFIBV4 ← Replicated to Bridge-1   [Bridge-1 | Bridge-0]
Bridge-0        0           MFIBV4 ← Not replicated here      [NP 1 | NP 0]
Bridge-1        11          MFIBV4 ← Replicated to NP 2 and 3 [NP 3|NP 2]
RP/0/RSP0/CPU0:asr9k-3#
```


Cisco live!

MGID/FGID and NP

```
RP/0/RSP0/CPU0:asr9k-3#show mfib hardware route olist 225.0.0.1 location 0/4/CPU0
------ SNIP----
Source: 142.0.0.2       Group: 225.0.0.1       Mask: 64  RPF Int: Te0/5/0/1
  Route Information
  ------------------------------------------------------------------------
  B  S  DC PL PR PF DR RI          FS      G       M
  ------------------------------------------------------------------------
  F  F  F  F  F  F  F  0xe000100  0x40   19918  3797  ←FGID and MGID values
  ------------------------------------------------------------------------
  Interface Information
  ------------------------------------------------------------------------
  NP Intf             OT  U      T  IC B
  ------------------------------------------------------------------------
  2  Gi0/4/0/10.101   REG 85     1  F  F      ←  NP and Outgoing port info
  3  Gi0/4/0/3.102    REG 109    1  F  F      ←  NP and Outgoing port info
  ------------------------------------------------------------------------
  OLIST counts
  --------------------------------------------------------
  NP:         0     1     2     3
  Count:      0     0     1     1  ← Shows 1 port from NP 2 and 3 interested in traffic.
  --------------------------------------------------------
RP/0/RSP0/CPU0:asr9k-3#
```

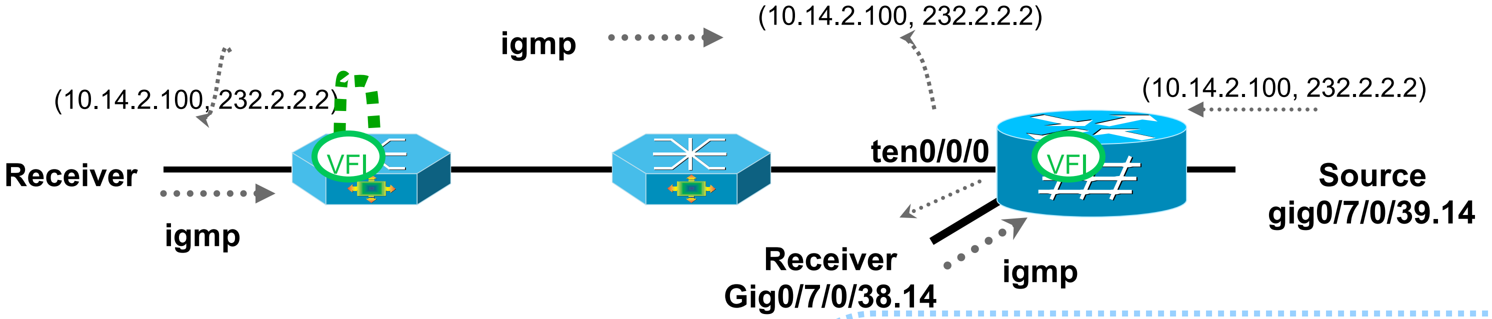
Legend to previous output

```
-----------------------------------------------------------------------------
Legend:
Route Information
    NP:   NP ID                      B:    BACL check
    S:    RPF Interface signal       DC:   Directly connected
    PL:   Punt to LC CPU             PR:   Punt to RP
    PF:   Punt if forwarded          DR:   Drop all
    RI:   RPF interface              FS:   Fabric slotmask
    G:    Multicast group ID         M:    Multicast Leaf Index
    T:    Table ID for lookup        OC:   Count of OLIST members
    Base: Base of the statistics pointer NI:   Not Installed

Interface Information
    NP:   NP ID                      Intf: Interface
    U:    uIDB index                 OT:   OLE Type
    T:    Table ID                   IC:   HW IC flag
    B:    HW BACL bit                EU:   Interface uIDB index
    IB:   Bundle interface           EH:   In HW OLIST table
    OIDX: OLIST index on NP          PT:   Punt table entry
    Base: Statistics Ptr base        RM:   Remote FGID (Pri/Back)

Software OLIST Information
    SW OC: Software OLIST counts     HW OC: Hardware OLIST counts
    T:     Table ID                  SD:    Send direct flag
-----------------------------------------------------------------------------
```

Example 1 – L3 Multicast PIM SM
show CLI – check the counters [1]

```
RP/0/RSP0/CPU0:asr9k-3#show mfib hardware route statistics 225.0.0.1 142.0.0.2 loc 0/5/CPU0
LC Type: Typhoon A9K-MOD160-SE
Source: 142.0.0.2  Group: 225.0.0.1  Mask:64
  -------------------------------------------------------------------------
  NP            R(packets:bytes)/F(packets:bytes)/P(packets)/ID(packets)/ED(packets)
  -------------------------------------------------------------------------
  0             406759:18710914 / 0:0 / 0 / 0 / 0   ← THIS NP is receiving traffic from wire
  1             0:0 / 0:0 / 0 / 0 / 0
  2             0:0 / 0:0 / 0 / 0 / 0
  3             0:0 / 0:0 / 0 / 0 / 0
  -------------------------------------------------------------------------
RP/0/RSP0/CPU0:asr9k-3#show mfib hardware route statistics 225.0.0.1 142.0.0.2 loc 0/4/CPU0
LC Type: Trident A9K-40GE-E
--------------------------------------------------------------------------
Source: 142.0.0.2  Group: 225.0.0.1  Mask:64
  -------------------------------------------------------------------------
  NP            R(packets:bytes)/F(packets:bytes)/P(packets)/ID(packets)/ED(packets)
  -------------------------------------------------------------------------
  0             0:0 / 0:0 / 0 / 0 / 0
  1             0:0 / 0:0 / 0 / 0 / 0
  2             0:0 / 434208:19973568 / 0 / 0 / 0  ← This NP is sending traffic out on wire
  3             0:0 / 443309:20392214 / 0 / 0 / 0  ← This NP is sending traffic out on wire
  -------------------------------------------------------------------------
  Interface Statistics:
  -------------------------------------------------------------------------
  C   Interface       F/P/D (packets:bytes)
  -------------------------------------------------------------------------
  2   Gi0/4/0/10.101  434208:19973568 / 0:0 / 0:0  ← Outgoing interface on the NP2
  3   Gi0/4/0/3.102   443309:20392214 / 0:0 / 0:0  ← Outgoing interface on the NP3
  -------------------------------------------------------------------------
RP/0/RSP0/CPU0:asr9k-3#
```

Example 2 – L2 Multicast IGMP Snooping




```
interface GigabitEthernet0/7/0/39.12
l2transport
 encapsulation dot1q 12
 rewrite ingress tag pop 1 symmetric


interface GigabitEthernet0/7/0/38.12
 encapsulation dot1q 12
 rewrite ingress tag pop 1 symmetric
```

```
igmp snoop profile igmp-prf1
igmp snoop profile igmp-prf2
        mrouter

l2vpn
 bridge group viking-demo
  bridge-domain 12
   igmp snooping profile igmp-prf1
   interface GigabitEthernet0/7/0/38.12
   interface GigabitEthernet0/7/0/39.12
     igmp snooping profile igmp-prf2

   vfi vpls-12
    neighbor 10.0.0.1 pw-id 12
```

Example 2 – L2 Multicast
Show CLIs: sh igmp snooping summ stats

```
#sh igmp snooping summary statistics

Traffic Statistics (elapsed time since last cleared 00:30:52):
                        .....
                              Received  Reinjected  Generated
  Messages:                          5           0          3
  IGMP General Queries:              3           0          0
  IGMP Group Specific Queries:       0           0          0
  IGMP G&S Specific Queries:         0           0          0
  IGMP V2 Reports:                   2           0          0
    IGMP V3 Reports:                 0           0          3
    IGMP V2 Leaves:                  0           0          0
    IGMP Global Leaves:              0           -          0
    PIM Hellos:                      0           0          -
   Rx Packet Treatment:
    Packets Flooded:                             0
    Packets Forwarded To Members:                0
    Packets Forwarded To Mrouters:               0
    Packets Consumed:                            5
   Rx Errors:
    None
   Tx Errors:
    None
```

Example 2 – L2 Multicast
Show CLIs: sh igmp snooping …

```
#sh igmp snooping bridge-domain

Bridge:Domain          Profile  Act  Ver  #Ports  #Mrtrs  #Grps   #SGs
--------------------   -------- ---  ---  ------  ------  -----   ----
Viking-demo:12         prof1    Y    v3        2       1      2      0


#sh igmp snooping group

Key: GM=Group Filter Mode, PM=Port Filter Mode
Flags Key: S=Static, D=Dynamic, E=Explicit Tracking

                    Bridge Domain Viking-demo:12

Group          Ver GM Source        PM Port                          Exp  Flg
-----          --- -- ------        -- ----                          ---  ---
239.1.1.1       V3  EX *            EX GigabitEthernet0/0/0/6        104   D
239.1.2.1       V3  EX *            EX GigabitEthernet0/0/0/6        104   D
```

Example 2 – L2 Multicast
Show CLIs: sh l2vpn forwarding …

```
#sh l2vpn forwarding mroute ipv4 loc 0/0/cpu0
Bridge-Domain Name: Viking-demo:12
  Prefix: (0.0.0.0,224.0.0.0/4)           <- Default route
  Bridge Port:
    GigabitEthernet0/0/0/4

Bridge-Domain Name: Viking-demo:12
  Prefix: (0.0.0.0,239.1.1.1/32)
  Bridge Port:
    GigabitEthernet0/0/0/6
    GigabitEthernet0/0/0/4

Bridge-Domain Name: Viking-demo:12
  Prefix: (0.0.0.0,239.1.2.1/32)
  Bridge Port:
    GigabitEthernet0/0/0/6
    GigabitEthernet0/0/0/4
```

Example 2 – L2 Multicast
Show CLIs: sh l2vpn forwarding …

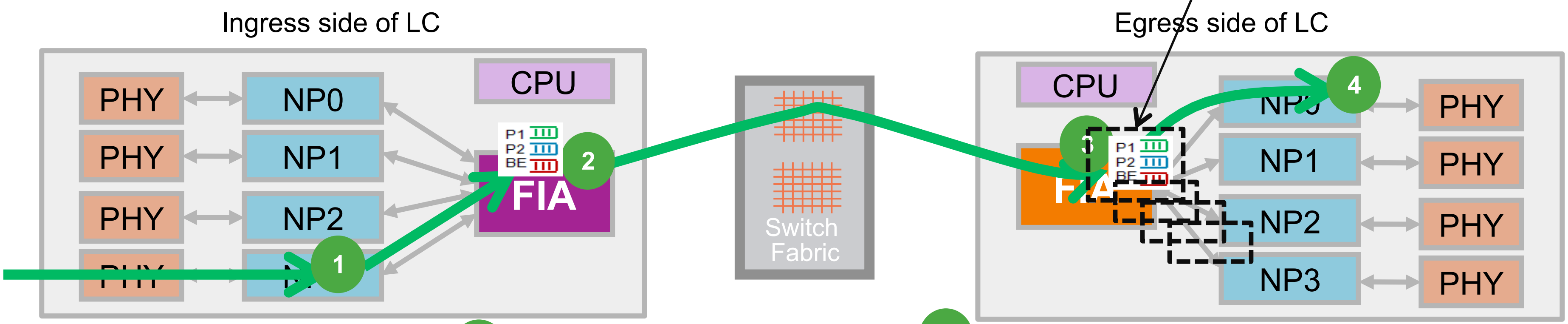
```
#sh l2vpn forwarding mroute ipv4 group 239.1.1.1 hardware ingress loc 0/0/cpu0
Bridge-Domain Name: Viking-demo:12

  ……
 S: Source, G: Group, Pr: Prefix Length, C: Chip ID, R: Received,
 FF: Forwarded to fabric, P: Punted to CPU, D: Dropped, F: Forwarded
 ……
S: *  G: 239.1.1.1  Pr:32
  -----------------------------------------------------------------------
  C    R(packets:bytes)/FF(packets:bytes)/P(packets)/D(packets)
  -----------------------------------------------------------------------
  0    0:0 / 0:0 / 0 / 0
  1    0:0 / 0:0 / 0 / 0
  2    0:0 / 0:0 / 0 / 0
  3    944768:58575616 / 944768:76526208 / 0 / 0        <- Ingress/Fabric
  -----------------------------------------------------------------------
  XID Statistics:
  -----------------------------------------------------------------------
  XID-ID          Stats Ptr  F/P/D (packets:bytes)
  -----------------------------------------------------------------------
  0x1             0x54c98     944768:58575616 / 0:0 / 0:0    <- Egress
  0x2             0x54c9c     0:0 / 0:0 / 0:0
```

QOS architecture

System QoS Refresh

End-to-End priority (P1,P2, Best-effort) propagation →
Guarantee bandwidth, low latency for high priority traffic at any congestion point
3 strict priority level across all internal HW components

One Queue set (4 queues) per each NP on the LC



1 Ingress (sub-)interface QoS Queues

2 Virtual Output Queues

3 Egress FIA Queues

4 Egress (sub-)interface QoS Queues

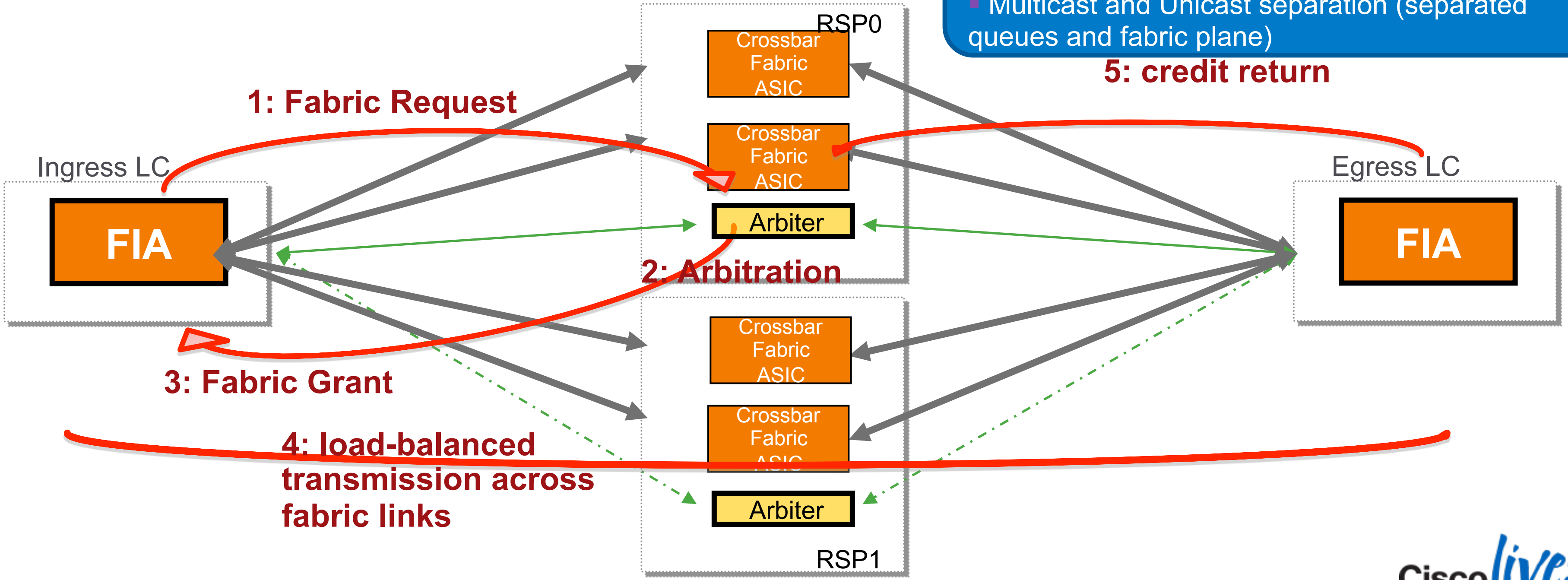
Configure with
Ingress MQC 4-layer hierarchy
Two strict high priority + Normal priority

Implicit Configuration
Two strict high priority +
Normal priority

Configure with
Egress MQC
4-layer hierarchy
Two strict high priority +
Normal priority

System QoS Refresh – Fabric Bandwidth Access Overview

- 3 strict priority scheduling/queueing
- Back pressure and virtual output queue
- Multicast and Unicast separation (separated queues and fabric plane)



Arbitration & Fabric QoS

- Arbitration is being performed by a central high speed arbitration ASIC on the RSP
- At any time a single arbiter is responsible for arbitration (active/active "APS like" protection)
- The Arbitration algorithm is QOS aware and will ensure that P1 classes have preference over P2 classes, both of which have preference over non-priority classes
- Arbitration is performed relative to a given the egress VQI

System QoS Refresh (3) – Backpressure and VoQ Mechanism

Egress NP congestion → → backpressure to ingress FIA →

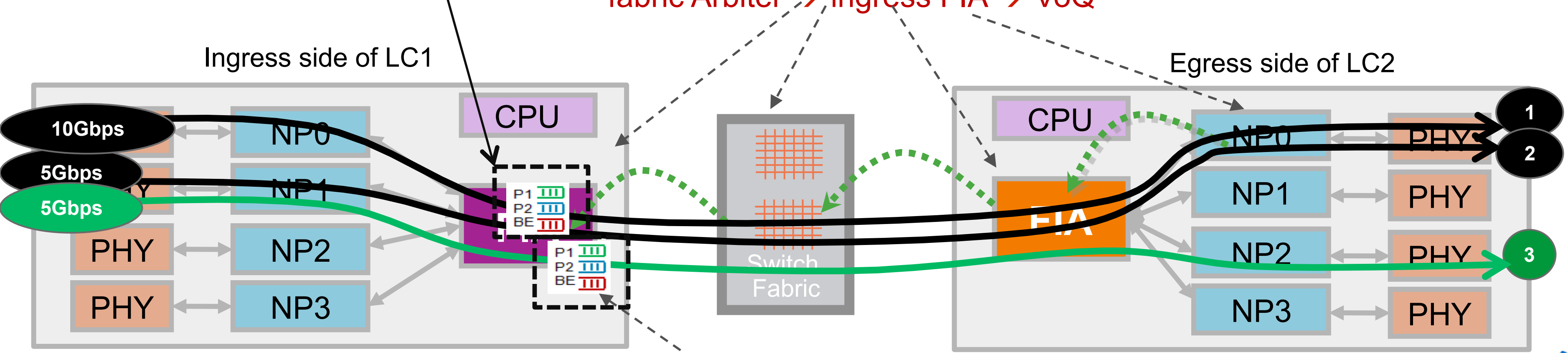
Packet is en-queued in the dedicated VoQ →

No impact of the packet going to different egress NP →

No head-of-line-block issue

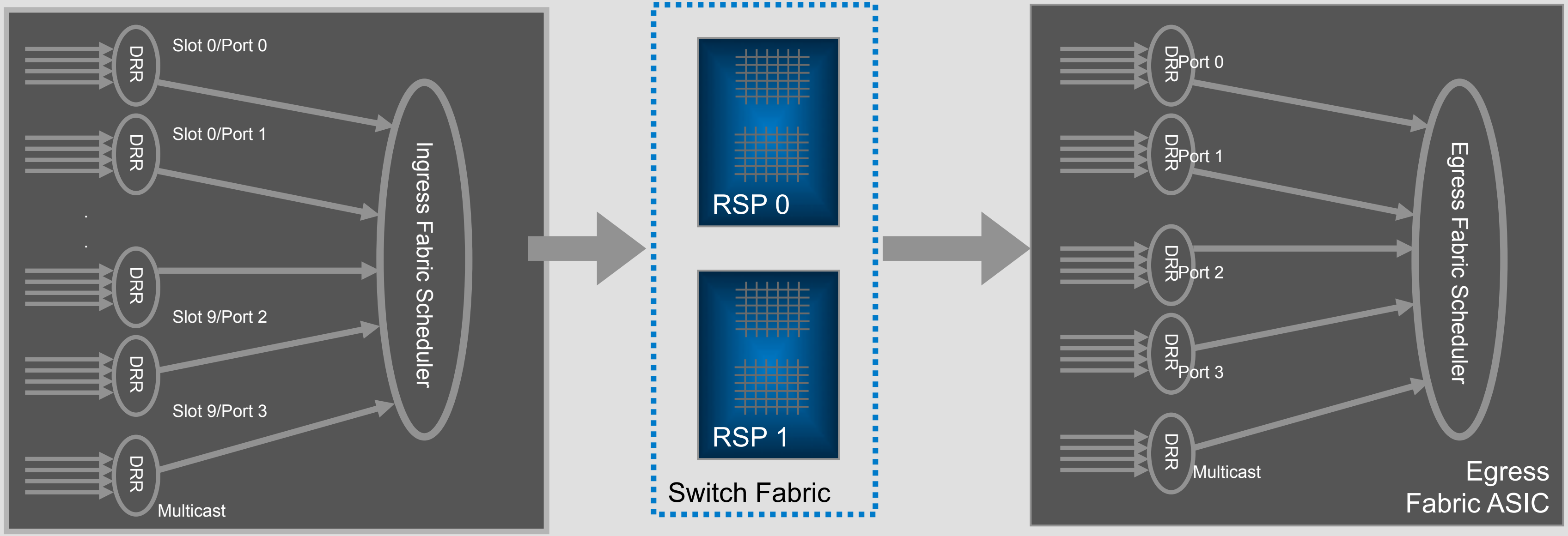
One VoQ set (4 queues) per each NP in the system

Backpressure: egress NP → egress FIA → fabric Arbiter → ingress FIA → VoQ



Packet going to different egress NP put into different VoQ set → Congestion on one NP won't block the packet going to different NP

Linecard QoS
Switch Fabric Queuing mechanisms



136 ingress VoQ used:

- 8 dest LCs * 4 10G ports/LC * 4 classes/port** == 128 VoQ for LCs
- 2 dest RSPs * 1 10G port/RSP * 4 classes/port == 8 VoQ for RSPs
- 4 multicast queues

20 egress fabric queues:

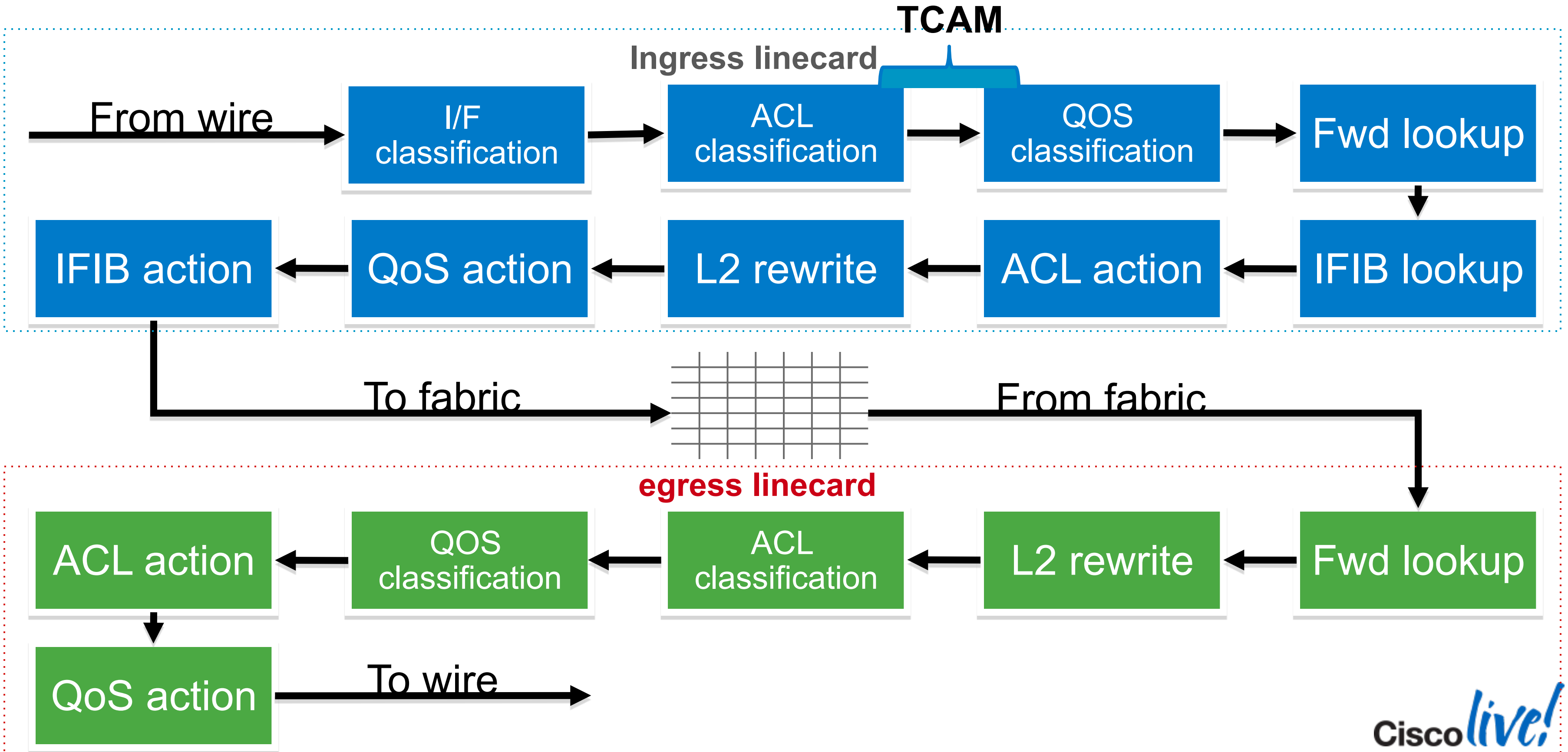
- 4 classes/port * 4 ports/LC (unicast) == 16
- 4 multicast classes == 4

higher density cards will have correspondingly larger numbers of VoQ's

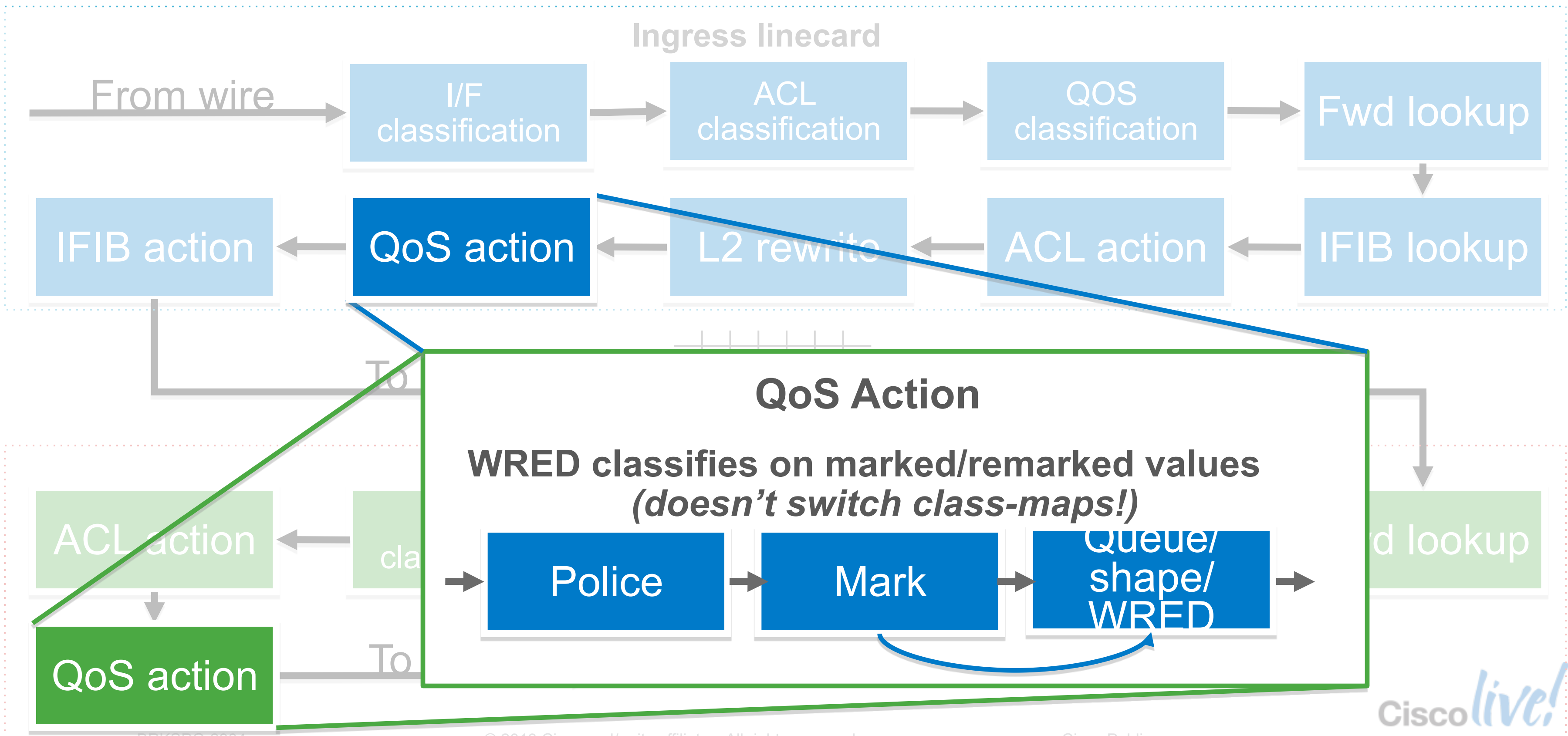
MQC to System QOS mapping

- ASR 9000 supports traffic differentiation at all relevant points within the system
  - P1/P2/LP differentiation or P1/LP differentiation support throughout the system
- Classification into these priorities is based on input MQC classification on the **ingress** linecard into P1, P2, Other
  - Once a packet is classified into a **P1 class** on ingress it will get mapped to PQ1 queue along the system qos path
  - Once a packet is classified into a **P2 class** on ingress it will get mapped to PQ2 queue along the system qos path, unless no MP is implemented. In this case HP would be used for P2.
  - Once a packet is classified into a **non-PQ1/2** class on ingress it will get mapped to LP queue along the system qos path
- Note: The marking is implicit once you assign a packet into a given queue on ingress; its sets the fabric header priority bits onto the packet.
  - no specific "set" action is required

Feature order on ASR 9000 NP (simplified)



Feature order on ASR 9000 NP
QoS Action Order



Injected packets

- In general are injected "to-wire" (same as Pak Priority in IOS)
- Means that all features are bypassed.
- Including QOS
- Few exceptions
  - ICMP
  - BFD echo responses
  - Netflow

CoPP / LPTS

- "Control Plane Policing" and "Local Packet Transport Service"
- Policing of control plane protocols and punted packets is supported
- CoPP is performed by NP, i.e in hardware
- Policer Values configurable
  - ✓ but with very sensible defaults that rarely need to be changed!
- 8 Priorities in towards CPU, CPU will honor priorities when accepting packets for processing

ASR 9000 QOS Implicit Trust

- For Bridged packets on ingress – outermost COS would be treated as trusted.
- For Routed packets on ingress – DSCP/Precedence/outermost EXP would be treated as trusted based on packet type.
- Default QOS will be gleaned from ingress interface before QOS marking is applied on the ingress policymap.
- By default ASR 9000 would never modify DSCP/IP precedence of a packet without a policy-map configured.
- Default QOS information would be used for impositioned fields only

ASR 9000 Linecard/NP QoS Overview

Typhoon System QoS Overview

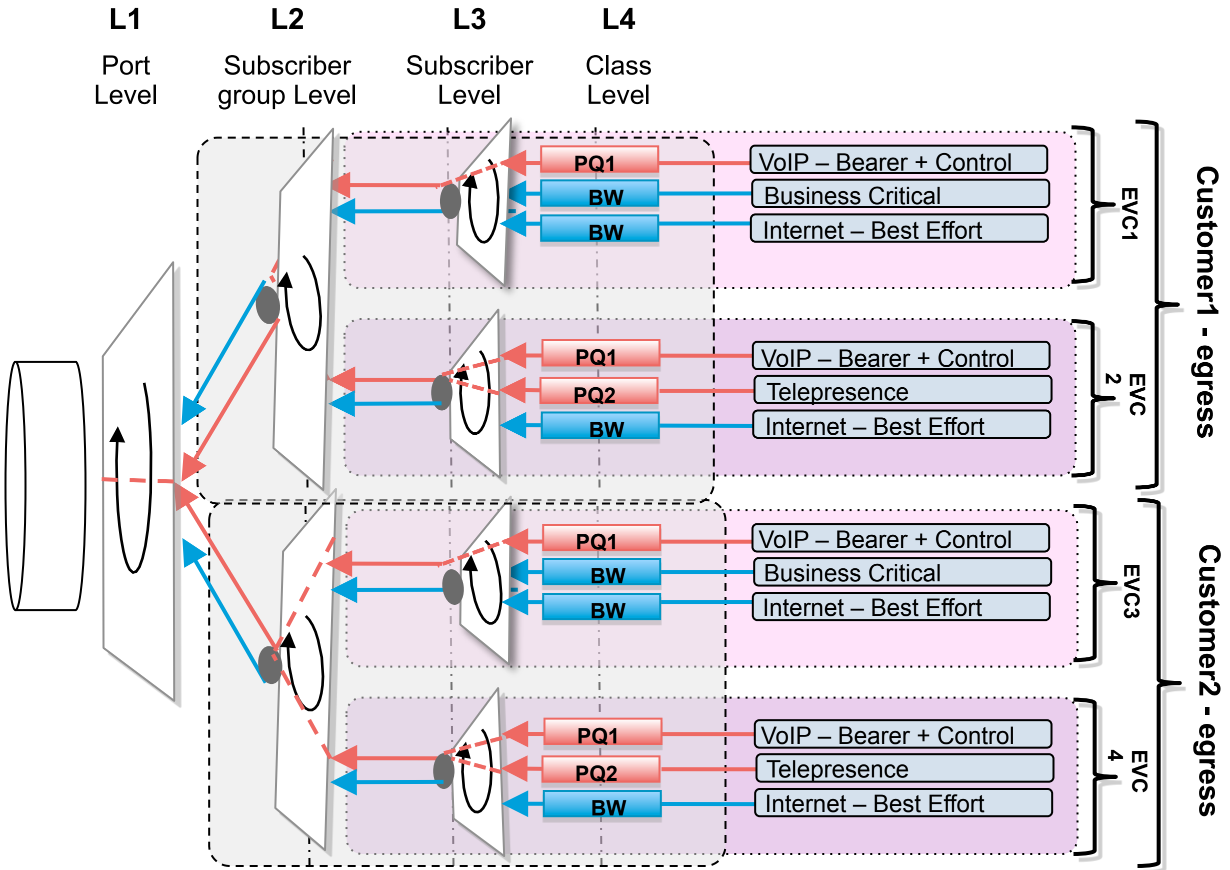
- Typhoon system (new fabric, new LC) has the same internal system qos and back pressure mechanism as existing system.
- On Trident LCs, VoQ and FIA egress queue set is per NP basis.
  - NP is 1:1 for 10GE ports
- On the new LC system, NP is designed for multiple 10G ports, 40G, and 100G port. sets of VQIs are used to represent 10/40/100G ports
  - Each 10G port is 1:1 mapped to one VQI
  - Each 40G port is mapped to 8 VQI
  - Each 100G port is mapped to 16 VQI
  - VQI's used to load balance across internal connections

Typhoon QoS Overview

- Super-set of existing Trident linecard QoS functionality
  - Dedicated TM for queuing
  - Fabric/internal QoS mechanism
  - Flexible 4-level H-qos ingress and egress
- Higher scale
  - Higher queue and policer scale
  - More granular bandwidth control for both policing and queuing
  - Higher buffer size
- Additional new feature capability
  - Conform-aware policer (a/k/a Coupled Policer)
  - 4 strict priority: P1, P2, P3 and normal priority
- Ingress TM for <=30G configs only
  - No input shaping on high-NP loading configs (36x10G, 8x10 MPA, 40G MPA)

ASR 9000 Hierarchical Traffic Management Infra

4 Layer Hierarchy Overview



Note: We count hierarchies as follows:
4L hierarchy = 3 Level nested p-map
3L hierarchy = 2 level nested p-map

L1 level is not configurable but is implicitly assumed

Hierarchy levels used are determined by how many nested levels a policy-map is configured for and applied to a given subinterface

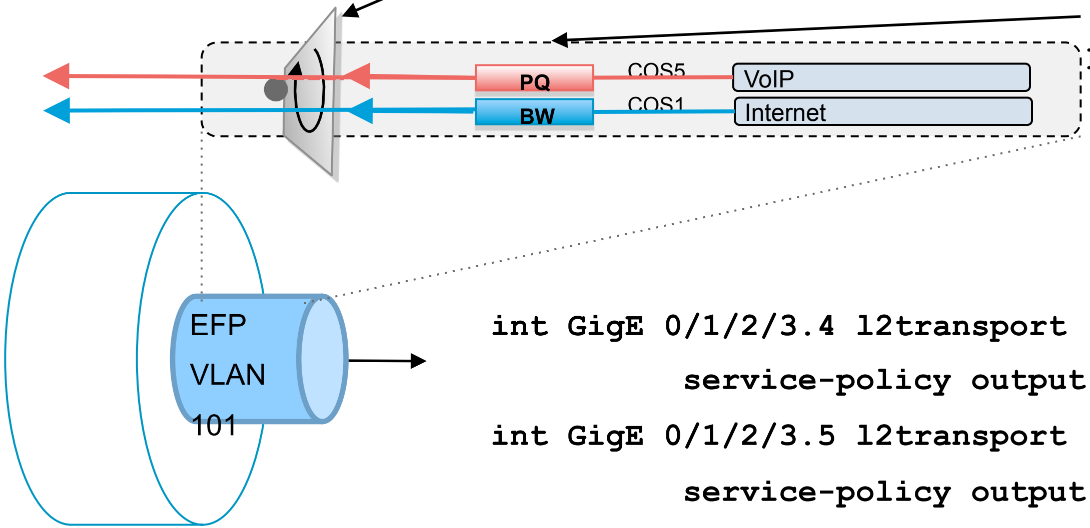
Max 8 classes (L4) per subscriber level (L3) are supported

Cisco live!

3 Layer Hierarchy Example

- Objective: Apply a SLA to an EFP with parent shape/ bandwidth/BRR and child class based queuing

```
policy parent
  class-default
    shape average 100 mbps
    bandwidth 50 mbps
    bandwidth-remaining-ratio 50
    service-policy child

policy child
  class-voip {classify on cos=5}
    priority level 1
    police 20 mbps
  class-internet {classify on cos=1}
    bandwidth 10
```




```
int GigE 0/1/2/3.4 l2transport
          service-policy output parent
int GigE 0/1/2/3.5 l2transport
          service-policy output parent
```

Increased Priority Queues

- Trident –Max of 8 Child Queues per parent , with 1 Priority 1, 1 Priority 2, and 6 Normal-priority queues (including class-default)
- Typhoon – Max 8 Child Queues per Parent – Choices based on user config in policy.
  - 1 Priority 1, 2 Priority 2 and 5 Normal-priority
  - 1 Priority 1, 1 Priority 2, 1 Priority 3, 5 Normal-Priority (Egress only)
  - 1 Priority 1, 1 Priority 2, and 6 Normal-priority

ASR 9000 QOS Functional Details

ASR9K QoS Classification Criteria

- Very flexible L2/L3 field classification on L2 interfaces
  - Inner/outer cos
  - Inner/Outer vlan *
  - DEI*
  - Outer EXP
  - Dscp/Tos
  - TTL, TCP flags, source/destination L4 ports
  - Protocol
  - Source/Destination IPv4
  - Source/Destination MAC address*
  - Discard-class
  - Qos-group
  - match all/match any

- Note:
  - Not all fields are supported on L3 interfaces*
  - Some fields don't make sense on ingress (e.g. dicard-class, qos-group)
  - MPLS classification is based on EXP only

ASR9K QoS - Classification Formats

- Per Policy-map a given classification format is chosen by SW, i.e a given policy-map can only classify based on a single format

	Format 0	Format 1	Format 2	Format 3
Fields supported	-IPV4 source address (Specific/ Range)[1] -IPV4 Destination address (Specific/ Range) -IPV4 protocol -IP DSCP / TOS / Precedence -IPV4 TTL -IPV4 Source port (Specific/Range) -IPV4 Destination port (Specific/ Range) -TCP Flags -QOS-group (output policy only) -Discard-class (output-policy only)	-Outer VLAN/COS/DEI -Inner VLAN/COS -IPV4 Source address (Specific/ Range) -IP DSCP / TOS / Precedence -QOS-group (output policy only) -Discard-class (output policy only)	-Outer VLAN/COS/DEI -Inner VLAN/COS -IPV4 Destination address (Specific/ Range) -IP DSCP / TOS / Precedence -QOS-group (output policy only) -Discard-class (output policy only)	-Outer VLAN/COS/DEI -Inner VLAN/COS -MAC Destination address -MAC source address -QOS-group (output policy only) -Discard-class (output policy only)

ASR9K QoS - Packet marking details

- "settable" packet fields:
  - dscp/precedence
  - EXP imposition
  - EXP topmost
  - cos inner/outer
  - qos-group
  - discard-class
- ASR9K supports maximum of 2 fields per class-map. The same 2 fields can be placed in any combination below
  - - 2 sets per police-conform/exceed/violate
  - - 2 sets without policing.
  - Note: In MPLS context only EXP marking is supported
  - *Remember that mpls encapped packets can't match on L3 criteria (same for ACL)*

ASR9K QoS - Policing details

- RFC 2698 supported (2r3c) and 1r2c
- Ingress & egress policing supported
- General Rule: Policing required on priority queues.
  - Priority level 2 classes can also accept shaping instead of policing.
- Granularity of 8Kbps supported (typhoon, 64k on trident)
- 2-level nested policy maps supported
  - Note: policers at parent and child work independently
- 64k policers per NP (shared for ingress/egress) on extended linecards
- Policer actions supported:
  - transmit
  - drop
  - set (implicitly behaves like set and transmit)
  - each color can have **two** set actions:

```
Policy-map parent
  Class class-default
       Police rate 10 Mbps  peak-rate 20 mbps
           conform-action set dscp af12
           conform-action set cos 2
           exceed-action set dscp af13
           exceed-action set cos 3
```

Normal Hierarchical Policer

```
policy-map child
 class class1
  police rate 20 mbps peak-rate 50 mbps
 class class2
  police rate 30 mbps peak-rate 60 mbps

policy-map parent
 class class-default
    police rate 60 mbps
    service-policy child
```

At parent level, if it's over the CIR, packet will be dropped randomly. There is no awareness which packet to be dropped

Conform Aware Policer

```
policy-map child
 class class1
  police rate 20 mbps peak-rate 50 mbps
 class class2
  police rate 30 mbps peak-rate 60 mbps

policy-map parent
 class class-default
  service-policy child

  police rate 60 mbps
   child-conform-aware
```

Parent CIR must > aggregated child CIR
Parent police only support 1R2C, child police support all: 1R2C, 2R3C, or 1R3C

If drop happen at parent level, it will drop child out-of-profile packet, but guarantee the child in-profile packet

Common Policer problems

- Note that all L2 headers are included, added to the payload and that packet size is depleting the token bucket (applies to shaping also). Only IFG and CRC are not accounted for.
- Incorrect burst size configuration, allow for some excess burst to “catch up”.
- Mistake between 2 or 3 rate policers (exceed action drop)
- Trident’s policer can’t go negative, Typhoon can borrow
  - This means that policer behavior is slightly different between the 2 hardware

ASR 9000 QoS - Queue scheduling

- “shape” for a shaped PIR for a graceful enforcement of a maximum bandwidth“
  - shaping at all configurable levels
  - Min. granularity: 64kbps (L3, L4, 256kbps for L2)
- priority levels: priority level 1, priority 2, minBw/CIR and Bw remaining
- “bandwidth” (minBw) for a CIR guarantee relative to the parent hierarchy level
  - Min. RATE: 64kbps (8k granularity)
- bandwidth remaining ratio/percent” for the redistribution of excess bandwidth that is available after PQ classes have been scheduled
  - configurable ratio values 1-1020
- Two parameter scheduler support at class level and subscriber group level (L4, L2):
  - Shape & BwR (ratio / percent)
  - Shape & MinBw (absolute / percent)
  - Not supported: BwR & MinBw on the same class

ASR 9000 QoS - congestion management/buffering details

- WRED based on: *DSCP, IPP, EXP, COS, discard-class*
- default queue-limit -to prevent buffer exhaustion- is 100ms of service rate (service rate is the sum of guaranteed bw/bwr assigned to a class)
- WRED configuration unit options are: bytes, kbytes, mbytes, us, ms, packets
  - These values will be rounded up to a set of pre-defined profiles ranging from 8 kB to 262144 kB
  - The actual implementation uses 512 byte buffer particles
- Novelty: ASR 9000 supports WRED on shaped PQ2 classes.
  - ✓ Can be used for differentiation of two kinds of priority within the PQ2 class

Absolute vs Percentage

- All relevant policy actions support both, absolute and percentage based configuration:
  - shape
  - bandwidth
  - Police
  - bandwidth remaining*
- For tri-rate Copper SFPs (10/100/1000) percentage based QOS will be adjusted automatically based on the selected rate

*Note: Bandwidth remaining supports ratio/percent, not absolute bandwidth

Show/debug QOS commands

show running-config	
show running-config policy-map <policyname>	Policy map configuration
show running-config class-map <classmap>	Class map configuration
show running-config interface <interface>	Interface running configuration
show policy-map interface <interface> [iNPt | output]	Policy-map statistics on a particular non-bundle interface
show policy-map interface <bundle-interface> [iNPt|output] member	Policy-map statistics on a member of bundle interface
show qos interface <interface> <iNPt|output> [member <interface>]	Displays hardware and software configured values of each class for a service-policy on an interface
show qos-ea interface <interface> <iNPt|ouput> [member <interface>] [detail]	Displays the detailed information of hardware and software configured paramters in each class of a service-policy on an interface
show qos summary <police|policy|queue> [interface <interface>] [output| iNPt] [member <interface>]	Lists the summary of all queues or policers or interfaces for a policy
show qoshal tm-config <all|counters|fcu|general|priority|shape|topology| wfq|wred> np <np> tm <tm>	Displays generic NP TM config
show qoshal <wfq|wred|wred-scale|shape|police|police-node> np <np> tm <tm> level <level> profile <profile> <num-of-profiles> [hw|sw]	Displays various profiles configured in sw and hw and the values of each profile

Show/debug QOS commands - contd

<table>
<tr><td>show qoshal resource summary [np <np>]</td><td>Displays the summary of all the resources used in hardware and software for QoS such number of policy instances, queues, profiles</td></tr>
<tr><td>show qoshal fcu <limits|status|profile></td><td>Displays all Traffic Manager (TM) Flow control related info</td></tr>
<tr><td>show qoshal ha chkpt <all|<chkpt-tbl-name> {all|<recid>|info}</td><td>Display HA related info for PRM QoS HAL</td></tr>
<tr><td>show qos-ea ha state</td><td>Displays the HA State of process QoS EA whether it can accept the service-policies</td></tr>
<tr><td>show qos-ea ha chkpt <all|<chkpt-tbl-name> {all|<recid>|info}</td><td>Display HA Chkpt related info for all the chkpt tables for QoS EA</td></tr>
<tr><td>show qos-ea trace {all|errors|events|internal}</td><td>Displays the trace of errors or events or internal events of QoS EA process</td></tr>
<tr><td>show prm server trace hal</td><td>Displays all the trace info of PRM QoS HAL thread</td></tr>
<tr><td>debug qos-ea all</td><td>Debug commands for qos ea process</td></tr>
<tr><td>debug qoshal <level|module|events> <word></td><td>Debug commands for PRM qos HAL</td></tr>
<tr><td>debug prm server hal <all|error|events></td><td>Debug commands for PRM qos HAL API</td></tr>
</table>

Troubleshooting Back-pressure Issues

- **Check if you are seeing FIA drops**

```
RP/0/RSP1/CPU0:ios#show drops location 0/0/CPU0
=== snip ===
FIA 0 Drops:
----------------------------------------------------------------
Ingress Drops                                          287078960
Egress Drops                                           1
Total Drops                                            287078961
Ingress Generic Hard Drop-2                            287078960
Egress Mcast RxFab Hdr-1                               1
----------------------------------------------------------------
```

- **Check if any VQI is dropping packet**

```
RP/0/RSP1/CPU0:ios#show controller fabric fia q-depth location 0/0/CPU0
FIA  0
VoQ   | ddr | pri | pkt_cnt
------+-----+-----+---------
23    | 0   | 2   | 118
Total Pkt queue depth count = 118  ← Packets in the queue. Not good.
```

Troubleshooting Back-pressure Issues

- **Check if you are seeing FIA drops**

```
RP/0/RSP1/CPU0:ios#show controllers pm interface tenGigE 0/5/0/0 loc 0/5/CPU0
Ifname(1): TenGigE0_5_0_0, ifh: 0xe000100 :
switch_fabric_port 0x17  ← VQI 23 is for interface ten0/5/0/0
RP/0/RSP1/CPU0:ios#
```

- **Check egress NP TM Drops:**

- RP/0/RSP1/CPU0:ios#show controllers NP tm counters all location 0/5/CPU0

```
                Node: 0/5/CPU0:
==== TM Counters (NP 3 TM 1) ====
 TM Counters: commit_xmt_paks: 1509333316
 excess_xmt_paks:  67641555690
 Total Transmitted paks: 69150889006
 wred_drop paks: 2441836834 timeout_drop 0 intf_drop 0
==== TM Counters (NP 3 TM 2) ====
 TM Counters: commit_xmt_paks: 0
 excess_xmt_paks:  0
 Total Transmitted paks: 0
 wred_drop paks: 0 timeout_drop 0 intf_drop 0
RP/0/RSP1/CPU0:ios#
```

What consumes a queue

- Bandwidth, Priority and Shaping will consume a queue
- On ingress, priority setting will not consume a queue

RP/0/RSP0/CPU0:A9K-BNG#show qos int g 0/0/0/0 out | i "**QueueID|Level|Class**"

Thu Mar 28 13:48:56.683 EDT

Level: 0 Policy: **SHAPE Class**: class-default

QueueID: N/A

Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 1

Level: 1 Policy: child Class: class1

Parent Policy: SHAPE Class: class-default

QueueID: 136 (Priority 1)

Level: 1 Policy: child Class: class2

Parent Policy: **SHAPE Class**: class-default

QueueID: 138 (Priority Normal)

Bandwidth: 0 kbps, BW sum for Level 1: 0 kbps, Excess Ratio: 70

Class name

Queuing level

QueueID And priority class

Child class belonging to parent class

Computed BW ratio (based on class rate over parent shape rate

What is programmed in HW?

COMMAND: **show qos int g 0/0/0/0 out**

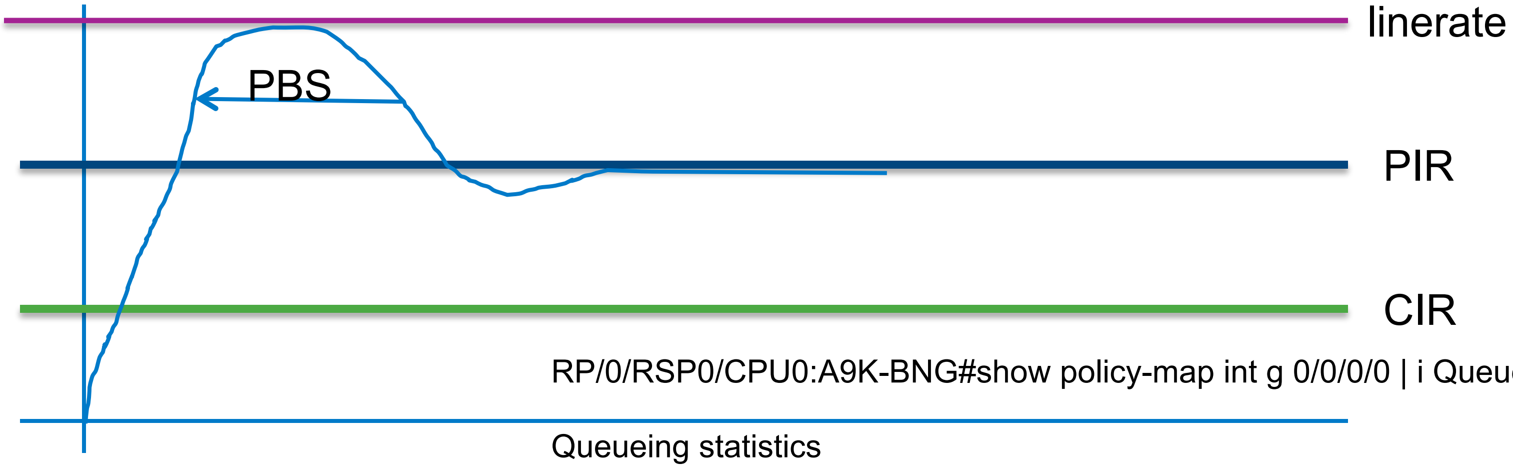
```
-------------------------------------------------------------------
Level: 0 Policy: xtp Class: class-default
QueueID: N/A
Shape CIR : NONE
Shape PIR Profile : 0/4(S) Scale: 195 PIR: 199680 kbps  PBS: 2496000 bytes
WFQ Profile: 0/9 Committed Weight: 10 Excess Weight: 10
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 1
-------------------------------------------------------------------
```

```
policy-map xtp
 class class-default
  service-policy xt
  shape average 200 mbps
 !
 end-policy-map
```

- Rate is rounded to the nearest 8k or 64k value
- Shape sets PIR
- PBS is default rate of 100msec of configured shape rate
- BW is zero or 64k, only applicable in oversubscription at sum of parent levels

Shaping with PIR/PBS and CIR

- Shaper peaks to linerate for pbs time
- Should allow some burst to get to PIR faster
- CIR is ignored, will result in queue(exceed) counts, but they don't mean drops!



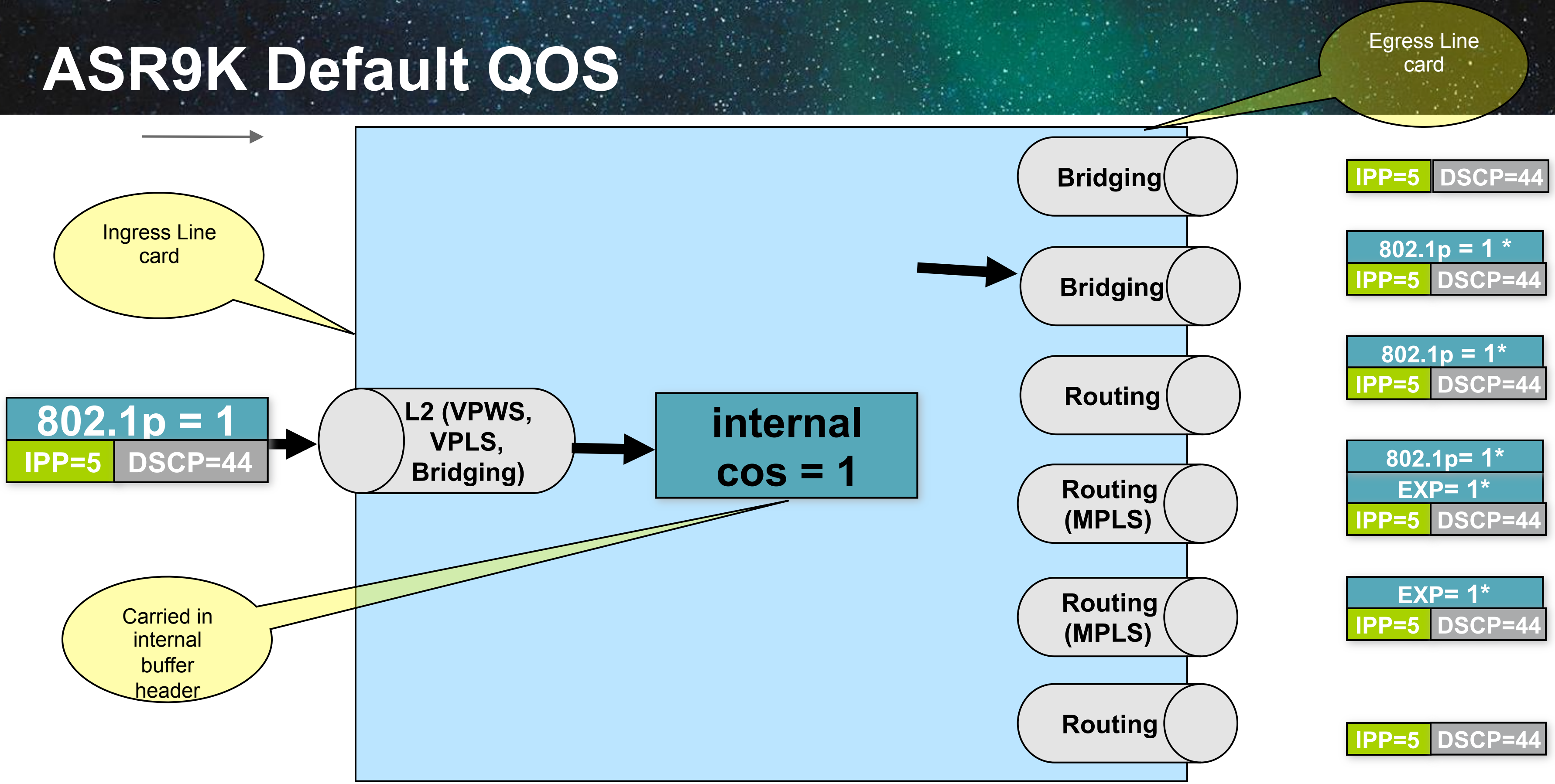
RP/0/RSP0/CPU0:A9K-BNG#show policy-map int g 0/0/0/0 | i Queue

Queueing statistics
  Queue ID : 136
  Queue(conform) : 0/0 0
  Queue(exceed) : **0/0** **0**

QOS summary

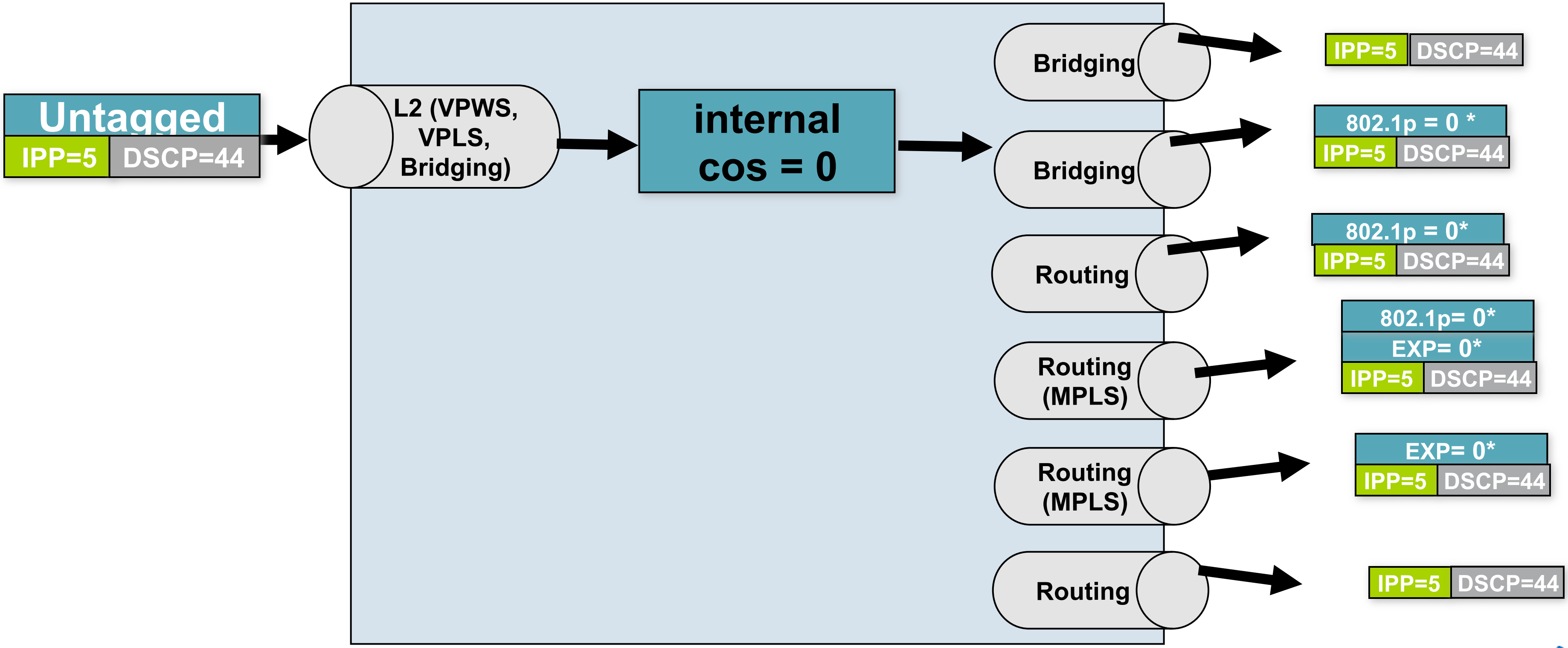
- All Ethernet linecards support Queuing, Marking and Policing.
- Some high speed linecards do not support ingress Queuing (but support policing and marking).
  - Because their ingress TM (Traffic Manager) is disabled
- To guarantee priority end to end, make sure high priority traffic is marked on ingress (This will not burn a queue)
- https://supportforums.cisco.com/docs/DOC-15592

ASR9K Default QOS



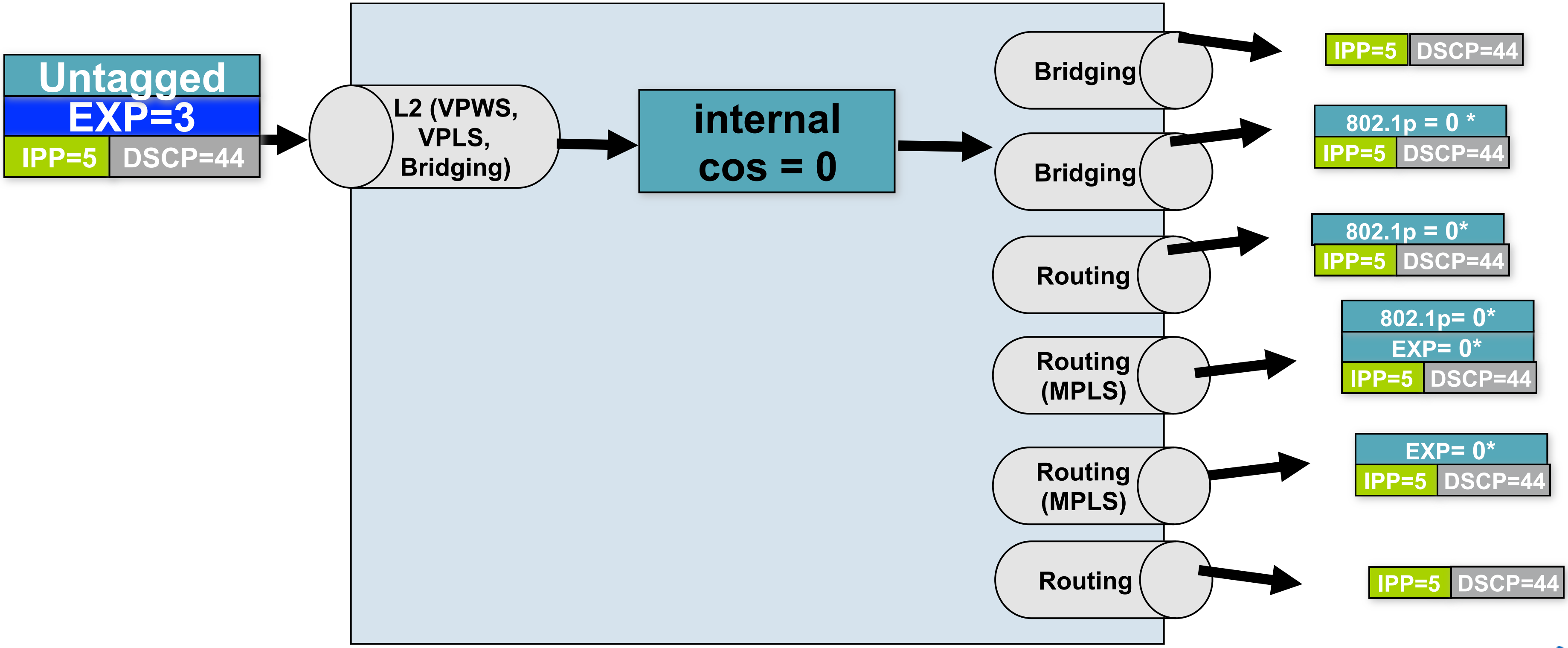
Note: VPWS will be treated like a L2 operation on ingress - Applies for all tags/labels in the stack that get imposed. Not for VLAN translation. Bridging on egress without adding an vlan header is an hypothetical case – in case we have a need.
IPP = IP Precedence, showing IPP & DSCP seperately since policymap can treat precedence and dscp separately as required.

ASR9K Default QOS



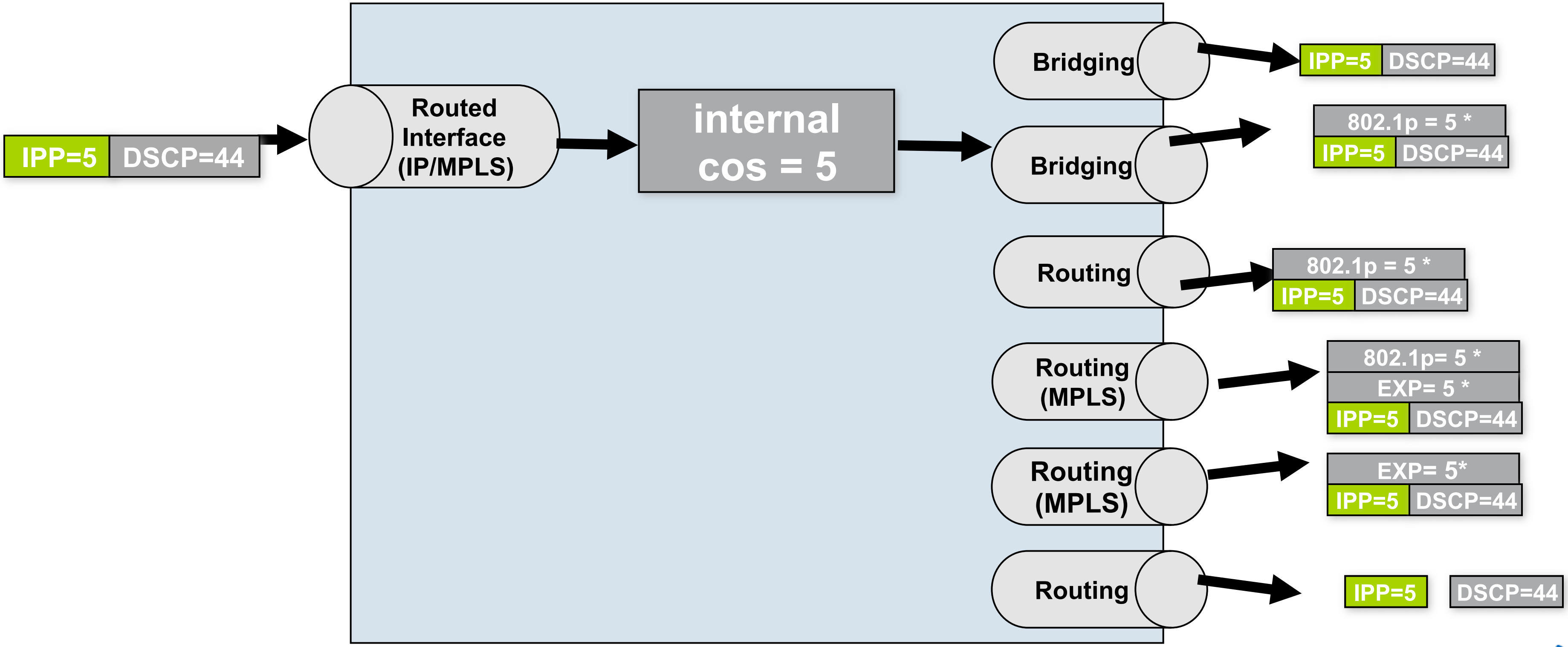
Note: Trust cos in case of bridged interfaces in ingress. For untagged packets use cos = 0.
* - Applies for all tags/labels in the stack that get imposed.

ASR9K Default QOS



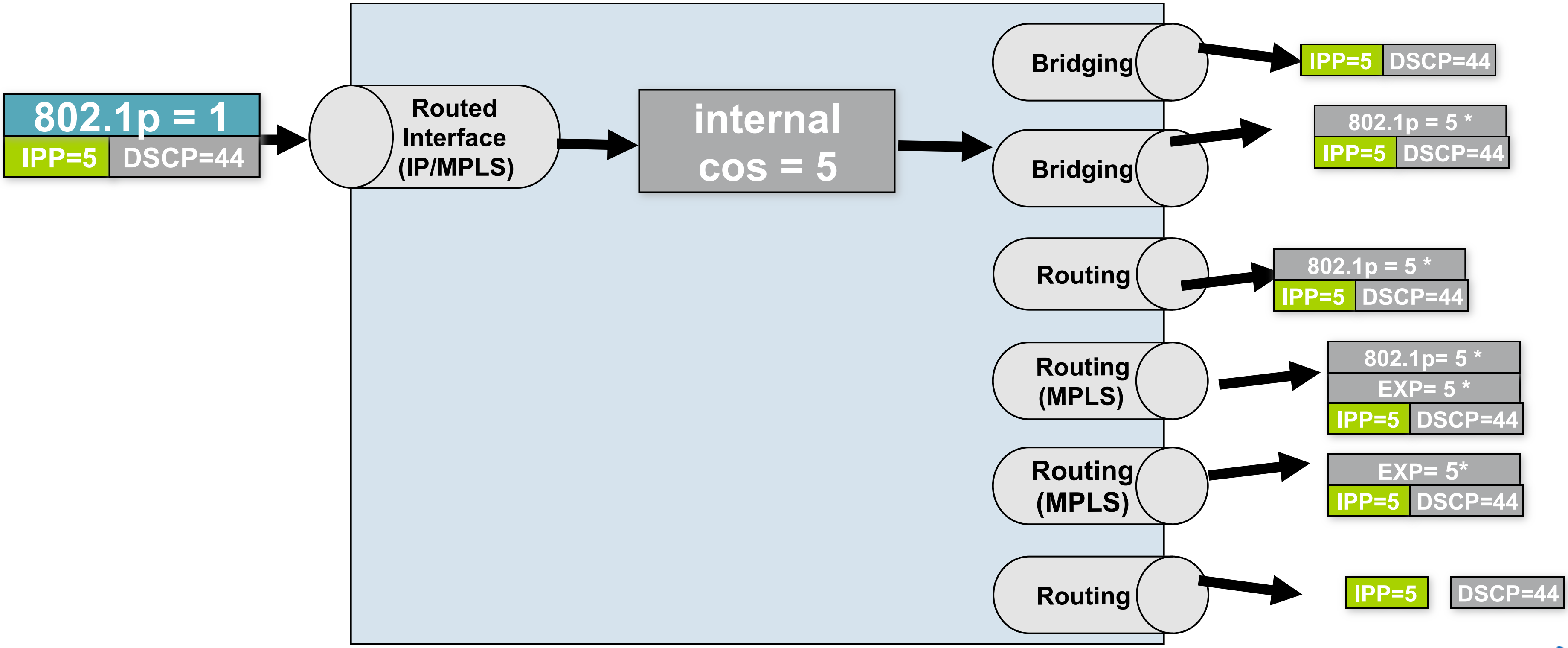
Note: Trust cos in case of bridged interfaces in ingress. For untagged packets use cos = 0.
•- Applies for all tags/labels in the stack that get imposed.
•- Explicit NULL EXP is treated the same as an topmost EXP of non NULL labels.

ASR9K Default QOS



Note: Trust dscp in case of routed interfaces in ingress. For Non IP packets use cos = 0
* - Applies for all tags/labels in the stack that get imposed.

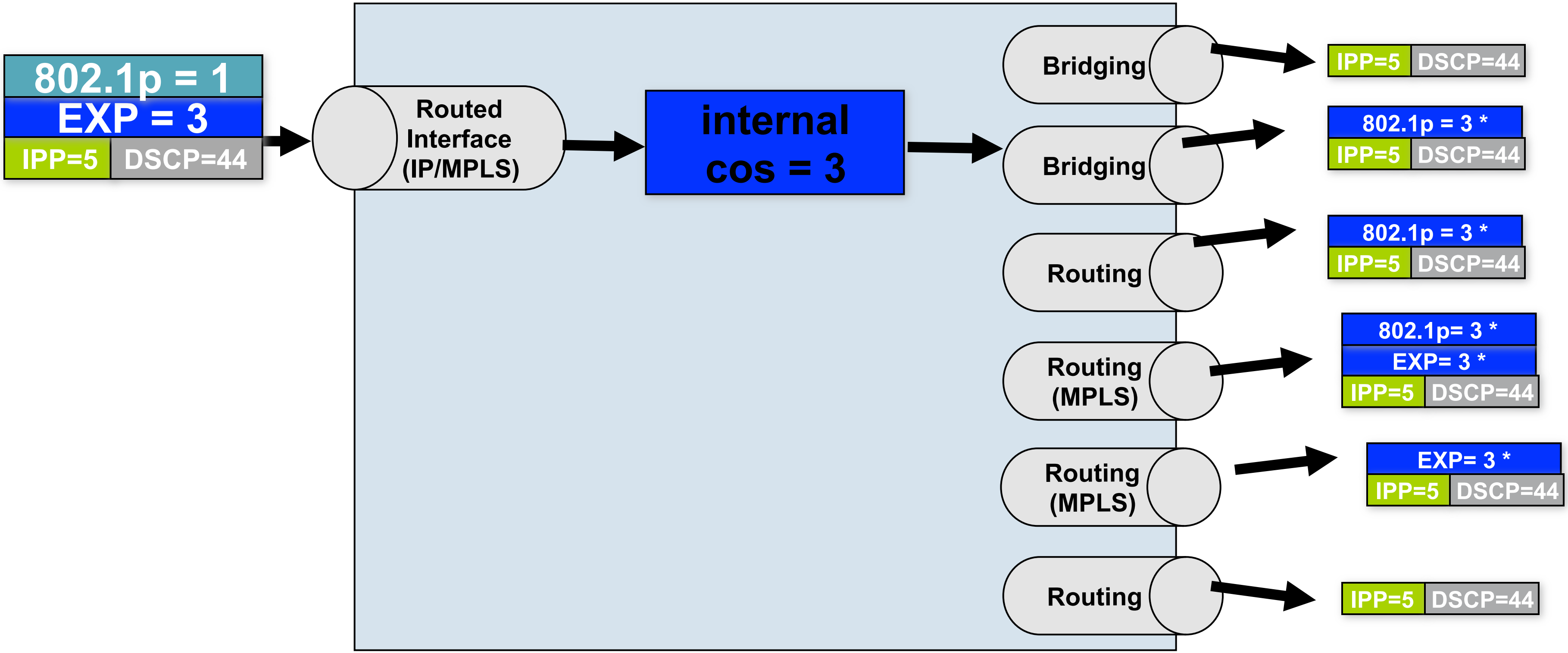
ASR9K Default QOS



Note: Trust dscp in case of routed interfaces in ingress. For Non IP packets use internal dscp= 0

* - Applies for all tags/labels in the stack that get imposed.

ASR9K Default QOS



Note: Trust EXP/dscp in case of routed interfaces in ingress. For Non IP packets use internal dscp= 0. Do not overwrite DSCP fields exposed during disposition – to support pipe mode by default.

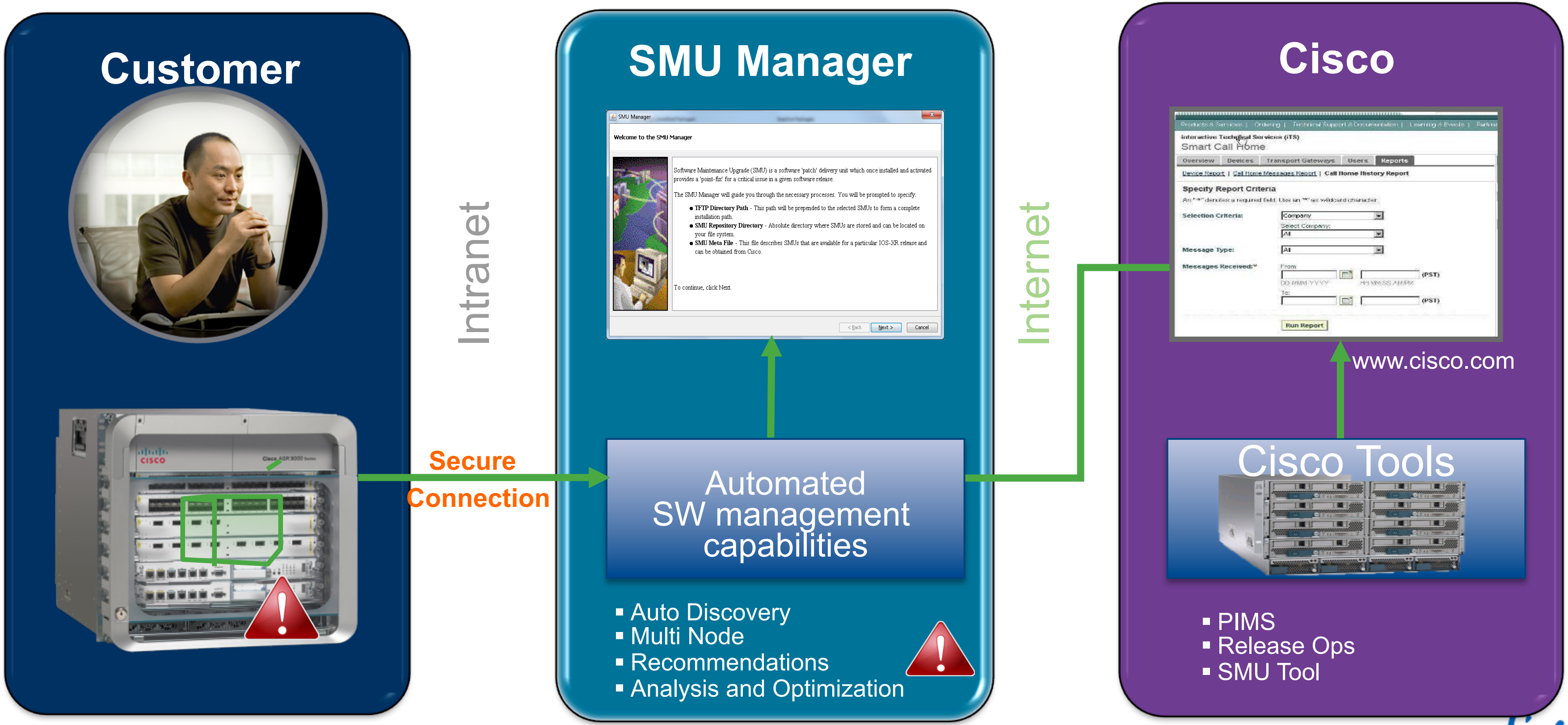
* - Applies for all tags/labels in the stack that get imposed.

Few words IOS-XR and IOS differences

What are key differences between IOS and XR

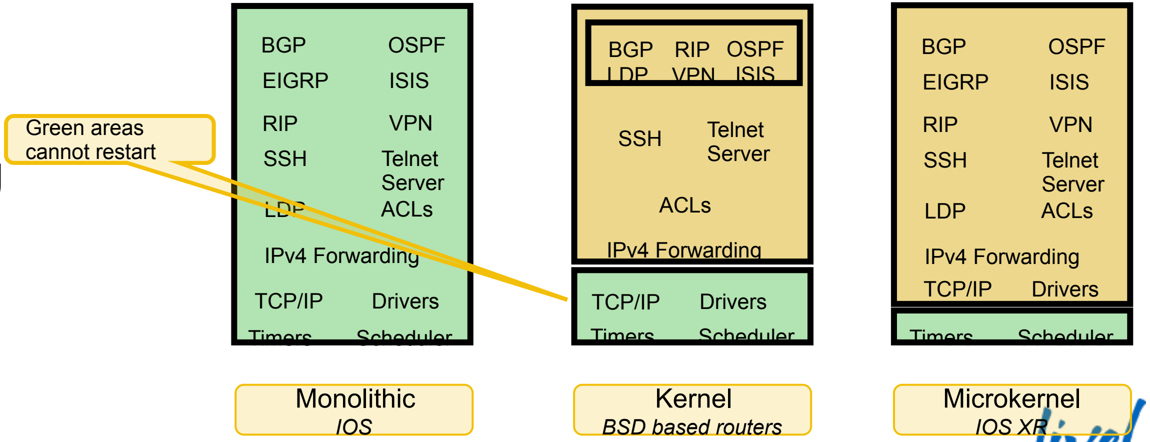
- Micro kernel vs Monolithic
  - Process crashes are confined in XR
  - Ability to patch individual processes (via SMU's) (SMU manager tool!)
- SNMP architectural differences (caching)
- IPC (inter process communications)
- Memory management and CPU utilization
- EVC model (as opposed to IEEE in IOS)
- Routing protocol behavioral differences
  - E.g. RPL instead of route-maps
  - E.g. BGP no sync and deterministic MED is always on things like that
- Task based command author
- Two stage commit
- **Google ASR9000 ios to xr migration guide**

SMU Management Architecture



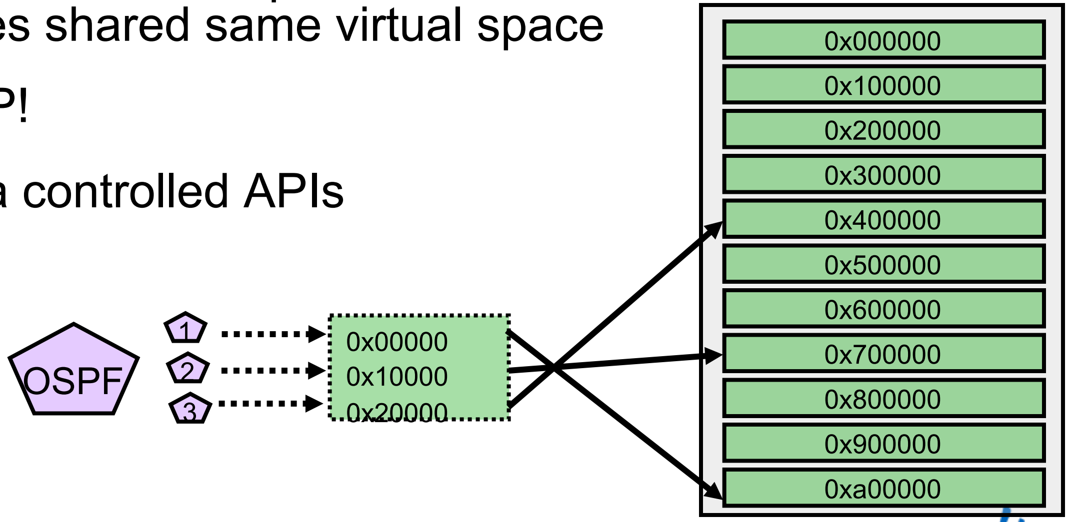
MicroKernel instead of Monolithic

- Complete Micro Kernel allowing for individual process restarts
- No runaway processes
- One misbehaving process will not affect another
- Patchable at the individual process level
- Process isolation
- Process restart
- Preemptive multitasking



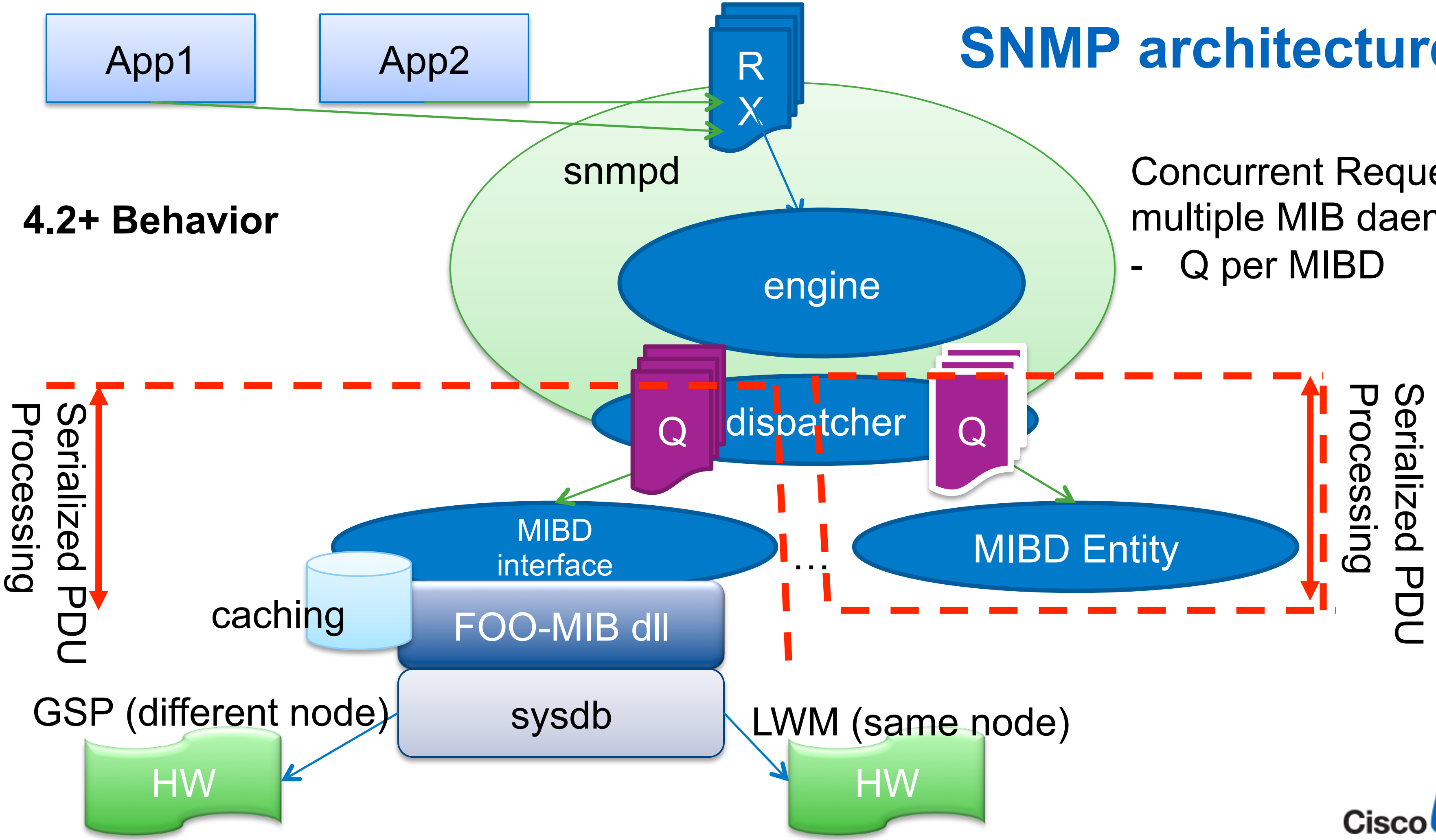
Virtual memory spaces and allocation

- Each process has its own dedicated memory space
- Mapped to real HW addresses invisible to process
- One process cannot corrupt another's memory
  - Process can only access virtual space
  - In IOS – all processes shared same virtual space
- No more SYS-MEMDUMP!
- Comm. between procs via controlled APIs



SNMP architecture

4.2+ Behavior



Concurrent Requests to multiple MIB daemons

- Q per MIBD

ASR 9000 Flexible Ethernet SW Infrastructure
(“EVC” SW Infrastructure)

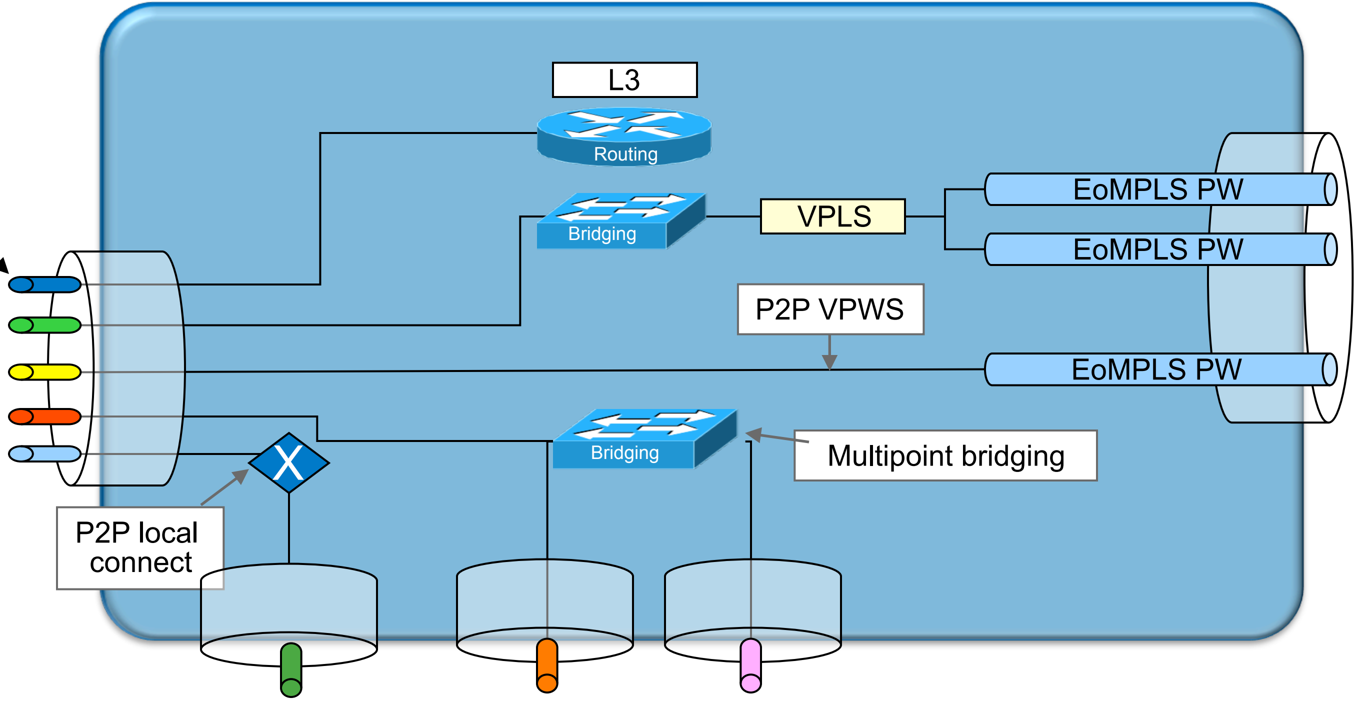
EFP (**E**thernet **F**low **P**oint) or sub-interface

Flexible VLAN tag classification

Flexible Ethertype (.1Q, QinQ, .1ad)

Flexible VLAN tag manipulation

1



2

Flexible service mapping and multiplexing

L2 and L3, P2P and MP services concurrently on the same port

Cisco live!

Flexible Service – L2VPN P2P

EFP configuration example

```
Interface gig 0/0/0/1.101 l2transport
encapsulation dot1q 101 second 10
rewrite ingress pop 2 Symmetric

Interface gig 0/0/0/2.101 l2transport
encapsulation dot1q 101
rewrite ingress pop 1 Symmetric

Interface gig 0/0/0/3.101 l2transport
encapsulation dot1q 102
rewrite ingress push dot1q 100 Symmetric
```

L2VPN P2P service configuration example

```
l2vpn
 xconnect group cisco
   p2p service1 ← local connect
     interface gig 0/0/0/1.101
     interface gig 0/0/0/2.101
   p2p service2 ← VPWS
     interface gig 0/0/0/3.101
     neighbor 1.1.1.1 pw-id 22
   p2p service3 ← PW stitching
     neighbor 2.2.2.2 pw-id 100
     neighbor 3.3.3.3 pw-id 101
```

- Two logical ports (EFP or PW) form one EVC (Ethernet virtual circuit)
- No MAC learning/forwarding involved

IOS-XR vs. IOS EVC Comparison

- **Common part**
  - Both share the same EVC SW infrastructure
  - Feature parity for the flexible VLAN tag classification, VLAN tag rewrite and service mapping

- **7600 IOS**
  - VLAN tag classification, rewrite, service mapping are all done on the port level (with some exceptions), which is classic IOS CLI
  - Introduced “service instance” configuration mode for better L2VPN scale
  - Legacy switchport feature support in parallel (but can’t co-exist with EVC on the same port)
  - IEEE trunks
  - Interface VLAN

- **ASR 9000 IOS-XR**
  - De-couple port level and service configuration. VLAN tag classification and rewrite are done at port level. L2VPN services are configured at “l2vpn” module
  - Uniform “sub-interface” CLI for both L2 and L3 service, no additional “service instance” structure
  - Common Infrastructure for native L2 and MPLS based L2VPN service
  - EFP based access model.
  - Bridge domain per vlan
  - BVI

EVC Configuration Comparison (1) – L2VPN P2P service

<table>
<tr><th></th><th>ASR 9000</th><th>7600</th></tr>
<tr><td>Local Connect</td><td rowspan="3"><u>EFP configuration under interface</u>

Including VLAN tag encapsulation, tag rewrite, Qo/ACL features, etc

Interface gig 0/0/0/1.101 l2transport
encapsulation dot1q 101 second 10
rewrite ingress tag pop 2 Symmetric

Interface gig 0/0/0/2.101 l2transport
encapsulation dot1q 101
rewrite ingress tag pop 1 Symmetric

<u>Service configuration under “l2vpn”</u>

l2vpn
xconnect group cisco
p2p service1 ← local connect
interface gig 0/0/0/1.101
interface gig 0/0/0/2.101
p2p service2 ← EoMPLS
interface gig 0/0/0/3.101
neighbor 1.1.1.1 pw-id 22
p2p service3 ← PW stitching
neighbor 2.2.2.2 pw-id 100
neighbor 3.3.3.3 pw-id 101</td><td>interface GigabitEthernet4/1/0
service instance 101 ethernet
encapsulation dot1q 101 second 10
rewrite ingress tag pop 2 Symmetric

interface GigabitEthernet4/1/1
service instance 100 ethernet
encapsulation dot1q 100
rewrite ingress tag pop 1 Symmetric

connect eline-101 GigabitEthernet4/1/0 101 GigabitEthernet4/1/1 100</td></tr>
<tr><td>EoMPLS</td><td>interface GigabitEthernet4/1/1
service instance 11 ethernet
encapsulation dot1q 101 second-dot1q 60-70
xconnect 10.0.0.3 101 encapsulation mpls</td></tr>
<tr><td>PW stitching</td><td>l2 vfi tac-training point-to-point
neighbor 10.0.2.3 3001 encapsulation mpls
neighbor 10.0.2.2 3000 encapsulation mpls

[note] require BGP configuration if it’s for inter-AS</td></tr>
</table>

Flexible Service – L2VPN Multi-Point

EFP configuration example

```
Interface gig 0/0/0/1.101 l2transport
encapsulation dot1q 101
rewrite ingress pop 1 Symmetric

Interface gig 0/0/0/2.101 l2transport
encapsulation dot1q 101
rewrite ingress pop 1 Symmetric

Interface gig 0/0/0/3.101 l2transport
encapsulation dot1q 102
rewrite ingress push dot1q 100 Symmetric
```

- More than two logical ports (EFP or PW) belong to the same bridge domain
- MAC learning/forwarding involved
- Bridge-domain is global significant, VLAN ID is local port scope

L2VPN MP service configuration example

```
l2vpn
 bridge group cisco
  bridge-domain domain1 ← local bridging
   Interface gig 0/0/0/1.101
      split-horizon group ← no bridging among same SHG
   Interface gig 0/0/0/2.101
      split-horizon group

  bridge-domain domain2 ← vpls
   Interface gig 0/0/0/1.101
   Interface gig 0/0/0/2.101
   vfi cisco
    neighbor 192.0.0.1 pw-id 100
    neighbor 192.0.0.2 pw-id 100

  bridge-domain domain3 ← h-vpls
   Interface gig 0/0/0/1.101
   neighbor 192.0.0.3 pw-id 100 ← spoke PW
   vfi cisco ← core PWs
    neighbor 192.0.0.1 pw-id 100 ← core PW
    neighbor 192.0.0.2 pw-id 100
```

CLI Comparison (4) – SVI

ASR 9000 IRB/BVI* Example (equivalent to 7600 SVI feature)

```
Interface gig 0/0/0/1.50 l2transport
encapsulation dot1q 50
rewrite ingress tag pop 1 Symmetric

Interface gig 0/0/0/2.50 l2transport
encapsulation dot1q 50
rewrite ingress tag pop 1 Symmetric
```

```
l2vpn
 bridge group cisco
 bridge-domain domain50
   Interface gig 0/0/0/1.50
   Interface gig 0/0/0/2.50
   routed interface bvi 20

Interface bvi 20
  ipv4 address 1.1.1.1 255.255.255.0
```

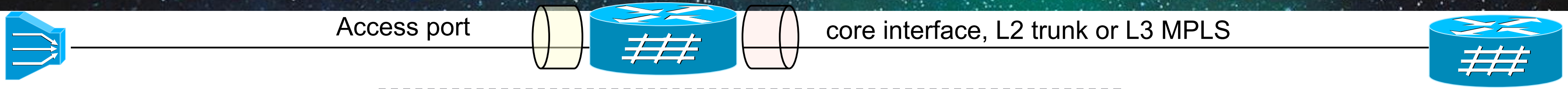
7600 SVI example

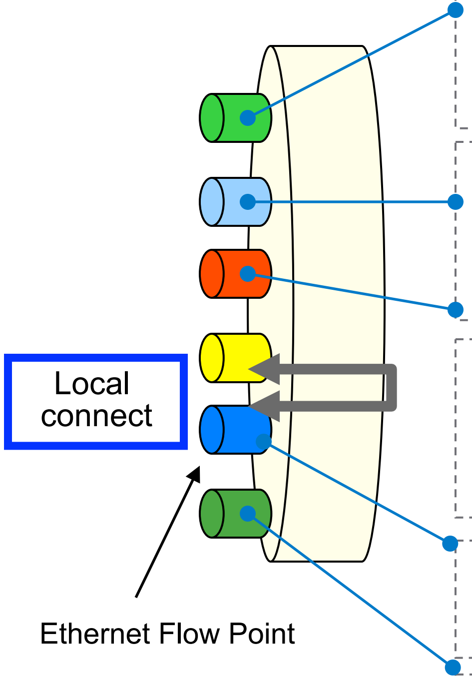
```
interface gig 1/2
  switchport
  switchport mode trunk
  switchport trunk allow vlan 50-1000
```

```
interface GigabitEthernet4/1/0
 service instance 2 ethernet
  encapsulation dot1q 50
  rewrite ingress tap pop 1 sym
  bridge-domain 50

Interface vlan 50
  ip address 1.1.1.1 255.255.255.0
```

*QOS policing and ACL supported on BVI starting XR43.
(features replicated to all npu's with EFPs in that BD!

Multiple Services on the same port example






```
interface gig 0/0/0/1.1 l2transport
  encapsulation dot1q 20 second-dot1q 10
  rewrite ingress tag pop 1 sym
l2vp
  bridge group cisco
  bridge-domain cisco
    interface gig 0/0/0/1.1
    interface …
```

Local Bridging

```
interface gig 0/0/0/1.2 l2transport
  encapsulation dot1q 11-100
  rewrite ingress tag push dot1q 101
L2vpn
 xconnecct group
  p2p eline-1
   interface gig 0/0/0/1.2
   neighbor 1.1.1.1 pw-id 101
```

E-LINE (VPWS)

```
Interface gig 0/0/0/1.3 l2transport
  encapsulation dot1q 101 second-dot1q 10
  rewrite ingress tag translate 2-to-1 100
l2vpn bridge group vpls
bridge-domain vpls
  interface gig 0/0/0/1.3
  vfi APPLE
    neighbor 20.20.20.20 pw-id 200
```

E-LAN (VPLS)

```
l2vpn
 xconnect group LocalConnect
  p2p someone
   interface GigabitEthernet0/0/0/1.5
   interface GigabitEthernet0/0/0/1.6
```

Local connect

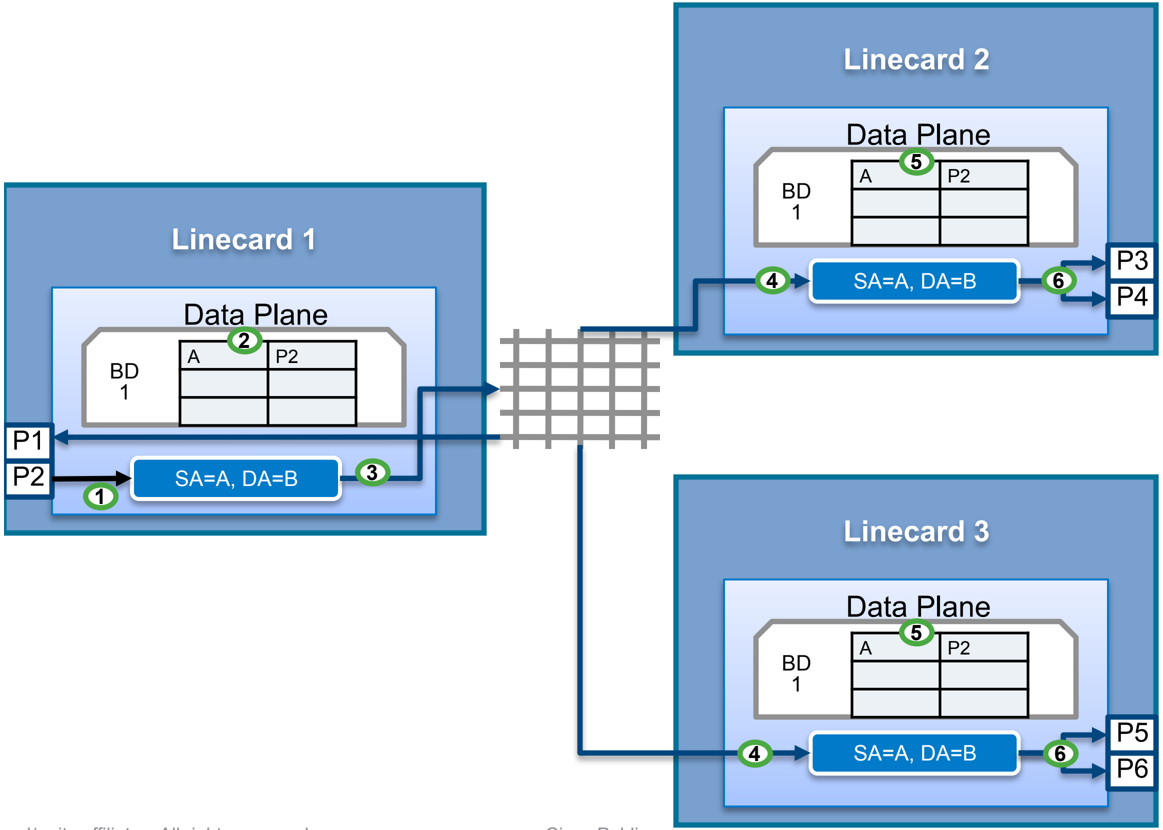
```
interface gig 0/0/0/1.100
  encapsulation dot1q 200 second 200
  ipv4 address 1.1.1.1 255.255.255.0
```

L3 service

MAC Learning – Learn from Data Plane Flooding
DMAC unknown/broadcast

Precondition: SMAC unknown, DMAC unknown/broadcast

1. Frame with unknown SMAC & DMAC address enters the system on LC1 into BD1
2. MAC lookup, MAC table on LC1 is updated with SMAC (ingress data-plane learning)
3. Since DMAC is unknown, frame is flooded towards linecards which participate in BD and to locally attached ports
4. LC2 and LC3 receive flooded frame copy with unknown SMAC & DMAC into BD1
5. MAC lookup, MAC table on LC2, LC3 is updated with SMAC (egress data-plane learning)
6. Since DMAC is unknown, frame is flooded towards local bridge ports on BD1



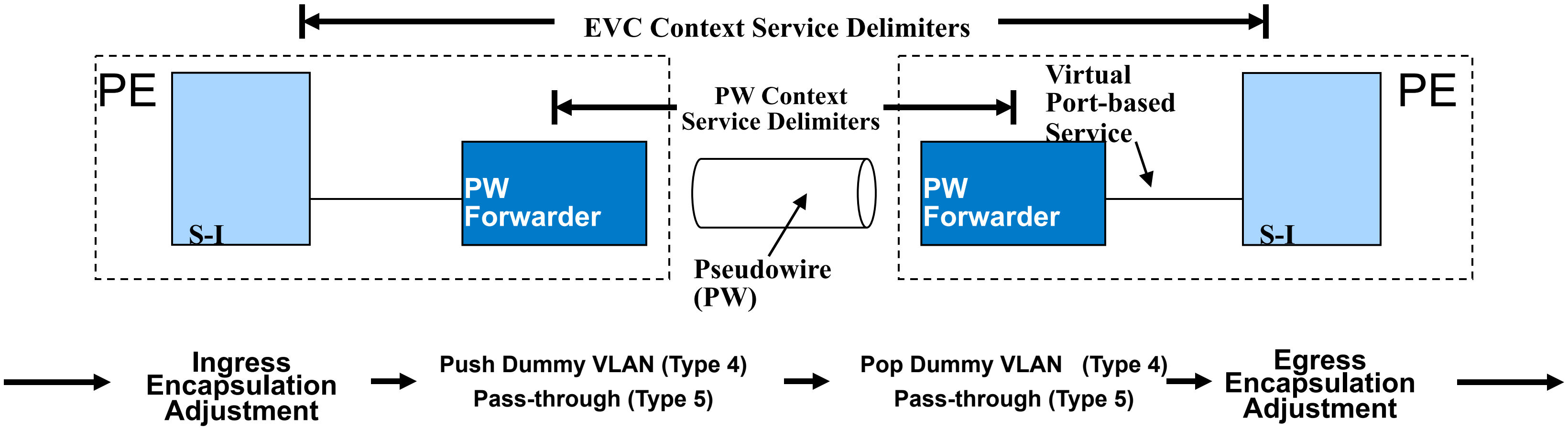
MAC withdrawal / flush

- A Flush is done on a per port basis, but with a mac wildcard.
- This means that a vpls ldp mac withdrawal message is sent to flush basically all macs in the Bridge domain.
- This means that the Bridge domain will start to flood for a little bit, but this is no problem considering we have hardware learning.

- Pay attention to the MAC_MOVE np counter
- MAC_NOTIFY is an update for learning a new mac. The npu will generate and flood a mac-notify to all npu's in the system (regardless whether they have a bridge-domain or not)

VLAN rewrite Considerations

VLAN Tags and Pseudowires



- EVC Encapsulation Adjustment is independent of negotiated Pseudowire (PW) Type; PW type dictates VLAN adjustment in PW Forwarder only
- For Ethernet PW (Type 5), frames pass through PW Forwarder with the Ethernet header unmodified
- For VLAN PW (Type 4), the PW Forwarder adds Dummy VLAN in imposition path and rewrites that VLAN in disposition path
- Golden rule, always "pop" the service delimit VLAN tag regardless of the VC type

References

- ASR9000/XR Feature Order of operation
- ASR9000/XR Frequency Synchronization
- ASR9000/XR: Understanding SNMP and troubleshooting
- Cisco BGP Dynamic Route Leaking feature Interaction with Juniper
- ASR9000/XR: Cluster nV-Edge guide
- Using COA, Change of Authorization for Access and BNG platforms
- ASR9000/XR: Local Packet Transport Services (LPTS) CoPP
- ASR9000/XR: How to capture dropped or lost packets
- ASR9000/XR Understanding Turboboot and initial System bring up
- ASR9000/XR: The concept of a SMU and managing them
- ASR9000/XR Using MST-AG (MST Access Gateway), MST and VPLS
- ASR9000/XR: Loadbalancing architecture and characteristics
- ASR9000/XR Netflow Architecture and overview
- ASR9000 Understanding the BNG configuration (a walkthrough)
- ASR9000/XR NP counters explained for up to XR4.2.1
- ASR9000/XR Understanding Route scale
- ASR9000/XR Understanding DHCP relay and forwarding broadcasts
- ASR9000/XR: BNG deployment guide

References

- ASR9000/XR: Understanding and using RPL (Route Policy Language)
- ASR9000/XR What is the difference between the -p- and -px- files ?
- ASR9000/XR: Migrating from IOS to IOS-XR a starting guide
- ASR9000 Monitoring Power Supply Information via SNMP
- ASR9000 BNG Training guide setting up PPPoE and IPoE sessions
- ASR9000 BNG debugging PPPoE sessions
- ASR9000/XR : Drops for unrecognized upper-level protocol error
- ASR9000/XR : Understanding ethernet filter strict
- ASR9000/XR Flexible VLAN matching, EVC, VLAN-Tag rewriting, IRB/BVI and defining L2 services
- ASR9000/XR: How to use Port Spanning or Port Mirroring
- ASR9000/XR Using Task groups and understanding Priv levels and authorization
- ASR9000/XR: How to reset a lost password (password recovery on IOS-XR)
- ASR9000/XR: How is CDP handled in L2 and L3 scenarios
- ASR9000/XR : Understanding SSRP Session State Redundancy Protocol for IC-SSO
- ASR9000/XR: Understanding MTU calculations
- ASR9000/XR: Troubleshooting packet drops and understanding NP drop counters
- Using Embedded Event Manager (EEM) in IOS-XR for the ASR9000 to simulate ECMP "min-links"
- XR: ASR9000 MST interop with IOS/7600: VLAN pruning

Summary

So what have we discussed today

- ASR9000 architecture overview
  - Fabric and Linecards
- How the NPU forwarders work
- How to troubleshoot the ASR9000 packet forwarding issues
- Loadbalancing
- Punt Path
- Multicast
- QOS architecture
- Quick Comparison between IOS and XR
- L2VPN/EVC configuration model and Mac learning

Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Cisco Daily Challenge points for each session evaluation you complete.
- Complete your session evaluation online now through either the mobile app or internet kiosk stations.

Maximize your Cisco Live experience with your free Cisco Live 365 account. Download session PDFs, view sessions on-demand and participate in live activities throughout the year. Click the Enter Cisco Live 365 button in your Cisco Live portal to log in.

CISCO™