TOMORROW
starts here.

Cisco ASR 9000 System Architecture

BRKARC-2003

Xander Thuijs CCIE#6775 Principal Engineer

Highend Routing and Optical Group

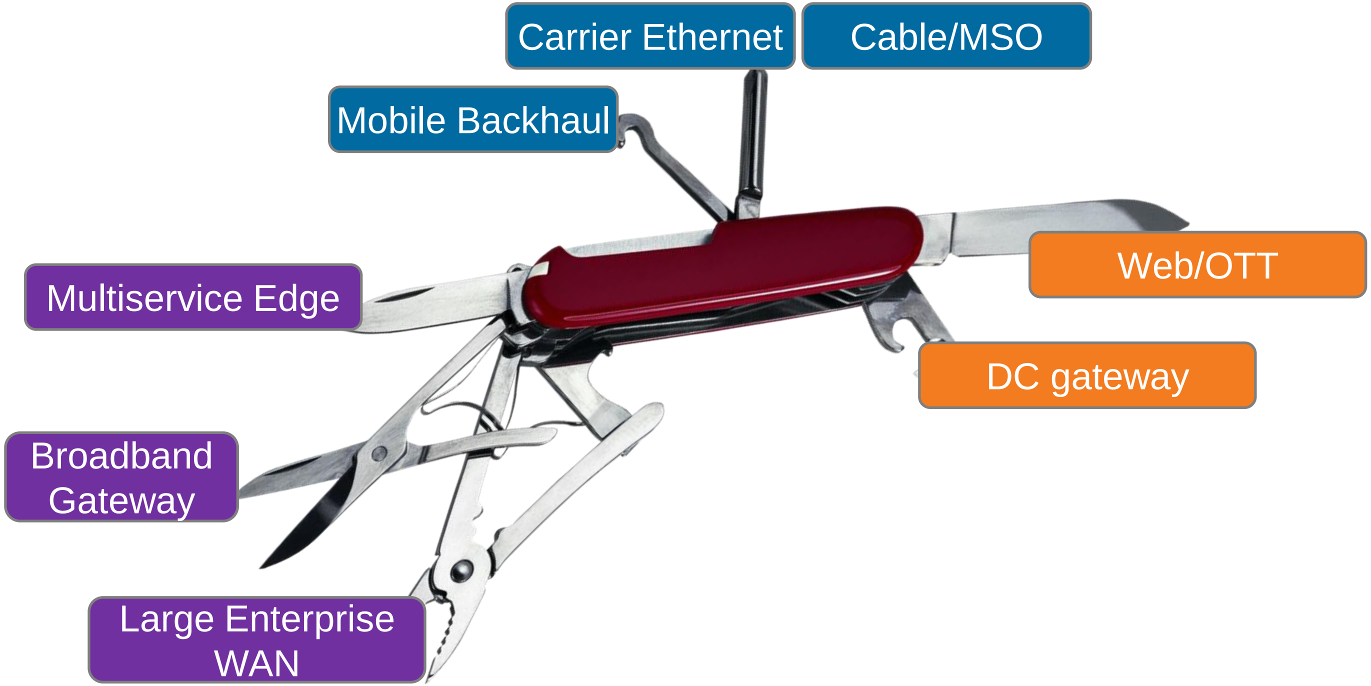
Dennis Cai, Distinguished Engineer, Technical Marketing

CCIE #6621, R&S, Security

Cisco live!

Swiss Army Knife Built for Edge Routing World

Cisco ASR9000 Market Roles



1. High-End Aggregation & Transport

1. Mobile Backhaul
2. L2/Metro Aggregation
3. CMTS Aggregation
4. Video Distribution & Services

2. DC Gateway Router

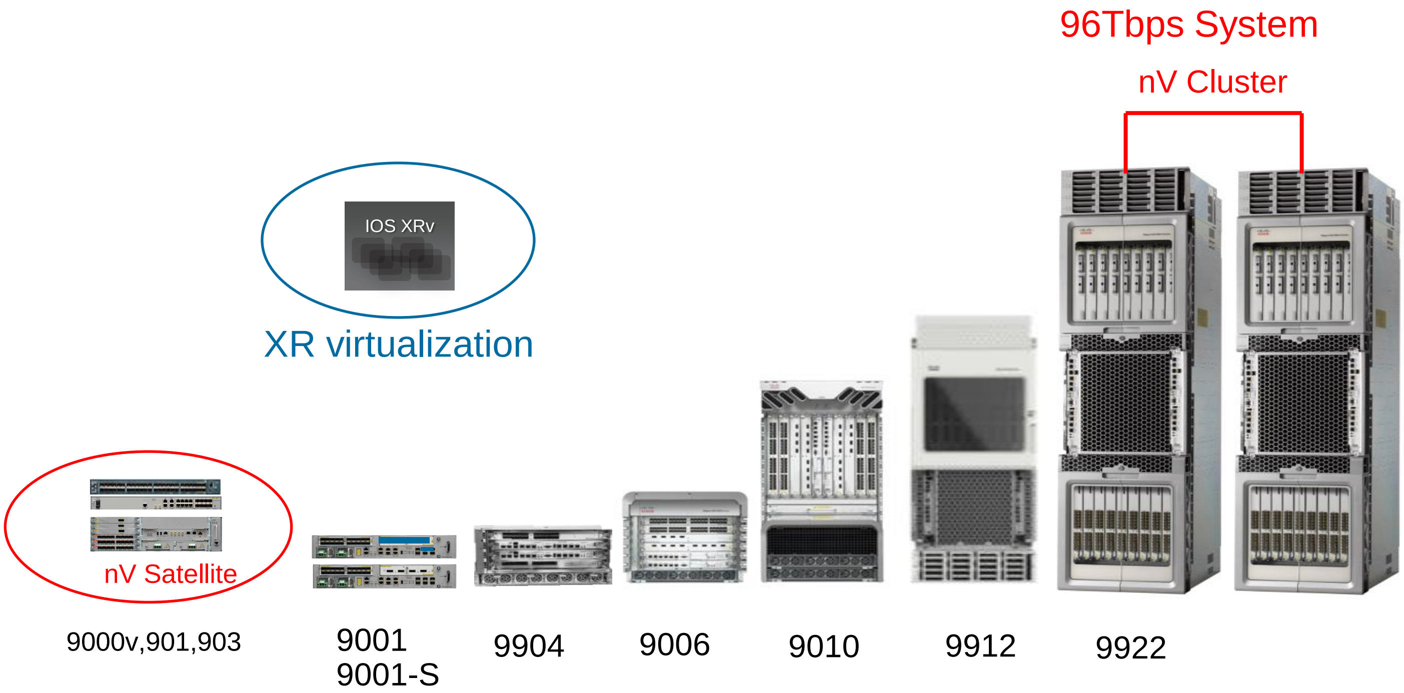
1. DC Interconnect
2. DC WAN Edge
3. WEB/OTT

3. Services Router

1. Business Services
2. Residential Broadband
3. Converged Edge/Core
4. Enterprise WAN

Scalable System Architecture and Portfolio

Physical and Virtual



Other ASR9000 or Cisco IOS XR Sessions

… you might be interested in ☺

- BRKSPG-2904 - ASR-9000/IOS-XR Understanding forwarding, troubleshooting the system and XR operations
- TECSPG-3001: Advanced - ASR 9000 Operation and Troubleshooting
- BRKSPG-2202: Deploying Carrier Ethernet Services on ASR9000
- BRKARC-2024: The Cisco ASR9000 nV Technology and Deployment
- BRKMPL-2333: E-VPN & PBB-EVPN: the Next Generation of MPLS-based L2VPN
- BRKARC-3003: ASR 9000 New Scale Features - FlexibleCLI(Configuration Groups) & Scale ACL's
- BRKSPG-3334: Advanced CG NAT44 and IOS XR Deployment Experience

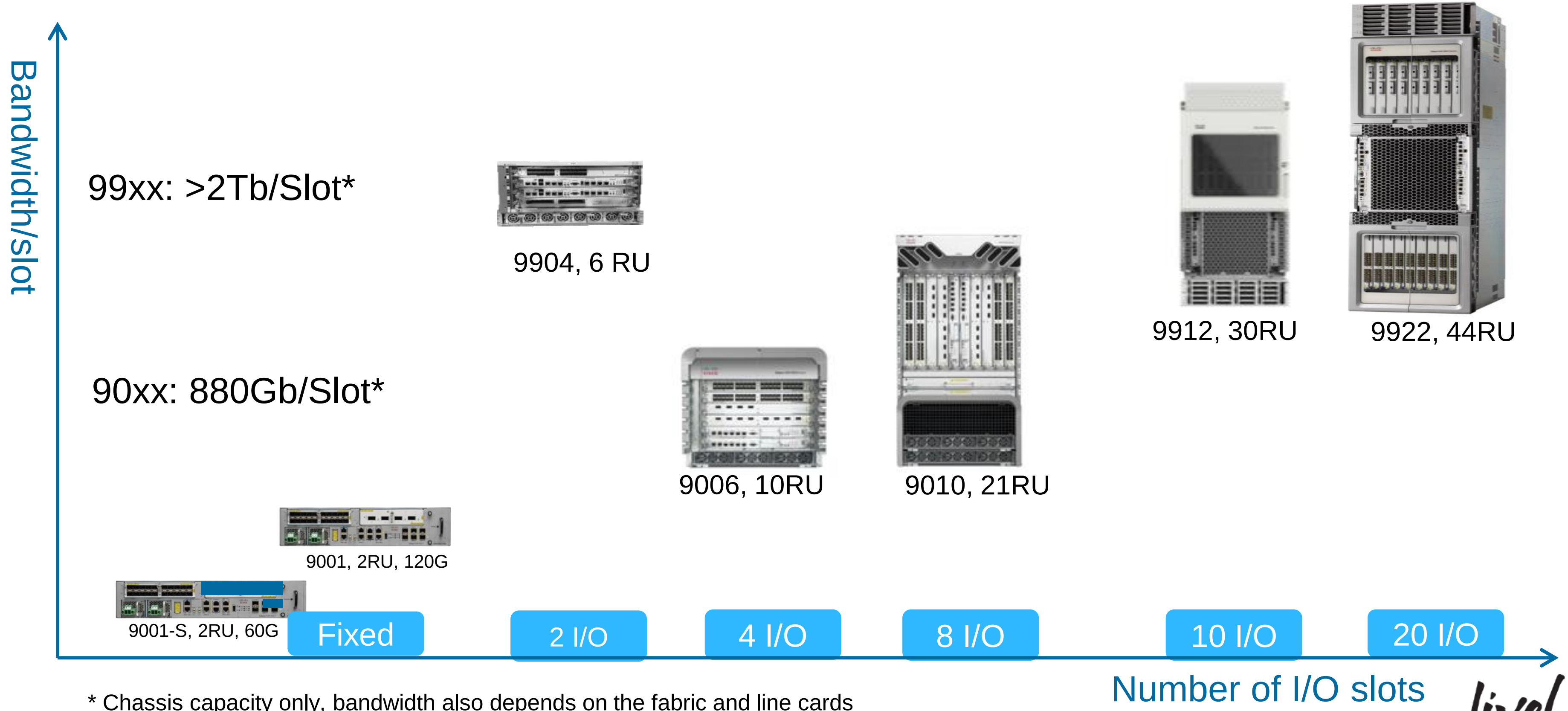
Agenda

- **ASR9000 Hardware System Architecture**
  - HW Overview
  - HW Architecture
- **ASR 9000 Software System Architecture**
  - IOS-XR
  - Control and Forwarding: Unicast, Multicast, L2
  - Queuing
- ASR 9000 Advanced System Architecture
  - OpenFlow
  - nV (Network Virtualization)

ASR9000 Hardware System Architecture (1)
HW Overview

ASR 9000 Chassis Overview

Common software image, architecture, identical software features across all chassis



* Chassis capacity only, bandwidth also depends on the fabric and line cards

ASR 9010 and ASR 9006 Chassis

Shipping since day 1

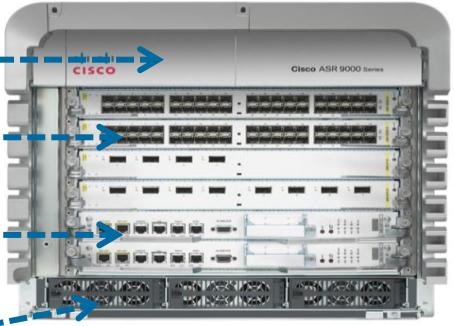
Front-to-back air flow
with air flow baffles,
13RU, vertical

Side-to-back airflow, 10 RU

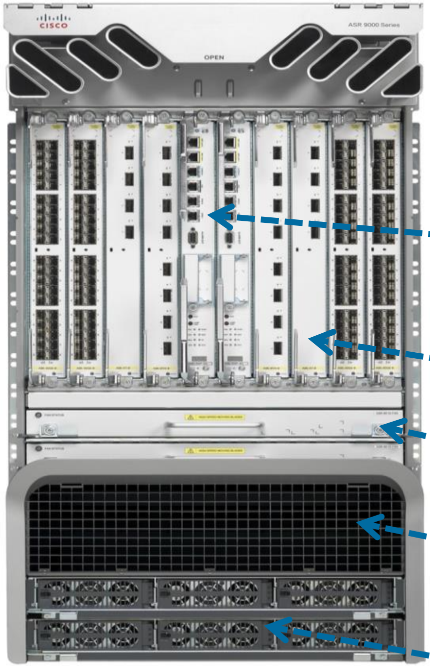
System fan trays
(2x)

Line Card
(0-3)

**RSP (0-1)
(integrated
switch fabric)**



**V1 power shelf: 3 Modular V1 PS
V2 power shelf: 4 Modular V2 PS**



21RU

Front-to-back
airflow

**RSP (0-1)
(integrated
switch fabric)**

Line Card
(0-3, 4-7)

System fan trays
(2x)

Air draw

**2 power shelves
6 V1 or 8 V2 PS**

ASR 9001 Compact Chassis

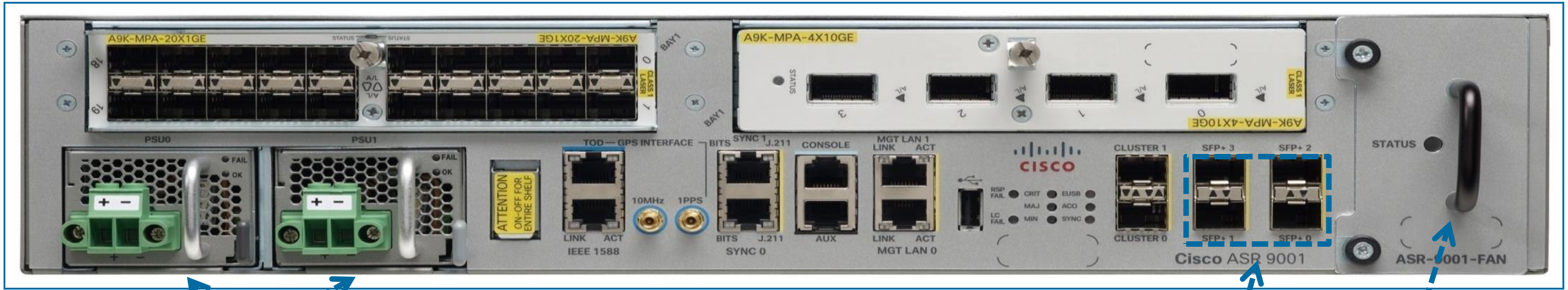
Shipping since IOS-XR 4.2.1
May 2012

Side-to-Side airflow
2RU

Front-to-back air flow with air flow baffles, 4RU, require V2 fan

Sub-slot 0 with MPA

Sub-slot 1 with MPA



Redundant
(AC or DC)
Power Supplies
Field Replaceable

Supported MPAs:

20x1GE
2x10GE
4x10GE
1x40GE

Fixed 4x10G SFP+ ports

Fan Tray
Field Replaceable

ASR 9001-S Compact Chassis

Shipping since IOS-XR 4.3.1
May 2013

Side-to-Side airflow
2RU

Front-to-back air flow with air flow baffles, 4RU, require V2 fan

Supported MPAs:

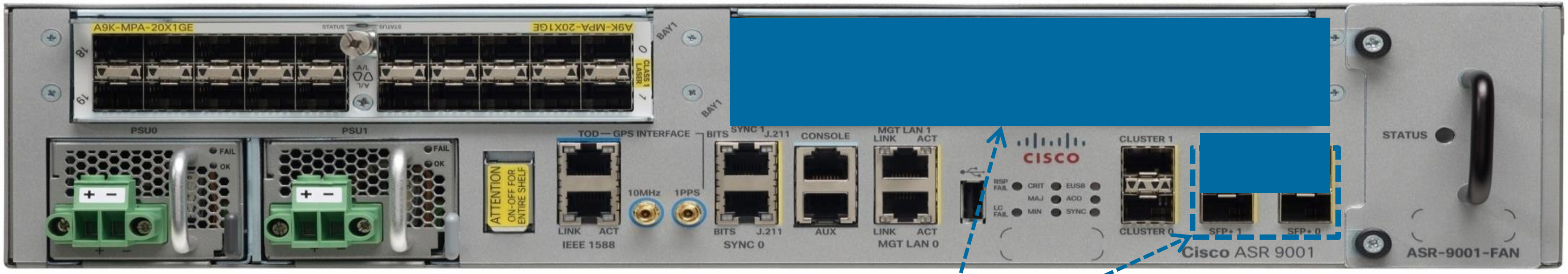
20x1GE
2x10GE
4x10GE
1x40GE

Pay As You Grow

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

Sub-slot 0 with MPA

Sub-slot 1 with MPA

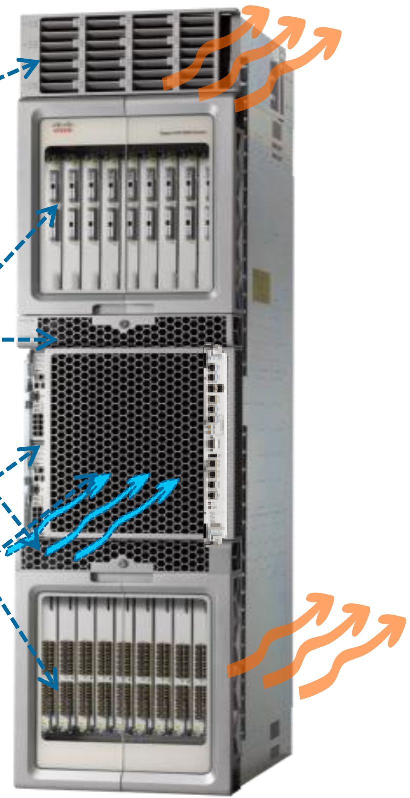


60G bandwidth are disabled by software. SW license to enable it

ASR 9922 Large Scale Chassis

Shipping since IOS-XR 4.2.2
August 2012

Features	Description
Power	4 Power Shelves, 16 Power Modules 2.1 KW DC / 3.0 KW AC supplies N+N AC supply redundancy N:1 DC supply redundancy
Fan	4 Fan Trays Front to back airflow
I/O Slots	20 I/O slots
Rack Size	44 RU
RP	1+1 RP redundancy
Fabric	6+1 fabric redundancy.
Bandwidth	Phase 1: 550Gb per Slot Future: 2+Tb per Slot
SW	XR 4.2.2 – August 2012

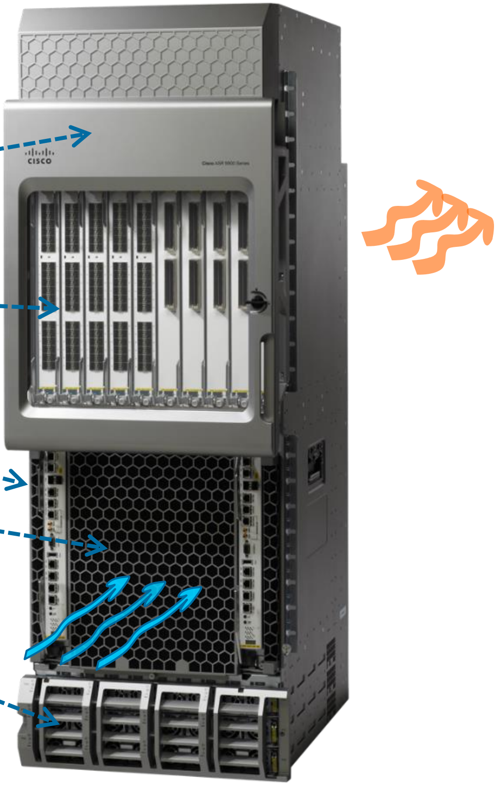


Fully loaded Engineering testbed

ASR 9912 Large Scale 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



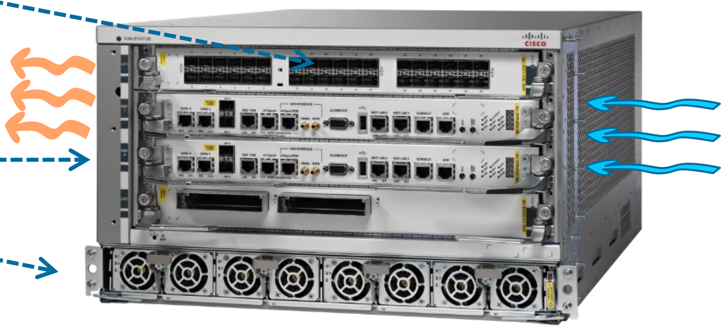
ASR 9904

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 per Slot (440G/slot with existing Line cards) Future capability: 1.7 Tb per Slot
SW	XR 5.1.0 – August 2013

Front-to-back air flow with air flow baffles, 10RU

Side-to-Side airflow
6RU



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

Fan is chassis specific

Power Supply

DC Supplies

A B — 1.5 kW*

A B — 2.1 kW

AC Supplies

A — 3 kW

B — 3 kW

* Version 1 only

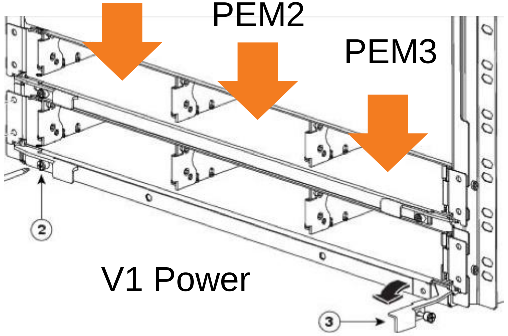
- Single power zone
- All power supplies run in active mode
- Power draw shared evenly
- 50 Amp DC Input or 16 Amp AC for Easy CO Install

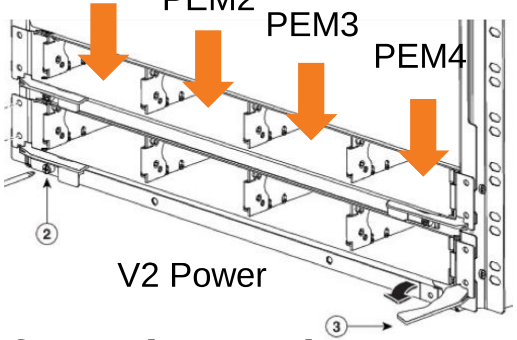
V2 power supply is common across all modular chassis

Version 1 Power vs Version 2 Power System

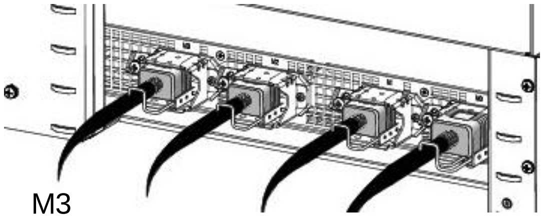
Power Switch:
V1 → in the back
V2 → in the front

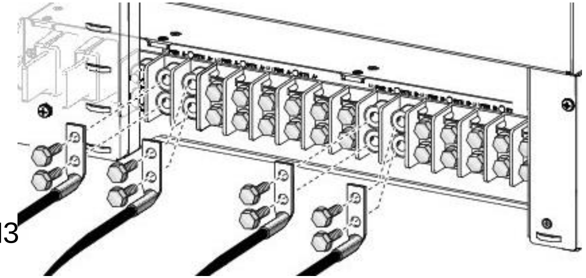
PEM Insertion from the Front





Power Feed Cabling from the Back

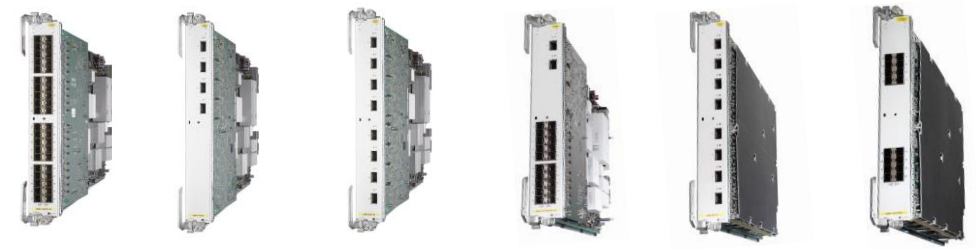




ASR 9000 Ethernet Line Card Overview

First-generation LC (Trident*)

-L, -B, -E



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

Second-gen LC (Typhoon)

-TR, -SE

A9K-24x10GE A9K-2x100GE

A9K-36x10GE

A9K-MOD160

A9K-MOD80

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

* Trident 10G line cards EoS/EoL:
http://www.cisco.com/c/en/us/products/routers/asr-9000-series-aggregation-services-routers/eos-eol-notice-c51-731288.html

Trident vs. Typhoon – Features

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

- Some features are not available yet in SW, although it will be supported on Typhoon hardware
- This is not the complete feature list

Modular SPA Linecard

20Gbps, feature ritch, high scale, low speed Interfaces

Quality of Service

- 128k Queues
- 128k Policers
- H-QoS
- Color Policing

Scalability

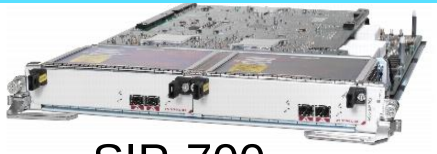
- Distributed Control and Data Plane
- 20Gbits, 4 SPA Bays
- L3 i/f, route, session protocol – scaled for MSE needs

High Availability

- IC-Stateful Switch Over Capability
- MR-APS
- IOS-XR base for high scale and Reliability

Powerful & Flexible QFP Processor

- Flexible uCode Architectue for Feature Richness
- L2 + L3 ServicesL FR, PPP, HDLC, MLPPP, LFI
- L3VPN, MPLS, Netflow, 6PE/6VPE



SIP-700

SPAs

SPA Support

- ChOC-3/12/48 (STM1/4/16)
- POS: OC3/STM1, OC12/STM4, OC-48/STM16, OC192/STM64
- ChT1/E1, ChT3/E3, CEoPs, ATM

ASR 9000 Optical Interface Support

NEW
XR5.1.1

Some new additions:
- 100Gbase-ER4 CFP
- Tunable SFP+
- CWDM 10G XFP+
- …

- All Linecards use Transceivers
- Based on Density and Interface Type the Transceiver is different
  – 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)]

SFP, SFP+

XFP

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

Integrated Services Module (ISM)

Application Domain

- Linux Based
- Multi-Purpose Compute Resource:
  - Used for Network Positioning System (NPS)
  - Used for Translation Setup and Logging of CGN Applications

IOS-XR Router Domain

- IOS-XR
- Control Plane
- Data Forwarding
- L3, L2 (management)
- IRB (4.1.1)
- Hardware Management

20M+ active translations
100s of thousands of subscribers

1M+ connections/second
validated for 14Gbps per ISM

Carrier Grade v6 (CGv6) Overview

IOS XR Releases		IPv4 & IPv6 Coexistence	IPv4 over IPv6 Network	IPv6 over IPv4 Network
4.2.0	NAT 444	Dual Stack		
4.2.1	DS-Lite		DS-Lite	
4.3.0		Stateful NAT64	Stateless 46 (dIVI/MAP-T)	
4.3.1			MAP-E	6RD

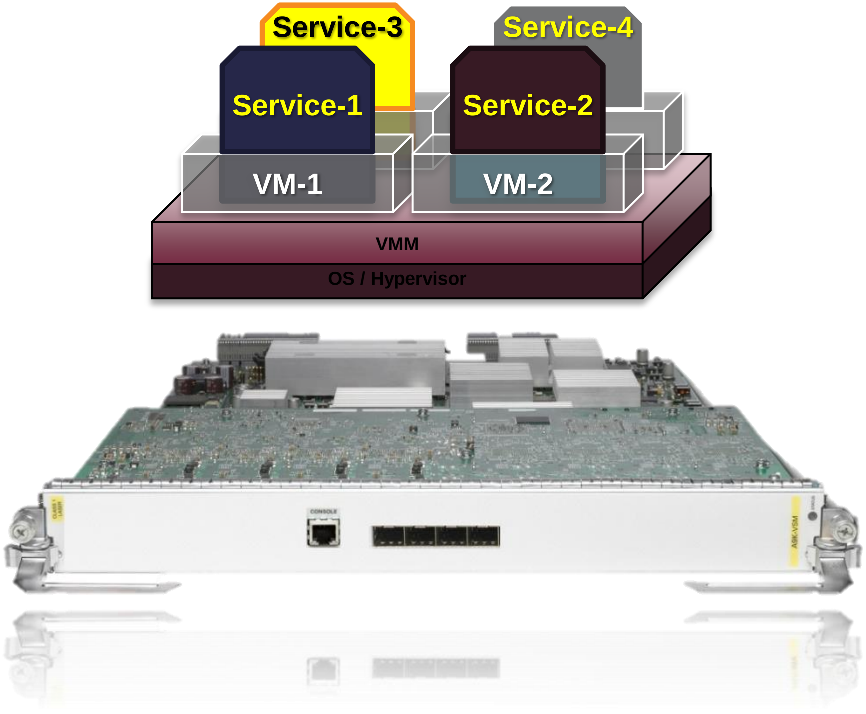
Stateful Transition Technologies - NAT444, DS-Lite & NAT64
Stateless Transition Technologies –
- MAT-T, MAP-E, 6RD
- Stateless implementation Inline on Typhoon LCs
- No requirement for Logging

ISM
2^{nd} Gen Eth Linecards

Cisco live!

Virtual Services Module (VSM)

Supported since IOS XR 5.1.1

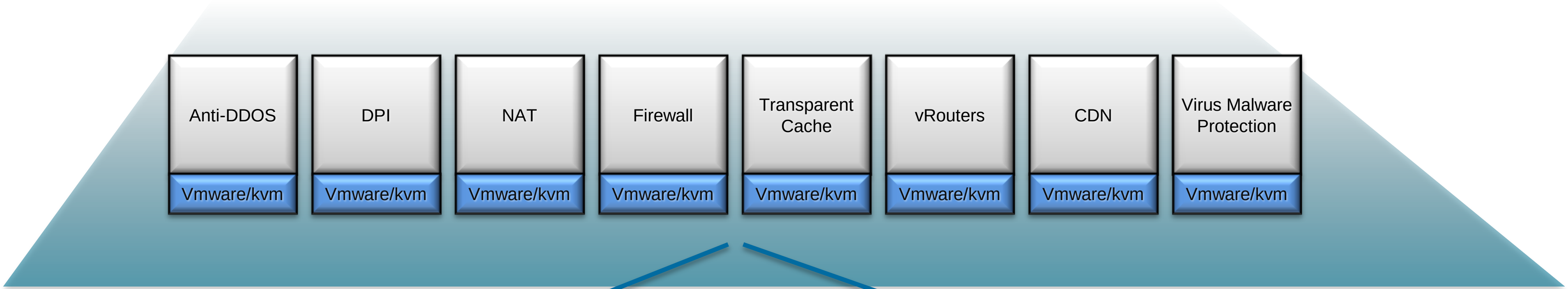


ASR 9000 VSM

- Data Center Compute:
  - 4 x Intel 10-core x86 CPU
- 2 Typhoon NPU for hardware network processing
  - 120 Gbps of Raw processing throughput
- HW Acceleration
  - 40 Gbps of hardware assisted Crypto throughput
  - Hardware assist for Reg-Ex matching
- Virtualization Hypervisor (KVM)
- Service VM life cycle management integrated into IOS-XR
- Services Chaining
- SDN SDK for 3rd Party Apps (OnePK)

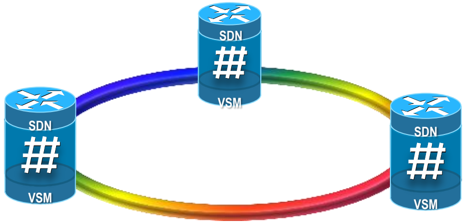
Cisco ASR 9000 Service Architecture Vision*

Flexible NfV placement for optimal Service Delivery



Decide per NfV function
Where to place it based
on service logic requirements

- **Low Latency**
- **Simplified Service Chaining**
- **Router integrated Management Plane**
- **Hardware assists**

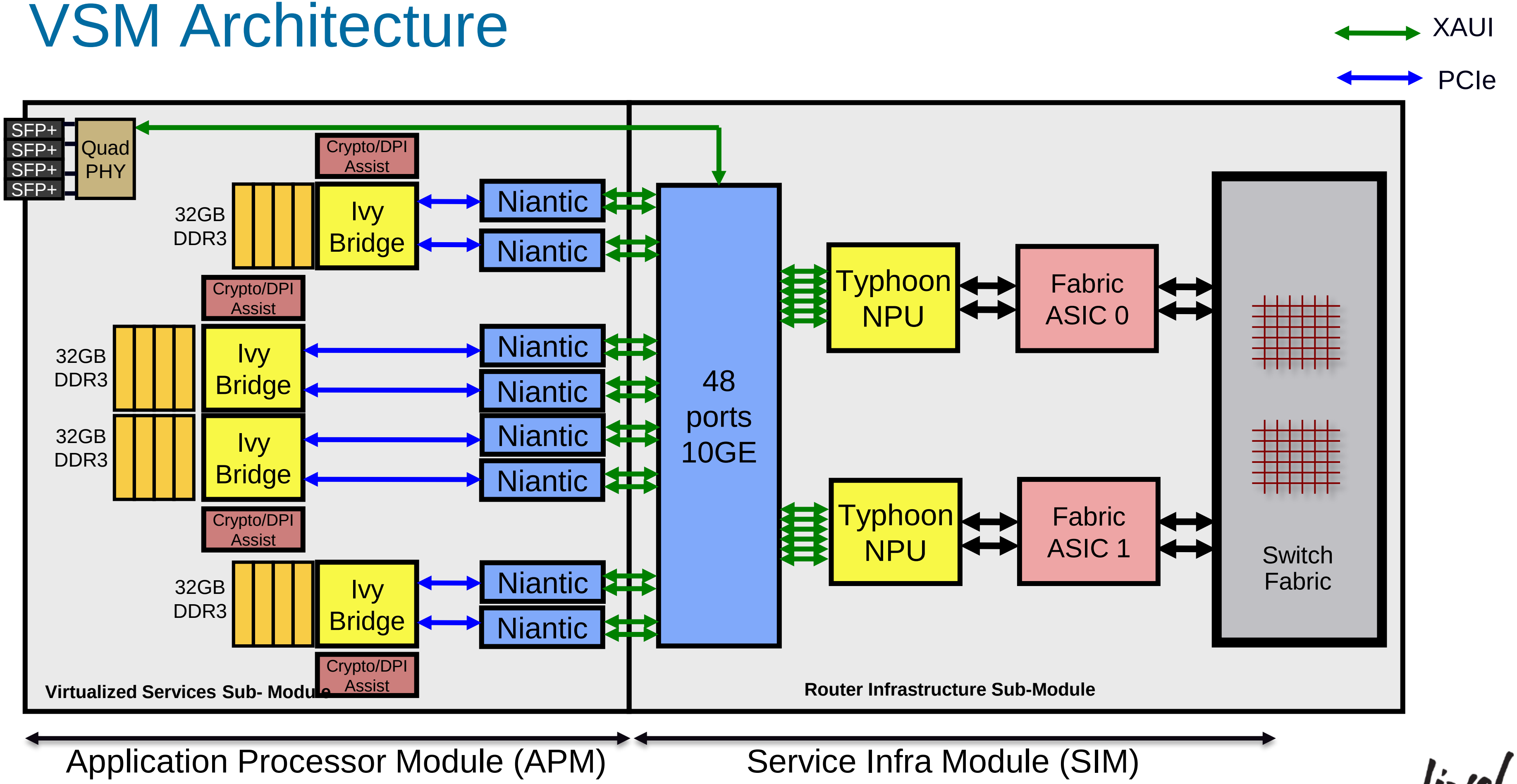


- **Elastic Scale & Throughput**
- **Cloud based operational model**

* Not all applications are supported in existing release

Cisco live!

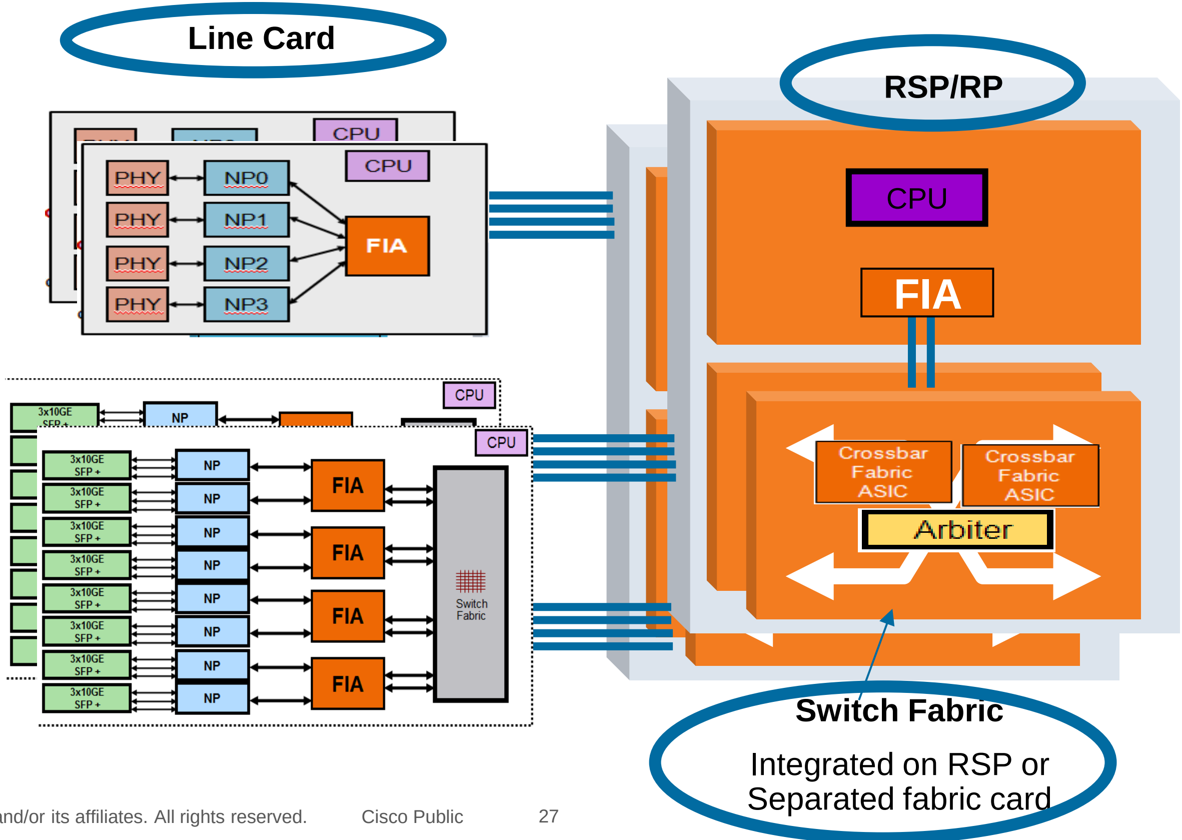
VSM Architecture



ASR9000 Hardware System Architecture (2)

HW Architecture

Cisco ASR 9000 Hardware System Components



Route Switch Processors (RSPs) and Route Processors (RPs)

RSP used in ASR9904/9006/9010, RP used in ASR9922/9912

	9006/9010 RSP	9904/9006/9010 RSP440	9912/9922-RP
	First generation RP and fabric ASIC	Secondary generation RP and fabric ASIC	
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 (fabric integrated on RSP)	220G + 220G (9006/9010) 385G + 385G (9904) (fabric integrated on RSP)	660G+110G (separated fabric card)

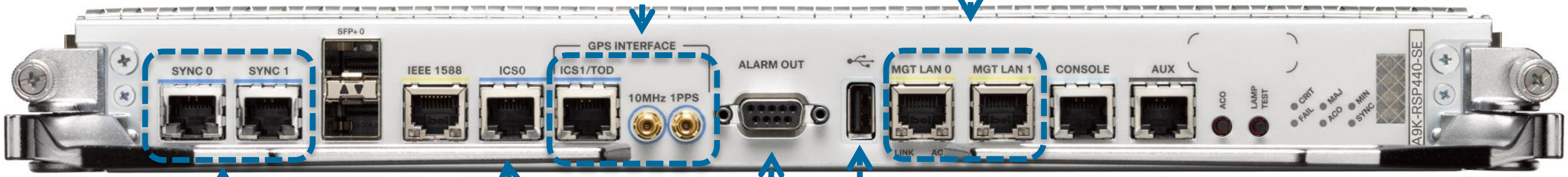
Identical

RSP440 – Faceplate and Interfaces

2x 1G
nV Edge EOBC

GPS Interface
ToD, 1pps, 10Mhz

2x Mgmt Eth



BITS/DTI/J.211

PTP*

nV Edge Sync*

Alarms

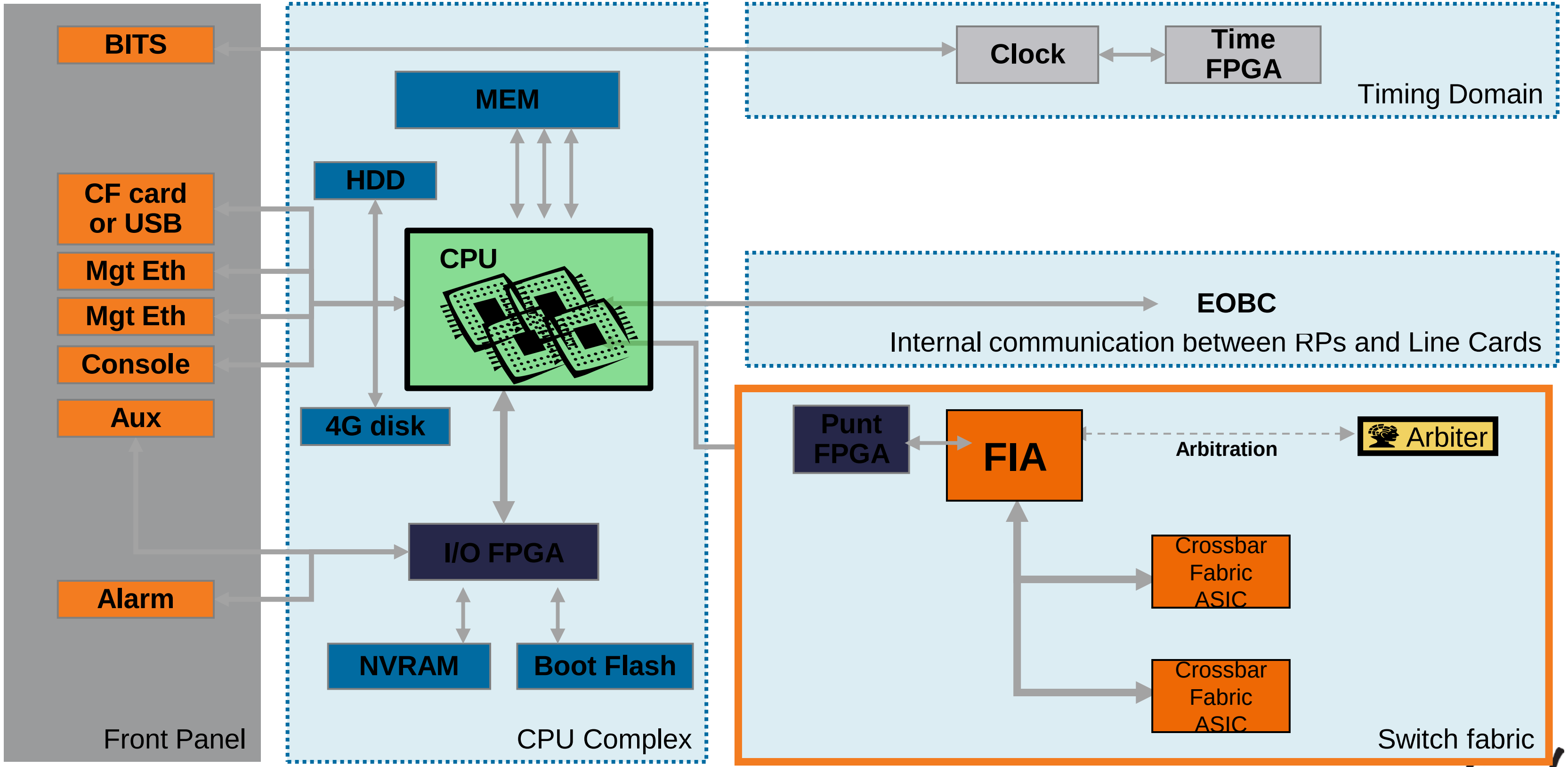
USB

Console & Aux

Status LEDs

* Future SW support

RSP Engine Architecture



ASR 9000 Switch Fabric Overview

Integrated fabric/RP/LC

9001, 2RU, 120G

9001-S, 2RU, 60G

Fabric is integrated on RSP

1+1 redundancy

9904

RSP440: 385G+385G /slot

9006 9010

RSP440: 220G+220G /slot

RSP: 92G+92G* /slot

Separated fabric card

6+1 redundancy

9912 9922

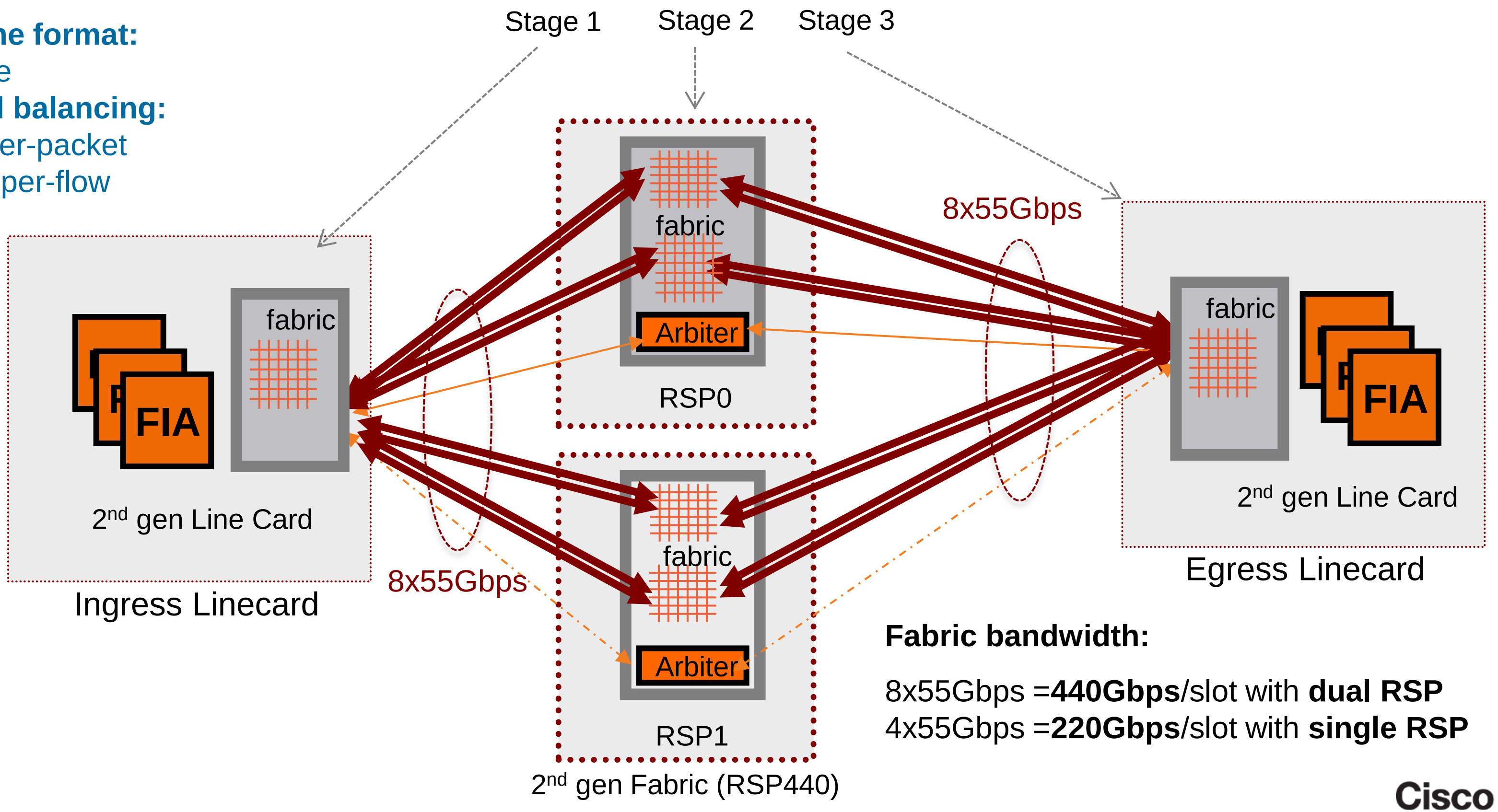
660G+110G /slot

* First generation switch fabric is only supported on 9006 and 9010 chassis.
It’s fully compatible with all existing line cards

ASR 9006/9010 Switch Fabric Overview

3-Stage Fabric

Fabric frame format:
Super-frame
Fabric load balancing:
Unicast is per-packet
Multicast is per-flow

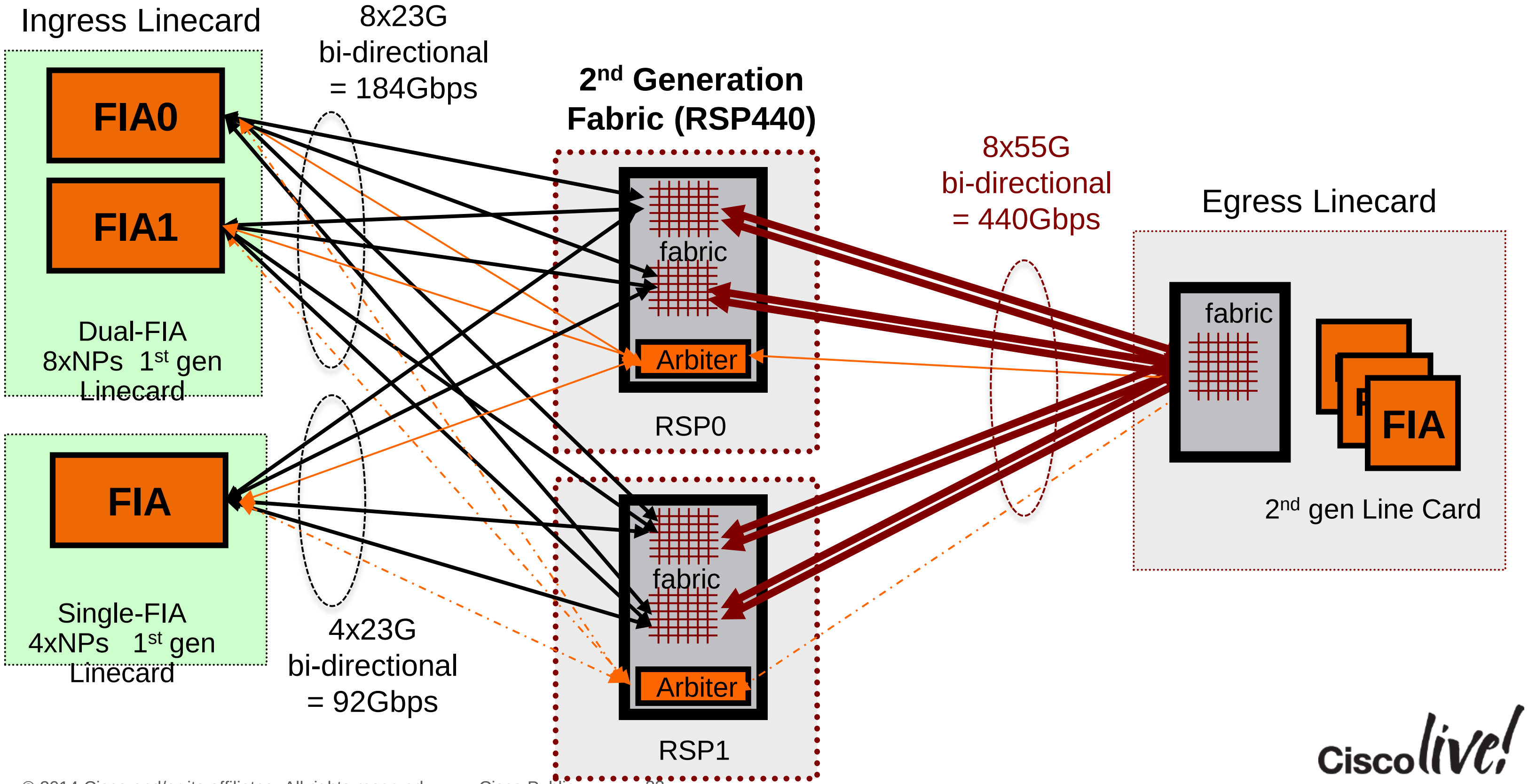


Fabric bandwidth:

8x55Gbps =**440Gbps**/slot with **dual RSP**
4x55Gbps =**220Gbps**/slot with **single RSP**

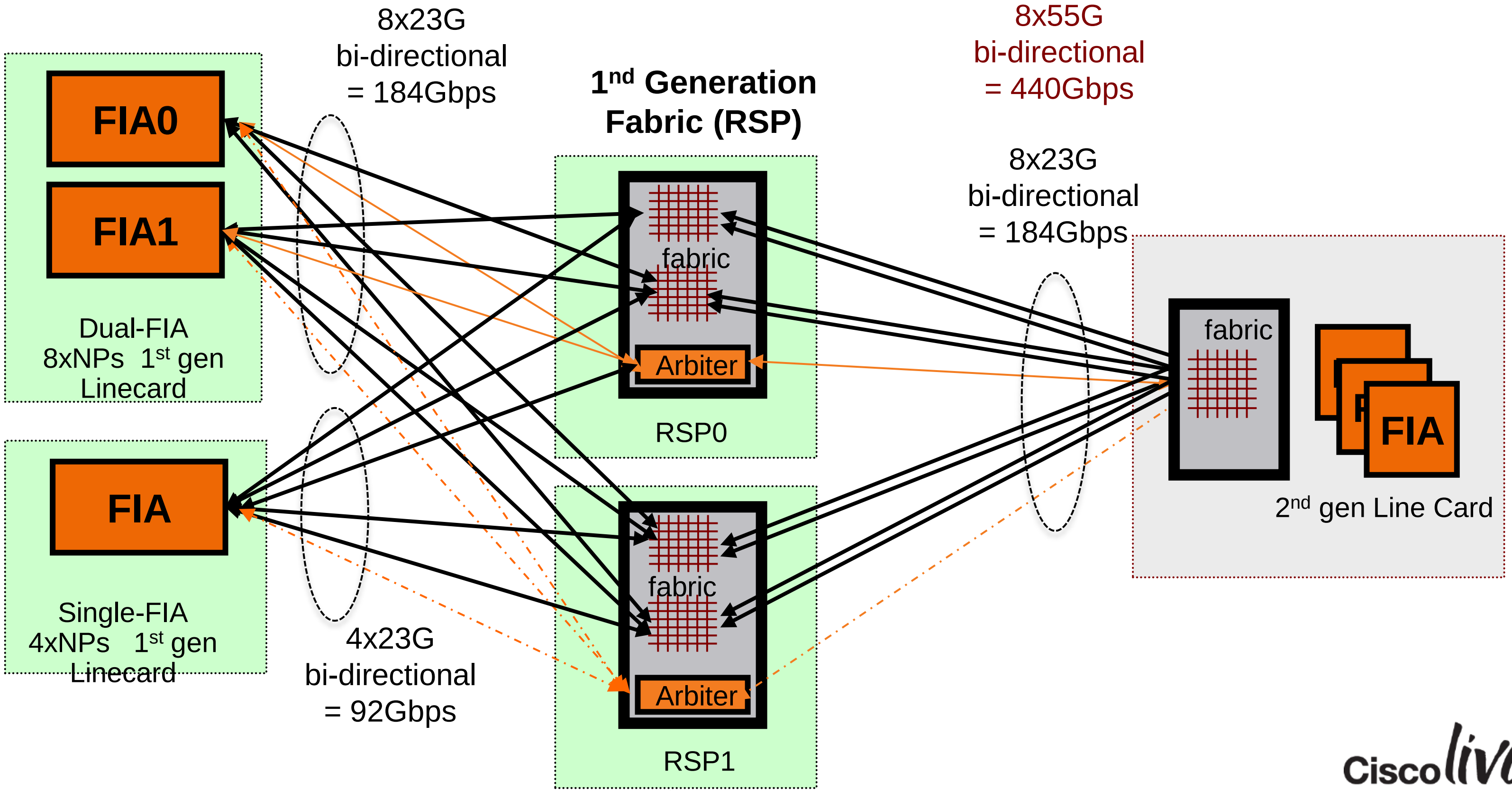
1st/2nd Generation switch fabric compatibility

System With 2nd Generation Fabric



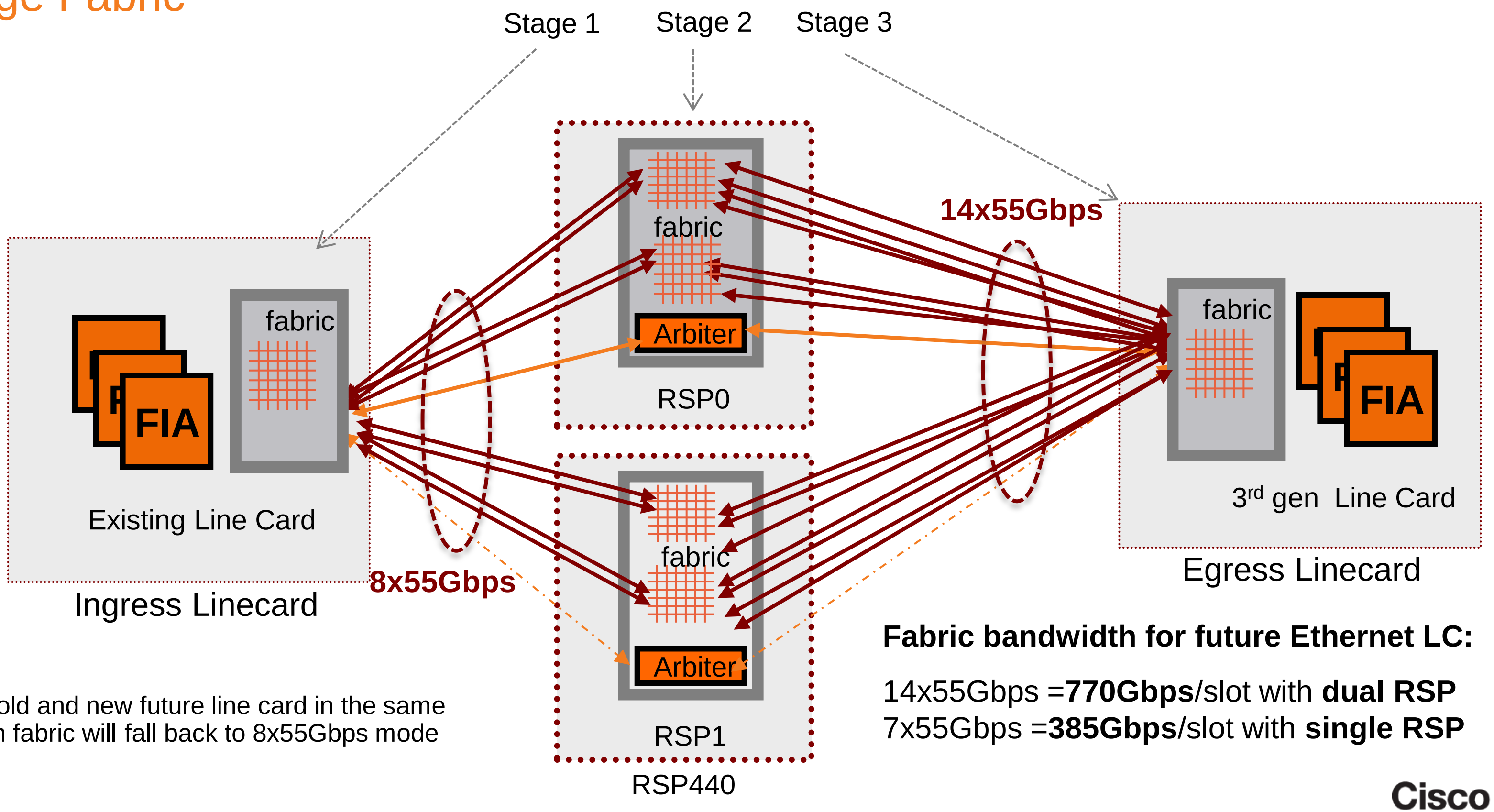
1st/2nd Generation switch fabric compatibility

System with 1st Generation Fabric



ASR 9904 Switch Fabric Overview

3-Stage Fabric



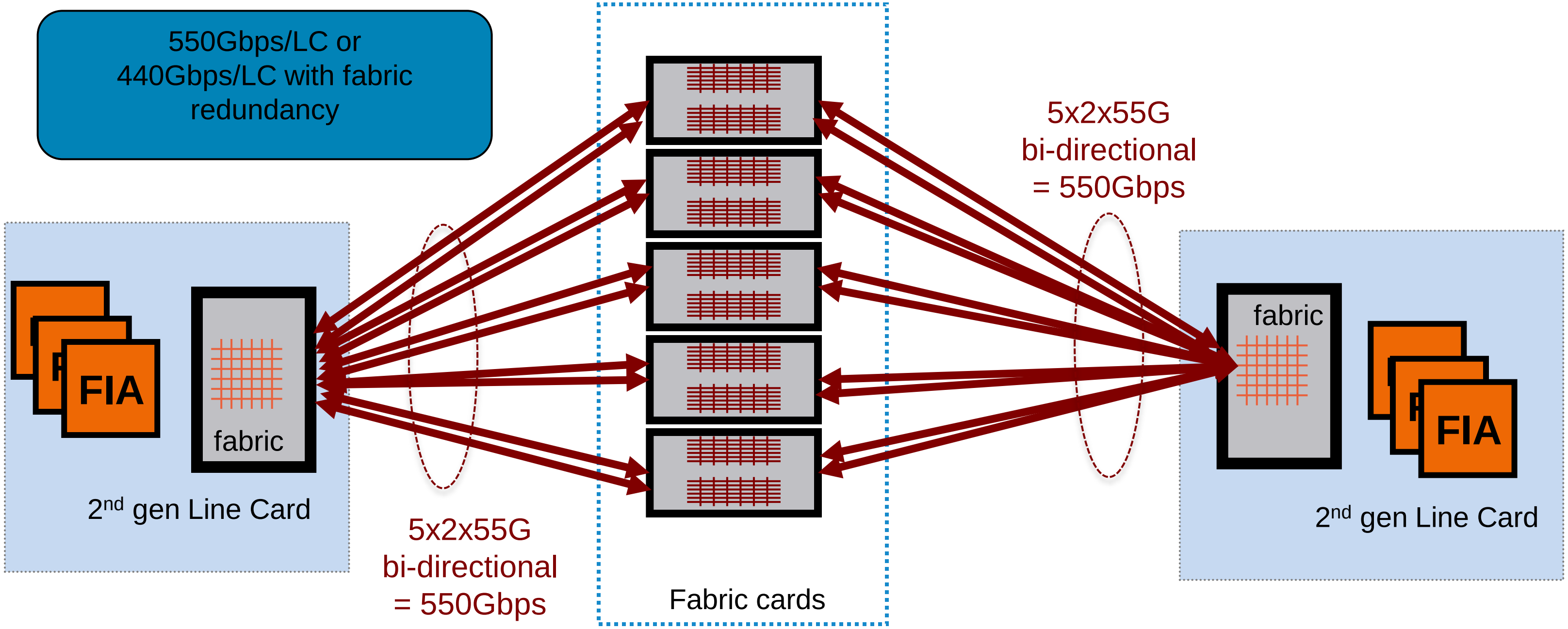
Fabric bandwidth for future Ethernet LC:

14x55Gbps =**770Gbps**/slot with **dual RSP**
7x55Gbps =**385Gbps**/slot with **single RSP**

Note, if mix old and new future line card in the same system, then fabric will fall back to 8x55Gbps mode

ASR 9912/9922 Fabric Architecture: 5-plane System

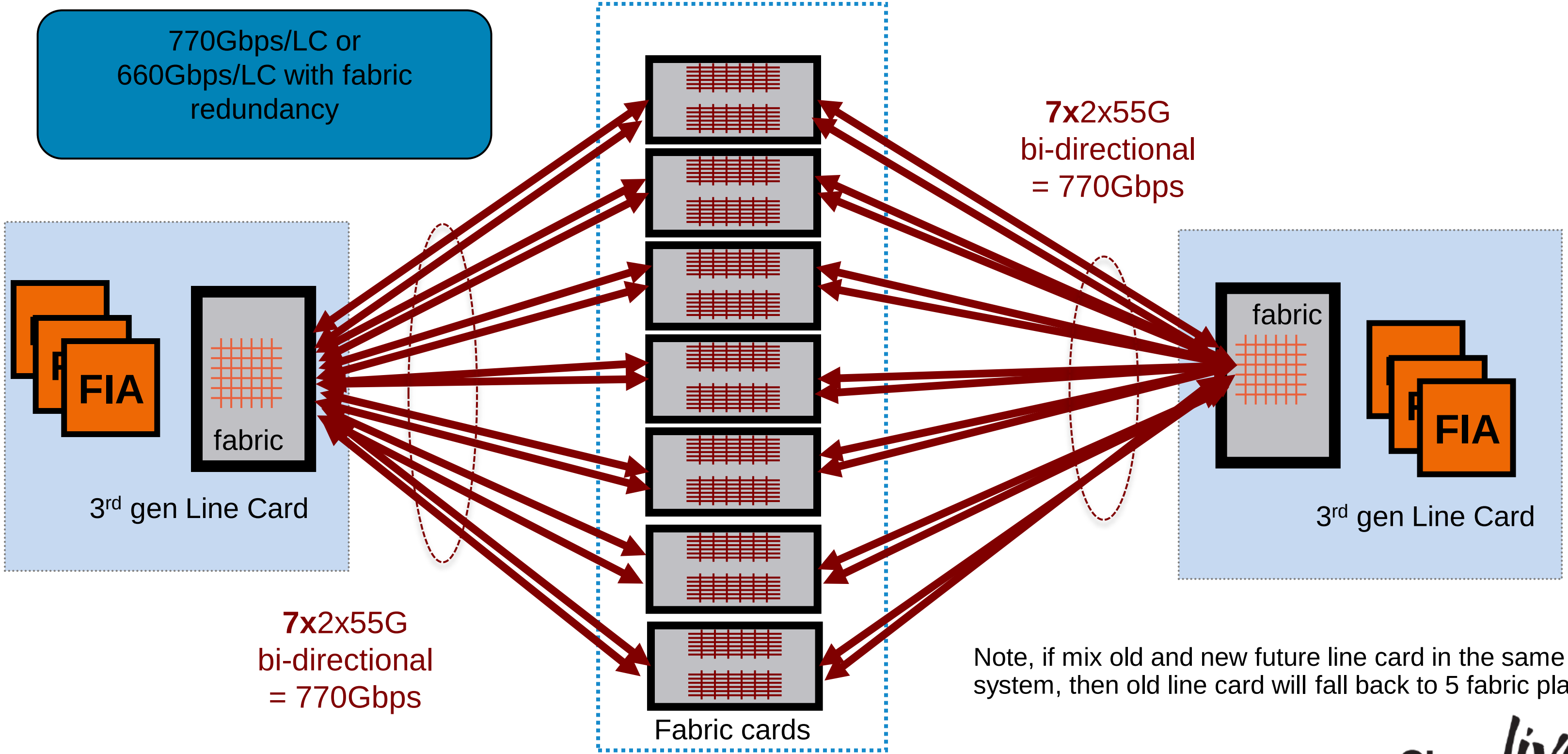
Supported Today



ASR 9912/9922 Fabric Architecture: 7-plane System

Supported in future

770Gbps/LC or
660Gbps/LC with fabric
redundancy

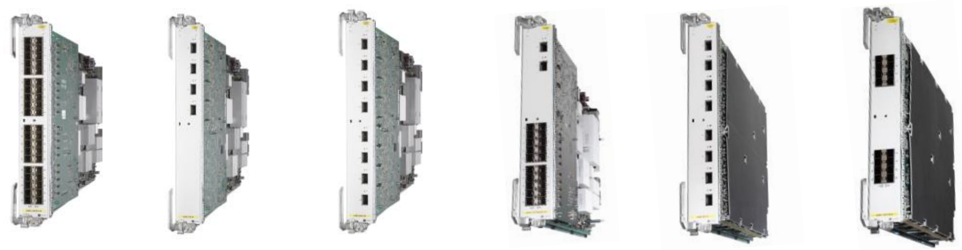


Note, if mix old and new future line card in the same system, then old line card will fall back to 5 fabric plane

ASR 9000 Ethernet Line Card Overview

-L, -B, -E

First-generation LC
Trident NPU:
15Gbps, ~15Mpps,
bi-directional



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

-TR, -SE

Second-gen LC
Typhoon NPU:
60Gbps, ~45Mpps,
bi-directional

A9K-24x10GE

A9K-2x100GE
(A9K-1x100G)

A9K-36x10GE

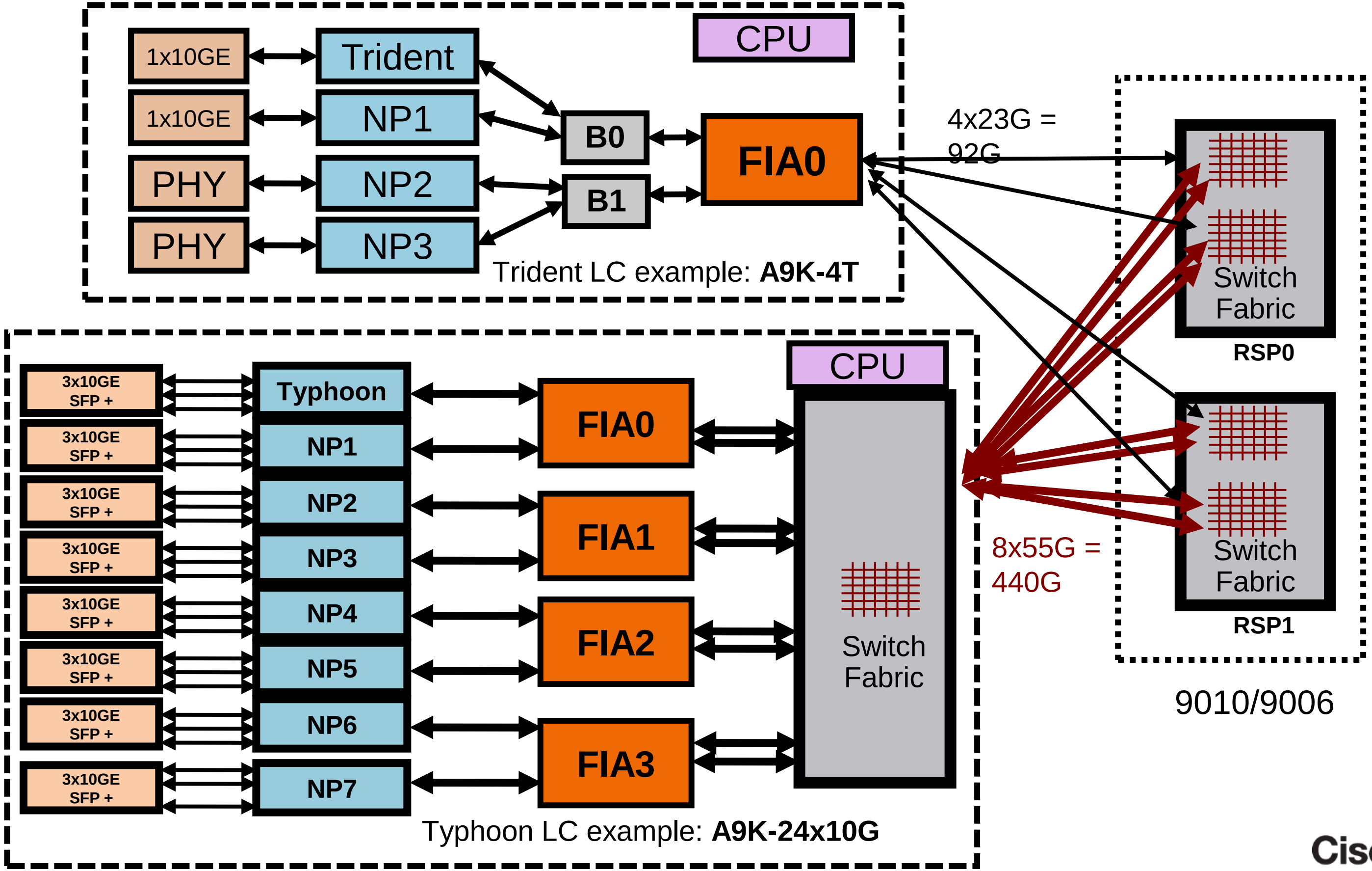
A9K-MOD160

A9K-MOD80

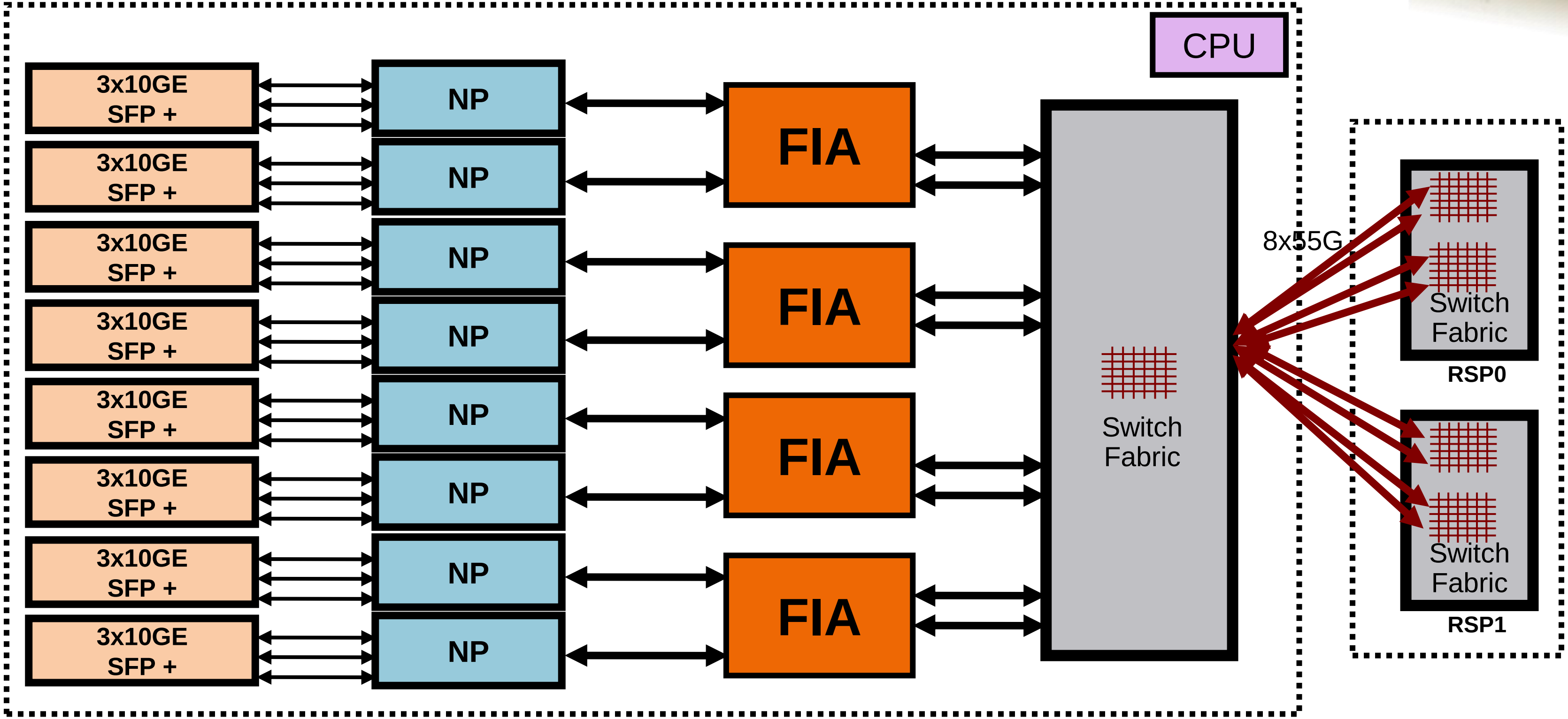
MPAs
20x1GE
2x10GE
4x10GE
8x10GE
1x40GE
2x40GE

-L: low queue, -B: Medium queue, -E: Large queue, -TR: transport optimized, -SE: Service edge optimized

ASR 9000 Line Card Architecture Overview



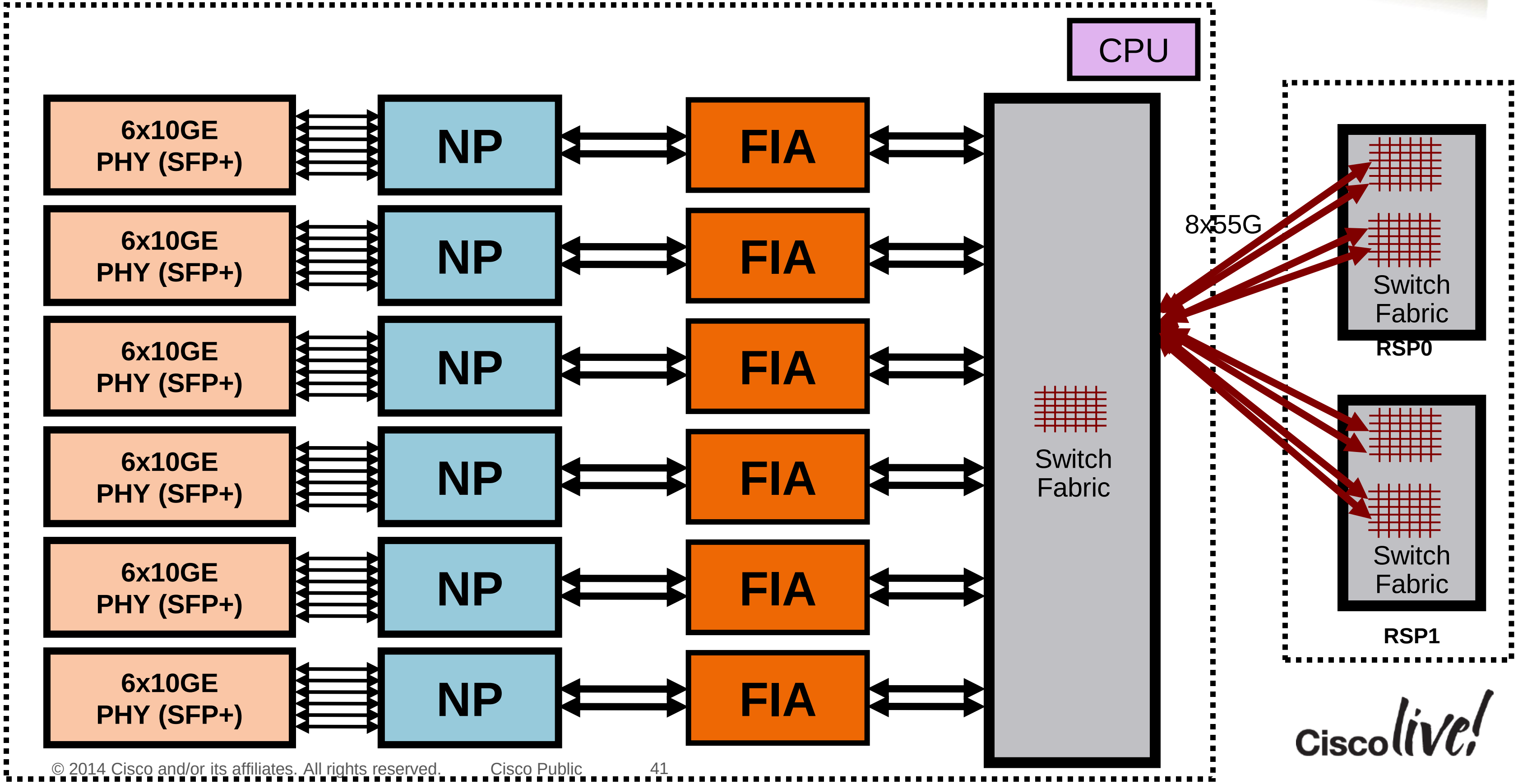
24port 10GE Linecard Architecture



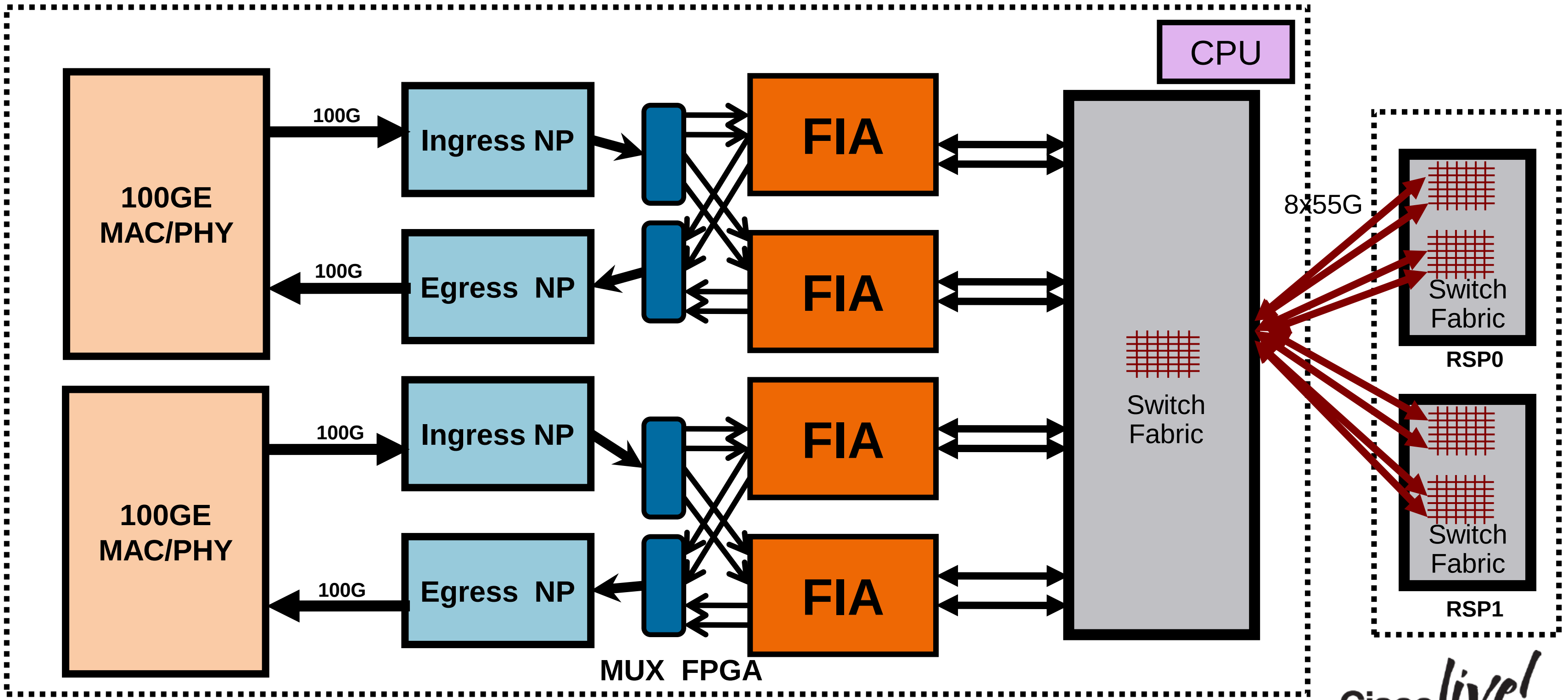
Each NP: 60Gbps bi-directional
120Gbps uni-directional

Each FIA: 60Gbps bi-directional

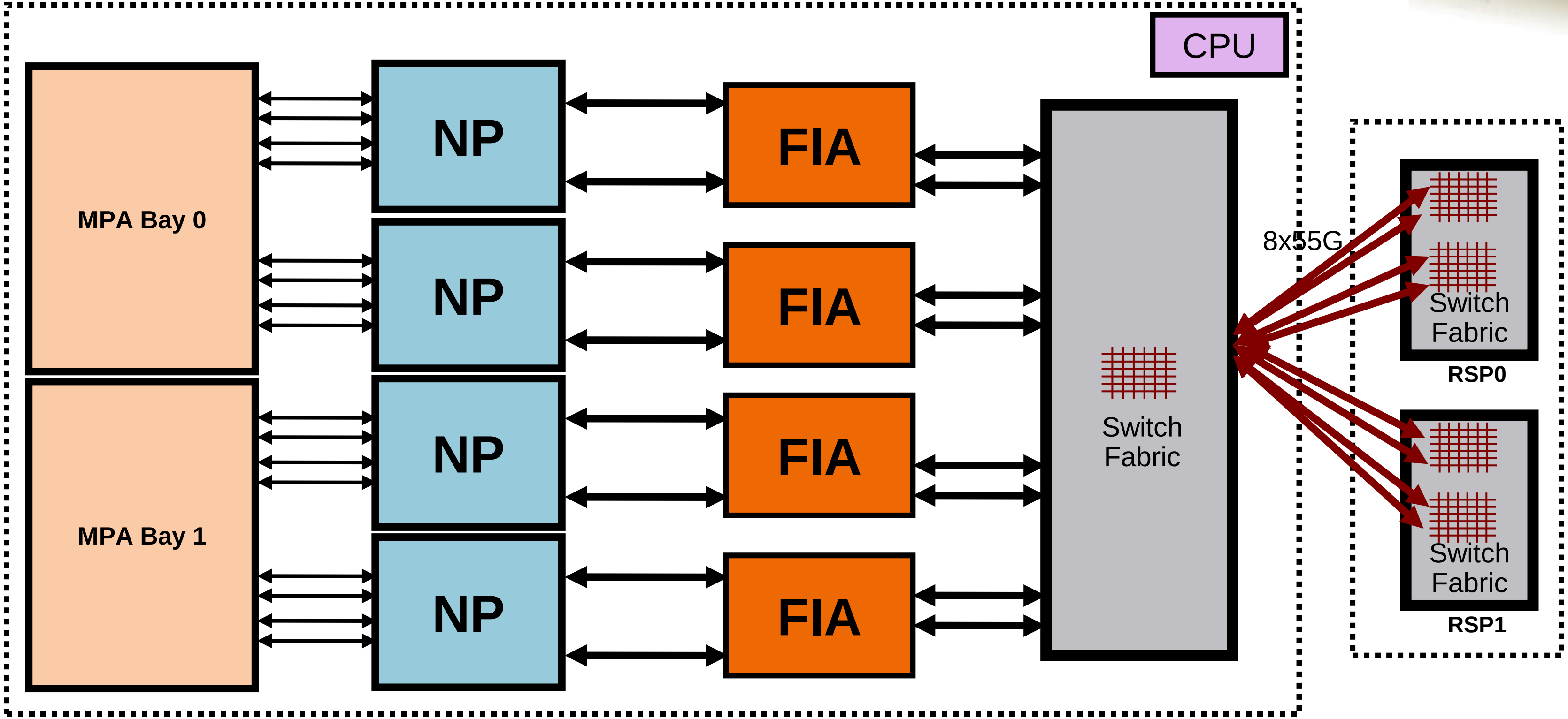
36port 10GE Linecard Architecture



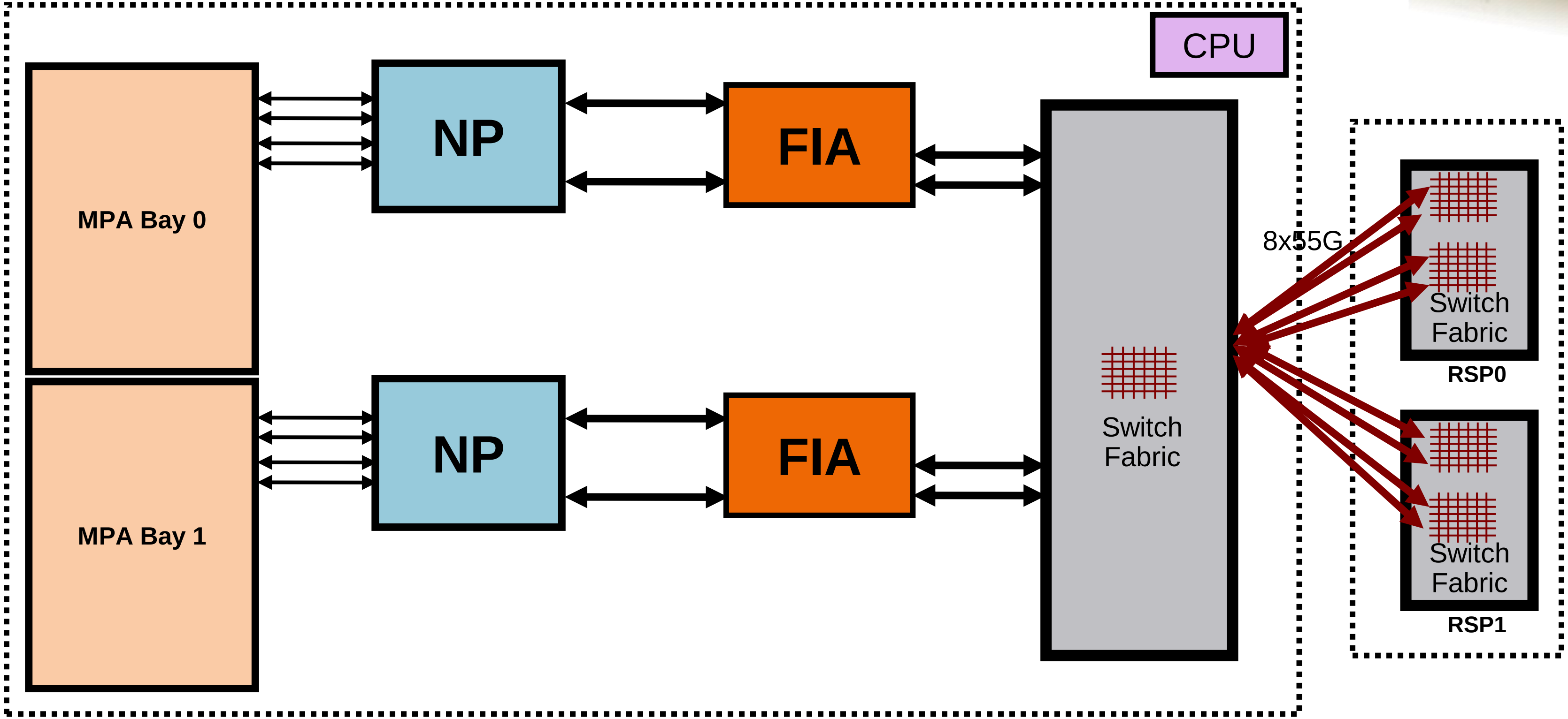
2port 100GE Linecard Architecture



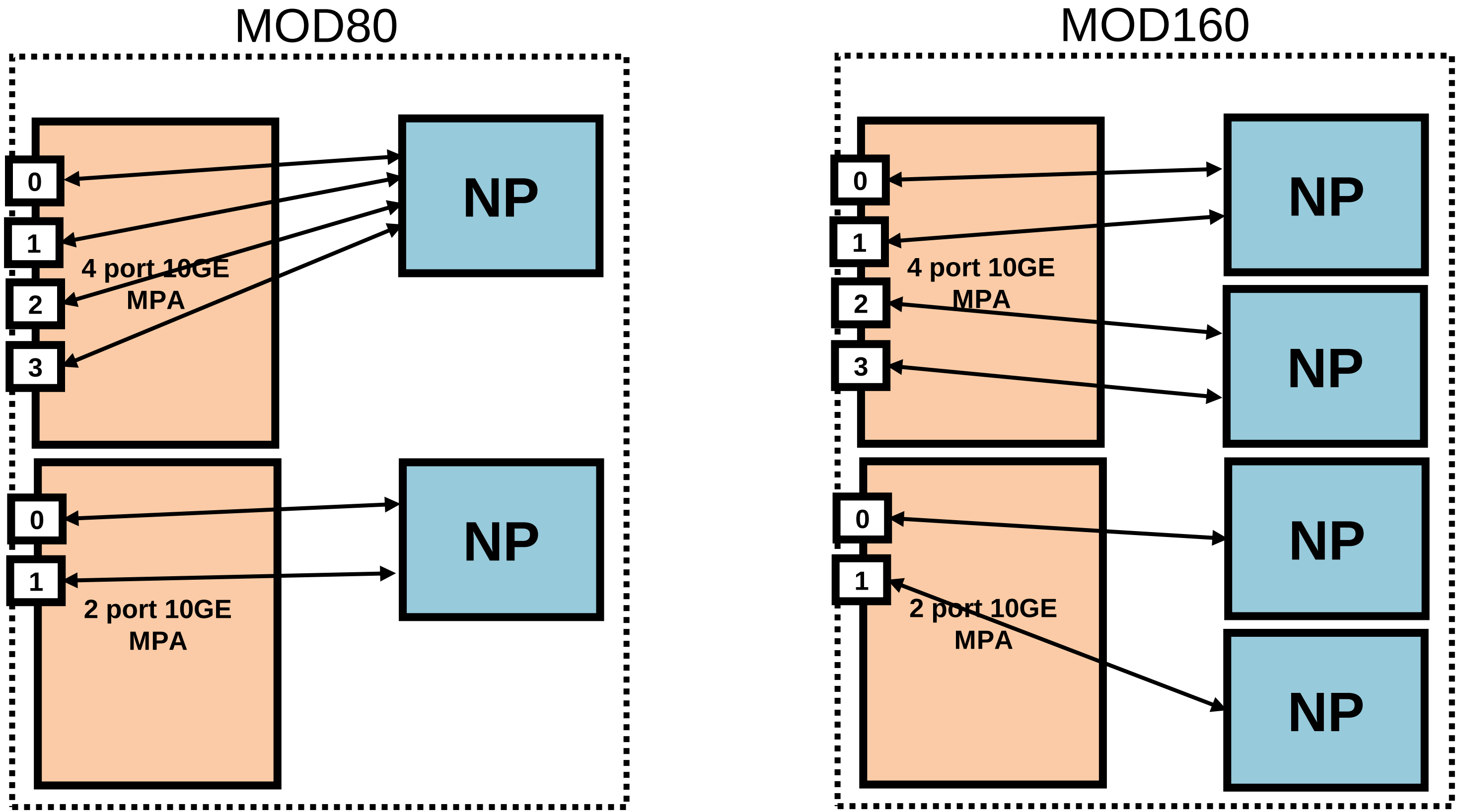
Module Cards – MOD160



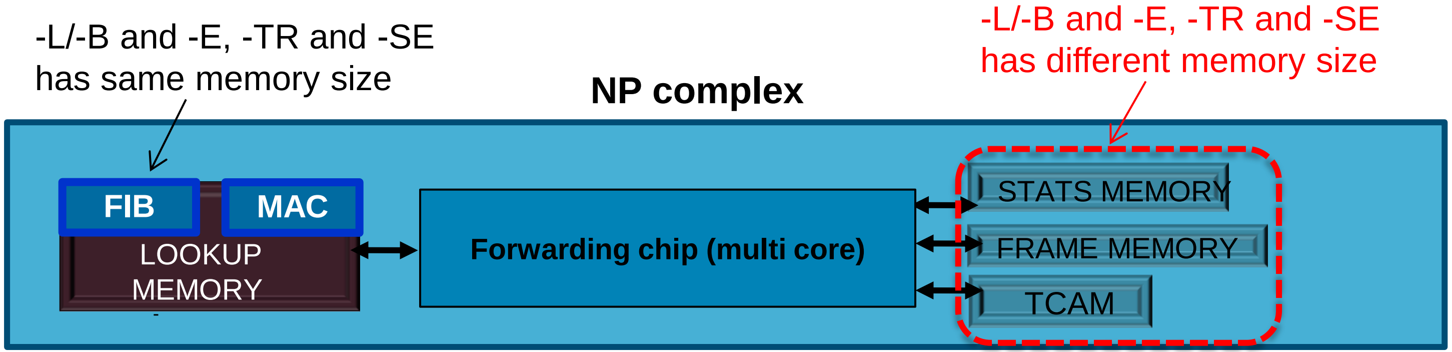
Module Cards – MOD80



MPA Port Mapping Examples for 10GE Ports



Network Processor Architecture Details

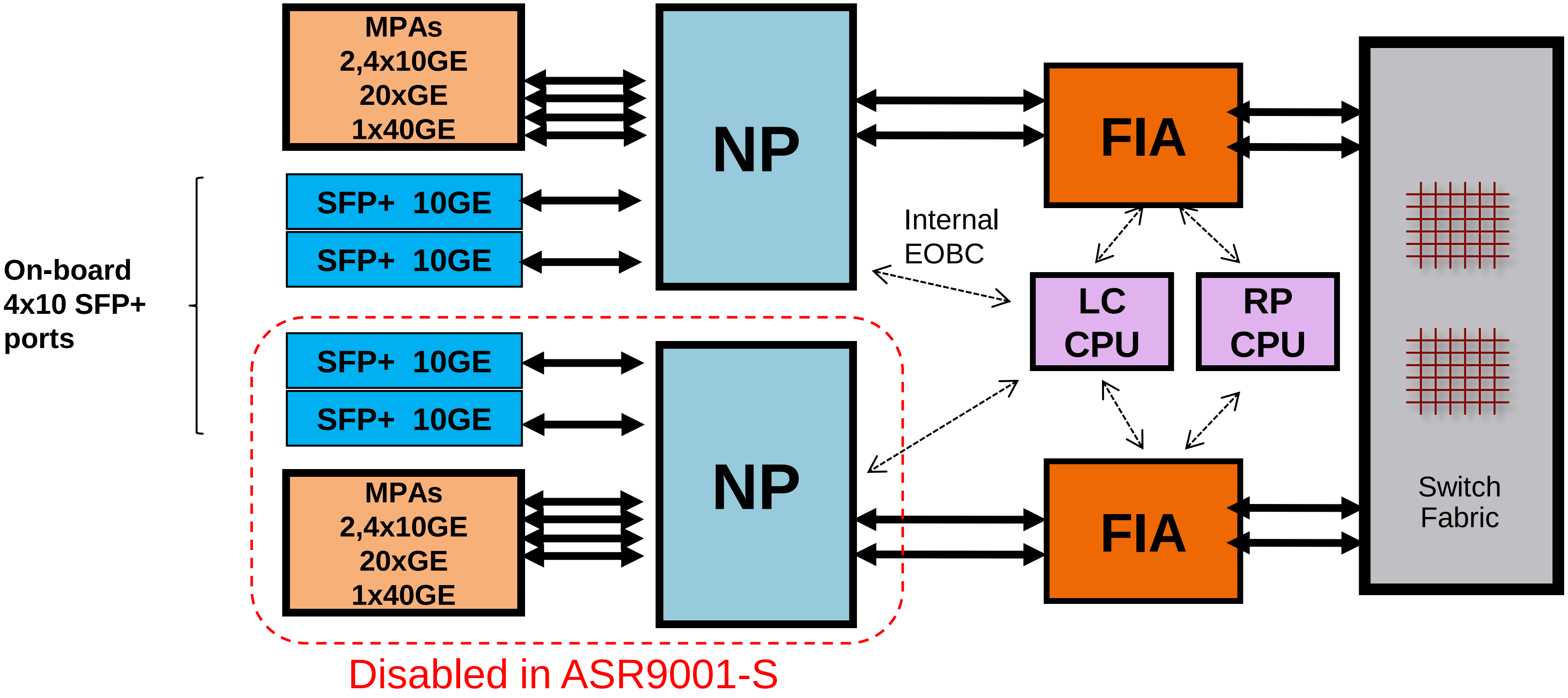


- TCAM: VLAN tag, QoS and ACL classification
- Stats memory: interface statistics, forwarding statistics etc
- Frame memory: buffer, Queues
- Lookup Memory: forwarding tables, FIB, MAC, ADJ
- -TR/-SE, -L/-B/-E
  - Different TCAM/frame/stats memory size for different per-LC QoS, ACL, logical interface scale
  - Same lookup memory for same system wide scale mixing different variation of LCs doesn't impact system wide scale

-L: low queue, -B: Medium queue, -E: Large queue, -TR: transport optimized, -SE: Service edge optimized

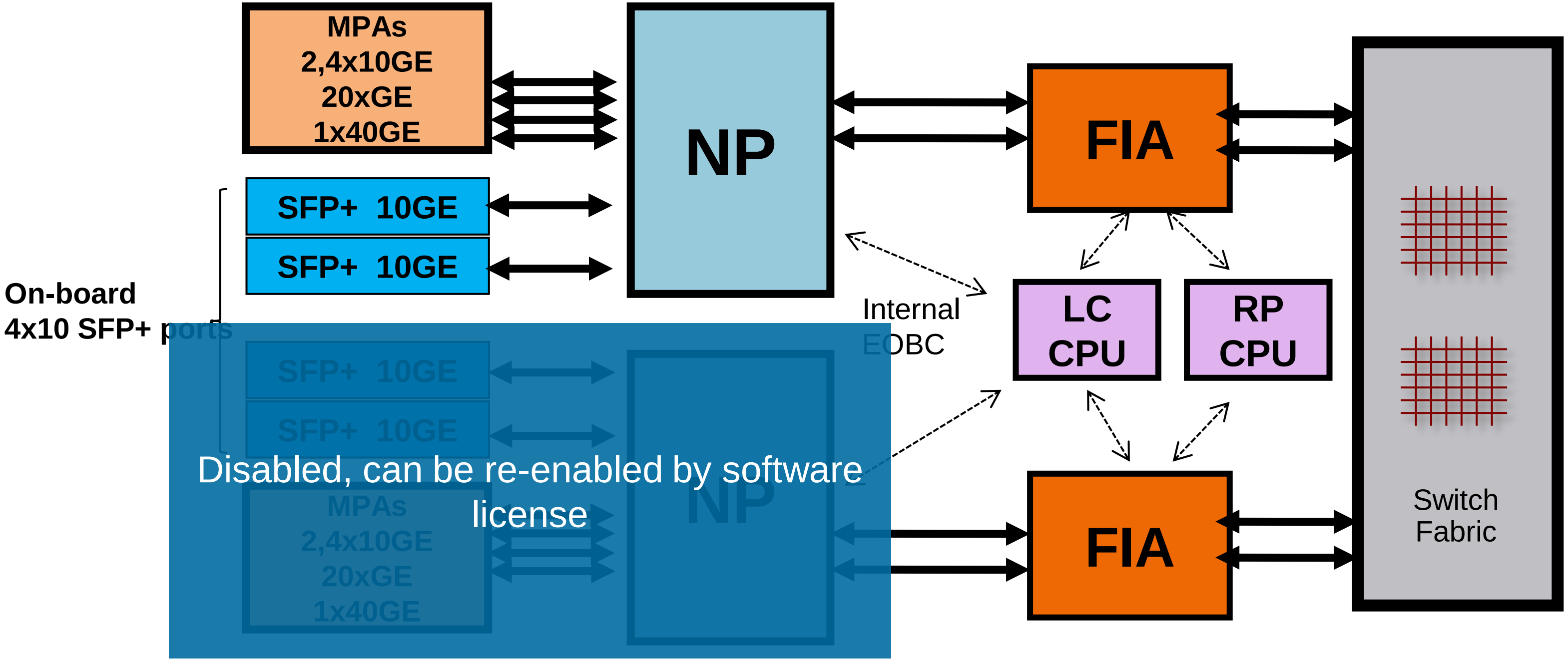
ASR9001 Architecture

Identical HW Components as the Modular Systems



ASR 9001/9001-S Architecture

Identical HW Components as the Modular Systems



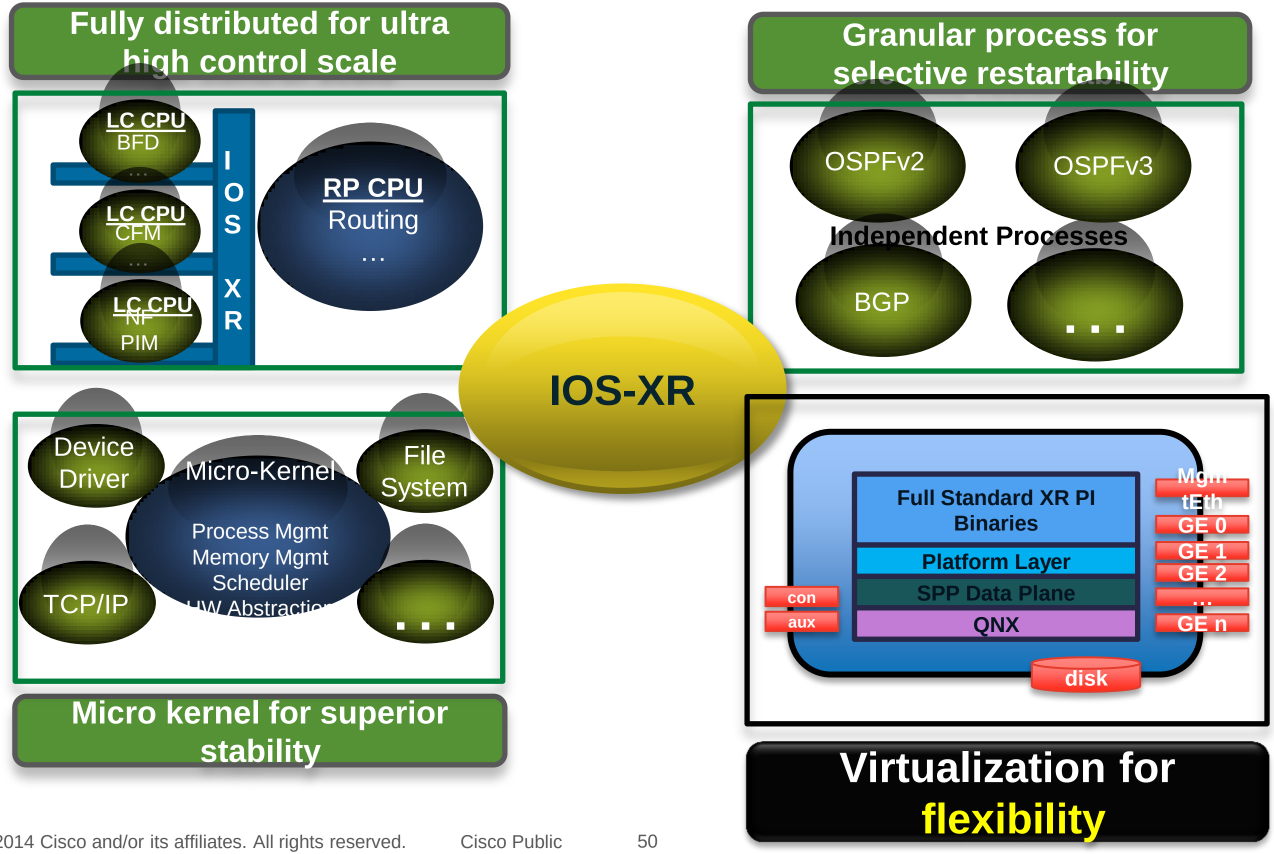
ASR 9001/9001-S architecture is based on Typhoon line card, second generation fabric ASIC and RSP

ASR 9000 Software System Architecture (1)

IOS-XR

Industry Hardened IOS XR

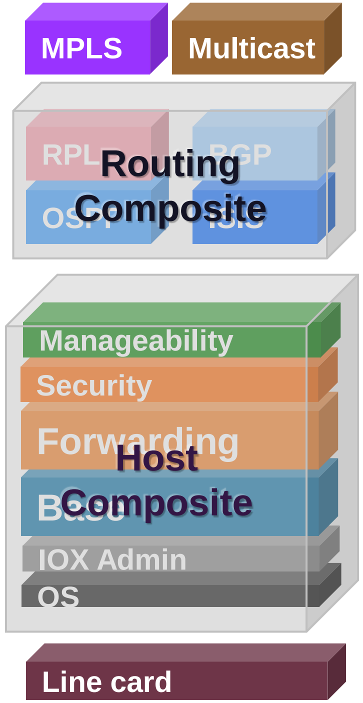
Micro Kernel, Modular, Fully Distributed, Moving towards Virtualization



Cisco live!

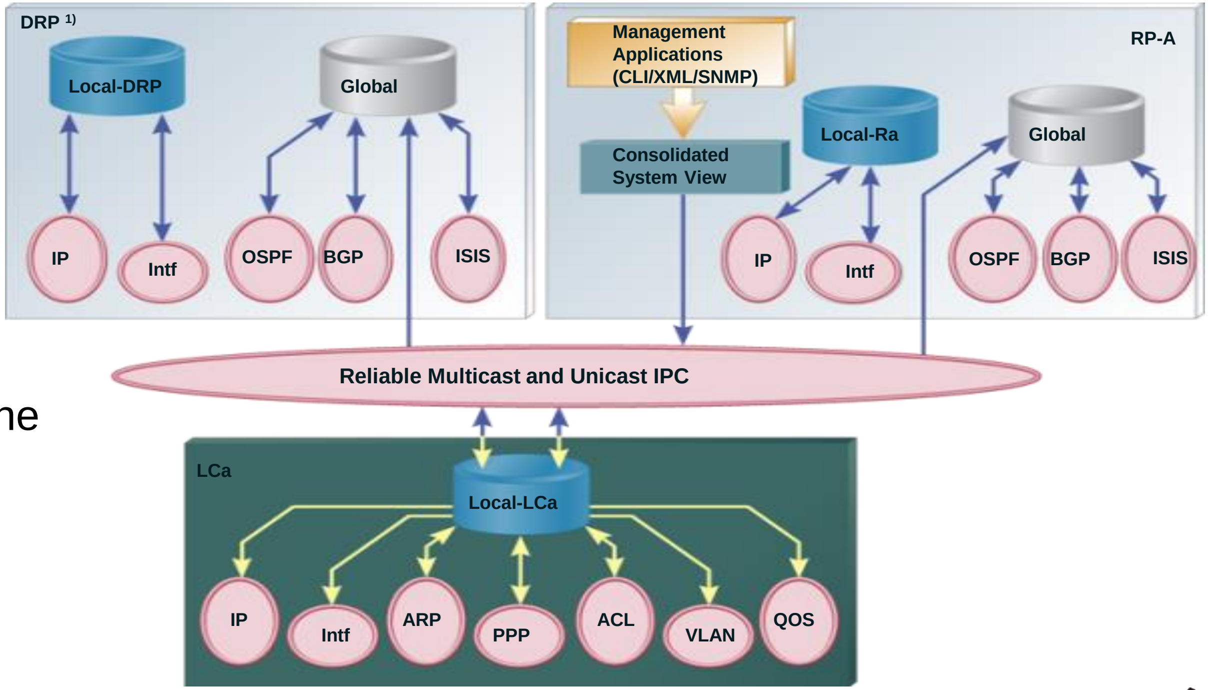
Cisco IOS-XR Software Modularity

- Ability to upgrade independently MPLS, Multicast, Routing protocols and Line Cards
- Ability to release software packages independently
- 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
- CLI, SNMP and XML/Netconf Access for EMS/NMS

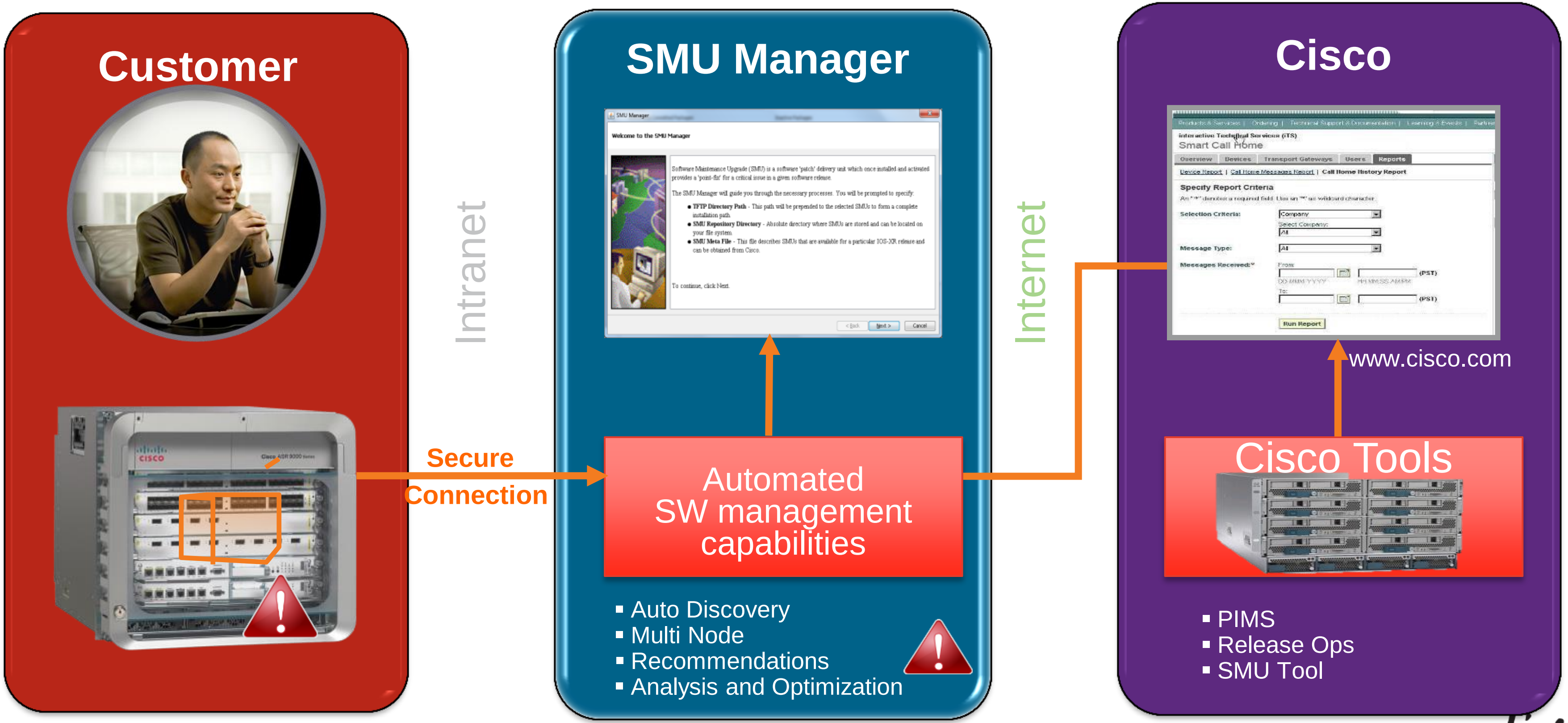


1) DRPs are only supported in CRS

Software Maintenance Updates (SMUs)

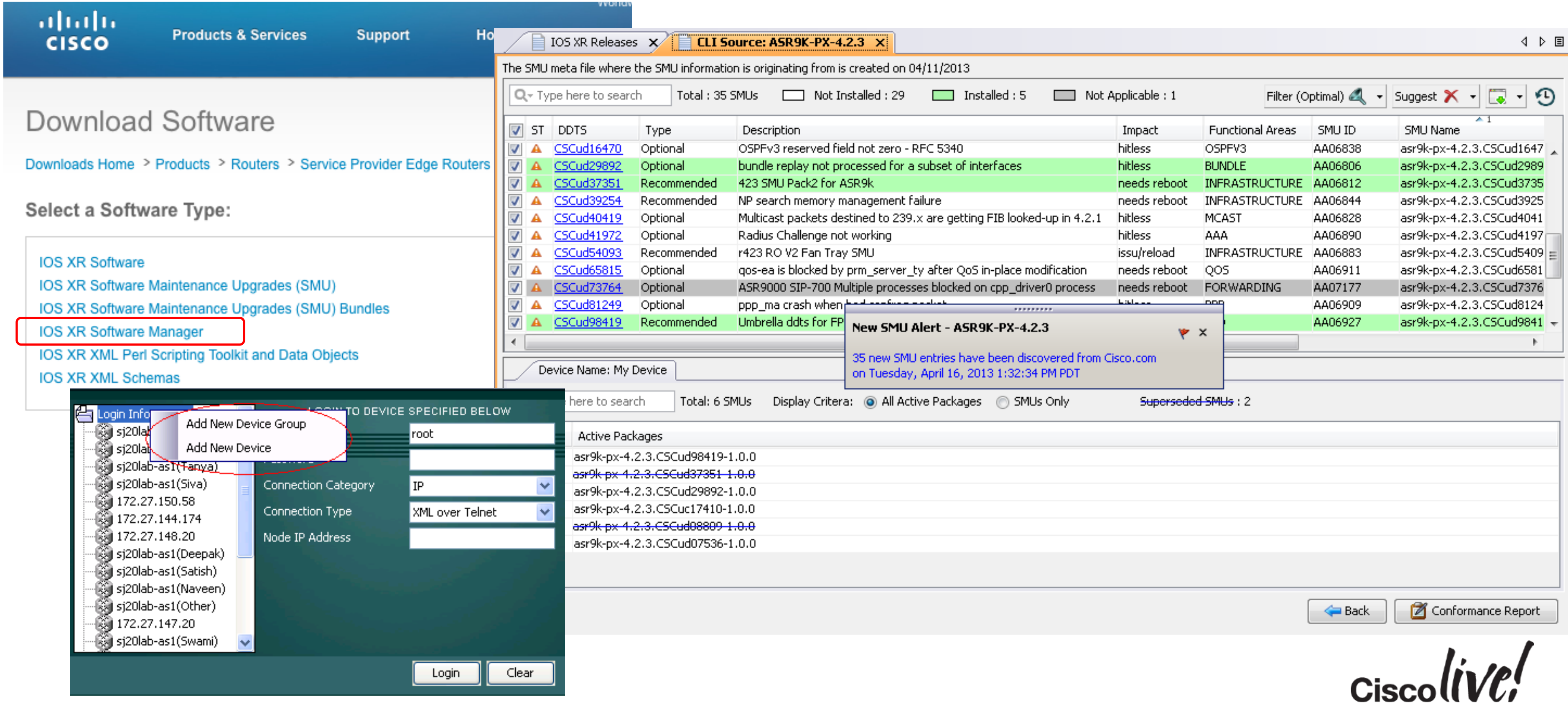
- Allows for software package installation/removal leveraging on Modularity and Process restart
- Redundant processors are not mandatory (unlike ISSU) and in many cases is non service impacting and may not require reload.
- Mechanism for
  - delivery of critical bug fixes without the need to wait for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

SMU Management Architecture



Introducing Cisco Software Manager

Available on CCO in the Downloads Section for ASR9000



Cisco Virtualization Technologies

Platform Virtualization

IOS XR

VM-based tool: IOS XRv
FCS Target: 5.1.1

NX-OS

VM-based tool: NX-OSv
Target: H2FY13

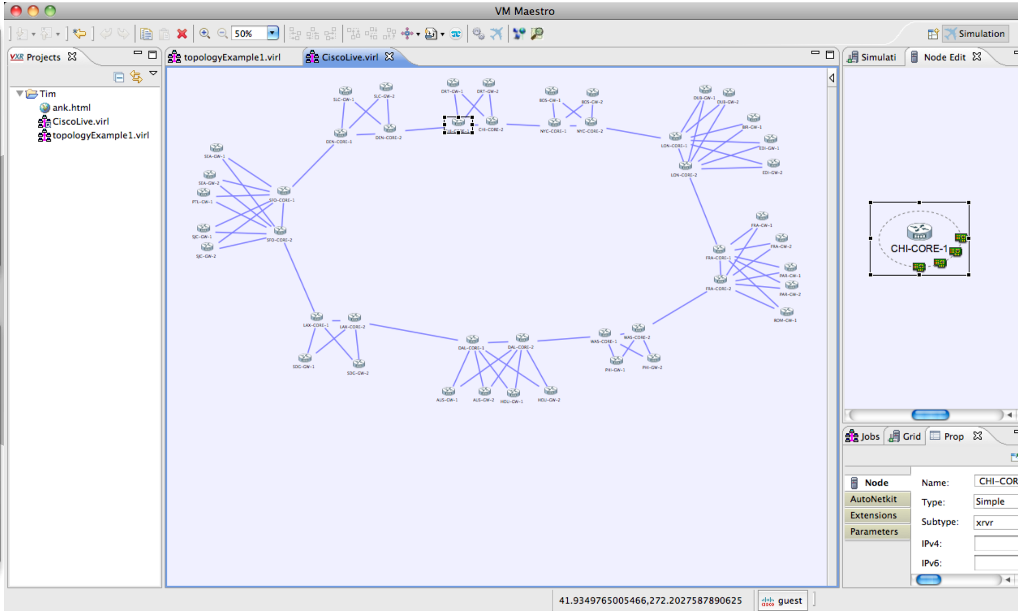
IOS XE

VM-based tool: CSR1000v
FCS: Q2CY13

IOS

VM-based tool: IOSv
H2FY13

Cisco Modeling Lab (CML)



IOS-XRv

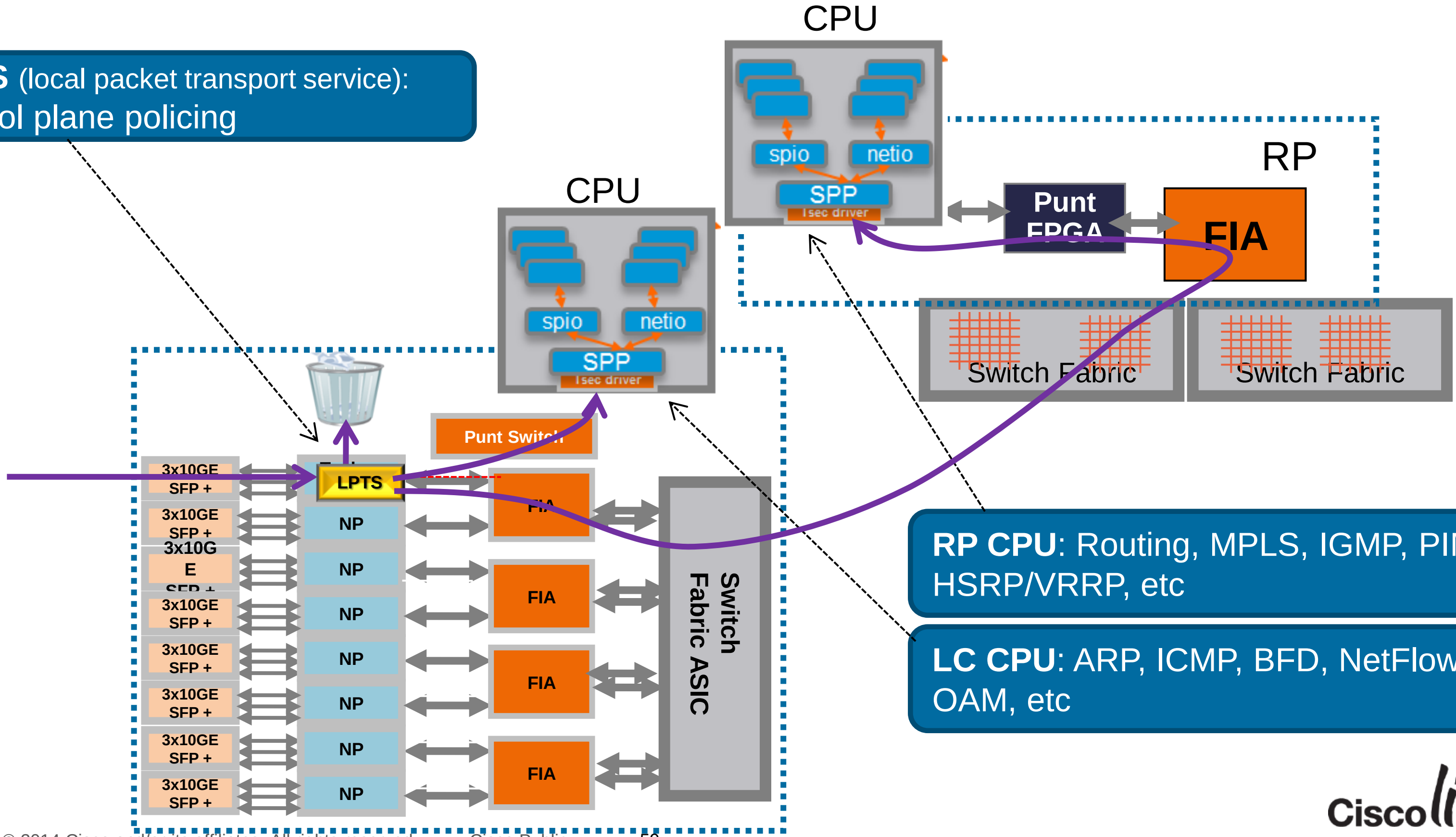
- **Cisco IOS XRv supported since 5.1.1**
  - Control plane only. Virtual data plane on the roadmap
  - Initial application: BGP router reflect, Cisco Modeling Lab (CML)
  - Release Notes: http://www.cisco.com/en/US/partner/docs/ios_xr_sw/iosxr_r5.1/general/release/notes/reln-xrv.html
  - Demo Image: https://upload.cisco.com/cgi-bin/swc/fileexg/main.cgi?CONTYPES=Cisco-IOS-XRv
  - Installation Guide: http://www.cisco.com/en/US/docs/ios_xr_sw/ios_xrv/install_config/b_xrvr_432.html
  - Quick Guide to ESXi: https://supportforums.cisco.com/docs/DOC-39939
- **Cisco Modeling Lab (CML)**
  - CML is a multi-purpose network virtualization platform that provides ease-of-use to customers wanting to build, configure and Test new or existing network topologies. IOS XRv Virtual XR platform is now available
  - http://www.cisco.com/en/US/docs/ios_xr_sw/ios_xrv/install_config/b_xrvr_432_chapter_01.html

ASR 9000 Software System Architecture (2)

Control Plane and Forwarding Plane

ASR9000 Fully Distributed Control Plane

LPTS (local packet transport service): control plane policing

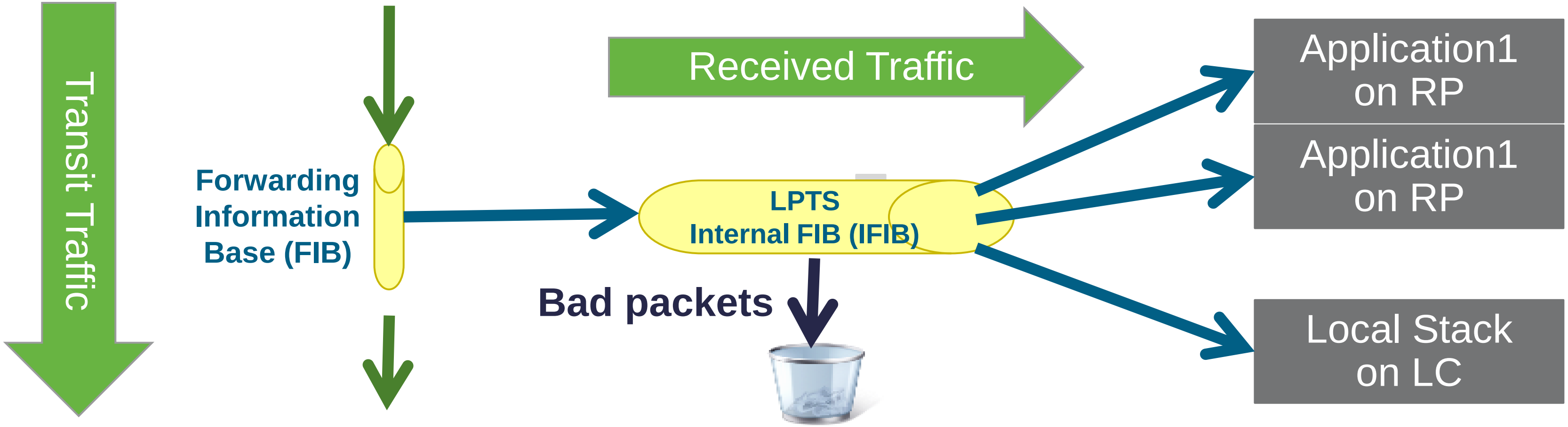


RP CPU: Routing, MPLS, IGMP, PIM, HSRP/VRRP, etc

LC CPU: ARP, ICMP, BFD, NetFlow, OAM, etc

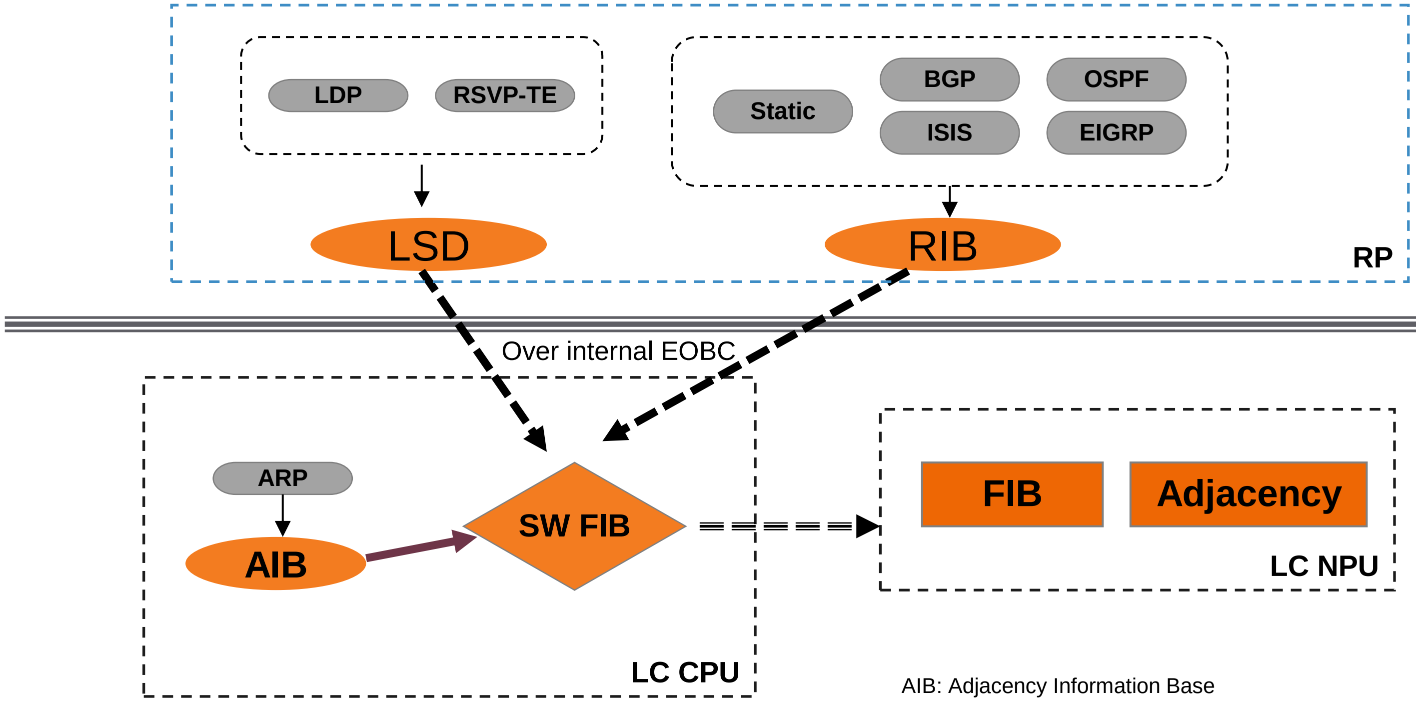
Local Packet Transport Services (LPTS)

“The” Control Plane Protection



- 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

Layer 3 Control Plane Overview



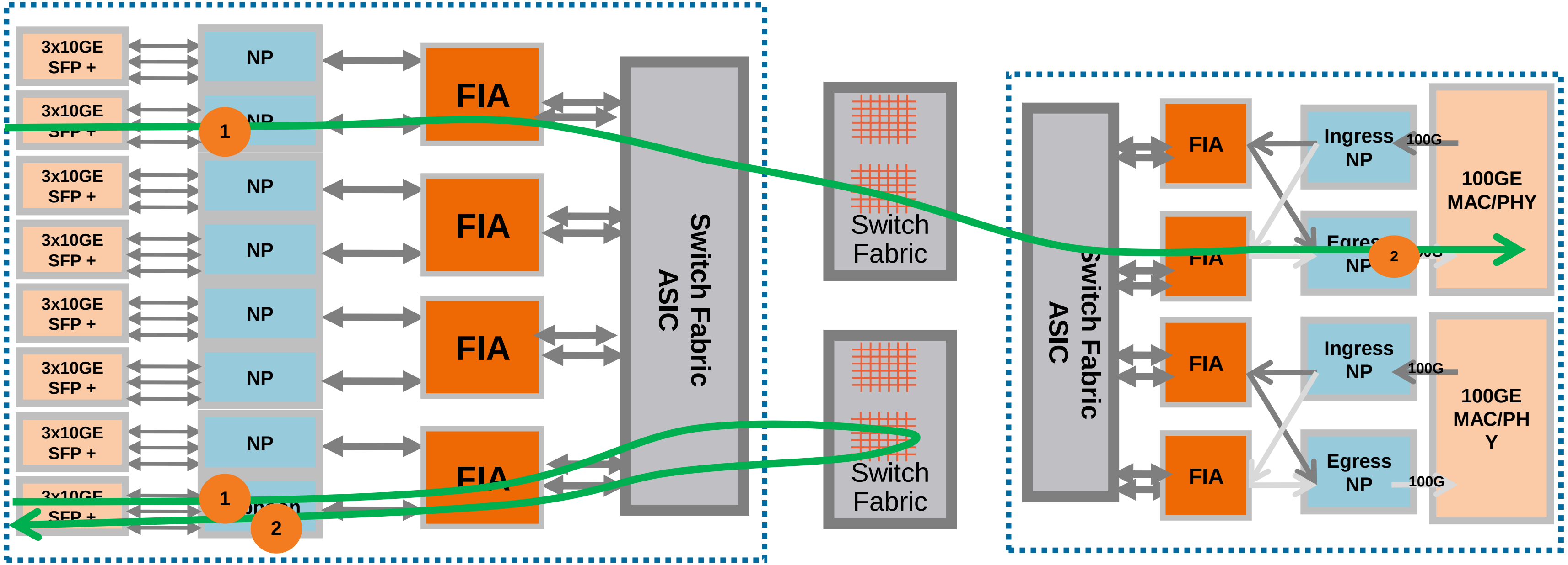
Selective VRF download per Line card for high scale

Hierarchical FIB table structure for prefix independent convergence: TE/FRR, IP/FRR, BGP, Link bundle

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

IOS-XR Two-Stage Forwarding Overview

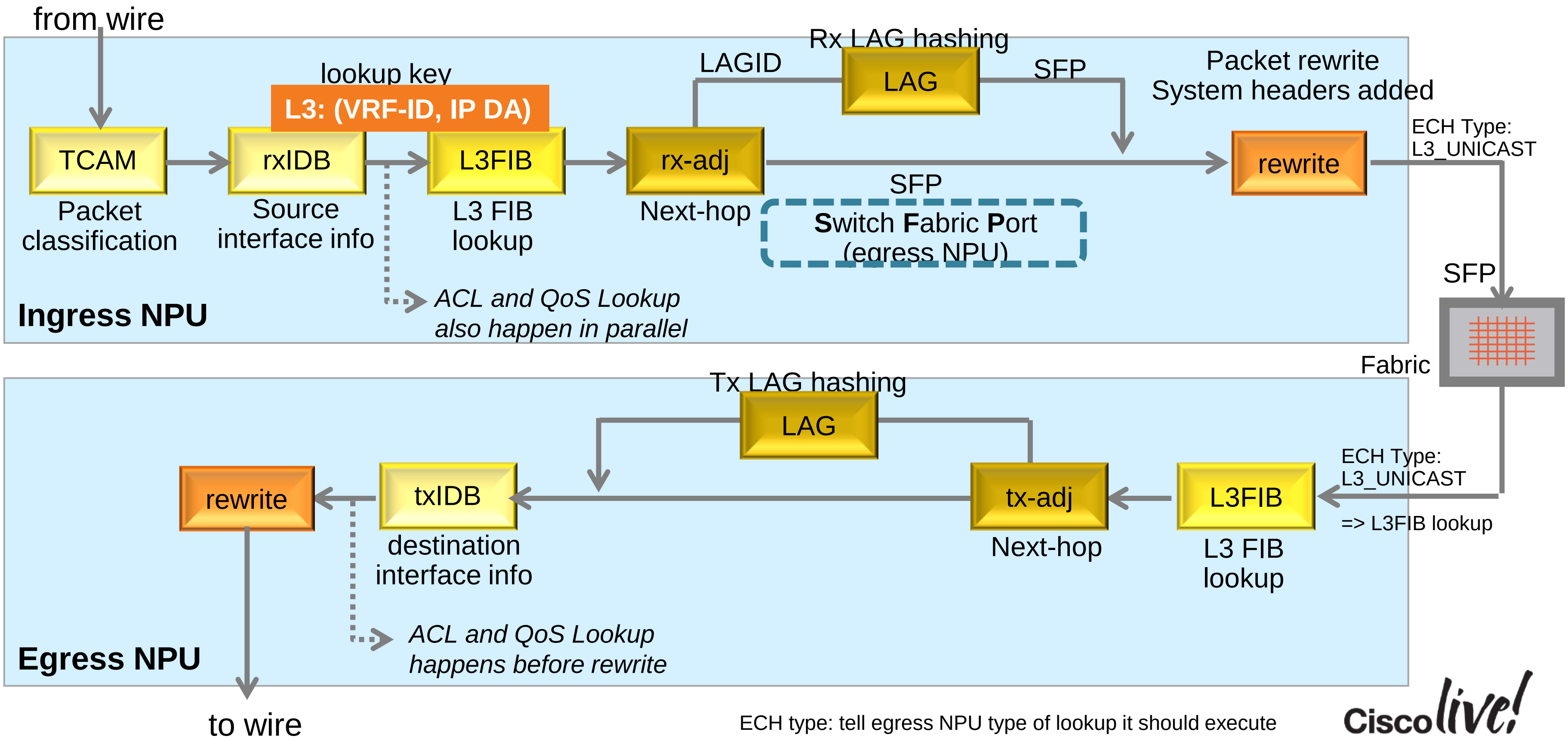
Scalable and Predictable



Uniform packet flow for simplicity and predictable performance

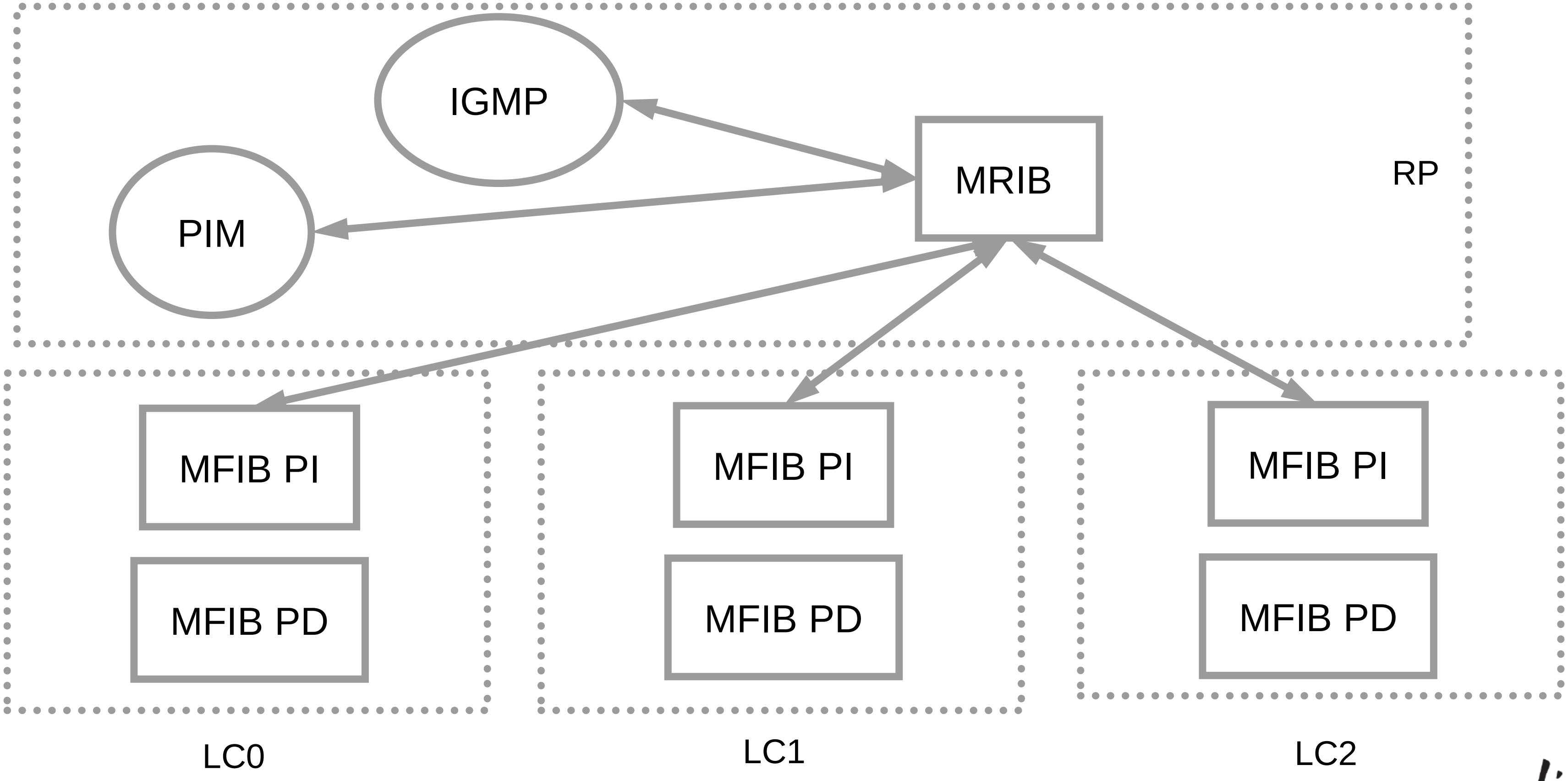
L3 Unicast Forwarding

Packet Flow (Simplified) Example



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

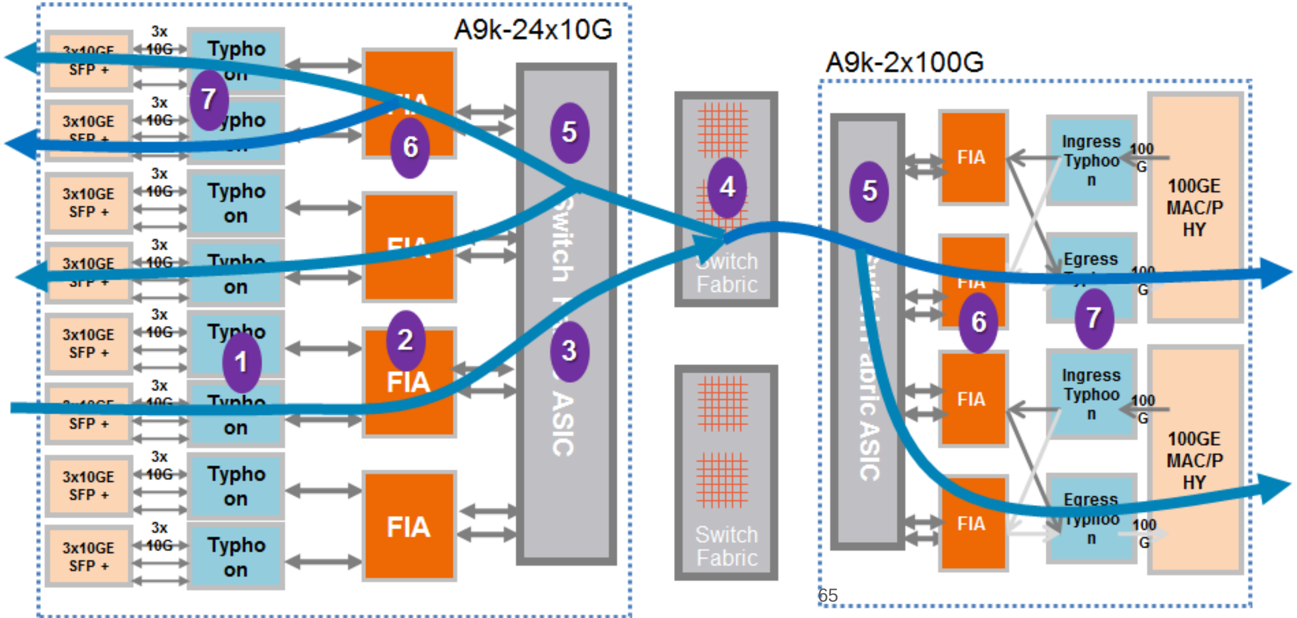
L3 Multicast Software Architecture – MRIB/MFIB



Multicast Replication Model Overview

2-Stage Replication

- Multicast Replication in ASR9k is like an SSM tree
- 2-stage replication model:
  - **Fabric to LC replication**
  - **Egress NP OIF replication**
- ASR9k doesn't use inferior "binary tree" or "root uniary tree" replication model

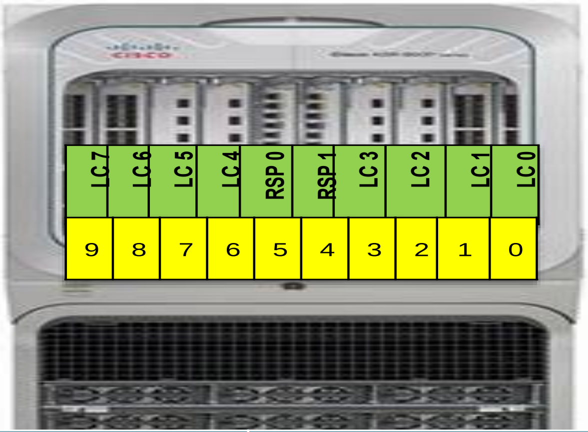


Important ASR9k MFIB Data-Structures

- FGID = Fabric Group ID
  1. FGID Index points to (slotmask, fabric-channel-mask)
  2. Slotmask, fabric-channel-mask = simple bitmap
- MGID = Multicast Group ID (S,G) or (*,G)
- 4-bit RBH
  1. Used for multicast load-balancing chip-to-chip hashing
  2. Computed by ingress NP ucode using these packet fields:
  3. IP-SA, IP-DA, Src Port, Dst Port, Router ID
- FPOE = FGID + 4-bit RBH

FGID (Slotmask)

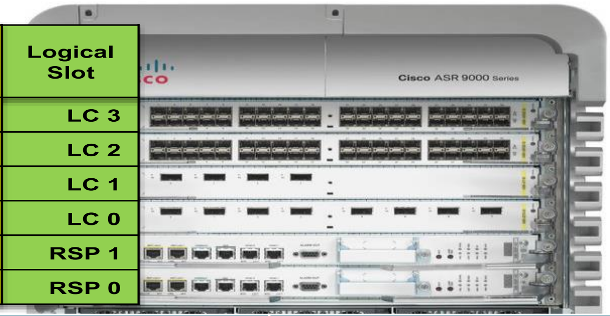
FGIDs: 10 Slot Chassis



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

FGIDs: 6 Slot Chassis

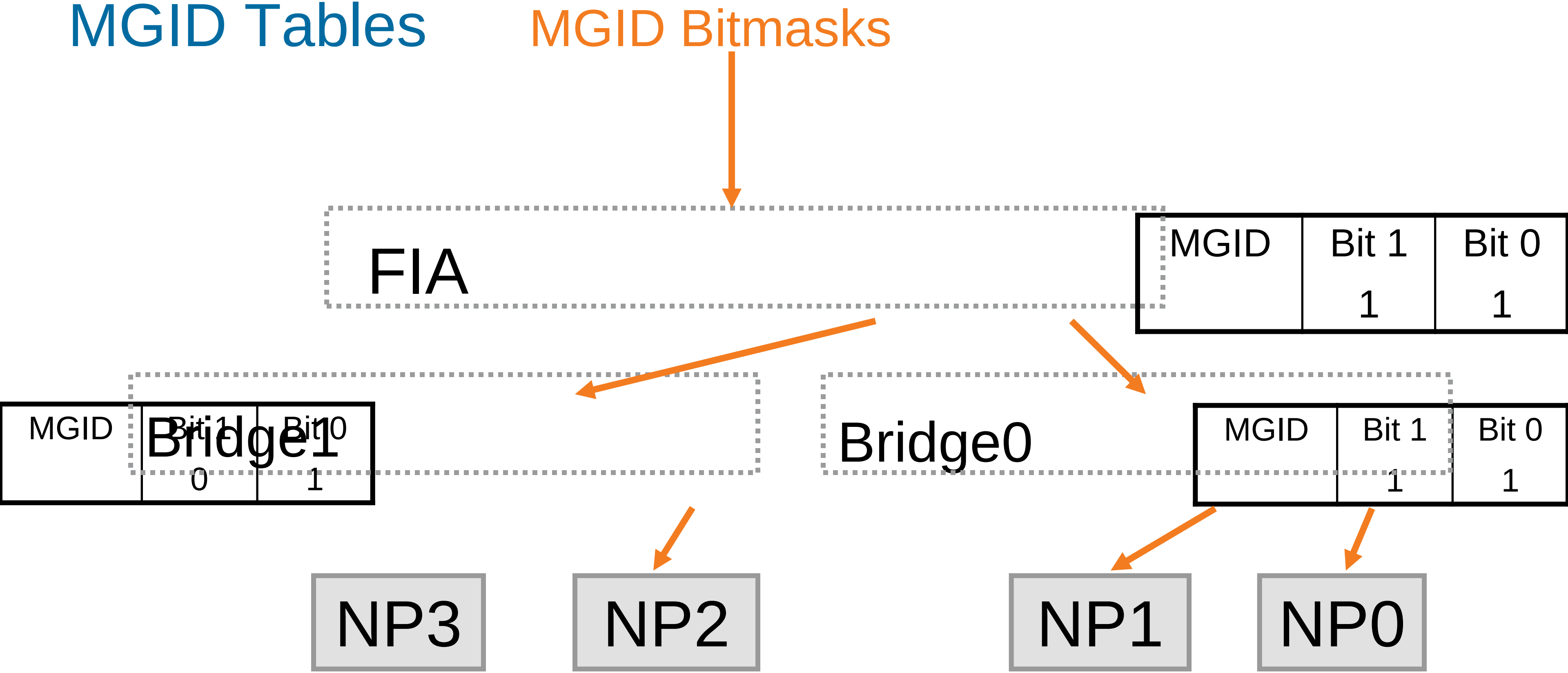
Phy Slot Number	Logical Slot
5	LC 3
4	LC 2
3	LC 1
2	LC 0
1	RSP 1
0	RSP 0



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

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

MGID Tables

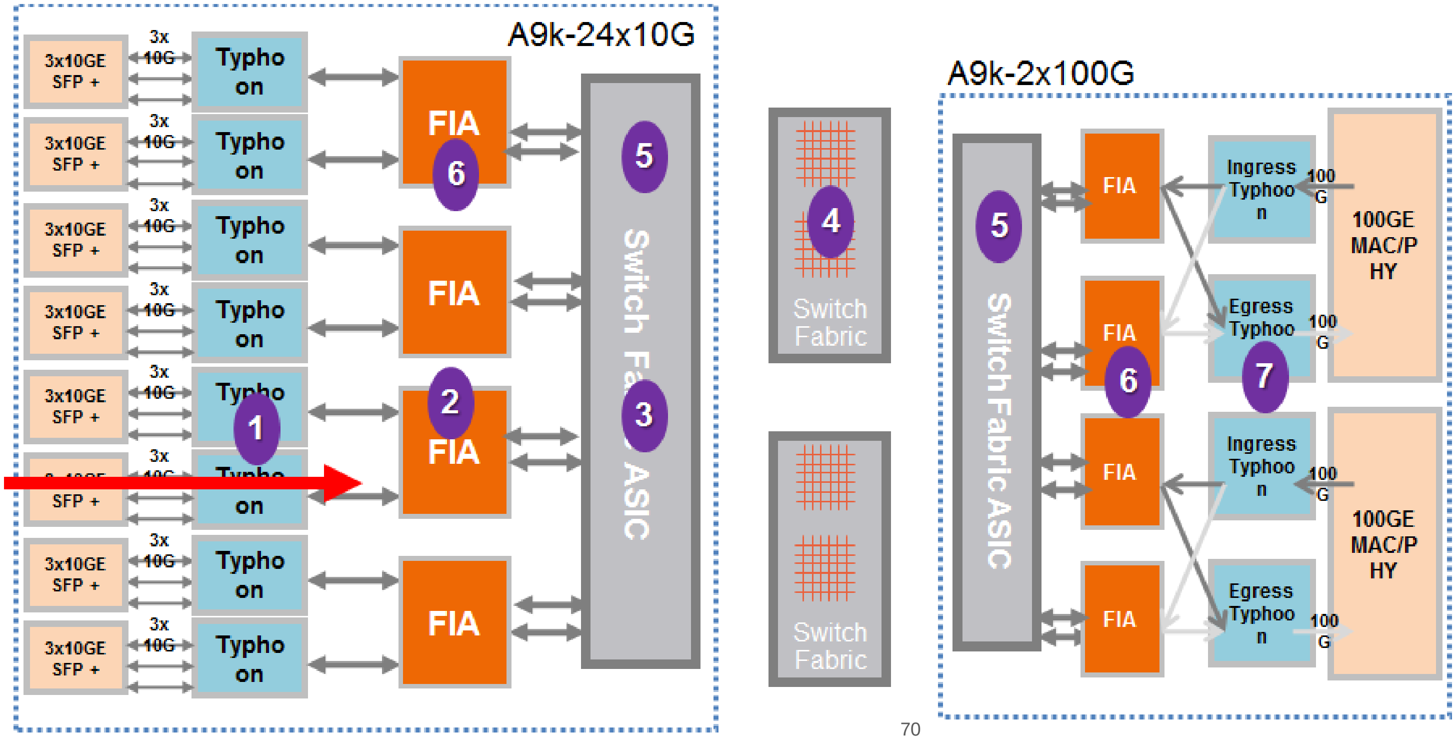

MGID Bitmasks
FIA
MGID | Bit 1 | Bit 0
1 | 1
Bridge1
MGID | Bit 1 | Bit 0
0 | 1
Bridge0
MGID | Bit 1 | Bit 0
1 | 1
NP3
NP2
NP1
NP0

MGID Allocation in ASR9k

- A MGID is allocated per L2/L3/MPLS multicast route
- Typhoon LCs support 512k MGIDs per system which are allocated by the MGID server
- They are fully backward compatible to Trident (1st Gen) and SIP700 cards
- MGID space allocation is as follows:
    1. 0 – (32k-1): Bridge domains in mixed LC system
    2. 32k – (64k-1): IP and L2 multicast in mixed LC system
    3. 64k – (128k-1): Reserved for future Bridge domain expansion on Typhoon LCs
    4. 128k – (512k-1): IP and L2 multicast on Typhoon LCs

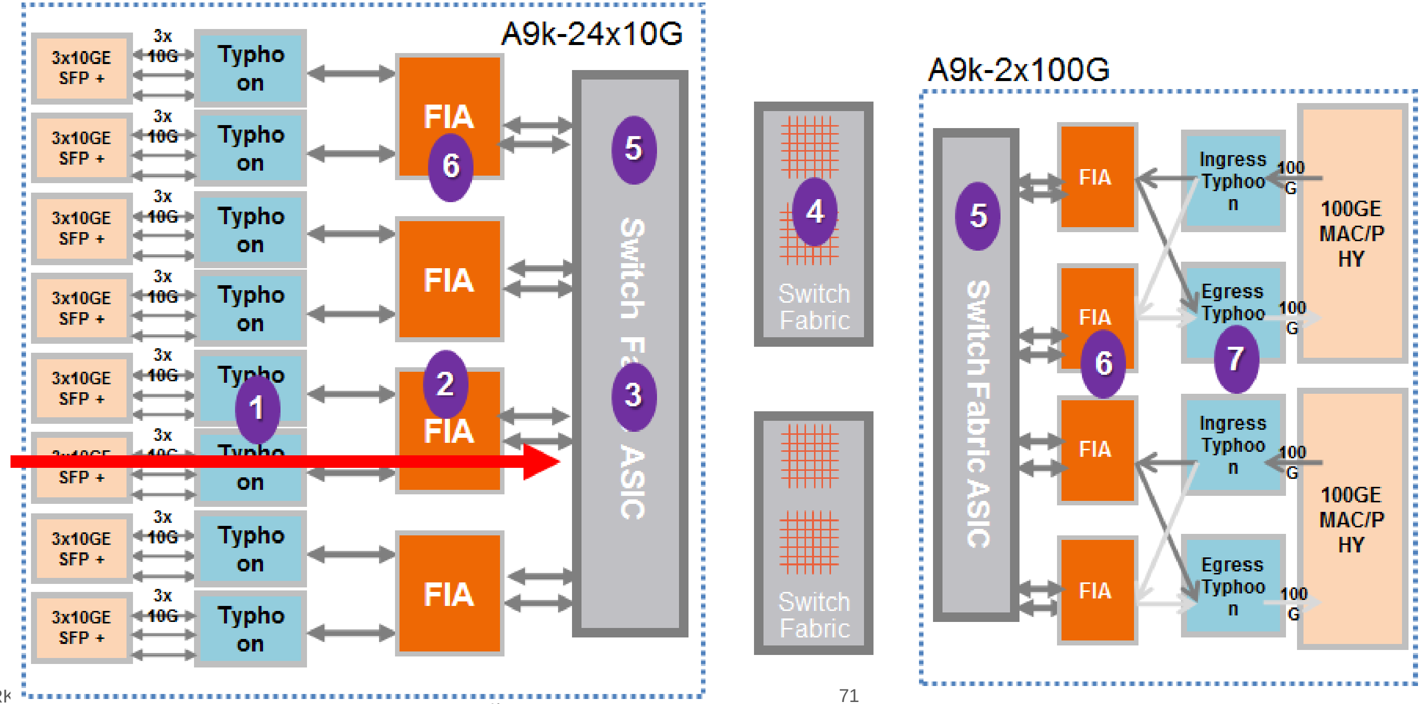
Multicast Replication Model Overview
Step 1

- **Ingress NPU:**
  1. MFIB (S,G) route lookup yields {FGID, MGID, Olist, 4-bit RBH} data-structures
  2. Ingress NPU adds **FGID**, MGID, 4-bit RBH in fabric header to FIA



Multicast Replication Model Overview
Step 2

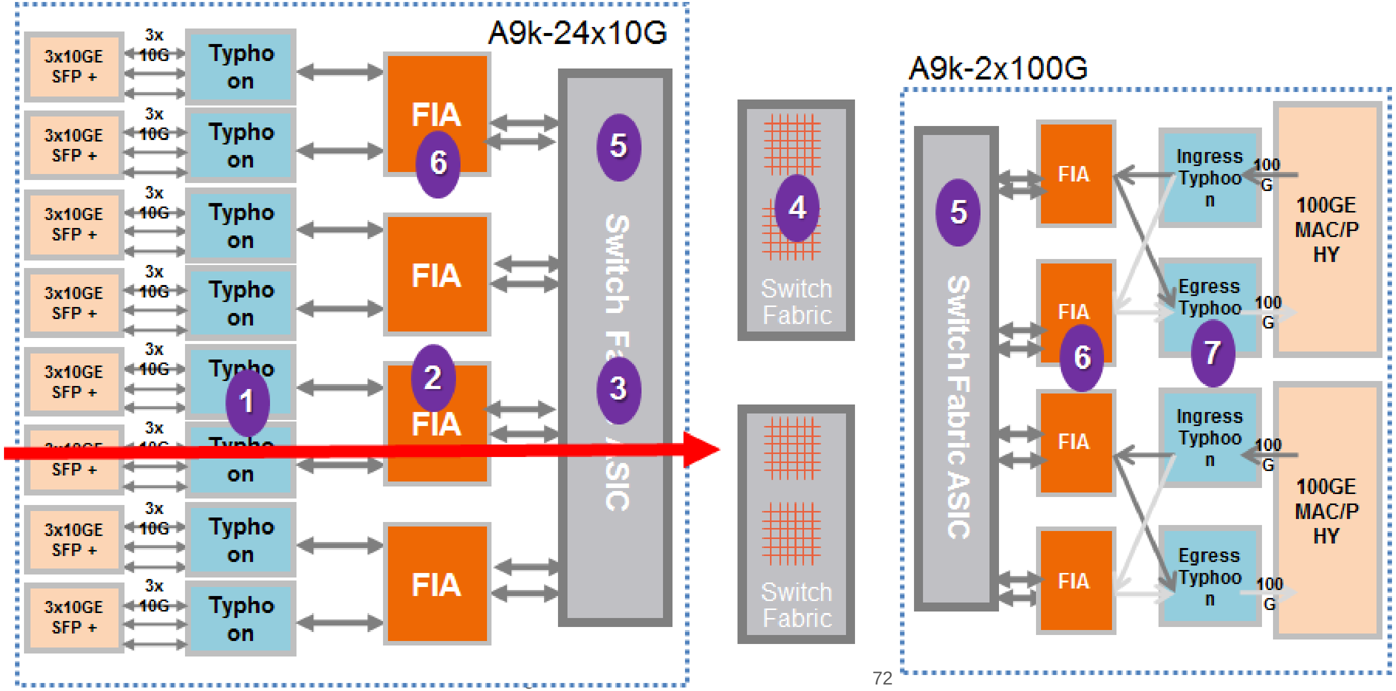
- **Ingress FIA:**
  1. Load-balance multicast traffic from FIA to LC Fabric



Multicast Replication Model Overview

Step 3

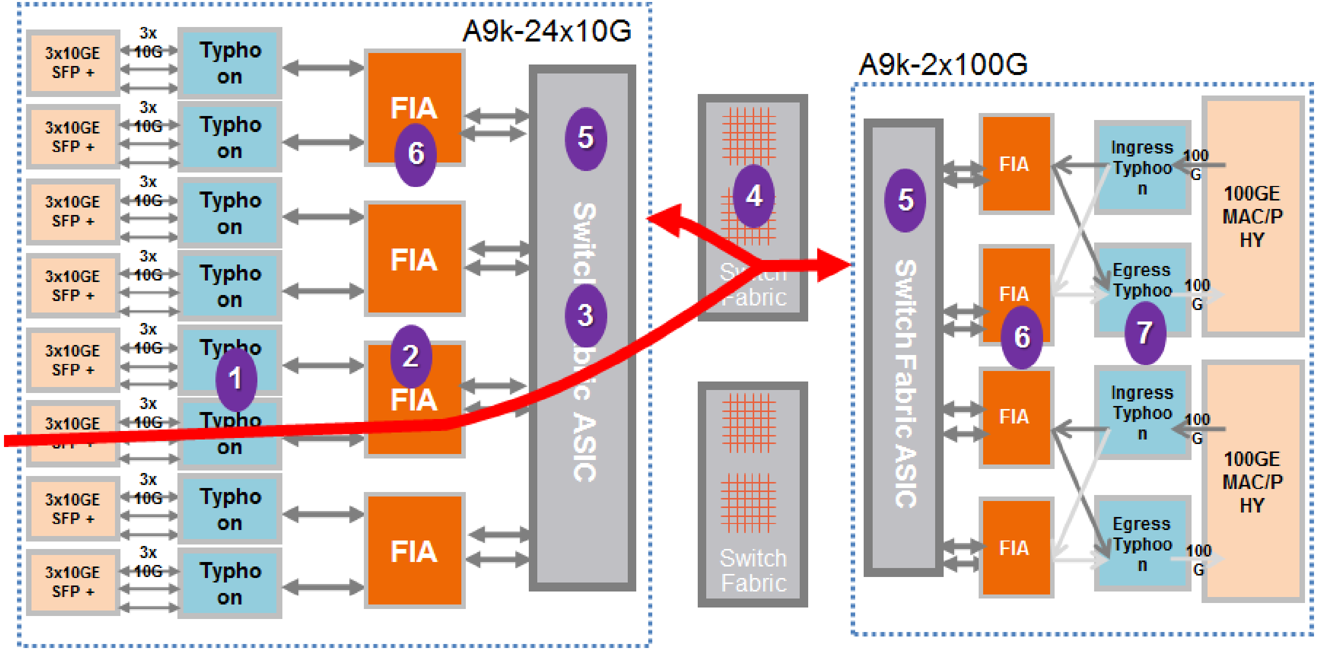
- **Ingress LC Fabric:**
  1. Reads FPOE bits in the fabric header AND reads 3-bits of derived RBH
  2. It will load-balance MGID towards any of the 8 fabric channels
  3. Now it send traffic to central fabric over 1 of the fabric channels per MGID
     - (Note: there are only upto 8 fabric-channel links to central fabrlc)



Cisco live!

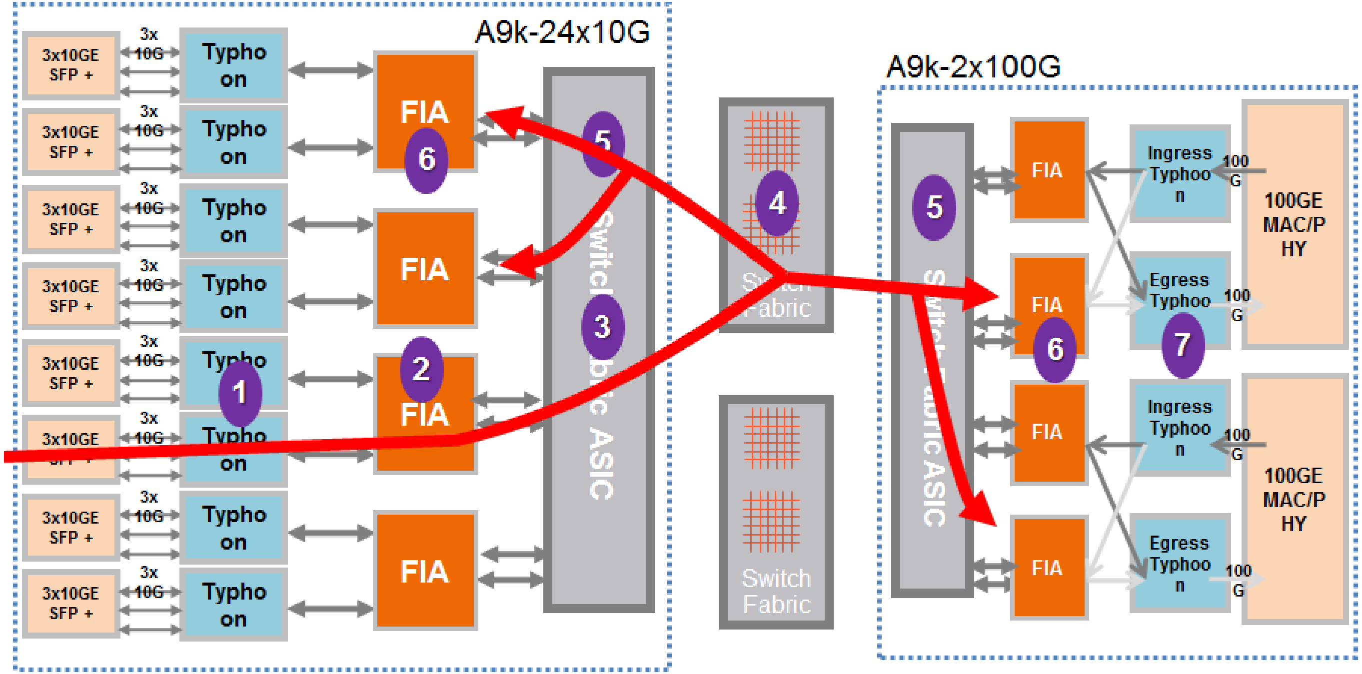
Multicast Replication Model Overview
Step 4

- **RSP Fabric Replication to Egress LC Fabric:**
  1. Receives 1 copy from ingress LC
  2. Reads fabric header FGID slotmask value to lookup the FPOE table to identify which fabric channel output ports to replicate to
  3. Now it replicates 1 copy to egress LCs with multicast receivers



Multicast Replication Model Overview
Step 5

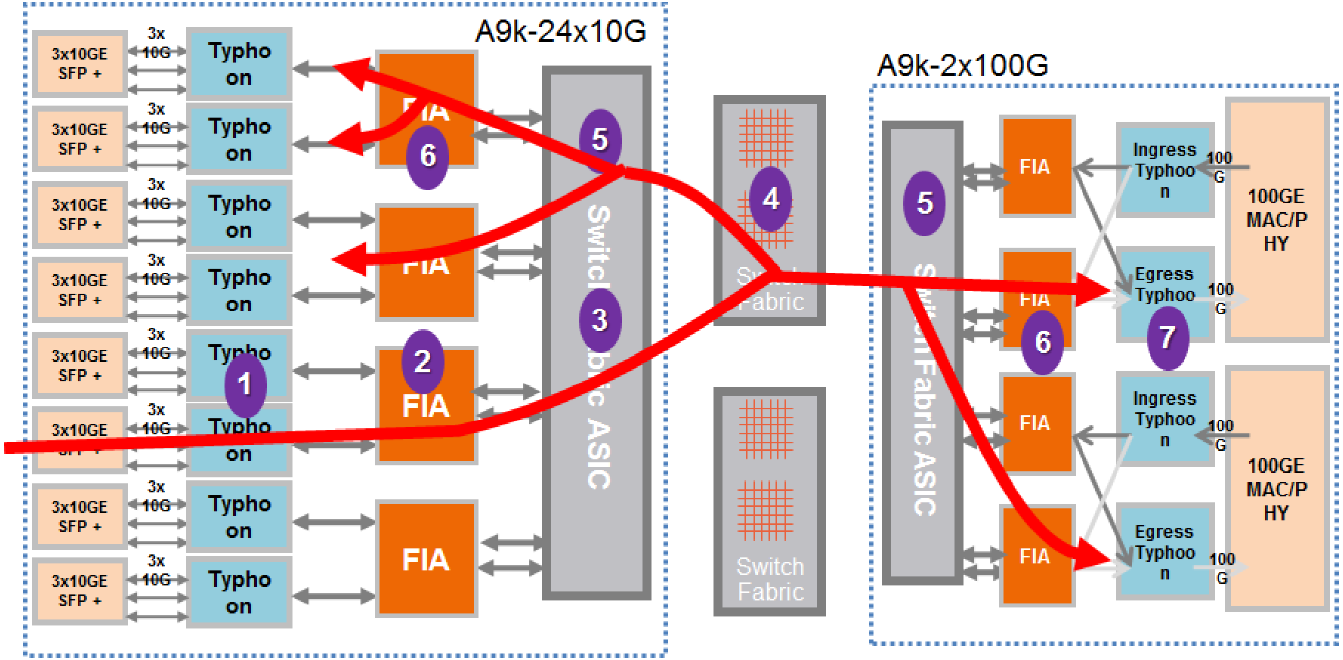
- **Egress LC Fabric Replication to FIA:**
  1. Egress LC fabric is connected to all the FIAs (ie. upto 6 FIAs in A9k-36x10G) card
  2. All MGIDs (ie. mroute) are mapped into 4k FPOE table entries in LC fabric
  3. Looks up FPOE index and replicate the packets mapped to egress FIAs with MGID receiver



Multicast Replication Model Overview

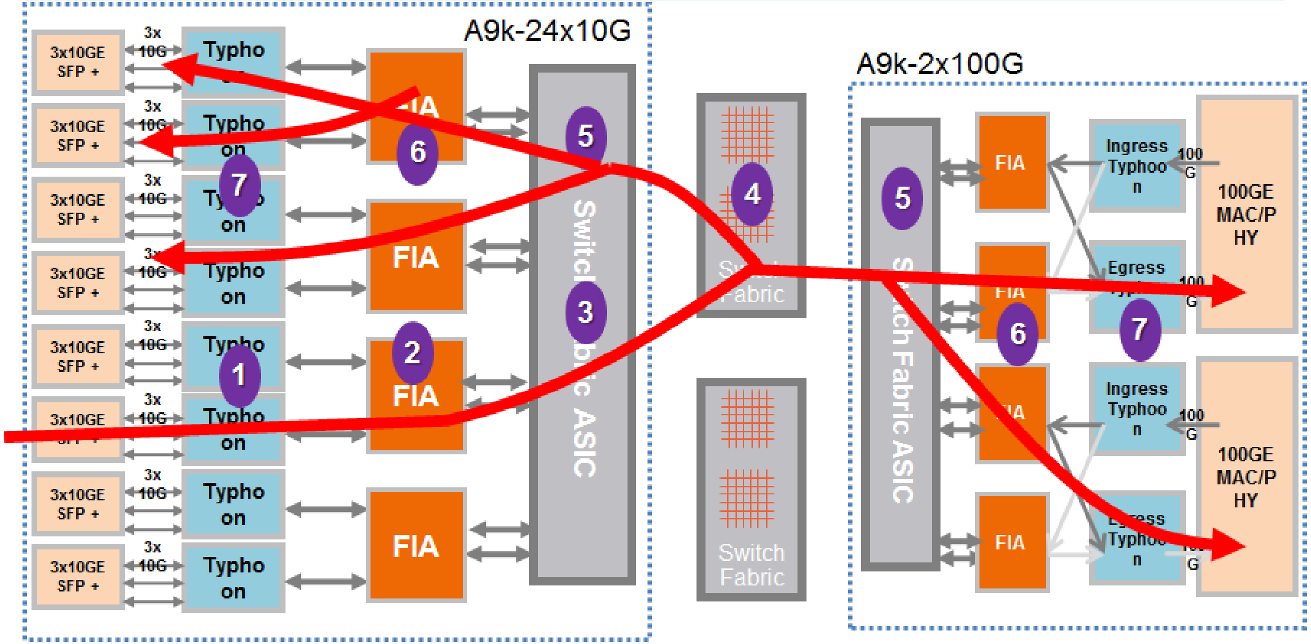
Step 6

- **Egress FIA Replication to Typhoon NPU**
  1. Egress FIA has 256k MGIDs (ie. mroutes), 1 MGID is allocated per mroute
  2. Each MGID in the FIA is mapped to its local NPUs
  3. Performs a 19-bit MGID lookup of incoming mcast packet from LC fabric
  4. Replicates 1 copy to each Typhoon NPU with mroute receivers



Multicast Replication Model Overview
Step 7

- **Egress Typhoon NPU Multicast OIF Replication**
  1. Egress NPU performs L2/L3/MPLS multicast OIF replication (2nd stage lookup)
  2. MGID lookup yields OIF count (ie. replication interface count)
  3. When OIF count == 1, then NPU replicate all L2/L3/MPLS multicast traffic in 1st pass
  4. When OIF count > 1, then NPU replicate all L2/L3/MPLS multicast traffic in 2nd pass
  5. (S,G), (*,G)



Cisco live!

L2 Service Framework: Cisco EVC

Most Flexible Carrier Ethernet Service Architecture: any service any port, any VLAN to any VLAN

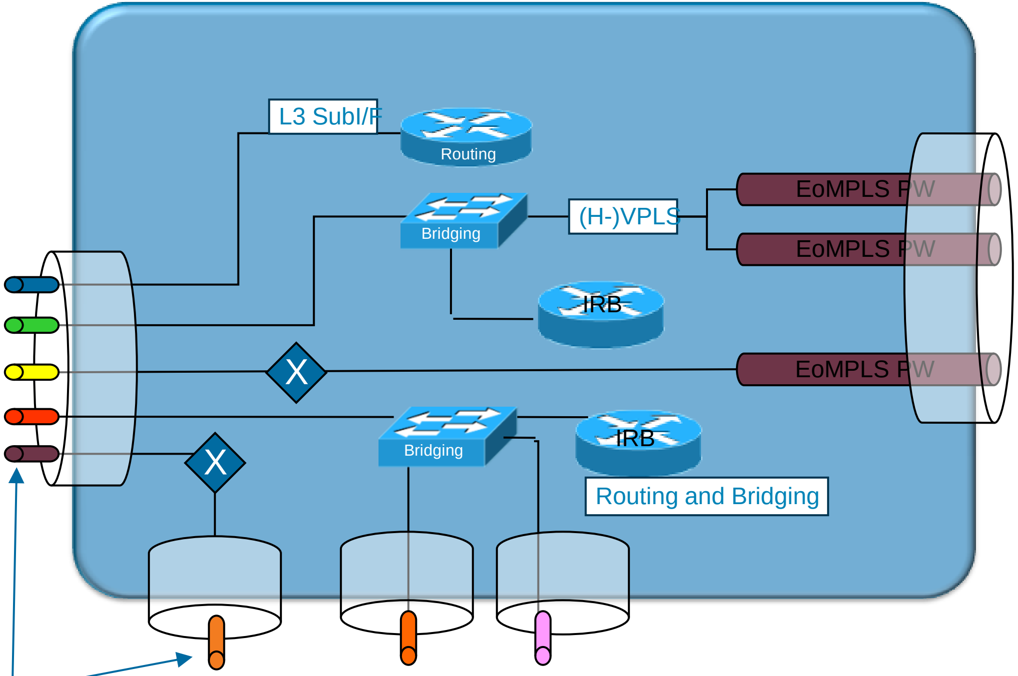
VLAN tag local significant

Flexible VLAN tag classification

Flexible VLAN tag rewrite

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

1



L2 or L3 sub-interfaces (802.1a/qinq/.1ad)

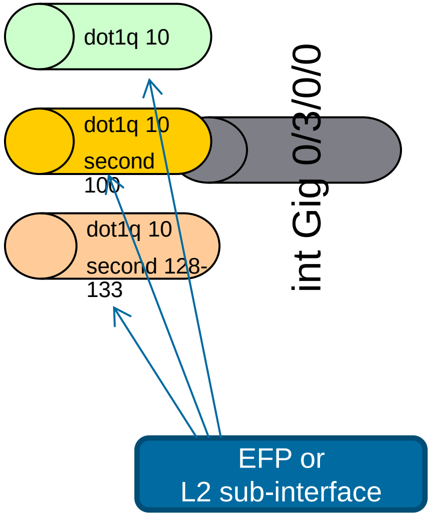
2

Flexible service mapping and multiplexing. Support all standard based services concurrently on the same port:

Regular L3, L2 interface/sub-interface
Integrated L2 and L3 – IRB/BVI
Mixed L2 and L3 sub-interfaces on the same port

Flexible VLAN Tag Classification

```
RP/0/RSP0/CPU0:PE2-asr(config)#int gig 0/3/0/0.100 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 10
  comma  comma
  exact  Do not allow further inner tags

RP/0/RSP0/CPU0:PE2-asr(config-subif)#encapsulation dot1q 10 second-dot1q 100 ?
  comma  comma
  exact  Do not allow further inner tags

RP/0/RSP0/CPU0:PE2-asr(config-subif)#encapsulation dot1aq 10 second-dot1q 128-133 ?
  comma  comma
  exact  Do not allow further inner tags
```



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 ?
  dot1ad  Push a Dot1ad tag
  dot1q   Push a Dot1Q tag
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

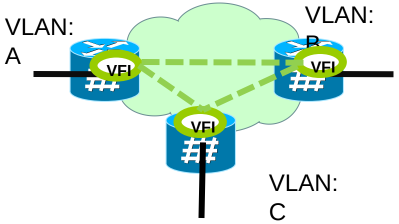
Tag translation

1-1

1-2

2-1

2-2



Any VLAN to any VLAN: single or double tags, dot1q or dot1ad

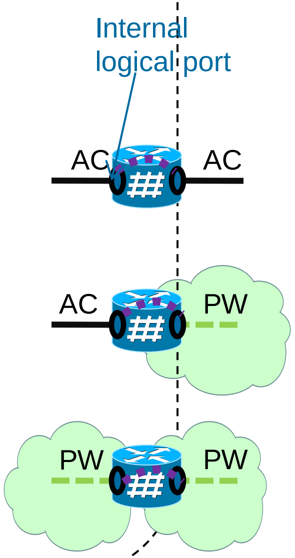
L2VPN P2P

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
```

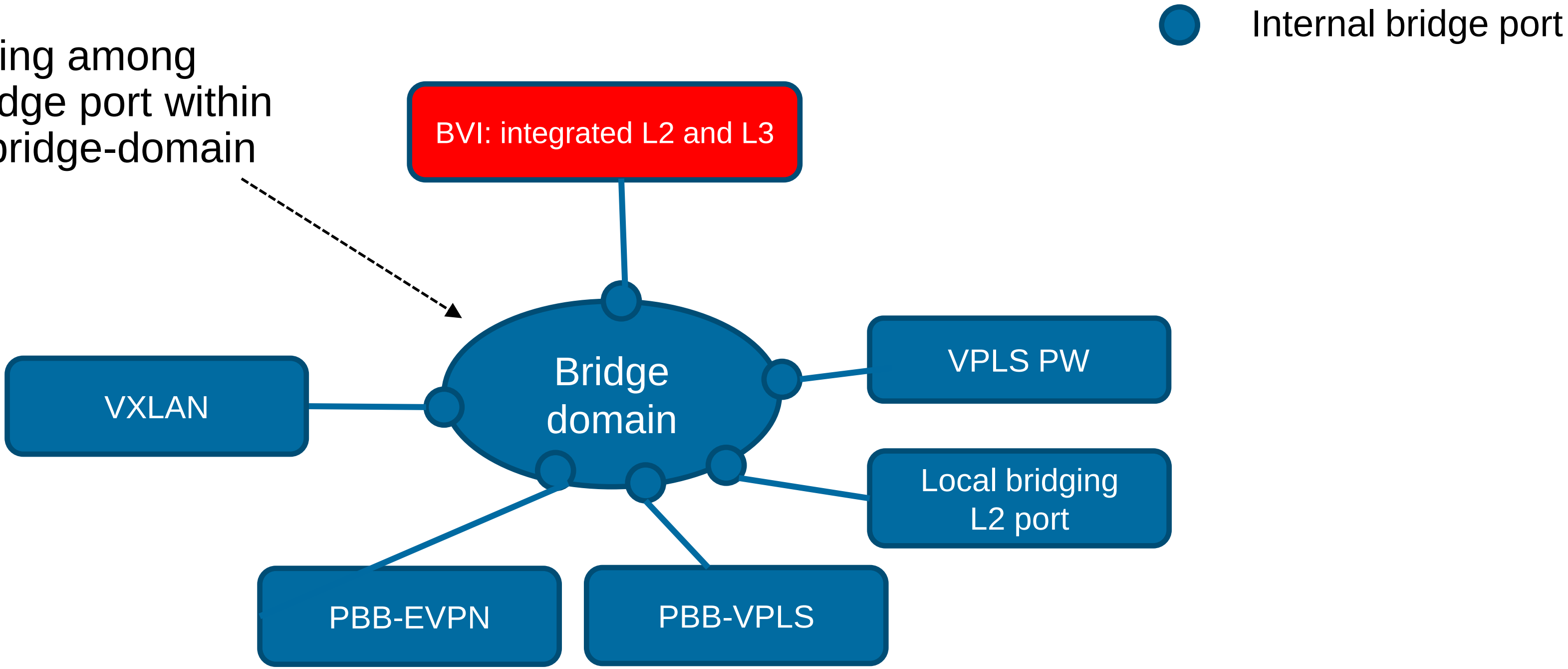
L2VPN P2P service configuration example

```
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 Multipoint Bridging Architecture

MAC bridging among internal bridge port within the same bridge-domain



* Not in 5.2.0

L2VPN Multi-Point (1): local bridging, vpls, h-vpls

EFP configuration example

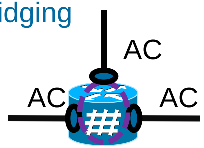
Interface gig 0/0/0/1.101 l2transport
encapsulation dot1q 101
rewrite ingress pop 1 Symmetric

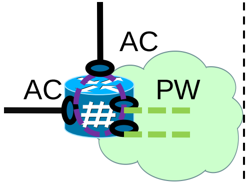
Interface gig 0/0/0/2.101 l2transport
encapsulation dot1q 101
rewrite ingress pop 1 Symmetric

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
   Interface gig 0/0/0/2.101
   Interface gig 0/0/0/3.101

  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
   neighbor 192.0.0.3 pw-id 100 ← spoke PW
   vfi cisco
    neighbor 192.0.0.1 pw-id 100
    neighbor 192.0.0.2 pw-id 100
```



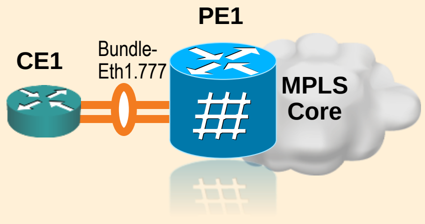


A Simple PBB-EVPN CLI Example

Please refer to session xxx for details:

MINIMAL Configuration

Default B-MAC SA
Auto RT for EVI
Auto RD for EVI
Auto RD for Segment Route



PE1

```
interface Bundle-Ether1.777 l2transport
  encapsulation dot1q 777

l2vpn
 bridge group gr1
  bridge-domain bd1
   interface Bundle-Ether1.777
   pbb edge i-sid 260 core-bridge-domain core_bd1

 bridge group gr2
  bridge-domain core_bd1
   pbb core
    evpn evi 1000

router bgp 64
 address-family l2vpn evpn
 !
 neighbor <x.x.x.x>
  remote-as 64
  address-family l2vpn evpn
```

PBB B-component
No need to define B-VLAN

Mandatory - Globally unique identifier for all PEs in a given EVI

BGP configuration with new EVPN AF

VXLAN L3 Gateway CLI Example

```
RP/0/0/CPU0:r1(config)# interface nve 1
RP/0/0/CPU0:r1(config-if)# encapsulation vxlan
RP/0/0/CPU0:r1(config-if)# source-interface loopback 0
RP/0/0/CPU0:r1(config-if)# vni 65001-65010 mcast 239.1.1.1
RP/0/0/CPU0:r1(config-if)# vni 65011 mcast 239.1.1.2
! 1:1 or N:1 mapping between VNIs and vxlan multicast delivery group
```

```
RP/0/0/CPU0:r1(config)#l2vpn
RP/0/0/CPU0:r1(config-l2vpn)#bridge group customer1
RP/0/0/CPU0:r1(config-l2vpn-bg)#bridge-domain cu-l3vpn
RP/0/0/CPU0:r1(config-l2vpn-bg-bd)#member vni 65001
RP/0/0/CPU0:r1(config-l2vpn-bg-bd)#routed interface 101
```

```
RP/0/0/CPU0:r1(config)#interface BVI 101
RP/0/0/CPU0:r1(config-if)#ipv4 address 100.1.1.1/24
RP/0/0/CPU0:r1(config-if)#ipv6 address 100:1:1::1/96
! Can apply any existing features like QoS, ACL, Netflow, etc under BVI interface
```

VXLAN L2 Gateway CLI Example

```
RP/0/0/CPU0:r1(config)# interface nve 1
RP/0/0/CPU0:r1(config-if)# encapsulation vxlan
RP/0/0/CPU0:r1(config-if)# source-interface loopback 0
RP/0/0/CPU0:r1(config-if)# vni 65001-65010 mcast 239.1.1.1
RP/0/0/CPU0:r1(config-if)# vni 65011 mcast 239.1.1.2
! 1:1 or N:1 mapping between VNIs and vxlan multicast delivery group
```

```
RP/0/0/CPU0:r1(config)#l2vpn
RP/0/0/CPU0:r1(config-l2vpn)#bridge group customer1
RP/0/0/CPU0:r1(config-l2vpn-bg)#bridge-domain cu-l2vpn
RP/0/0/CPU0:r1(config-l2vpn-bg-bd)#interface GigabitEthernet0/2/0/0.100
RP/0/0/CPU0:r1(config-l2vpn-bg-bd)#member vni 65001
```

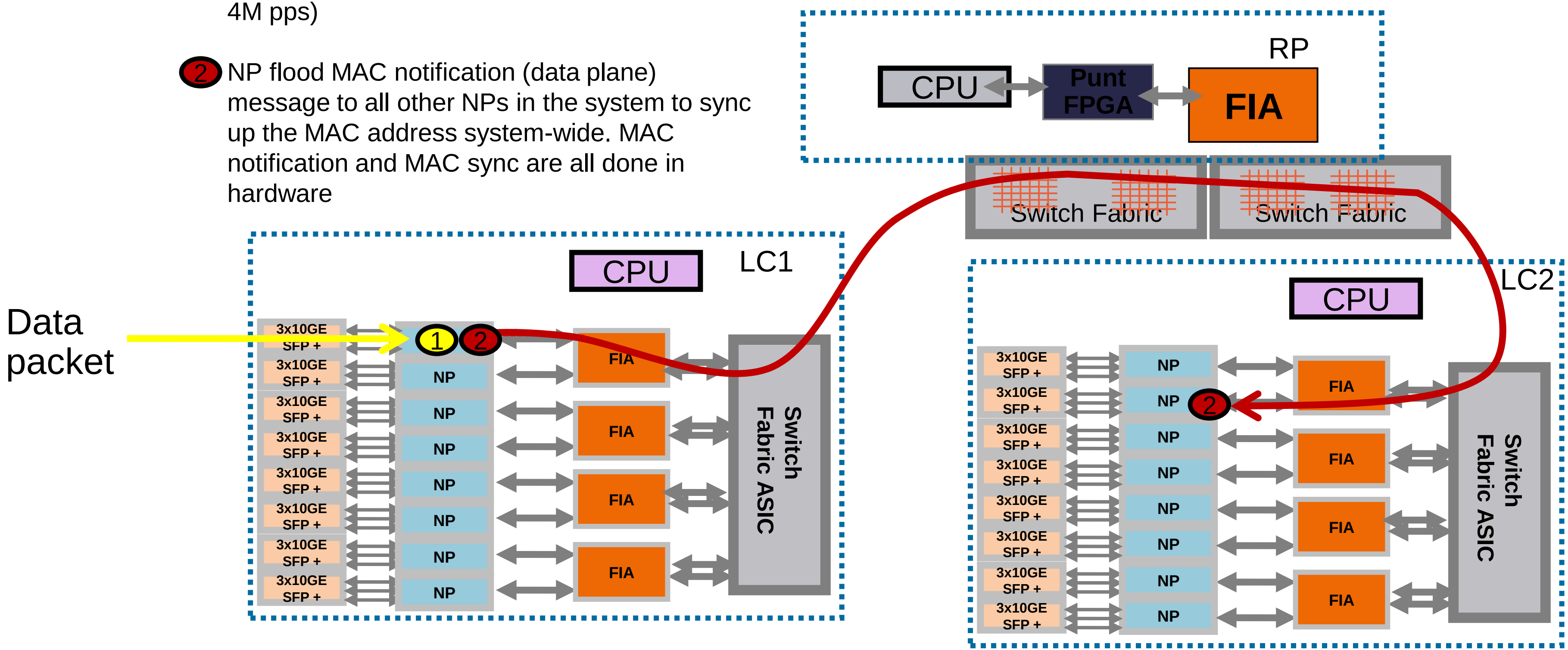
```
RP/0/0/CPU0:r1(config)#interface GigabitEthernet0/2/0/0.100 l2transport
RP/0/0/CPU0:r1(config-subif)#dot1q vlan 100
```


MAC Learning and Sync

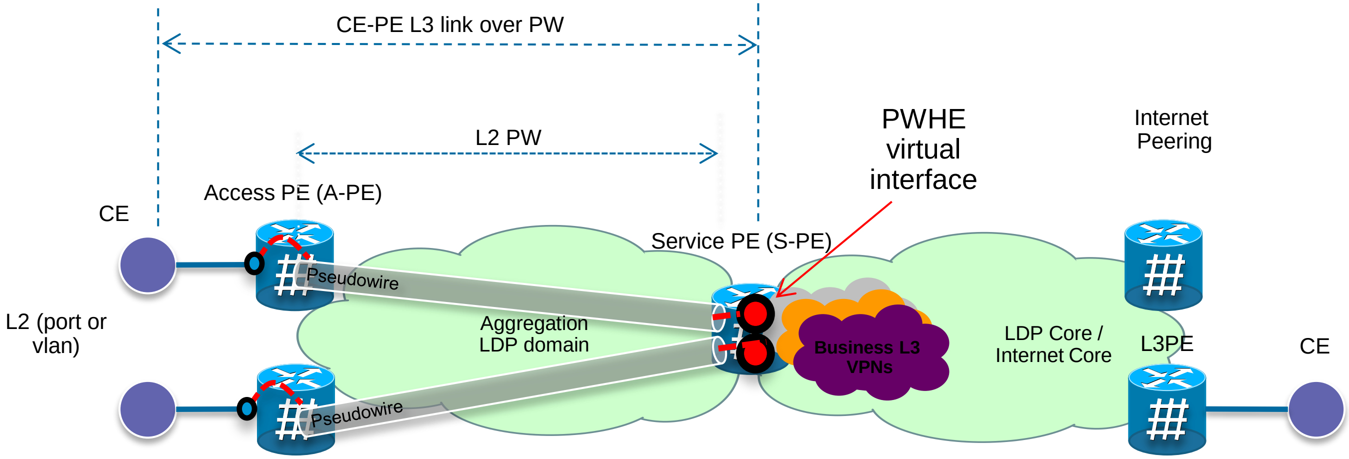
Hardware based MAC learning: ~4Mpps/NP

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

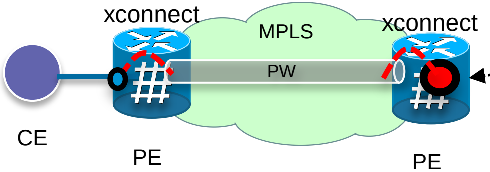


Virtual Service Interface: PWHE Interface



- 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

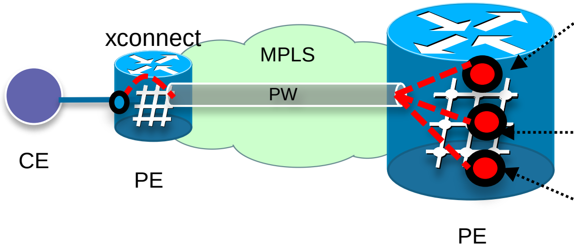
PWHE Configuration Examples



PWHE L3 interface Example

```
interface pw-ether 200
    vrf vrf0001
    ipv4 address 11.0.0.1 255.255.255.0
    ipv6 address 2001:da1::1/64
    load-interval 30
```

```
l2vpn
 xconnect group pwhe
  p2p pwhe-red
   interface pw-ether 100
   neighbor 100.100.100.100 pw-id 1
```



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
```

```
l2vpn
 xconnect group pwhe
  p2p pwhe-red
   interface pw-ether 100
   neighbor 100.100.100.100 pw-id 1

xconnect group cisco
p2p service2
       Interface pw-ether 100.200
       neighbor 1.1.1.1 pw-id 22

bridge-domain domain2
   Interface pw-ether 100.300
   vfi cisco
    neighbor 192.0.0.1 pw-id 100
    neighbor 192.0.0.2 pw-id 100
```


Cisco live!

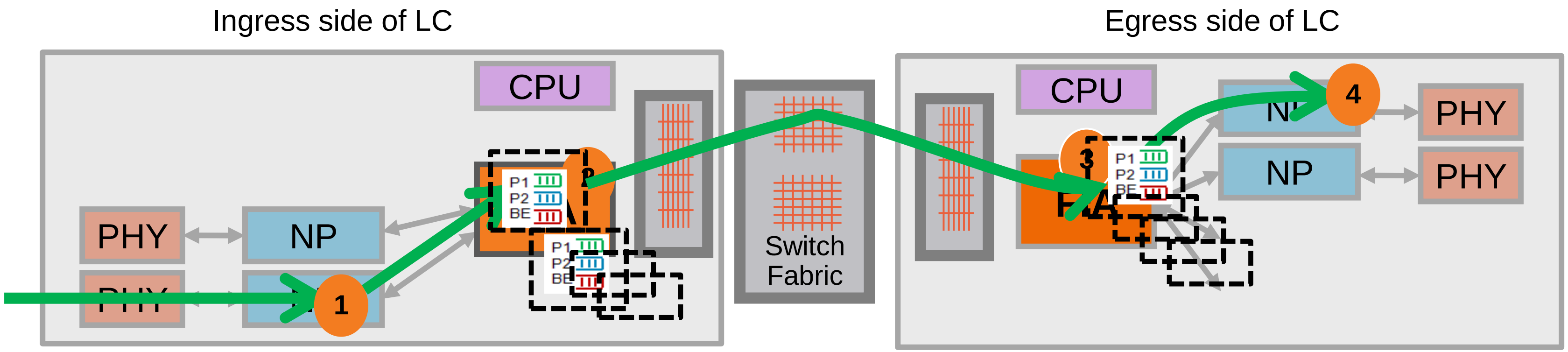
ASR 9000 Software System Architecture (3)

Queuing

System QoS Overview

Port/LC QoS and Fabric QoS

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



1 Ingress Port QoS

2 4 VOQ per each virtual port in the entire system

Up to 4K VOQs per FIA

3 4 Egress Queues per each virtual port, aggregated rate per NP

4 Egress Port QoS

Line Card QOS Overview (1)

- The user configure QoS policy using IOS XR MQC CLI
- QoS policy is applied to interface (physical, bundle or logical*), attachment points
  - Main Interface

    MQC applied to a physical port will take effect for traffic that flows across all sub-interfaces on that physical port
    - ✓ will NOT coexist with MQC policy on sub-interface **
    - ✓ you can have either port-based or subinter-face based policy on a given physical port
  - L3 sub-interface
  - L2 sub-interface (EFP)
- QoS policy is programmed into hardware microcode and queue ASIC on the Line card NPU

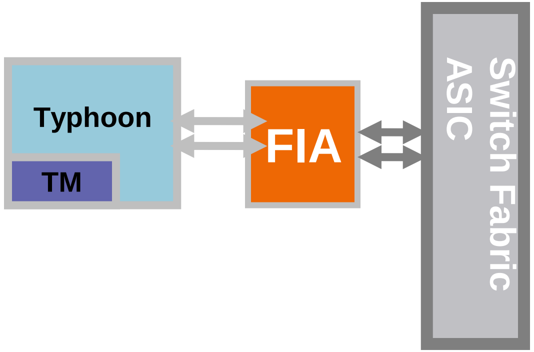
* Some logical interface could apply qos policy, for example PWHE and BVI

** it could have main interface level simple flat qos co-exist with sub-interface level H-QoS on ingress direction

Line Card QoS Overview (2)

Dedicated queue ASIC – **TM (traffic manager)** per each NP for the QoS function

-SE and –TR* LC version has different queue buffer/memory size, different number of queues



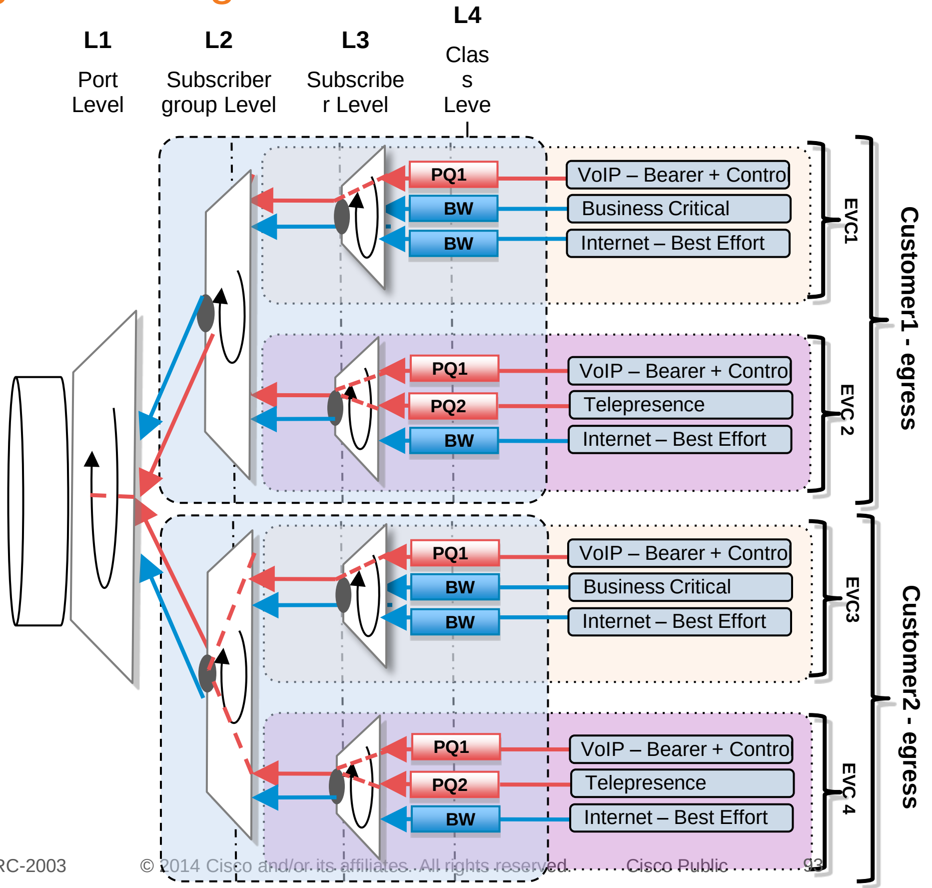
- High scale
  - –Up to 3 Million queues per system (with -SE linecard)
  - –Up to 2 Million policers per system (with -SE linecard
- Highly flexible: 4 layer hierarchy queuing/scheduling support
  - –Four layer scheduling hierarchy →Port, Subscriber Group, Subscriber, Class
  - –Egress & Ingress, shaping and policing
- Three strict priority scheduling with priority propagation
- Flexible & granular classification, and marking
  - –Full Layer 2, Full Layer 3/4 IPv4, IPv6

* 8 queues per port

LC QoS Overview (3): 4-Level Hierarchy QoS

Ingress* & Egress Direction

* Certain line card doesn't support ingress queuing



4-Level H-QoS supported in ingress and egress direction

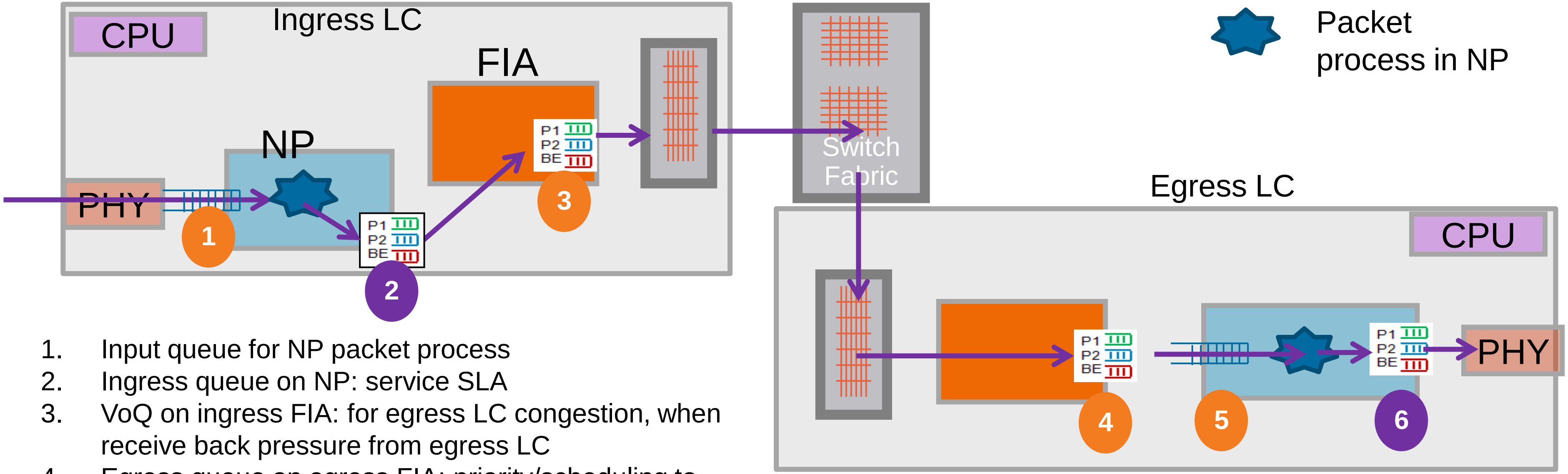
Note: We count hierarchies as follows:
4L hierarchy = 3 Level nested p-map
3L hierarchy = 2 level nested p-map

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

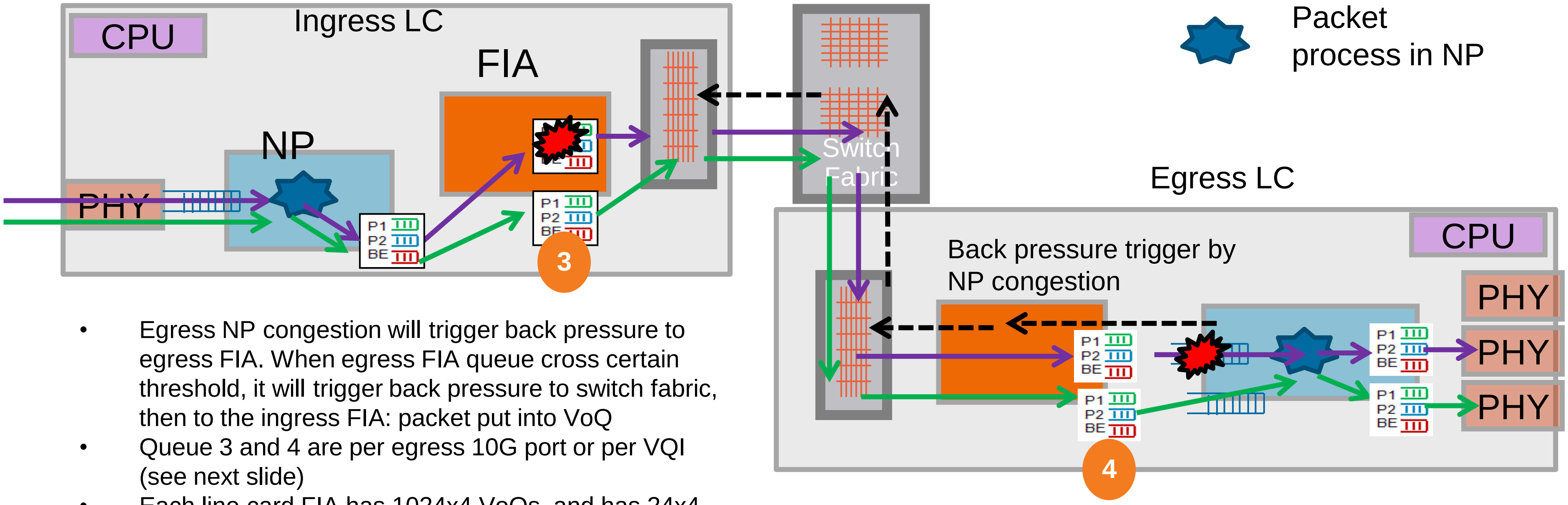
Internal QoS: End-to-End System Queuing



1. Input queue for NP packet process
2. Ingress queue on NP: service SLA
3. VoQ on ingress FIA: for egress LC congestion, when receive back pressure from egress LC
4. Egress queue on egress FIA: priority/scheduling to egress NP
5. Input queue for NP packet process. When queue build up, it will trigger back pressure to FIA
6. Egress queue on NP: link congestion and service SLA

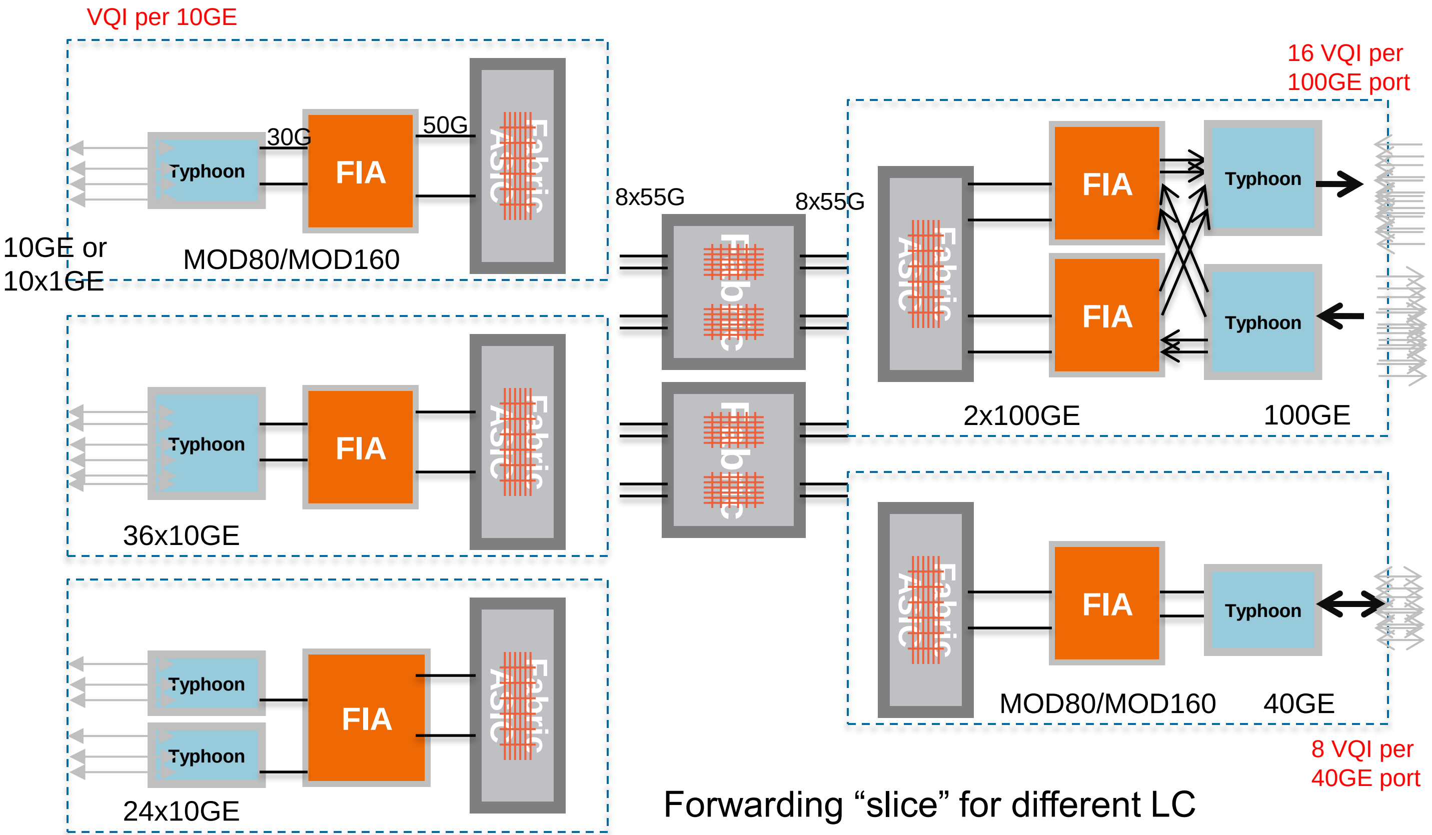
- Queue 2, 3,4 (Ingress NP queue, VoQ, FIA egress queue) has 3 strict priority: P1, P2 and BE
- Queue 6 (Egress NP queue) has two options: 2PQ+BEs or 3PQ+BEs
- Queue 2 and 6 are user configurable, all others are not
- Queue 3 and 4 priority is determined by queue 2: packet classified at ingress NP queue will be put into same level of priority on queue 3 and 4 automatically

Internal QoS: Back Pressure and VoQ



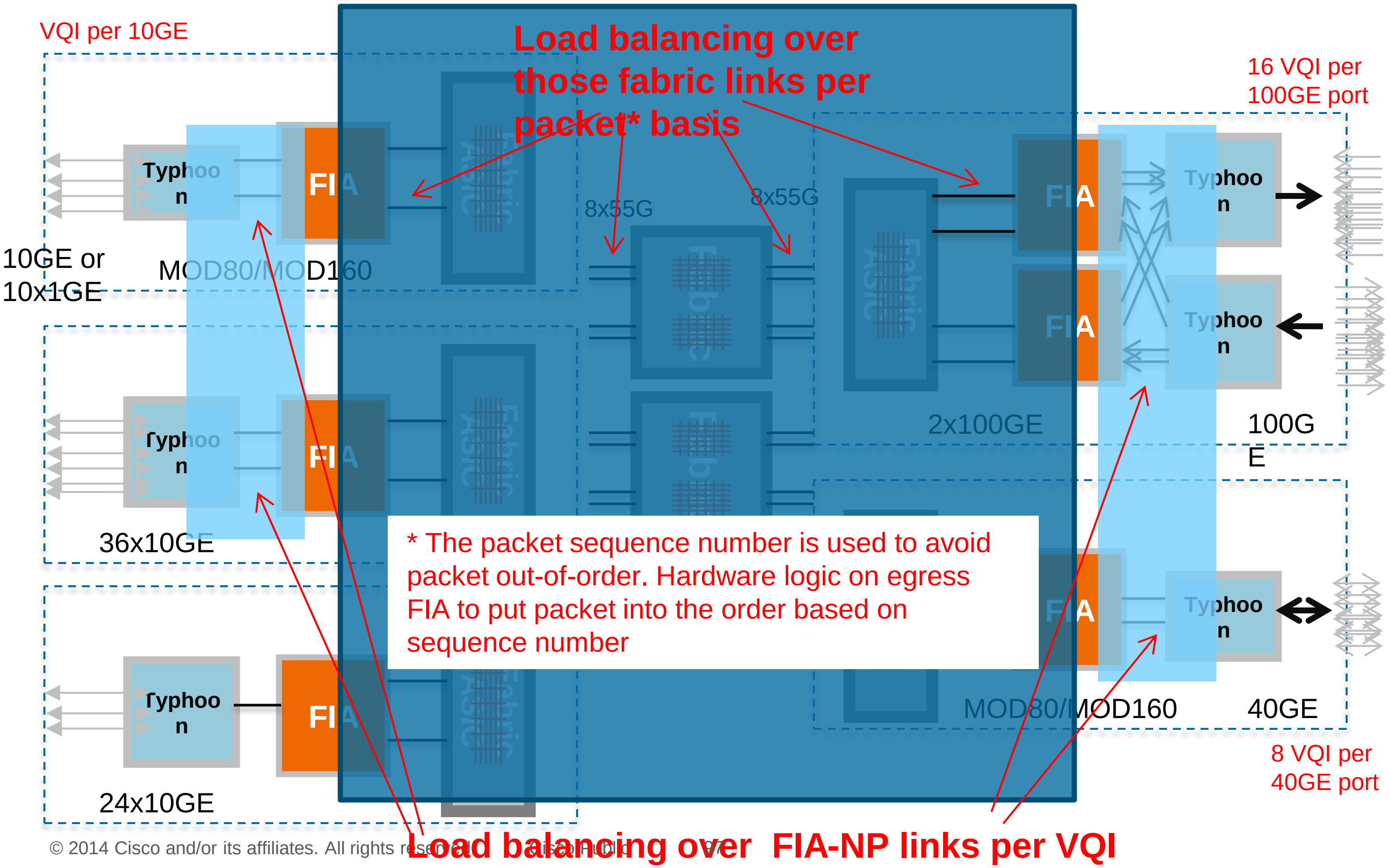
- Egress NP congestion will trigger back pressure to egress FIA. When egress FIA queue cross certain threshold, it will trigger back pressure to switch fabric, then to the ingress FIA: packet put into VoQ
- Queue 3 and 4 are per egress 10G port or per VQI (see next slide)
- Each line card FIA has 1024x4 VoQs, and has 24x4 egress queue
- Each FIA egress queue shape to 13G per VQI. If more than 13G hit, FIA will trigger back pressure
- One port congestion won't head of line block other egress port: purple port won't block green port in the above example, since they go through different VoQs

Understand VQI and Internal Link Bandwidth



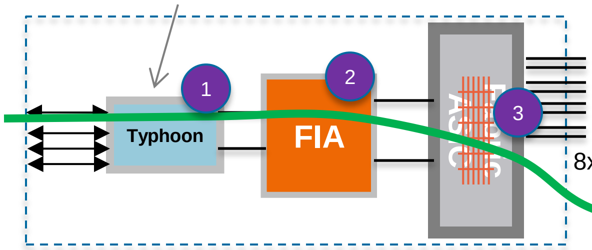
Forwarding “slice” for different LC

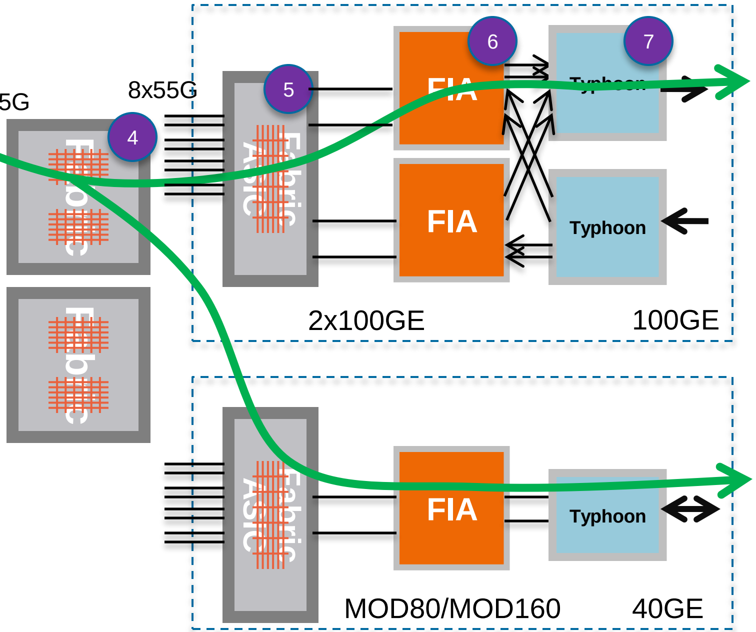
System Load Balancing – Unicast



System Load Balancing – Multicast

Ingress packet FLOW information is used to create 32bits hashing for all kinds of load balancing used in the system





1. NP load balance to FIA (if 2 links) based on the hash
2. FIA load balances over the 2 links to LC fabric using RBH (RBH is from the 32bits hash)
3. LC fabric load balances over multiple links to RSP fabric using modified RBH (RBH % Num_of_active fabric_paths)
4. RSP fabric replicates to egress LC fabric and load balances across 2 links which is selected in step 3
5. Egress LC fabric replicates to selected FIAs and load balances across 2 links to each selected FIAs using MGID
6. FIA replicates to selected NP (if connected to more than 1 NP). FIA load balances across two links to NP using MGID
7. NP replicates over multiple outgoing interfaces and load balance over link bundle member ports

ECMP and Bundle Load balancing

IPv6 uses first 64 bits in 4.0 releases, full 128 in 4.2 releases

ECMP Load balancing

A: IPv4 Unicast or IPv4 to MPLS (3)

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

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

B: IPv4 Multicast

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

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

C: MPLS to MPLS or MPLS to IPv4

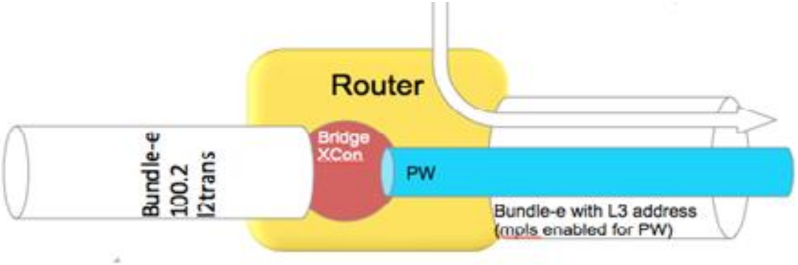
– # of labels <= 4 : same as IPv4 unicast (if inner is IP based, EoMPLS, etherheader will follow: 4th label+RID)

– # of labels > 4 : 4th label and Router ID on Trident card, 5th label and Router ID on Typhoon card

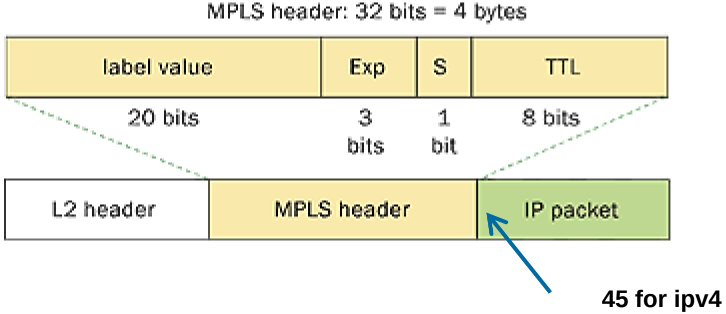
Bundle Load balancing

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

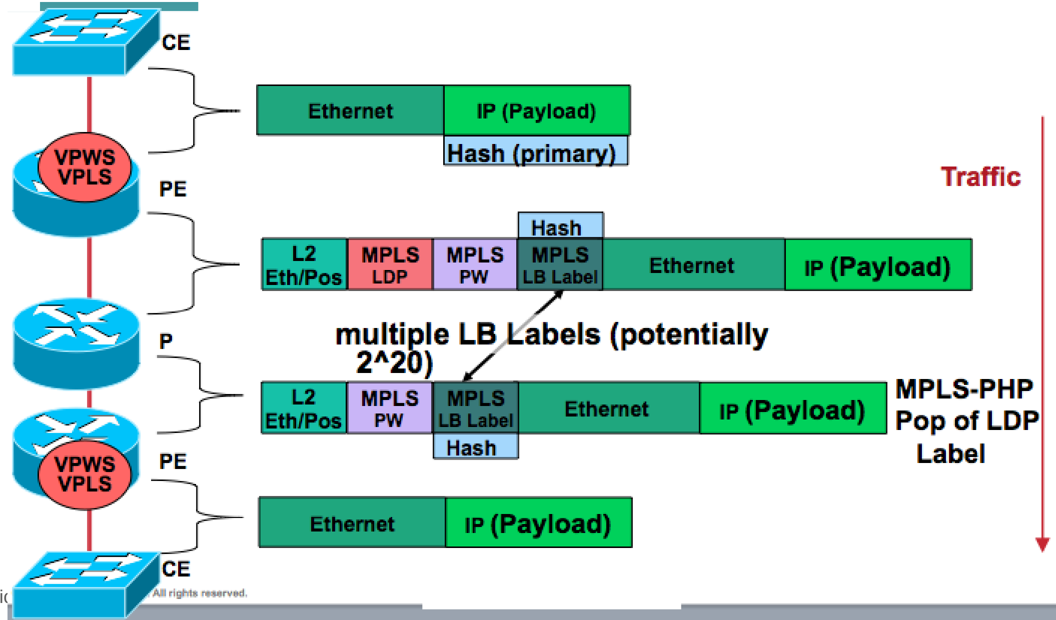
PW Load-balancing scenarios



MPLS/IP protocol stack



EoMPLS protocol stack



MPLS vs IP Based loadbalancing

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

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

GRE Tunnel Load Balancing Logic

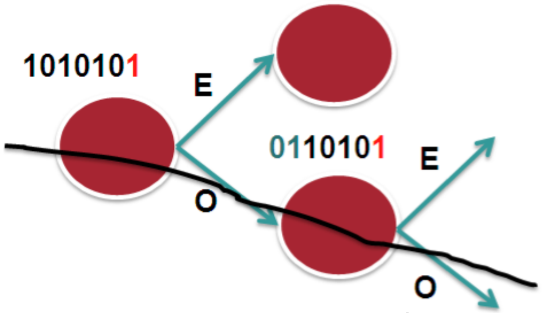
Headend:

- Always uses loadBalancing on inner header with the 5-tuple for IP packet.

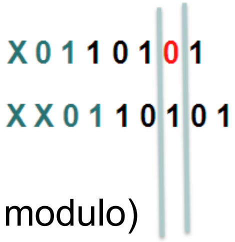
Transit Router:

- GRE+checksum (for IPv4 and IPv6 traffic) – Loadbalancing on inner SIP/DIP.
- GRE + Keepalive (for IPv4 traffic) – Loadbalancing on inner SIP/DIP.
- GRE + Sequence (for IPv4 and IPv6 traffic) – Loadbalancing on outer SIP/DIP.
- GRE + MPLS - Loadbalancing on outer SIP/DIP.
- GRE + Key (for IPv4 and IPv6 traffic) – LoadBalancing on outer SIP/DIP in 431. R510 uses inner SIP/DIP.
- Outer header ipv4 mcast address – Loadbalancing on outer SIP/DIP.

Loadbalancing ECMP vs UCMP and polarization



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



Great references

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

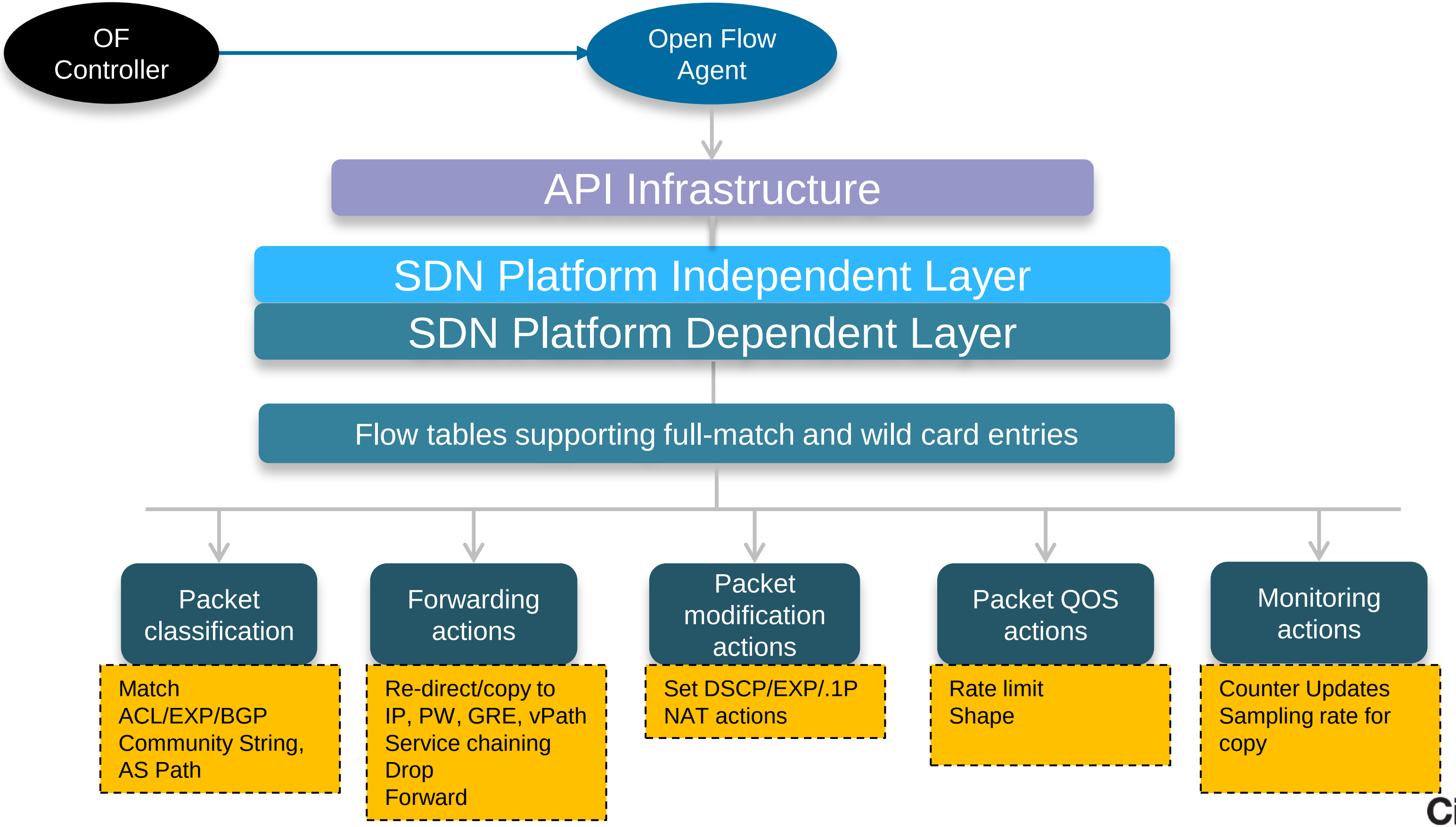
ASR 9000 Advanced System Architecture (1)

OpenFlow

OpenFlow Support on ASR9K

- HW requirement
  - All chassis type (nV cluster support is on roadmap)
  - Typhoon line card only, Trident line card and SIP-700 are not supported
- SW requirement
  - 5.1.1 early trial, 5.1.2 official support
  - Require asr9k-k9sec-px.pie (required for TLS encryption of the OF channel, which is turned on by default)
- Supported interface types
  - Physical interfaces/sub-int such as Gig/10G/40G/100G
  - Bundle interfaces/sub-int
  - Logical interface: BVI, PWHE interface/sub-int
  - Not supported: satellite interface, GRE, TE tunnel
- Hybrid Mode operation
  - OF switch function co-exist with existing ASR9K router functions
  - For example, some sub-interfaces can be part of the OF switch, while other sub-interfaces (on the same port) could be regular L2/L3 sub-interfaces

ASR9K OF/SDN Infrastructures



OpenFlow Configuration Examples

L2 or L2 with PWHE OF switch example:
An L2 only OpenFlow switch is attached to a bridge-domain as follows:

openflow switch 3 pipeline 129
bridge-group SDN-2 bridge-domain OF-2
controller 100.3.0.1 port 6634 max-backoff 8 probe-interval 5 pps 0 burst 0

L3 OF switch, global or vrf example:
L3_V4 switch can be attached either to a VRF or directly to layer 3 interfaces under global VRF. In case of VRF, all the interfaces in that VRF become part of the OpenFlow switch.

openflow switch 1 pipeline 131
vrf of-test
controller 100.3.0.1 port 6634 max-backoff 8 probe-interval 5 pps 0 burst 0

openflow switch 5 pipeline 132
controller 100.3.0.1 port 6633 max-backoff 8 probe-interval 5 pps 0 burst 0
interface GigabitEthernet0/7/0/1.8
interface GigabitEthernet0/7/0/1.9

Show/debug CLI Examples

Openflow show commands

show openflow switch <>
show openflow switch <> controllers
Show openflow switch <> ports
Show openflow switch stats
Show openflow switch flows
Show openflow interface switch <>
show openflow hardware capabilities pipeline <>
show table-cap table-type <>

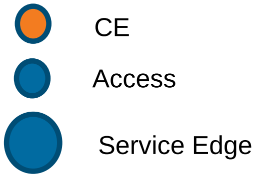
Debug commands for Open flow Agent

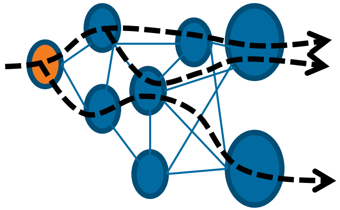
debug openflow switch ovs module ofproto level debug
debug openflow switch ovs module ofproto-plif level debug
debug openflow switch ovs module plif-onep level debug
debug openflow switch ovs module plif-onep-util level debug
debug openflow switch ovs module plif-onep-wt level debug

ASR 9000 Advanced System Architecture (2)

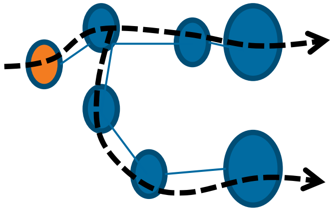
nV (network virtualization) Satellite and Cluster

What’s the story behind the nV?

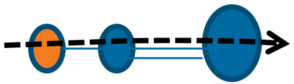




Example 1:
Complex, mesh network topologies, multiple paths, need network protocols



Example 2:
Ring topology, traffic direction: East or West, do I still need those network protocols?

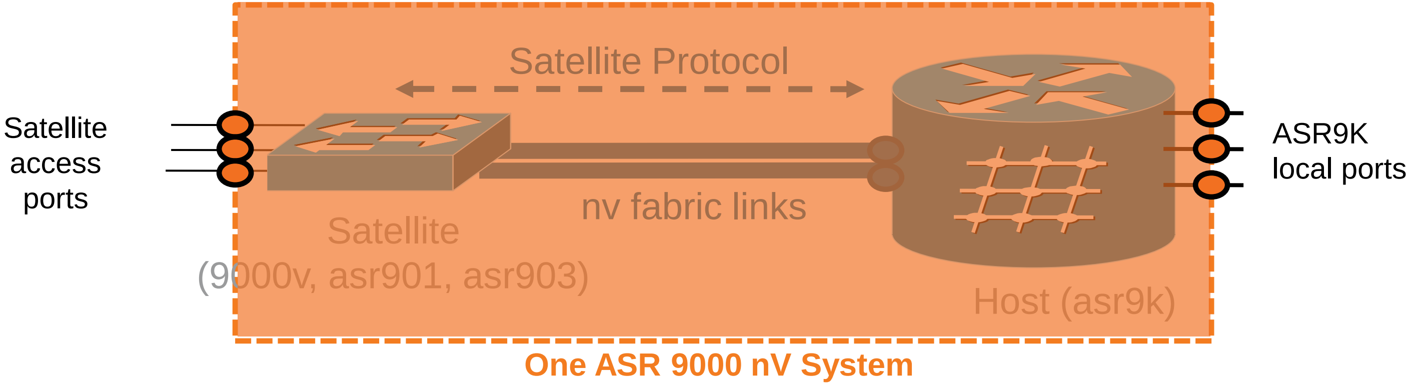


Example 3:
Even a simpler case: P2P topology. Why it need to run any protocol on the access device? Why it even need any forwarding table like FIB or MAC?

Satellite is network virtualization solution
which can dramatically simplify network for certain network topologies and traffic patterns

ASR 9000 nV Satellite Overview

Zero Touch, Fully Secure



- Satellite and ASR 9000 Host run **satellite protocol** for auto-discovery, provisioning and management
- Satellite and Host could be co-located or in different location. There is **no distance limitation** between satellite and Host
- The connection between satellite and host is called “**nv fabric link**”, which could be L1 or over L2 virtual circuit (future)

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 -48vDC Power Feeds
- Single AC power feed
- Max Power 210W
- Nominal Power 159 W

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 fiber SFP optics**
- **Speed/duplex auto negotiation**

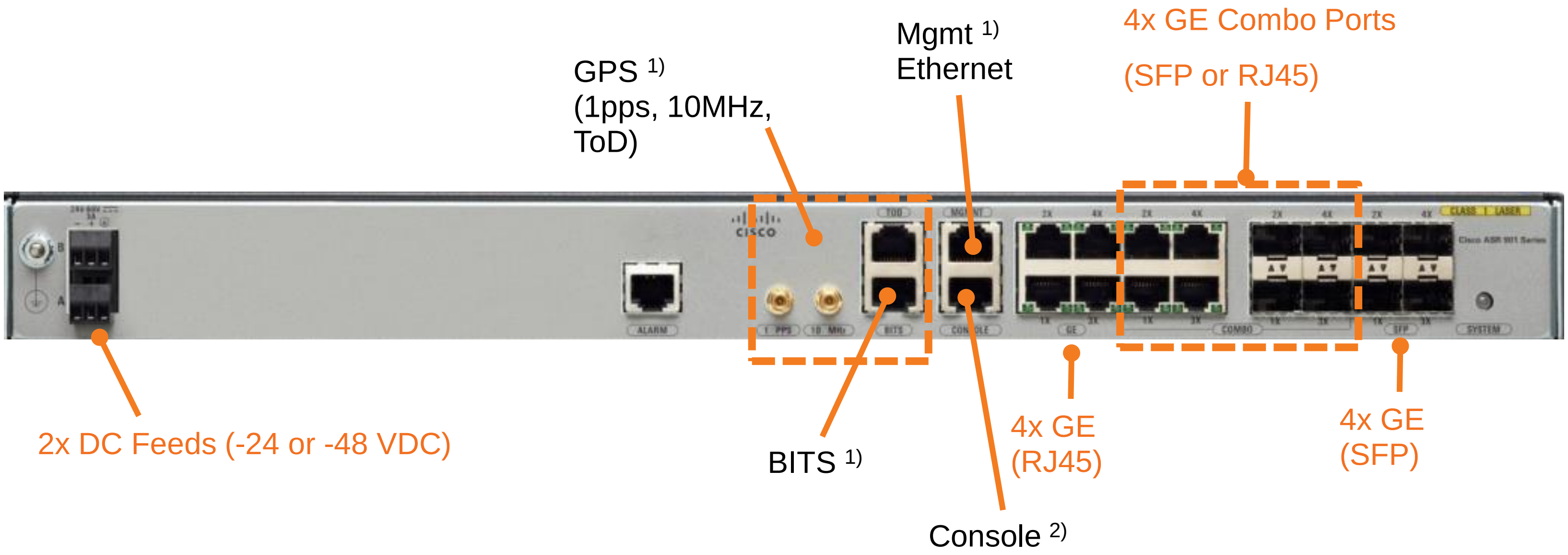
4x10G SFP+

- Initially used as Fabric Ports ONLY (could be used as access port in the future)
- **Copper and fiber 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

Router 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

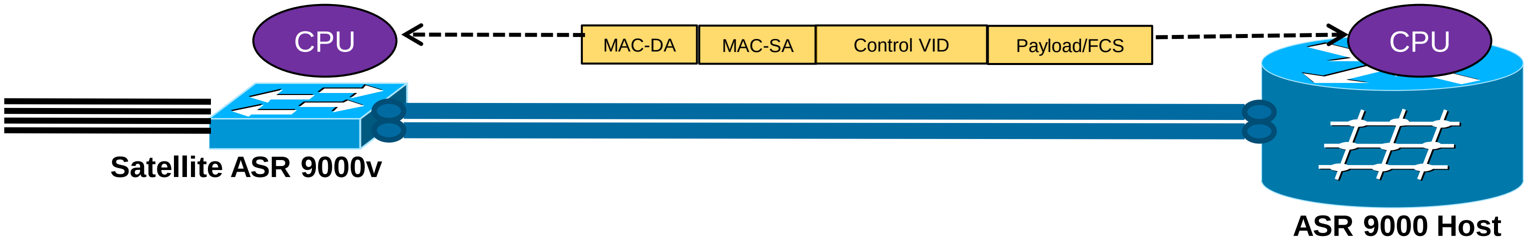
2x Power Modules

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

Fan Module

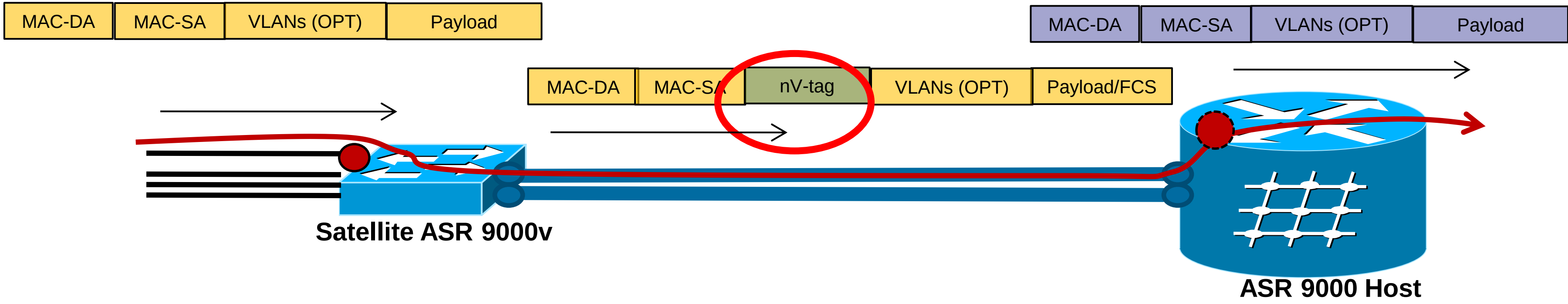
Satellite – Host Control Plane

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 standardized in the future
  - **Get/Set style messages** to provision the satellites and also to retrieve notifications from the satellite

Satellite – Host Data Plane Encapsulation



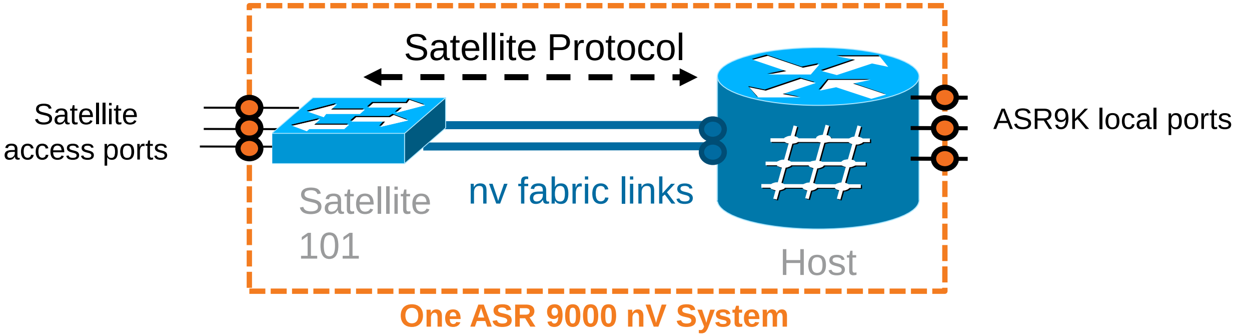
On the Satellite

- 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

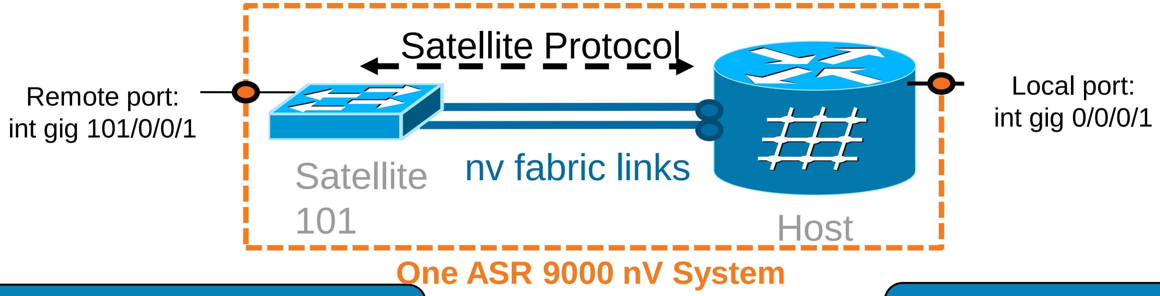
Initial Satellite Configuration




```
nv
 satellite 101 ← define satellite
  type asr9000v


interface TenGigE 0/2/0/2 ← configure satellite fabric port
  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



Satellite access port configuration examples

```
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
```

Local port configuration examples

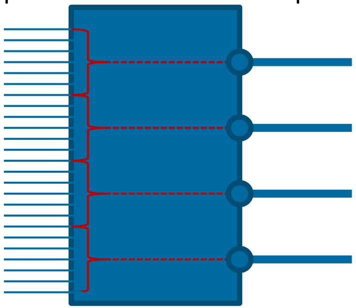
```
interface GigabitEthernet 0/0/0/1
  ipv4 address 2.2.2.2 255.255.255.0

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

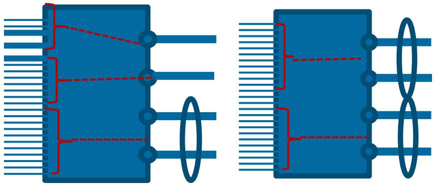
ASR9000v Example

44x1GE
Access ports

4x10GE
Fabric ports



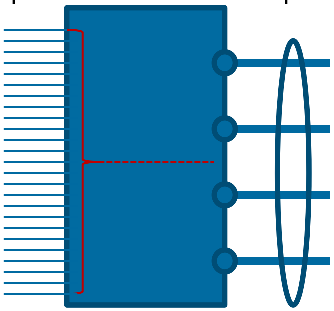
Mode 1: Static pinning
No fabric port redundancy



It can mix model 1 and
2 on the same satellite

44x1GE
Access ports

4x10GE
Fabric ports



Mode 2: Fabric bundle
Fabric port redundancy

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

- 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, Cisco 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 Host

```
RP/0/RSP0/CPU0:ios#show install active

  Node 0/RSP0/CPU0 [RP] [SDR: Owner]
    Boot Device: disk0:
    Boot Image: /disk0/asr9k-os-mbi-4.3.0/0x100000/mbiasr9k-rp.vm
    Active Packages:
      disk0:asr9k-mini-px-4.3.0
      disk0:asr9k-mpls-px-4.3.0
      disk0:asr9k-9000v-nV-px-4.3.0
      disk0:asr9k-asr901-nV-px-4.3.0
      disk0:asr9k-asr903-nV-px-4.3.0
      disk0:asr9k-fpd-px-4.3.0

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

9000v: Configure, Install and Ready-to-Go

Initial satellite configuration (on ASR9K Host → Rack & Plug → Go

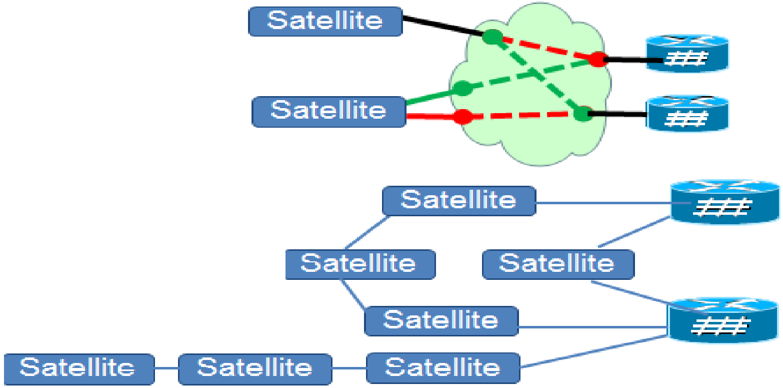
- Critical Error LED ON → bad hardware, RMA
- Major Error LED ON → Unable to connect to ASR9K host
  - Missing the initial satellite configuration?
  - L1 issue, at least one of the uplink port light green?
  - Security check (optional), is the satellite SN# correct?
- Status light green → ready to go, satellite is fully managed by Host

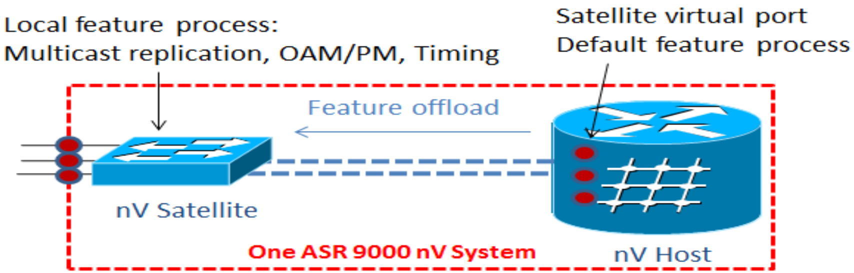
nV Satellite Evolution

Topology expansion*

High Dense
10G Satellite**

Feature offload***



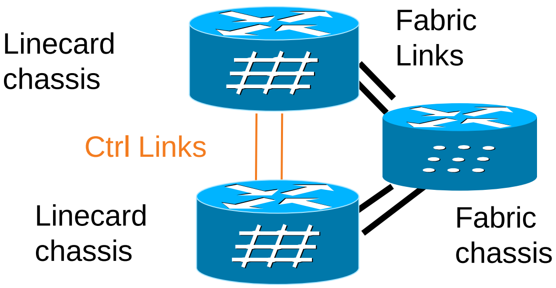


* Ring, L2 fabric, dual-hosts supported in 5.1.1 release

** high dense 10G satellite on the roadmap

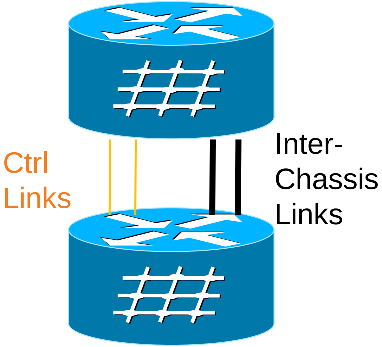
*** QoS offload in 5.1.1, SyncE offload in 5.2.0, Multicast offload in 5.2.2, others are on roadmap

ASR9000 nV Edge Overview



CRS Multi-Chassis

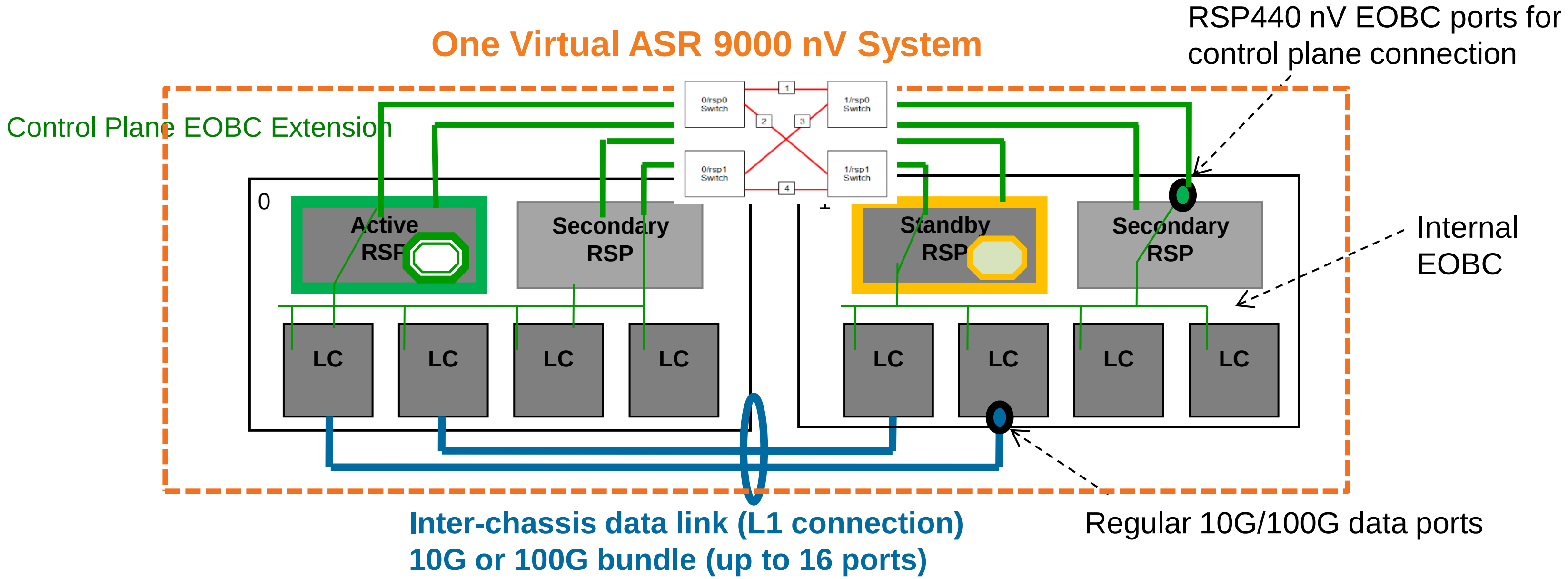
Leverage existing IOS-XR CRS multi-chassis SW infrastructure Simplified/Enhanced for ASR 9000 nV Edge



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



- 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
```

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
...
```

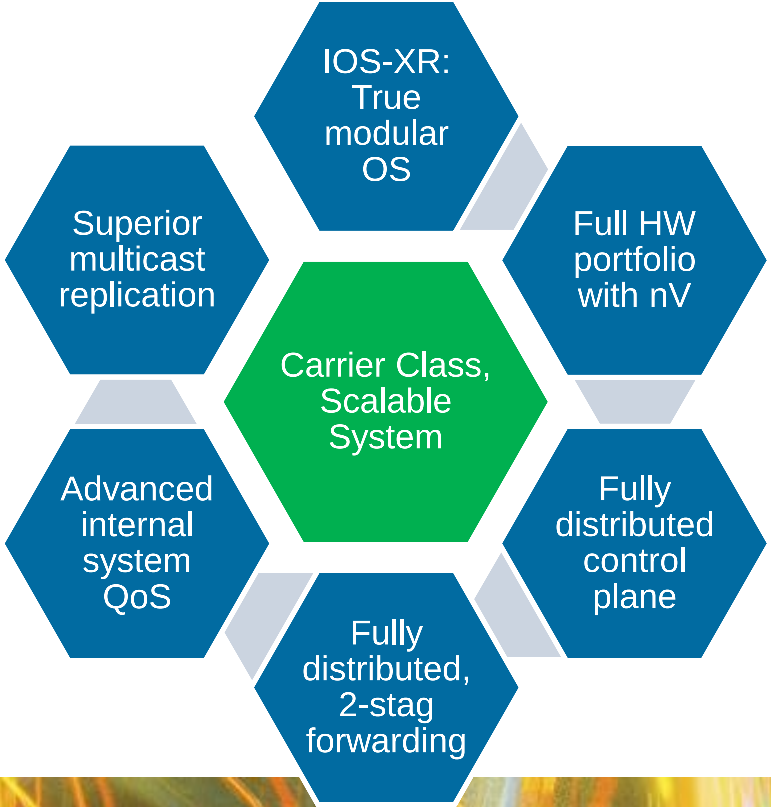
- Interfaces on a Satellite connected to the nV Edge Virtual System

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


nVSSU (nV System Software Upgrade)

- Existing nV cluster image upgrade: require reloading of both of the racks in the nV system
- nVSSU: a method of minimizing traffic downtime while upgrading a cluster system
  - Support “Any-to-Any” Release upgrade
  - Rack-by-Rack fully reload, so fully support XR Architecture releases, FPD upgrade, and Kernel upgrade
  - Traffic Outage estimated* < 1 sec. Topology loss < 5 min.
  - Traffic protection is via network switching
- Upgrade Orchestration is performed off-router via a set of Python scripts
- Feature roadmap:
  - Limited support in IOS-XR 5.2.2 release. Generic support will be in later release

* May subject to change depends on the scale and feature set



nV, XRv, OF, VXLAN and a lot more …

References

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

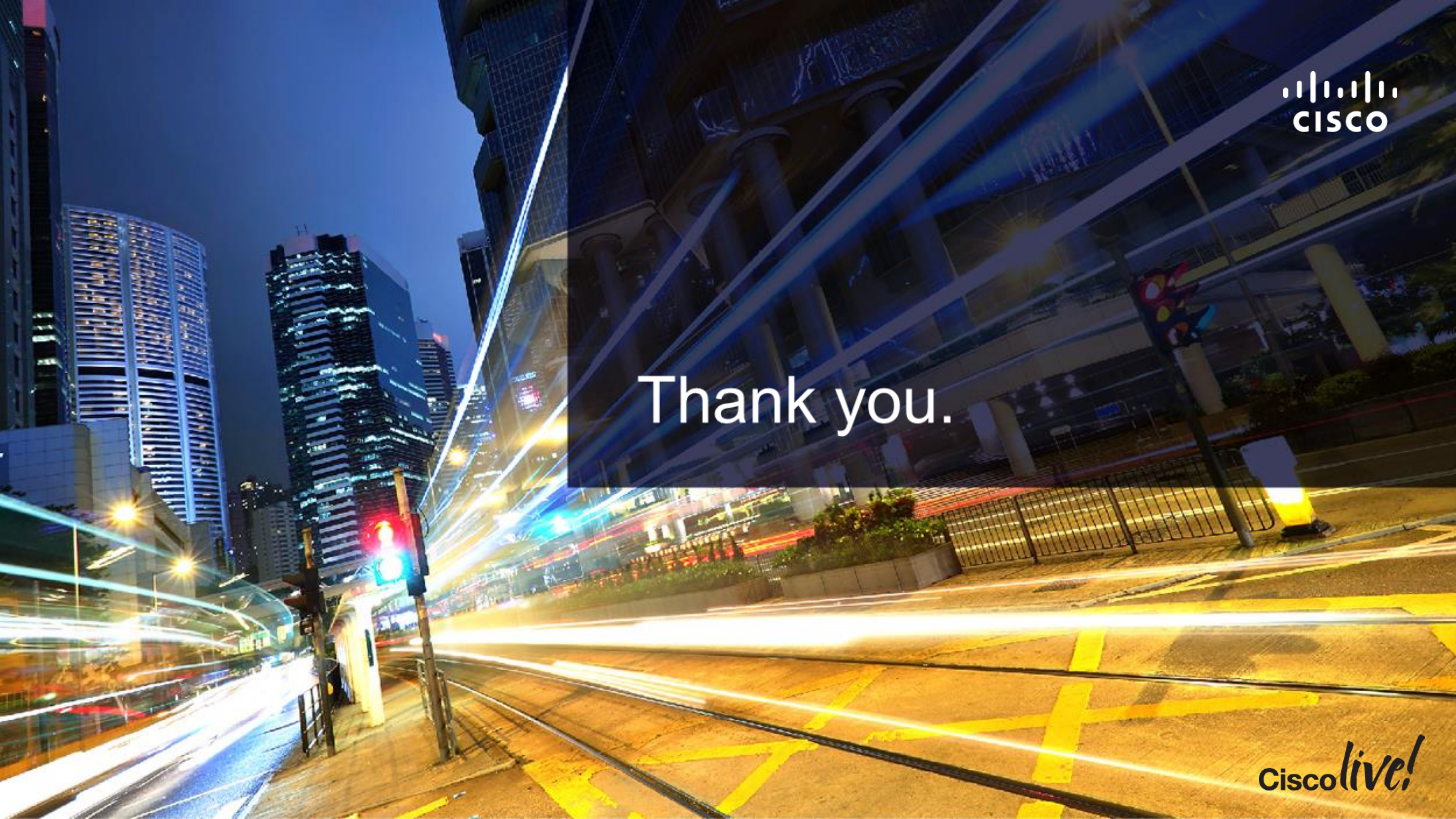
References

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

Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Complete your session evaluation through the Cisco Live mobile app or visit one of the interactive kiosks located throughout the convention center.

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


Thank you.