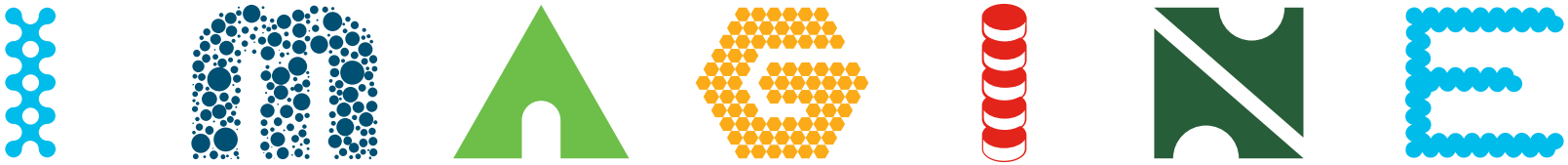


INTUITIVE

Cisco live!

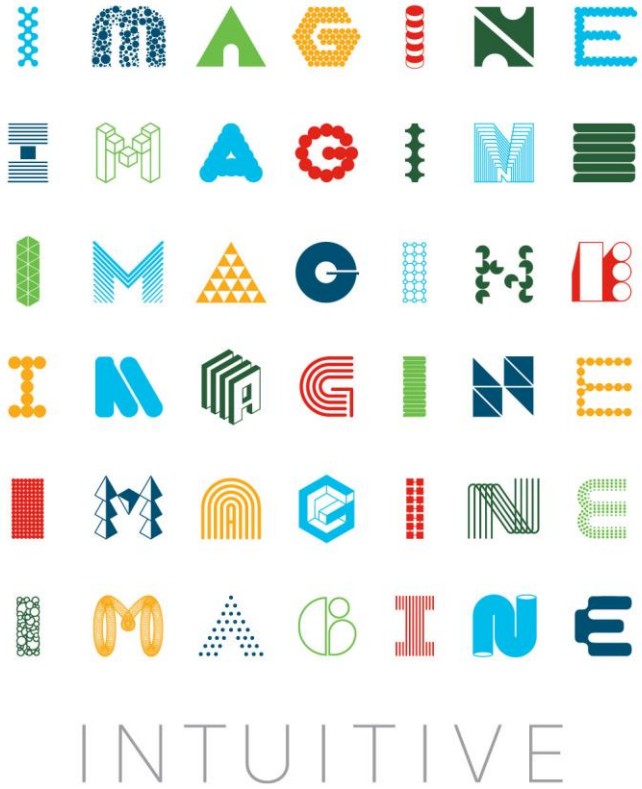
June 10-14, 2018 • Orlando, FL

#CLUS

ASR9000 Selected Topics and Troubleshooting

Nikolay Karpyshev, CCIE 43340
Engineering, SP Routing

BRKSPG-2904



Agenda

- cXR to eXR migration
- Turboboot
- Fabric interworking 5 vs 7
- Troubleshooting Packet Drops
- XR Embedded Packet Tracer
- Load-balancing
- PWHE Load Balancing and Troubleshooting
- Fragmentation
- HSRP/VRRP and Restrictive Timers
- QOS queuing/burst/buffering
- Software Management
- Dealing with punt fabric/automation
- News Digest

Cisco Webex Teams

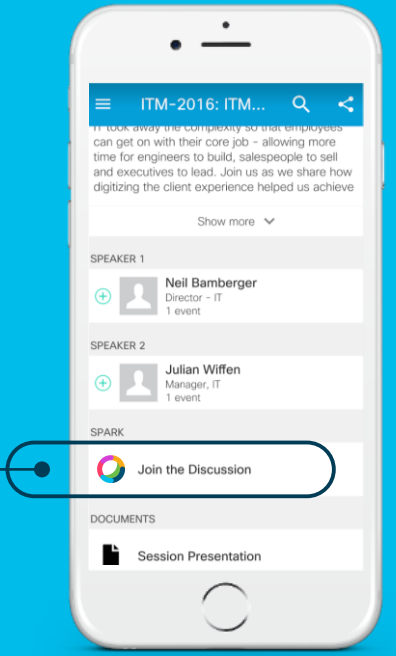
Questions?

Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

How

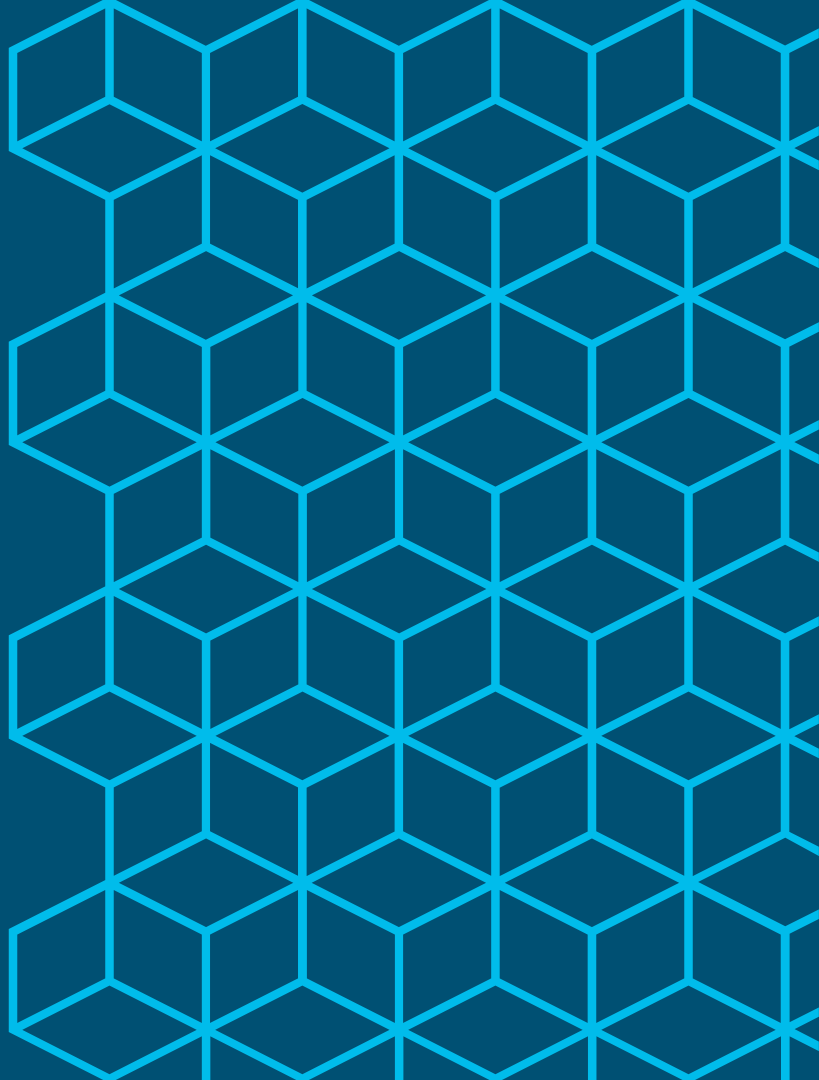
1. Find this session in the Cisco Events App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space

Webex Teams will be moderated by the speaker until June 18, 2018.



cs.co/ciscolivebot#BRKSPG-2904

cXR to eXR migration



What Is eXR and When To Use It?

- Evolved XR -- Linux 64bit kernel with VM's
- Starting with 6.1.x, eXR and cXR images are built for ASR9000.
- NCS5500 runs eXR only
- Typhoon and Tomahawk together can only run classic XR 32 bit.
- eXR requires:
  - Tomahawk or Lightspeed line cards
  - RSP880 / RP2

- More memory per process (BGP?)
- Linux tools (install using RPM)
- New hardware (e.g. Lightspeed line card for asr9000)
- Certain functionality (e.g. ISSU, Golden ISO, iPXE ZTP)

Migrating to eXR: need to knows

- Starting XR 6.3.2 images are signed with a “REL” key.
- New key introduced to prevent code overwrites (in particularly) eXR
  - which was exploited one time in SECCON.
- Since classic and eXR releases prior to 6.3.2 are signed differently and 6.3.2 onwards has the new key
  - security violation is detected failing the install.
- Bridge SMU **CSCvf01652** required to allow the migration on your *FROM* release to accept the different signatures

How To Migrate – CSM Server

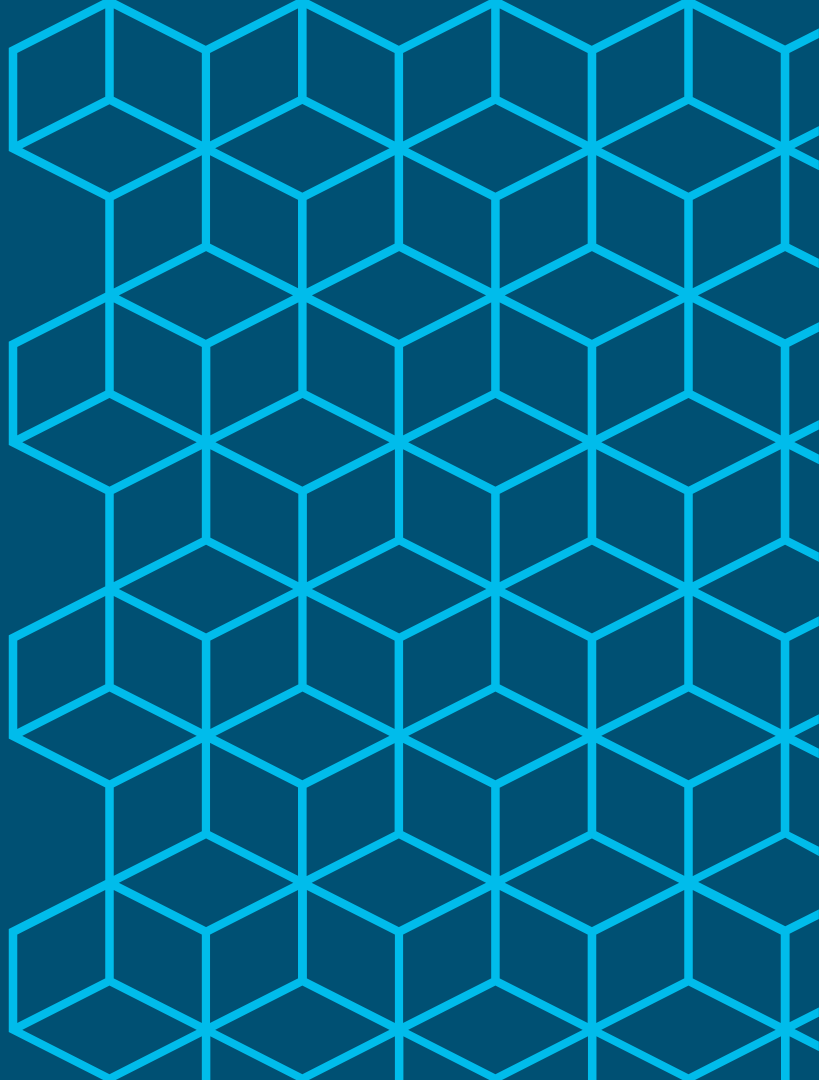
- Used by Cisco test teams for migration, upgrade and patch install
- Documentation included in the CSM Server installation
- CSM Server Demo:
  - https://www.youtube.com/watch?v=isxN08x-mr4

Great overview in only 4 minutes!

- CSM Server documentation:
  - https://supportforums.cisco.com/document/13154846/cisco-software-manager-csm-33-overview-documentation
- Migration documentation:
  - http://www.cisco.com/c/en/us/td/docs/routers/asr9000/migration/guide/b-migration-to-ios-xr-64-bit.html
- Video "ASR9K IOS XR 32 bit to 64 bit Migration using CSM Server":
  - https://youtu.be/RVgR0TdbpVw

NEW!!!

Turboboot



Turboboot

- Fresh install, booting the mini.vm from rommon
- Boot process recovery

Legacy

- Rommon Variables
- Define Network IP settings
- Set TFTP environment
- Set turboboot variable
- Boot mini.vm file

USB

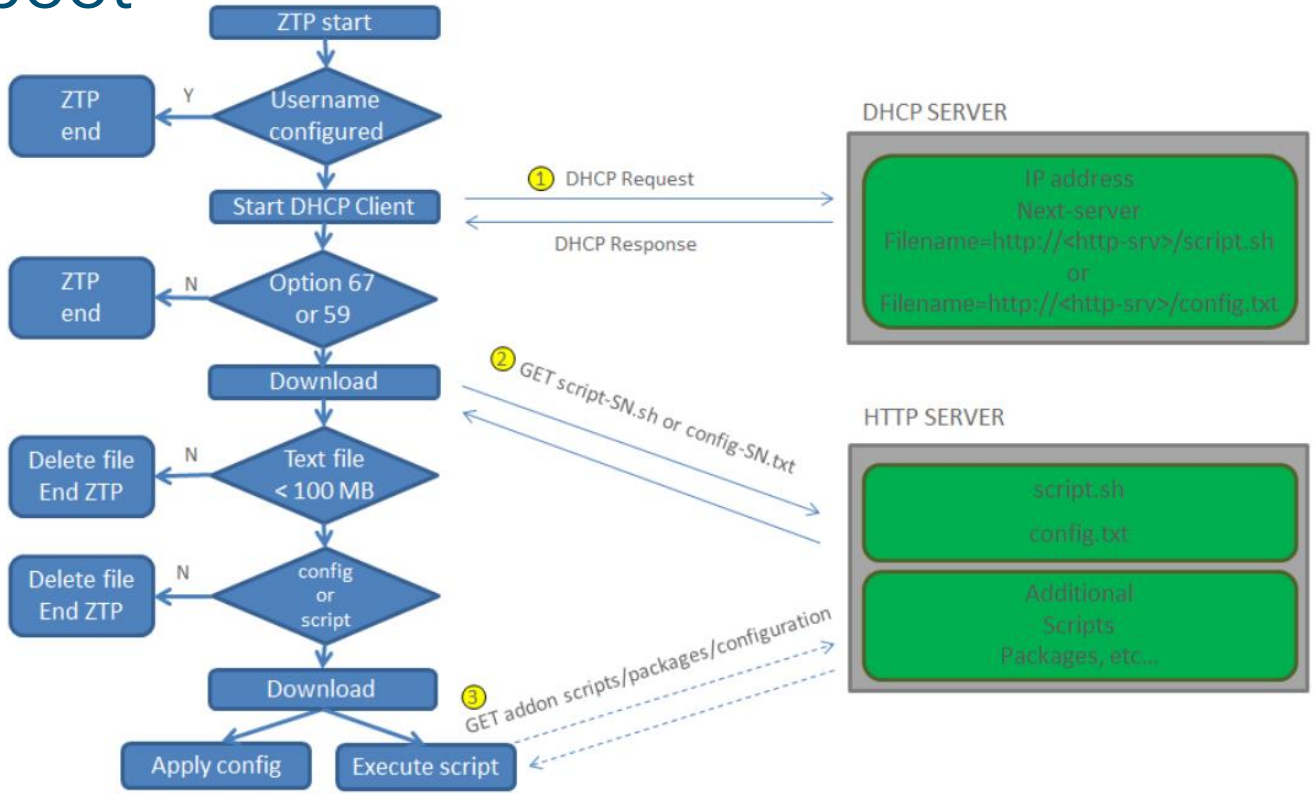
- From RSP-440 (and 9001 with rommon 2.03)
- Place file on USB
  - rommon> mediaboot usb:\release_mini.vm
- Later revisions of the rommon:
  - rommon> boot usb:/<file>
  - rommon> boot disk1:/

  use dev to see mapping

https://supportforums.cisco.com/t5/service-providers-documents/asr9000-xr-understanding-turboboot-and-initial-system-bring-up/ta-p/3146024

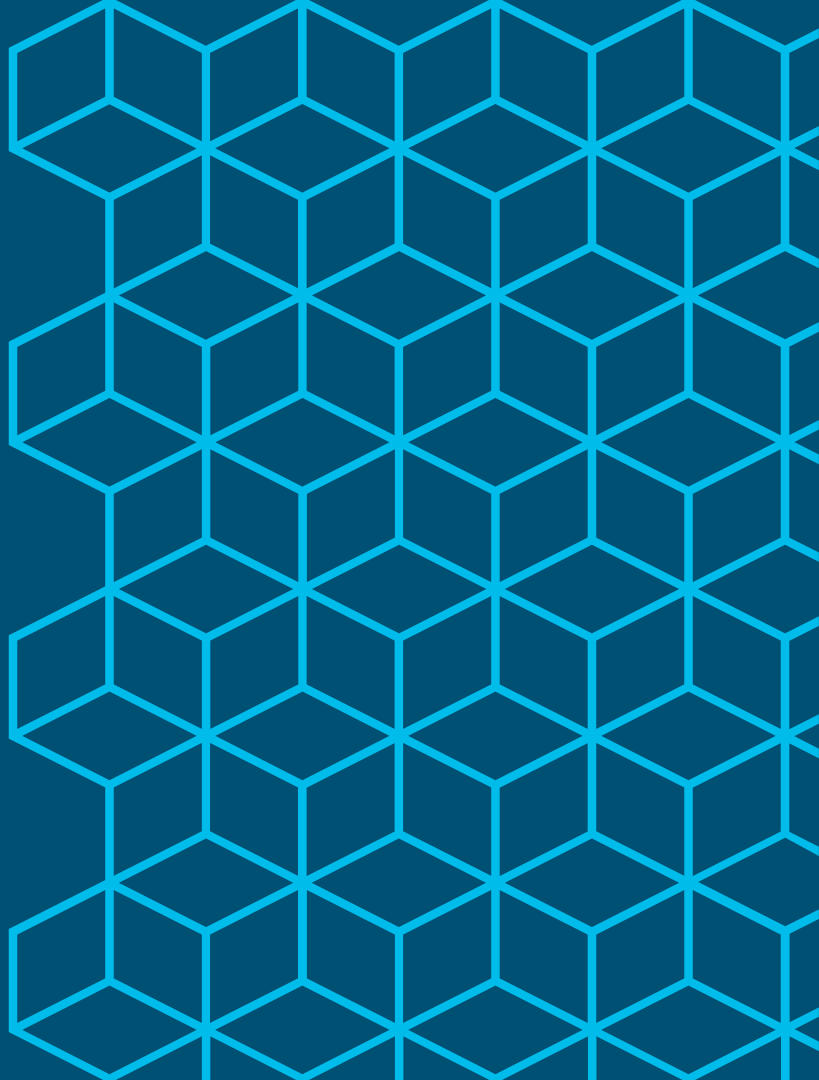
eXR iPXE ZTP boot

- ISO from TFTP/USB
- ZTP starts on first boot
- Can be forced from Admin Config



https://supportforums.cisco.com/t5/service-providers-documents/ipxe-boot-in-asr9k/ta-p/3363335
https://xrdocs.github.io/software-management/tutorials/2016-07-27-ipxe-deep-dive/
https://xrdocs.github.io/software-management/tutorials/2016-08-26-working-with-ztp/

ASR9k Fabric Interworking



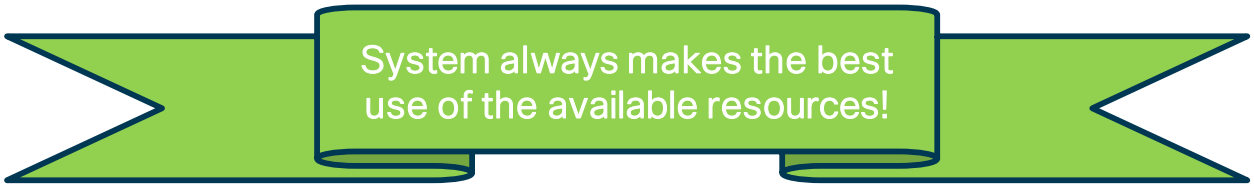
Benefits of ASR9k 7-Fabric Architecture

- Line rate performance for 12X100GE Skyhammer linecard
  - 5 fabrics (230 Gbps each, total 1150 Gbps) not enough for 12 ports
- Better fabric redundancy for 8X100GE linecards
  - Can lose two more fabrics before throughput impacted
- Line rate interoperability between 12X100GE and 8X100GE linecards in corner cases
  - When 12X100GE card sends traffic to 5-fabric 8X100GE cards, only the first five fabrics can be used thus can not send 1.2 Tbps linerate

Unicast 5/7-Fabric Interworking

- A9K linecards are 5 fabric (48x10, MOD200/400, 4x100, A9K-8x100)
- A99 linecards are 7 fabric (A99-8x100, 12x100)

- If Ingress and Egress Linecards are 7-Fabric cards:
  - Unicast traffic will use all 7 fabric cards
- If Ingress and/or Egress Linecard is 5-Fabric card:
  - Unicast traffic will use the first 5 fabric cards

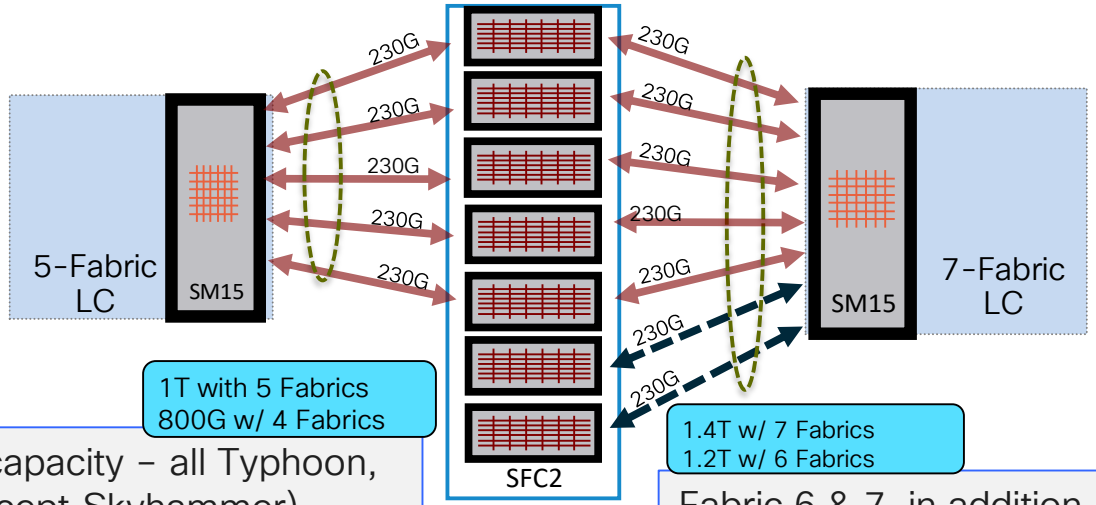


Fabric Mode CLI Commands in A99 Chassis Types

```
RP/0/RSP1/CPU0:ASR9K-2(admin-config)#fabric enable mode ?
  A99-highbandwidth  A99 High bandwidth cards only
  highbandwidth      High bandwidth cards only
```

- Default:
  - Max 1024 VQI per system
  - Multicast traffic uses the first 5 fabric cards
- Highbandwidth:
  - Max 2048 VQI per system (➔ only Tomahawk and RP2 allowed)
  - Multicast traffic uses the first 5 fabric cards
- A99-highbandwidth:
  - Max 2048 VQI per system
  - Multicast traffic uses all 7 fabric cards (➔ only A99 Tomahawk and RP2 allowed)

ASR99xx – LC and Fabric Operation

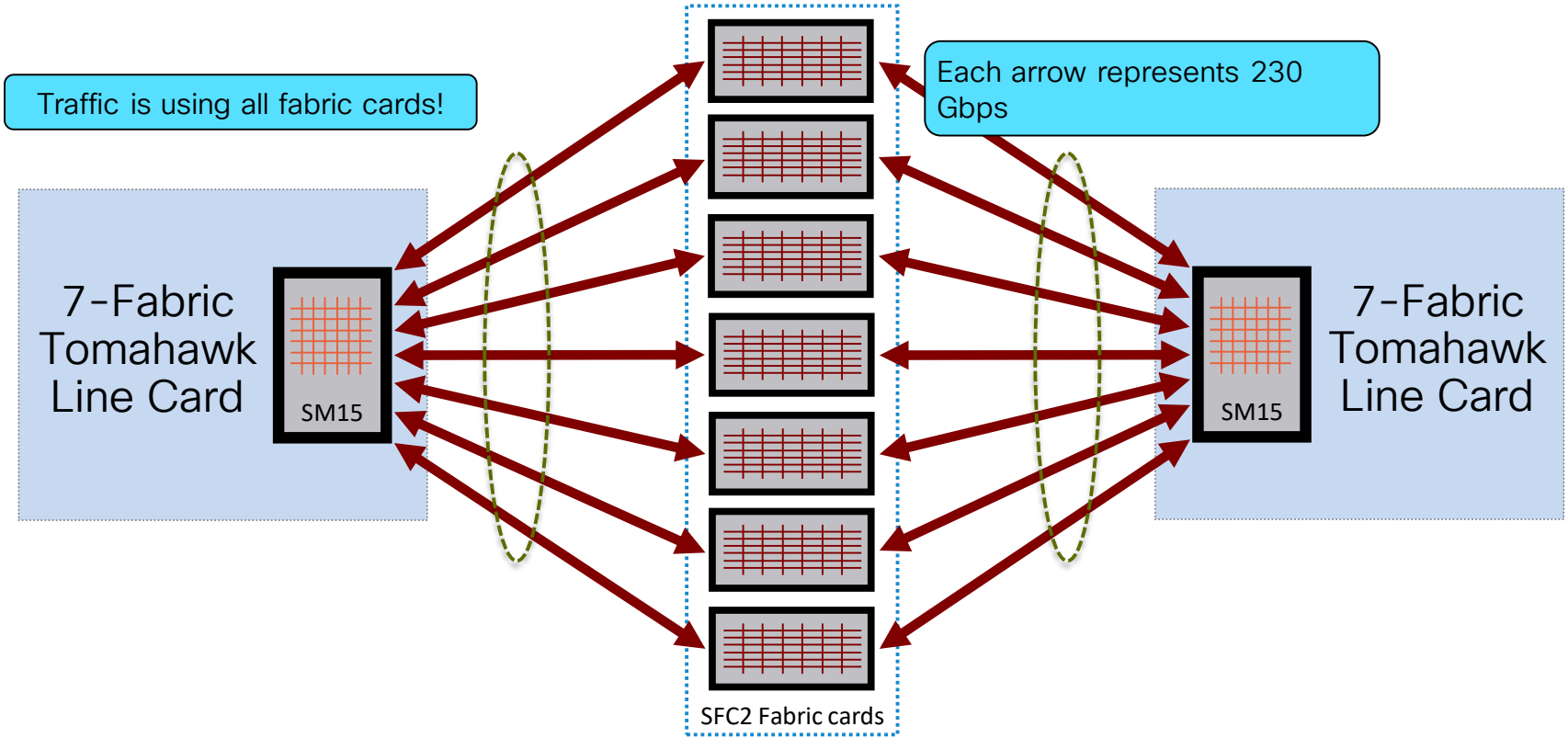


800G linerate /LC capacity – all Typhoon, Tomahawk LCs (except Skyhammer) interoperate at full throughput

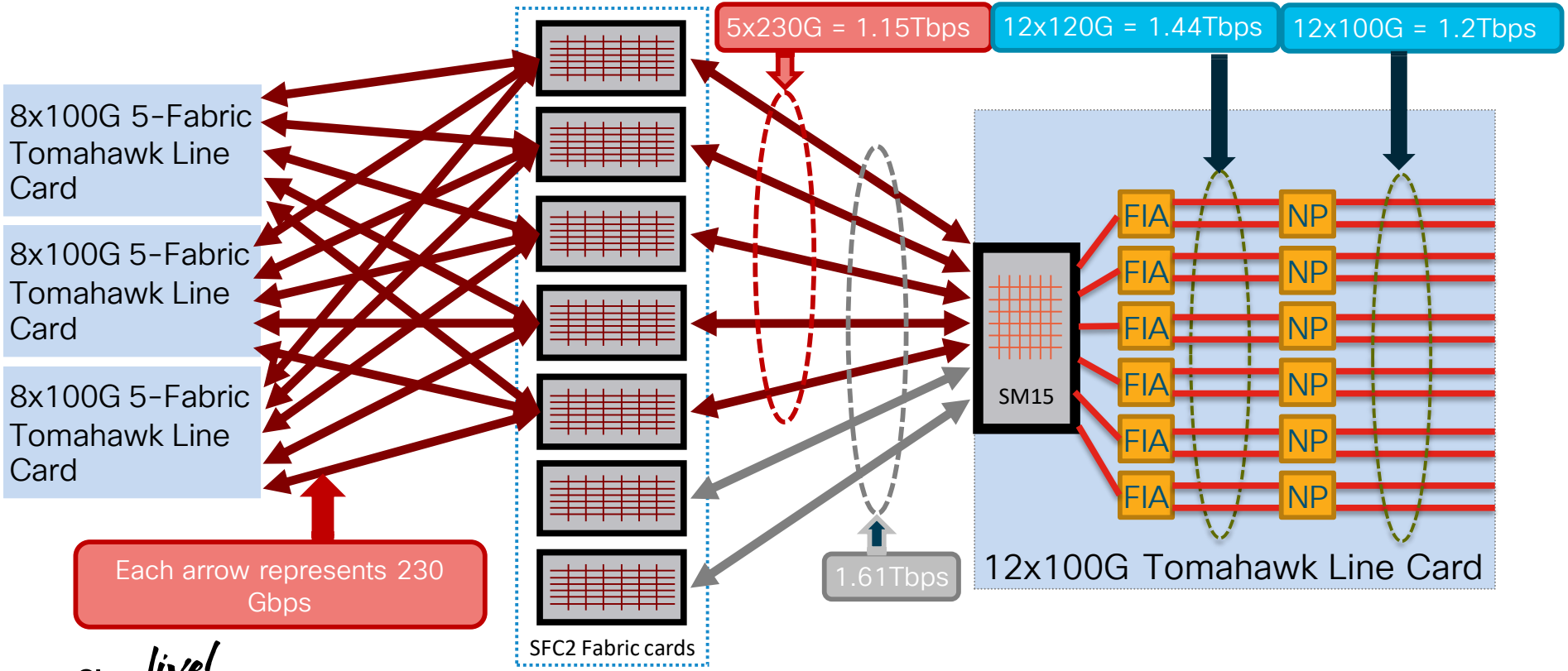
Fabric 6 & 7, in addition, provide per linecard capacity to get to 1.2T

7-fabric LCs uses 5 fabrics to interoperate with other 5-fab LCs

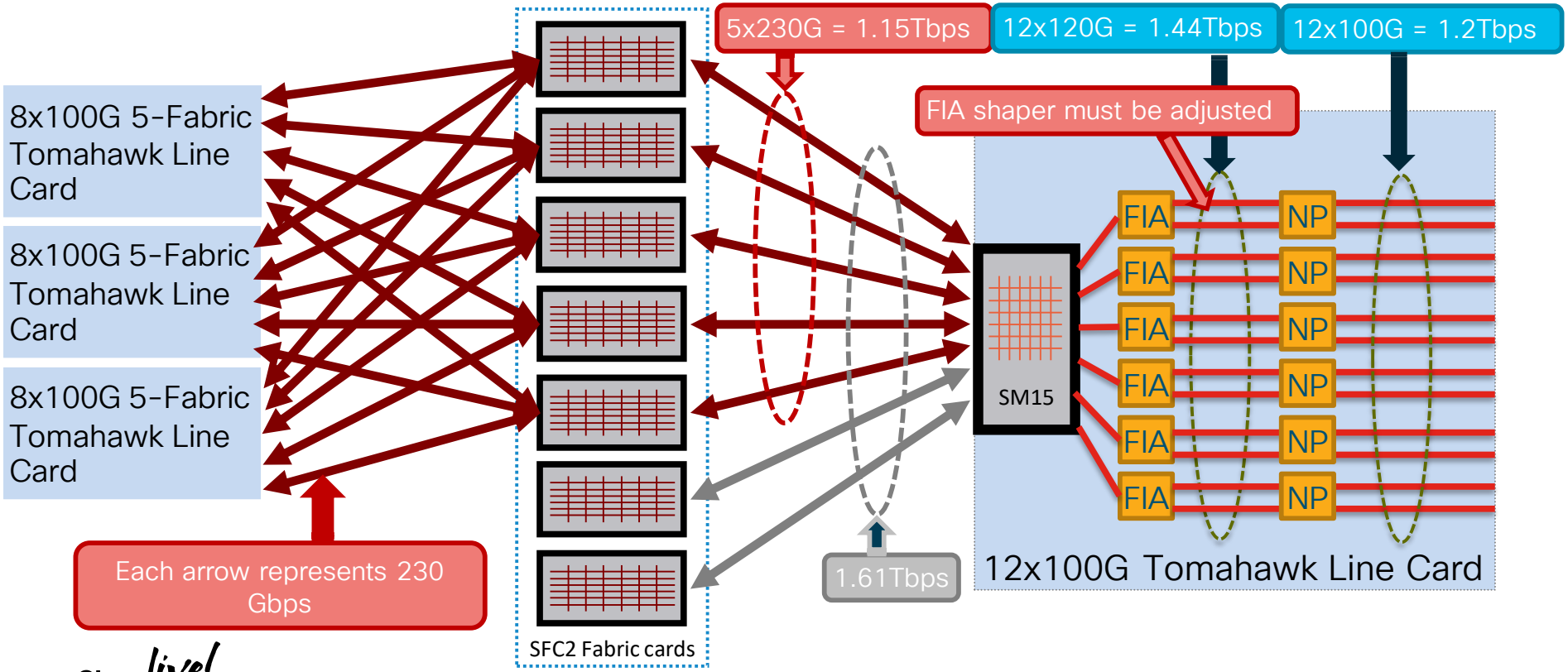
Fabric Interworking: 7-Fab LC to 7-Fab LC



Fabric Interworking: 5-Fab LC to 12x100G LC



Fabric Interworking: 5-Fab LC to 12x100G LC



Fabric Interworking: 5-Fab LC to 12x100G LC

- FIA shaper is applied by default on 12x100G line cards
- A99 chassis with 5 fabric cards or more:
  - 83Gbps per 100G port (total of 996 Gbps; fabric conn 5x230Gbps = 1.15Tbps Gbps)
- Any chassis with 4 fabric cards (asr9010, asr9006 with dual RSP880):
  - 71Gbps per 100G port (total of 852 Gbps; fabric connection 4x230Gbps = 920 Gbps)
- Syslog:
  - LC/0/0/CPU0:Dec 27 12:05:16.429 EST: pfm_node_lc[299]: %FABRIC-FIA-1-RATE_LIMITER_ON : Set|fialc[163907]|0x1072000|Insufficient fabric capacity for card types in use - FIA egress rate limiter applied
- Checking the shaper rate:
  - **show controllers fabric fia information location** <*location*>
  - **show controller fabric fia trace location** <*location*> | **include "shape_RL"**

Fabric Interworking: 5-Fab LC to 12x100G LC

```
RP/0/RSP1/CPU0:WEST-PE_ASR9K-2#admin show inventory | i Chassis
NAME: "chassis ASR-9010-AC", DESCR: "ASR 9010 8 Line Card Slot Chassis with V1 AC PEM"
RP/0/RSP1/CPU0:ASR9K-2#show controllers fabric fia information location 0/7/CPU0
 ********** FIA-0 **********
Category: bandwidth-0
                          BW if0          120
                          BW if1          23
 ********** FIA-1 **********
Category: bandwidth-1
                          BW if0          23
                          BW if1          120
 ********** FIA-2 **********
Category: bandwidth-2
                          BW if0          0
                          BW if1          0
 ********** FIA-3 **********
Category: bandwidth-3
                          BW if0          71
                          BW if1          71
```

One port in admin down

➔ Full BW allowed to the other port

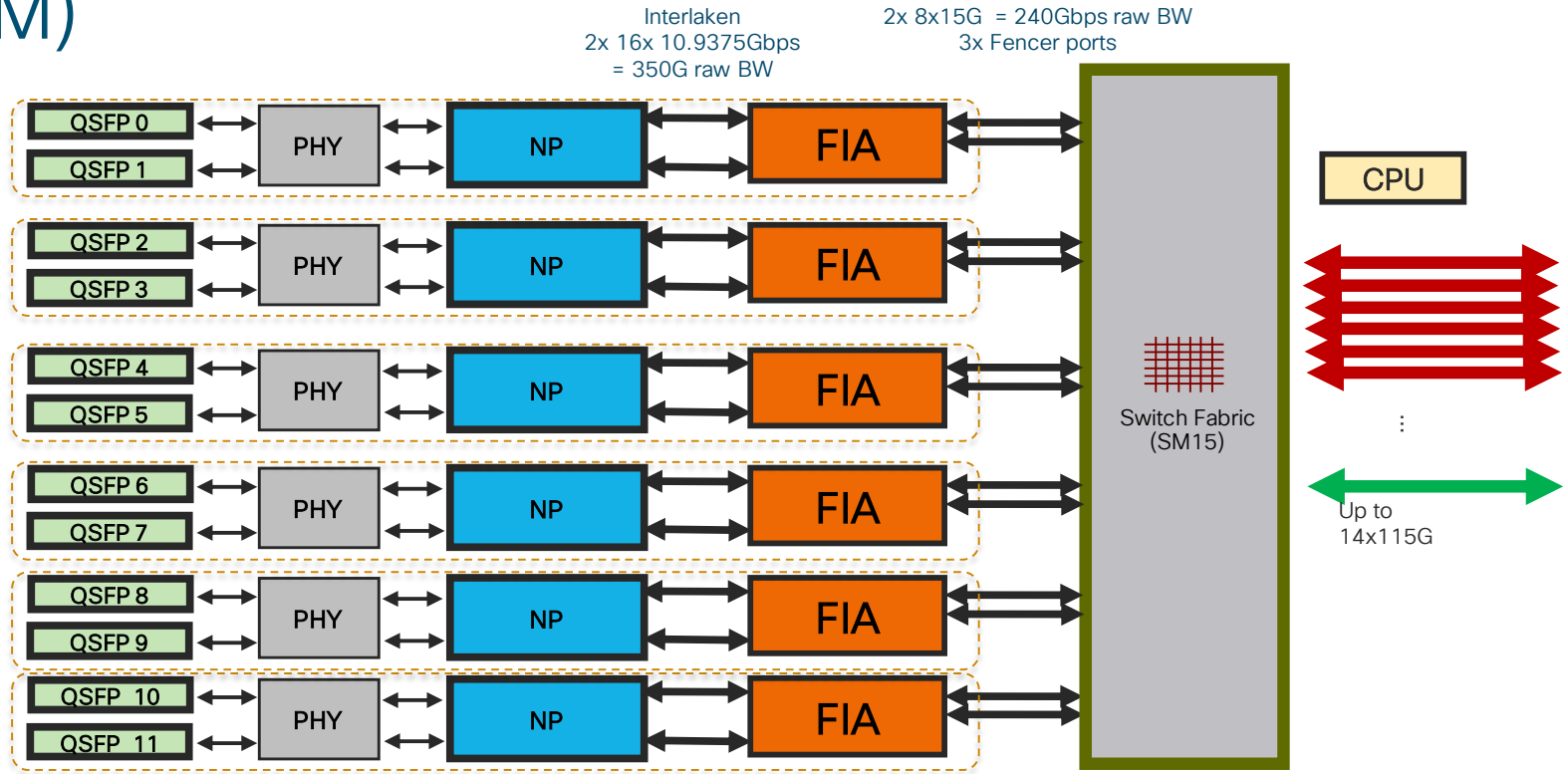
Slice powered down

BW split between two ports

Fabric Interworking: 5-Fab LC to 12x100G LC

- FIA policer can be disabled on A99 chassis
- Configuration command (note: not an admin command):
  - hw-module fia-intf-policer disable
- When can I use it?
  - When most traffic is between 12x100G line cards
  - E.g.: Only one line card in chassis is a 5-fabric card
  - E.g.: All 5-fabric cards in the chassis are MOD200

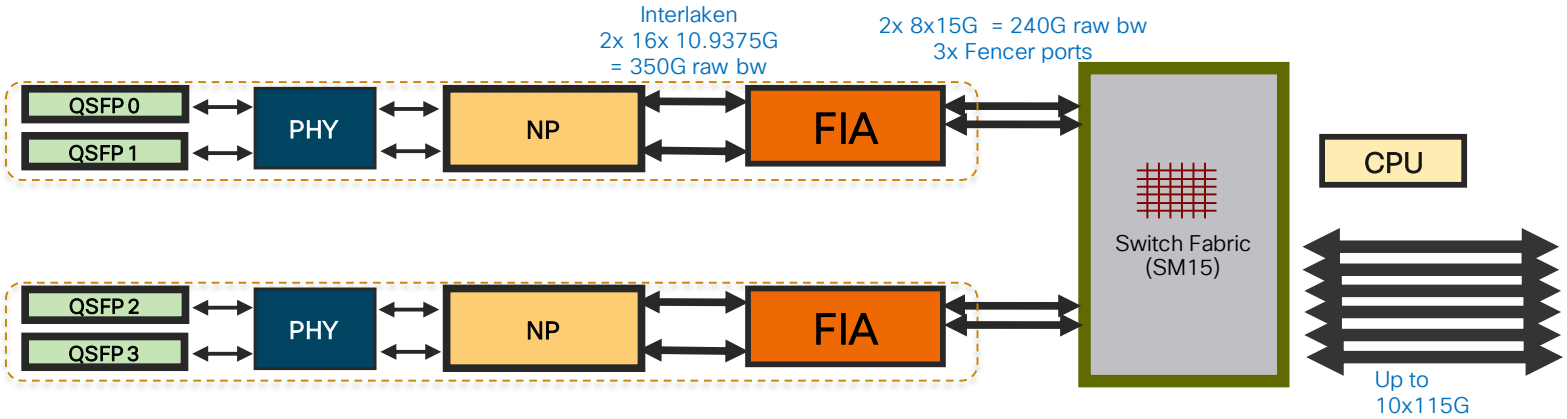
12-port 100GE linecard architecture (No External TCAM)



- No external TCAM on this card. Only 5Mb internal TCAM
- Due to limited TCAM only L3 Transport/LSR features supported

4x100GE LAN linecard architecture (No External TCAM)

Mini-skyhammer



Cost Optimized Mini Skyhammer A9K-4X100GE

FCS: Dec' 17
Rel 6.3.2

HW Specs	• Tomahawk based, 5-fabric linecard with: 2x NPU and 5Mbit internal TCAM • PID: A9K-4x100GE - Supported on all ASR 90xx and 99xx systems
Positioning	• TR scale limits (reduced scale for TCAM features)
Features	• Parity with Tomahawk TR linecards at FCS • Features not supported: VidMon, Cluster, LISP, NSH, BNG, Satellite, MPLS-TP
Scale	• Limited scale for TCAM dependent features; Rest of the scale on-par with other Tomahawk -TR cards
SW support	• cXR: 6.2.3, 6.3.2, 6.4.1 and onwards; SMU option available on demand for 6.1.4, 6.3.1 • eXR: 6.5.1. Tentative for 6.4.2

Not same as A9K-4x100GE-TR/SE Tomahawk card

Bandwidth/RSP on ASR9K chassis

Chassis	RSP880-LT	RSP880	A99-RSP
ASR 9910	230Gbps per RSP/SFC	NA	230Gbps per RSP/SFC
ASR 9906	230Gbps per RSP/SFC	NA	230Gbps per RSP/SFC
ASR 9904	440Gbps/RSP	440Gbps/RSP	NA
ASR 9010	440Gbps/RSP	440Gbps/RSP	NA
ASR 9006	440Gbps/RSP	440Gbps/RSP	NA

Cost Optimized RSP880-LT

Scale & Perf	• 2X more Bandwidth than RSP440 – 880Gbps • 2X more RIB scale • Similar memory and scale to RSP880
Operational Efficiency	• RSP880-LT will be supported ASR 9006, ASR 9010, ASR 9904, ASR 9910, ASR 9906 chassis
Power	• ~30% lower Power consumption than RSP880

Cost Optimized RSP880-LT

Scale & Perf

- 2X more Bandwidth than RSP440 – 880Gbps
- 2X more RIB scale
- Similar memory and scale to RSP880

Operational Efficiency

- R
  9

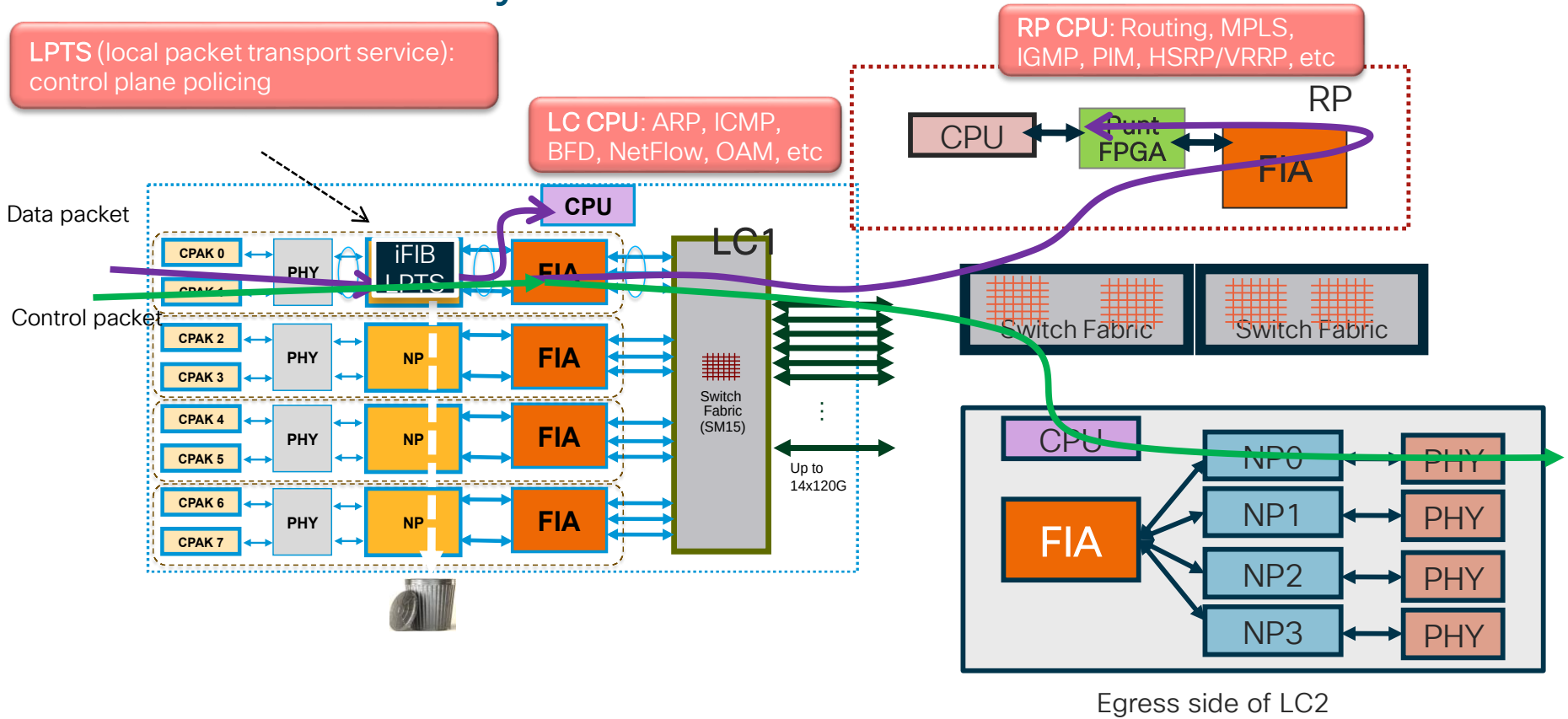
Power

- ~

RSP880	RSP880-LT
8 Cores, 1.9Ghz	4 Cores, 2.4Ghz
4 SFP+ external ports	No external SFP+ ports
2x32GB SSD	2x128GB SSD
eXR 6.1.2	eXR 6.4.1
3rd Party Application	None

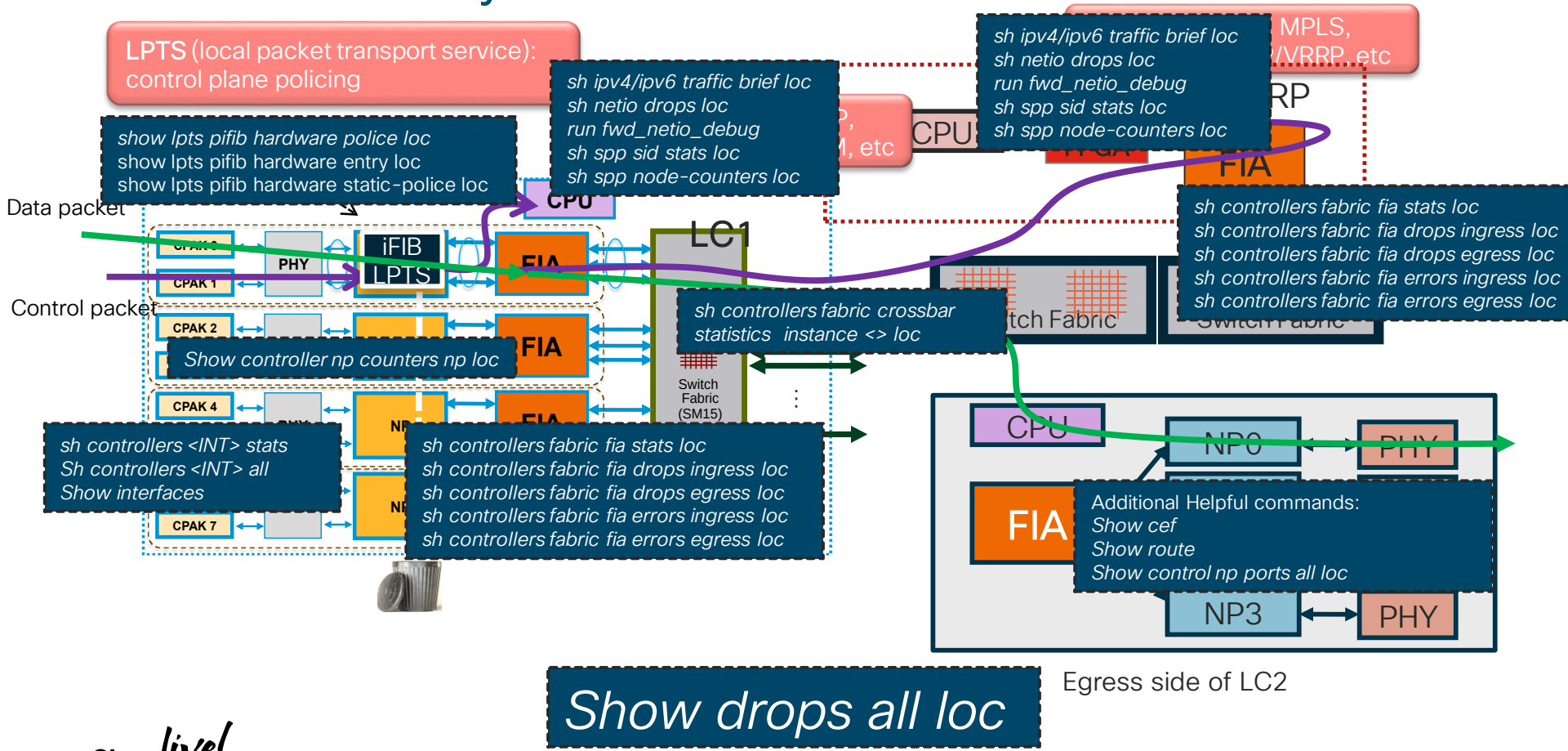
Troubleshooting Packet Drops

ASR9000 Fully Distributed Control Plane



Egress side of LC2

ASR9000 Fully Distributed Control Plane



Input Drops Troubleshooting

GigabitEthernet0/0/1/6.1 is up, line protocol is up
<..output omitted..>
307793 packets input, 313561308 bytes, 227987 total input drops

Troubleshooting this?

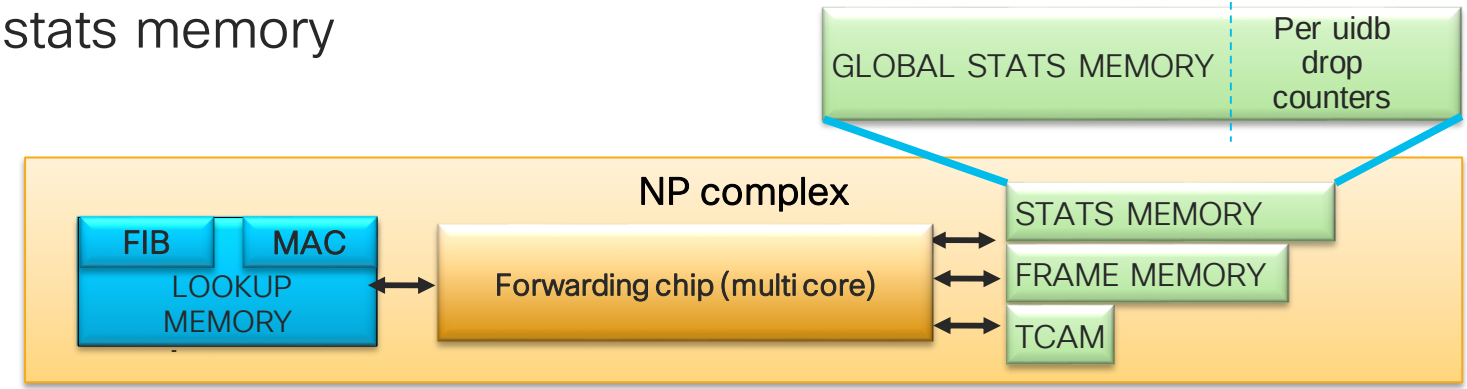
Piece of cake starting with IOS XR 5.3.3!!!

Packet drops troubleshooting tools in IOS XR 5.3.3 and later:

- Per uidb drop counter monitoring: "monitor np interface"
- Pervasive dropped packet capture: "show controller np capture"
- Available on both Tomahawk and Typhoon
- "monitor np counter" still available for all other counters

Monitor NP Interface

- Part of the stats memory carved out for per-uidb drop counters
- UIDB == µIDB == **M**icro **I**nterface **D**escriptor **B**lock
  - NP's view of an interface
- Only one uidb at the time per LC can be monitored
- Drop counters that are updated for selected uidb are not updated in the global stats memory



Monitor NP Interface

```
RP/0/RSP0/CPU0:our9001#monitor np interface g0/0/1/6.1 count 2 time 1 location 0/0/CPU0
Monitor NP counters of GigabitEthernet0_0_1_6.1 for 1 sec

<..output omitted..>
                          ****  Sun Jan 31 22:14:32 2016 ****

Monitor 2 non-zero NP1 counters:  GigabitEthernet0_0_1_6.1
Offset  Counter                                FrameValue   Rate (pps)
--------------------------------------------------------------------------------
 262 RSV_DROP_MPLS_LEAF_NO_MATCH_MONITOR                  101        49
1307 PARSE_DROP_IPV4_CHECKSUM_ERROR_MONITOR                101        50

(Count 2 of 2)
RP/0/RSP0/CPU0:our9001#
```

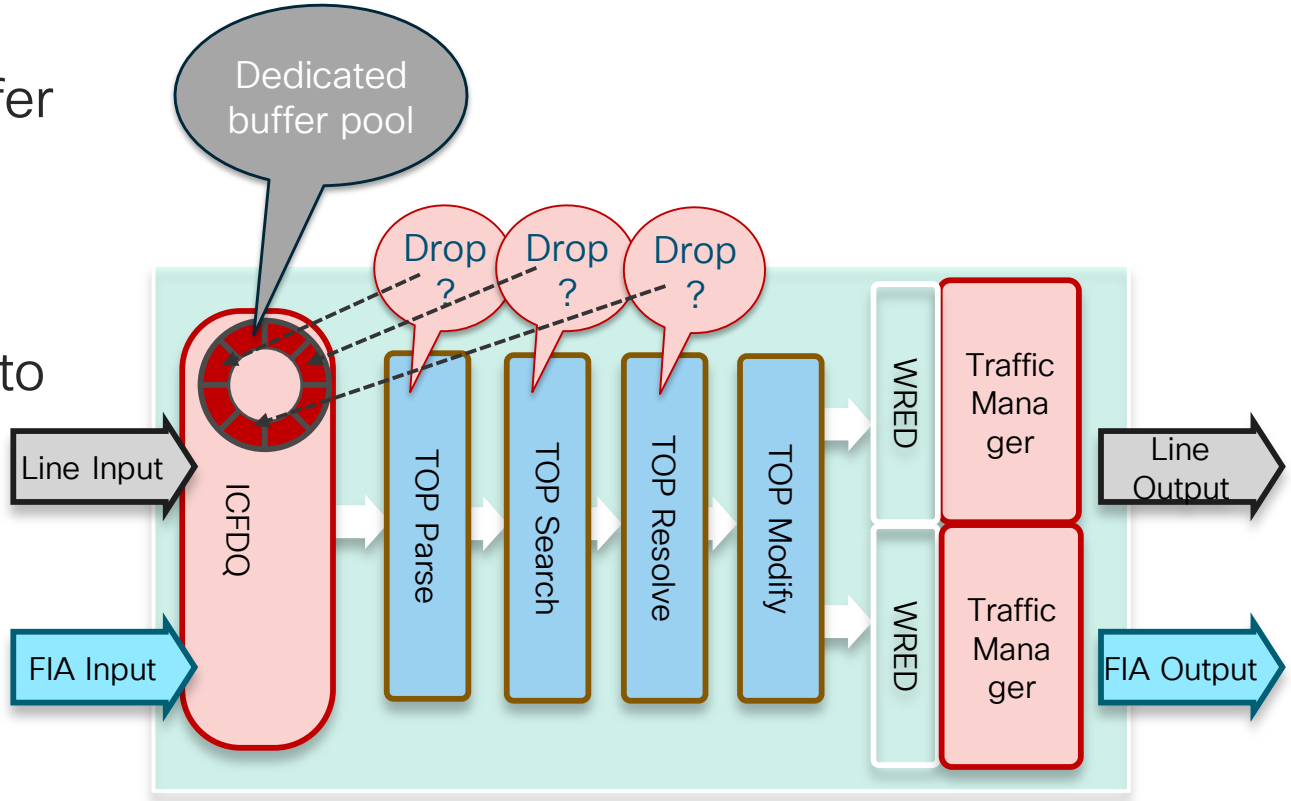
Monitor NP Interface

```
RP/0/RSP0/CPU0:our9001#monitor np interface g0/0/1/6.1 count 2 time 1 location 0/0/CPU0
Monitor NP counters of GigabitEthernet0_0_1_6.1 for 2 sec

<..output omitted..>


Monitor 2 non-zero N
Offset  Counter
------------------
 262 RSV_DROP_MP
1307 PARSE_DROP_

(Count 2 of 2)
RP/0/RSP0/CPU0:ou
```

- Counters reported with '_MONITOR' appendix
  - These counters are not added to global NP counters
- By default runs one capture during 5 seconds (configurable count and time)
- One session at the time per LC
- Supports physical and BE (sub)interfaces
  - Physical (sub)int: monitoring runs on NP that hosts the interface
  - BE(sub)int: monitoring runs on all NP that host the members
- Applicable only to ucode stages where uidb is known
  - Works perfectly for input drops troubleshooting and some output drops

Pervasive Capture of Dropped Packets

- Dedicated pool of buffer pointers
- Instead of dropping – first RFD buffer of a packet is enqueued into a dedicated pool



Show Controllers NP Capture

```
RP/0/RSP0/CPU0:our9001#sh controllers np capture np1 location 0/0/CPU0

NP1 capture buffer has seen 426268 packets - displaying 32

Sun Jan 31 22:55:13.935 : RSV_DROP_MPLS_LEAF_NO_MATCH
 From                              : 1222 byte packet on NP1
0000: 84 78 ac 78 ca 3e 30 f7 0d f8 af 81 81 00 03 85
0010: 88 47 05 dc 11 ff 45 00 00 64 01 ae 00 00 ff 01
0020: 62 c3 ac 12 00 02 ac 10 ff 02 00 00 02 3a 00 0a
<..output omitted..>
RP/0/RSP0/CPU0:our9001#sh controllers np capture np1 help location 0/0/CPU0

 NP1 Status            Capture Counter Name
 --------------------+-----------------------------
 Capturing             PARSE_UNKNOWN_DIR_DROP
 Capturing             PARSE_UNKNOWN_DIR_1
<...output omitted..>
RP/0/RSP0/CPU0:our9001#sh controllers np capture np1 filter RSV_DROP_MPLS_LEAF_NO_MATCH disable location 0/0/CPU0

 Disable NP1 packet capture for: RSV_DROP_MPLS_LEAF_NO_MATCH
```

Show Controllers NP Capture

```
RP/0/RSP0/CPU0:our9001#sh controllers np capture np1 location 0/0/CPU0

NP1 capture buffer has seen 426268 packets - displaying 32

Sun Jan 31 22:55:13.935 : RSV_DROP_MPLS_LEAF_NO_MATCH
 From GigabitEthernet0_0_1_6: 1222 byte packet on NP1
000
001
002
<..o
RP/

 NP
 ---
 Cap
 Cap
<...c
RP/

 Dis
```

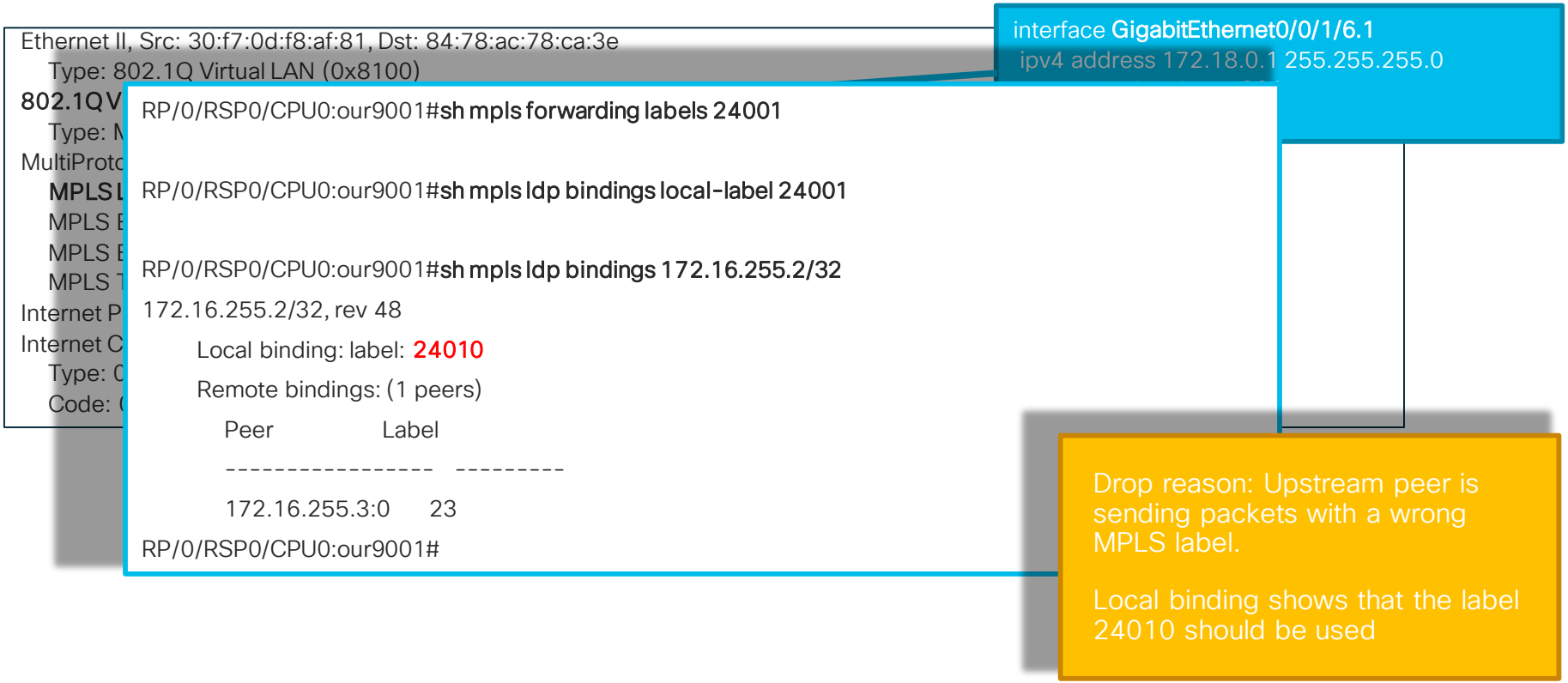
- Circular buffer captures the recently dropped packets
  - Tomahawk: 128 buffers
  - Typhoon: 32 buffers
- Enabled by default – no configuration required!
- Works at port-level
  - L2 encapsulation is included in the dump → you can figure out the sub-interface if you decode the packet dump
- In case of packets spanning more than one buffer, only the first buffer is captured
- Filtering is supported – you can select which drop reasons not to capture
  - Run the 'help' option to see the eligible counters and their status

Show Controllers NP Capture – Next Steps

```
interface GigabitEthernet0/0/1/6.1
 ipv4 address 172.18.0.1 255.255.255.0
 encapsulation dot1q 901
!
```

```
Ethernet II, Src: 30:f7:0d:f8:af:81, Dst: 84:78:ac:78:ca:3e
    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 901
    Type: MPLS label switched packet (0x8847)
MultiProtocol Label Switching Header, Label: 24001, Exp: 0, S: 1, TTL: 255
    MPLS Label: 24001
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 255
Internet Protocol, Src: 172.18.0.2 (172.18.0.2), Dst: 172.16.255.2 (172.16.255.2)
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0 ()
```

Troubleshooting Input Drops – Next Steps



Monitor NP Counter

- Available since 4.3.x
- ACL with capture can be used to filter packets you want to match
  - IPv4/v6 ACL can still be used for matching if the MPLS stack is one level deep
- All captured packets are dropped!!!
- NP reset is required upon capture completion
  - ~50ms traffic outage on Typhoon, ~150 on Tomahawk

Monitor NP Counter

```
RP/0/RSP0/CPU0:our9001#monitor np counter ACL_CAPTURE_NO_SPAN.1 np1 location 0/0/CPU0

Warning: Every packet captured will be dropped! If you use the 'count'
      option to capture multiple protocol packets, this could disrupt
      protocol sessions (eg, OSPF session flap). So if capturing protocol
      packets, capture only 1 at a time.

      Additional packets might be dropped in the background during the
      capture; up to 1 second in the worst case scenario. In most cases
      only the captured packets are dropped.

Warning: A mandatory NP reset will be done after monitor to clean up.
      This will cause ~50ms traffic outage. Links will stay Up.
 Proceed y/n [y] >
 Monitor ACL_CAPTURE_NO_SPAN.1

<...output omitted..>

Cleanup: Confirm NP reset now (~50n
 Ready? [enter] >
RP/0/RSP0/CPU0:our9001#
```

Always allow NP to be reset after the capture!!!

```
ipv4 access-list CL16
 10 permit icmp host 172.18.0.2 host 172.16.255.2 capture
!
interface GigabitEthernet0/0/1/6.1
 ipv4 access-group CL17 ingress

RP/0/RSP0/CPU0:our9001#sh controllers np counters np1 location 0/0/CPU0 | i SPAN
483  ACL_CAPTURE_NO_SPAN                              14859          3
```

“Show drops all” enhancement

- Supported starting with 5.3.0
- Uses a ‘grammar’ file to combine outputs of other show commands
  - Grammar file can be modified to suite particular troubleshooting tasks
  - System will look for it at two locations:
    1. disk0a:/usr/packet_drops.list
    2. /pkg/etc/packet_drops.list (default)
- “show drops all commands” shows called constituent commands

```
RP/0/RP0/CPU0:ios#sh drops all commands
Module                CLI
[arp]                 show arp traffic
[cef]                 show cef drops
[fabric]              show controllers fabric fia drops egress
...
[spp]                 show spp client detail
[spp]                 show spp ioctrl
```

- Use the ongoing keyword to show delta

“Show drops all” sample output (2)

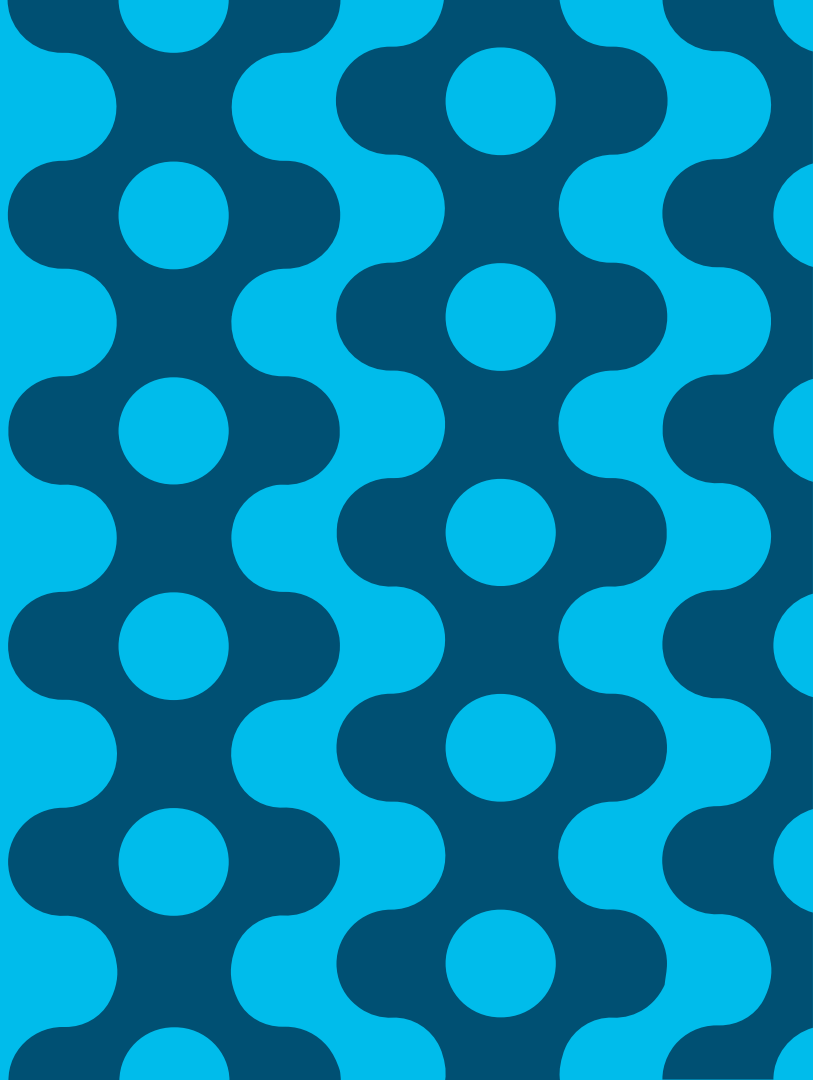
```
RP/0/RSP0/CPU0:ASR9000# show drops all ongoing location 0/0/CPU0
 ====================================
Checking for ongoing drops on 0/0/CPU0
====================================

show arp traffic:
[arp:ARP] IP Packet drop count for node 0/0/CPU0: +52

show cef drops:
 [cef:0/0/CPU0] No route drops packets : +1352

show netio drops:
[netio:Interface: FINT0/0/CPU0] /pkg/lib/libl2_adj_fint_netio.dll: +1351

show controller np counters:
 [np:NP0] PARSE_ING_DISCARD: +10824
[np:NP0] PARSE_ING_L3_CFM_DROP: +5412
[np:NP0] UNKNOWN_L2_ON_L3_DISCARD: +5412
[np:NP0] RSV_DROP_IN_L3_NOT_MYMAC: +5413
 [np:NP0] ARP_EXCD: +108238
[np:NP0] MDF_PUNT_POLICE_DROP: +108237
```

Troubleshooting NP Forwarding

1. Identify interface in question.
2. Identify the mapping from interface to NPU.
3. Examine NP counters.
4. Look for rate counters that match lost traffic rate.
   - If none of the counters match the expect traffic, check for drops at interface controller
5. Lookup the counter description.
6. If required capture the packet hitting the counter (Typhoon/Tomahawk only).
   - If troubleshooting drops use the new tools “monitor np interface” and “show controller np capture”.
7. If packets are forwarded to the fabric, run fabric troubleshooting steps.
8. Identify egress NP and repeat steps 3 to 6.

http://docwiki.cisco.com/wiki/Finding_Packet_Drops_-_ASR9K

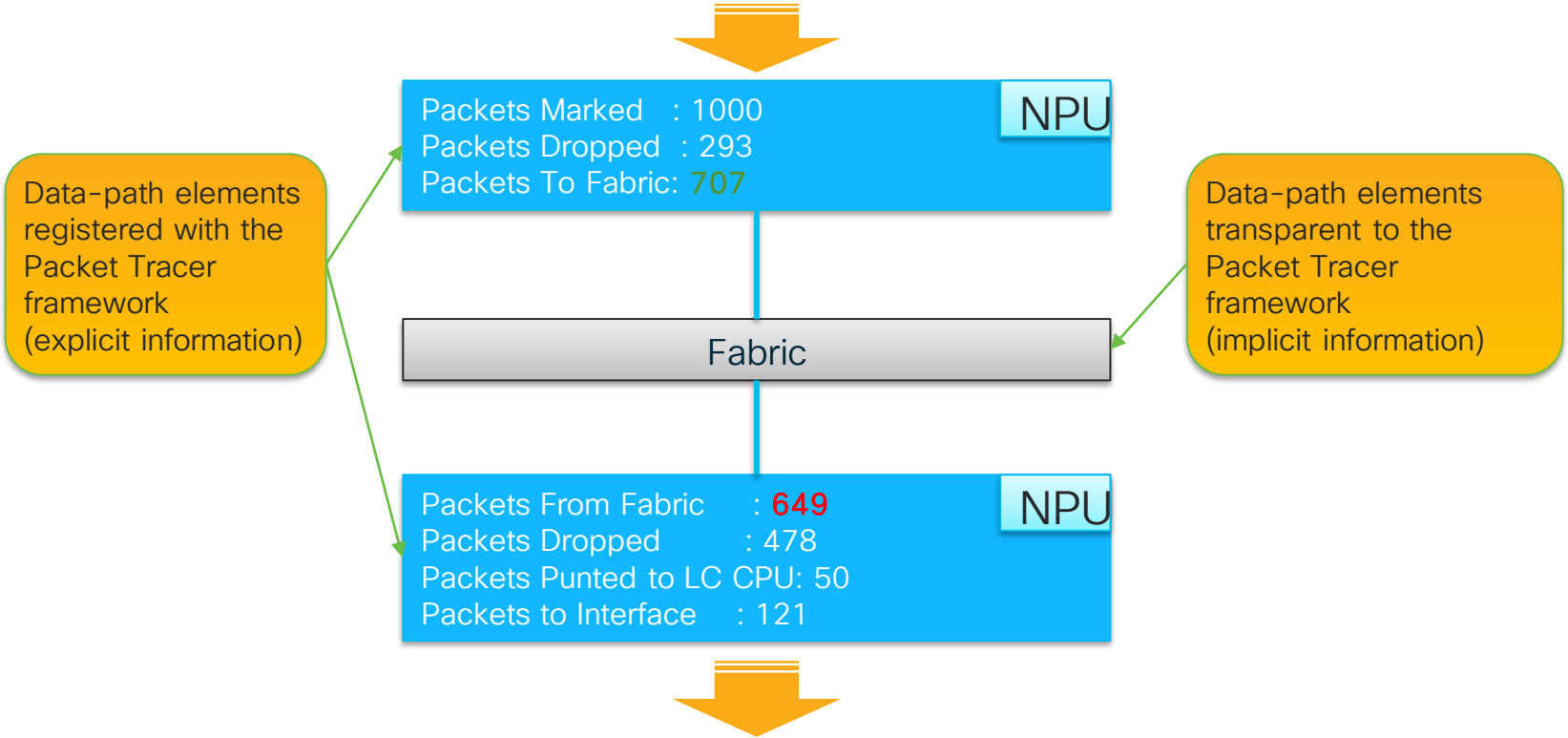
XR Embedded Packet Tracer



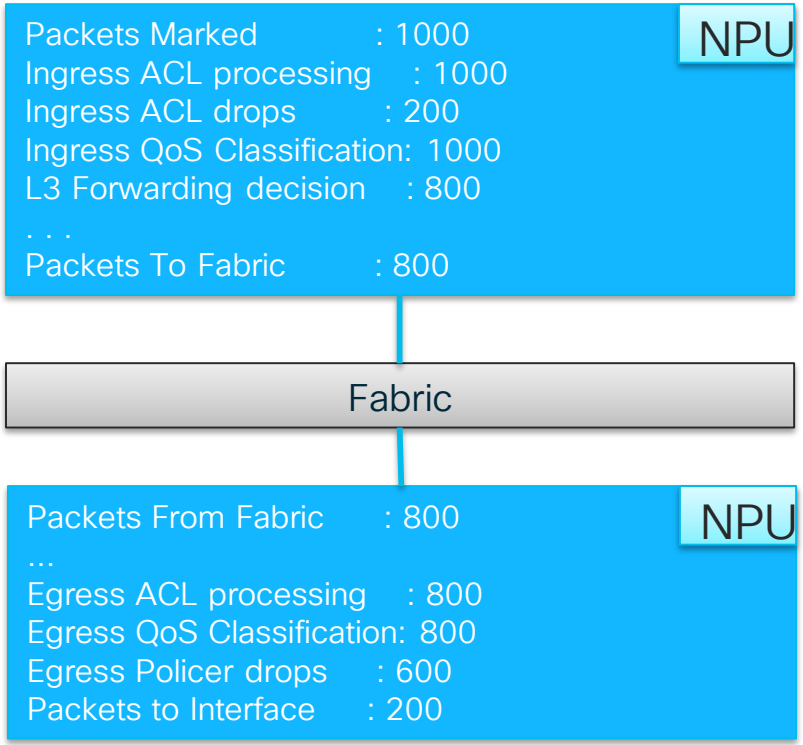
XR Embedded Packet Tracer

- XR Embedded Packet Tracer (EMT) empowers users to:
  - Follow a given packet flow through the router
  - Perform in-depth triaging of packet forwarding issues in data, punt and inject path
- Common Packet Tracer framework used by all XR platforms
  - Common slow path tracing capabilities across all platforms
  - Data-path tracing capabilities differ between platforms
- Key functionalities:
  - Trace packets through the router.
  - User defined conditions to match packets of interest.
  - Network Protocol agnostic (supports offset/value/mask as condition)
  - Works from user (cisco-support privilege) mode (no debug nor configuration required)
  - Provide insight into the features executed on the packet, with timestamps and feature processing result (phase II)

Simple Trace Through Data-path

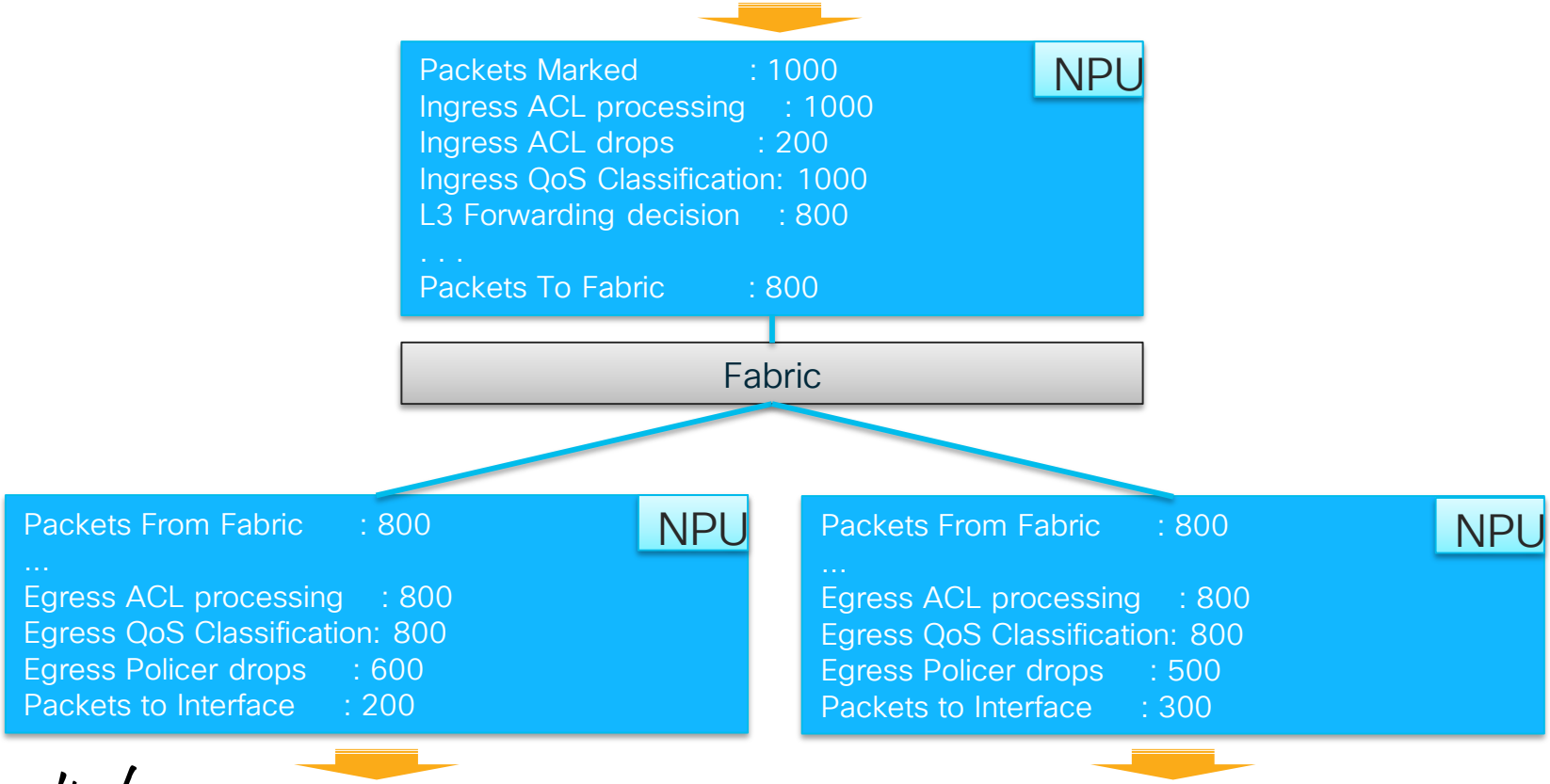


Packet Tracer Granularity

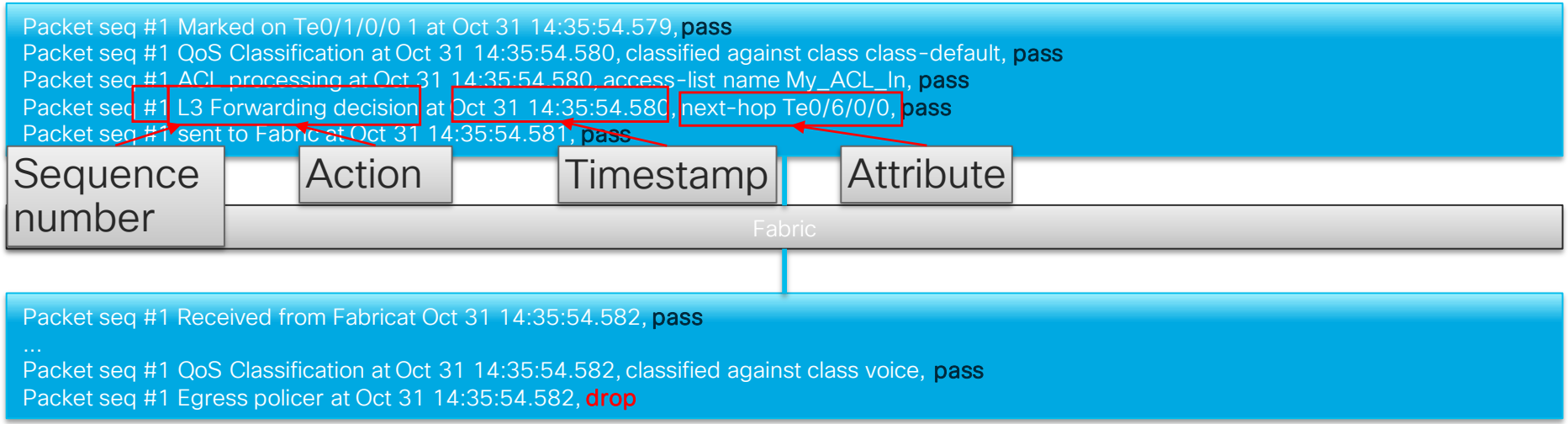


- Packet Tracer Granularity dictated by the number of registered counters

Multicast? – It just works! ☺



Detailed Trace



- Every marked packet is assigned a unique sequence number
- In detailed mode only one packet marking element allowed in the system
- **Not all platforms will support this mode!** (fabric header must carry the seq#)

Set The Packet Trace Condition

- Syntax:

```
packet-trace condition <keyword> <string>
```

- Supported filter types:
  - Raw (Offset/Value/Mask):

```
packet-trace condition interface Te0/2/0/0
packet-trace condition offset 12 value 0x8847 mask 0xFFFF
packet-trace condition offset 14 value 0x04231000 mask 0xFFFFF000
packet-trace condition offset 18 value 0x03E81000 mask 0xFFFFF000
```

Logical AND

  - Canonical:

```
packet-trace condition interface Te0/2/0/0
packet-trace condition ipv4 offset 22
packet-trace condition ipv4 source 10.10.10.10/32
packet-trace condition ipv4 destination 10.20.20.0 mask 255.255.252.0
packet-trace condition tcp port source 80
```

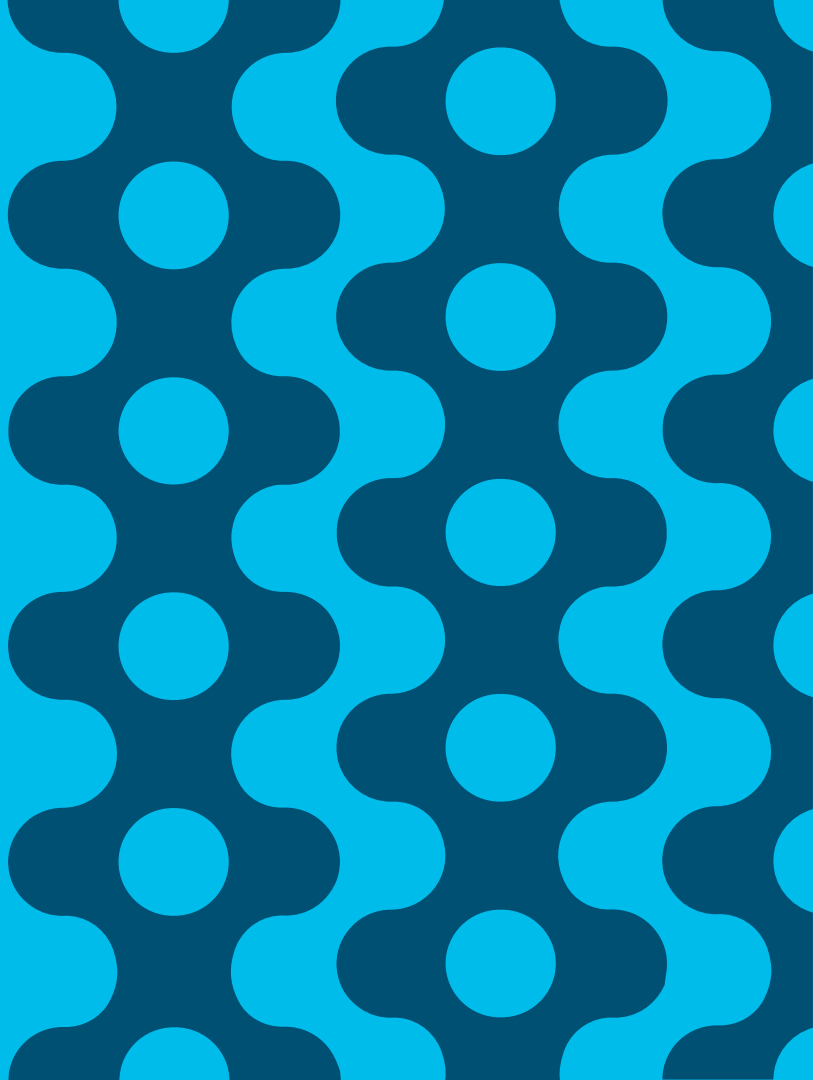
Logical AND

XR Embedded Packet Tracer

- What can you expect at FCS
- Platforms: asr9k
- Capabilities: data-path tracing only, basic counters

How useful do you find this feature?
Leave us your feedback via Cisco Webex Teams or supportforums

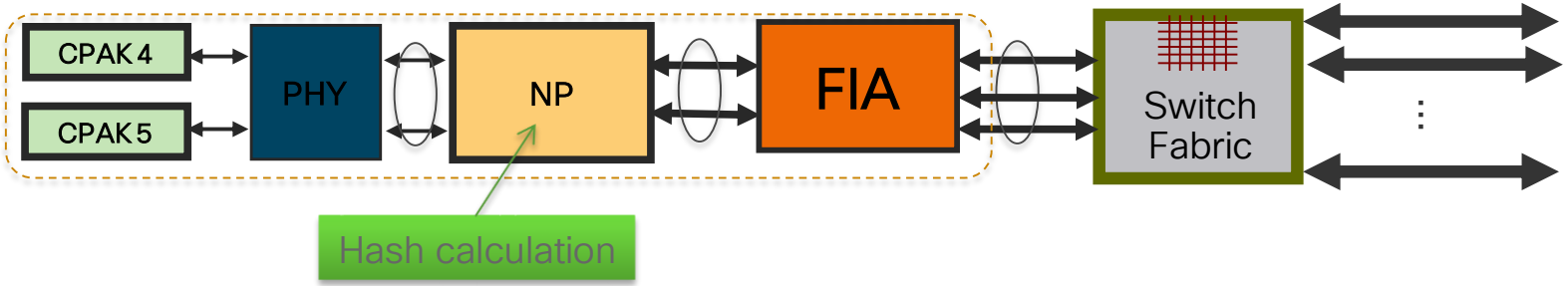
ASR9000 Load Balancing



Ciscolive!

ASR9k Load-balancing Essentials

NP µcode hash



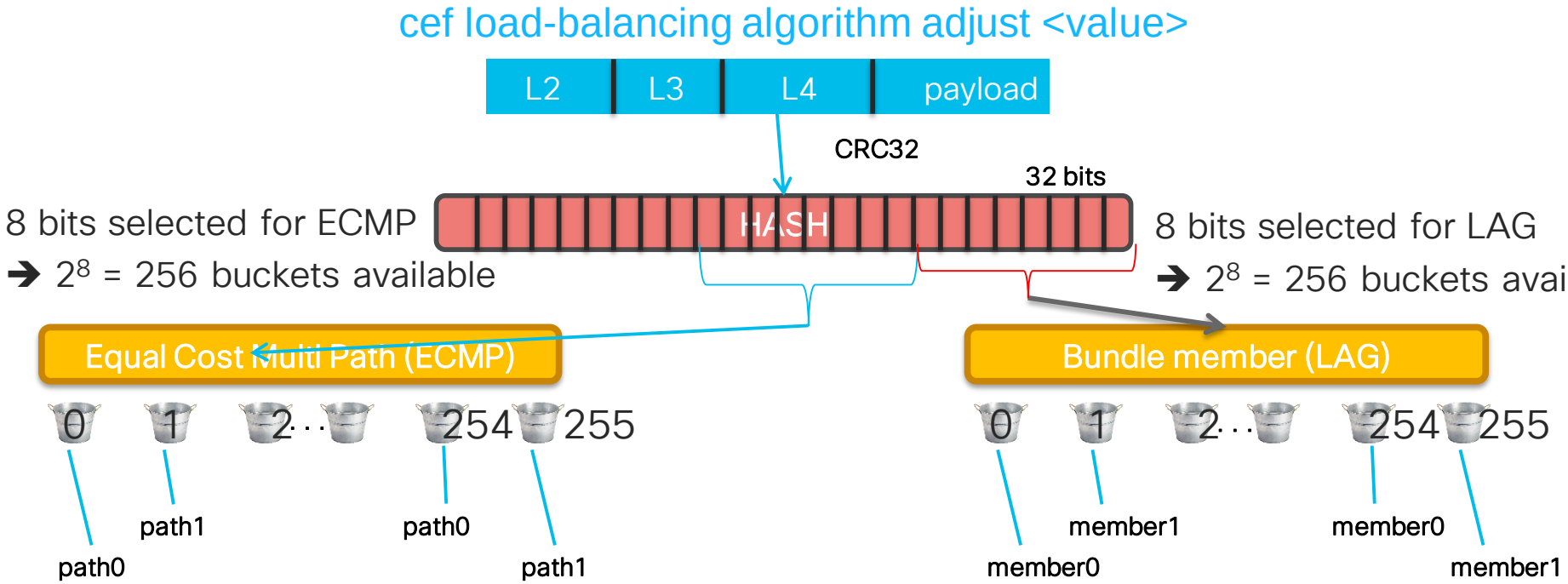
- Every packet received on an NPU undergoes a 32-bit CRC based **hash** computation.
  - Fields used as input for the hash calculation depend on packet encapsulation
- Different sets of 8 bits of the hash are used for different purposes:
  - Bundle member selection
  - Non recursive Path (IGP/OSPF/ISIS)
  - Recursive path (eg bGP)

HASH_LAG_MASK(x) ((x) & 0xFF)
HASH_IGP_MASK(x) (((x) & 0xFF00) >>8)
HASH_BGP_MASK(x) (((x) & 0x1FE000) >> 13)

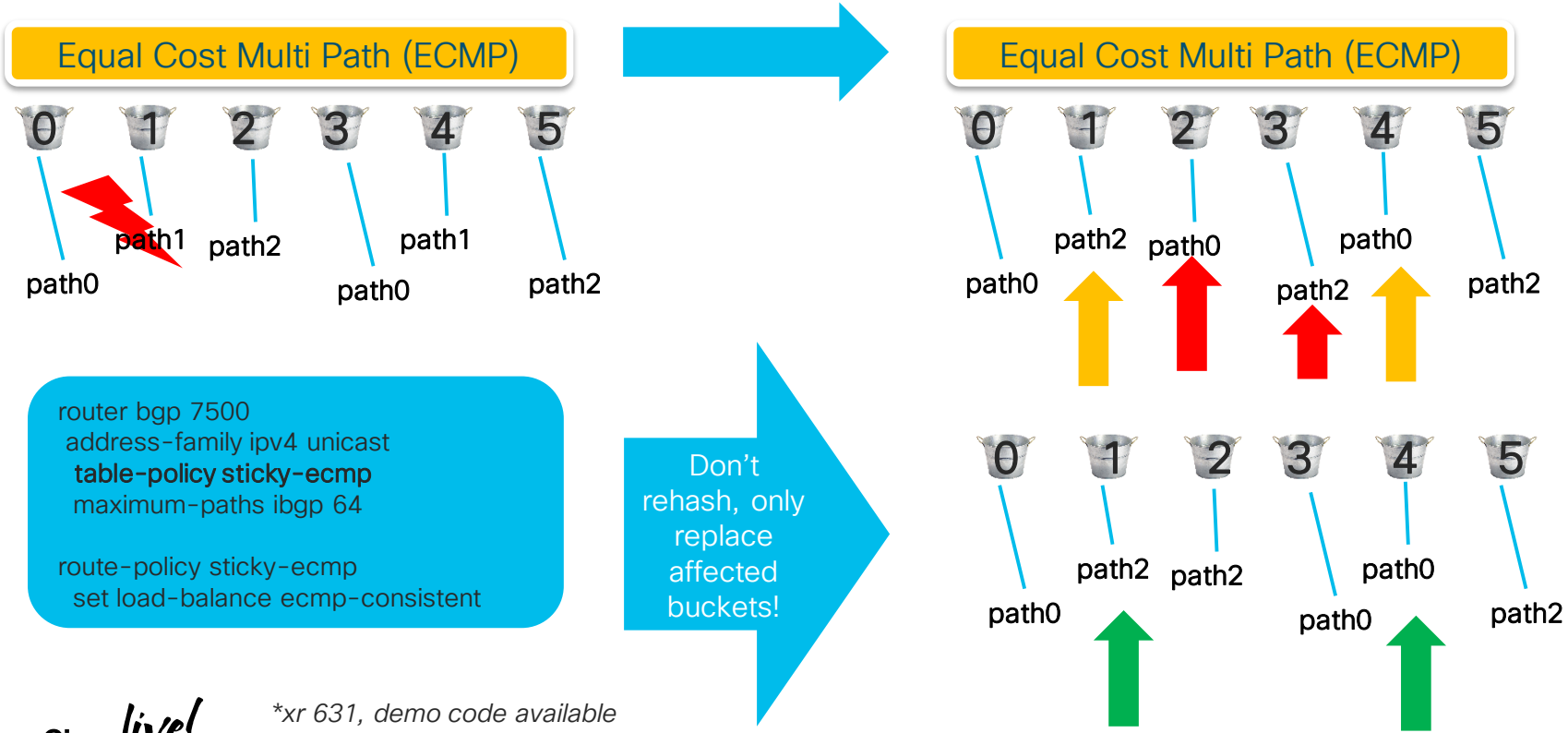
  - VQI selection in case of 100G/40G Typhoon egress interface

ASR9k Load-balancing Essentials

Concept of a hash and its usage: ECMP and LAG



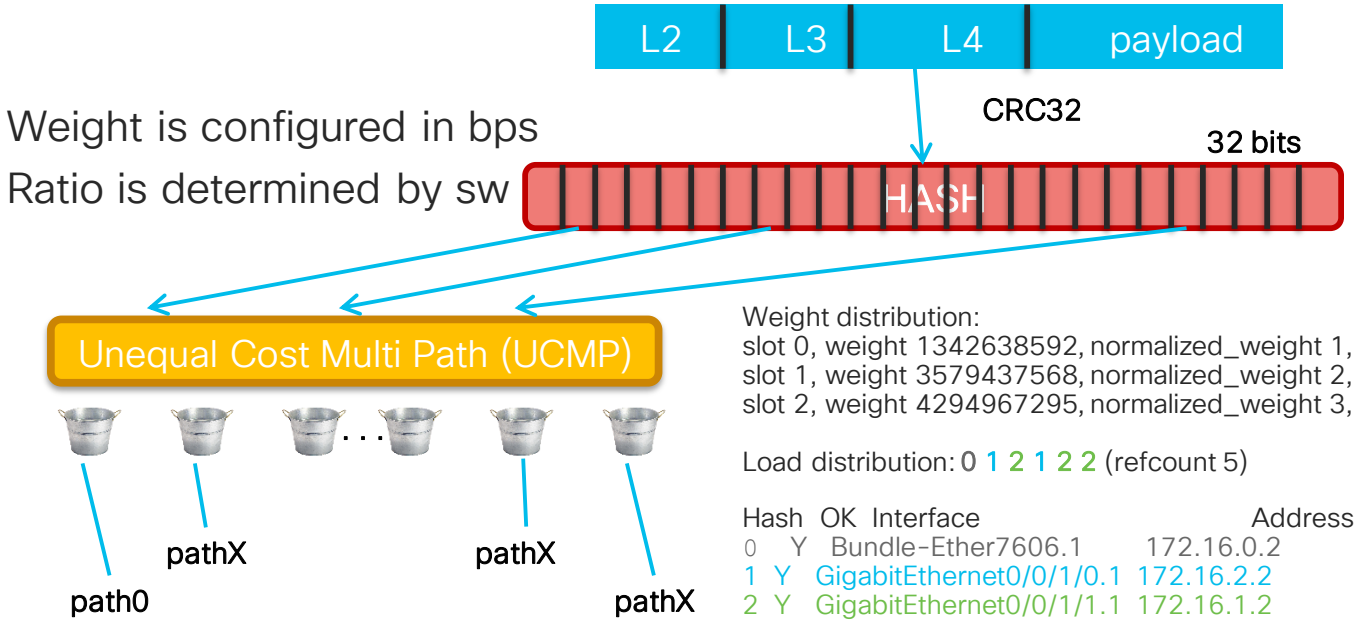
Rehashing consequences and sticky ECMP*



*xr 631, demo code available

ASR9k Load-balancing Essentials

Concept of a hash and its usage: UCMP



Weight distribution:
slot 0, weight 1342638592, normalized_weight 1, class 0
slot 1, weight 3579437568, normalized_weight 2, class 0
slot 2, weight 4294967295, normalized_weight 3, class 0

Load distribution: 0 1 2 1 2 2 (refcount 5)

Hash	OK	Interface	Address
0	Y	Bundle-Ether7606.1	172.16.0.2
1	Y	GigabitEthernet0/0/1/0.1	172.16.2.2
2	Y	GigabitEthernet0/0/1/1.1	172.16.1.2
3	Y	GigabitEthernet0/0/1/0.1	172.16.2.2
4	Y	GigabitEthernet0/0/1/1.1	172.16.1.2
5	Y	GigabitEthernet0/0/1/1.1	172.16.1.2

Load distribution per path depends on relative (normalised) weight, pattern repeated over buckets

ECMP/UCMP Hash Input

A: IPv4 Unicast or IPv4 to MPLS

- 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, next-hop of RPF

C: MPLS to MPLS or MPLS to IPv4

- # of labels <= 4:
  - Payload is IP: same as IPv4 unicast
  - Payload is other: 4th label, Router ID
- # of labels > 4 : Inner most label and Router ID

Bundle Hash Input

L3 bundle:

- follows "A" or "C", depending on packet type

L2 access bundle:

- SMAC, DMAC, RID (default)
- IP SA, IP DA, RID (if configured under l2vpn)

MPLS enabled core facing bundle with L2 access:

- PW label (default)
- SMAC, DMAC, RID (if configured under l2vpn)
- IP SA, IP DA, RID (if configured under l2vpn)

L2VPN vs L3VPN MPLS Loadbalancing

- **Labeled packet**: 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
- On P it can be IP version (in MPLS/IP) or it can be the DMAC (in EoMPLS).
- RULE: If you have EoMPLS **AND** MACs starting with a 4 or 6. You HAVE to use **Control-Word**
- Control Word inserts zeros after the inner label showing the P nodes to go for label based balancing.

- In EoMPLS, inner label is VC label – LB per VC then. More granular spread can be achieved with FAT PW

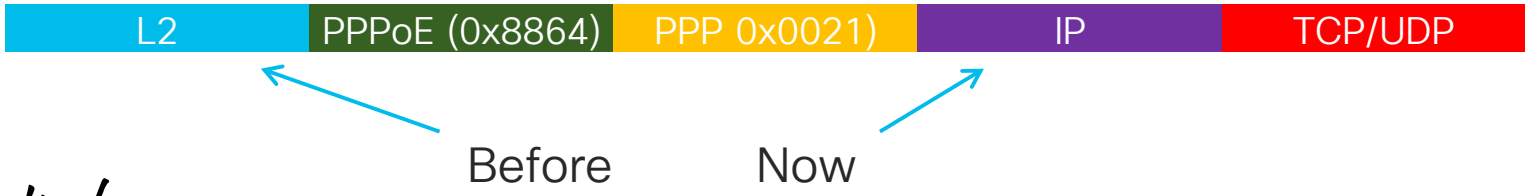
PPPoE Load-balancing

Rel 6.5.1

- Originally treated PPPoE as L2 in terms of LB
- Use cases:

  BNG deployment scenario, i.e. where subscriber traffic is carried through an L2VPN MPLS access network.

  - Label imposition on access PE. FAT label based on DMAC/SMAC will yield one flow per subscriber
  - Label disposition on access PE with Bundle towards DSLAM. Bundle member selection will be based on a very small set of SMAC+DMAC
- Using L3+L4 behind PPPoE header for hash calculation would yield better load balancing



ASR9000 L2VPN Load-Balancing (cont.)

- ASR9000 PE Imposition load-balancing behaviors
  - Per-PW based on MPLS VC label (default)
  - Per-Flow based on L2 payload; i.e.
    DMAC, SMAC, RID
  - Per-Flow based on L3/L4 payload; i.e.
    L3 D_IP / L3 S_IP / L4 D_port / L4 S_port(1), RTR ID
- ASR9000 PE Disposition load-balancing behaviors
  - Per-Flow load-balancing based on L2 payload; i.e. DMAC, SMAC (default)
  - Per-Flow load-balancing based on L3/L4 payload; i.e. L3 D_IP / L3 S_IP / L4 D_port / L4 S_port

PE Per-Flow load-balance based on L2

```
!
l2vpn
 load-balancing flow src-dst-mac
```

PE Per-Flow load-balance based on L3/L4

```
!
l2vpn
 load-balancing flow src-dst-ip
```

(1) Typhoon/Tomahawk LCs required for L3&L4 hash keys. Trident LCs only capable of using L3 keys

Loadbalancing knobs and what they affect

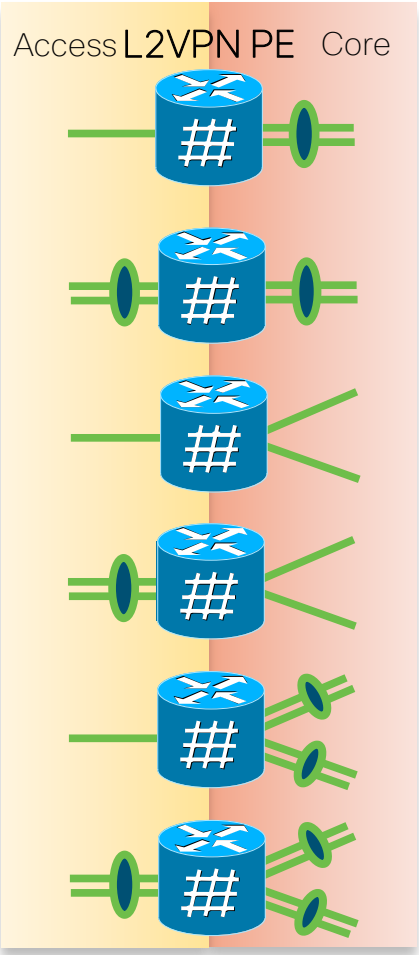
- L2vpn loadbalancing src-dest-ip
  - For L2 bundle interfaces egress out of the AC
  - FAT label computation on ingress from AC towards core
    - Note: upstream loadbalancing out of core interface does not look at fat label (inserted after hash is computed)
- On bundle (sub)interfaces:
  - Loadbalance on srcIP, dest IP, src/dest or fixed hash value (tie vlan to hash result)

  Used to be on L2transport only, now also on L3 (XR53).
- GRE (no knob needed anymore)
  - Encap: based on incoming ip
  - Decap: based on inner ip
  - Transit: based on inner payload if incoming is v4 or v6 otherwise based on GRE header
    - So running mpls over GRE will result in LB skews!
    - Tunnelkey possible via **cef loadbalancing fields l4 gtp**

Loadbalancing FAQ

- If packets are fragmented, L4 is omitted from the hash calculation
- Include IPv6 flow label in hash calculation (XR release 6.0 and later)

  cef load-balancing fields ipv6 flow-label
- Show cef exact route or bundle hash BE<x> can be used to feed info and determine actual path/member
  - shadow calculation; *should* yield the same result as HW
- Mixing Bundle members between trident/typhoon/tomahawk is not recommended.

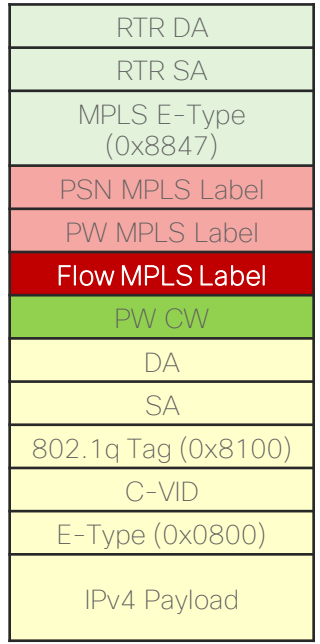


FAT vs Entropy label

Problem: How LSRs load-balance traffic from PW flows across core ECMPs and Bundle interfaces?

- L3 flows - we effectively use 5-tuple on LSR (IP/L4port/router ID).
  Thus we don't need additional overhead entropy can bring
- L2 flows - We use FAT as cant use 5-tuple
- RFC6391: Only PEs push/pop Flow label
- P routers not involved in any signaling/handling/manipulation of Flow label

Item	FAT	Entropy
Neighbor support required?	N	Y
Loadbalance for	L2 only	L2 + MPLS/IP
Use case (XR)	L2: FAT L3 native ECMP (IP)	P router support only (understands entropy, no insertion, no PE support)



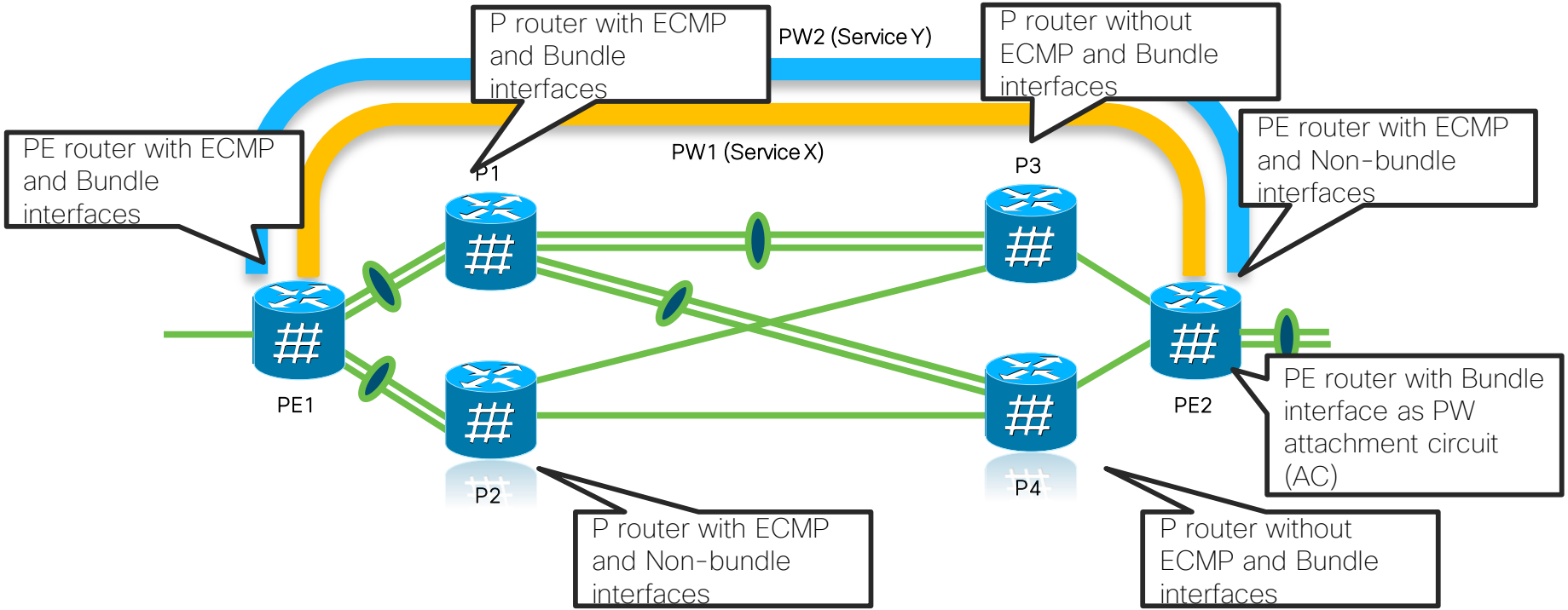
Flow Aware Transport PWs (cont.)

- ASR9000 PE capable of negotiating (via LDP – RFC6391) the handling of PW Flow labels
- ASR9000 also capable of manually configure imposition and disposition behaviors for PW Flow labels
- Flow label value based on L2 or L3/L4 PW payload information
- ASR9000 PE capable of load-balancing regardless of the presence of Flow Label
  - Flow label aimed at assisting P routers

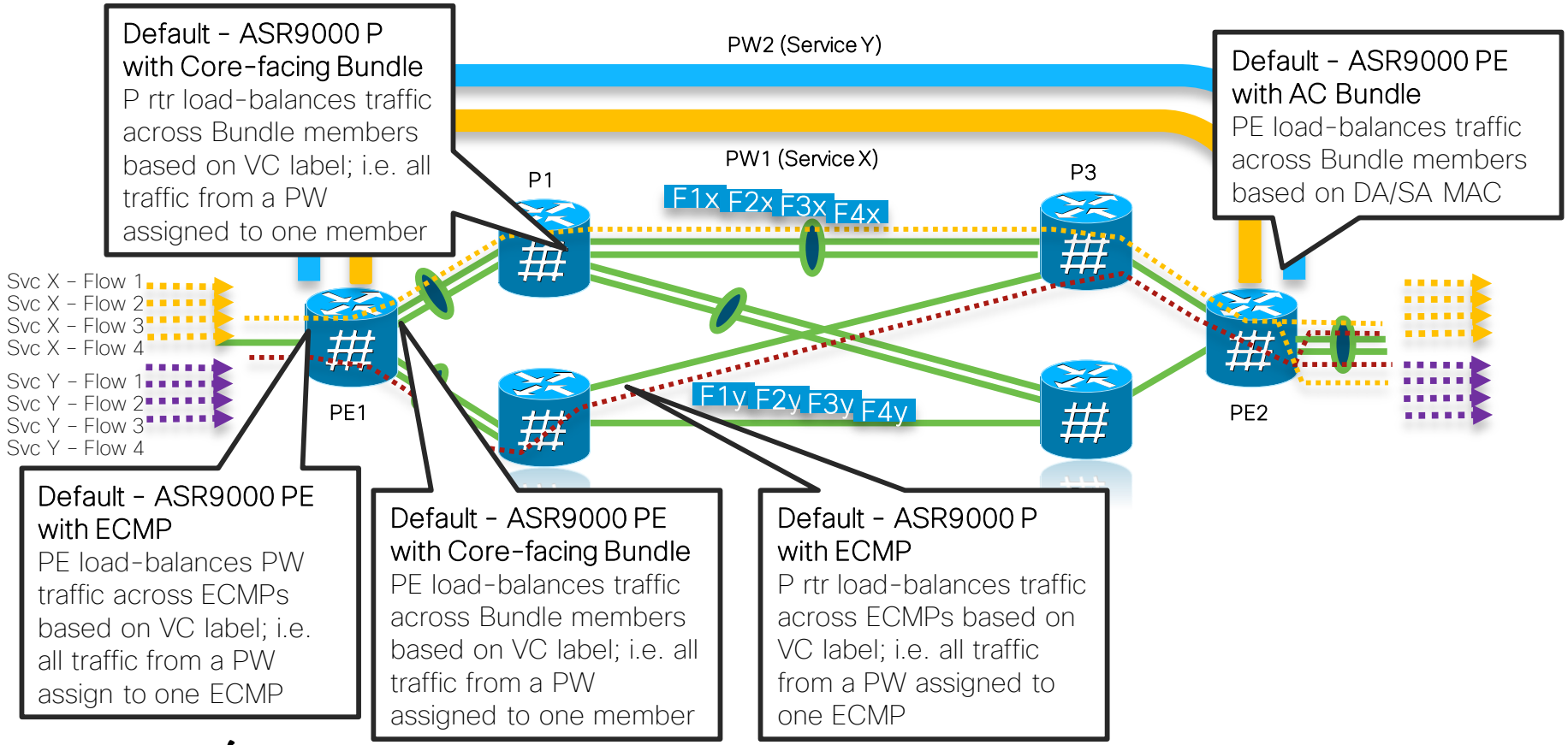
```
!
l2vpn
 pw-class sample-class-1
  encapsulation mpls
   load-balancing flow-label both
 !
 pw-class sample-class-1
  encapsulation mpls
   load-balancing flow-label tx
 !
 pw-class sample-class-1
  encapsulation mpls
   load-balancing flow-label rx
```

```
!
l2vpn
 pw-class sample-class
  encapsulation mpls
   load-balancing flow-label both static
 !
```

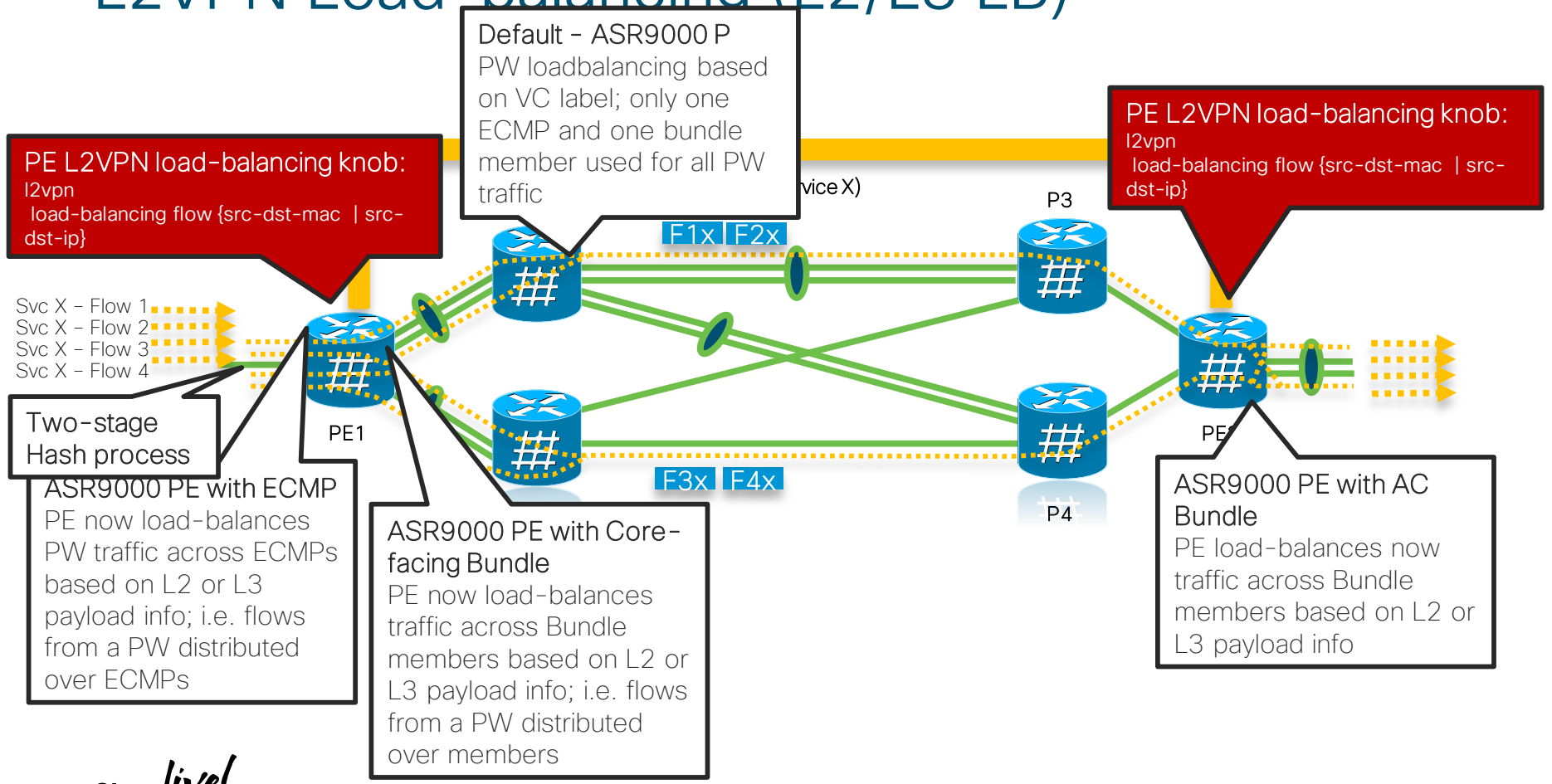
L2VPN Load-balancing (E2E Scenario)



L2VPN Load-balancing (Per-VC LB)



L2VPN Load-balancing (L2/L3 LB)



L2VPN Load-balancing (L2/L3 LB + FAT)

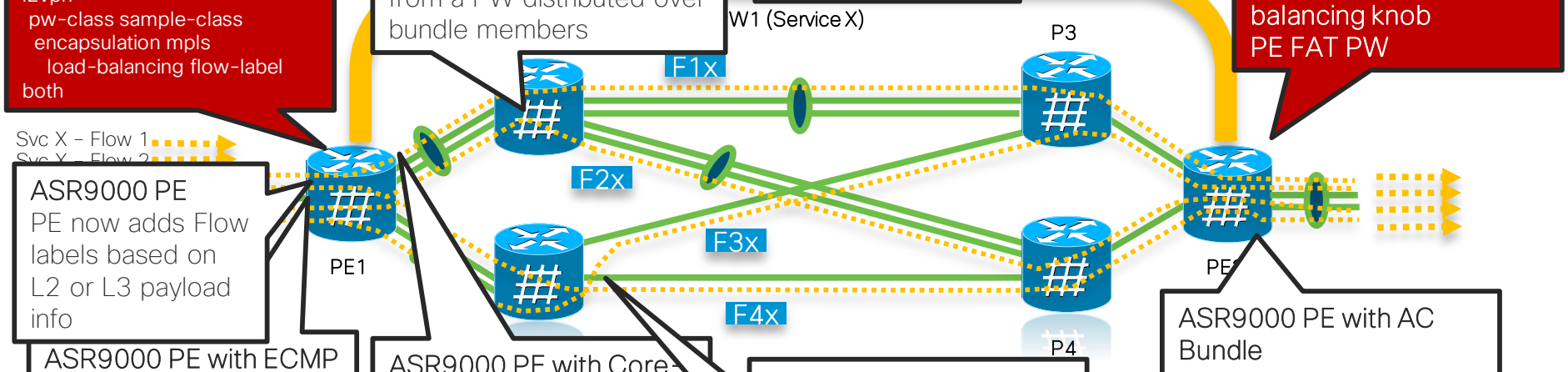
PE L2VPN load-balancing knob (same as before)

PE FAT PW
```
l2vpn
 pw-class sample-class
  encapsulation mpls
   load-balancing flow-label
both
```

ASR9000 P with Core-facing Bundle
PW loadbalancing based on Flow label; i.e. flows from a PW distributed over bundle members

No new configuration required on P routers

PE L2VPN load-balancing knob
PE FAT PW



ASR9000 PE
PE now adds Flow labels based on L2 or L3 payload info

ASR9000 PE with ECMP
PE now load-balances PW traffic across ECMPs based on L2 or L3 payload info; i.e. flows from a PW distributed over ECMPs

ASR9000 PE with Core-facing Bundle
PE now load-balances traffic across Bundle members based on L2 or L3 payload info; i.e. flows from a PW distributed over members

ASR9000 P with ECMP
P rtr now load-balances traffic across ECMPs based on Flow label; i.e. flows from a PW distributed over ECMPs

ASR9000 PE with AC Bundle
PE load-balances now traffic across Bundle members based on L2 or L3 payload info

L2VPN LB Summary

- ASR9000 as L2VPN **PE** performs multi-stage hash for ECMPs / Bundle
  - User-configurable hash keys allows to use L2 fields or L3/L4 fields in PW payload
- ASR9000 (as PE) complies with RFC6391 (FAT PW) to POP/PUSH Flow labels
  - PE load-balancing is performed irrespective of Flow PW label presence
  - FAT PW allows for load-balancing of PW traffic in the Core
  - Cisco has prepared a draft to address current gap of FAT PW for BGP-signaled PWs
- ASR9000 as L2VPN **P** router :
  - Always use bottom of stack MPLS label for hashing
  - Bottom of stack label could be PW VC label or FAT label for L2VPN

Satellite ICL Bundle Load-Balancing

Static
(port, VC, etc. based)

Port hash

Satellite ICL Bundle

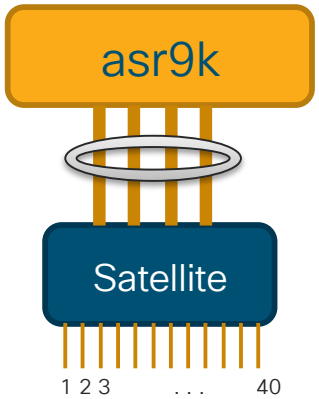
ICL bundle member selection

hash_value = (satellite_port_number + 2) % number_of_ICL_members

Eg: on a bundle ICL with 4 members, hash value of satellite port 13 is:

X = (13 + 2) % 4 = 15 % 4 = 3

# of ICL members	Satellite Port	Hash
4	1	3
4	2	0
4	3	1
4	4	2



ASR9000 Unicast Fabric Load Balancing

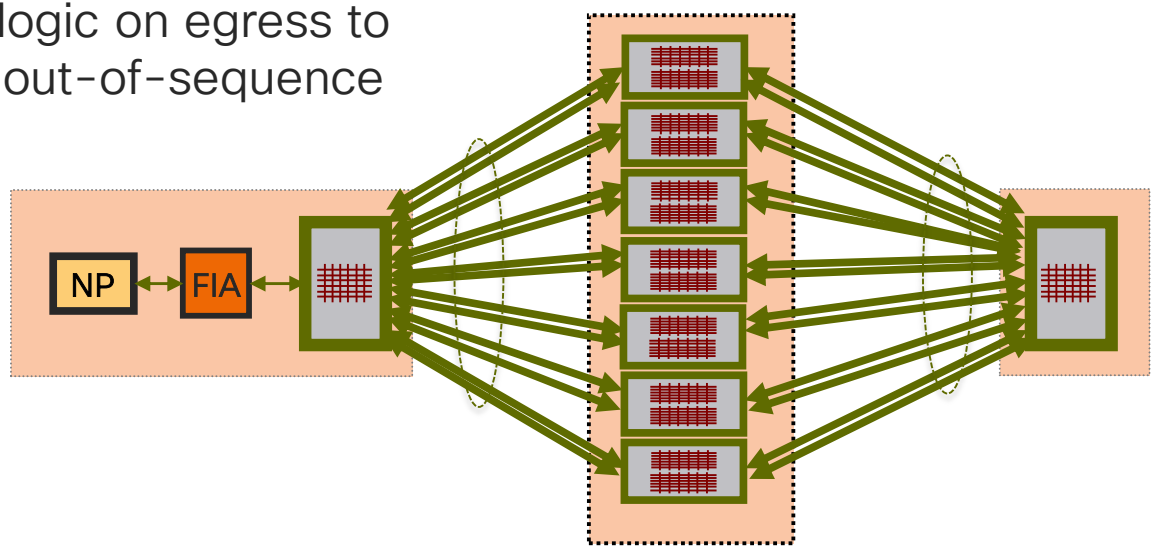
Static
(port, VC, etc.
based)

Round robin

Ingress FIA

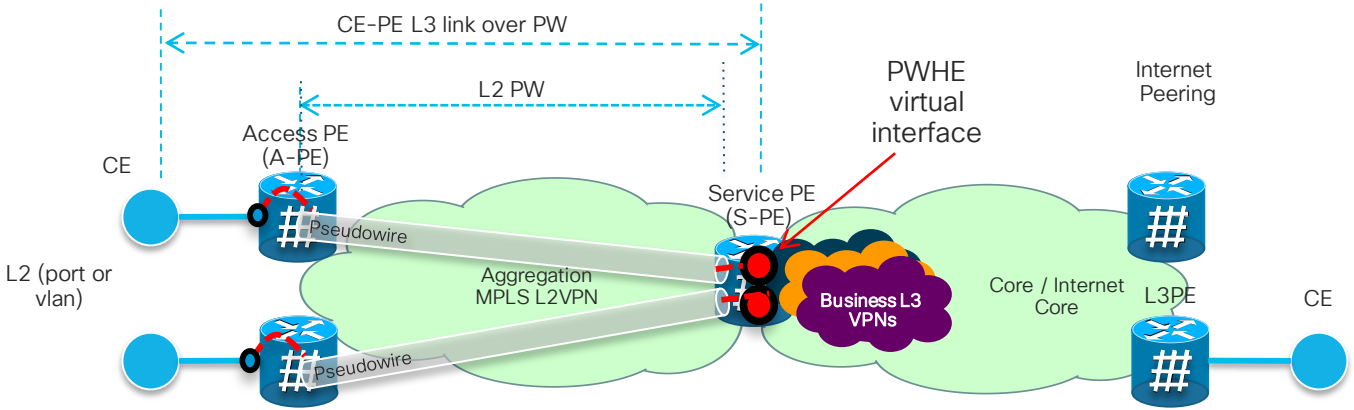
Fabric link selection

- Round-robin across all fabric links; optimal fabric BW utilisation
- Special logic on egress to prevent out-of-sequence



PWHE Load Balancing

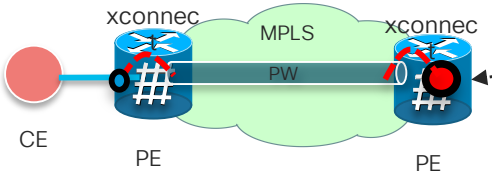
Pseudo-Wire Head-End



- Unified MPLS end-to-end transport architecture
- Flexible service edge placement with virtual PWHE interface
  - L2 and L3 interface/sub-interface
  - Feature parity as regular L3 interface: QoS, ACL, Netflow, BFD, etc
  - CE-PE routing is over MPLS transport network. It doesn't need direct L3 link any more
- CE-PE virtual link is protected by the MPLS transport network

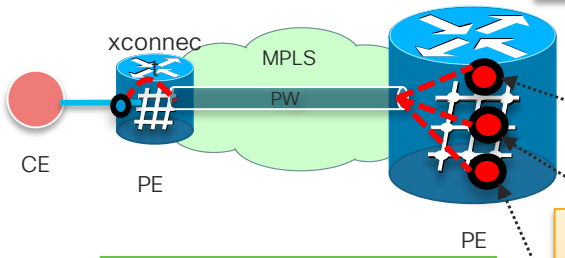
Pseudo-Wire Head-End Configuration Examples

Bind PW-Ether to MPLS Interfaces



PWHE L3/L2 Main Interface Example

```
interface pw-ether 200
 vrf vrf0001
 ipv4 address 11.0.0.1/24
 ipv6 address 2001:da1::1/64
 attach generic-interface-list pwhe_gi_2
!
generic-interface-list pwhe_gi_2
 interface Bundle-Ether100
 interface TenGigE0/5/0/12
```



PWHE L3/L2 Sub-interface Example

```
interface pw-ether 100.100
  encap dot1q 100
  vrf vpn-red
  ipv4 address 10.1.1.2/24
```

```
interface pw-ether 100.200 l2transport
  encap dot1q 200
!
interface pw-ether 100.300 l2transport
  encap dot1q 300
```

Bind PW-Ether to L2 VC

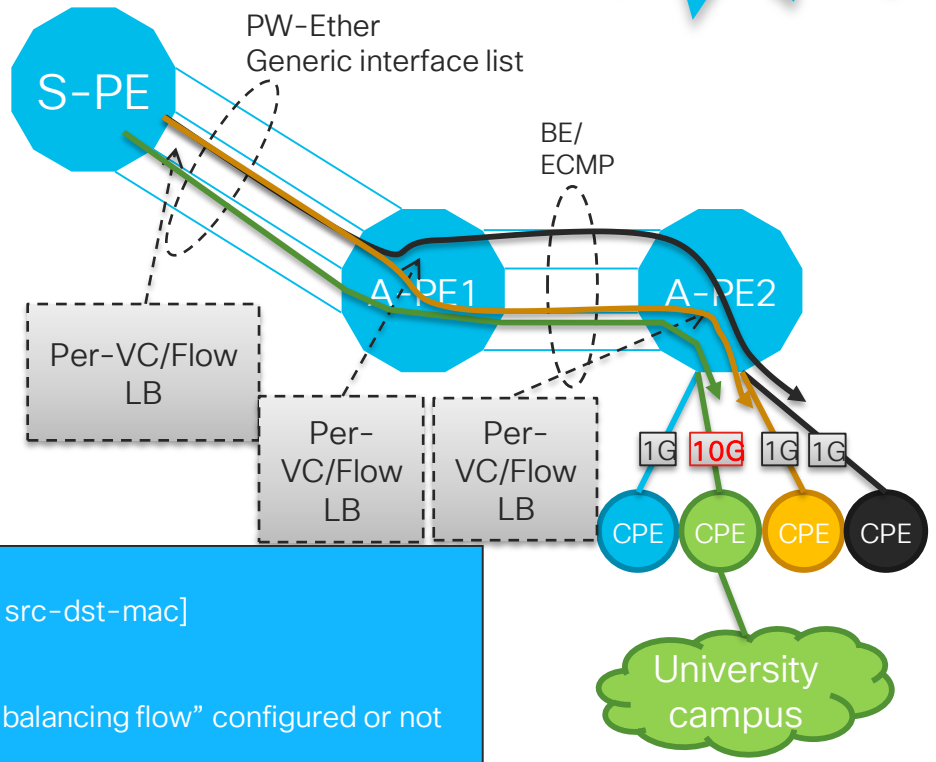
```
l2vpn
 xconnect group pwhe
  p2p pwhe-red
   interface pw-ether 200
   neighbor 100.100.100.100 pw-id 1
```

```
l2vpn
 xconnect group cisco
  p2p service2
   interface PW-Ether100.200
   neighbor ipv4 1.1.1.1 pw-id 22
 bridge group group1
  bridge-domain domain2
   interface PW-Ether100.300
   vfi cisco
    neighbor 192.0.0.1 pw-id 100
    neighbor 192.0.0.2 pw-id 100
```

PWHE Load-Balancing Today

Rel 6.5.1

- Before 6.5.1 Only Per-VC load-balancing
  - L2VPN load balancing config is ignored
  - Flow label config in pw-class is ignored
- From 6.5.1 supports both load-balancing and FAT



a) **no config** -> PW-label (aka VC-label) based hashing
b) **Global config mode only**: l2vpn load-balancing flow [src-dst-ip | src-dst-mac]
   -> Flow based hashing
c) **l2vpn pw-class mode**<>
   encapsulation mpls load-balancing pw-label, no matter if "load-balancing flow" configured or not
   -> PW-label based hashing

PWHE Troubleshooting

```
RP/0/RP0/CPU0:ASR9922#show interface pw-ether 300
<..>
   Generic-Interface-List: pwhe_gi_2
```

Find the related generic interface list

```
RP/0/RP0/CPU0:ASR9922#show generic-interface-list name pwhe_gi_2
Wed Dec 27 12:25:42.062 EST
generic-interface-list: pwhe_gi_2 (ID: 3, interfaces: 5)
   Bundle-Ether100 - items pending 0, downloaded to FIB
   TenGigE0/5/0/12 - items pending 0, downloaded to FIB
   TenGigE0/5/0/13 - items pending 0, downloaded to FIB
   TenGigE0/5/0/14 - items pending 0, downloaded to FIB
   TenGigE0/5/0/15 - items pending 0, downloaded to FIB
 Number of items: 1
 List is downloaded to FIB
```

Topology?

Is this the full list of interfaces on which the MPLS packets from A-PE can be received?

Is this the full list of output interfaces towards PW tail-end?

Check route towards tail-end

```
RP/0/RP0/CPU0:ASR9922#show l2vpn xconnect interface PW-Ether 300
XConnect               Segment 1                  Segment 2
Group     Name       ST   Description          ST     Description          ST
------------------------  -----------------------------  -----------------------------
pwhe_lb   pwhe_lb    UP   PE300                UP     8.8.8.8              300    UP
------------------------  -----------------------------  -----------------------------
```

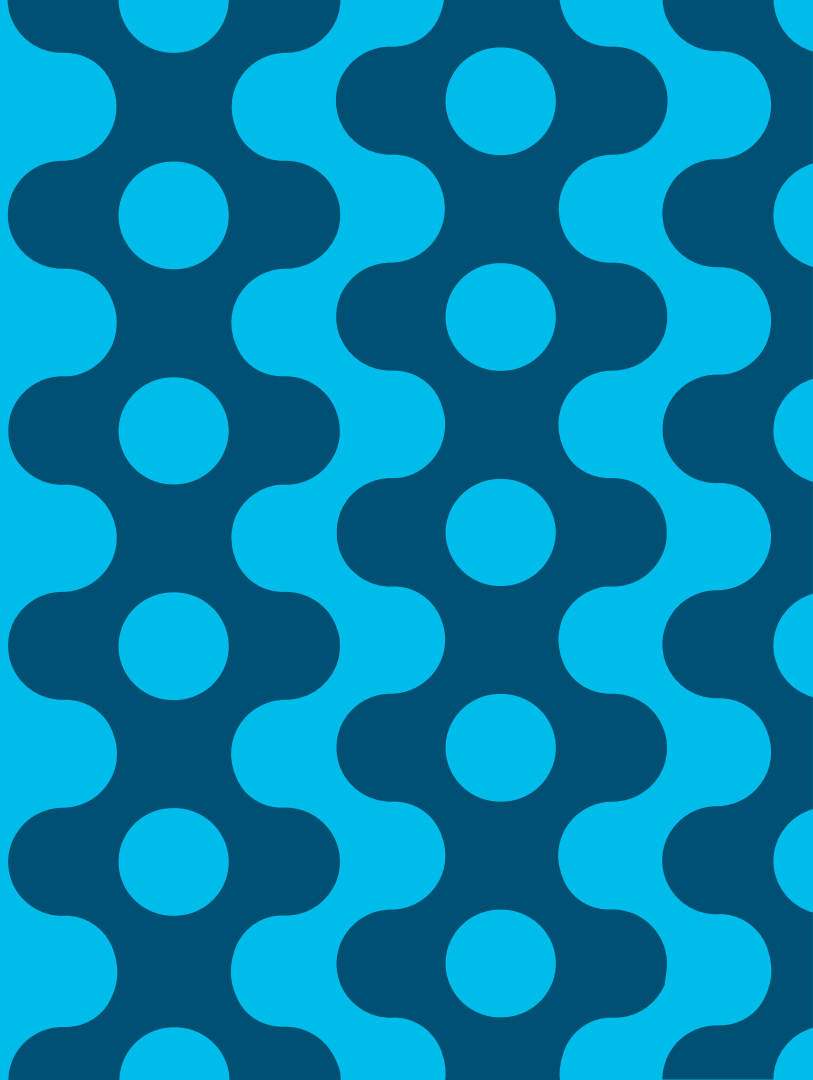
PWHE Troubleshooting

show generic-interface-list [name <*name*>]
show im database interface pw-ether <*n*> verbose
show interface pw-ether <*n*> detail
show proc pwhe_ma

show l2vpn ea pwhe summary location <*location*>
show l2vpn pwhe interface pw-ether <*n*> detail
show l2vpn generic-interface-list private
show l2vpn ma pwhe interface pw-ether <*n*> private
show l2vpn ma pwhe peer private
show l2vpn ma pwhe trace

show mpls lsd forwarding pw-list

HSRP/VRRP and Restrictive Timers

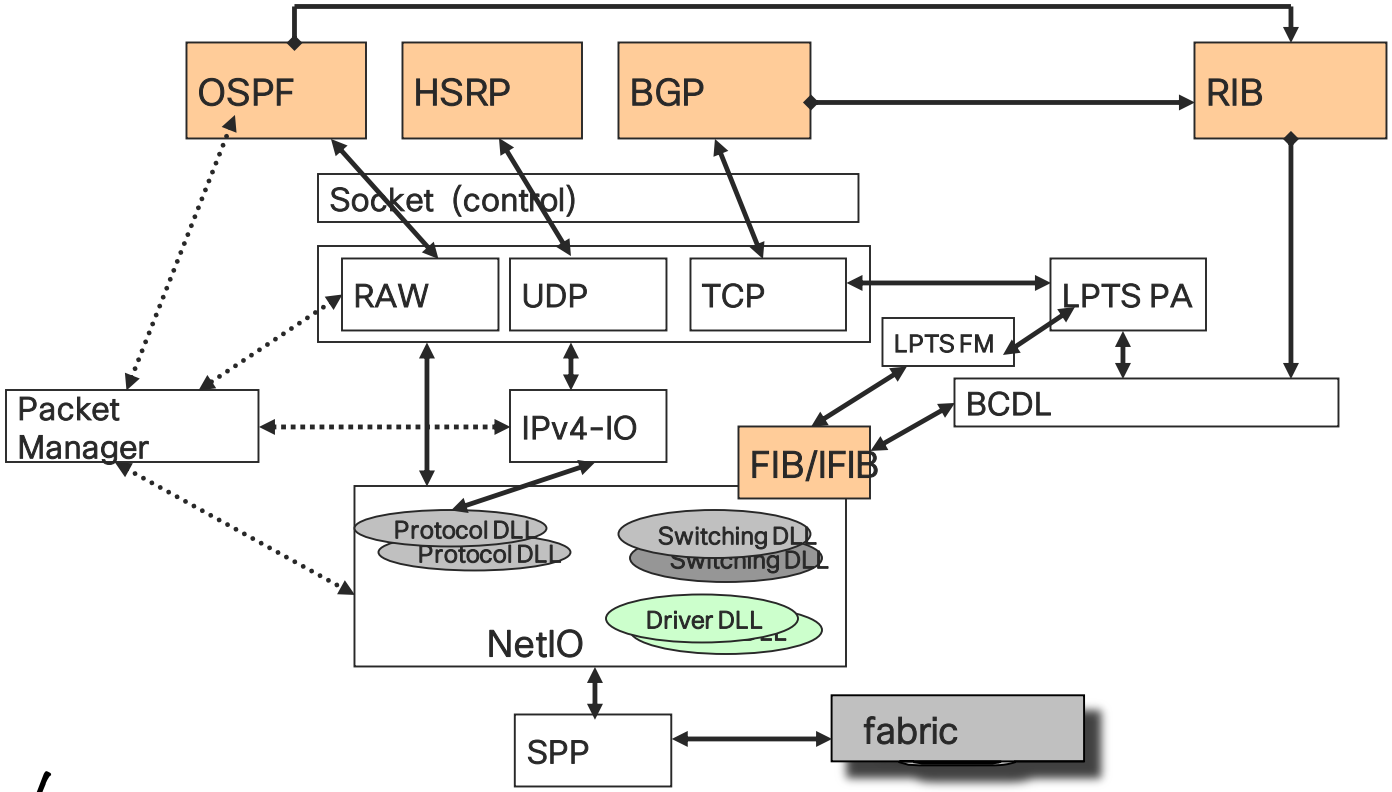


ASR9k Punt/Inject Terminology

- Punt packet:
  - Any packet that NP hands over to the LC or RP CPU is a punt packet
  - Examples: packets destined to the router; packets with destination IP that requires ARP resolution; etc.
- Inject packet:
  - Any packet sent to NP from LC or RP CPU

- NetIO process:
  - Netio process runs on every RP and LC CPU
  - Handles all protocol packets
  - 1 high and 1 normal priority output queue for injected packets
- SPP:
  - Software Packet Path
  - Optimized packet processing
  - CPU pipe-lining of instructions and data cache usage

XR Control Plane Components



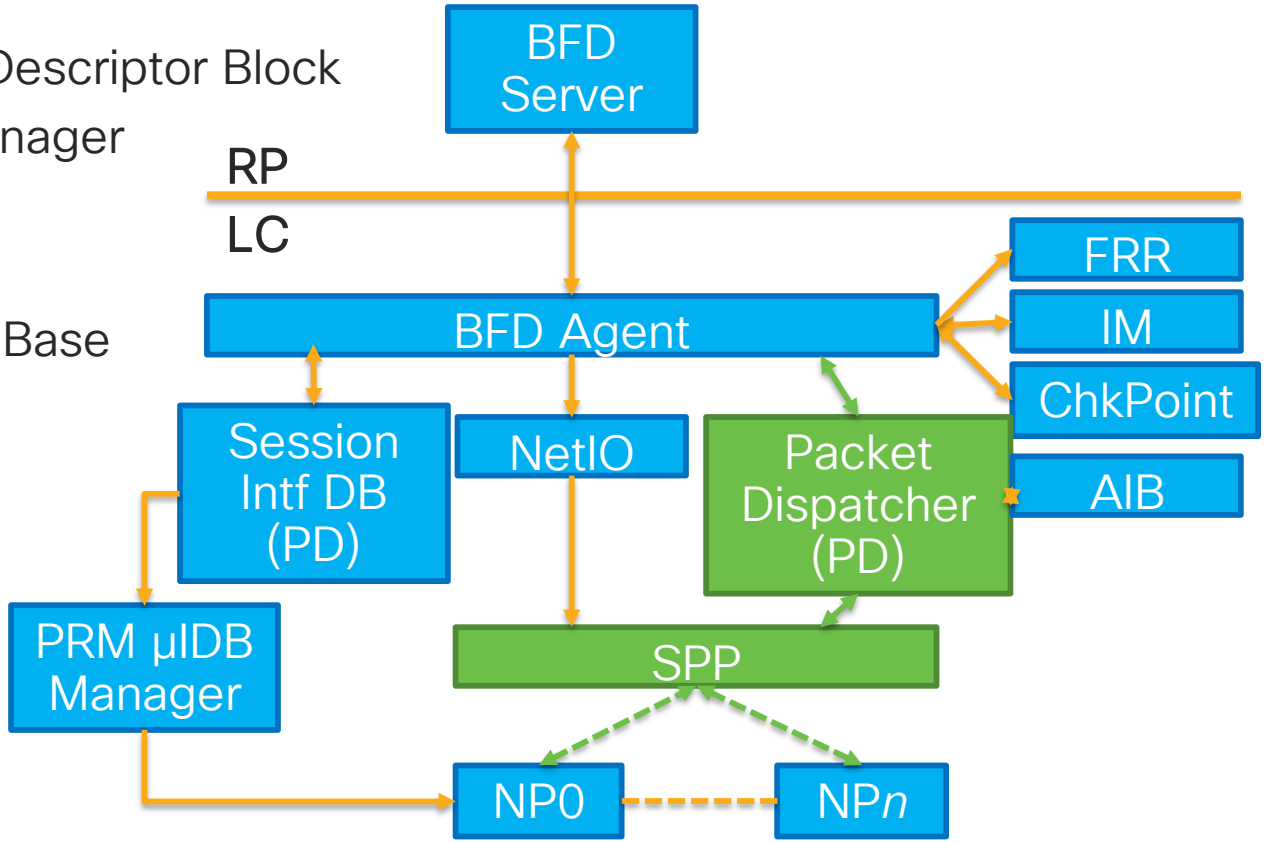
HSRP/VRRP vs BFD Packet Processing

- HSRP runs at normal priority
- Not designed to have restrictive timers
- HSRP Tx hello packet path:
- hsrp → udp → ipv4_io → netio → spp → punt-fpga → fia_rsp → xbar → fia_lc → np → wire
- Performance of the inject path depends on HSRP scale and other processes running on the RP

- BFD process designed for Fast lightweight generic failure detection.
- Hello packet Rx/Tx always on LC CPU
- Threads of BFD process responsible for hello packet exchange run at higher priority
- Session state maintenance and interaction with clients is on RP (bfd server)
- BFD does NOT transmit packets through S/W stack (netio)

BFD Packet Processing Flow

µIDB: Microcode Interface Descriptor Block
PRM: Platform resource manager
FRR: Fast Reroute
IM: Interface Manager
AIB: Adjacency Information Base
SPP: Software Packet Path
NP: Network Processor



BVI Inject Path Difference

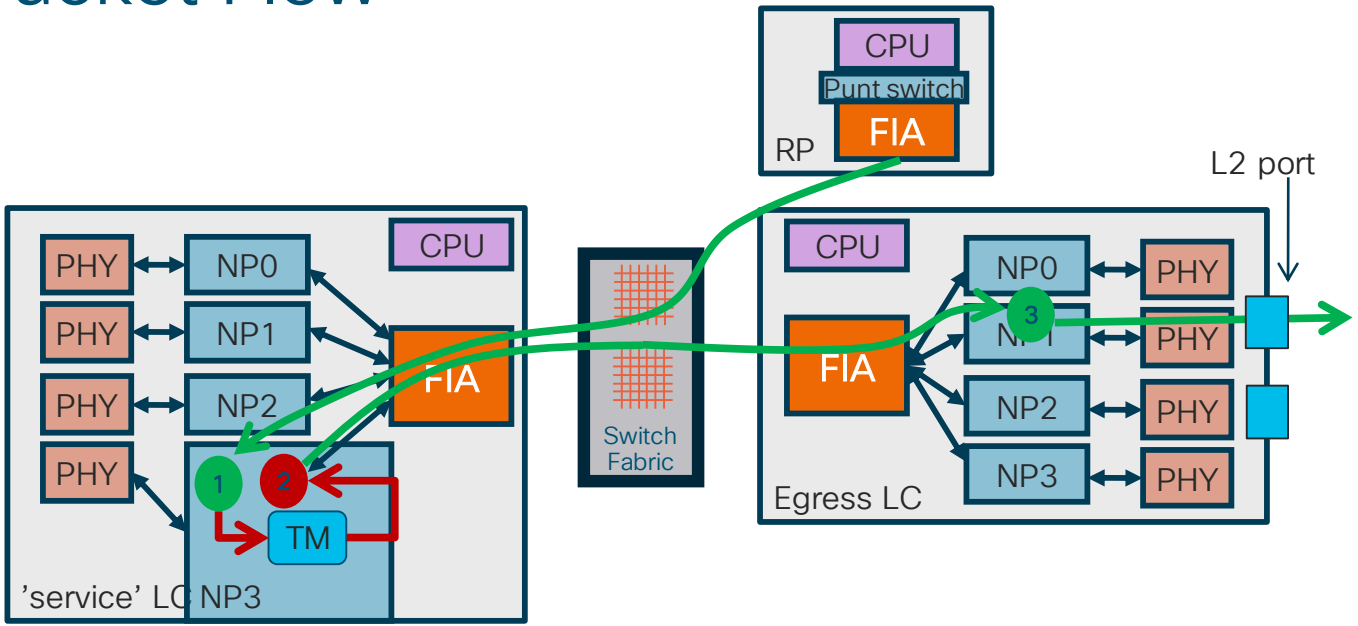
- L2FIB is maintained only on LCs RSP/RP can't select the AC when injecting packets
- RSP delegates L2FIB lookup to a "service card": typically NP0 of the first LC in the chassis

```
RP/0/RSP0/CPU0:PE1#sh adjacency bvi-fia
Service card for BVI interfaces on node 0/RSP0/CPU0
      MPLS packets: Slot 0/1/CPU0 VQI 0xc6
      Non-MPLS packets: Slot 0/1/CPU0 VQI 0xc6

      ----- Service Card List -----

          0/RSP0/CPU0              VQI: Not Applicable
       0/RSP1/CPU0              VQI: Not Applicable
       0/0/CPU0                 VQI: Not Applicable
       0/1/CPU0                 VQI: 0xc6
       0/2/CPU0                 VQI: 0x138
       0/3/CPU0                 VQI: Not Applicable
       0/FT0/SP                 VQI: Not Applicable
       0/FT1/SP                 VQI: Not Applicable
```

(asr9006 chassis; slots 0 and 3 empty)

BVI Inject Packet Flow

- Unicast L3 → L2



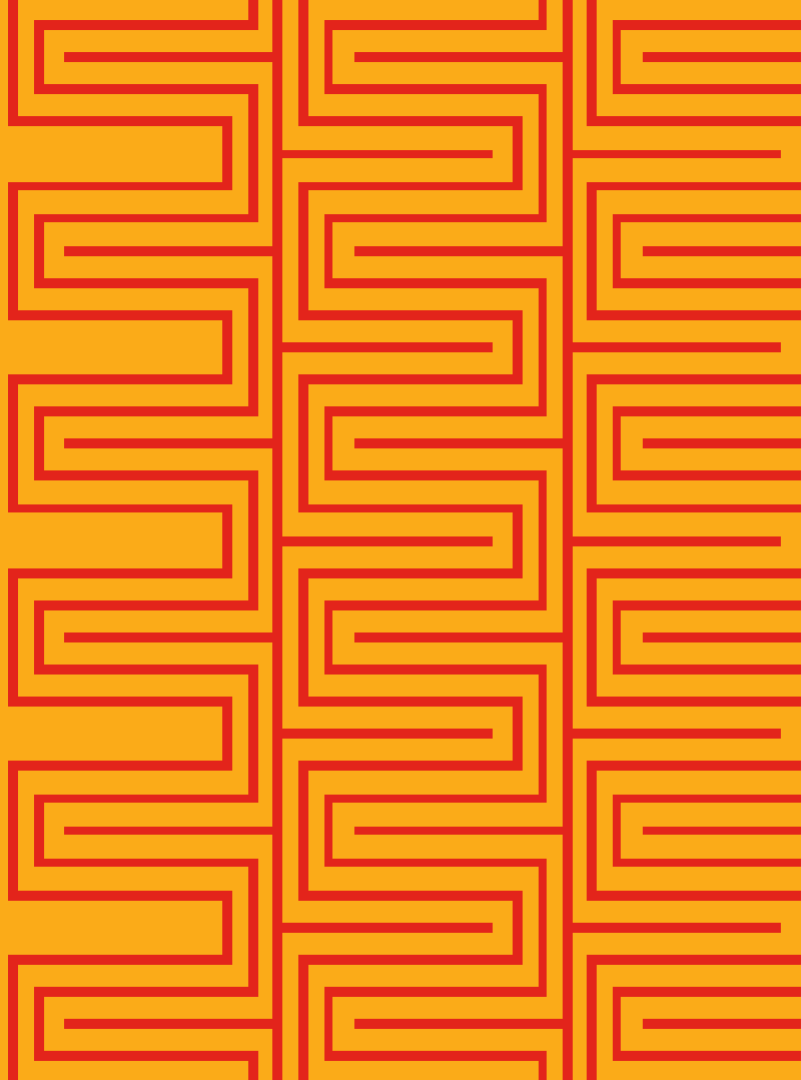
1 • Ingress L3 Lookup & L2 rewrite (including egress BVI counters)

TM • TM packet replication and loop packet back to NP

2 • Ingress L2 Lookup

3 • Egress Regular L2 Lookup

Fragmentation on ASR9k

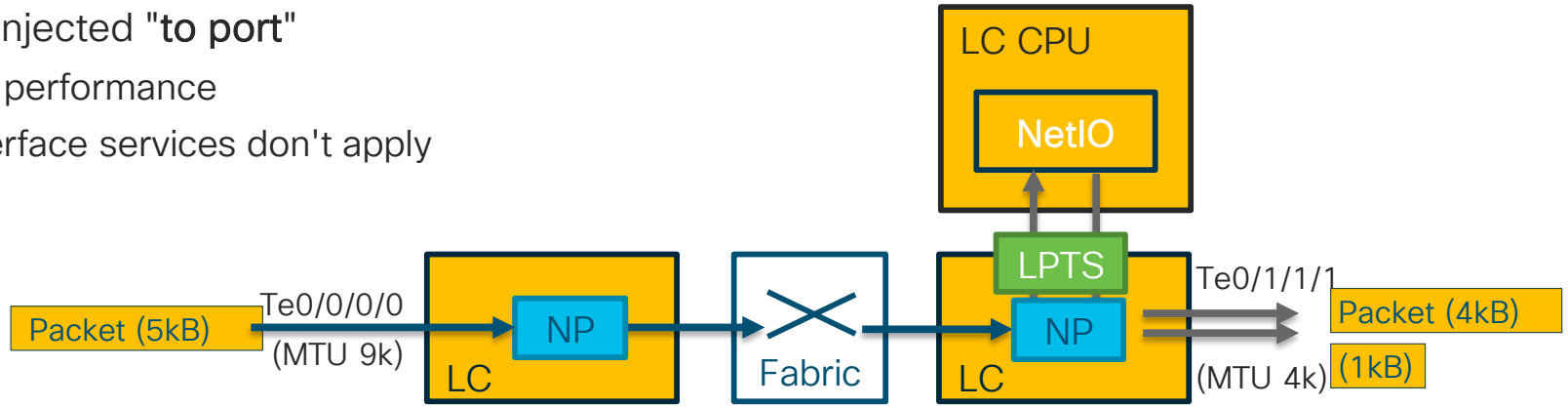


Fragmentation

- Fragmentation is an exception path on all XR platforms.
  - Taking an exception path can lead to packets out of order:
    - Packet A.1 from flow A requires fragmentation, takes exception path.
    - Packet A.2 received shortly after A.1 does not require fragmentation.
    - Packet A.2 is forwarded before A.1 because exception path is slower.
- Fragmentation can be supported only on IPv4, PPPoE and MPLS
  - IPv6 protocol does not define fragmentation.

Fragmentation on ASR9k

- ASR9k is a two-stage forwarding platform → fragmentation is an egress feature.
- Packets that exceed MTU are punted to LC CPU through LPTS static policers:
  IPV4_FRAG_NEEDED_PUNT,PPPOE_FRAG_NEEDED_PUNT,MPLS_FRAG_NEEDED_PUNT
- NetIO fragments the packet and takes the forwarding decision based on SW FIB
- L2 rewrite in NetIO (Adjacency database)
- Packet is injected "**to port**"
  - Benefits performance
  - Sub-interface services don't apply



Tomahawk In-Line Fragmentation on ASR9k

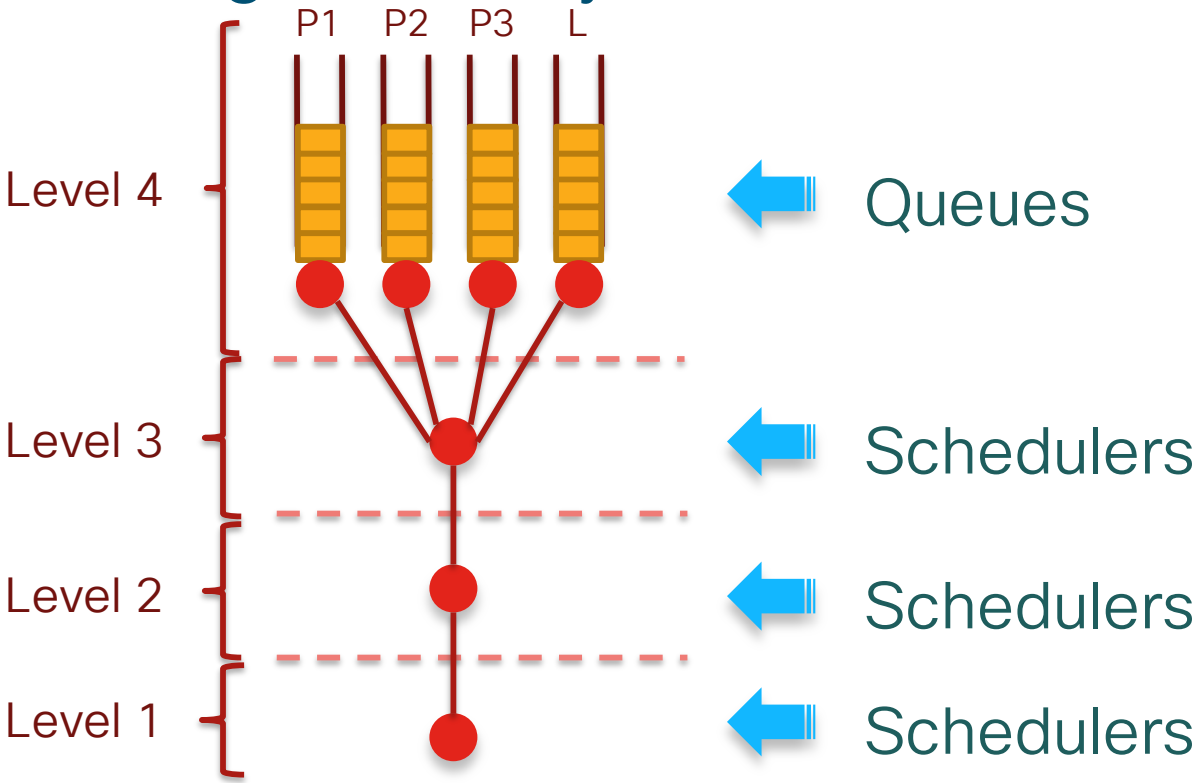
- Restrictions:
  - Works only on main interfaces
  - Works only on clear IPv4 packets
- Related NP counters:
  - MODIFY_CANNOT_FRAGMENT_DROP
    - Frames dropped due to errors during fragmentation
  - MODIFY_FRAGMENT_PROCESSING
    - Frames entering fragmentation processing code
  - MODIFY_FRAGMENTS_SENT

QOS:
Queuing and Scheduler Hierarchy

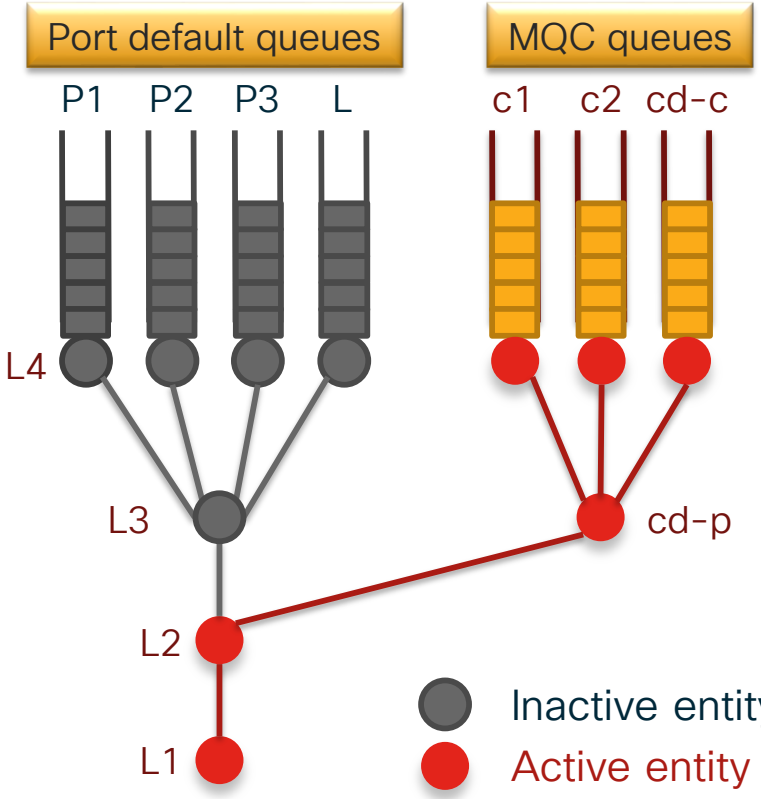
Queuing and Scheduling Hierarchy

- At which level packets from different classes and sub-interface compete with each other
- High priority will trump all other traffic of lower priority
- Priority propagation is enabled by default;
- cannot be turned off.
- Bandwidth remaining sets CIR and gives weight to class/sub-interface

Queuing Hierarchy Of Default Interface Queues



MQC Hierarchy in Queuing ASIC

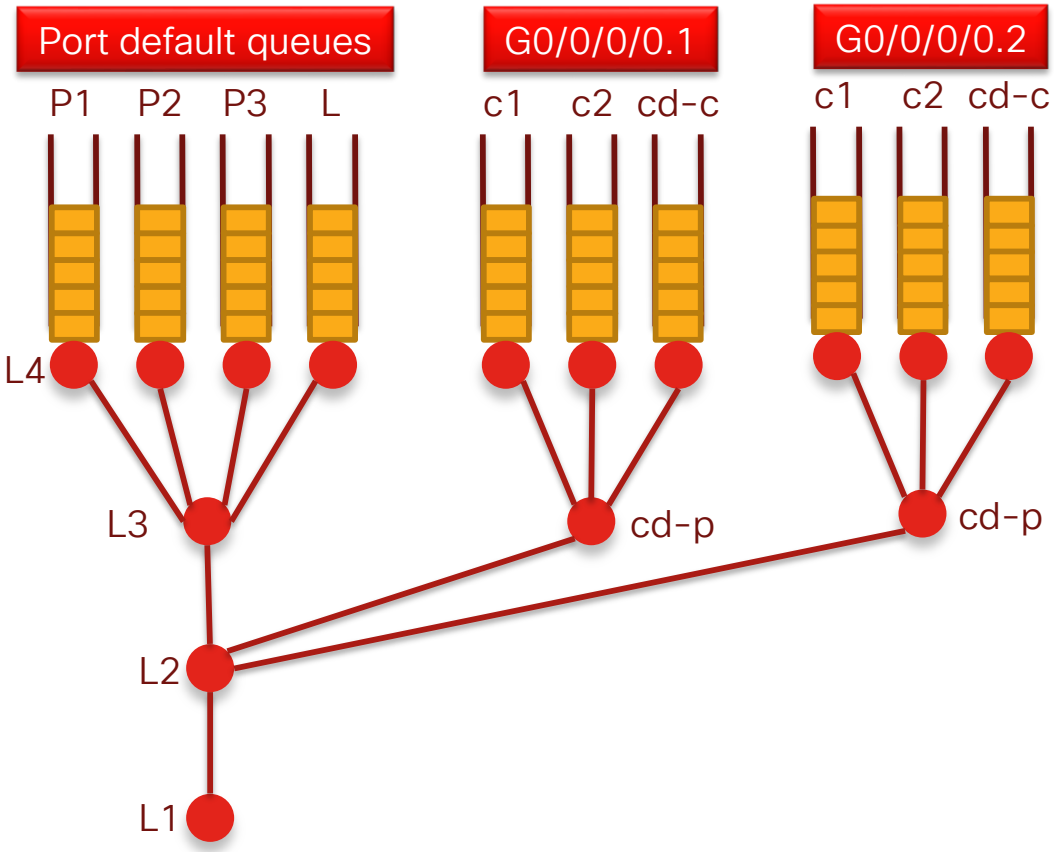



```
policy-map child
 class c1
  priority level 1
  police rate 640 kbps
 class c2
  bandwidth 20 mbps
 class class-default                cd-c
  bandwidth 1 mbps
!
policy-map parent
 class class-default                cd-p
  shape average 35 mbps
  service-policy child
!
interface GigabitEthernet0/0/0/0
 service-policy output parent
```

Inactive entity (P1 queue used for high-priority injected traffic bypassing egress QoS)

Active entity

MQC Hierarchy in Queuing ASIC

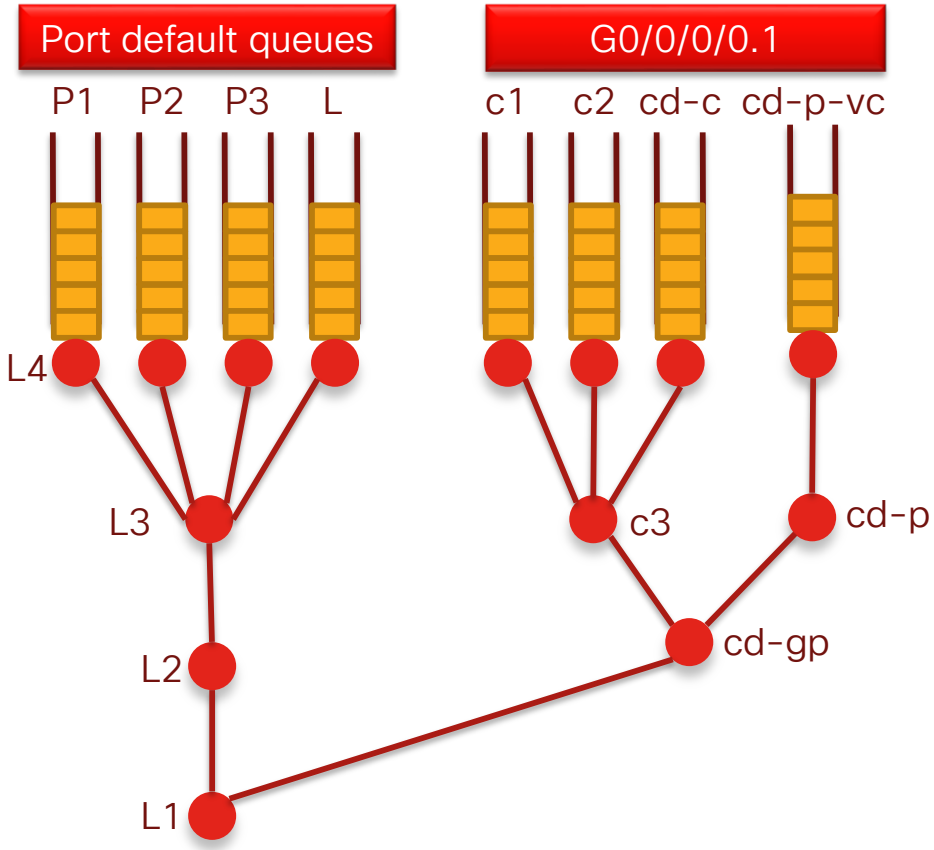



```
policy-map child
 class c1
  priority level 1
  police rate 640 kbps
 class c2
  bandwidth 20 mbps
 class class-default                    <-- cd-c
  bandwidth 1 mbps
!
policy-map parent
 class class-default                    <-- cd-p
  shape average 35 mbps
  service-policy child
!
interface GigabitEthernet0/0/0/0.1
 service-policy output parent
!
interface GigabitEthernet0/0/0/0.2
 service-policy output parent
```

Inactive entity

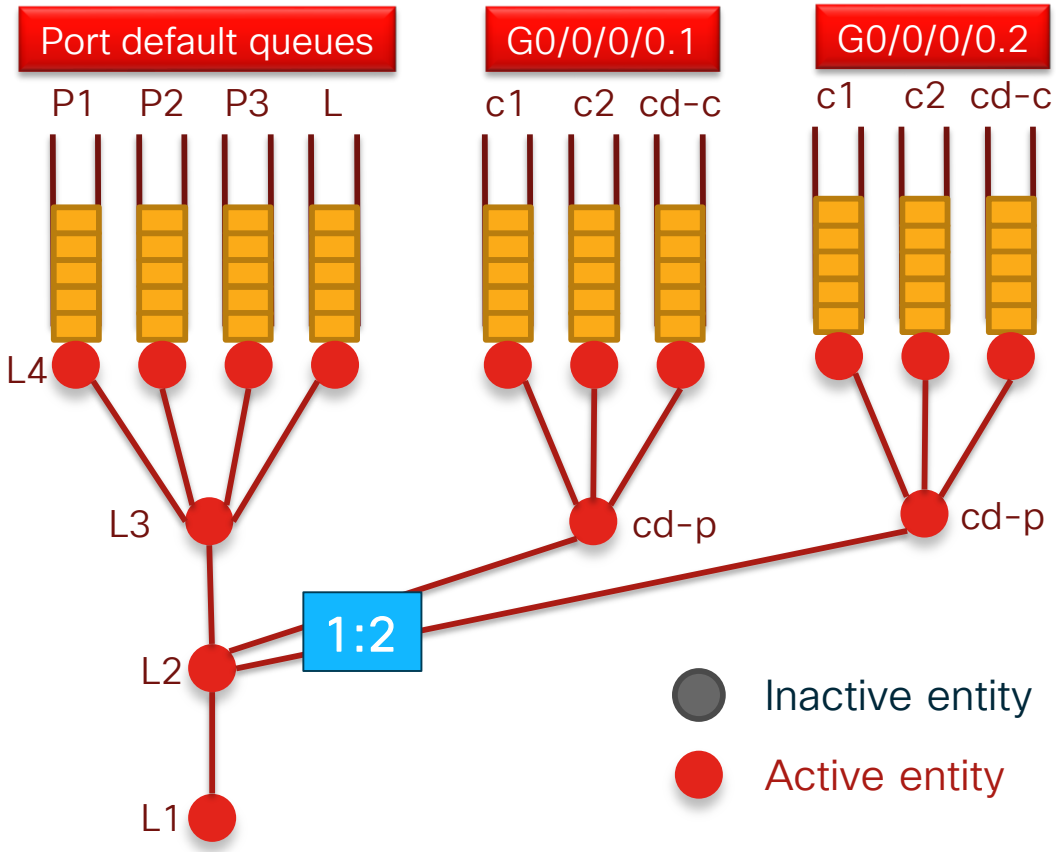
Active entity

MQC Hierarchy in Queuing ASIC




```
policy-map child
 class c1
  priority level 1
  police rate 640 kbps
 class c2
  bandwidth 20 mbps
 class class-default                    <- cd-c
  bandwidth 1 mbps
!
policy-map parent
 class c3
  shape average 35 mbps
  service-policy child
 class class-default                    <- cd-p
  bandwidth 1 mbps
!
policy-map grand-parent                 <- cd-gp
 class class-default
  shape average 35 mbps
  service-policy parent
!
interface GigabitEthernet0/0/0/0.1
 service-policy output grand-parent
```

MQC Hierarchy in Queuing ASIC




```
policy-map child
 class c1
  priority level 1
  police rate 640 kbps
 class c2
  bandwidth 20 mbps
 class class-default                    <-- cd-c
  bandwidth 1 mbps
!
policy-map parent
 class class-default                    <-- cd-p
  bandwidth remaining percent 20
  service-policy child

policy-map parent2
 class class-default
 bandwidth remaining percent 40
 service-policy child

!
interface GigabitEthernet0/0/0/0.1
 service-policy output parent
!
interface GigabitEthernet0/0/0/0.2
 service-policy output parent2
```

Default Interface Queues

Old format was:
show qoshal default-queue subslot 1 port 0 location 0/0/CPU0

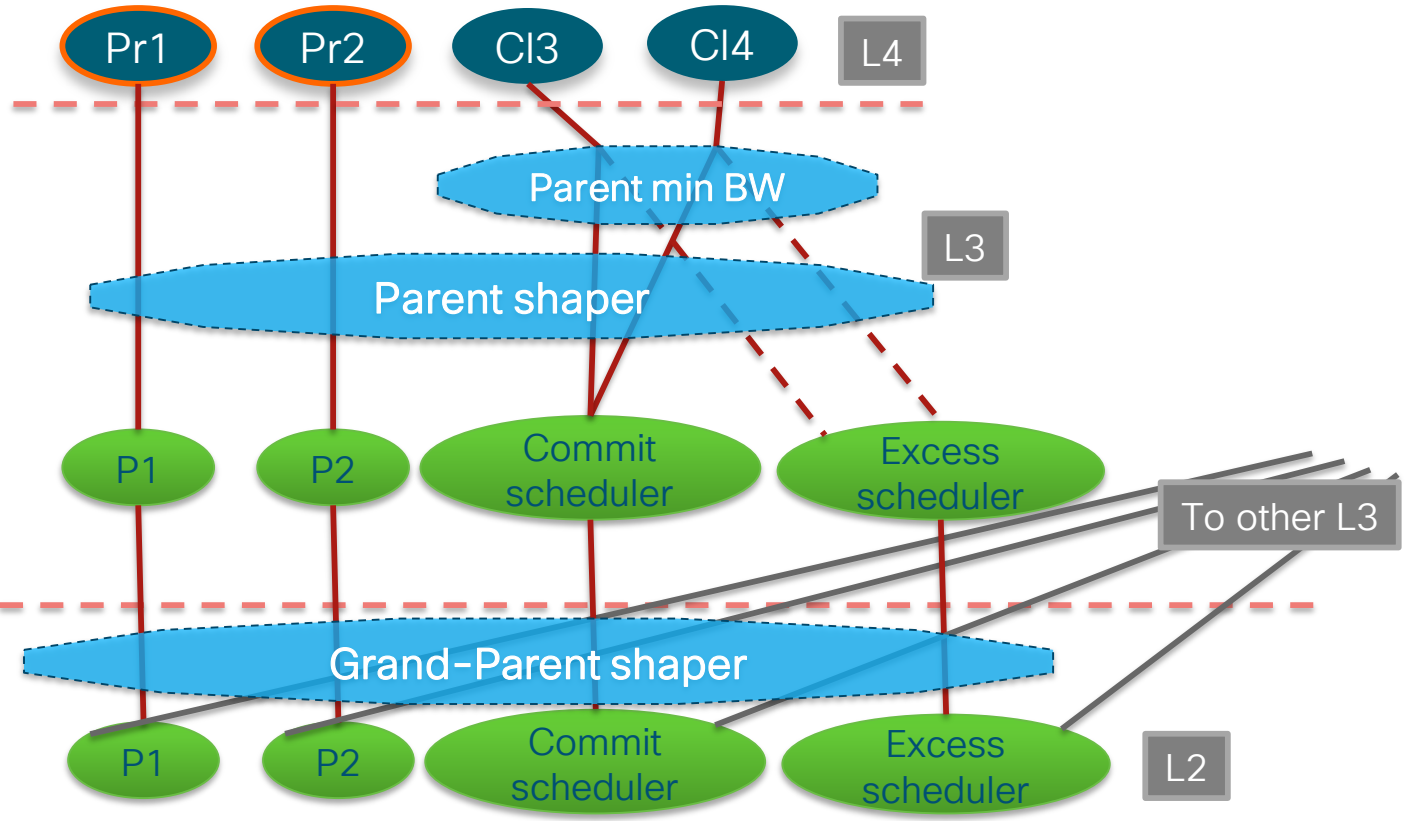
```
RP/0/RSP0/CPU0:A9K#show qoshal default-queue interface Gig0/0/1/0
Thu Apr  3 06:00:08.931 UTC
Interface Default Queues : Subslot 1, Port 0
============================================================
Port 64 NP 1 TM Port 16
   Ingress: QID 0x20000 Entity: 1/0/2/4/0/0 Priority: Priority 1 Qdepth: 0
         StatIDs: commit/fast_commit/drop: 0x690000/0x660/0x690001
         Statistics(Pkts/Bytes):
           Tx_To_TM  0/0 Fast TX: 425087/91780021
           Total Xmt 425087/91780021 Dropped 0/0

   Ingress: QID 0x20001 Entity: 1/0/2/4/0/1 Priority: Priority 2 Qdepth: 0
         <...>
   <...>
   Egress: QID 0x20020 Entity: 1/0/2/4/4/0 Priority: Priority 1 Qdepth: 0
         StatIDs: commit/fast_commit/drop: 0x6900a0/0x663/0x6900a1
         Statistics(Pkts/Bytes):
           Tx_To_TM  0/0 Fast TX: 412811/68090497
           Total Xmt 412811/68090497 Dropped 0/0

   Egress: QID 0x20021 Entity: 1/0/2/4/4/1 Priority: Priority 2 Qdepth: 0
         <...>
    <...>
```

current number of packets in the queue

TX statistics

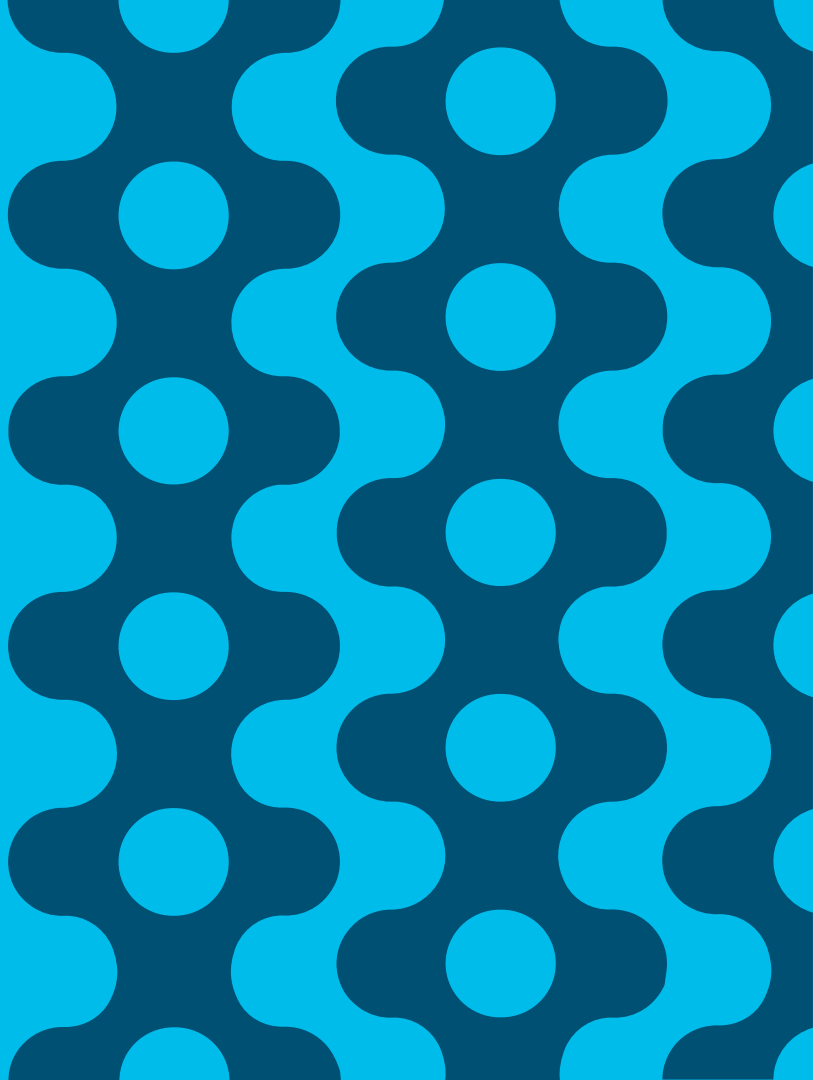
NP/TM/Chunk/Level/Index/Offset

Priority Propagation



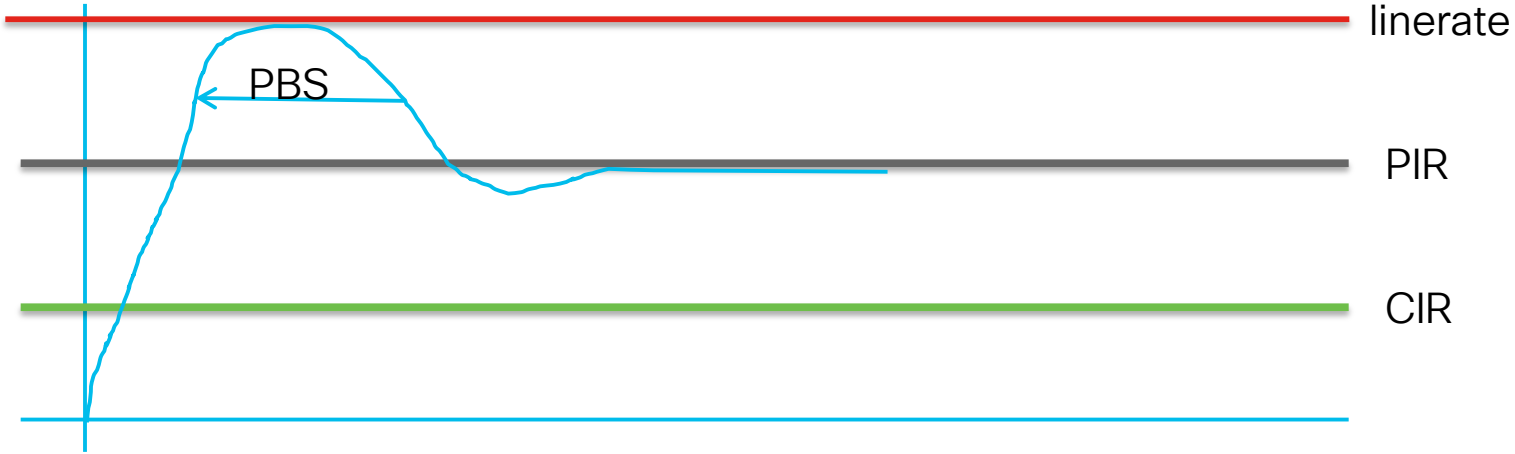

```
policy-map child
 class Pr1
  police rate 64 kbps
  priority level 1
 class Pr2
  police rate 10 mbps
  priority level 2
 class Cl3
  bandwidth 3 mbps
class Cl4
  bandwidth 1 mbps
!
policy-map parent
 class parent1
  shape average 25 mbps
  service-policy child
 class parent2
  shape average 25 mbps
  service-policy child
!
policy-map grand-parent
 class class-default
 shape average 500 mbps
 service-policy parent
```

QOS *Tokens buckets policers bursts*

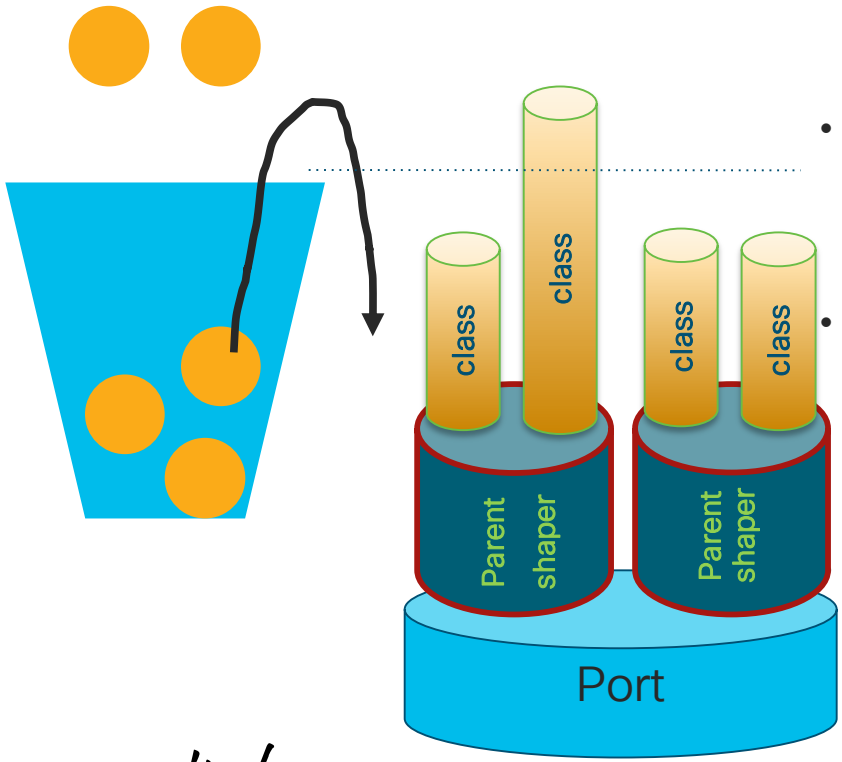


Shaping with PIR/PBS and CIR

- 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!



Shaping bucket



- Every Tc, a share of the shape rate is added as tokens to the bucket. (CIR/Tc tokens).
- When a class is assigned a “release packet” due to its bw ratio, then at transmission a number of tokens is taken out
- If the shaper is reaching max rate, it doesn’t dequeue a packet from the child class and therefore will start to buffer in the child class:
- If Bucket is empty (and no burst available) packet is buffered. This is called q-depth. The larger the q-depth, the more packet buffers used and (the longer the “yellow tail”).

(NOTE: the parent shaper doesn’t buffer (it just says, don’t dequeue yet to the child class).

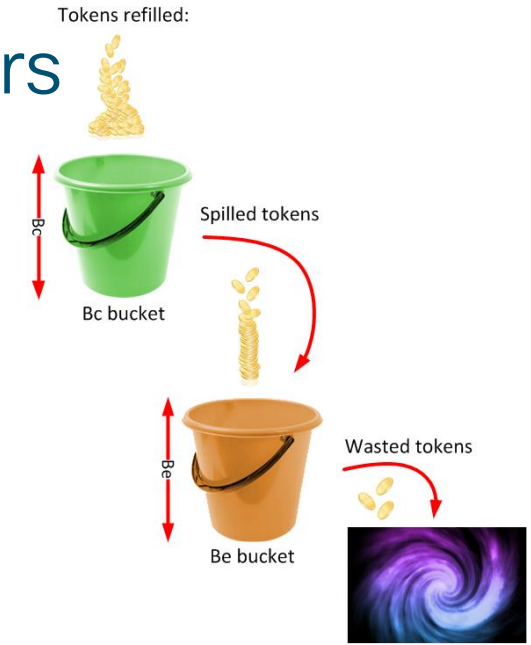
Policers

conform

exceed

violate

Single rate/3color



Dual rate/3color

police rate:

100 kbps **burst 10 bytes peak-burst 5 bytes**

police **rate 100 kbps peak-rate 200 kbps**

Conform Aware Policer

```
policy-map child
 class c1
  police rate 20 mbps peak-rate 50 mbps
 class c2
  police rate 30 mbps peak-rate 60 mbps
!
policy-map parent
class class-default
  service-policy child
  police rate 60 mbps
    child-conform-aware
```

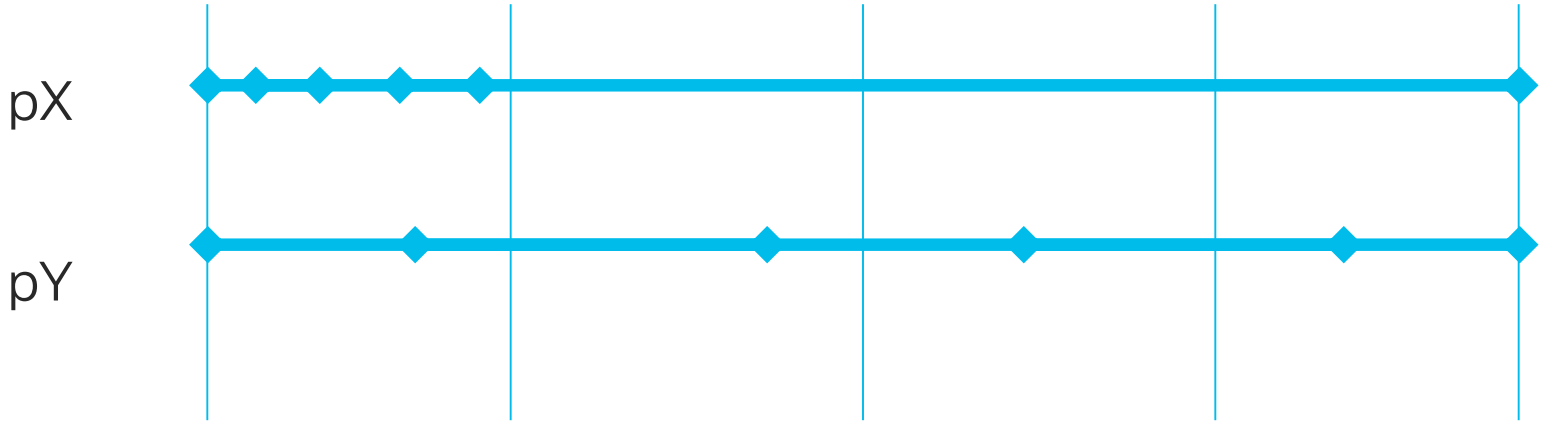
Parent CIR > Aggregated child CIR

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

(admin-config)#

On the topic of Token Refresh

- If the rate is 5 ps. I can send 5 at T0ms and stay silent until T1s. This is the same as sending 1 at times T0, T200msec, T400msec….



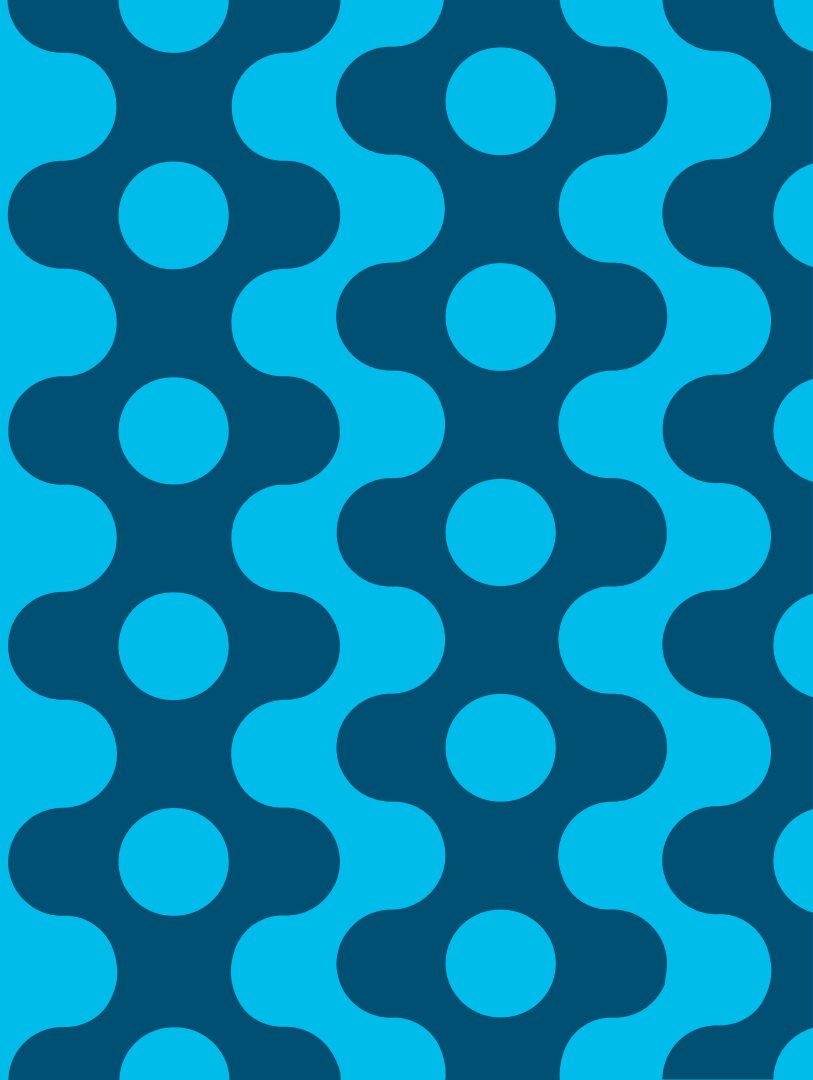
If token refreshing 4 times a second, I would add 1.25 tokens per refresh. Pattern X would call 4 packets exceed

Since A9K has a fixed token refresh time regardless of rate, it could exhibit transmission of pX.

A receiver may not tolerate that burst. Solution lowburst mode on a9k for frequent token refresh.

admin-config# *hw-module location qos-mode lowburst-enabled*

QoS:
Queue Limits/Occupancy
Buffer Limits/Occupancy

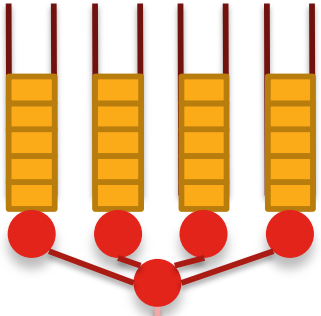


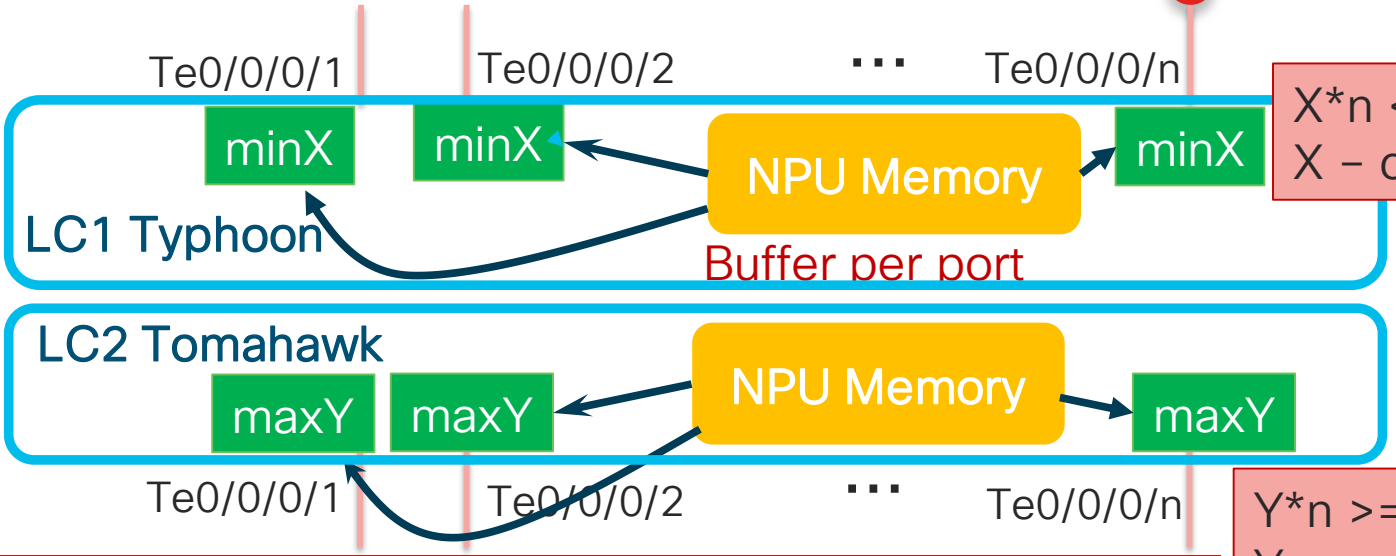
Queue Accounting Vs. Port Accounting

- Queue occupancy is accounted in units of 64 bytes
  - "Inst-queue-len (Packets)" field in "show policy-map interface"
  - 2 packets of 80 bytes each - Inst-queue-len shows 3 (3 x 64 = 192 bytes).
  - Compared against queue-limit and random-detect configured on the policy-map
  - Starting with 5.1.1 (5.3.0 for TH) can be calculated in numbers of 256-byte
    - hw-module all qos-mode wred-buffer-mode

- Port-level occupancy is measured in number of 256 byte buffers
  - User can over-subscribe the queue-limits on individual queues.
  - Limit exists on the number of buffers per port, per direction.
  - The limit itself is configured as linear curve with Min & Max:

Buffer Assignment Per Port





X*n < NPU mem
X – can be oversubscribed

Y*n >= NPU mem
Y – can't be oversubscribed

From 6.2.2 can double on TH card:
hw-module location <> qos-mode port-limit-oversubscribe

Port Buffer Occupancy

From "show qoshal default-queue interface Gig0/0/1/0"

```
RP/0/RSP0/CPU0:A9K#sh qoshal entity np 1 tm 0 chunk 2 level 4 index 4 0 hierarchy loc 0/0/CPU0
<...>
Np: 1 TM: 0 Chunk: 0 Level: 0 Index: 1 Offset 0
<...>
   Egress: Entity: 1/0/0/0/11/0 Qdepth: 0
   <...>
   [D] WRED template - ID: 2 Curve: 0 Min/Mid/Max 657/657/851 MidDrop/MaxDrop 0/102
   [D] WRED scale - ID: 3 Value: 132096
   [D] WRED template - ID: 2 Curve: 1 Min/Mid/Max 657/657/851 MidDrop/MaxDrop 0/102
   [D] WRED scale - ID: 3 Value: 132096
   [D] WRED template - ID: 2 Curve: 2 Min/Mid/Max 657/657/851 MidDrop/MaxDrop 0/102
   [D] WRED scale - ID: 3 Value: 132096
   [D] WRED template - ID: 2 Curve: 3 Min/Mid/Max 657/657/851 MidDrop/MaxDrop 0/102
   [D] WRED scale - ID: 3 Value: 132096
```

Current buffer occupancy in numbers of 256-byte particles

Buffer limit in numbers of 256-byte particles

PPM of WRED Scale

- Two curves used by default:
  - One for normal priority, and one for high priority (levels 1, 2 and 3).
- Per-Priority Buffer-Limits mode is available in XR Release 5.1.1 → all 4 curves in use

  hw-module all qos-mode per-priority-buffer-limit

  show qoshal qos-mode location <loc>

MQC – Queue Occupancy

```
RP/0/RSP0/CPU0:av-asr9001#sh policy-map interface g0/0/1/0 output

GigabitEthernet0/0/1/0 output: core-parent

Class af21
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :              9443/9480772              0
    Transmitted         :              4646/4664584              0
    Total Dropped       :              2397/2406588              0
    Policy core-child Class TGN
      Classification statistics        (packets/bytes)     (rate - kbps)
        Matched             :              9443/9480772              0
        Transmitted         :              4646/4664584              0
        Total Dropped       :              2397/2406588              0
      Queueing statistics
        Queue ID                         : 131714
        High watermark                   : N/A
        Inst-queue-len  (packets)        : 1814
        Avg-queue-len                    : N/A
        Taildropped(packets/bytes)       : 2397/2406588
        Queue(conform)      :            4646/4664584              0
        Queue(exceed)       :              0/0                     0
        RED random drops(packets/bytes)      : 0/0
<..>
```

```
policy-map core-child
 class TGN
  bandwidth percent 10
 !
 class class-default
  bandwidth percent 1
 !
policy-map core-parent
 class af21
  service-policy core-child
  shape average 10 mbps
 !
 class class-default
  shape average 100 mbps
```

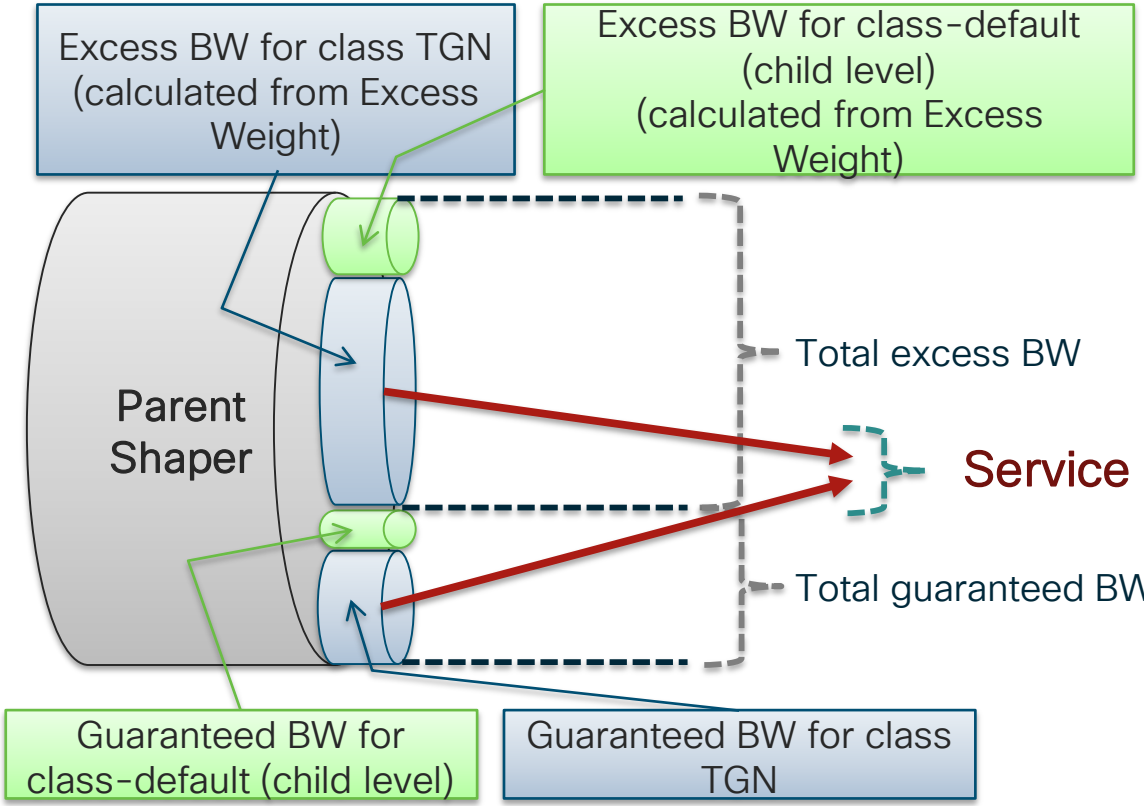
Current queue occupancy in 64-byte units

MQC – What Is Programmed In Hardware

```
RP/0/RSP0/CPU0:av-asr9001#sh qos interface Gig0/0/1/0 output
Interface: GigabitEthernet0_0_1_0 output
Bandwidth configured: 1000000 kbps Bandwidth programed: 1000000 kbps
ANCP user configured: 0 kbps ANCP programed in HW: 0 kbps
Port Shaper programed in HW: 0 kbps
Policy: core-parent Total number of classes: 4
------------------------------------------------------------------------
Level: 0 Policy: core-parent Class: af21
QueueID: N/A
Shape CIR : NONE
Shape PIR Profile : 2/3(S) Scale: 156 PIR: 9984 kbps  PBS: 124800 bytes
WFQ Profile: 2/9 Committed Weight: 10 Excess Weight: 10
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 1
------------------------------------------------------------------------
Level: 1 Policy: core-child Class: TGN
Parent Policy: core-parent Class: af21
QueueID: 131746 (Priority Normal)
Queue Limit: 114 kbytes Abs-Index: 27 Template: 0 Curve: 0
Shape CIR Profile: INVALID
WFQ Profile: 2/76 Committed Weight: 91 Excess Weight: 91
Bandwidth: 1000 kbps, BW sum for Level 1: 1100 kbps, Excess Ratio: 1
------------------------------------------------------------------------
Level: 1 Policy: core-child Class: class-default
Parent Policy: core-parent Class: af21
QueueID: 131747 (Priority Normal)
Queue Limit: 12 kbytes Abs-Index: 7 Template: 0 Curve: 0
Shape CIR Profile: INVALID
WFQ Profile: 2/8 Committed Weight: 9 Excess Weight: 9
Bandwidth: 100 kbps, BW sum for Level 1: 1100 kbps, Excess Ratio: 1
```

```
policy-map core-child
 class TGN
  bandwidth percent 10
 !
 class class-default
  bandwidth percent 1
 !
policy-map core-parent
 class af21
  service-policy core-child
  shape average 10 mbps
 !
 class class-default
  shape average 100 mbps
```

Service Rate




```
policy-map core-child
 class TGN
  bandwidth percent 10
 !
 class class-default
  bandwidth percent 1
 !
policy-map core-parent
 class af21
  service-policy core-child
  shape average 10 mbps
 !
 class class-default
  shape average 100 mbps
```

Queue Limit and Service Rate

- Queue Limit is by default 100ms worth of Service Rate
- Service Rate is the sum of minimum guaranteed bandwidth and bandwidth remaining assigned to a given class
- Bandwidth remaining = Parent Max Rate – Sum of Guaranteed BW at child
  - Parent Max Rate is typically the parent shaper rate or physical interface BW (no PIR)
- Bandwidth remaining assigned to a given class = BR * Excess Weight / Sum of Excess Weights at child level

$$\text{Service Rate} = \text{Guaranteed BW} + \frac{(\text{Parent Max Rate} - \sum \text{Guaranteed BW}) * \text{Excess Weight}}{\sum \text{Excess Weights}}$$

Committed & Excess Weight

- Configured BW allocation is always translated into a WFQ weight
  - Bandwidth Remaining Ratio is directly translated into WFQ weight
  - Bandwidth Percentage is translated as percentage of 1024
- Same value is used for Committed Weight and Excess Weight
  - → excess bandwidth is distributed among classes in the same ratio as the guaranteed bandwidth
- Committed Weight: within parent bandwidth
- Excess Weight: exceeding parent bandwidth, but within parent shape
- Values are pre-fit; the one closest to user configured ratio is picked

Queue Limit Calculation

```
Level: 0 Policy: core-parent Class: af21
Shape PIR Profile : 2/3(S) Scale: 156 PIR: 9984 kbps  PBS: 124800 bytes
------------------------------------------------------------------------
Level: 1 Policy: core-child Class: TGN
Queue Limit: 114 kbytes Abs-Index: 27 Template: 0 Curve: 0
WFQ Profile: 2/76 Committed Weight: 91 Excess Weight: 91
Bandwidth: 1000 kbps  BW sum for Level 1: 1100 kbps  Excess Ratio: 1
------------------------------------------------------------------------
Level: 1 Policy: core-child Class: class-default
WFQ Profile: 2/8 Committed Weight: 9 Excess Weight: 9
```

```
policy-map core-child
 class TGN
  bandwidth percent 10
 !
 class class-default
  bandwidth percent 1
 !
policy-map core-parent
 class af21
  service-policy core-child
  shape average 10 mbps
 !
 class class-default
  shape average 100 mbps
```

- **Queue Limit is 100ms worth of Service Rate**

  Queue Limit = Service Rate * 100ms

- **Service Rate is the sum of minimum guaranteed bandwidth and bandwidth remaining assigned to a given class**
  - Parent BW = 9984 kbps
  - Guaranteed BW of class TGN = 1000 kbps
  - Sum of guaranteed BW at Level1 = 1100 kbps
  - Total remaining BW at Level1 = 9984 – 1100 = 8884 kpbs
  - Remaining BW of class TGN = (91 / (91 + 9)) * 8884 kpbs = 8084 kbps
  - Service Rate of class TGN = 1000 + 8084 = 9084 kbps
  - Queue Limit of class TGN = 9084 / 10 = 908400 [bits] = 113550 [Bytes] ~= **114 kB**

Configuring The Queue Limit

- Queue Limit can be configured manually
- Supported units:
  - Bytes
  - Kilobytes
  - Megabytes
  - Milliseconds
  - Packets (default)
    - For conversion into 64-Byte units of queue occupancy, packet size of 256 bytes is presumed
  - Microseconds
- Queue Limit sizes are pre-fit
  - The one closest to the calculated queue limit is picked for HW programming

Configuring The Queue Limit

```
RP/0/RSP0/CPU0:av-asr9001#sh qos interface Gig0/0/1/0 output location 0/0/CPU0
<. . .>
Level: 1 Policy: core-child Class: TGN
Parent Policy: core-parent Class: af21
Queue Limit: 2624 kbytes (10000 packets) Abs-Index: 90 Template: 0 Curve: 0
WFQ Profile: 2/76 Committed Weight: 91 Excess Weight: 91
Bandwidth: 1000 kbps, BW sum for Level 1: 1100 kbps, Excess Ratio: 1
```

- Presuming 256 Byte packet size
- Closest pre-fit value picked

```
<..>
policy-map core-child
 class TGN
  bandwidth percent 10
  queue-limit 10000 packets
 !
<..>
```

```
RP/0/RSP0/CPU0:av-asr9001#sh qos interface Gig0/0/1/0 output location 0/0/CPU0
<. . .>
Level: 1 Policy: core-child Class: TGN
Parent Policy: core-parent Class: af21
Queue Limit: 1120 kbytes (1000 ms) Abs-Index: 90 Template: 0 Curve: 0
WFQ Profile: 2/76 Committed Weight: 91 Excess Weight: 91
Bandwidth: 1000 kbps, BW sum for Level 1: 1100 kbps, Excess Ratio: 1
```

- 1000ms worth of Service Rate
  - 9084 [kbps] / 8 [bit/Byte] = 1135500 Bytes = 1135.5 kB
- Closest pre-fit value picked

```
<..>
policy-map core-child
 class TGN
  bandwidth percent 10
  queue-limit 1000 ms
 !
<..>
```

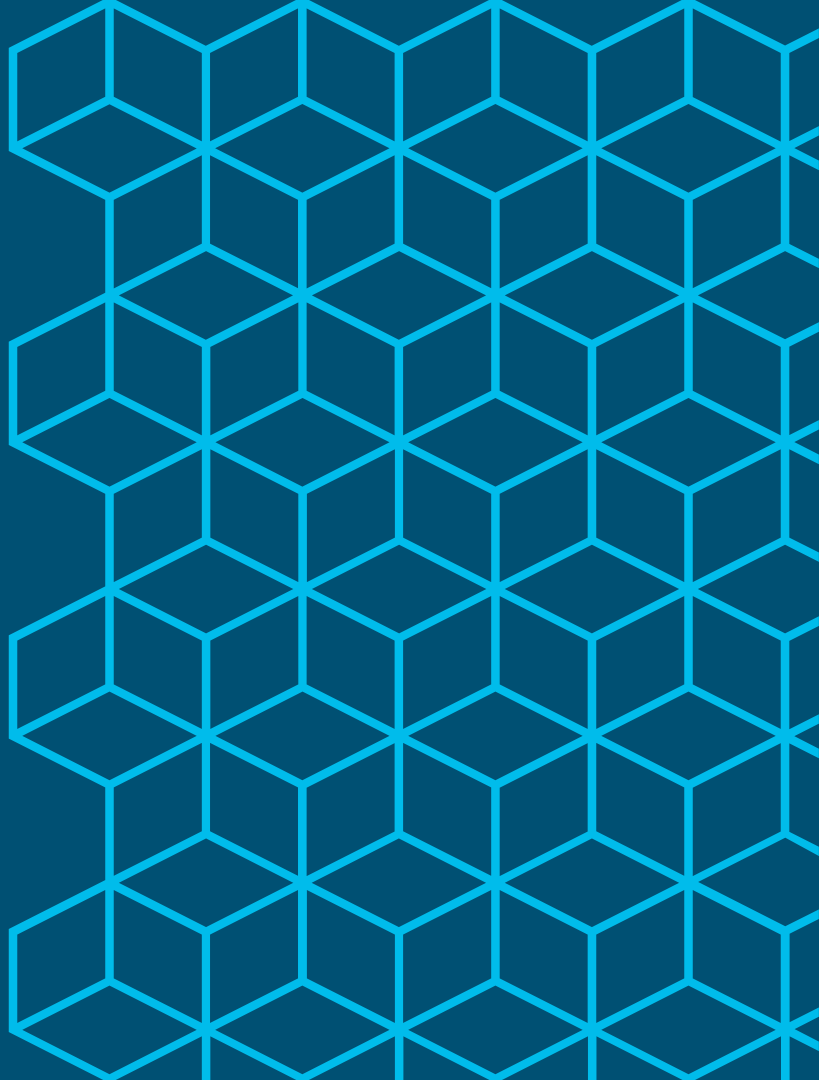
QoS Classification Formats

- A given QoS policy-map generally classifies based on a single classification format
- IPv4 and IPv6 classes can co-exist in the same policy

	Format 0	Format 1	Format 2	Format 3	Format 4
Fields supported	- IPV4 source address (Specific/Range)[1] - IPV4 Destination address (Specific/Range) - IPV4 protocol - IPV4 TTL - IPV4 Source port (Specific/Range) - IPV4 Destination port (Specific/Range) - TCP Flags - IP DSCP / TOS / Precedence - QOS-group (output policy only) - Discard-class (output-policy only) - EXP	- 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) - EXP	- 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) - EXP	- Outer VLAN/COS/DEI - Inner VLAN/COS - MAC Destination address - MAC source address - QOS-group (output policy only) - Discard-class (output policy only)	- IPV6 source address (Specific/Range) - IPV6 Destination address (Specific/Range) - IPV6 protocol - IPV6 TOS /EXP - IPV6 TTL - IPV6 Source port (Specific/Range) - IPV6 Destination port (Specific/Range) - TCP Flags - Outer VLAN/COS/DEI - Inner VLAN/COS - IPV6 header flags - QOS-group (output policy only) - Discard-class (output-policy only)

[1] All fields marked in green are defined using an ACL used for QOS classification.

Cisco Software Manager → CSM

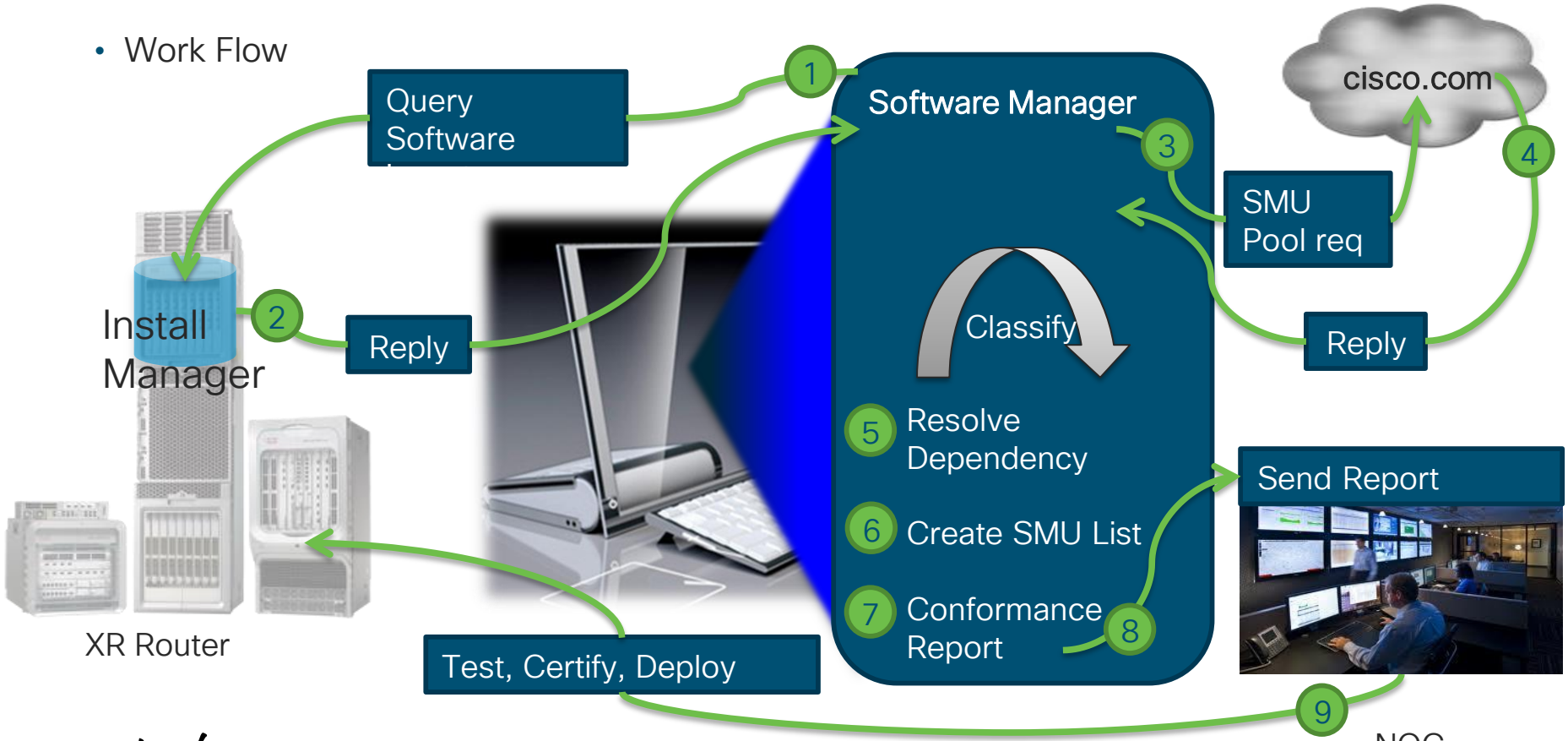


CSM Flavours

- CSM 2.0 (obsolete)
  - Standalone Java application for basic SMU management
- CSM Application (v3.6 – coming July)
  - Runs on Linux
  - MS Windows, MacOS support through OVA
  - Optimises SMU list per platform/release
- CSM Server (v3+)
  - Open source distribution
  - Runs on Linux

XR Software Manager Server

- Work Flow



What CSM Does

- Software Management:
  - Automated and Simplified image (releases and SMUs) retrieval, reporting and alerts
  - Pushes image to one or many devices
  - End to end SW management
  - Patch recommendation, and conformance reporting
  - Migration from 32-bit XR to 64-bit XR
- Operations Simplification:
  - Auto-updates: schedule installation, pre- and post- verifications
  - Easier access to image and patch details (documentation)
  - Multi-platform and multi-OS support
- Inventory Management:
  - Visibility into hardware, cards, slots, S/N, optic types

Solves For:

- Time consuming, manual, laborious, repetitive, error-prone SW installation
- Complicated patch dependencies
- High costs

Big Wins:

- Huge time and resource savings
- Up to 90% time savings on SW upgrades

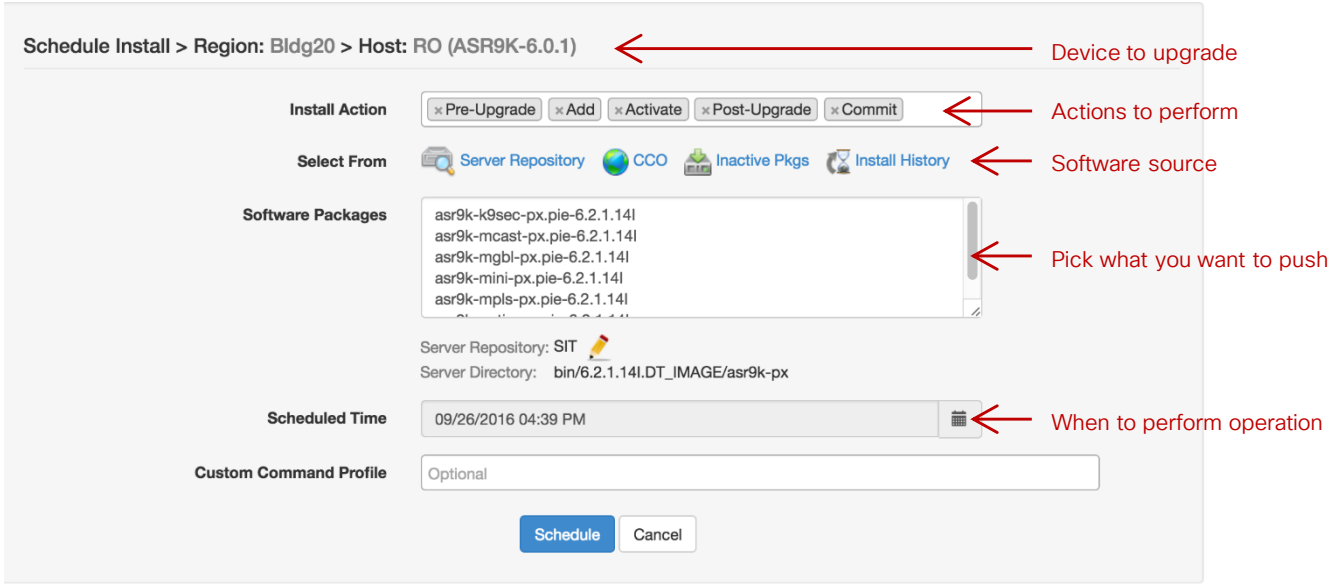
Management Aspects

- Manages thousands of devices by regions
  - Detects supported platforms /Daily software inventory retrieval
- Manages TFTP/FTP/SFTP server repositories
  - Browses/downloads/uploads software packages
- Manages users with different security privileges
  - Authenticates through CSM database or LDAP server
- Inventory manager
  - Collecting all HW details of devices, enforce FN, check underutilized HW
- Programmatic Interfaces
  - Supports RESTful APIs

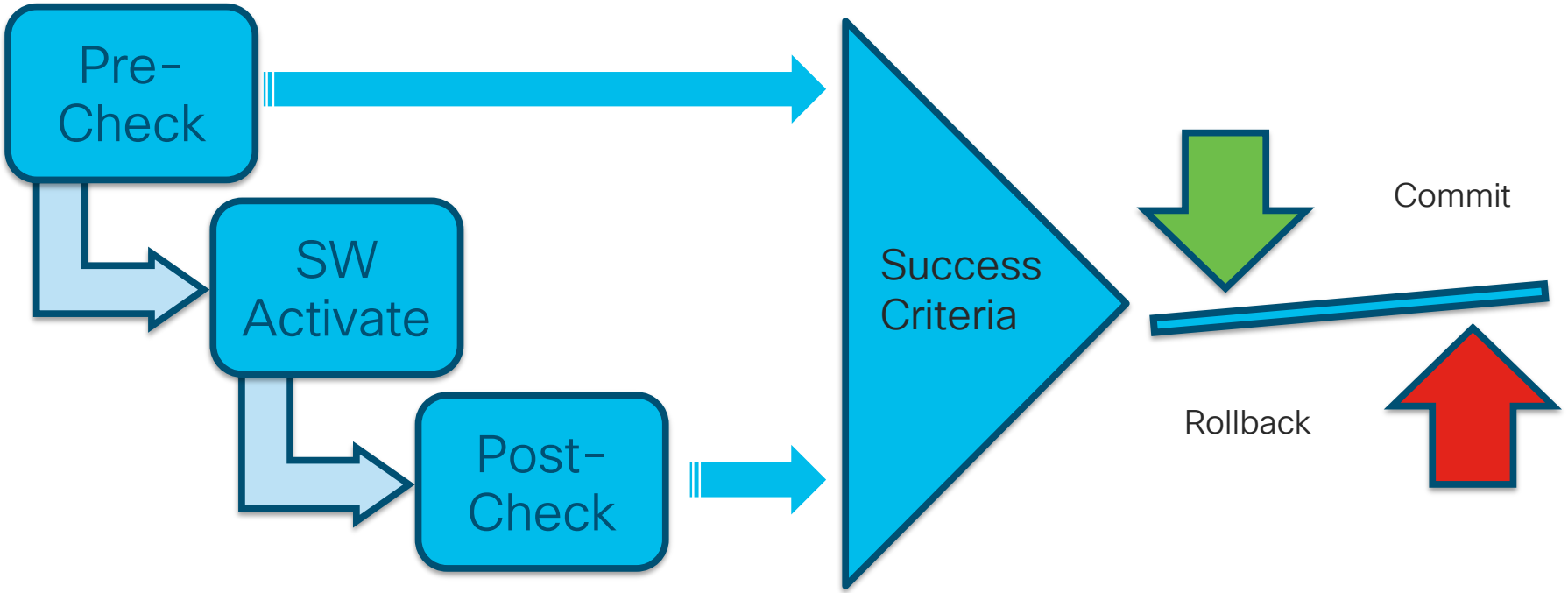
Supported Platforms (CSM v3.5)

- IOS XR 32-bit (ASR9K, CRS)
- IOS XR 64-bit (ASR9K-64, NCS1K, NCS5K, NCS5500, XRv9000)
- NG-XR (NCS6K)
- IOS-XE (ASR902, ASR903, ASR907, ASR920)
- IOS (ASR901)

CSM: How to Schedule a Release Installation



Software Upgrade Method of Procedure (MoP)



LDP Plugin Example (Post-Check)

Check if LDP is configured

Capture

Capture MPLS LDP Configuration
command: show running mpls ldp | i mpls ldp
store: **mpls_ldp_config**

Count

Count number of lines
Source: '**mpls_ldp_config**'

Notify

If counter = 0 then log "MPLS LDP not configured and finish
Else continue

Capture the neighbors

Capture

Capture MPLS LDP Neighbors' State
Command: show mpls ldp neigibor brief
Store: **post_ldp_neighbors_table**
Save as txt in the log dir

Count

Count the total number of LDP neighbors
Source: table: **post_ldp_neighbors_table**

Notify

If counter > 0 log "Total number of LDP neighbors"
Else warn there are no LDP neighbors

Compare Pre- and Post-results

Compare

Baseline: pre_ldp_neighbors_table
Sample: post_ldp_neighbors_table

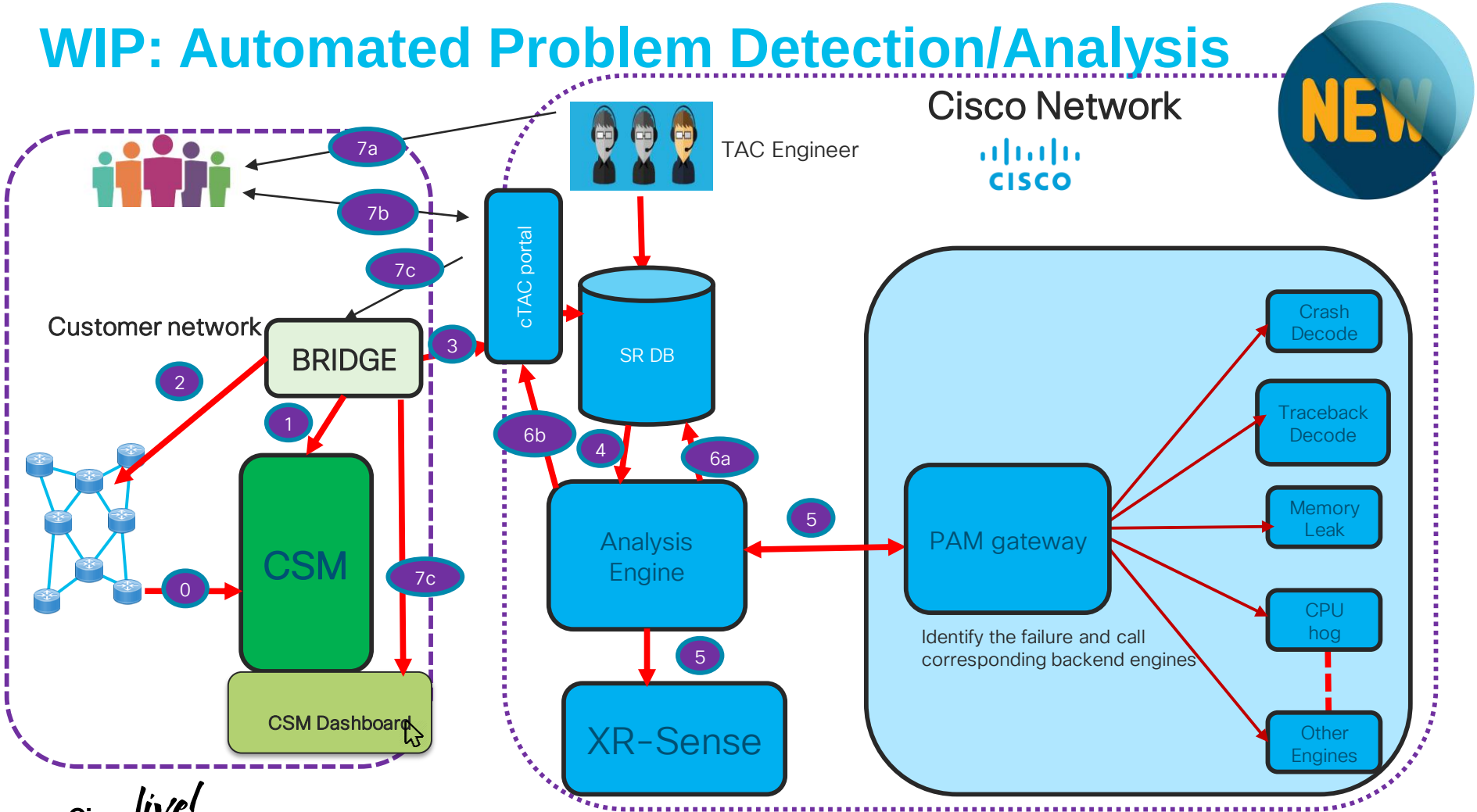
Notify

If missing neighbors in sample (post check) log warning
Log list of missing neighbors

Notify

If extra neighbors in sample (post check) log warning
Log list of extra neighbors

WIP: Automated Problem Detection/Analysis



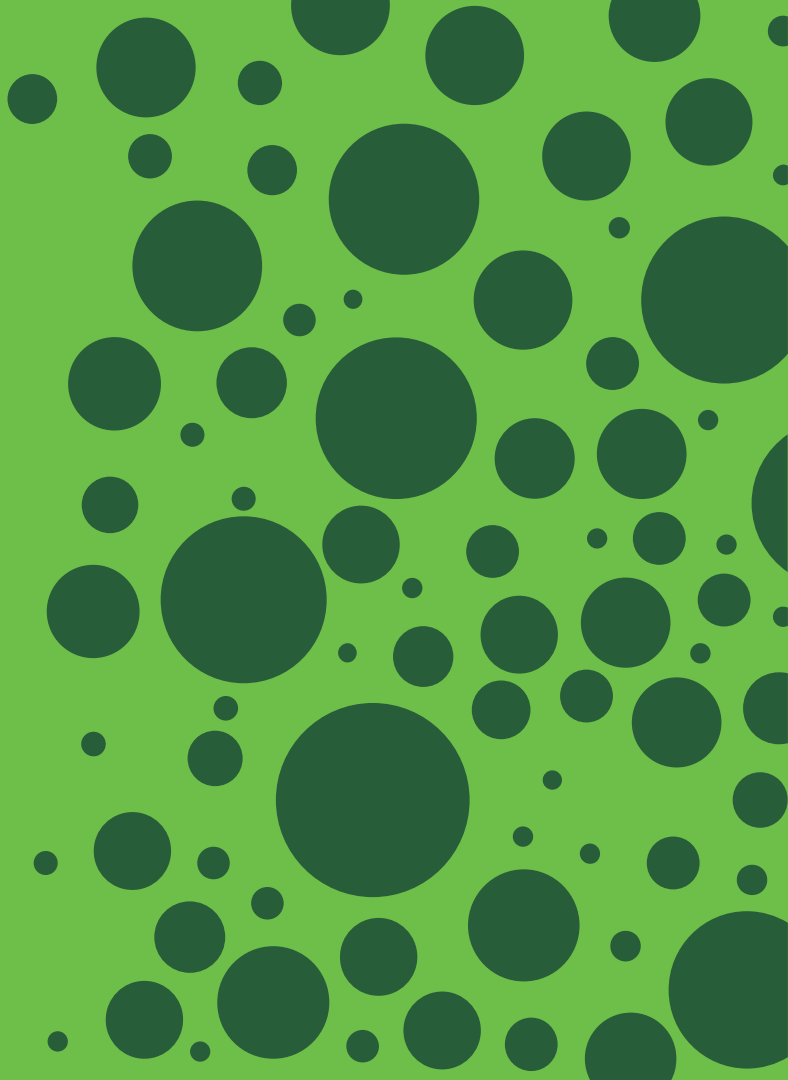
CSM is an official Cisco Tool

How to download	Download CSM: https://software.cisco.com/download/release.html?mdfid=282423206&softwareid=284777134&release=3.5&relind=AVAILABLE&rellifecycle=&reltype=latest
How to use	**CSM Server Documentation:** https://supportforums.cisco.com/document/13154846/cisco-software-manager-33-overview-documentation **CSM Server Videos:** Introduction to CSM Server: https://youtu.be/isxN08x-mr4 Getting Started with CSM Server: https://www.youtube.com/watch?v=omdpr3uP_b4 ASR9K IOS XR 32 bit to 64 bit Migration using CSM Server: https://youtu.be/RVgR0TdbpVw **CSM Application Video:** https://www.youtube.com/watch?v=PYO2Om-nUKQ
Support forum	https://supportforums.cisco.com/community/5996/xr-os-and-platforms

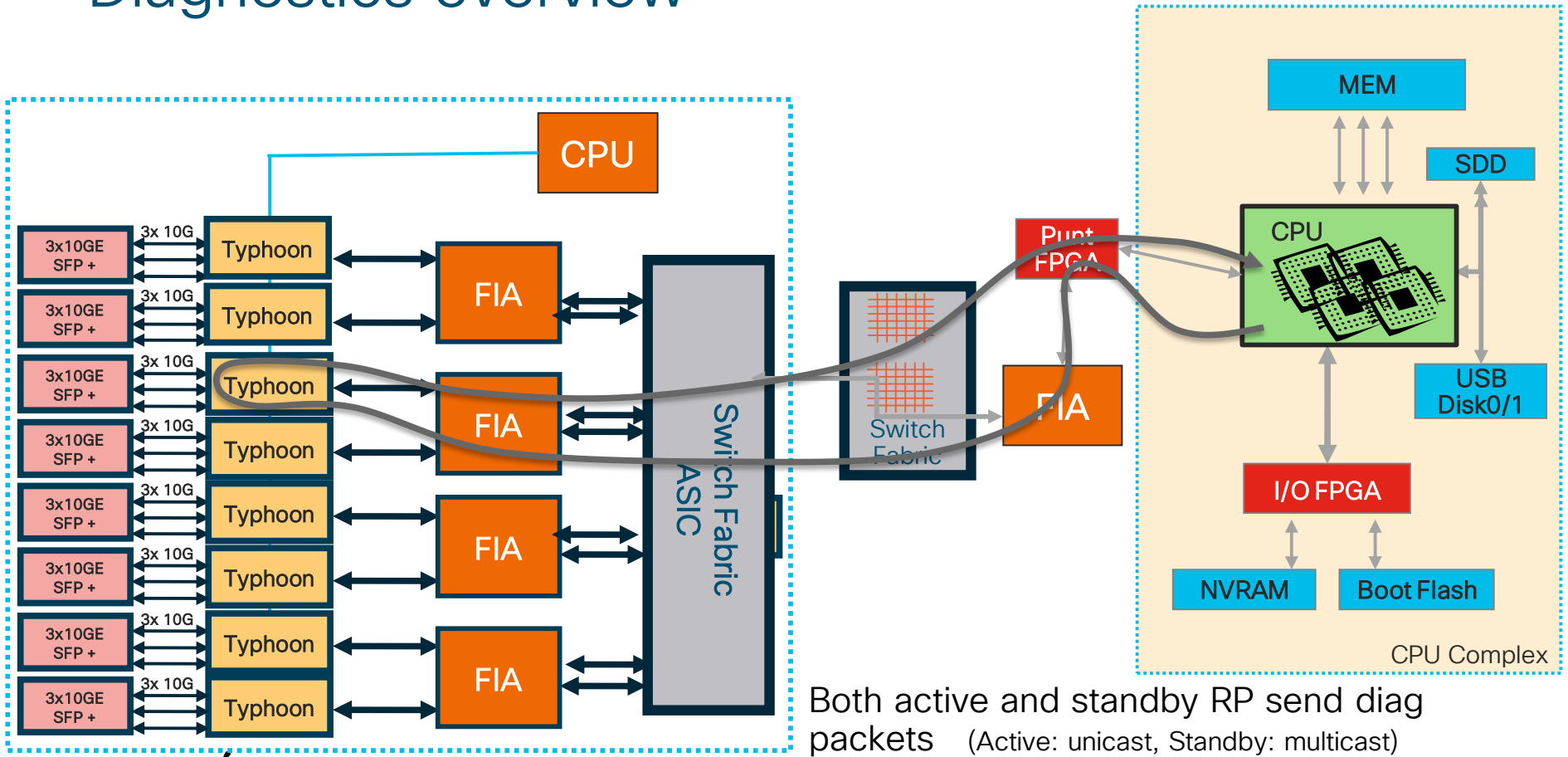
Supported on:

- CRS
- NCS (1K, 5K, 6K)
- 9XX Series
- ASR 9000
- More coming soon!

Punt Fabric Data Path Diagnostic



Diagnostics overview



Both active and standby RP send diag packets (Active: unicast, Standby: multicast)

Interpreting the message

%PLATFORM-DIAGS-3-PUNT_FABRIC_DATA_PATH_FAILED :
Set|online_diag_rsp[*237686*]|System Punt/Fabric/data Path
Test(*0x2000004*)|failure threshold is 3, (slot, NP) failed: (0/2/CPU0, 0)

- Syslog message formatted
- Set or Clear
- Specific test case failure
- *processID and test ID (not important)*
- Physical linecard reporting the error/NPU towards which the path is broken.
  - If a single NPU is listed: it may be an NP lockup
  - If multiple or all NPs are listed: troubleshoot the fabric path to the line card

```
RP/0/RSP0/CPU0:iox(admin)#show diagnostic content location 0/RSP0/CPU0
                                                    Test Interval Thre-
ID Test Name                          Attributes (day hh:mm:ss.ms shold)
==== ================================ ============ =====
11) PuntFabricDataPath ---------- *B*N****A 000 00:01:00.000 3
```

AutoShut

- Router can automatically shutdown all ports associated with the PUNT_FABRIC_DATA_PATH errors

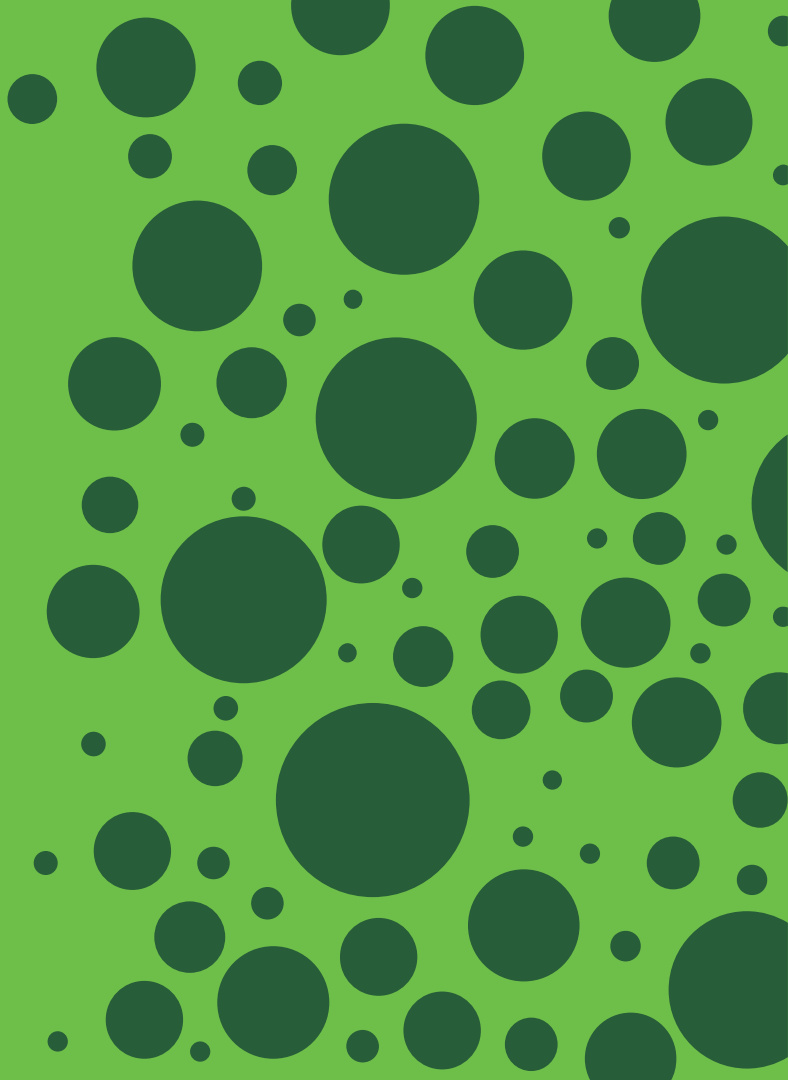
RP/0/RSP1/CPU0:COREBOT#admin show run
fault-manager datapath port shutdown <====== ADMIN config CLI

fialc(fia=oct fia#=2, slot=0)>RP/0/RSP1/CPU0:Feb 9 12:48:40.575 : pfm_node_rp[366]: %PLATFORM-DIAGS-3-PUNT_FABRIC_DATA_PATH_FAILED : Set|online_diag_rsp[204929]|System Punt/Fabric/data Path Test(0x2000004)|failure threshold is 3, (slot, NP) failed: (0/0/CPU0, 1)
LC/0/0/CPU0:Feb 9 12:48:40.666 : ifmgr[213]: %PKT_INFRA-LINEPROTO-5-UPDOWN : Line protocol on Interface HundredGigE0/0/0/0, changed state to Down

Punt Fabric Data Path Troubleshooting

- https://www.cisco.com/c/en/us/support/docs/routers/asr-9000-series-aggregation-services-routers/116727-troubleshoot-punt-00.html
- https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r4-0/troubleshooting/guide/tr40asr9kbook/tr40fab.pdf
- https://supportforums.cisco.com/t5/service-providers-documents/asr9000-xr-understanding-platform-diags-3-punt-fabric-data-path/ta-p/3134926

Feature Digest



Automatic Intf Shutdown on LC Insert

- Main principles:
  - Interface auto-shutdown is applied on interfaces without any configuration
  - Only explicit configuration is permanent
    - Otherwise the consistency of commit rollback may be disrupted
- Unwanted consequence best explained with this sequence:
  1. New LC is inserted
  2. Every interface on the LC is shutdown via auto-config
  3. Users apply some configuration to the interface (for example interface description)
  4. System or LC is reloaded
  5. Interface is NOT shutdown via auto-config because it has some configuration

Automatic Intf Shutdown on LC Insert Cont.

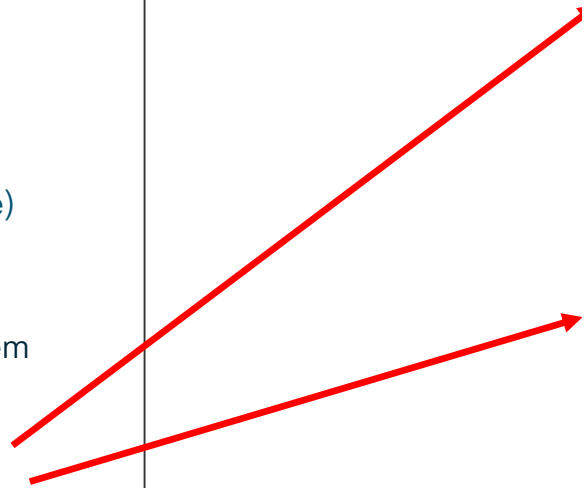
New Behaviour Starting With XR 6.5.1

- Main principles remain:
  - Interface auto-shutdown is applied on interfaces without any configuration
  - Only explicit configuration is permanent
    - Otherwise the consistency of commit rollback may be disrupted
- Auto-shutdown config is saved into a file
  - *<last_commit_ID+1>*.snd_shut_*<req-id>*_*<node-id>*
- Upon rollback this file is consulted and config is merged
  - ➔ consistent behaviour

DDTS Updates: Version Fidelity

What You Want To Know About a Bug?!

- Headline
- Release-note:
  - Symptom
  - Conditions
    - Conditions
    - Trigger
    - Impact
  - Workaround (preventive)
    - Preventive workaround
    - Recovery steps
  - Optional: Further Problem Description
- Impacted releases
- Impacted platforms



Tools & Resources

Bug Search Tool

Known Affected Releases: *(1)*

6.3.2.MPLS

Known Fixed Releases: *(7)*

6.1.4

6.4.1.23i.BASE

6.3.15.1i.BASE

6.3.2.19i.BASE

6.2.3.8i.BASE

This is SMU

Download software for Cisco ASR 9000 Series Aggregation Services Routers

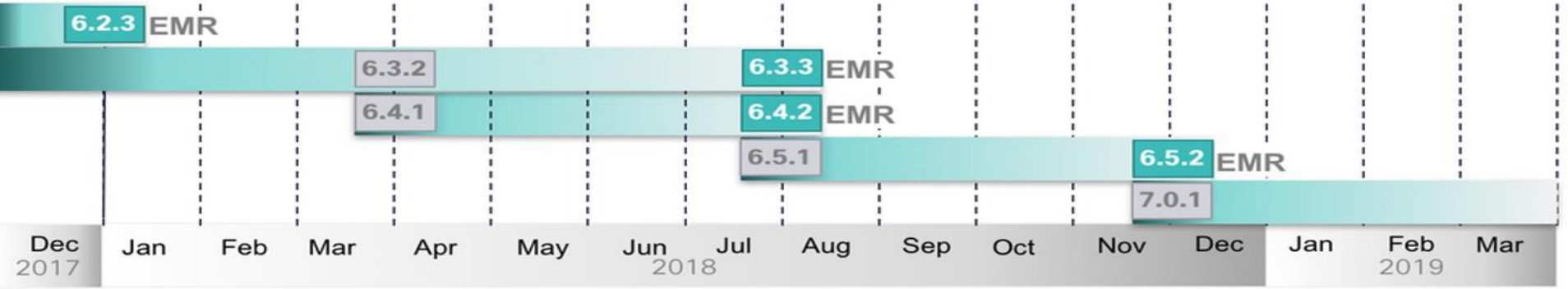
COMING SOON:
Potentially Affected

XR Release Selection

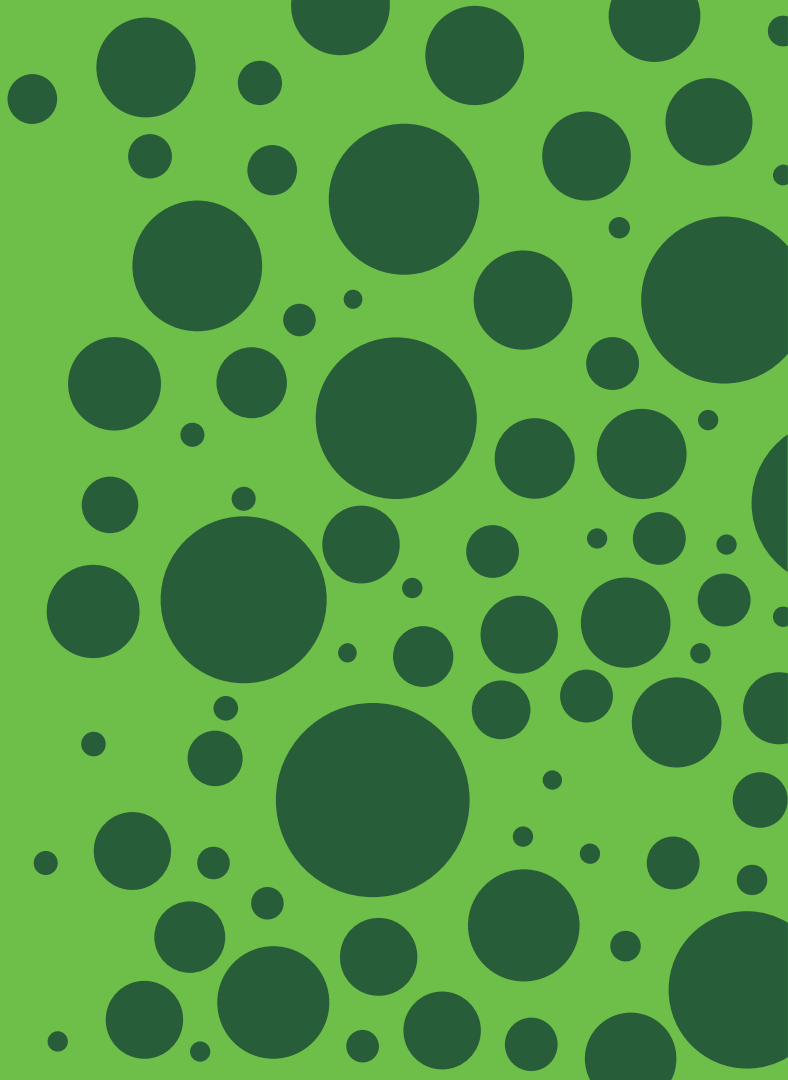
- Extended Maintenance Release (EMR) criteria:
  - No new features or new HW support
  - Incoming bug rate to drop for at least 90% from peak find rate for that release. Zero critical bugs.
- EMR target is x.x.3 (7 months after x.x.1); exceptionally x.x.4
  - EMR evaluation two months before FCS.
- Release recommendations for year 2018:
  - **ASR9000, CRS, NCS5000, NCS5500, NCS6000, XRv9000**: 6.1.4, 6.2.3, **EFT: 6.3.3, 6.4.2**
  - **ASR9000** with Trident generation HW: 5.3.4; See the related EoS-EoL notice.
- IOS XR EMR References:
  - https://supportforums.cisco.com/document/13212901/ios-xr-release-strategy-and-deployment-recommendation
  - http://www.cisco.com/en/US/prod/collateral/iosswrel/ps8803/ps5845/product_bulletin_c25-478699.html
  - Star icon  on ”Download Software” and “Software Research” portals on cisco.com
  - Search for “EMR” on https://supportforums.cisco.com/community/netpro/service-providers/ios-xr

XR Release Selection

	6.3.x	6.4.x	6.5.x
ASR9000	Yes	Yes	Yes
CRS	No	Yes	No
NCS5000	Yes	Yes (satellite only)	Yes
NCS5500	Yes	No	Yes
NCS6000	Yes	Yes	Yes
XRv9000	Yes	Yes	Yes



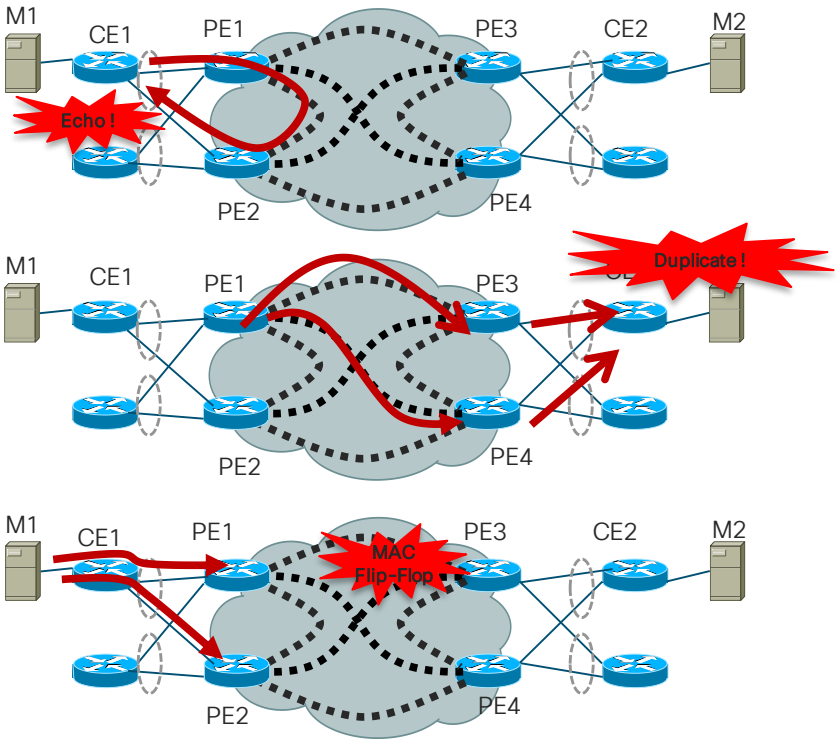
EVPN



Why was EVPN needed in 2012?

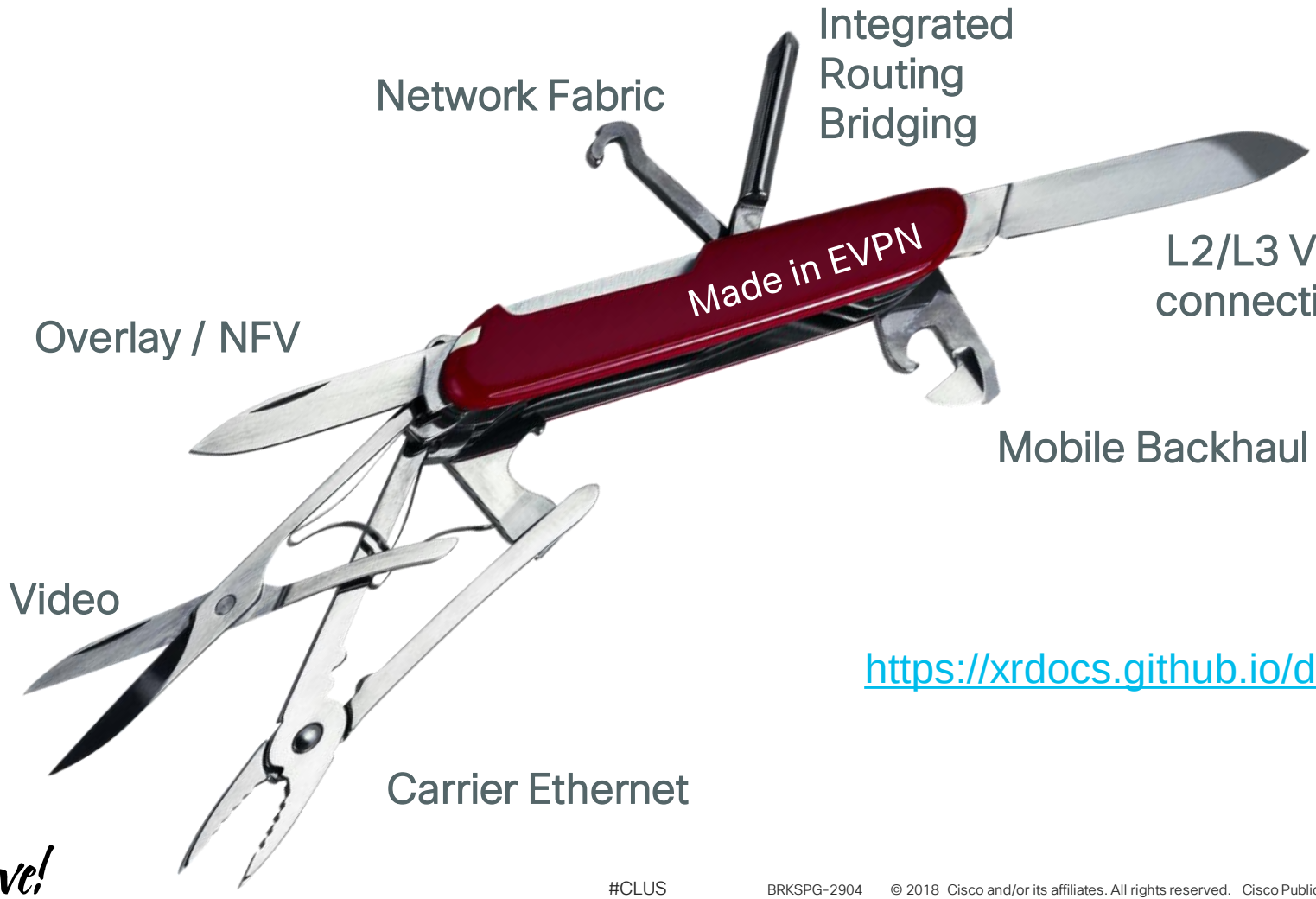
Existing VPLS solutions do not offer an All-Active per-flow redundancy

- Looping of Traffic Flooded from PE
- Duplicate Frames from Floods from the Core
- MAC Flip-Flopping over Pseudowire
- Rely on flooding and learning to build Layer 2 forwarding database



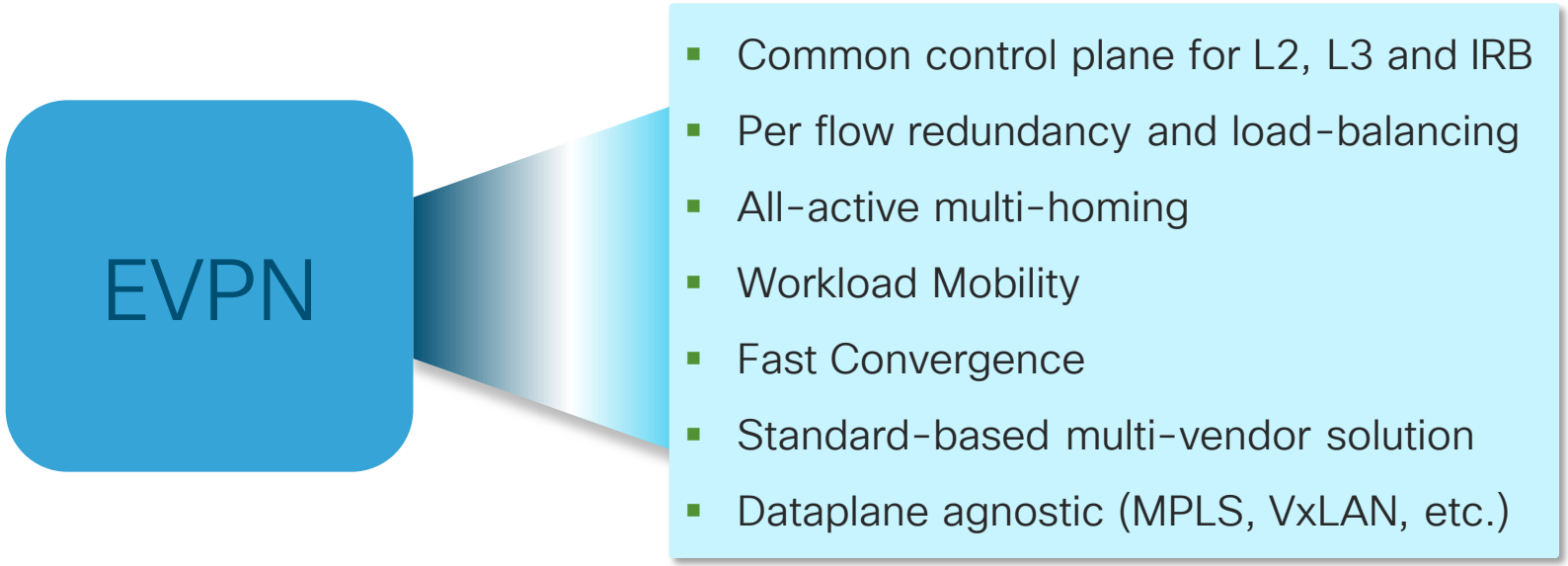
Why was EVPN needed in 2016?

- Network Operators have emerging needs in their network:
  - Data center interconnect operation (DCI)
  - Cloud and Services virtualization (DC)
  - Remove protocols and Network Simplification
  - Integrated of Layer 2 and Layer 3 Services over the same VPN



https://xrdocs.github.io/design/

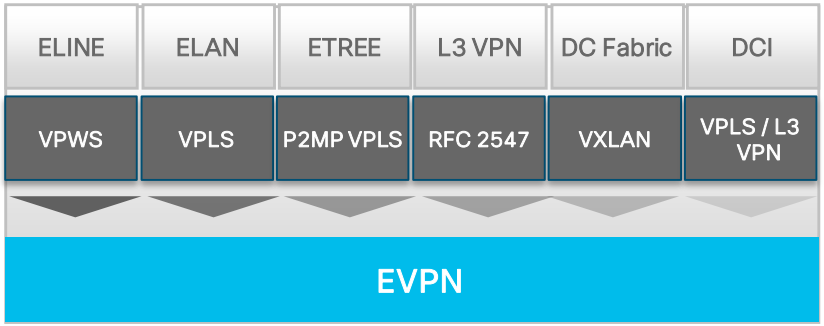
EVPN in Network Fabric



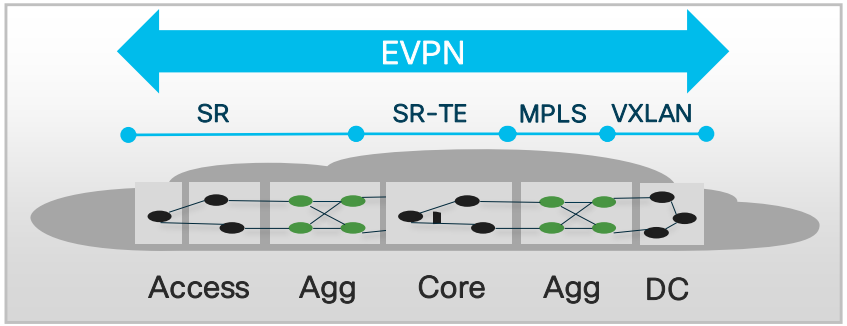
- EVPN with a choice of data plane encapsulation (MPLS/SR, VxLAN, SRv6) is the designed technology to address these requirements.

Ethernet VPN
Next generation network services

Single service for any application



E2E control and automation across domains



Optimized CapEx:
- Active-Active multi-homing
- Enhanced load balancing

Reduced OpEx:
- Single service, any application: faster time to market, certification
- E2E control and automation

- Increased Customer Value
  - Inter-domain SLA
- Better stability: no flood
- Granular policy control

EVPN Advantages:

Integrated Services

- Integrated Layer 2 and Layer 3 VPN services
- L3VPN-like principals and operational experience for scalability and control

Network Efficiency

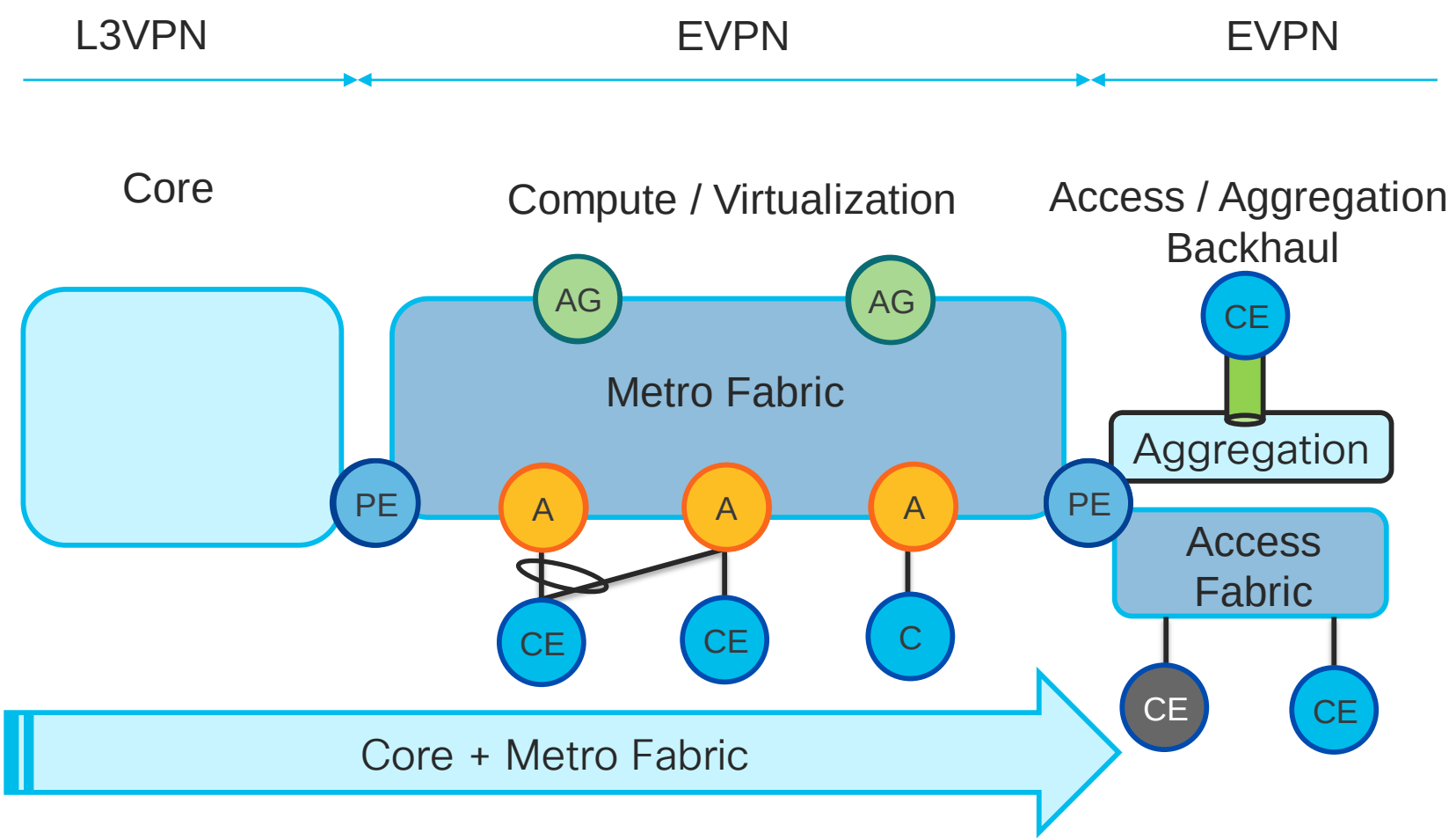
- All-active Multi-homing & PE load-balancing (ECMP)
- Fast convergence (link, node, MAC moves)
- Control-Place (BGP) learning. PWs are no longer used.
- Optimized Broadcast, Unknown-unicast, Multicast traffic delivery

Service Flexibility

- Choice of MPLS, VxLAN or PBB-MPLS data plane encapsulation
- Support existing and new services types (E-LAN, E-Line, E-TREE)
- Peer PE auto-discovery. Redundancy group auto-sensing

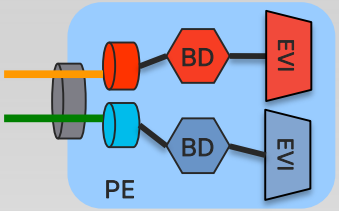
Investment Protection

- Fully support IPv4 and IPv6 in the data plane and control plane
- Open-Standard and Multi-vendor support



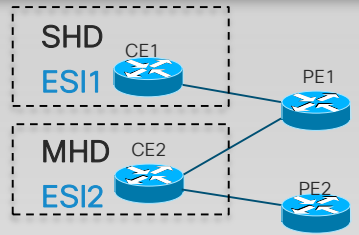
Concepts

EVPN Instance (EVI)



- EVI identifies a VPN in the network
- Encompass one or more bridge-domains, depending on service interface type
  - Port-based
  - VLAN-based (shown above)
  - VLAN-bundling

Ethernet Segment



- Represents a 'site' connected to one or more PEs
- Uniquely identified by a 10-byte global Ethernet Segment Identifier (ESI)
- Could be a single device or an entire network
  - Single-Homed Device (SHD)
  - Multi-Homed Device (MHD)
  - Single-Homed Network (SHN)
  - Multi-Homed Network (MHN)

BGP Routes

Route Types
[1] Ethernet Auto-Discovery (AD) Route
[2] MAC/IP Advertisement Route
[3] Inclusive Multicast Route
[4] Ethernet Segment Route
[5] IP Prefix Advertisement Route

- New SAFI [70]
- Routes serve control plane purposes, including:
  - MAC address reachability
  - MAC mass withdrawal
  - Split-Horizon label adv.
  - Aliasing
  - Multicast endpoint discovery
  - Redundancy group discovery
  - Designated forwarder election
  - IP address reachability
  - L2/L3 Integration

BGP Route Attributes

Extended Communities
ESI MPLS Label
ES-Import
MAC Mobility
Default Gateway
Encapsulation

- New BGP extended communities defined
- Expand information carried in BGP routes, including:
  - MAC address moves
  - Redundancy mode
  - MAC / IP bindings of a GW
  - Split-horizon label encoding
  - Data plane Encapsulation

Cisco Webex Teams

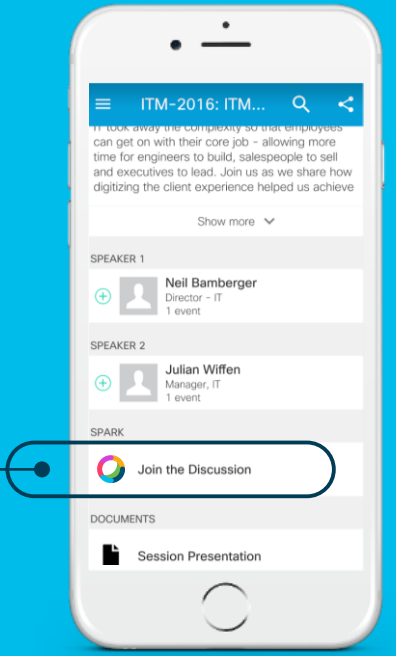
Questions?

Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

How

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

Webex Teams will be moderated by the speaker until June 18, 2018.



cs.co/ciscolivebot#BRKSPG-2904

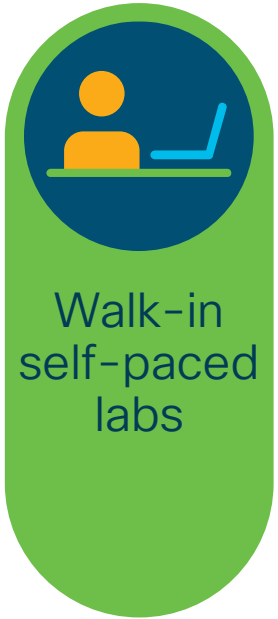
Complete your online session evaluation

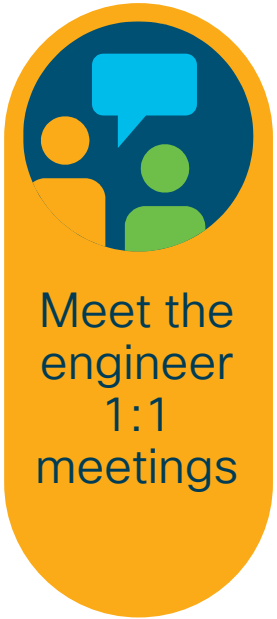
Give us your feedback to be entered into a Daily Survey Drawing.

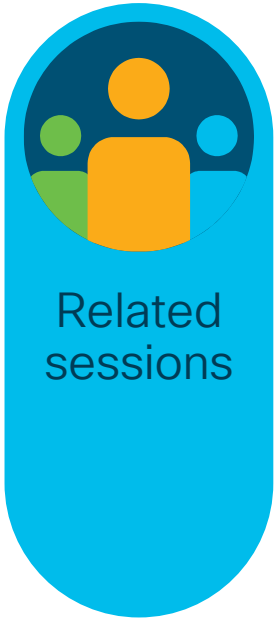
Complete your session surveys through the Cisco Live mobile app or on www.CiscoLive.com/us.

Don't forget: Cisco Live sessions will be available for viewing on demand after the event at www.CiscoLive.com/Online.

Continue your education







Thank you

INTUITIVE