
TOMORROW starts here.
cisco
Cisco live!

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

BRKSPG-2904

LJ Wobker

Technical Marketing Engineer

High End Routing & Optical Group

Cisco live!

Level Setting

- Apologies for the accent – throw things when I talk too fast.
- Impossible to give a truly thorough talk on all these topics ;-)
- Most of the output/show command slides are for reference
- Infinite time to answer questions...
- but not in this room within this specific session...
- Meet the engineer and/or World of Solutions is your friend

Cisco ASR9000 – Next-Gen Edge Routing Platform

Key Design Goals & System Benefits

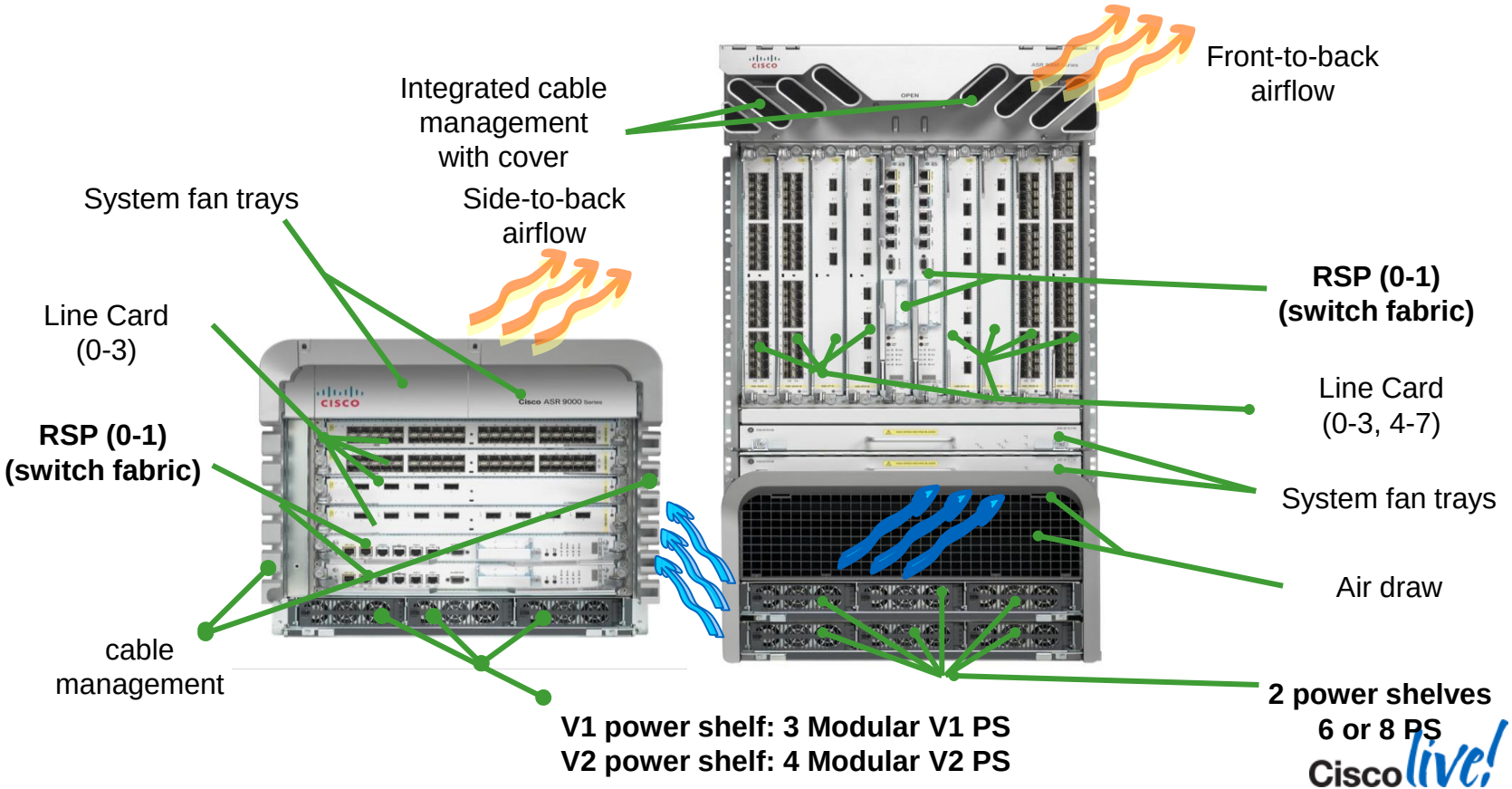
- Architectural Design for Longevity
- Product Portfolio with significant HW and SW commonality
- Highly integrated network processors
- Cisco IOS XR based
  - Truly modular, full distributed OS
  - Enhanced for the Edge (L2 and L3)
- nV (Network Virtualisation) for Operational Simplicity

Agenda

- ASR 9000 Hardware Overview
  - System Introduction and Chassis Overview
  - System Components: RPs, fabric, linecards
- IOS XR software overview
- ASR 9000 QoS architecture & configuration
- Cisco nV – Network Virtualisation
  - nV satellite
  - nV edge
- Q&A

ASR 9000 Hardware Overview

ASR 9010 and ASR 9006 Chassis (circa 2008)

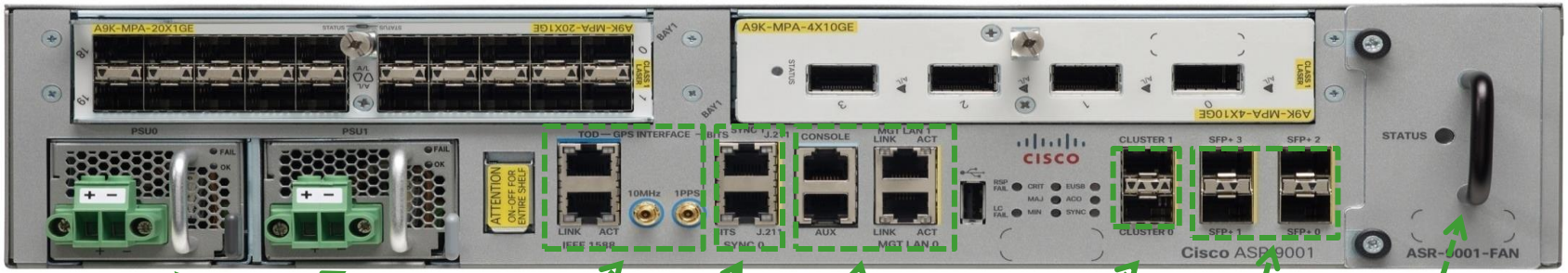


ASR 9001 Compact Chassis

Shipping Since XR 4.2.1, May 2012

Sub-slot 0 with MPA

Sub-slot 1 with MPA



Redundant (AC or DC) Power Supplies Field Replaceable

GPS, 1588

BITS

Console, Aux, Management

EOBC ports for nV Edge (2xSFP)

Fixed 4x10G SFP+ ports

Fan Tray Field Replaceable

ASR 9001-S Compact Chassis

Shipping Since XR4.3.1, May 2013

Supported MPAs:

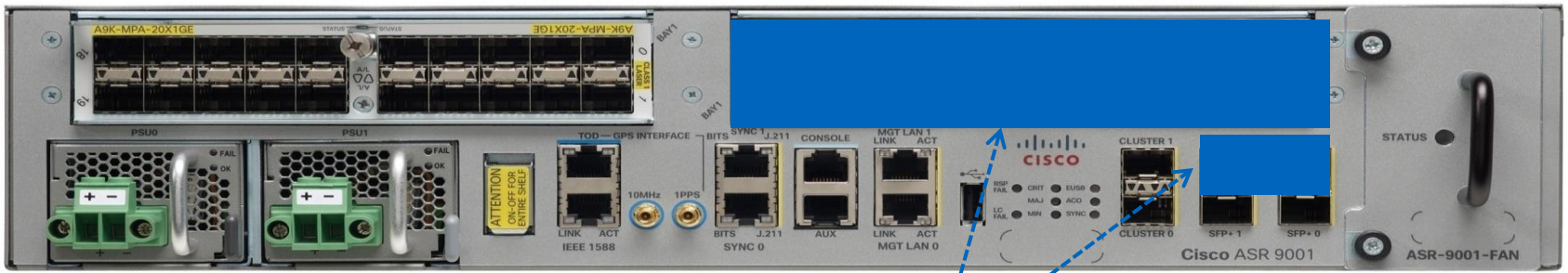
20x1GE
2x10GE
4x10GE
1x40GE

Pay As You Grow

- Low entry cost
- **SW License upgradable to full 9001**

Sub-slot 0 with MPA

Sub-slot 1 with MPA

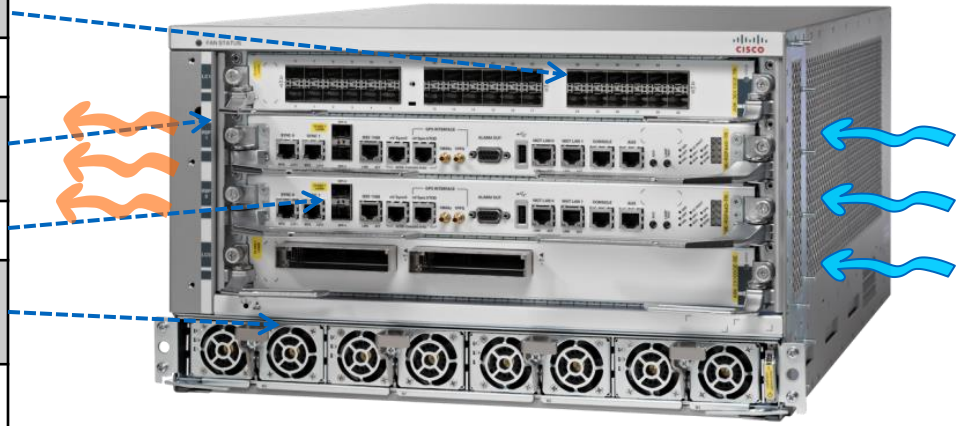


60G bandwidth are disabled by software. SW license to enable it

ASR 9904

Shipping Since 5.1.0, Sep 2013

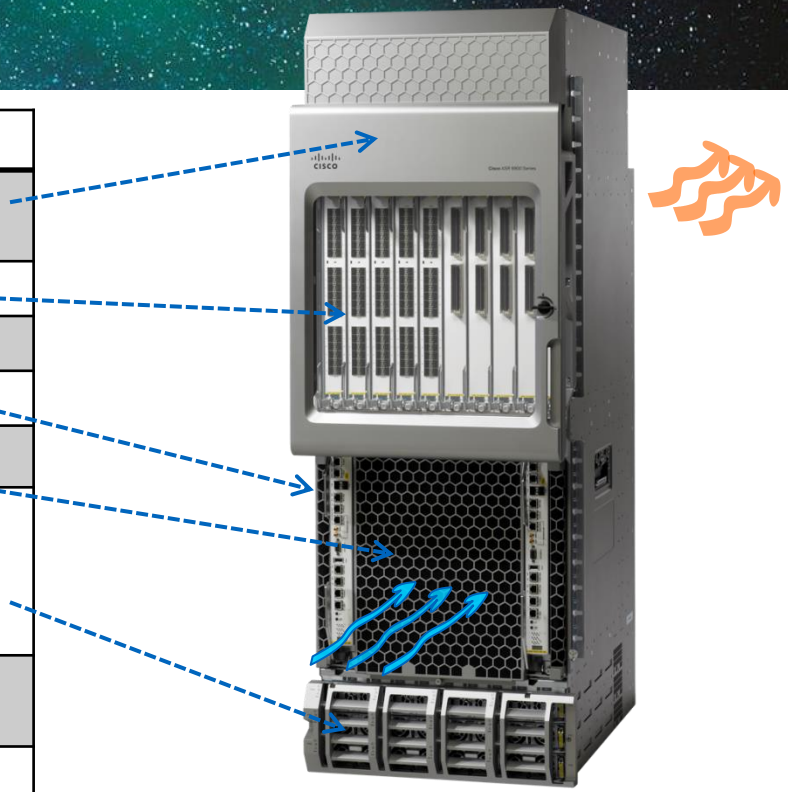
Feature	Description
I/O Slots	2 I/O slots
Rack size	6RU
Fan	Side to Side Airflow 1 Fan Tray, FRU
RSPs	RSP440, 1+1
Power	1 Power Shelf, 4 Power Modules 2.1 KW DC / 3.0 KW AC supplies
Fabric Bandwidth	Phase 1: 770G/slot Future capability: 1.7 Tb per Slot
SW	XR 5.1.0 – August 2013



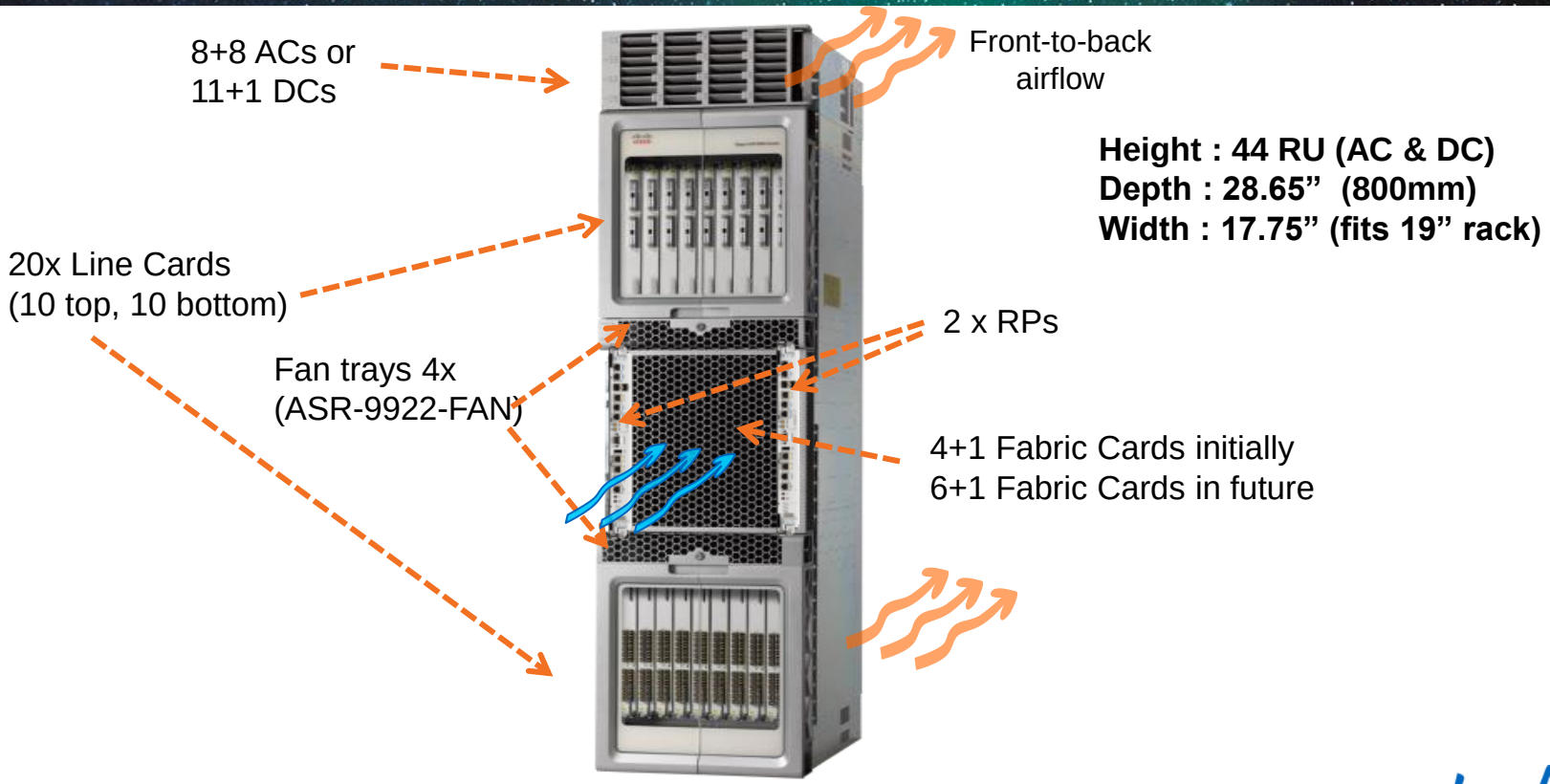
ASR 9912 Chassis

Shipping Since XR4.3.2 & 5.1.0, Sep 2013

Features	Description
Fan	2 Fan Trays Front to back airflow
I/O Slots	10 I/O slots
Rack Size	30 RU
RP	1+1 RP redundancy
Fabric	6+1 fabric redundancy
Power	3 Power Shelves, 12 Power Modules 2.1 KW DC / 3.0 KW AC supplies N+N AC supply redundancy N:1 DC supply redundancy
Bandwidth	Phase 1: 550Gb per Slot Future: 2+Tb per Slot
SW	XR 4.3.2 & 5.1.0



ASR 9922 Chassis



System Components

Power and Cooling

ASR-9010-FAN

ASR-9006-FAN

- Fans unique to chassis
- Variable speed for ambient temperature variation
- Redundant fan-tray
- Low noise, NEBS and OSHA compliant

Power Supply

DC Supplies

A
B
1.5 kW

A
B
2.1 kW

AC Supplies

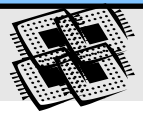
A
3 kW

B
3 kW

- Single power zone
- All power supplies run in active mode
- Power draw shared evenly
- 50 Amp DC Input or 16 Amp AC for Easy CO Install

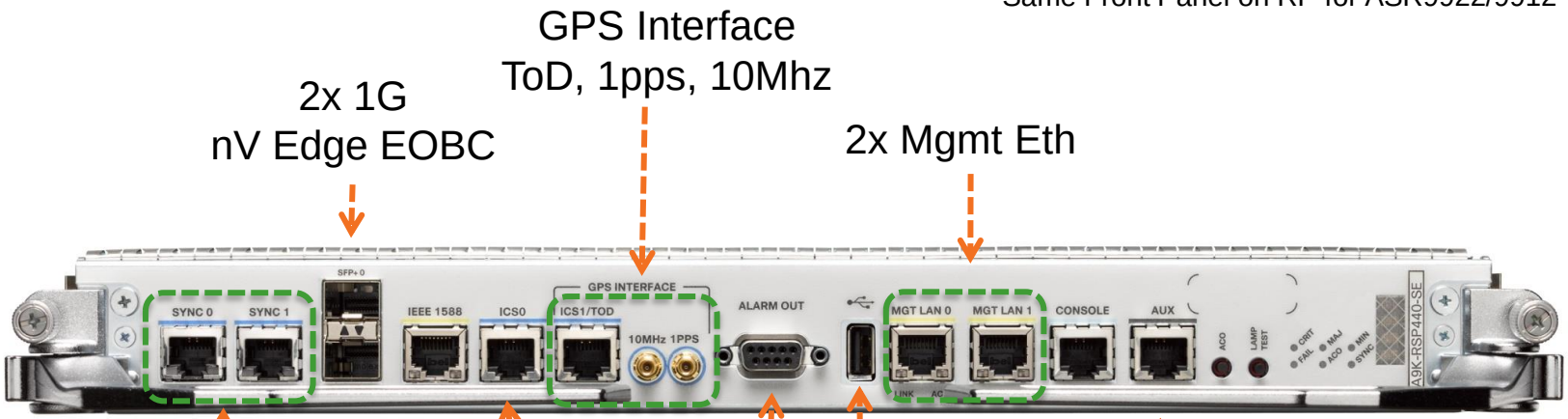
Control Processors (RP and RSP)

RSP used in ASR9006/ASR9010, RP used in ASR9922

	RSP	RSP440	9922-RP
Cores			
Processors	PPC/Freescale 2 Core 1.5GHz	Intel x86 4 Core 2.27 GHz	Intel x86 4 Core 2.27 GHz
RAM	RSP-4G: 4GB RSP-8G: 8GB	RSP440-TR: 6GB RSP440-SE: 12GB	-TR: 6GB -SE: 12GB
nV EOBC ports	No	Yes, 2 x 1G/10G SFP+	Yes, 2 x 1G/10G SFP+
Switch fabric bandwidth	92G + 92G (with dual RSP)	220+220G (with dual RSP)	660+110 (7-fabric model)

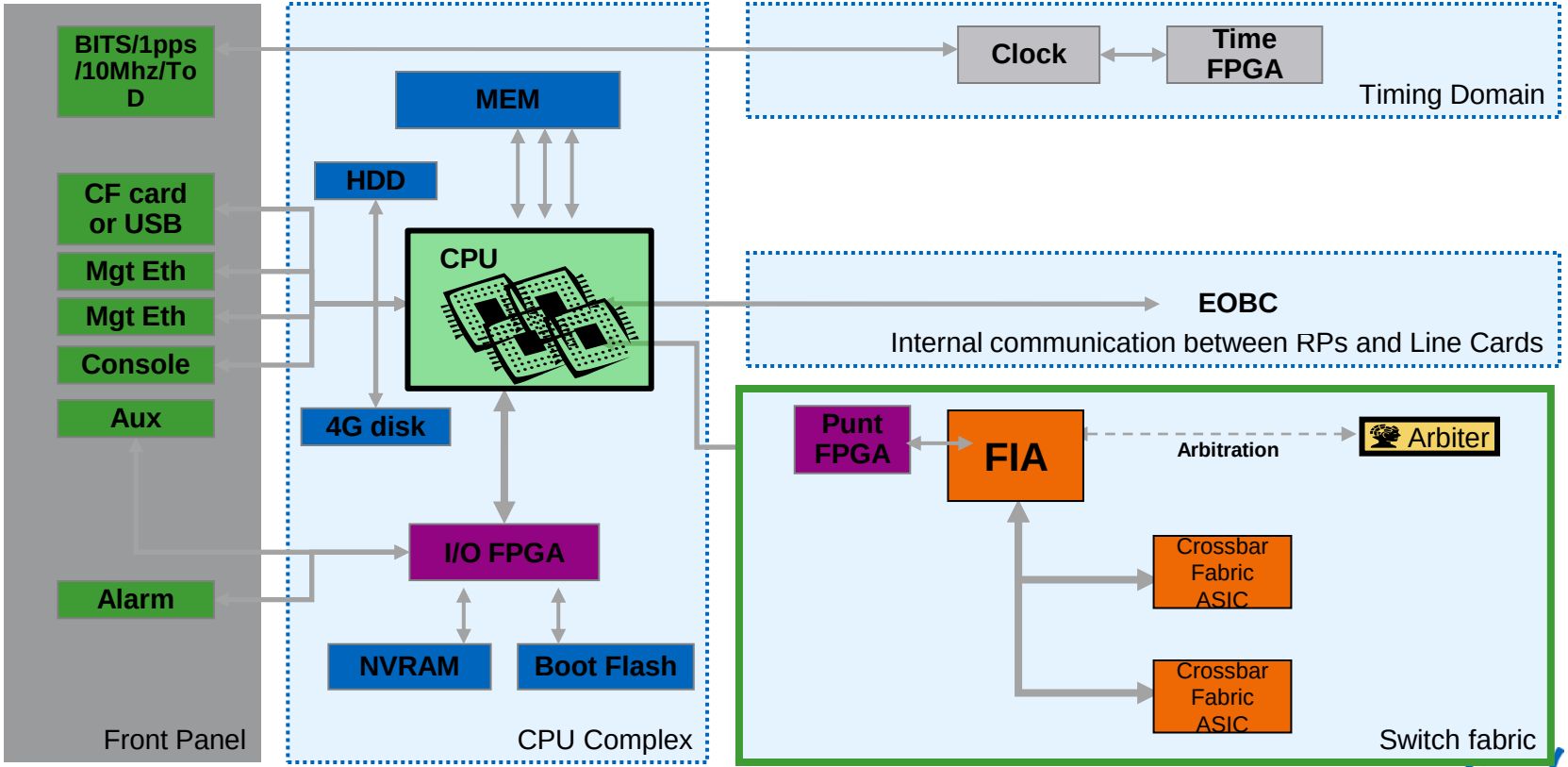
RSP440 – Faceplate and Interfaces

Same Front Panel on RP for ASR9922/9912



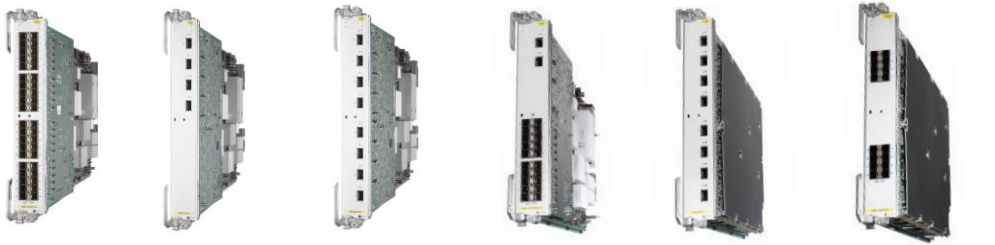
1) Future SW support

RSP Engine Architecture



ASR 9000 Ethernet Line Card Overview

First-generation LC (Trident)



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

-TR, -SE

Second-gen LC (Typhoon)

A9K-24x10GE | A9K-2x100GE

A9K-MOD160

A9K-MOD80

A9K-36x10GE

MPAs
20x1GE
2x10GE
4x10GE
8x10GE
1x40GE
2x40GE

ASR 9000 ISM (Integrated Service Module)

CDS Streaming:
TV and internet streaming
Error repair

CGN (carrier grade NAT):
NAT44, DS-Lite
NAT64

Feature	ASR 9000 ISM Capabilities
Applications	Ultra-Dense VoD, TV, Internet Streaming, Error Repair, CGv6
Bandwidth	30-40 Gbps streaming capacity ~3 Gbps cache fill rate
Compatibility	Works with all CDS appliances
Concurrent Streams	Up to 8,000 SD equivalent
Content Cache	3.2 TBytes at FCS - Modular Design
Video Formats	MPEG2 & AVC/H.264
Transport	MPEG over UDP / RTP
Session Protocols	RTSP / SDP
Environmental	NEBS / ETSI compliant

CDS: Manage 8,000 streams up to 40G per second
CGv6: 20M translations, 1M translations/sec., ~15Gbps throughput / ISM

ASR 9000 Optical Interface Support

- All linecards use transceivers
- Based on density and interface type
  - 1GE (SFP) T, SX, LX, ZX, CWDM/DWDM
  - 10GE (XFP & SFP+): SR, LR, ZR, ER, DWDM
  - 40GE (QSFP): SR4, LR4
  - 100GE (CFP): SR10, LR4, DWDM 1)

XFP

SFP, SFP+

QSFP

CFP

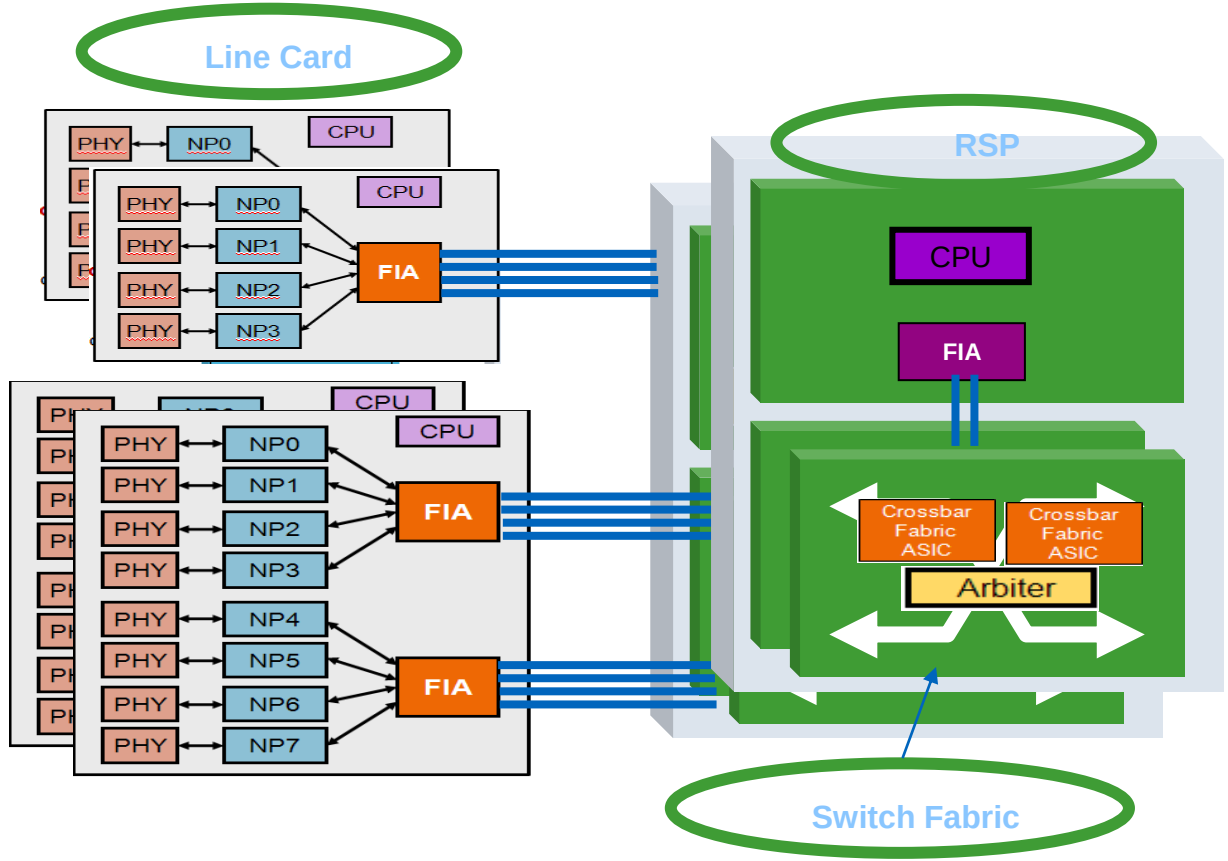
All 10G and 40G Ports do support G.709/OTN/FEC

1) Using Optical Shelf (ONS15454 M2/M6)

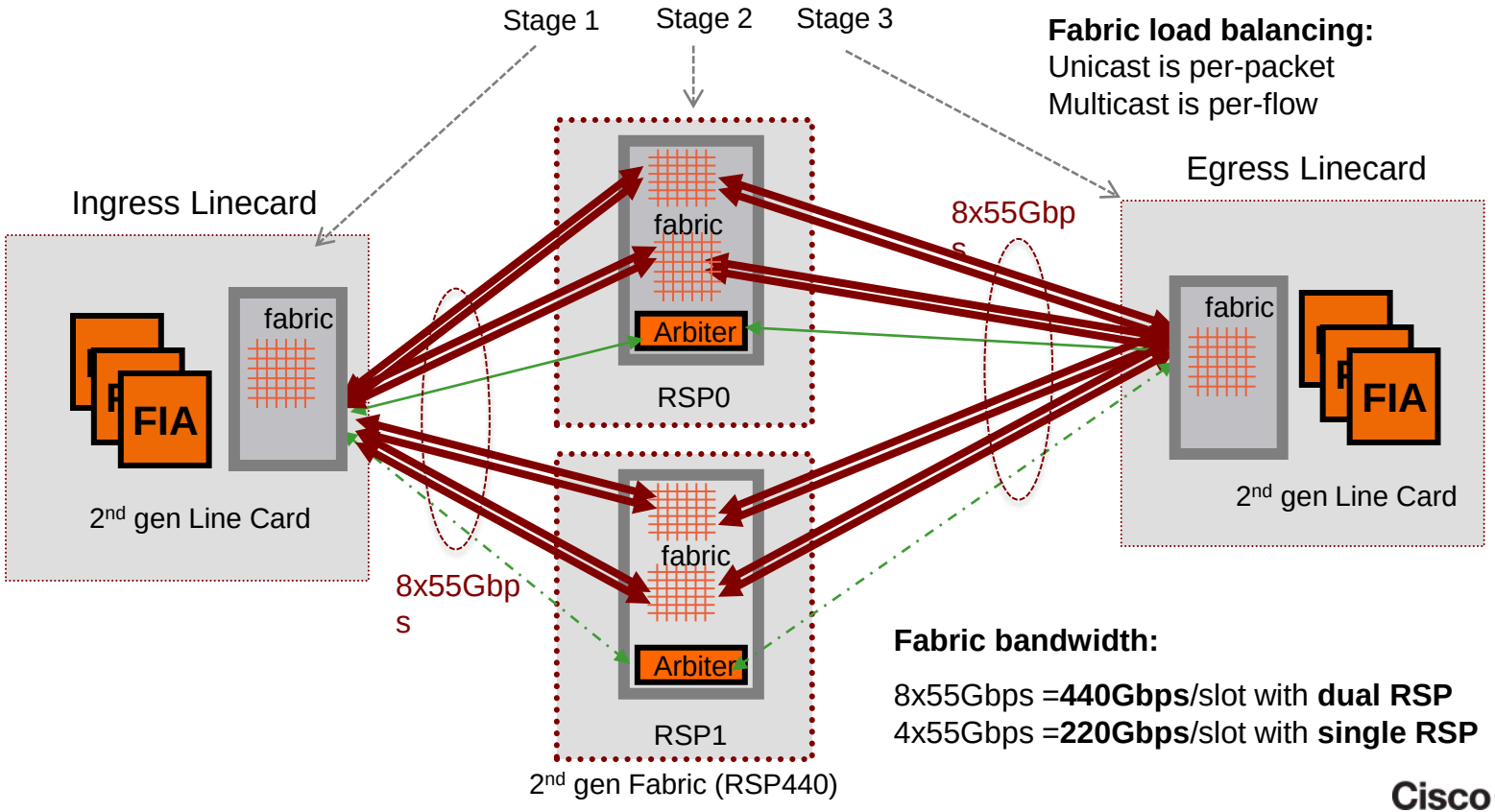
For latest Transceiver Support Information
http://www.cisco.com/en/US/prod/collateral/routers/ps9853/data_sheet_c78-624747.html

Fabric Architecture

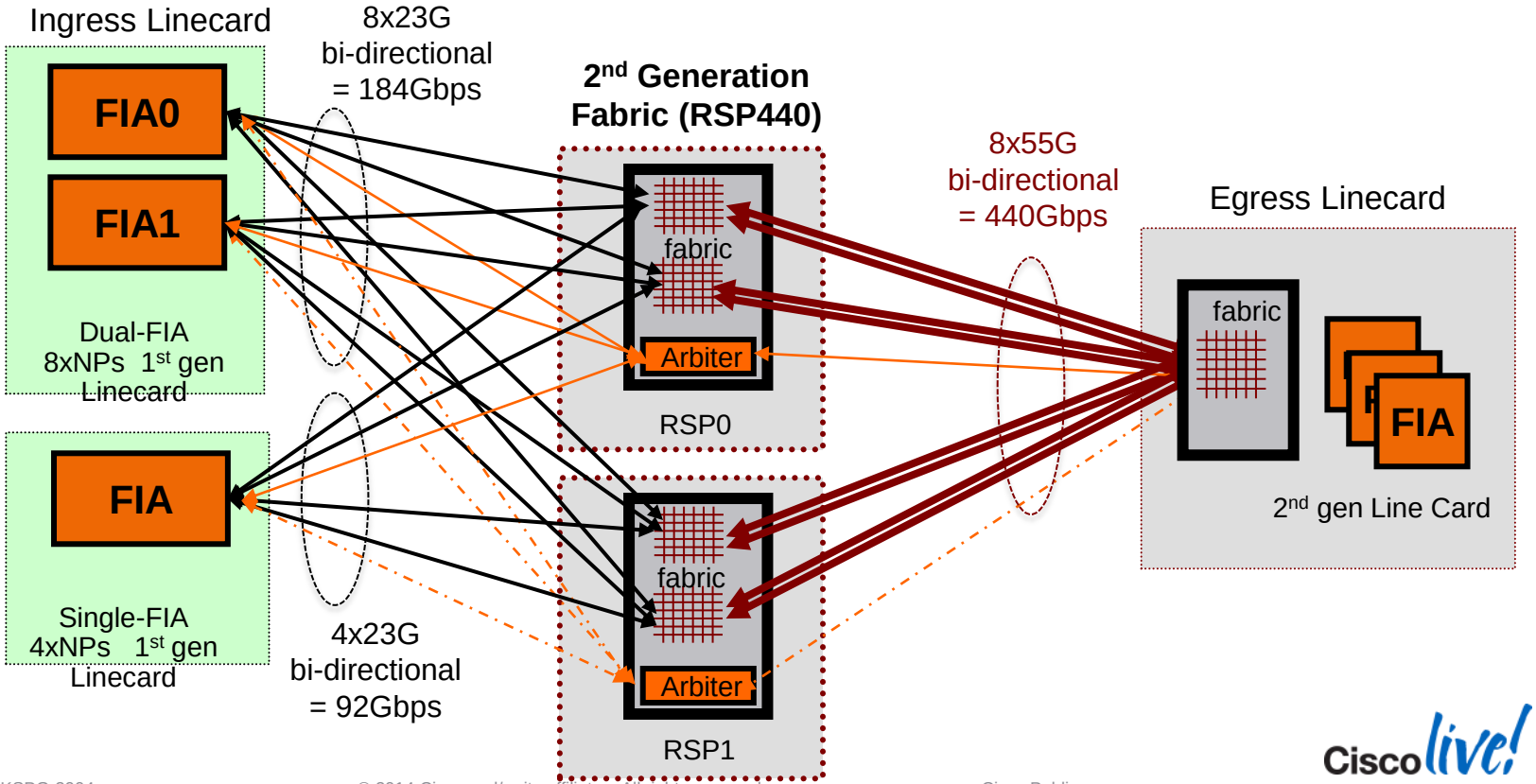
Cisco ASR9000 System Architecture



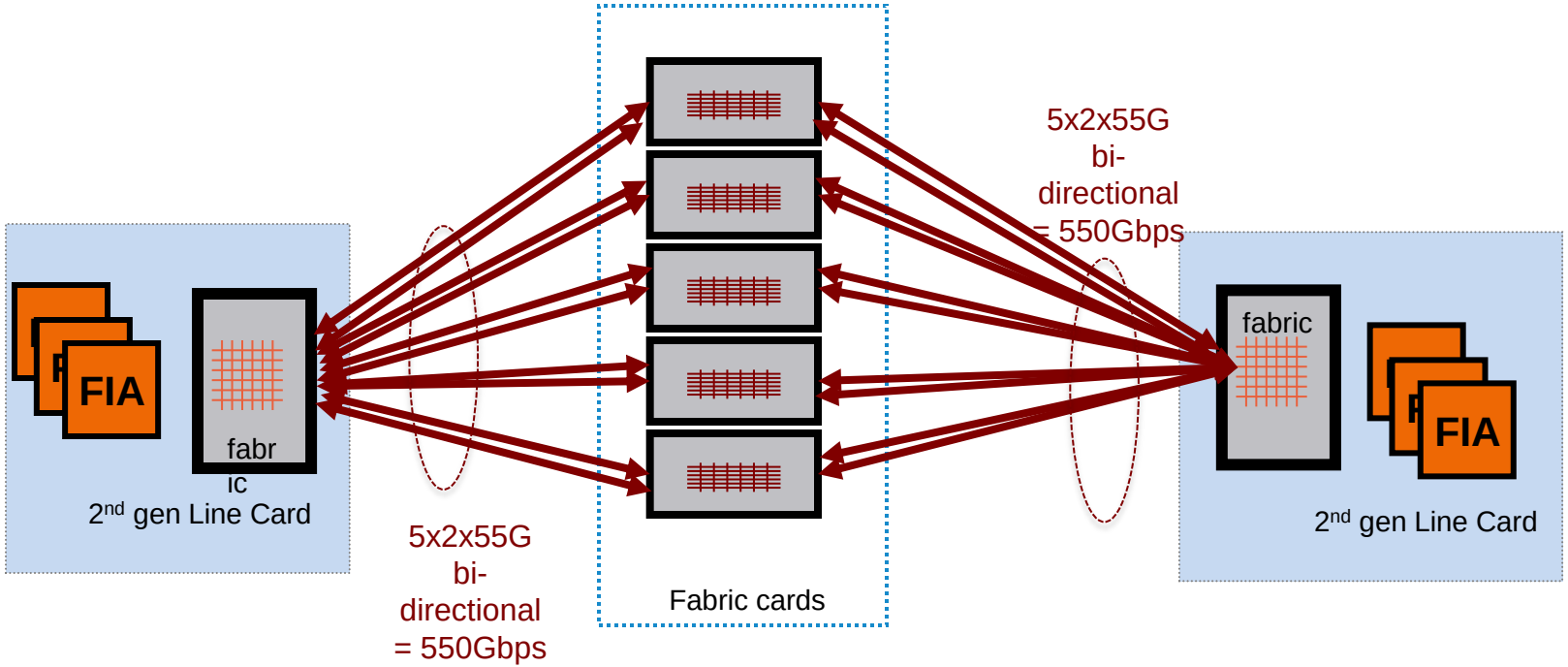
ASR9000 Switch Fabric Overview



1st/2nd Generation Switch Fabric Compatibility

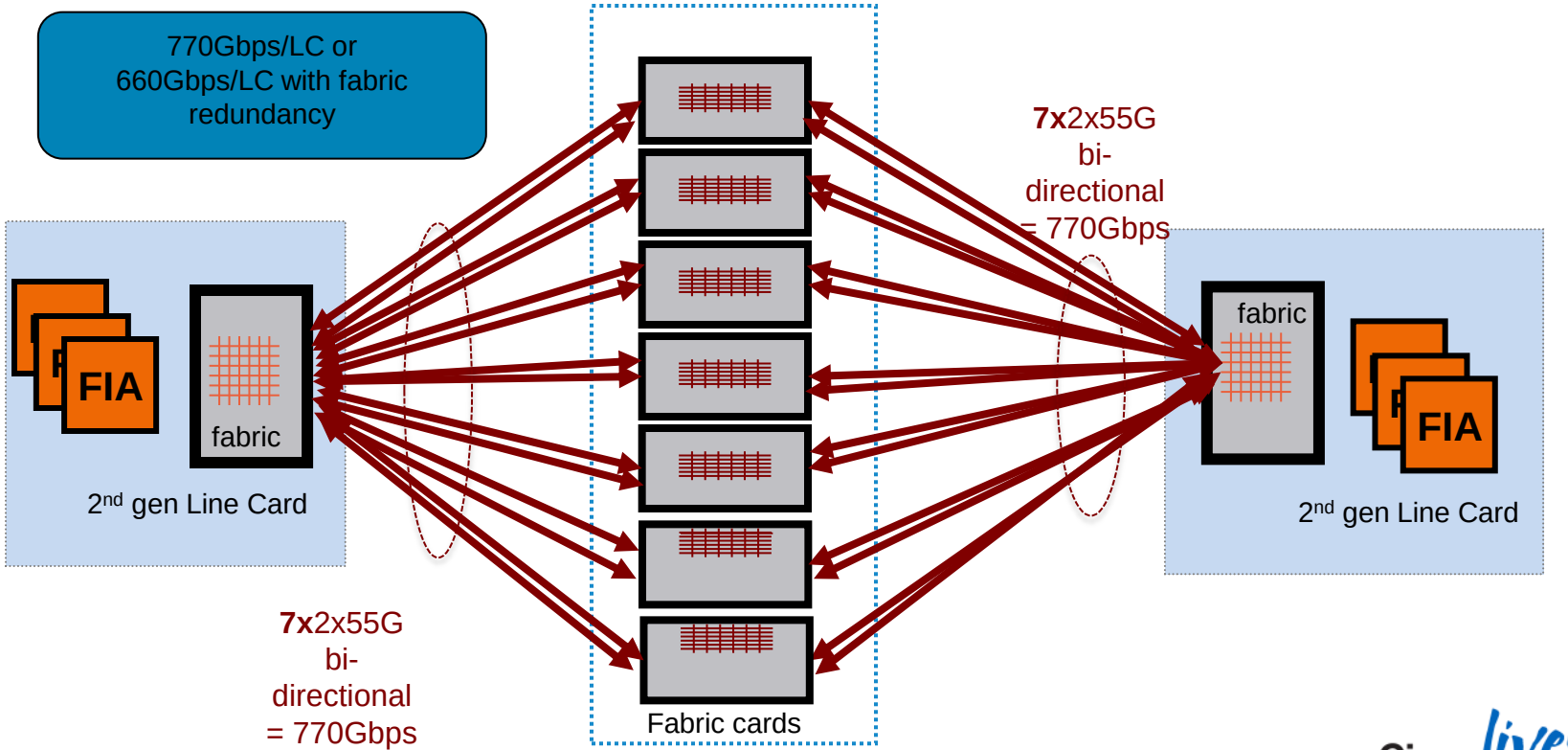


ASR 9912/9922 Fabric Architecture : 5-plane System



550Gbps/LC or 440Gbps/LC with fabric redundancy

ASR 9912/9922 Fabric Architecture : 7-plane System






Linecard Architecture

Generic Linecard Architecture – Components

Pluggable physical interfaces

PHY

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

CPU

CPU

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

Network Processor

NP

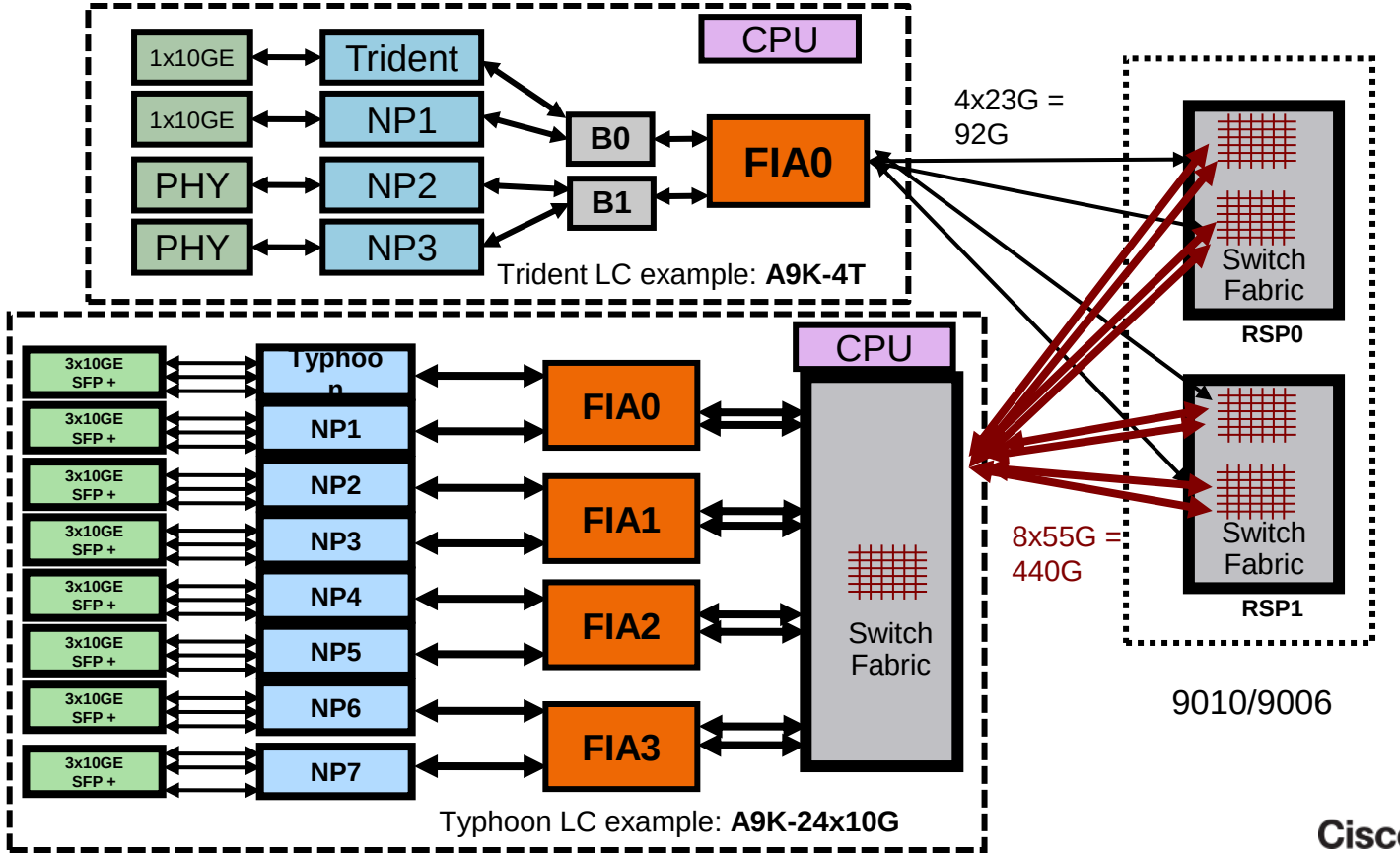
- forwarding and feature engine for the LC
- scales bandwidth via multiple NPs
  - up to 8 NPs/LC for performance vs. density options
- highly integrated silicon as opposed to multiple discrete components
  - shorter connections, faster communication channels
  - higher performance, density with lower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

Fabric Interface ASIC

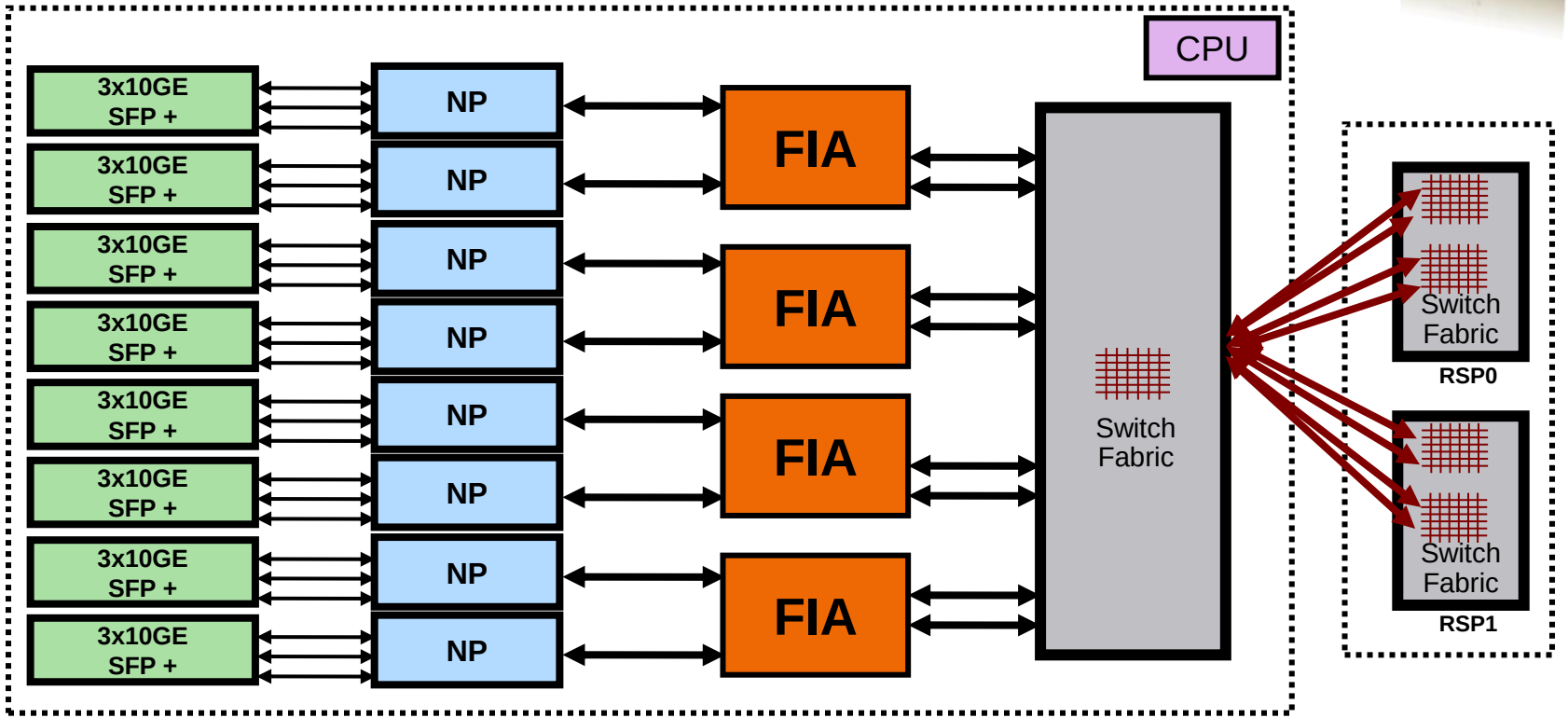
FIA

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

ASR 9000 Line Card Architecture Overview



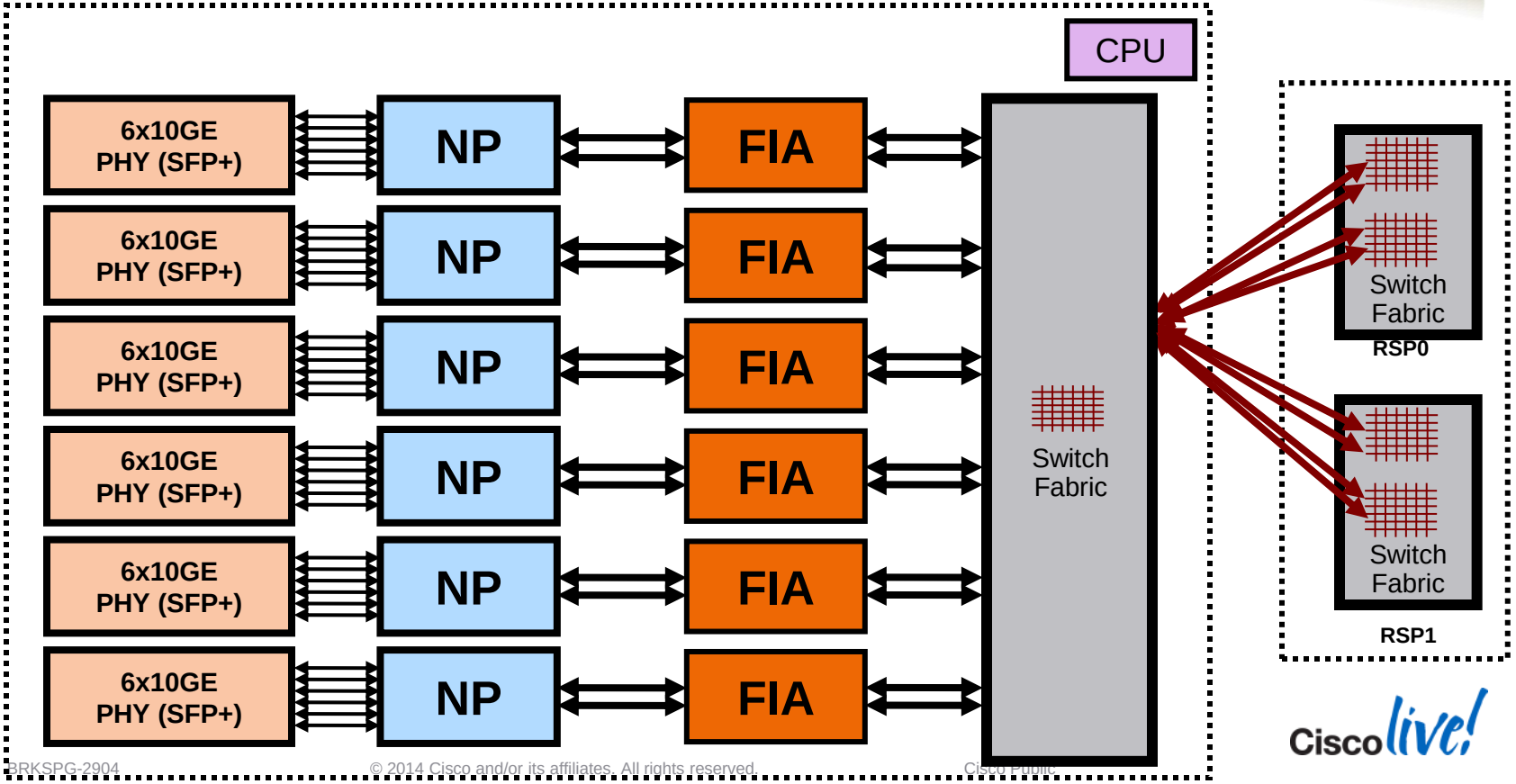
24port 10GE Linecard Architecture



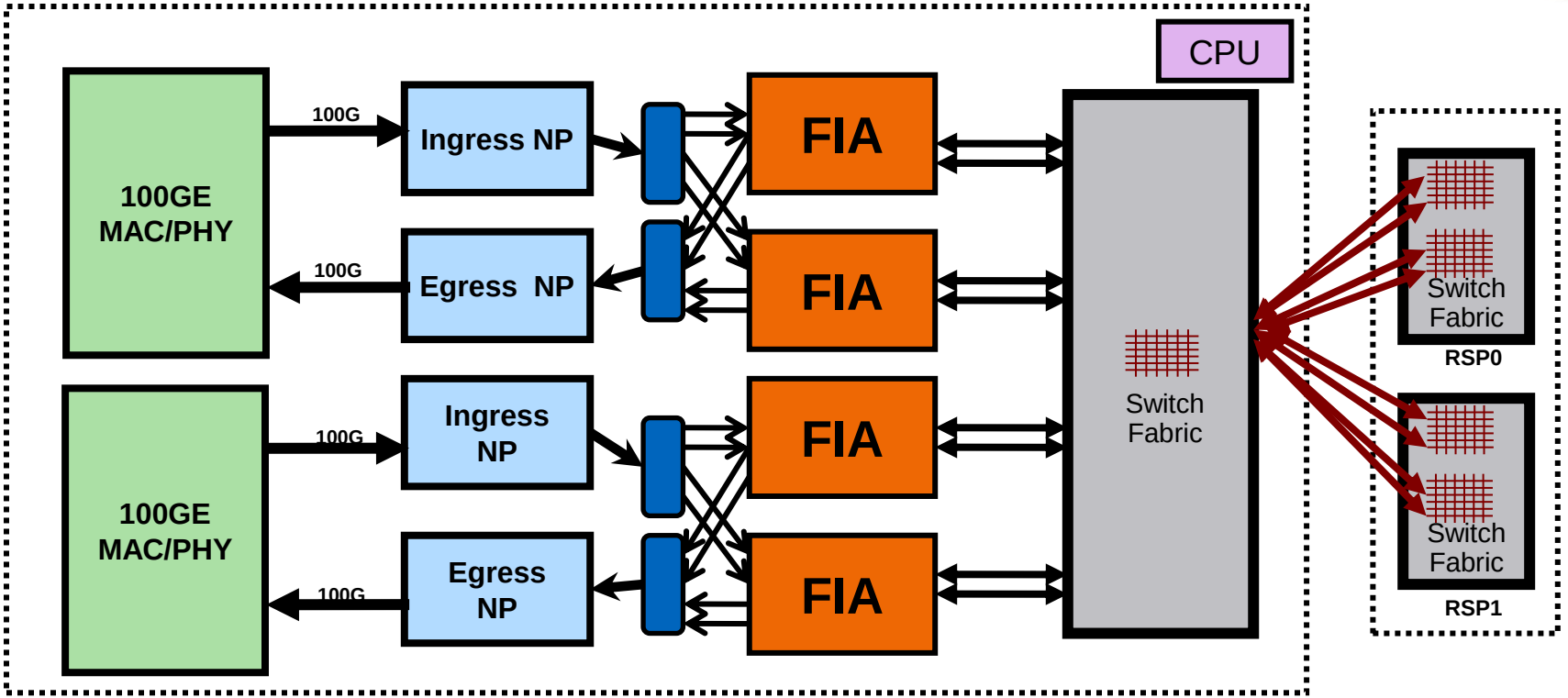
Each NP: 60Gbps bi-directional or 120Gbps uni-dir
2x45Mpps

Each FIA: >60Gbps bi-directional

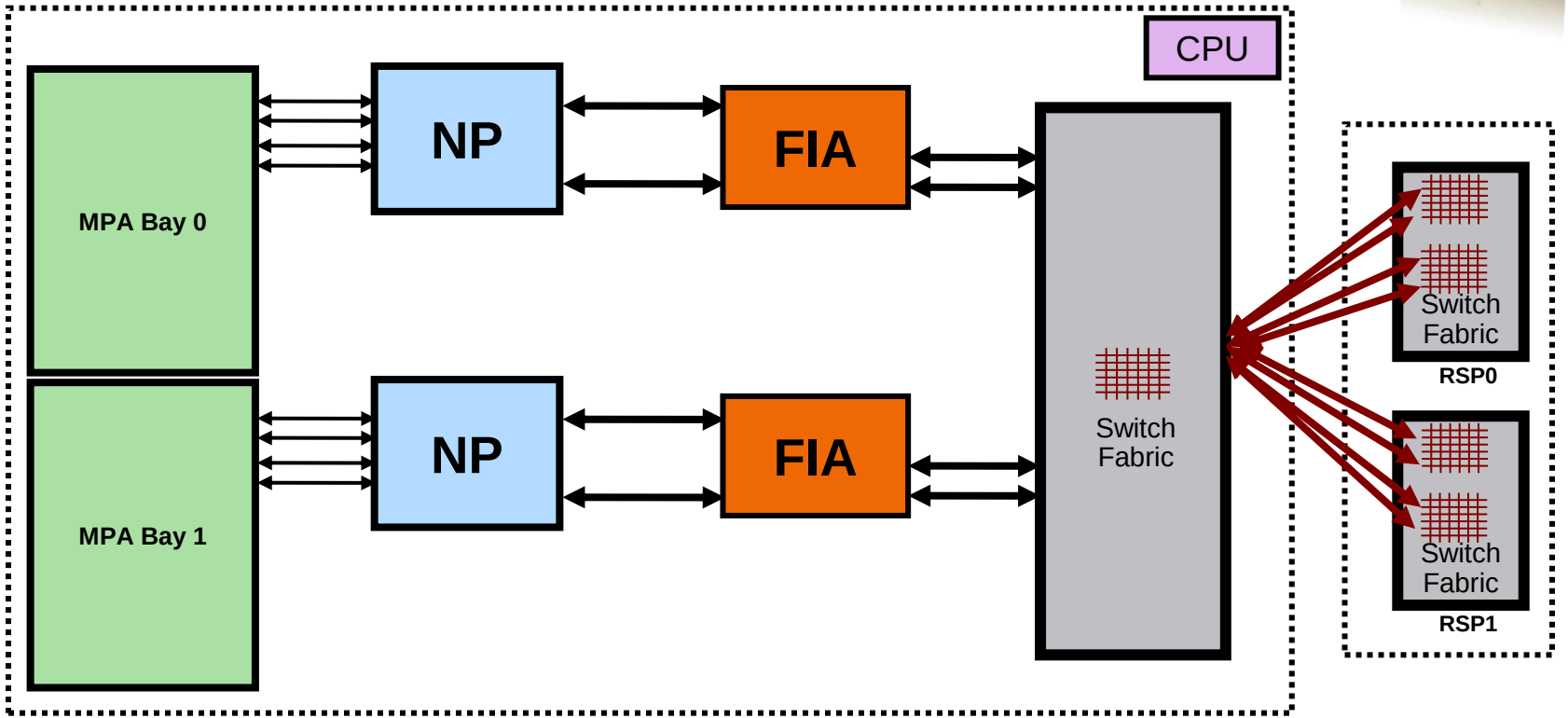
36port 10GE Linecard Architecture



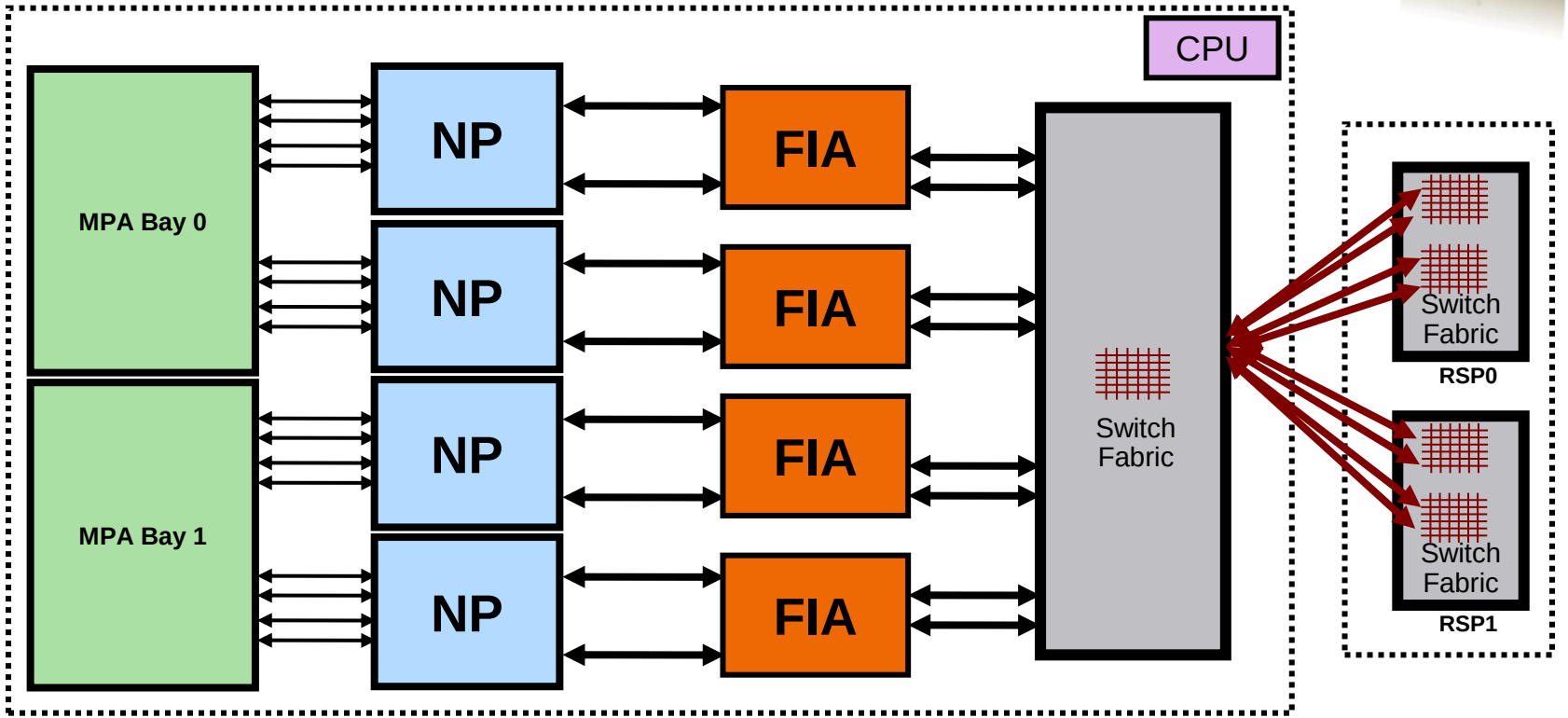
2port 100GE Linecard Architecture



Module Cards – MOD80

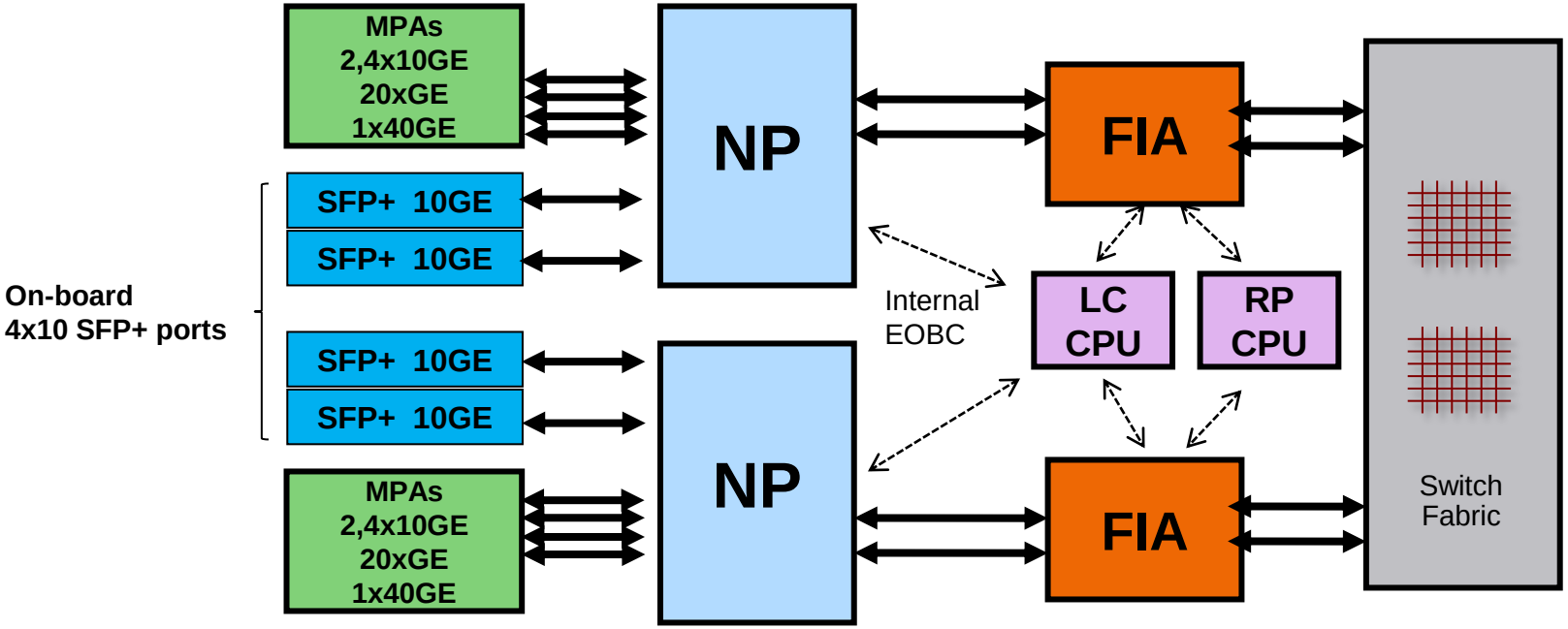


Module Cards – MOD160



ASR9001 Architecture

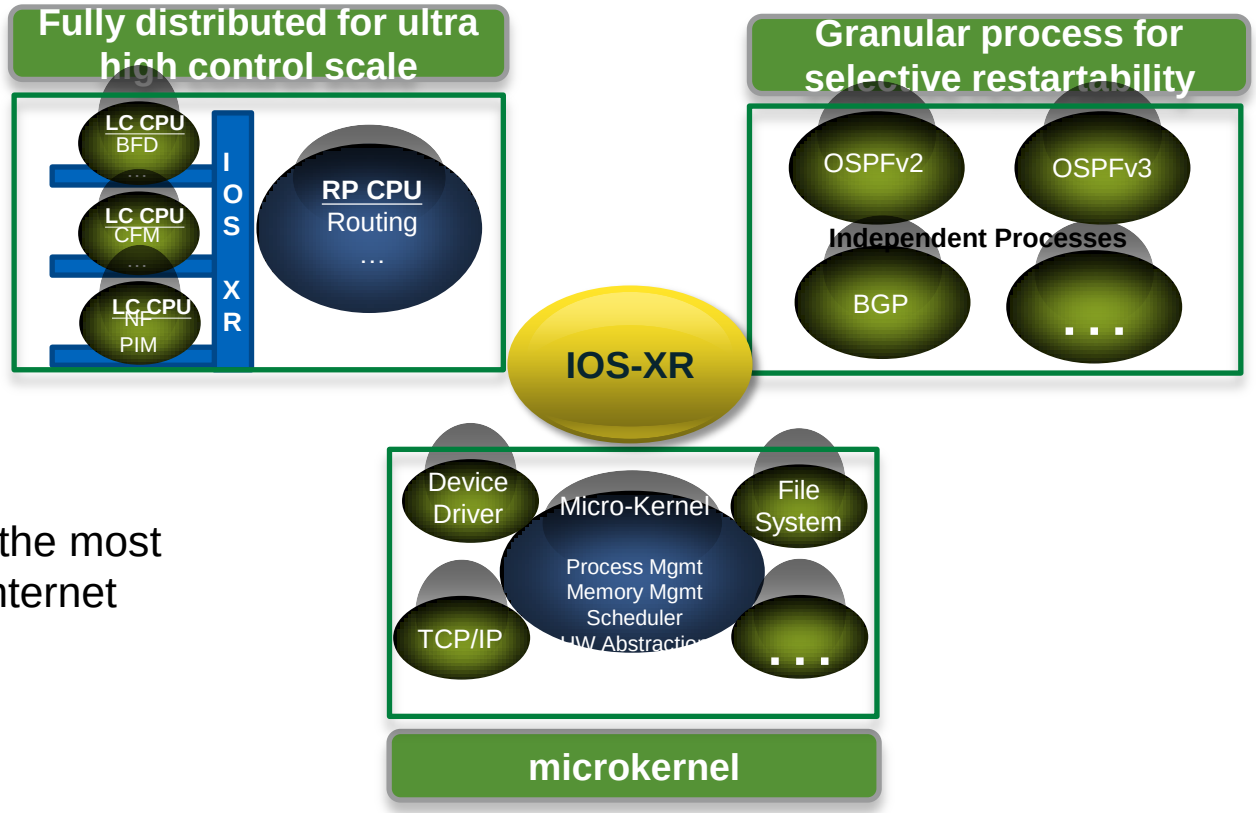
Same Hardware Components as the Modular Systems



Cisco IOS-XR Overview

Industry Hardened IOS XR

Modular, Fully Distributed



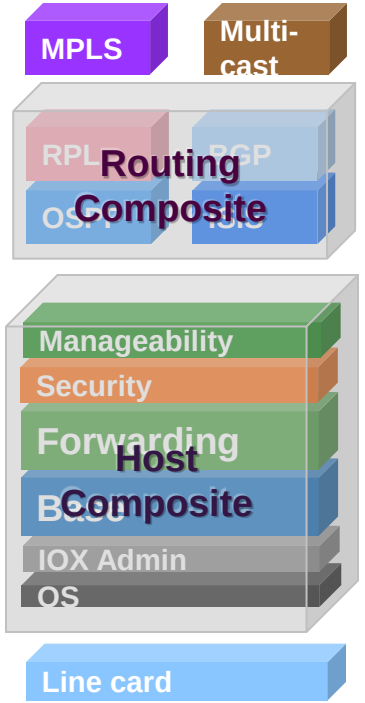
10+ years in the most demanding internet deployments

Cisco IOS-XR – High Level Benefits

- Modular — Runtime SW upgrade/downgrade support
- Distributed — scalable with multi chassis support
- Platform Independent — POSIX compliant
- Management Interface — Unified Data Model (XML)
- High Availability — Hot Standby and Process Restart
- Security — Control, Data and Management Plane

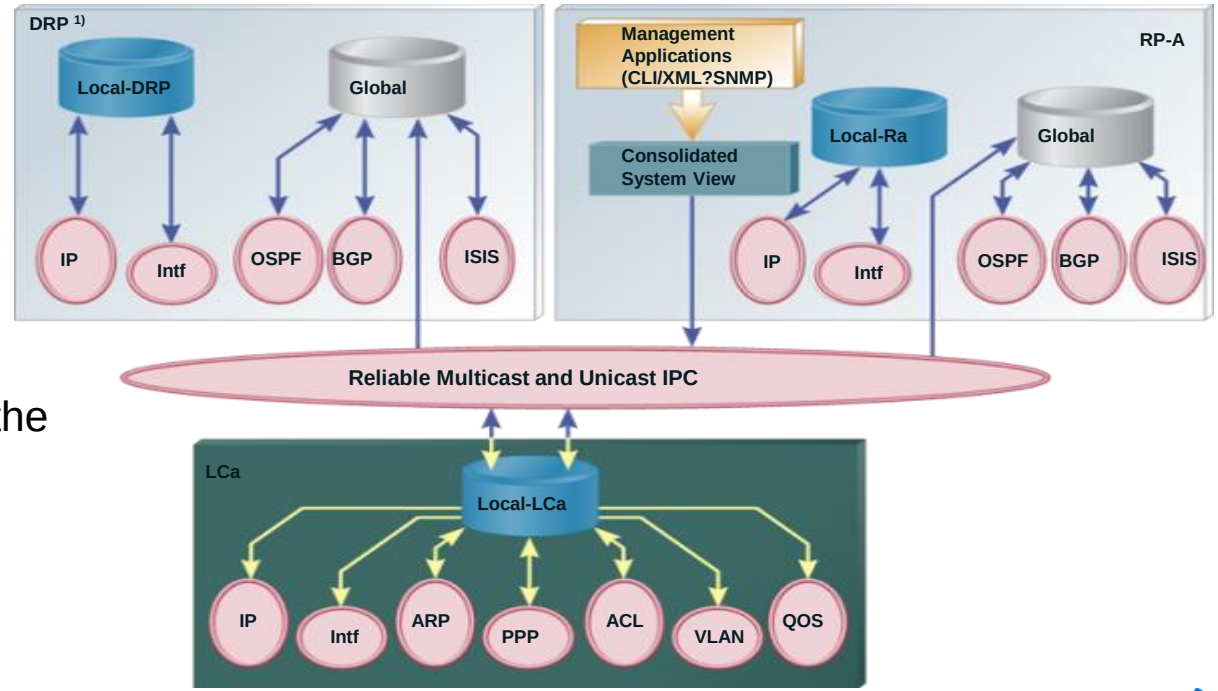
Cisco IOS-XR Software Modularity

- Ability to upgrade independently MPLS, multicast, routing protocols and linecards
- Ability to release software packages independently
- Ability to have composites into one manageable unit if desired
- Notion of optional packages if technology not desired on device : (Multicast, MPLS)



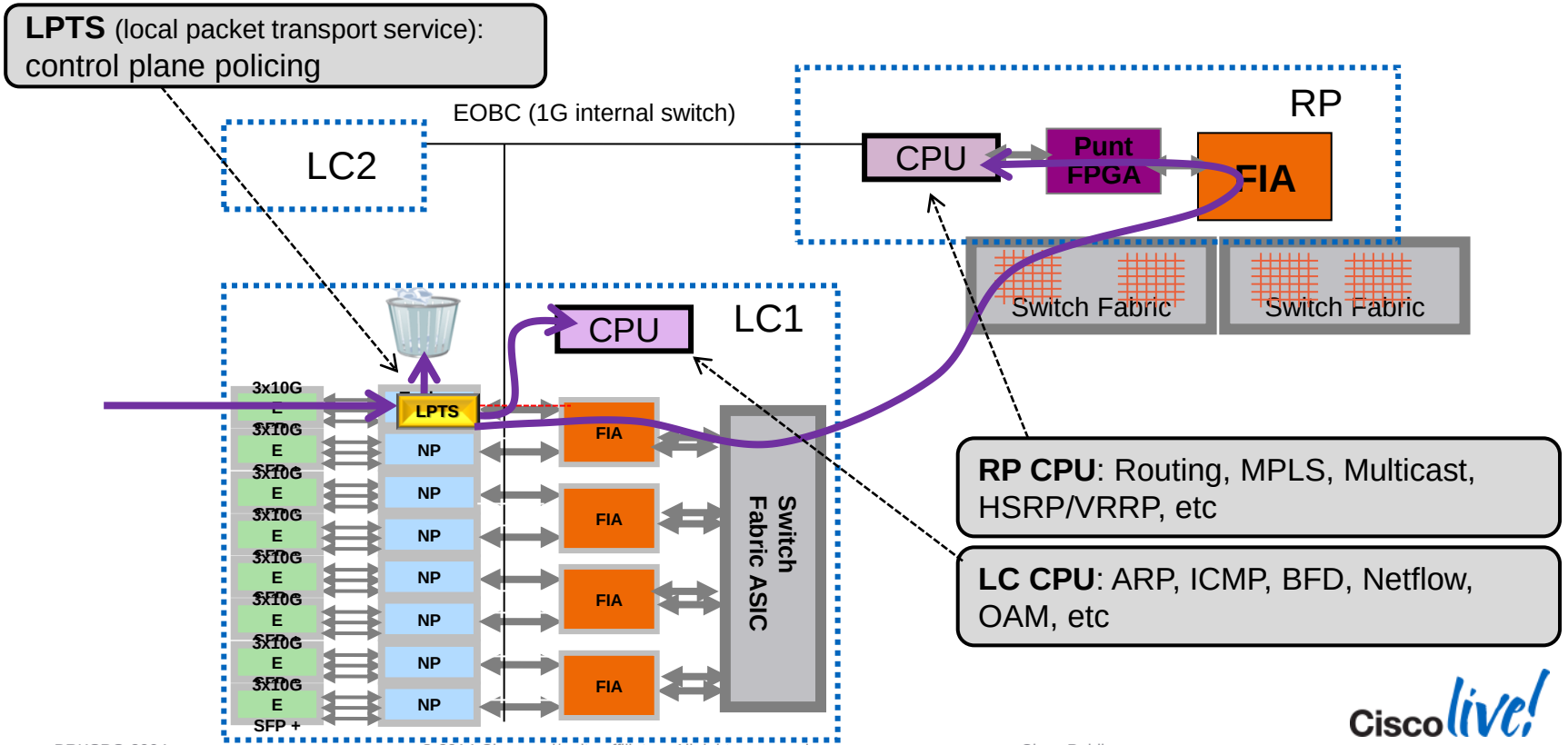
Distributed In-Memory Database

- Reliable Multicast IPC improves scale and performance
- Distributed data management model improves performance and Scale
- Single Consolidated view of the system eases maintenance

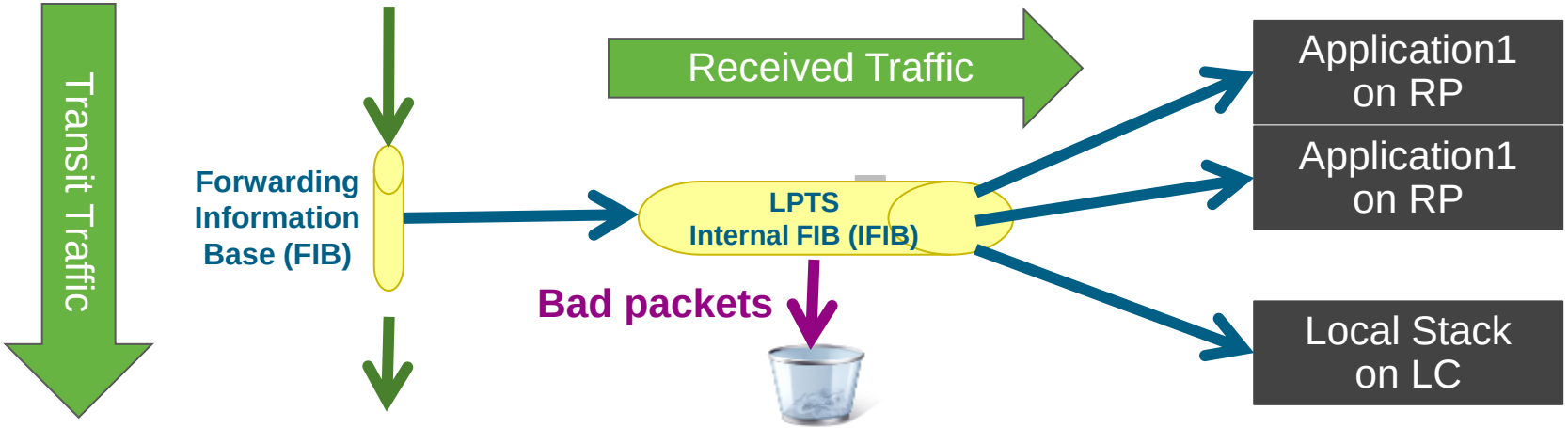


1) DRPs are only supported in CRS

ASR9000 Fully Distributed Control Plane

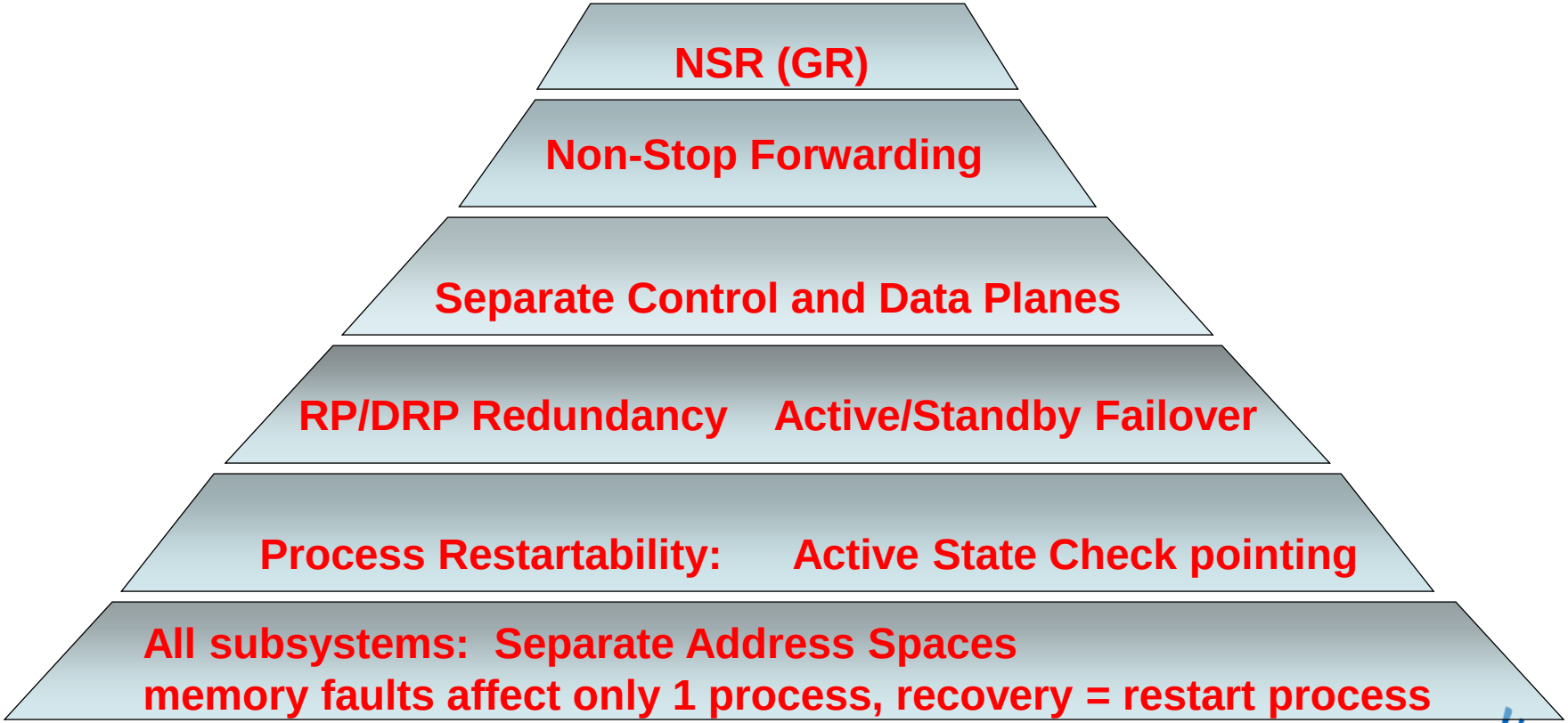


Local Packet Transport Services (LPTS)



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

IOS-XR High Availability Software Design Principles



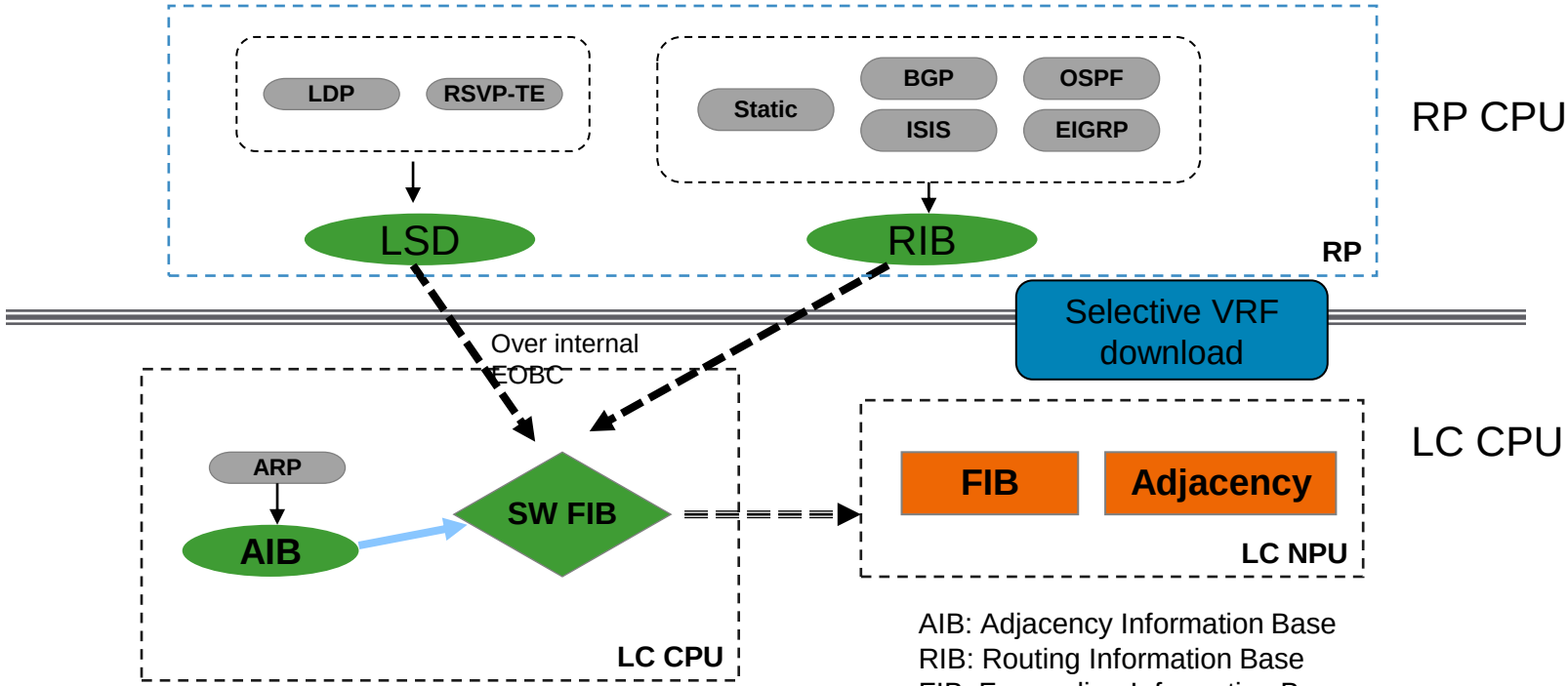
Software Maintenance Updates (SMUs)

- Allows for software package installation/removal leveraging on Modularity and Process restart
- Redundant processors are not mandatory (unlike ISSU) and in many cases is non service impacting and may not require reload.
- Mechanism for
  - delivery of software features (e.g. Multicast, MPLS)
  - delivery of critical bug fixes without the need to wait for next maintenance release




Control Plane and Packet Forwarding Infrastructure

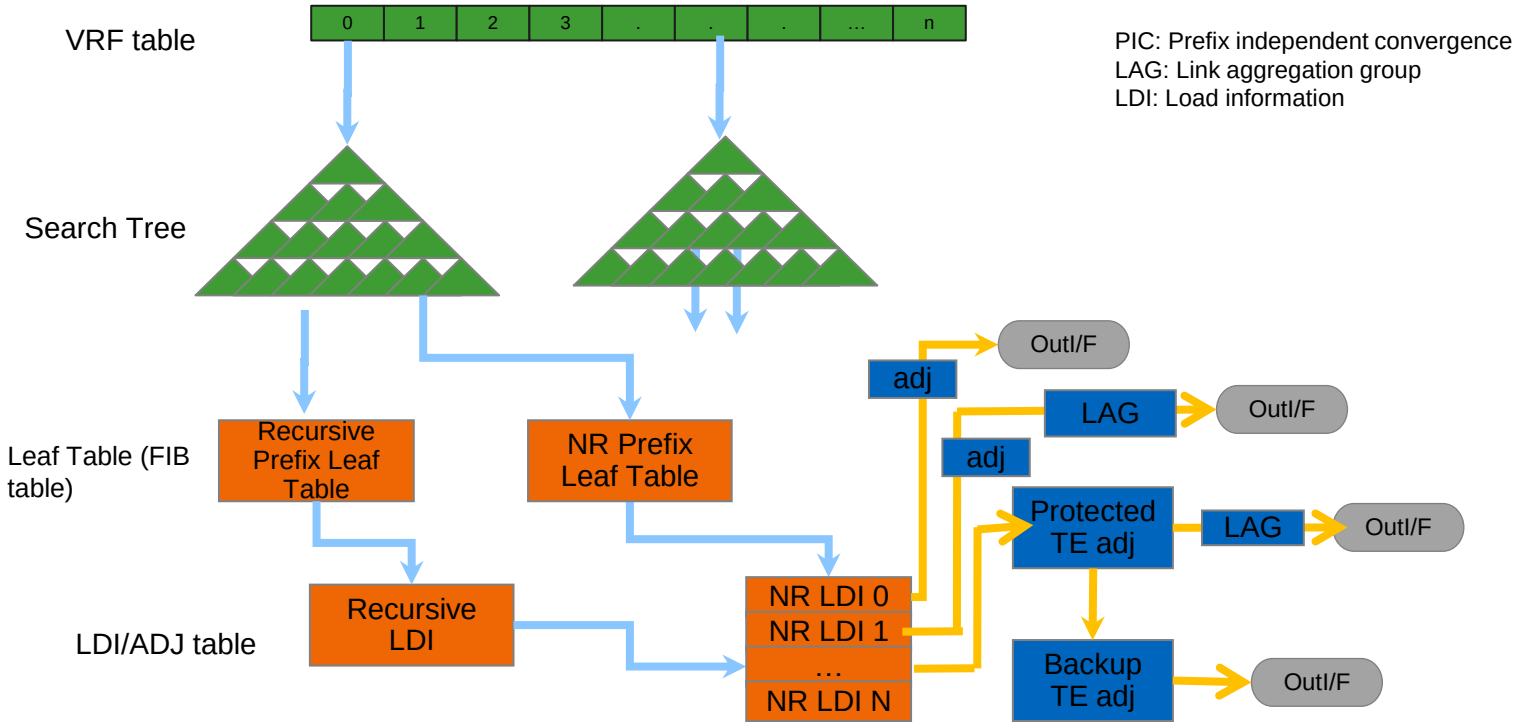
Layer 3 Control Plane Overview



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

Hierarchical Layer 3 Forwarding Data Structure

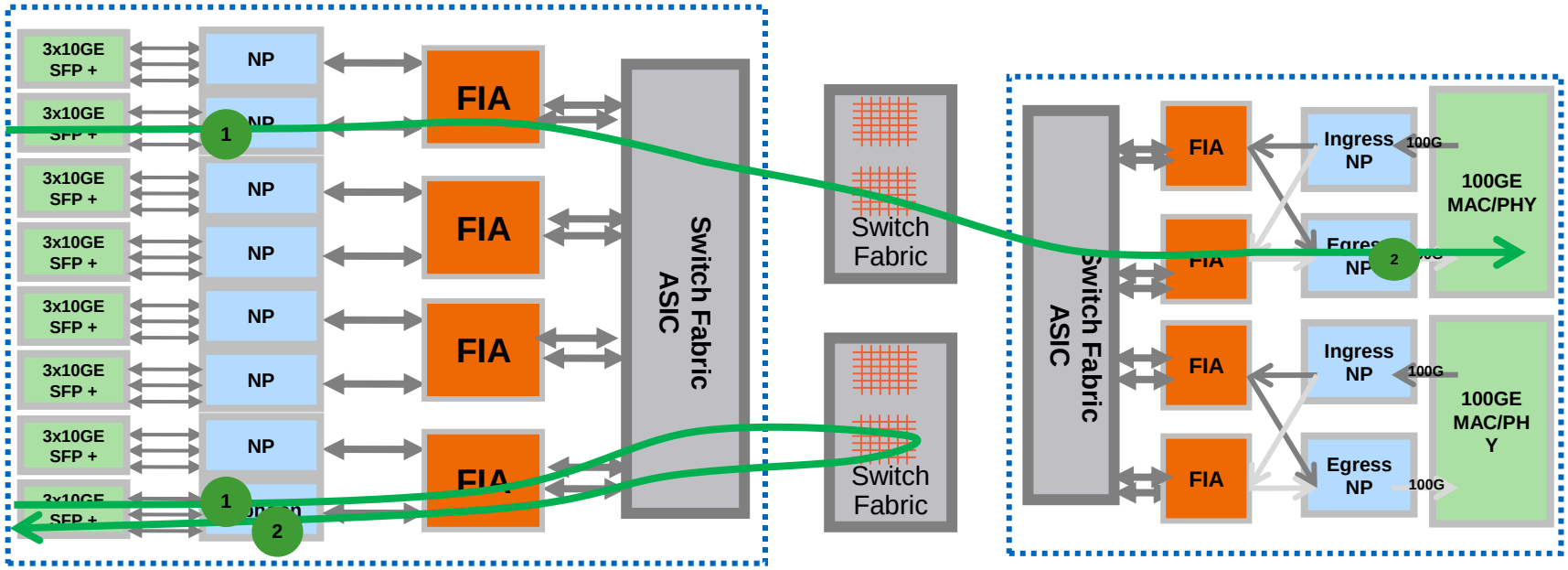
Enabling Prefix Independent Convergence for TE FRR, BGP, LAG



PIC: Prefix independent convergence
LAG: Link aggregation group
LDI: Load information

IOS-XR Two-Stage Forwarding Overview

Scalable and Predictable



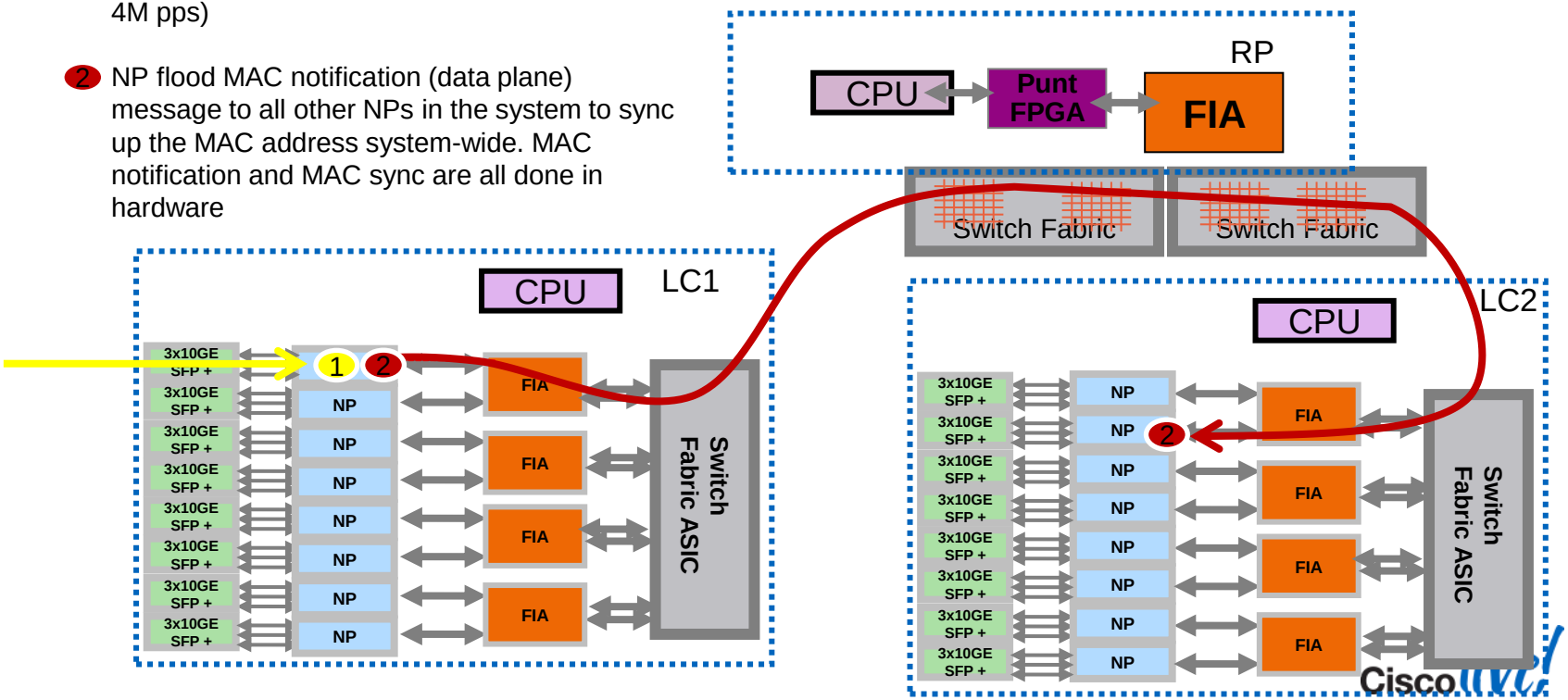
Uniform packet flow for simplicity and predictable performance

MAC Learning and Sync

Hardware based MAC learning: ~4Mpps/NP

1. NP learn MAC address in hardware (around 4M pps)

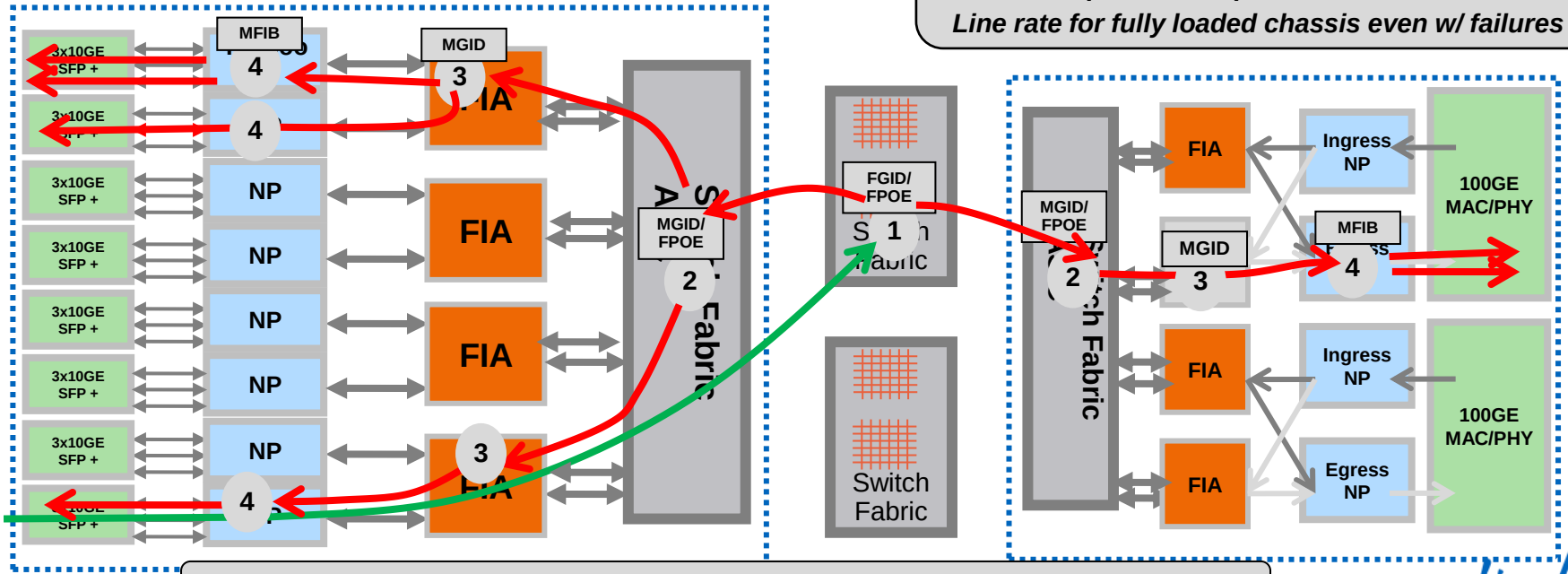
2. NP flood MAC notification (data plane) message to all other NPs in the system to sync up the MAC address system-wide. MAC notification and MAC sync are all done in hardware



Multicast Replication Overview

1. Fabric to LC Replication
2. LC fabric to FIA Replication
3. FIA to NP Replication
4. NP to egress port Replication

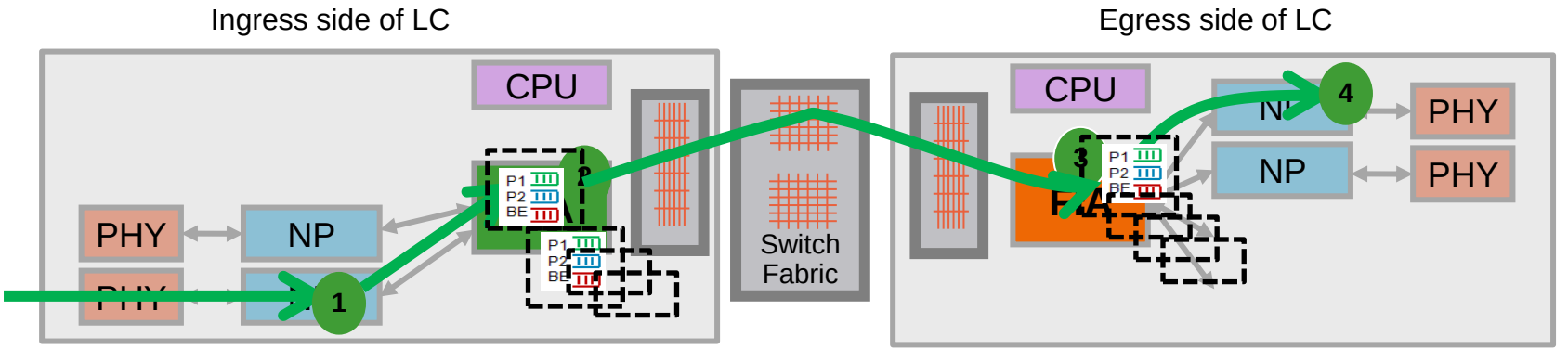
Efficient: replicate only where required
Simple: clean architecture with common codepaths
Consistent: predictable performance
Line rate for fully loaded chassis even w/ failures



Uniform packet flow for simplicity and predictable performance

Internal System QoS Overview

End-to-End priority (P1,P2, 2xBest-effort) propagation
Unicast VOQ and back pressure
Unicast and Multicast separation



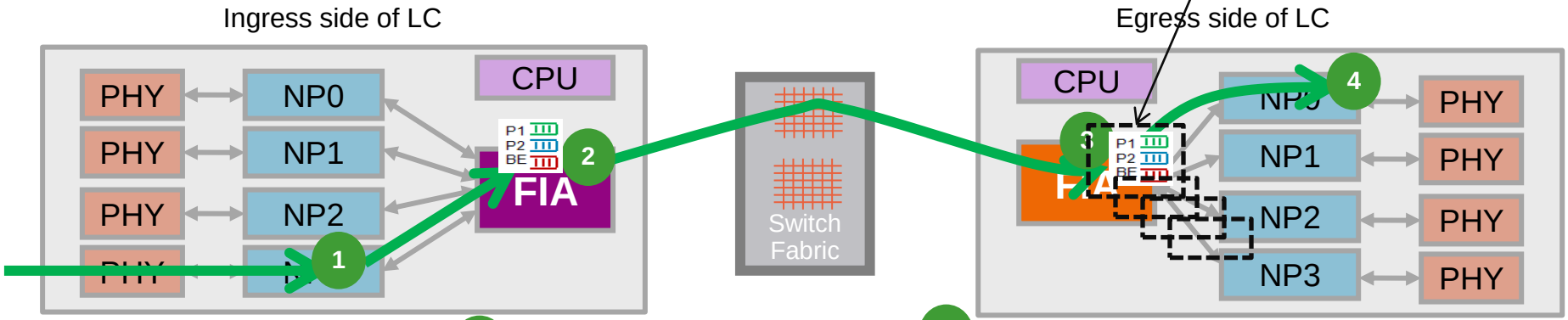



QOS Architecture

System QoS Refresh

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

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



1 Ingress (sub-)interface QoS Queues

2 Virtual Output Queues

3 Egress FIA Queues

4 Egress (sub-)interface QoS Queues

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

Implicit Configuration
Two strict high priority + Normal priority

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

Arbitration & Fabric QoS

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

System QoS Refresh (3) – Backpressure and VoQ Mechanism

Egress NP congestion → → backpressure to ingress FIA →

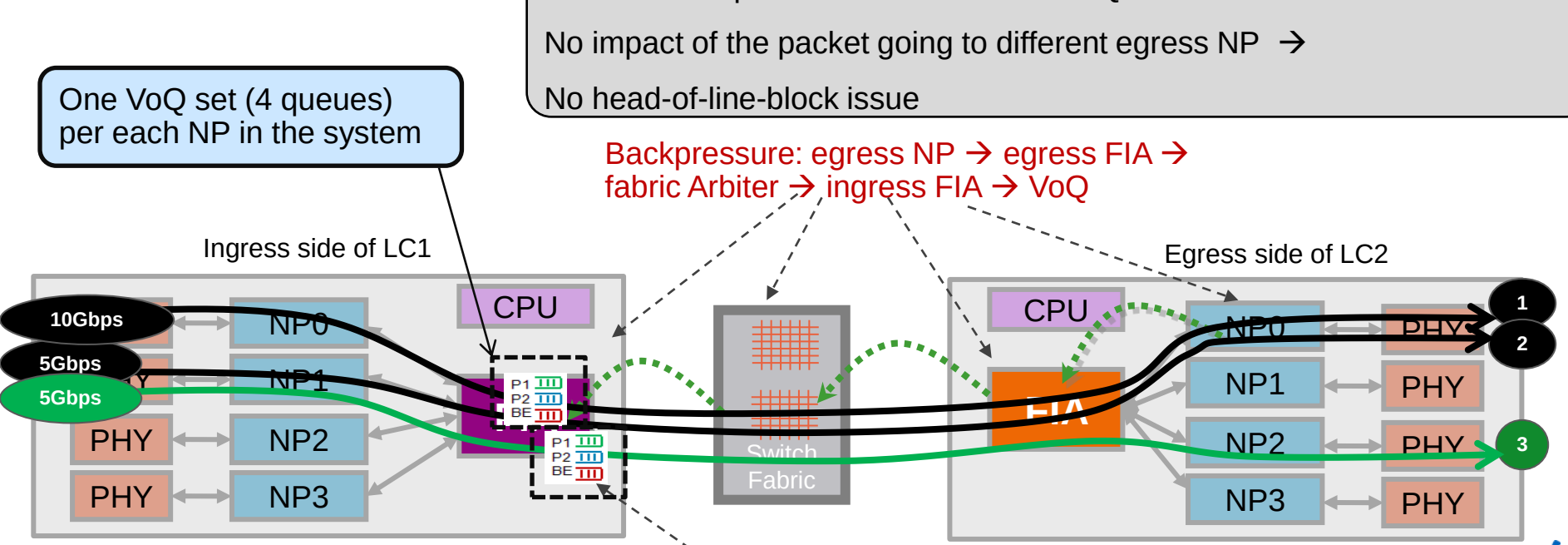
Packet is en-queued in the dedicated VoQ →

No impact of the packet going to different egress NP →

No head-of-line-block issue

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

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



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

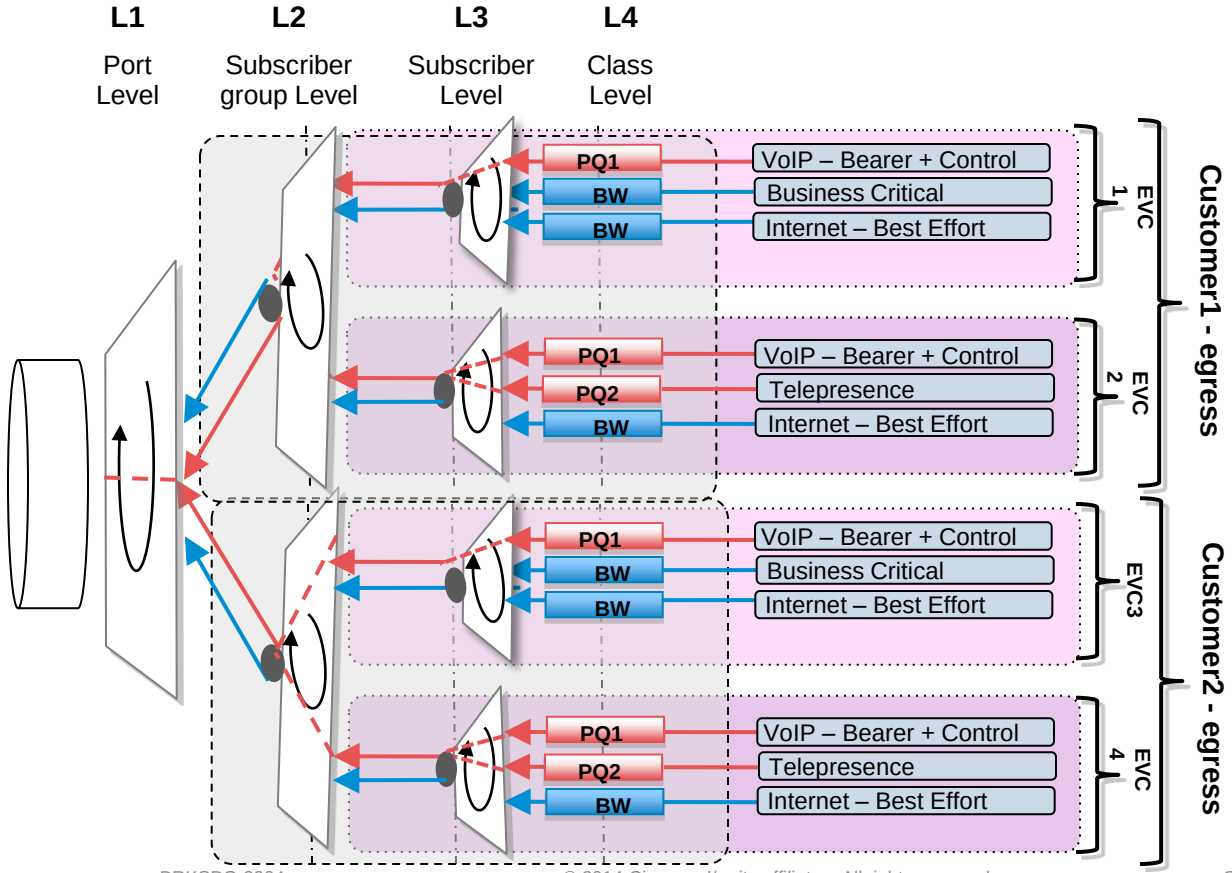
MQC to System QOS Mapping

- ASR 9000 supports traffic differentiation at all relevant points within the system
  - Priority awareness at all interfaces: P1 > P2 > Low/Normal
- Classification into these priorities is based on ***input MQC classification*** on the **ingress** linecard into P1, P2, Other
  - a packet classified into a **P1 class** on ingress is mapped to PQ1 system queue
  - a packet classified into a **P2 class** on ingress is mapped to PQ2 system queue
  - a packet classified into a **non-PQ1/2** class on ingress will get mapped to LP queue along the system qos path
- Note: The marking is implicit once you assign a packet into a given queue on ingress; its sets the fabric header priority bits onto the packet.
  - no specific “set” action is required – the priority level is taken from the MQC class configuration

ASR 9000 QOS Implicit Trust

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

4 Layer Hierarchy Overview



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

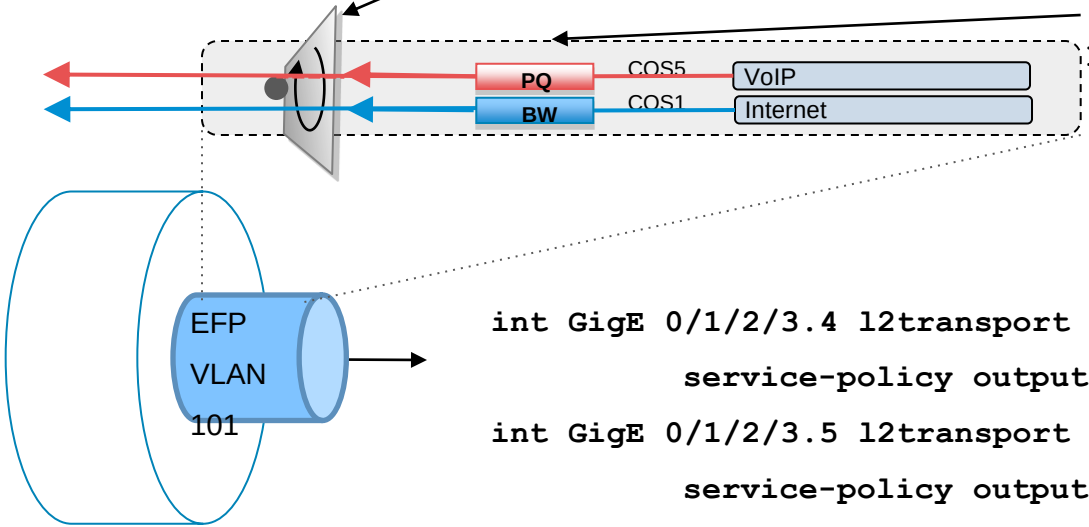
L1 level is not configurable but is implicitly assumed

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

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

3 Layer Hierarchy Example

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




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

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

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

ASR9K QoS Classification Criteria

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

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

ASR9K QoS - Classification Formats

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

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

ASR9K QoS - Packet Marking Details

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

ASR9K QoS - Policing Details

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

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

Normal Hierarchical Policer

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

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

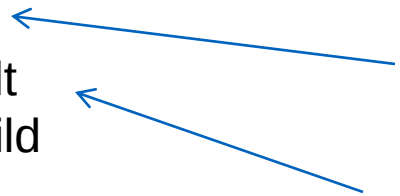
At parent level, if it's over the CIR, packet will be dropped randomly. There is no awareness which packet to be dropped

Conform Aware Policer

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

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

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



Parent CIR must > aggregated child CIR

Parent police only support 1R2C, child police support all: 1R2C, 2R3C, or 1R3C

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

ASR 9000 QoS - Queue Scheduling

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

Show/debug QOS Commands (Reference Only)

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

Show/debug QoS Commands (Reference Only)

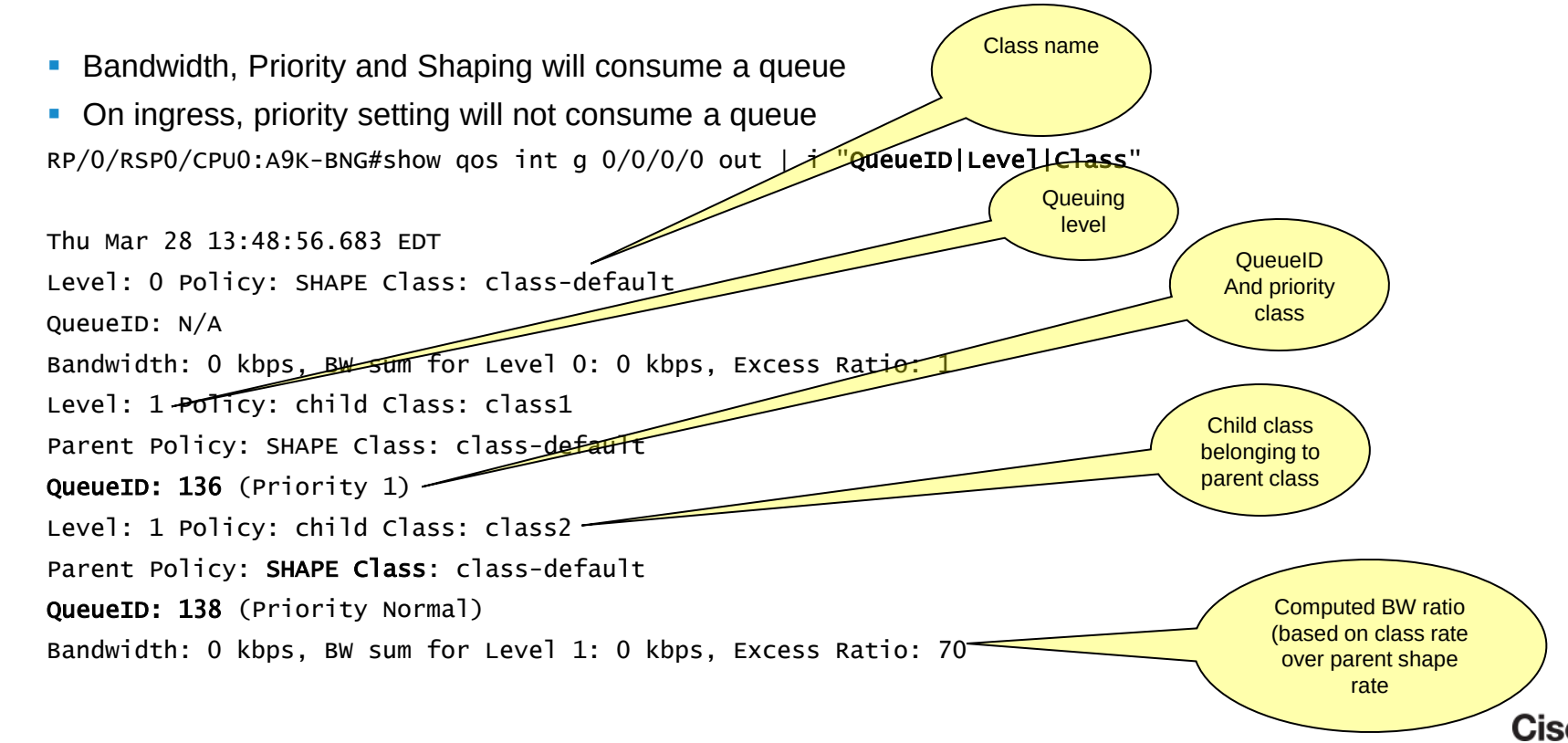
show qoshal resource summary [np <np>]	Displays the summary of all the resources used in hardware and software for QoS such number of policy instances, queues, profiles
show qoshal fcu <limits|status|profile>	Displays all Traffic Manager (TM) Flow control related info
show qoshal ha chkpt <all|<chkpt-tbl-name>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

What Consumes a Queue

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

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

Thu Mar 28 13:48:56.683 EDT
Level: 0 Policy: SHAPE Class: class-default
QueueID: N/A
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 1
Level: 1 Policy: child Class: class1
Parent Policy: SHAPE Class: class-default
QueueID: 136 (Priority 1)
Level: 1 Policy: child Class: class2
Parent Policy: SHAPE Class: class-default
QueueID: 138 (Priority Normal)
Bandwidth: 0 kbps, BW sum for Level 1: 0 kbps, Excess Ratio: 70
```



What is Programmed in HW?

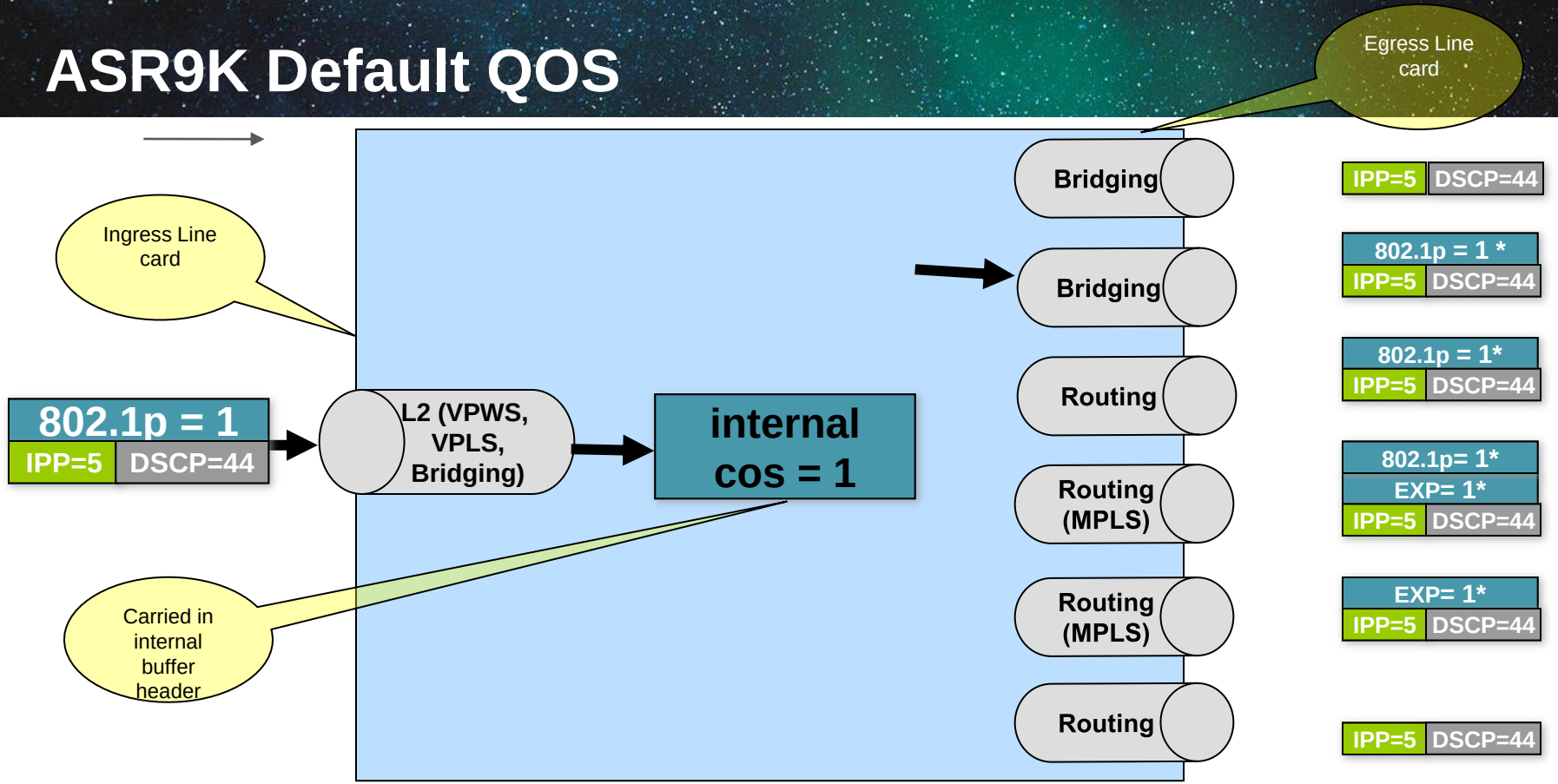
```
COMMAND: show qos interface gigE 0/0/0/0 out
------------------------------------------------------------------------
Level: 0 Policy: xtp Class: class-default
QueueID: N/A
Shape CIR : NONE
Shape PIR Profile : 0/4(S) Scale: 195 PIR: 199680 kbps  PBS: 2496000 bytes
WFQ Profile: 0/9 Committed Weight: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

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

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

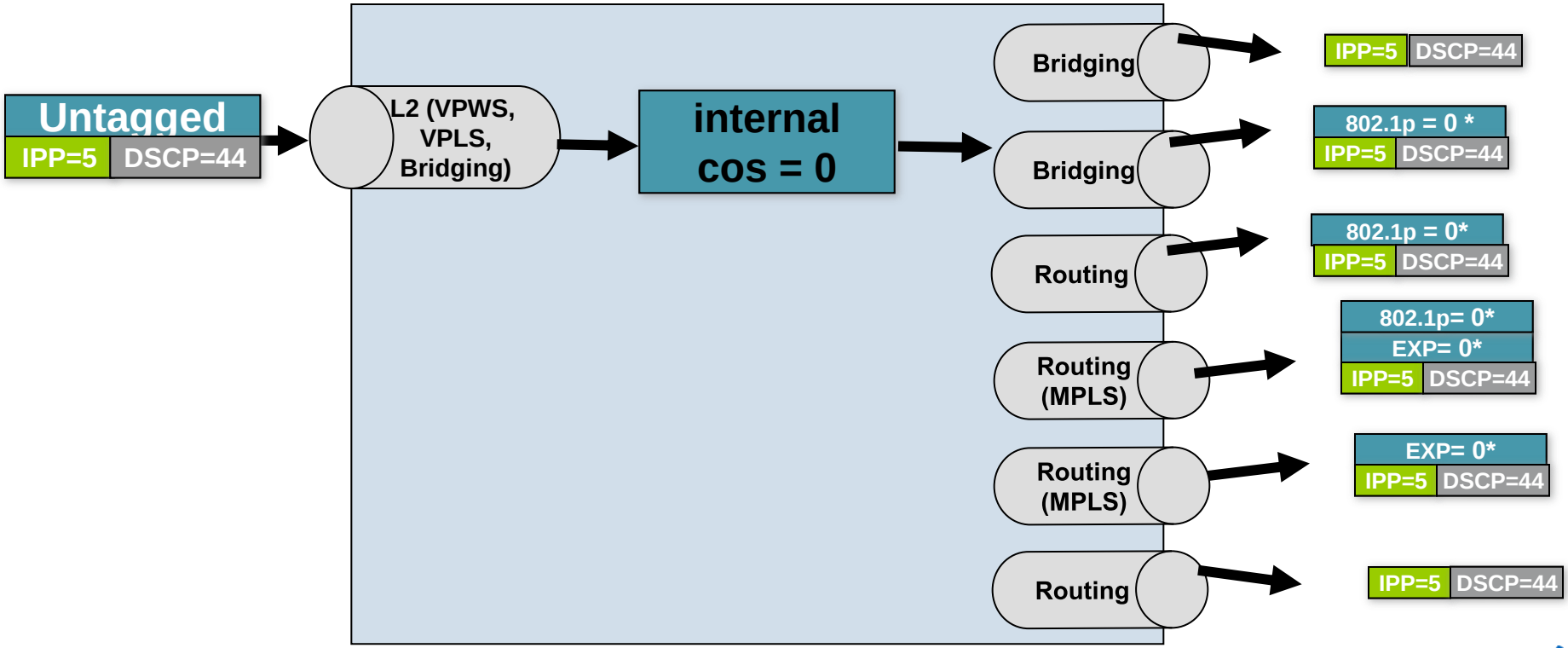
Note that all hardware parameters (WRED, queue, burst sizes, etc exist whether configured or not!

ASR9K Default QOS



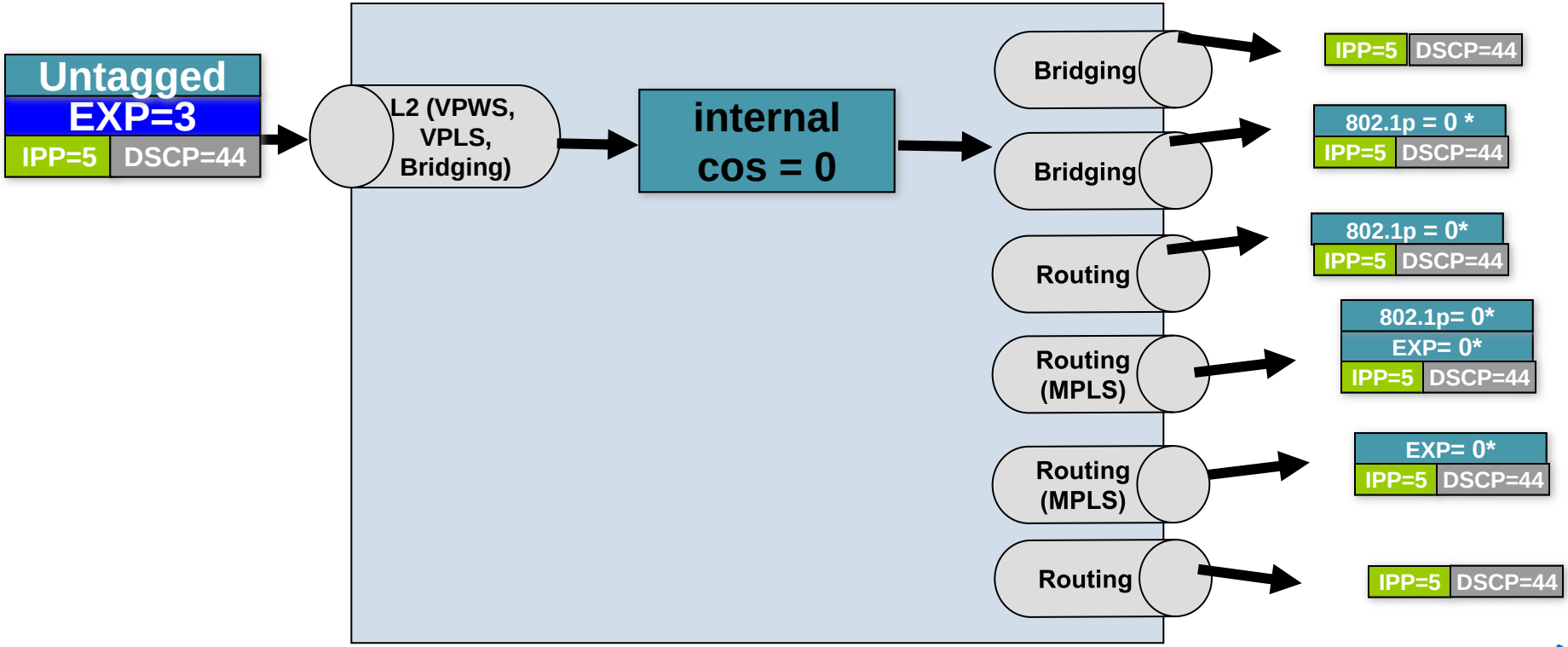
Note: VPWS will be treated like a L2 operation on ingress - Applies for all tags/labels in the stack that get imposed.
Not for VLAN translation. **Bridging on egress without adding an vlan header is an hypothetical case – in case we have a need.**
IPP = IP Precedence, showing IPP & DSCP seperately since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

ASR9K Default QOS



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

ASR9K Default QOS

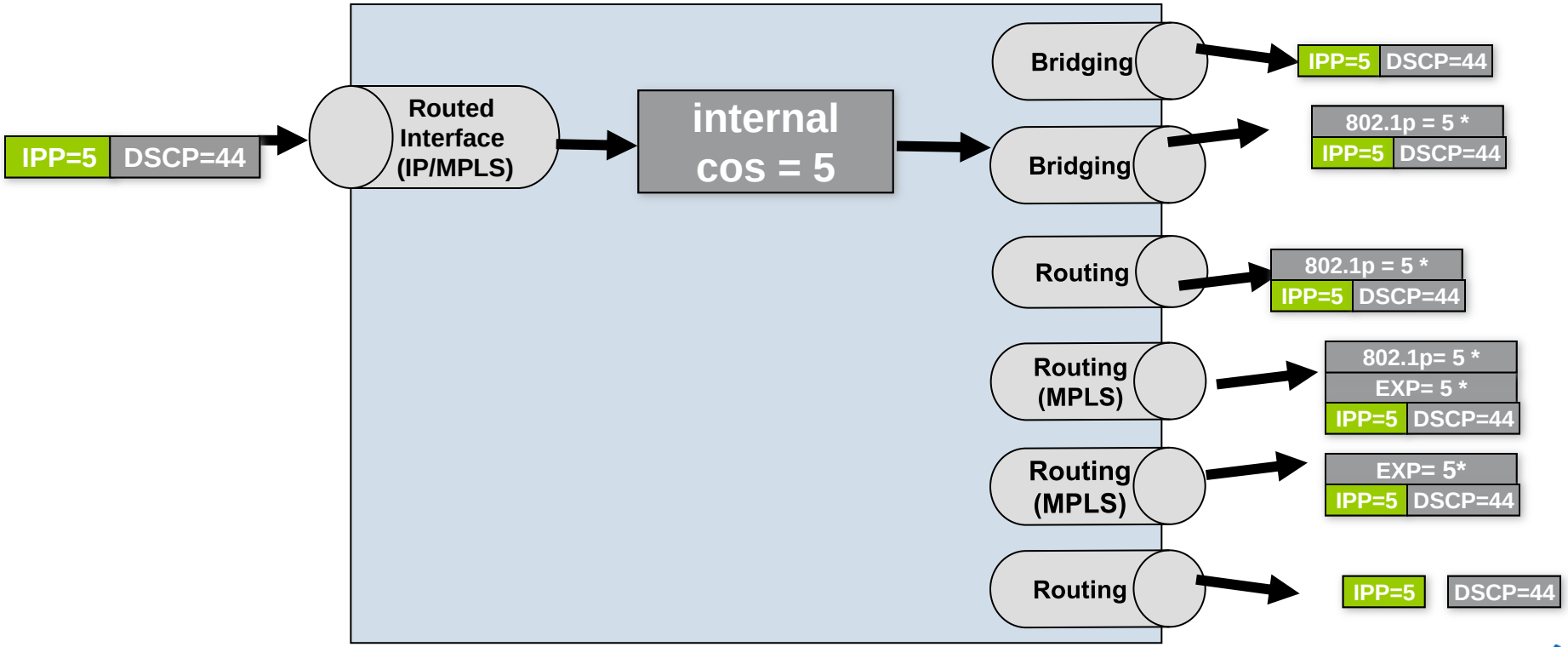


Note: Trust cos in case of bridged interfaces in ingress. For untagged packets use cos = 0.

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

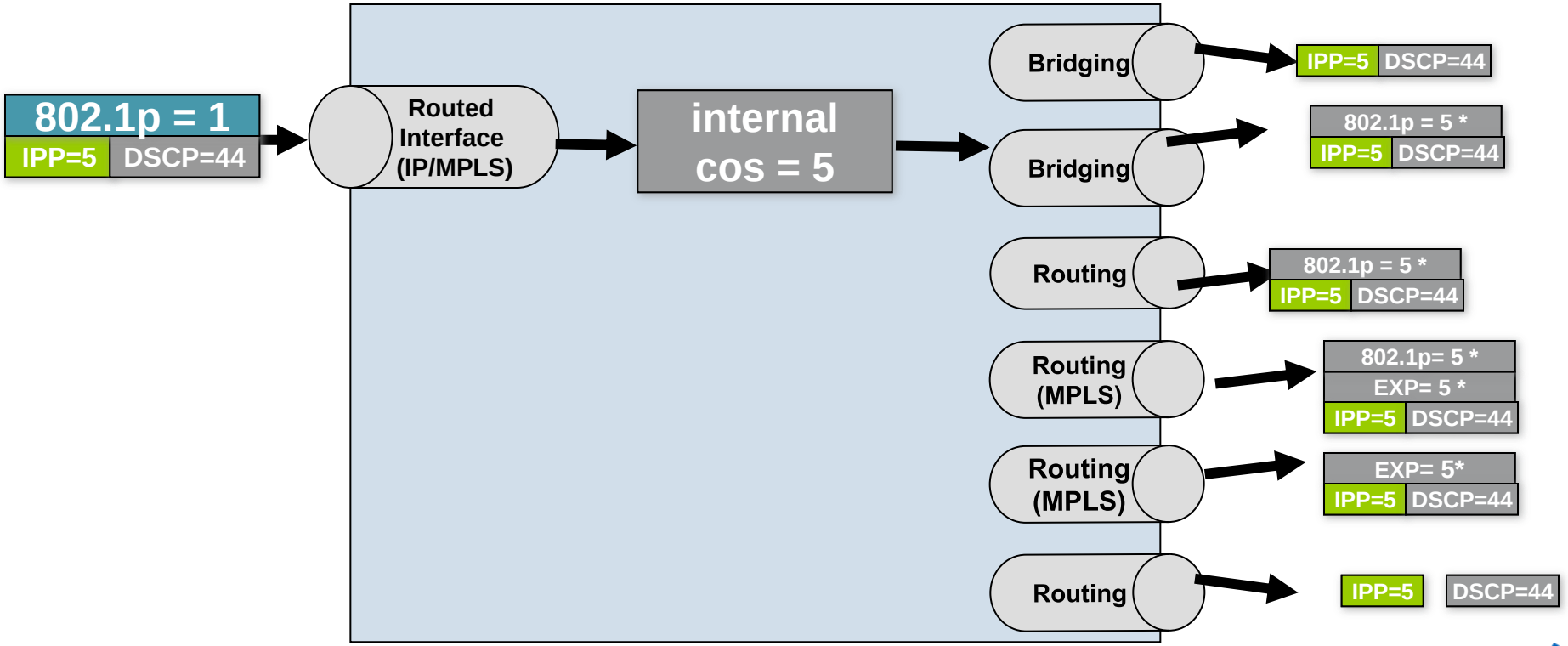
•- Explicit NULL EXP is treated the same as an topmost EXP of non NULL labels.

ASR9K Default QOS



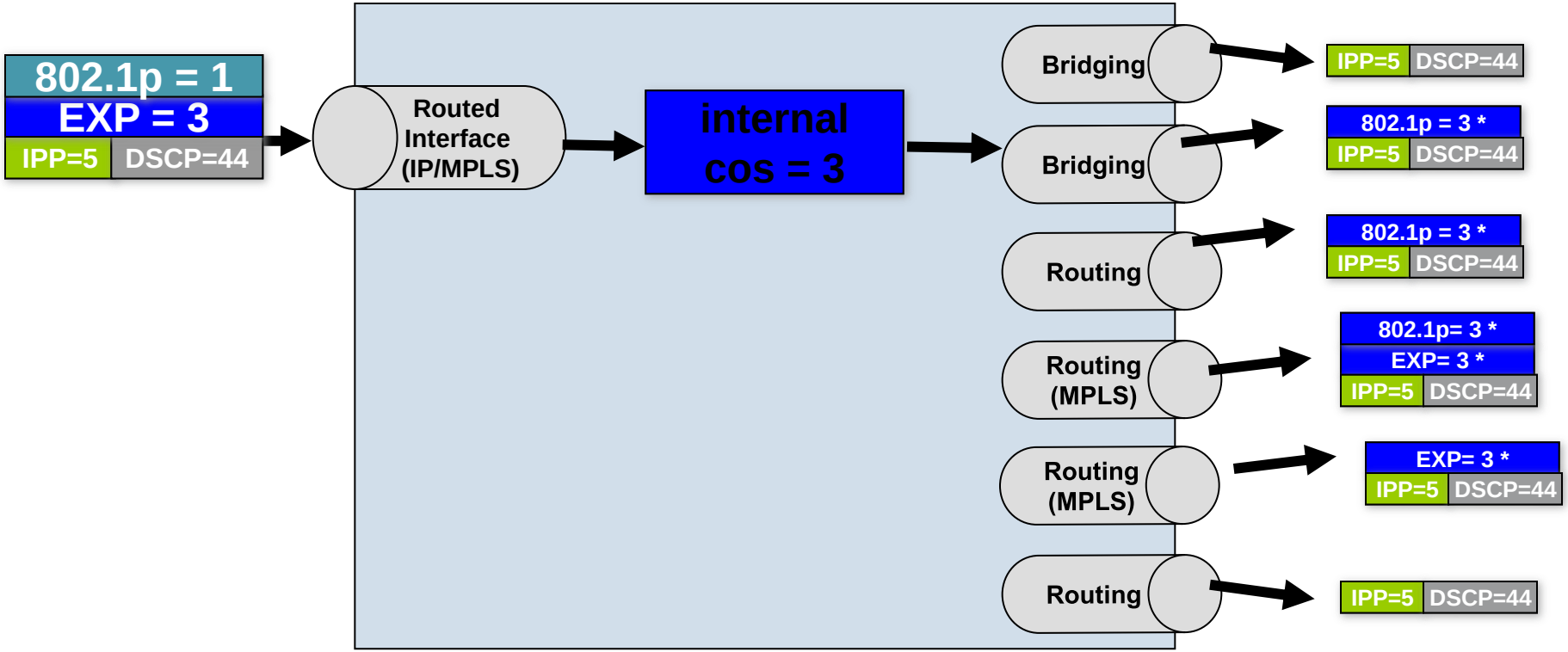
Note: Trust dscp in case of routed interfaces in ingress. For Non IP packets use cos = 0
*** - Applies for all tags/labels in the stack that get imposed.**

ASR9K Default QOS



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

ASR9K Default QOS



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

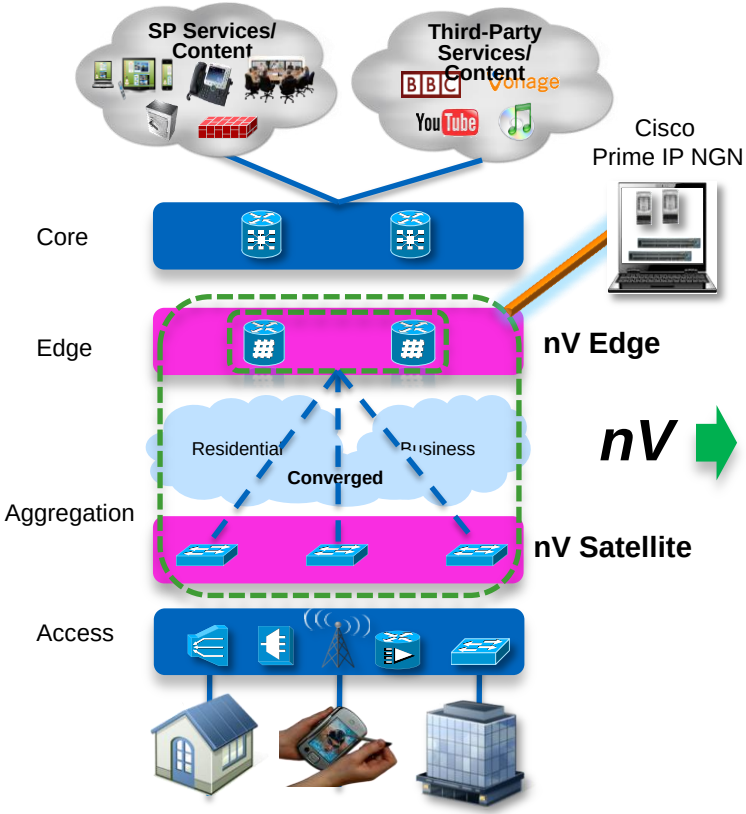
* - Applies for all tags/labels in the stack that get imposed.

ASR9000 nV – Network Virtualisation

ASR 9000 nV Technology Overview

Before: *nV* Technology

- Individual device to manage
- Complex network protocols
- Feature inconsistency, inter-operability
- Physical port limit



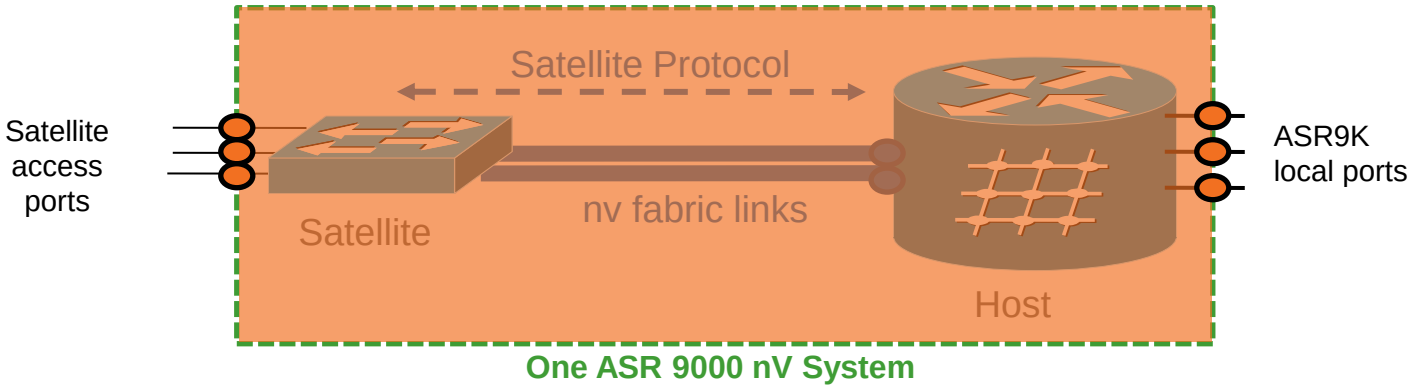
After: *nV* Technology

- One virtual system to manage
- No network protocols within virtual system, Remote satellite is plug-n-play, zero touch
- Single feature set, one release cycle
- Scale to 10 of 1000s physical ports

nV Satellite

ASR 9000 nV Satellite Overview

Plug and Play, Zero Touch Satellite Access Device



- Satellite and ASR 9000 Host run **satellite protocol** for auto-discovery, provisioning and management
- Satellite and Host may be in different locations. There is **no distance limitation** between satellite and Host
- The satellite<->host connection is called "**nv fabric link**", which may be L1 or L2.

Satellite access port have feature parity with ASR9K local ports
→ it works/feels just as local port

Satellite Hardware – ASR 9000v Overview

Power Feeds

- Redundant -48V DC power feeds
- Single 110-240VAC power feed
- Max Power 210W

1 RU ANSI & ETSI Compliant

Field Replaceable Fan Tray

- Redundant Fans
- ToD/PSS Output
- Bits Out

44x10/100/1000 Mbps Pluggables

- Full Line Rate Packet Processing and Traffic Management
- **Copper and fibre SFP optics**
- **Speed/duplex auto negotiation**

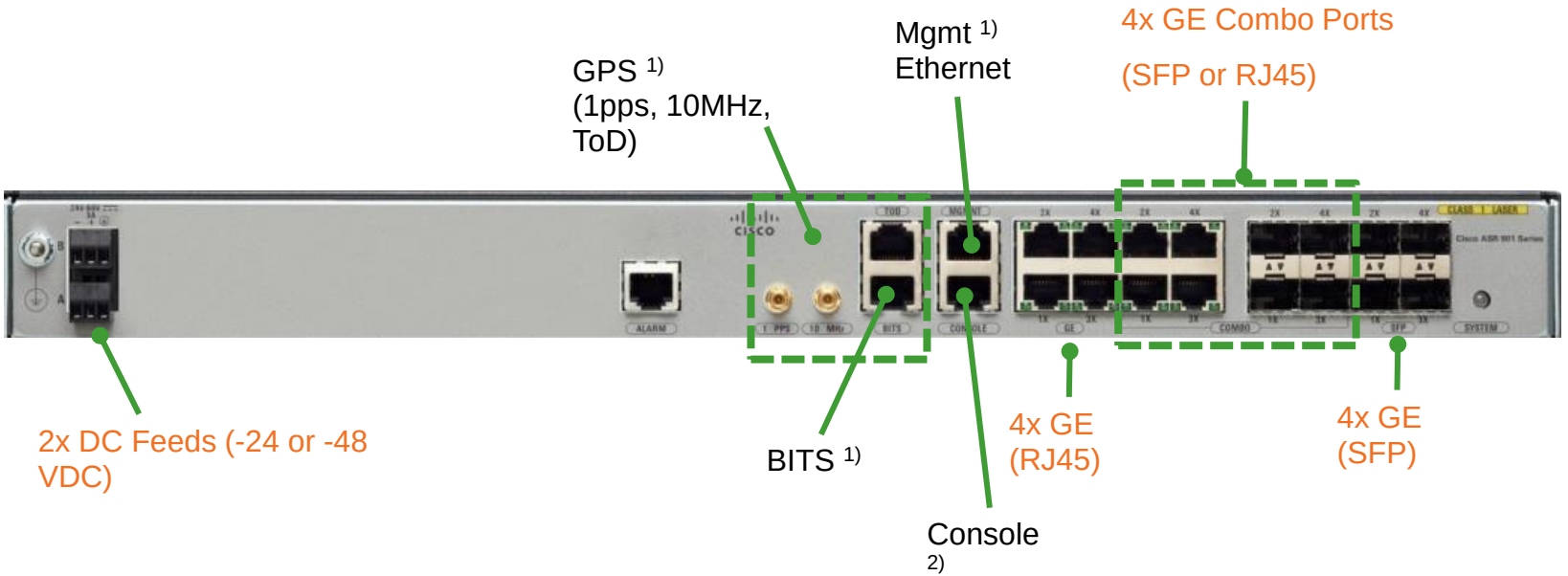
4x10G SFP+

- Initially used as Fabric Ports ONLY (could be used as access port in the future)
- **Copper and fibre SFP+ optics**

Industrial Temp Rated

- -40C to +65C Operational Temperature
- -40C to +70C Storage Temperature

Satellite Hardware – ASR901 Overview

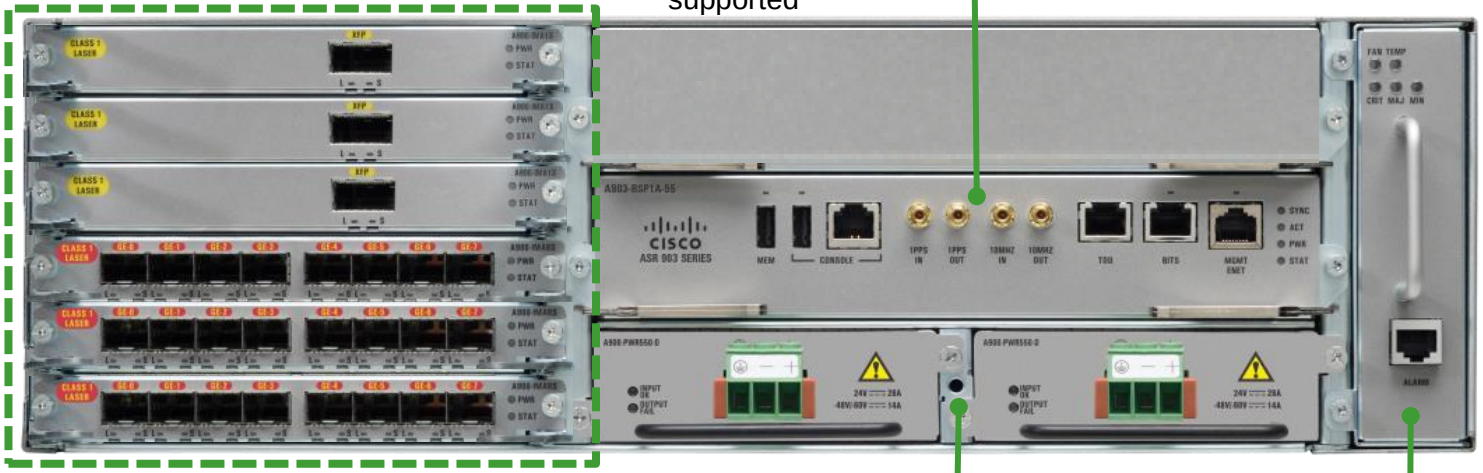


1) Not supported/used when operating in nV Satellite Mode
2) Used for low level debugging only

Satellite Hardware – ASR903 Overview

Route Switch Processor

- Currently only 1x RSP supported



Six I/O Modules

- 1 port 10GE Module (XFP) – nV fabric links only
- 8 port 1GE Module (SFP) – access ports only
- 8 port 1GE Module (RJ45) – access ports only

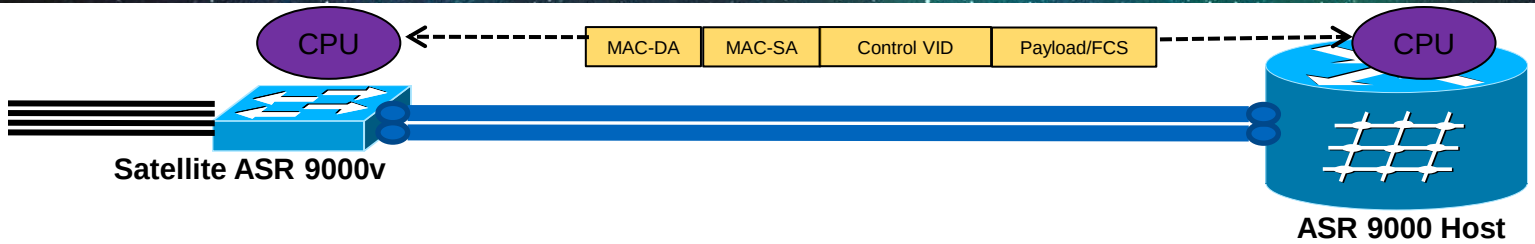
2x Power Modules

- DC PEM, 1x -24 or -48 VDC
- AC PEM, 1x 115..230 VAC

Fan Module

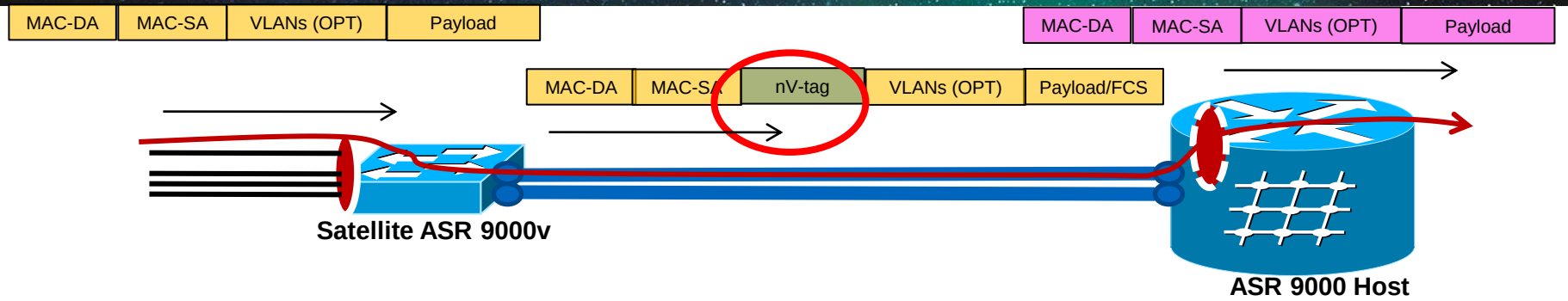
Satellite – Host Control Plane

Satellite Discovery and Control Protocol



- Discovery Phase
  - A **CDP-like** link-level protocol that discovers satellites and maintains a periodic heartbeat
  - **Heartbeat** sent once every second, used to detect satellite or fabric link failures. CFM based fast failure detection plan for future release
- Control Phase
  - Used for **Inter-Process Communication** between Host and Satellite
  - Cisco proprietary protocol over TCP socket, it could get standardised in the future
  - **Get/Set style messages** to provision the satellites and also to retrieve notifications from the satellite

Satellite – Host Data Plane Encapsulation



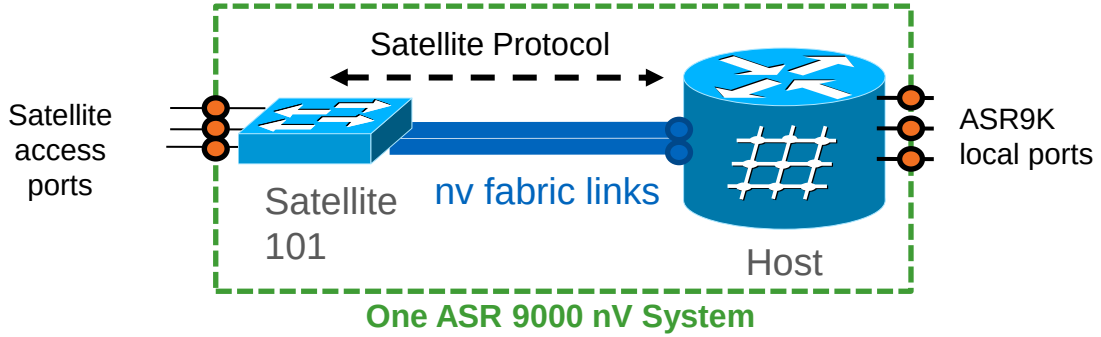
On the Satellite

- Satellite receives Ethernet frame on its access port
- **Special nV-tag** is added
- **Local xconnect** between access and fabric port (no MAC learning !)
- Packet is put into fabric port egress queue and transmitted out toward host

On the Host

- Host receives the packet on its satellite fabric port
- **Checks the nV tag**, then maps the frame to the corresponding satellite virtual access port
- Packet Processing identical to local ports (L2/L3 features, qos, ACL, etc all done in the NPU)
- Packet is forwarded out of a local, or satellite fabric port to same or different satellite

Initial Satellite Configuration

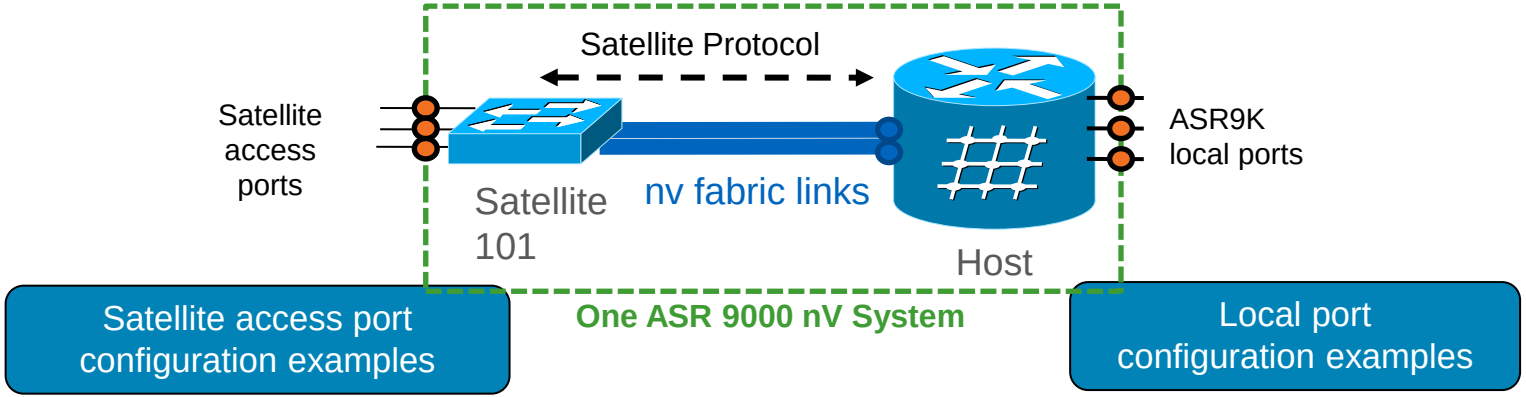



```
nv
 satellite 101 ← define satellite
  type asr9000v
  ipv4 address 10.0.0.101


interface TenGigE 0/2/0/2 ← configure satellite fabric port
  ipv4 point-to-point
  ipv4 unnumbered Loopback100
  nv
    satellite-fabric-link satellite 101
     remote-ports ← satellite to fabric port mapping
       GigabitEthernet 0/0/0-9
```

Satellite Port Configuration

Comparison to Local Port Configuration




```
interface GigabitEthernet 101/0/0/1
  ipv4 address 1.2.2.2 255.255.255.0

interface TenGig 101/0/0/1.1
  encapsulation dot1q 101
  rewrite ingress tag pop 1 sym

interface Bundle-ethernet 200
  ipv4 address 1.1.1.1 255.255.255.0

interface GigabitEthernet 101/0/0/2
  bundle-id 200
```

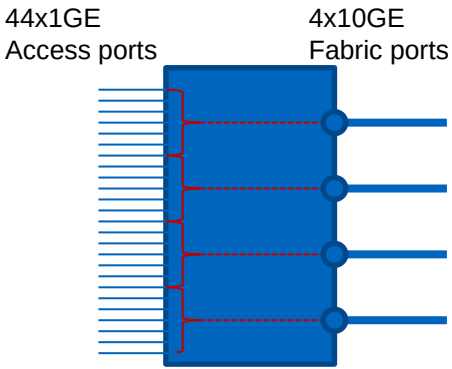
```
interface GigabitEthernet 0/0/0/1
  ipv4 address 2.2.2.2 255.255.255.0

interface TenGig 0/0/0/1.1
  encapsulation dot1q 101
  rewrite ingress tag pop 1 sym

interface Bundle-ethernet 100
  ipv4 address 1.1.1.1 255.255.255.0

interface GigabitEthernet 0/0/0/2
  bundle-id 100
```

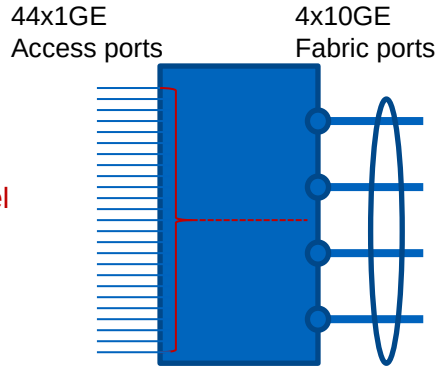
Satellite Deployment Models
ASR9000v Example



Mode 1: Static pinning
No fabric port redundancy

- Access ports are mapped to a single Fabric Link
- Fabric Link failure does bring Access Port down

It can mix model 1 and 2 on the same satellite



Mode 2: Fabric bundle
Fabric port redundancy

- Fabric links are forming a Link-Bundle
- Access port traffic is “hashed” across Bundle Members
- Fabric link failure keeps all Access Ports up, re-hashing of Traffic

Satellite Monitoring and Troubleshooting

- Normal operation, like show CLIs are done on the Host directly, for example
  - Satellite inventory reporting, environmental monitoring
  - Interface counts, stats
  - SNMP MIB
  - NMS support, including ACT, ANA/ PRIME
- Low level debug could still be done directly on the satellite device
  - User can telnet into satellite via out-of-band management console, or in-band from Host, and run regular show/debug CLIs

Satellite Software Management

Everything Controlled from the asr9k Host

```
RP/0/RSP0/CPU0:R1#sh install active
  Node 0/RSP0/CPU0 [RP] [SDR: Owner]
    Boot Device: disk0:
    Boot Image: /disk0/asr9k-os-mbi-4.2.1.22K.CSCtz10483-0.0.4.i/0x100305/mbiasr9k-rsp3.vm
    Active Packages:
      disk0:asr9k-px-4.2.1.22K.CSCtz10483-0.0.4.i
      disk0:asr9k-satellite-px-4.2.1.22K ← satellite image PIE
      disk0:asr9k-mini-px-4.2.1.22K
      disk0:asr9k-mpls-px-4.2.1.22K
      disk0:asr9k-mcast-px-4.2.1.22K
      disk0:asr9k-fpd-px-4.2.1.22K
```

Golden satellite image is always there in satellite flash card for image fall back

```
RP/0/RSP0/CPU0:R1#install nv satellite ?
  <100-65534>  Satellite ID
  all          All active satellites

RP/0/RSP0/CPU0:R1#install nv satellite 100 ?
  activate  Install a new image on the satellite, transferring first if necessary
  transfer  Transfer a new image to the satellite, do not install yet

RP/0/RSP0/CPU0:R1#install nv satellite 100 active
```

Satellite Plug and Play

Configure, Install and Ready-to-Go

- Critical Error LED ON → bad hardware, RMA

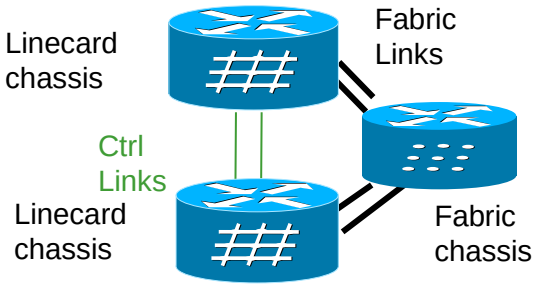
- Major Error LED ON → Unable to connect to ASR9K host
  - Missing the initial satellite configuration?
  - L1 issue, at least one of the uplink port light green?
  - Security check (optional), is the satellite SN# correct?

- Status light green → ready to go, satellite is fully managed by Host



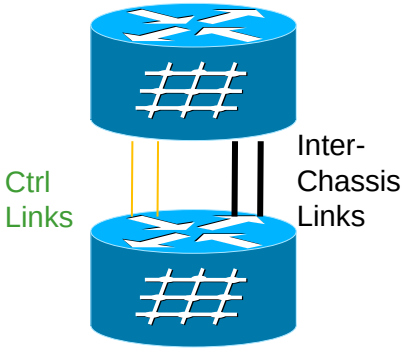

nV Edge

ASR9000 nV Edge Overview



CRS Multi-Chassis

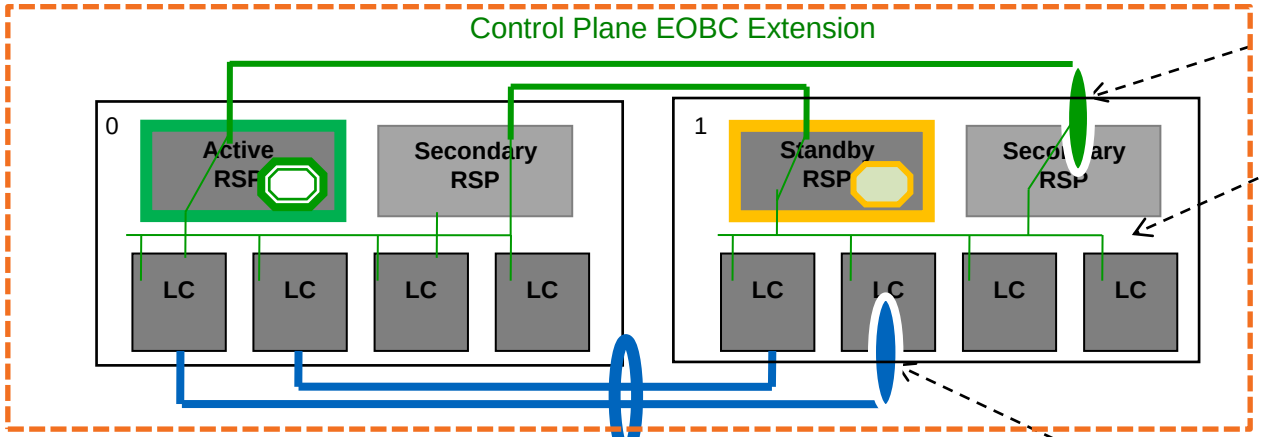
Leverage existing IOS-XR CRS multi-chassis SW infrastructure Simplified/Enhanced for ASR 9000 nV Edge



ASR 9000 nV Edge

Single control plane, single management plane, fully distributed data plane across two physical chassis → one virtual nV system

nV Edge Architecture Details



One Virtual ASR 9000 nV System

Inter-chassis data link (L1 connection)
10G bundle (up to 16 ports)

- Control plane connection: Active RSP and standby RSP are on the different chassis, they communicate via external EOBC links
- Data plane connection: bundle regular data links into special “nV fabric link” to simulate switch fabric function between two physical chassis for data packet
- Flexible co-located or different location deployment (upto 10msec latency)

nV Edge Configuration

- Configure nV Edge globally

```
nv
 edge-system
  serial FOX1437GC1R rack 1
  serial FOX1439G63M rack 0
```

← static mapping of chassis serial# and rack#

- Configure the inter-chassis fabric(data plane) links

```
interface TenGigE1/2/0/0
   nv edge interface
interface TenGigE0/2/0/0
   nv edge interface
```

- NO need to configure the inter-chassis control plane EOBC ports. It's plug-and-play ☺

After this configuration, rack 1 will reload and then join cluster after it boot up
Now you successfully convert two standalone ASR 9000 into one ASR 9000 nV Edge
As simple as this !!!

nV Edge Interface Numbering

- Interfaces on 1st Chassis (Rack 0)

```
GigabitEthernet0/1/1/0          unassigned      Up                    Up
GigabitEthernet0/1/1/1.1        unassigned      Shutdown              Down
...
```

- Interface on 2nd Chassis (Rack 1)

```
GigabitEthernet1/1/1/0          unassigned      Up                    Up
GigabitEthernet1/1/1/1.22       unassigned      Shutdown              Down
...
```

- Interfaces on a Satellite connected to the nV Edge Virtual System

```
GigabitEthernet100/1/1/0        unassigned      Up                    Up
GigabitEthernet100/1/1/1.123    unassigned      Up                   Up
...
```

nV Edge System Monitoring

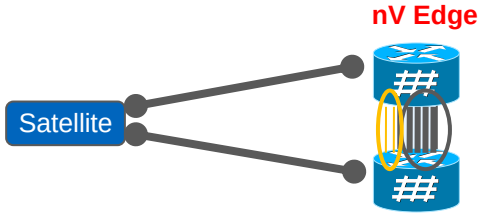
```
RP/0/RSP0/CPU0:ASR4-Rack0(admin)#show dsc
Thu Apr 12 03:01:12.225 UTC
-------------------------------------------------------------
           Node  (    Seq#)     Role      Serial# State
-------------------------------------------------------------
    0/RSP0/CPU0  (       0)   ACTIVE  FOX1545GRM1 PRIMARY-DSC
    0/RSP1/CPU0  (   31785)  STANDBY  FOX1545GRM1 NON-DSC
    1/RSP0/CPU0  (   31763)  STANDBY  FOX1325G77H NON-DSC
    1/RSP1/CPU0  (   32001)   ACTIVE  FOX1325G77H BACKUP-DSC

RP/0/RSP0/CPU0:ASR4-Rack0#show platform
Thu Apr 12 03:00:32.799 UTC
Node             Type                         State            Config State
--------------------------------------------------------------------------------
0/RSP0/CPU0      A9K-RSP440-SE(Active)        IOS XR RUN       PWR,NSHUT,NMON
0/RSP1/CPU0      A9K-RSP440-SE(Standby)       IOS XR RUN       PWR,NSHUT,NMON
0/0/CPU0         A9K-2x100GE-TR               IOS XR RUN       PWR,NSHUT,MON
0/1/CPU0         A9K-MOD160-TR                IOS XR RUN       PWR,NSHUT,NMON
0/1/0            A9K-MPA-2X40GE               DISABLED         PWR,SHUT,MON
0/1/1            A9K-MPA-20X1GE               OK               PWR,NSHUT,MON
0/3/CPU0         A9K-SIP-700                  IOS XR RUN       PWR,NSHUT,MON
0/3/0            SPA-8XOC12-POS               OK               PWR,NSHUT,MON
0/3/1            SPA-2XCHOC12/DS0             OK               PWR,NSHUT,MON
0/3/2            SPA-2XOC48POS/RPR            OK               PWR,NSHUT,MON
1/RSP0/CPU0      A9K-RSP440-SE(Standby)       IOS XR RUN       PWR,NSHUT,MON
1/RSP1/CPU0      A9K-RSP440-SE(Active)        IOS XR RUN       PWR,NSHUT,MON
```

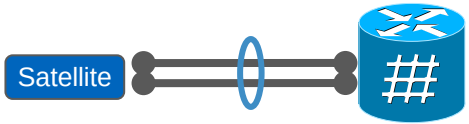
nV Topologies

Single-homed, static pinning

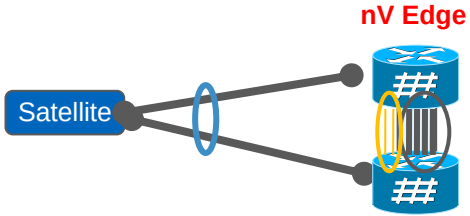
Dual-homed to nV Edge, static pinning



Single-homed, fabric bundle



Dual-homed to nV Edge, fabric bundle



EVC: Ethernet Virtual Circuits

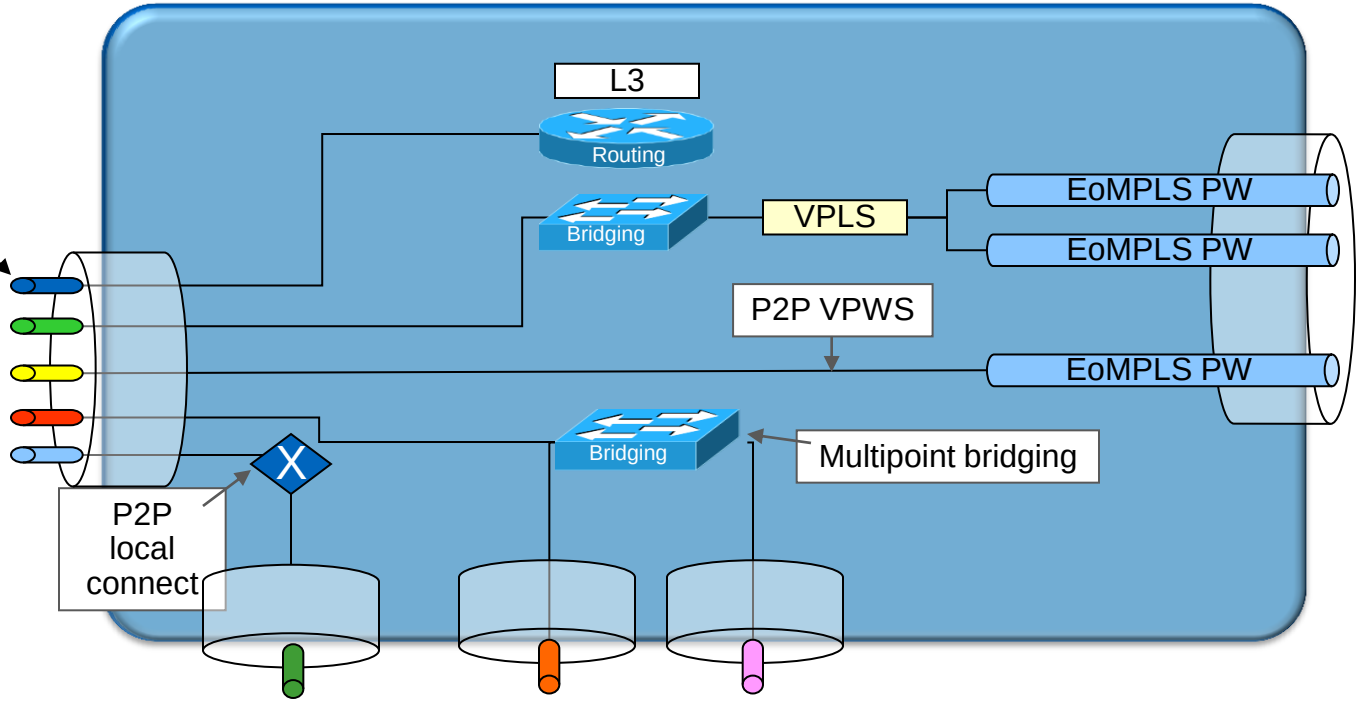
ASR 9000 Flexible Ethernet Infrastructure
("EVC" SW Infrastructure)

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

Flexible VLAN tag classification

Flexible VLAN tag rewrite

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



Flexible service mapping and multiplexing

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

Flexible VLAN Tag Classification

```
RP/0/RSP0/CPU0:PE2-asr(config)#int gig 0/0/0/4.100 l2transport
RP/0/RSP0/CPU0:PE2-asr(config-subif)#encapsulation ?
  default   Packets unmatched by other service instances
  dot1ad    IEEE 802.1ad VLAN-tagged packets
  dot1q     IEEE 802.1Q VLAN-tagged packets
  untagged  Packets with no explicit VLAN tag

RP/0/RSP0/CPU0:PE2-asr(config-subif)#encapsulation dot1q 100 ...
  comma  comma
  exact  Do not allow further inner tags

RP/0/RSP0/CPU0:PE2-asr(config-subif)#encapsulation dot1ad 20 dot1q 10-20 ?
  comma  comma
  exact  Do not allow further inner tags
```

- Double tag
- Single tag
- Multiple tag
- Range of tag
- .1q and/or .1ad
- Loose or exact match
- Untagged
- Default tag

Flexible VLAN Tag Rewrite

```
RP/0/RSP0/CPU0:PE2-asr(config)#int gig 0/0/0/4.100 l2transport

RP/0/RSP0/CPU0:PE2-asr(config-subif)#rewrite ingress tag ?
  pop        Remove one or more tags
  push       Push one or more tags
  translate  Replace tags with other tags

RP/0/RSP0/CPU0:PE2-asr(config-subif)#rewrite ingress tag pop ?
  1  Remove outer tag only
  2  Remove two outermost tags

RP/0/RSP0/CPU0:PE2-asr(config-subif)#rewrite ingress tag push dot1q 100 ?
  second-dot1q  Push another Dot1Q tag
  symmetric     All rewrites must be symmetric

RP/0/RSP0/CPU0:PE2-asr(config-subif)#rewrite ingress tag translate ?
  1-to-1  Replace the outermost tag with another tag
  1-to-2  Replace the outermost tag with two tags
  2-to-1  Replace the outermost two tags with one tag
  2-to-2  Replace the outermost two tags with two other tags
```

Pop tag 1 or 2

Push tag 1 or 2

Tag translation

- 1-1
- 1-2
- 2-1
- 2-2

Flexible Service – L2VPN P2P

EFP configuration example

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

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

Interface gig 0/0/0/3.101 l2transport
encapsulation dot1q 102-105
rewrite ingress push dot1q 100 Symmetric
```

L2VPN P2P service configuration example

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

Flexible Service – L2VPN Multi-Point

EFP configuration example

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

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

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

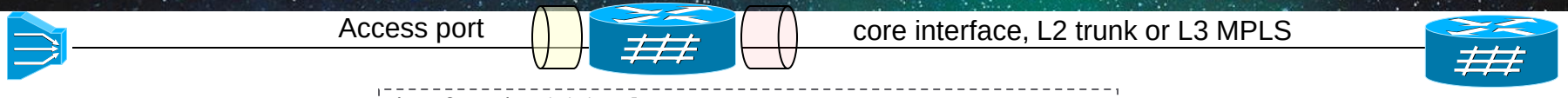
L2VPN MP service configuration example

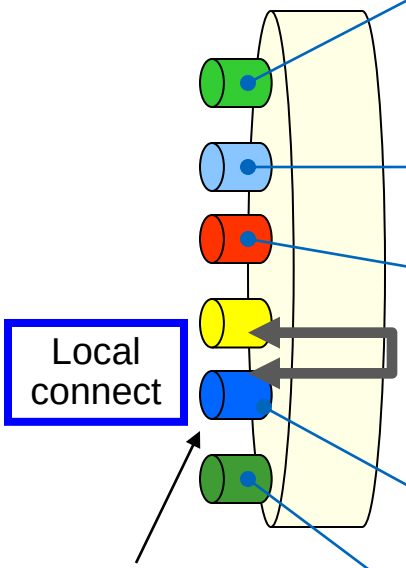
```
l2vpn
 bridge group cisco
  bridge-domain domain1 ← local bridging
   Interface gig 0/0/0/1.101
     split-horizon group ← no bridging among same SHG
   Interface gig 0/0/0/2.101
     split-horizon group

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

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

Multiple Services on the Same Physical Port






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

Local Bridging

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

E-LINE (VPWS)

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

E-LAN (VPLS)

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

Local connect

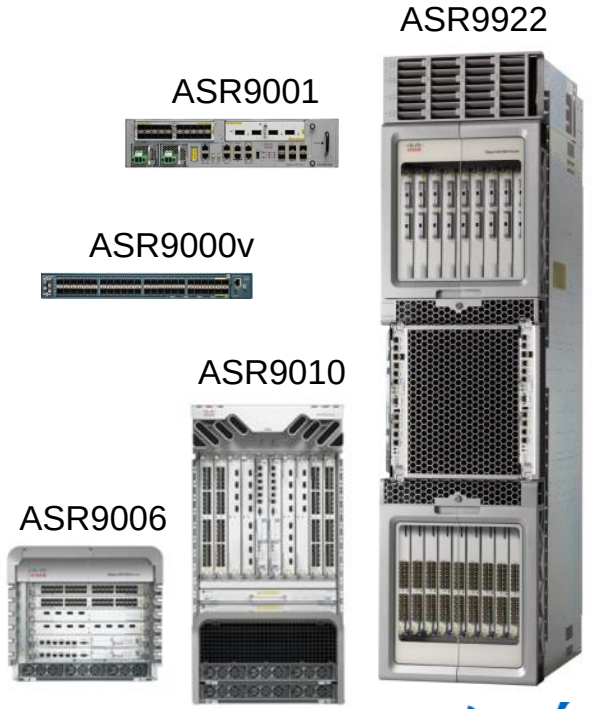
```
interface gig 0/0/0/1.100
  encapsulation dot1q 200 second 200
  ipv4 address 1.1.1.1 255.255.255.0
```

L3 service

Cisco ASR9000 – Next-Gen Edge Routing Platform

Key Design Goals & System Benefits

- Architectural Design for Longevity
- Product Portfolio with significant HW and SW commonality
- Highly integrated Network Processors for High Speed Scale and Feature Flexibility
- Cisco IOS XR based
  - Truly modular, full distributed OS
  - Enhanced for the Edge (L2 and L3)
- nV (Network Virtualisation) for Operational Simplicity






Q & A

Complete Your Online Session Evaluation

Give us your feedback and receive a Cisco Live 2014 Polo Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site www.ciscoliveaustralia.com/mobile
- Visit any Cisco Live Internet Station located throughout the venue

Polo Shirts can be collected in the World of Solutions on Friday 21 March 12:00pm - 2:00pm

Learn online with Cisco Live!

Visit us online after the conference for full access to session videos and presentations.
www.CiscoLiveAPAC.com


CISCO
TM