Cisco live!
February 20 - 24, 2017 · Berlin

CISCO

Your Time Is Now

Cisco IOS XE Routers ASR 1000 & ISR 4000 The Evolution of Converged Network Edge Architectures

David Roten, Technical Marketing Engineer
Marcelo Magno, Lead Product Manager

TECSPG-2401

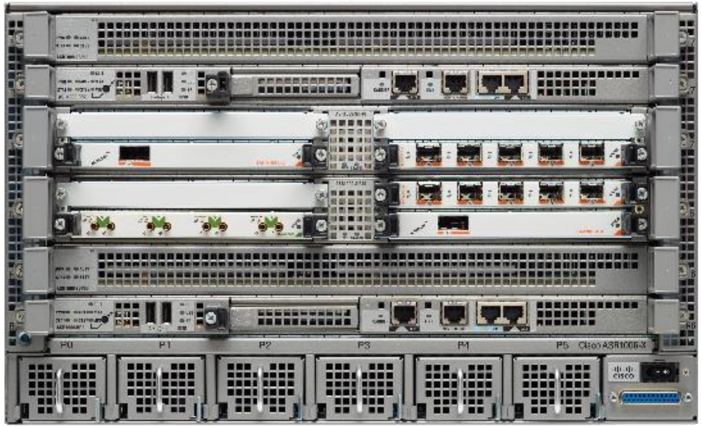
TECSPG-2401 Agenda

- Introduction, What’s new about IOS XE
- Software Architecture
  - ASR1000 / CSR1000V / ISR4000
- DRAM Demystified
- High Availability on ASR1000
- QoS, similarities and differences across platforms
- Performance
- Configuration Specifics

Introducing IOS XE

With ASR1000 and ISR4000 series routers

- 2007
  - ASR1000 introduced as the first routing platform using IOS XE software
- 2013
  - ISR4000 series routers inherit the new IOS architecture and married with the previous innovations from the ISR G2 series of routers
- 2014
  - 5 new ISR4000 series routers introduced to extend coverage through all branch connectivity needs
- 2015
  - 3 new ASR1000 chassis along with new MIP and Interfaces blades
- 2016
  - Two new fixed ASR1000-HX chassis introduced along with ISR4221



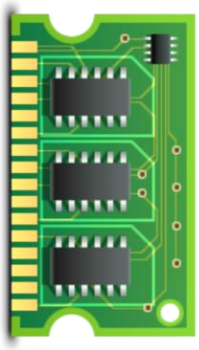
What is IOS XE

How is it different than Classic IOS at 30,000 feet?

- BIG differences!
- Linux is the underlying operating system for the chassis
- IOSd runs as a process in Linux
- Benefit from protected memory and process isolation
- Very familiar CLI (some things are best kept the same)
- Separation of control and data planes into discrete processes
- Multicore support for data plane
- Introduction of services plane in addition to control and data plane

Linux? where?

I don’t see a shell prompt anywhere!



- Linux, yes, but the only interface with the system is via IOSd
- IOSd presents the same CLI interface that everyone loves from other platforms like 7200, 7600, and ISR G2 routers
- Because IOSd is running as a discrete process it has protected memory that is isolated from crashes in other processes and failures in other components in the system.
- Individual software component upgrade opportunity
- With “`service internal`” and “`request platform software system shell`” commands you can find Linux. Don’t do it without a good reason. Here be dragons and you taste good with ketchup. Requires one-day license from TAC since you go well with ketchup.

Same CLI

Not like IOS-XR that looks like you understand it until you don’t…

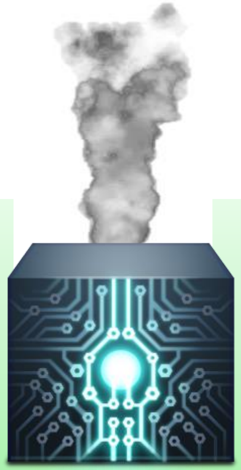
- In general configurations from Classic IOS platforms move forward to IOS XE without any changes
- There are certain features like QoS, carrier grade NAT (CGN), WAAS, CME that when moved forward are going to have slight variations or need updating to take advantage of new features
- More details on some of these later
- Cisco Active Advisor can analyze configurations from Classic IOS platforms and provide updated configurations for IOS XE platforms

  https://ciscoactiveadvisor.com/

Divided we stand, united we fall!

Wait, isn't it supposed to be the other way around?

- Classic IOS is a single threaded monolithic blob of code that has served us well for a long time
  - Impossible to separate control and data plane
  - Processors aren't getting faster so much, their number of cores are growing though



- Multi-core lets us to dedicate certain cores for control plane and others for data plane, i.e. no starving data plane for control plane

- Furthermore, we can use one chip architecture for control and a separate for data plane for mix and match to meet needs
- We have even created a services plane that can run alongside IOSd and not impact platform performance

Three legged stool

Balances a whole lot better than a two legged stool

- Previous router platforms had only the concept of a control plane and data plane

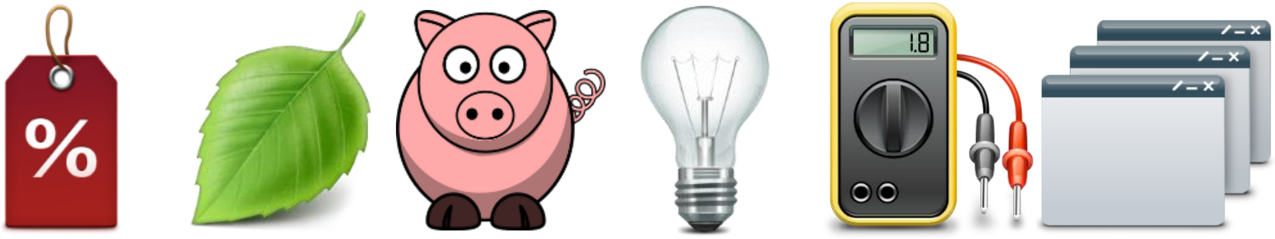
- IOS XE introduced the service-plane which allows for rich appliance type functions to be provided in the same sheet metal chassis as WAN edge functionality

- Consolidates equipment ownership, service contract management / expense, power, cooling, and space requirements

Services plane

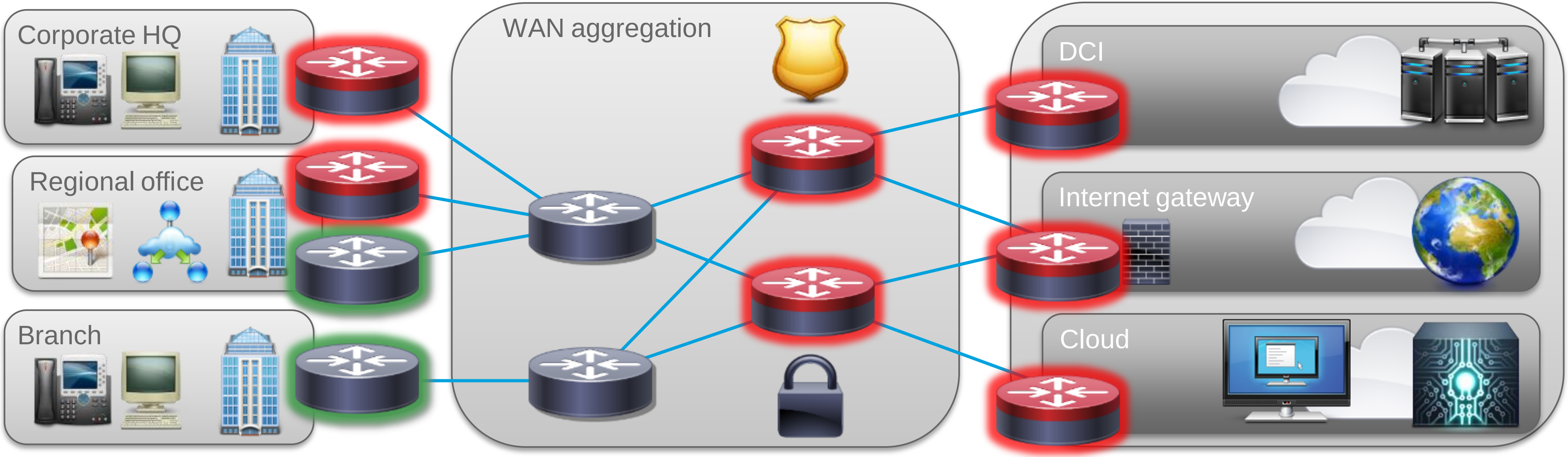
Coffee, tea, soft drinks, peanuts, watch your elbows please…

- All platforms have multiple cores on the control plane, no truck rolls needed!
- IOSd consumes one core with occasional use for extra cores for specific features
- Remaining cores are given to a hypervisor which can run dedicated applications to provide appliance like services
  - vWAAS
  - EnergyWise
  - SNORT
  - WireShark



- Single memory pool is used for Linux, IOSd, and the services plane

IOS XE in enterprise next generation networks

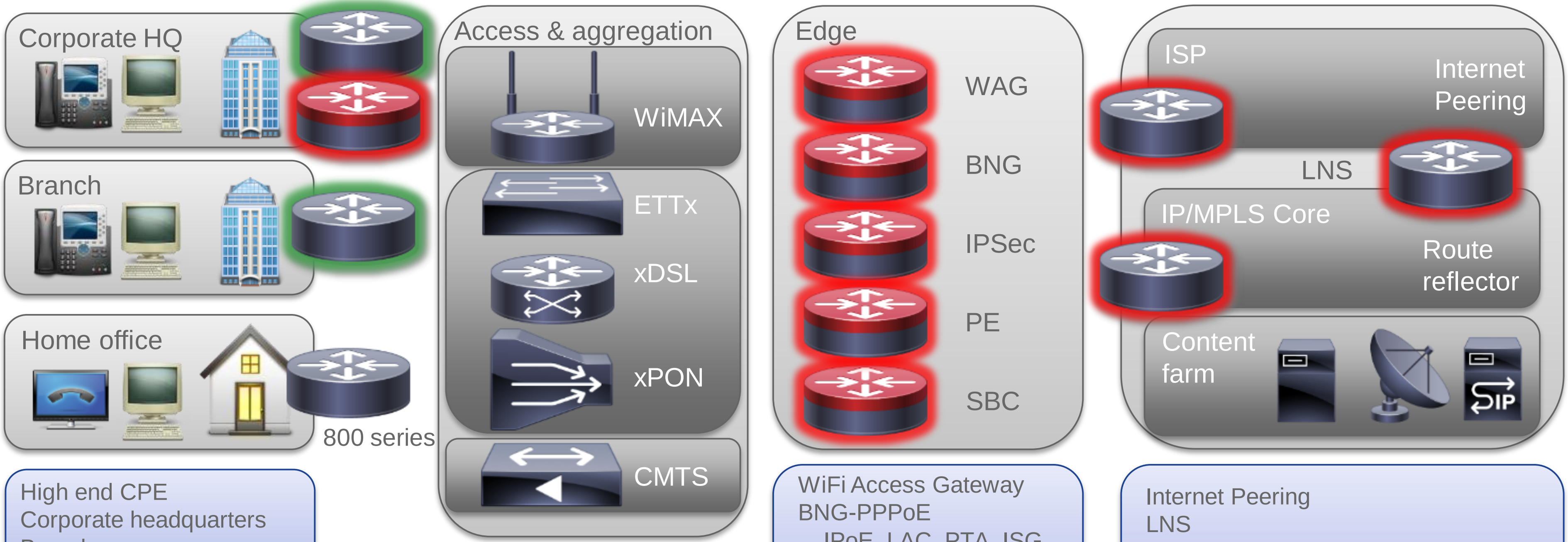


High Speed CPE
High-end Branch
Regional office
Corporate headquarters

WAN Aggregation
DMVPN / GETVPN / FlexVPN for access
IPSec

Data Center Interconnect
Internet gateway
Zone-Based Firewall
Cloud Services Edge

IOS XE in service provider NextGen networks

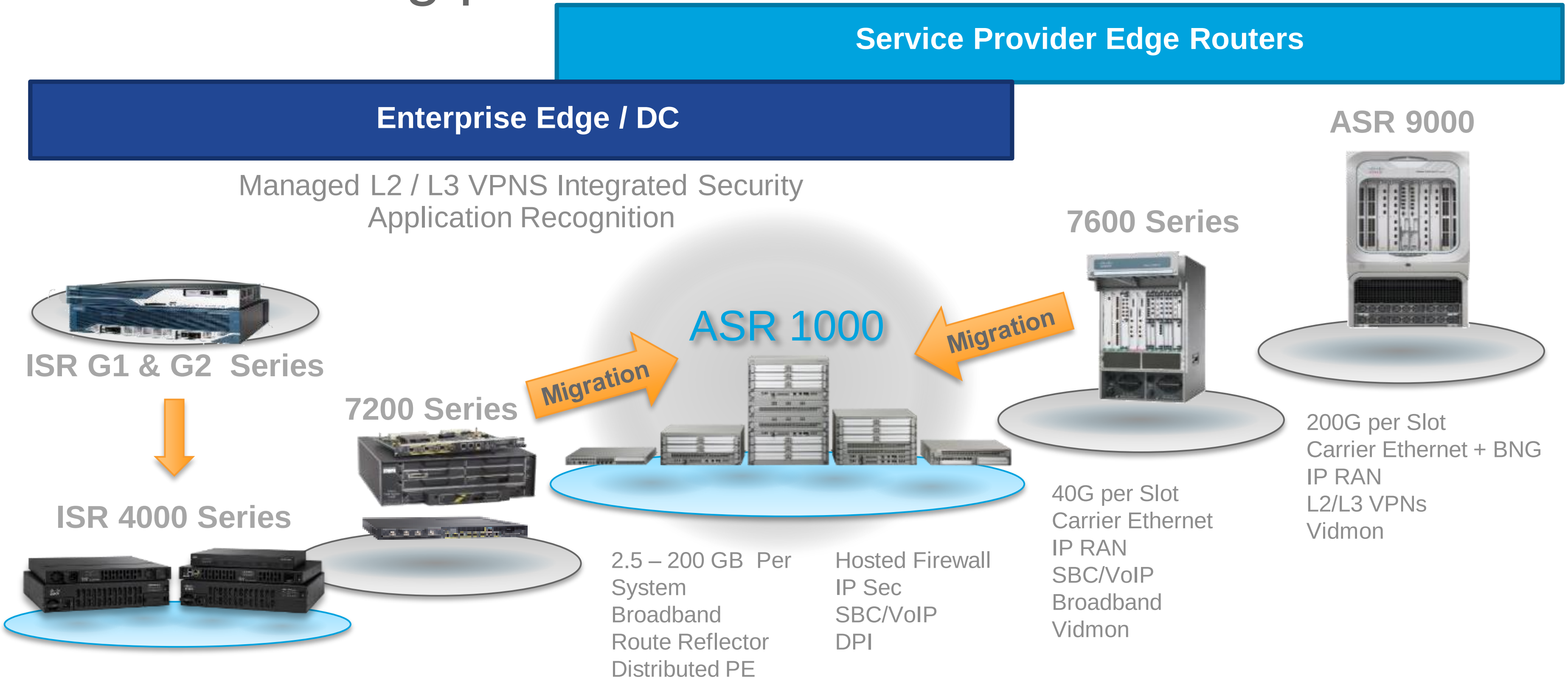


High end CPE
Corporate headquarters
Branch
Home office

WiFi Access Gateway
BNG-PPPoE
IPoE, LAC, PTA, ISG
IPSec Aggregator
PE (L3VPN and L3VPN)
VoIP SBC

Internet Peering
LNS
Route Reflector

Cisco’s routing portfolio



Service Provider Edge Routers

Enterprise Edge / DC

Managed L2 / L3 VPNS Integrated Security
Application Recognition

ISR G1 & G2 Series

ISR 4000 Series

7200 Series

Migration

ASR 1000

Migration

7600 Series

ASR 9000

2.5 – 200 GB Per System
Broadband
Route Reflector
Distributed PE

Hosted Firewall
IP Sec
SBC/VoIP
DPI

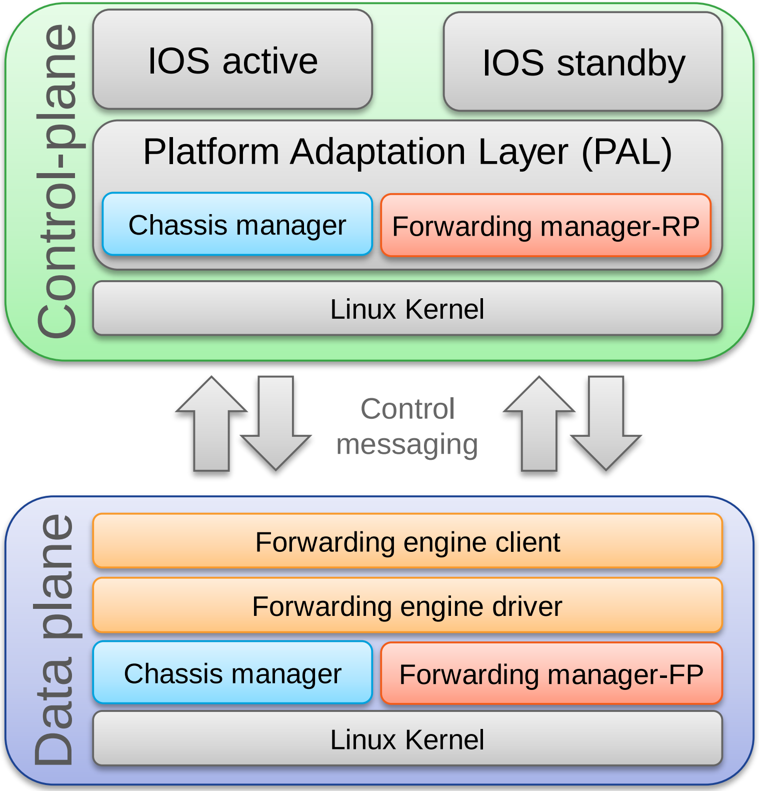
40G per Slot
Carrier Ethernet
IP RAN
SBC/VoIP
Broadband
Vidmon

200G per Slot
Carrier Ethernet + BNG
IP RAN
L2/L3 VPNs
Vidmon

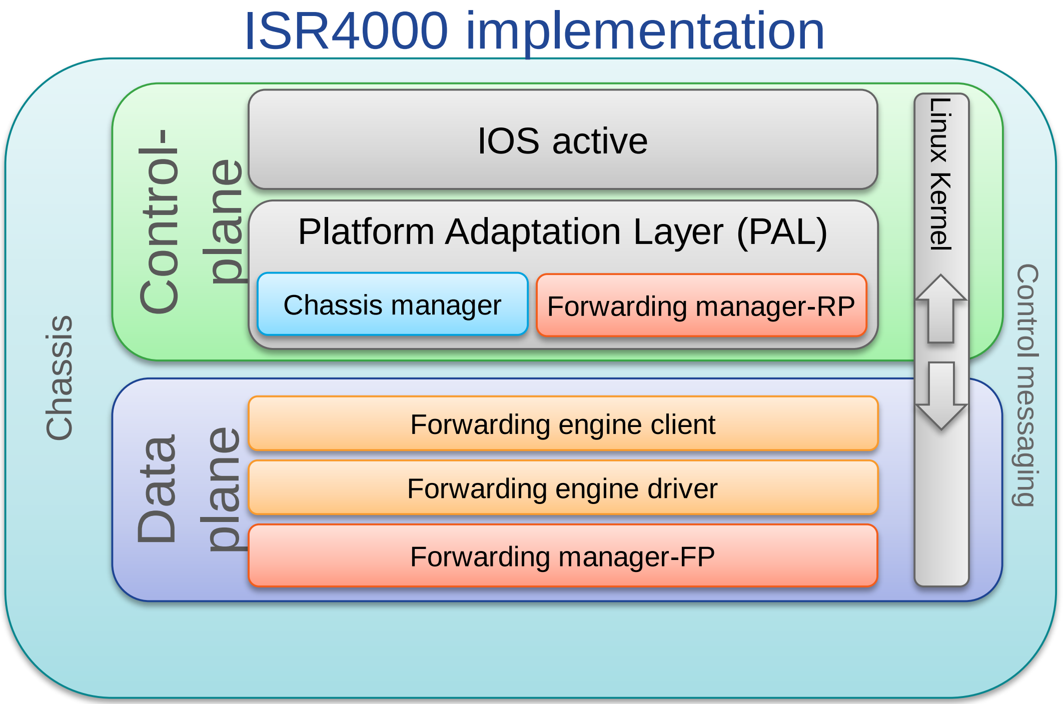
Software Architecture

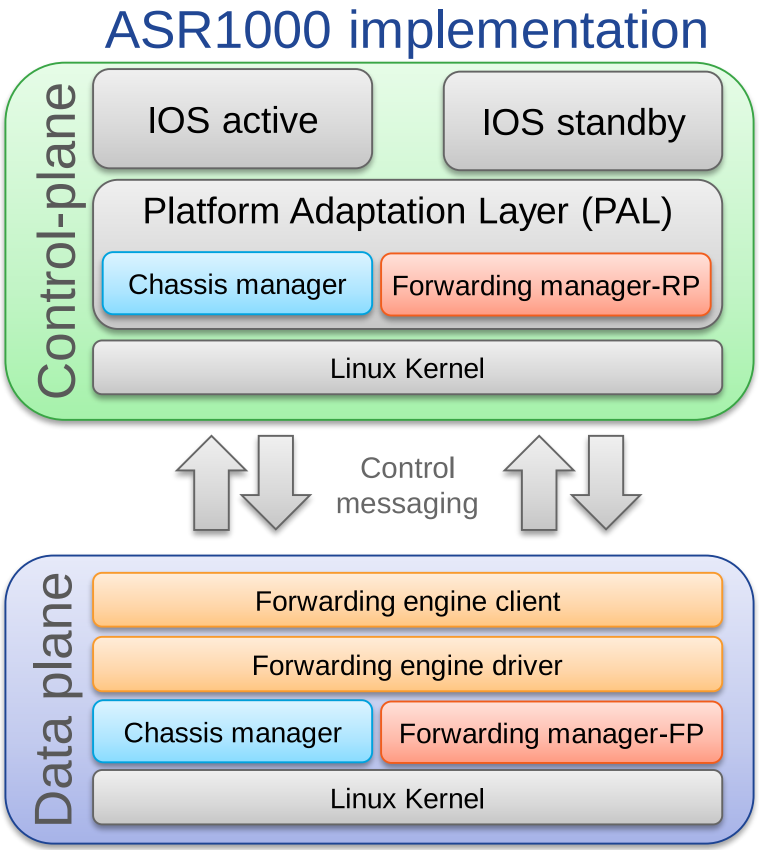
IOS XE software architecture

- IOS + IOS XE Middleware + Platform Software
- IOS runs as its own Linux process for control plane
- Linux kernel with multiple processes running in protected memory
- Fault containment, re-startability
- ISSU of individual SW packages
- With redundant data plane hardware packet loss can be as low as 50 ms at failover



IOS XE software architecture



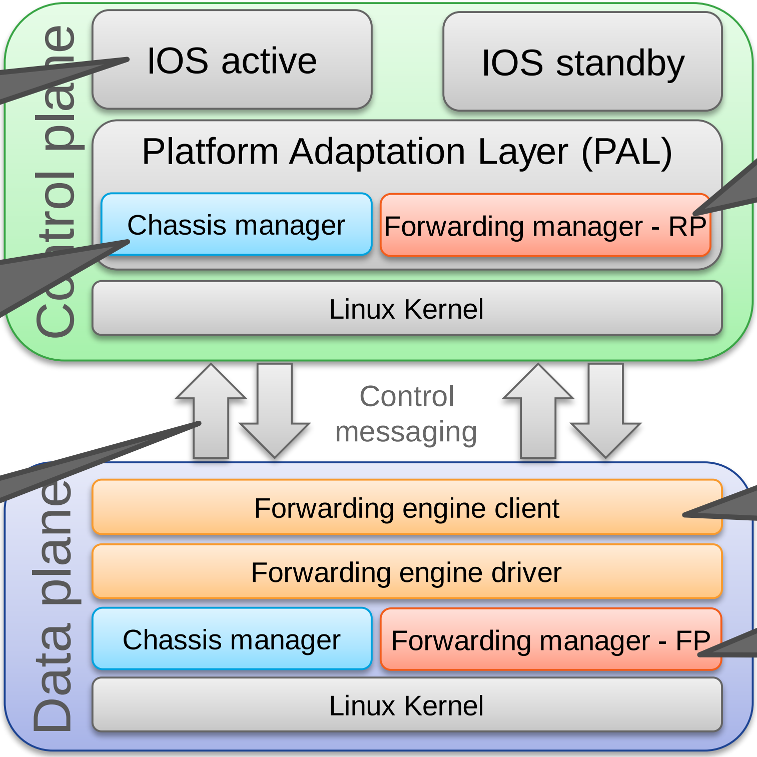


IOS XE architecture building blocks

- Runs Control Plane
- Generates configurations
- Maintains routing tables (RIB, FIB…)

- Initialization of RP processes
- Initialization of installed cards
- Detects and manages OIR of cards
- Manages system status, environmentals, power, EOBC

- All messaging done via IP datagrams in the kernel or over the backplane of the chassis



Control plane

IOS active

IOS standby

Platform Adaptation Layer (PAL)

Chassis manager

Forwarding manager - RP

Linux Kernel

Control messaging

Data plane

Forwarding engine client

Forwarding engine driver

Chassis manager

Forwarding manager - FP

Linux Kernel

- Provides abstraction layer between hardware & IOS
- Manages ESP redundancy
- Maintains copy of FIB and interface list
- Communicates FIB status to active & standby data plane FM

- Maintains copy of FIBs
- Programs forwarding plane and forwarding engine DRAM
- Statistics collection & RP communication

- Communicates with forwarding manager in control plane
- Provides interface to QFP client & driver

IOS XE Linux kernel

- Control CPUs run a Linux operating system kernel
  - handles process scheduling, memory management, interrupts
  - modular ASR1000 routers run multiple instances of Linux
  - fixed ASR1000s and ISR4000s run a single instance of Linux
- Includes a suite of low-level applications
  - Linux console access for debugging
  - base software is common, but may vary slightly on different platforms
- Connectivity to other system components via IPC
  - device drivers for EOBC, Hypertransport, PCIe
  - kernel is responsible for directing IPC messages to the respective software processes (IOS, chassis manager, etc.)
  - Implements punt-path for legacy data packets
- Pre-emptible (can interrupt and prioritize processes)

IOS XE IOSd

- IOSd runs as a process
  - IOS timing is governed by Linux kernel scheduling
- Provides virtualized management ports
  - no direct hardware component access
- Handles all control plane features
  - CLI, configuration processing, SNMP handling
  - running routing protocols & computing routes
  - session management
- Processes punted packets (legacy protocols, all protocols communications)
- Two IOS processes can run in parallel for software redundancy on non-redundant chassis
- Based on IOS 12.2SR features (includes 12.2SB and 12.4T-based features)

For your reference

IOS XE features

Routing & MPLS & L2 (IPv4 / IPv6)	IPv4 / IPv6 routing	CRoMPLS	BGP PIC Core	BGP PE-CE Opt.
	BGP, RIP, IS-IS, OSPF	EoMPLS	IPv4 selective Download	mVPN
	static routes	PW redundancy	Ethernet, POS, ATM	Half-duplex VRF
	GRE	MLPPP	GLBP, HSRP, VRRP	BGP Pic Best External
	MPLS LDP	GEC	IP event dampening	IPv4 over IPv6 Tunnels
	MPLS VPN	PBR	BFD for IS-IS, OSPF, Static (IPv4 & IPv6)	PfR
	Inter-AS & CsC	Netflow (v5, v8, v9)		L2TPv3
	MPLSoGRE	BGP policy accounting	WCCP	VxLAN
	MPLS TE FRR	BGP NSF	8000 eBGP/iBGP	EVPN
	VRF-aware features	BGP 4-byte AS (DOT)	4000 VRF	Segment Routing
Broadband	LAC, LNS, PTA	per-user Firewall	PPPoE Relay	ISG VRF-transfer
	L2TP	ANCP	ISG postpaid, tariff switching	VPND Multihop
	32K sessions with HA& QoS	dynamic Policies	ISG flow control	L2TP forwarding of PPP Tags
	AAA support	BNG clustering	VRF-aware ISG	PPPoEoA
	DHCP Relay for IPv4 & IPv6	template ACL	ISG-SCE control bus	IPv6 Broadband
	remote access MPLS	LI: Radius & SNMP	RADIUS COA / PoD	ANCP on ATM
	per-session QoS	PPPoE Tag support (RID, CID)	ISG single sign-on	
Multicast	PIM	IPv6 BSR	Multicast NAT	IGMPv2/v3
	PIM BiDir	MVPN	Multicast CAC	Extended ACL for Multicast
	IPv6 Multicast Routing	MVPN Extranet	MVPN NSF/SSO	
QoS	HQF support	dual/single rate 3 color policing	bandwidth remaining ratio	ATM service policies (VP/VC)
	2PQs, 128K queues	16K policy-maps	policy aggregation	NBAR
	MQC: classification / marking	1000 class-maps	ATM shaping per VP/VC	FPM
	egress queuing	3-level hierarchical scheduling	egress classification on QoS group	

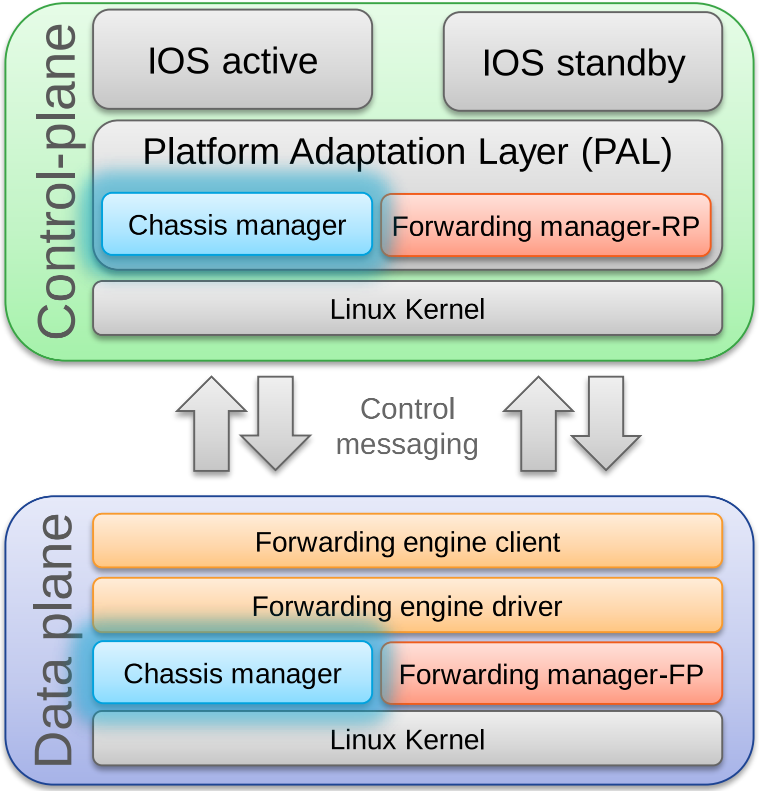
For your reference

IOS XE features

Security	hardware assisted IPSec IPSec VPN 3DES/AES DMVPN GETVPN	Zone-based Firewall NAT RTSP Firewall ALG Control Plane Policing	FIPS compliance IPv6 IPSec static VI VRF-aware zone-based Firewall VRF-aware NAT	DMVPN Hierarchical Hub VRF-aware IPSec VRF-aware Zone-based FW
SBC	Distributed and Integrated SBC Topology Identity hiding DoS Protection Pinhole/filter control SIP Signaling/latching	NAPT Megaco/H.248 Flow-based QoS control DBE control interface H.248, V4 transport, UDP, TCP, etc	Twice NAT for IPv4 No NAT for IPv6 H.248 ACK 3-way H.248 interim accounting SIP-H.323, H.323-H.323	Flexible header manipulation Privacy Header Signaling congestion control IPv6 support SBC Endpoint switching
HA	Config sync SNMP, ARP, NAT Stateful IS-IS	IPv6 FR, PPP, MLPPP, HDLC, VLAN DHCPv4/v6	IPSec MPLS, MPLS-VPN, LDP, VRF-lite	
Network management	LAN Management Solution Cisco Information Center QoS Policy Manager IP Solution Center	MPLS Diagnostics Expert Netflow Collector Cisco Security Manager Cisco Multicast Manager	Traffic Engineering Manger MPLS LSP Ping / Traceroute MIBs SNMP	Syslog VRF-aware NF Netconf/YANG Orchestration

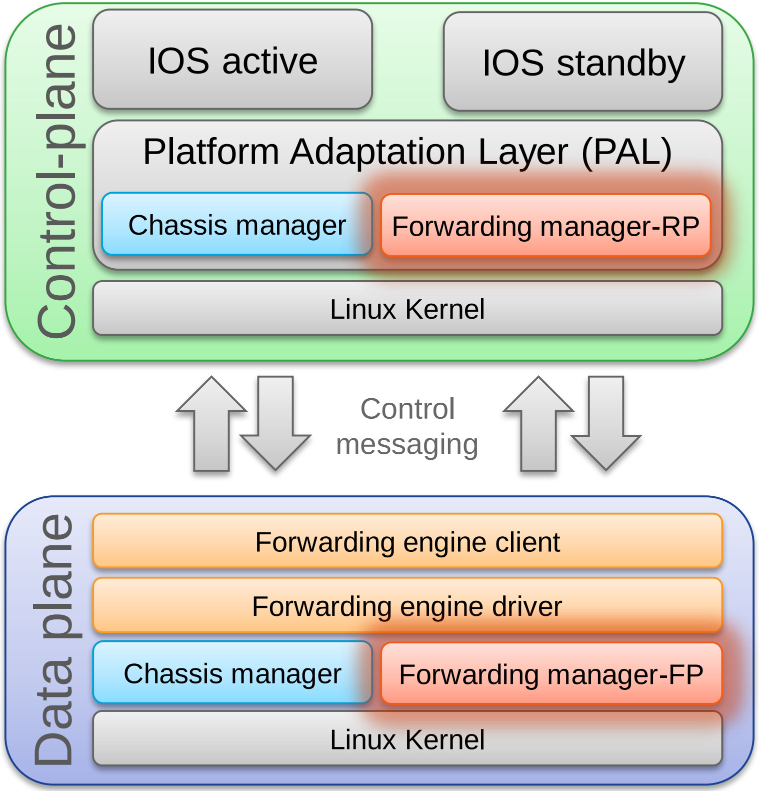
IOS XE chassis manager

- Initializes hardware and boots other processes
    - Manages EOBC switch on RP in ASR1000
    - Manages ESI links on RP/ESP/SIP in ASR1000
    - Manages timing circuitry
    - Controls reset, power-down of modules
    - Selects active/standby hardware, initiates failover
- Detects OIR
    - starts image download and boot process of the respective hardware component
- Communicates with IOS to make it aware of the hardware components
- Monitors environmental variables and alarms



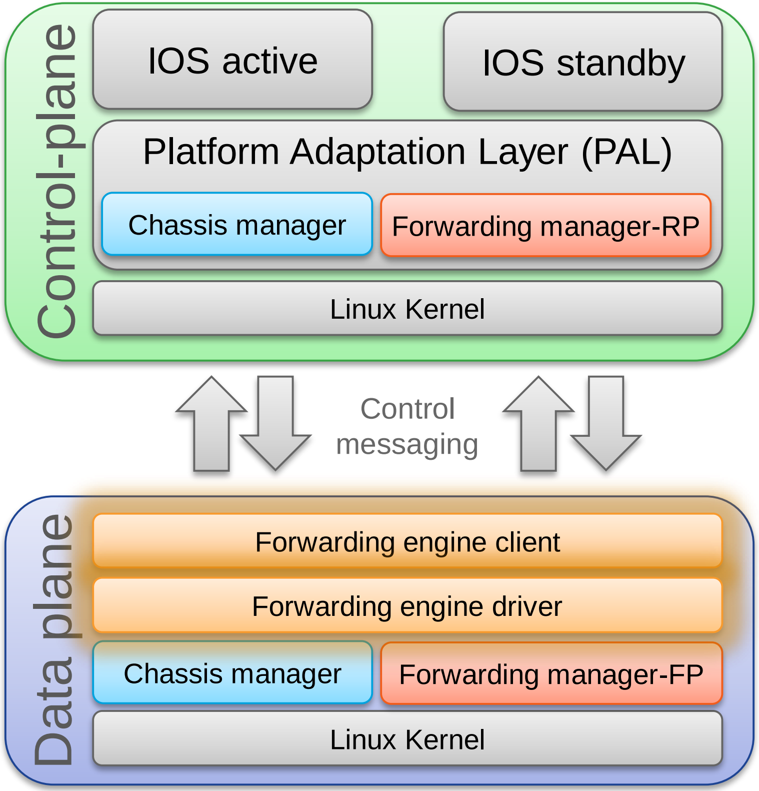
IOS XE forwarding manager

- FM in control plane communicates with peer FM processes in data plane
  - Distributed control function
- Propagates control plane operations to data plane
  - Exports forwarding information to / from IOS to data plane (CEF tables, ACLs, NAT, QoS hierarchies, etc.)
  - Maintains its own copy of forwarding state tables
- Communicates state information back to RP
  - statistics
- FM on the active control plane maintains state for both active and standby data plane hardware
  - Facilitates NSF after re-start with bulk-download of state information

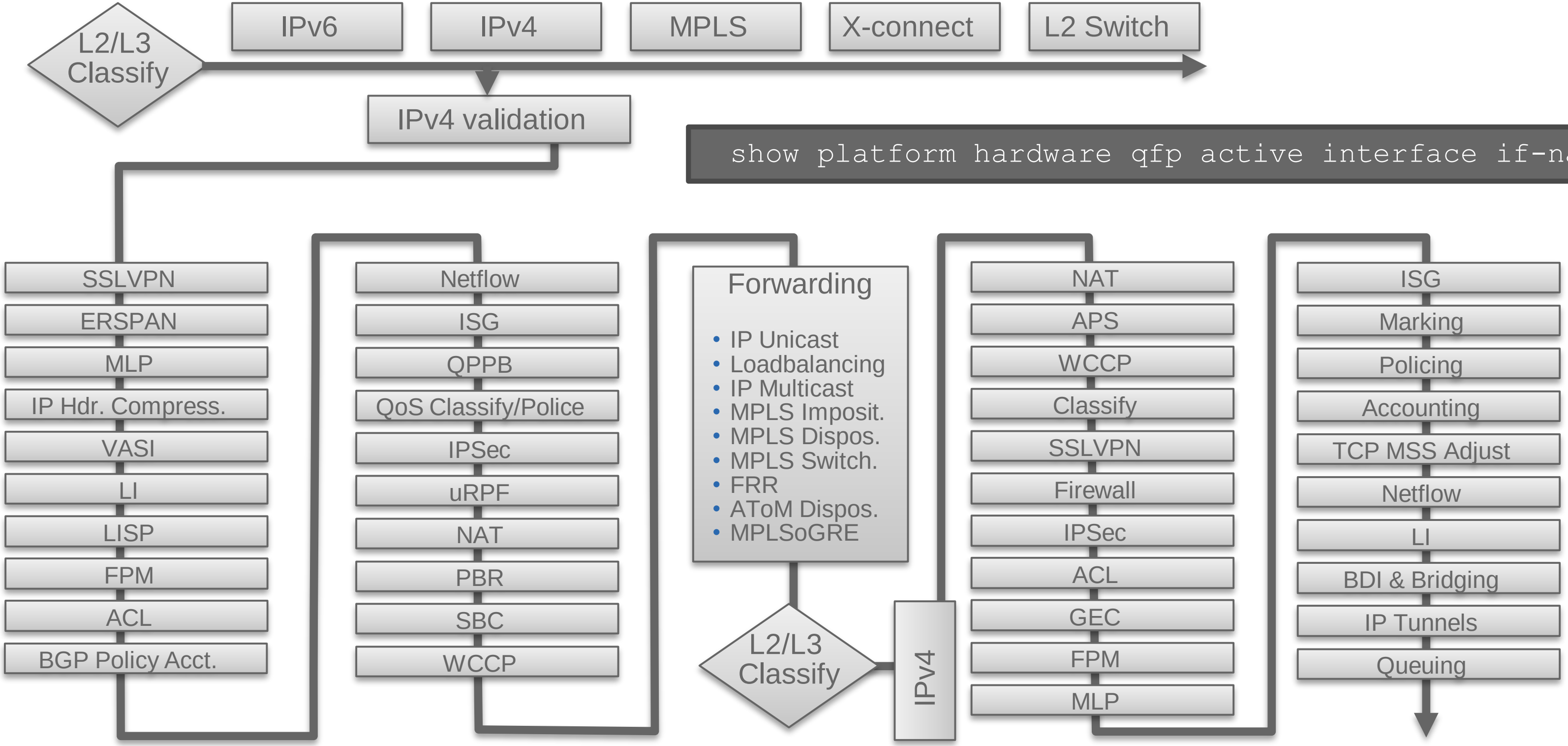


IOS XE forwarding engine control

- Forwarding engine client
  - Allocates and manages resources on forwarding engine (data structures, memory, scheduling hierarchy)
  - Receives requests from IOS via FM processes
  - Re-initializes FE and memory if a software error occurs
- Forwarding engine driver
  - Provides low-level access and control of FE
  - Provides communication path between FE client and actual forwarding engine via IPC
- Forwarding engine (runs µcode)
  - ASR1000 QFP implements data plane on PPEs
  - ISR4000 platforms use other multicore chips running the same microcode compiled for alternate processor



Feature Invocation Array – FIA



L2/L3 Classify

IPv6

IPv4

MPLS

X-connect

L2 Switch

IPv4 validation

SSLVPN

ERSPAN

MLP

IP Hdr. Compress.

VASI

LI

LISP

FPM

ACL

BGP Policy Acct.

Netflow

ISG

QPPB

QoS Classify/Police

IPSec

uRPF

NAT

PBR

SBC

WCCP

Forwarding

- IP Unicast
- Loadbalancing
- IP Multicast
- MPLS Imposit.
- MPLS Dispos.
- MPLS Switch.
- FRR
- AToM Dispos.
- MPLSoGRE

L2/L3 Classify

IPv4

NAT

APS

WCCP

Classify

SSLVPN

Firewall

IPSec

ACL

GEC

FPM

MLP

ISG

Marking

Policing

Accounting

TCP MSS Adjust

Netflow

LI

BDI & Bridging

IP Tunnels

Queuing

```
show platform hardware qfp active interface if-name <name>
```

System Architecture – ASR1000 Modular Platforms

ASR 1000 series

Compact, Powerful Router

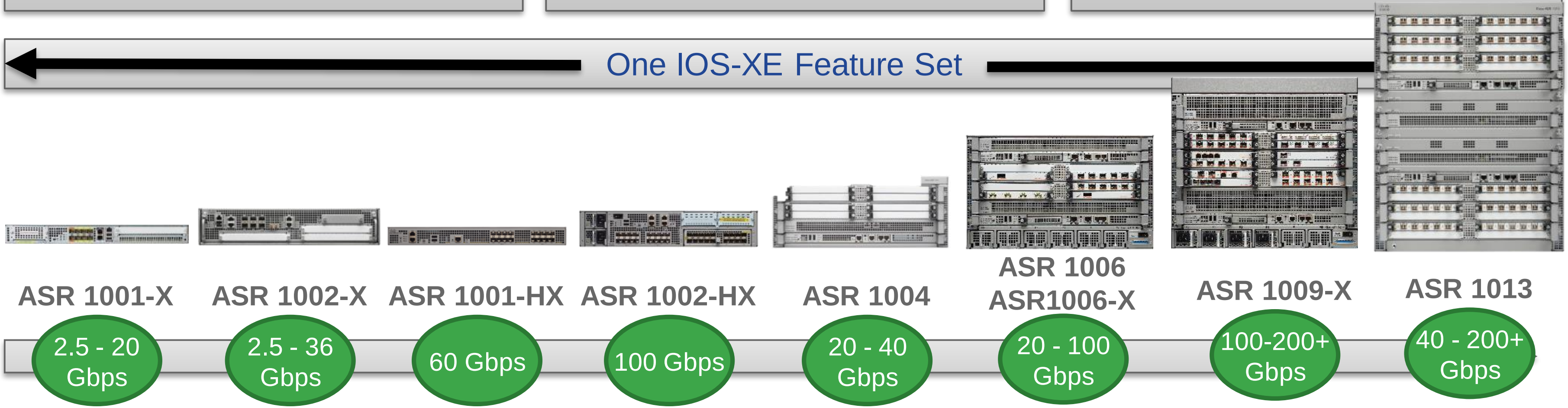
- Line-rate performance 2.5G to 200G+ with services enabled
- Hardware QoS engine with up 128K queues per ASIC
- Investment protection with modular engines, IOS CLI and SPAs for I/O

Business-Critical Resiliency

- Fully separated control and forwarding planes
- Hardware and software redundancy
- In-service software upgrades

Instant On Service Delivery

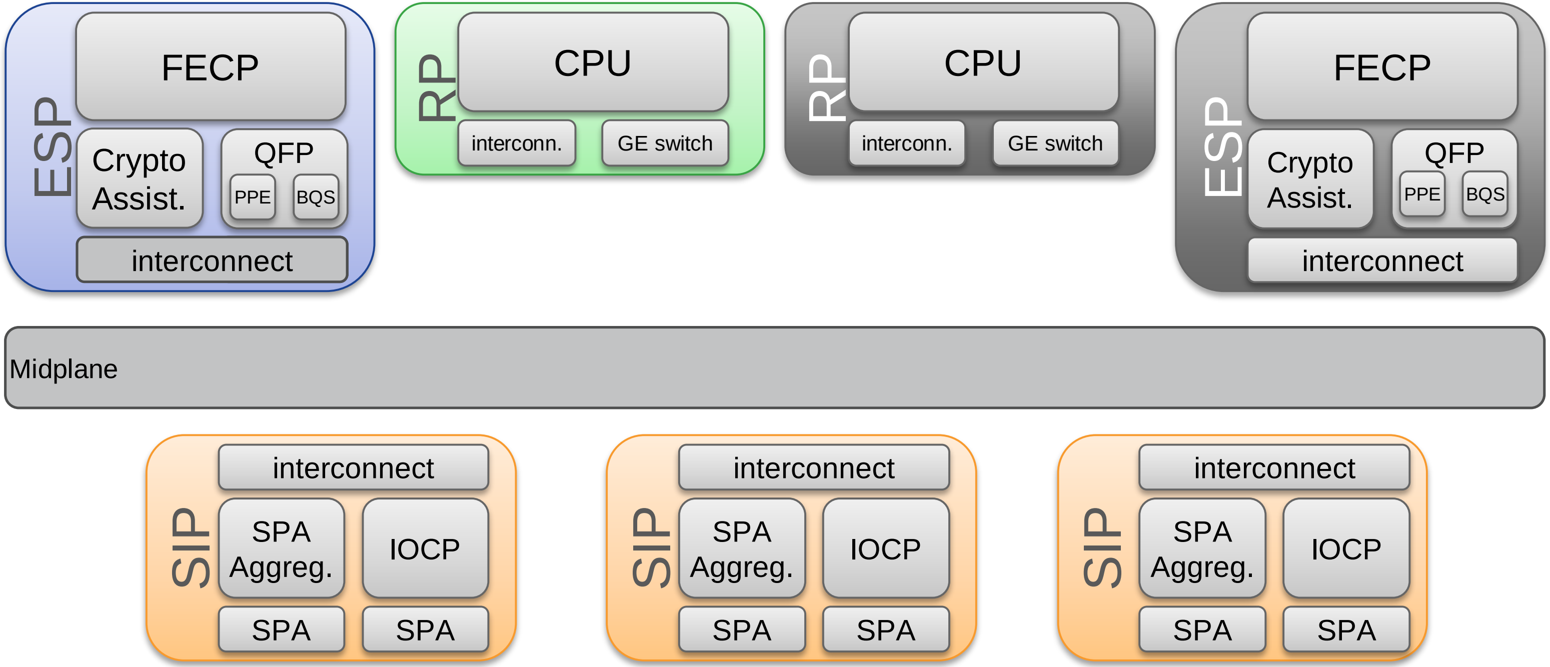
- Integrated firewall, VPN, encryption, NBAR, CUBE
- Scalable on-chip service provisioning through software licensing



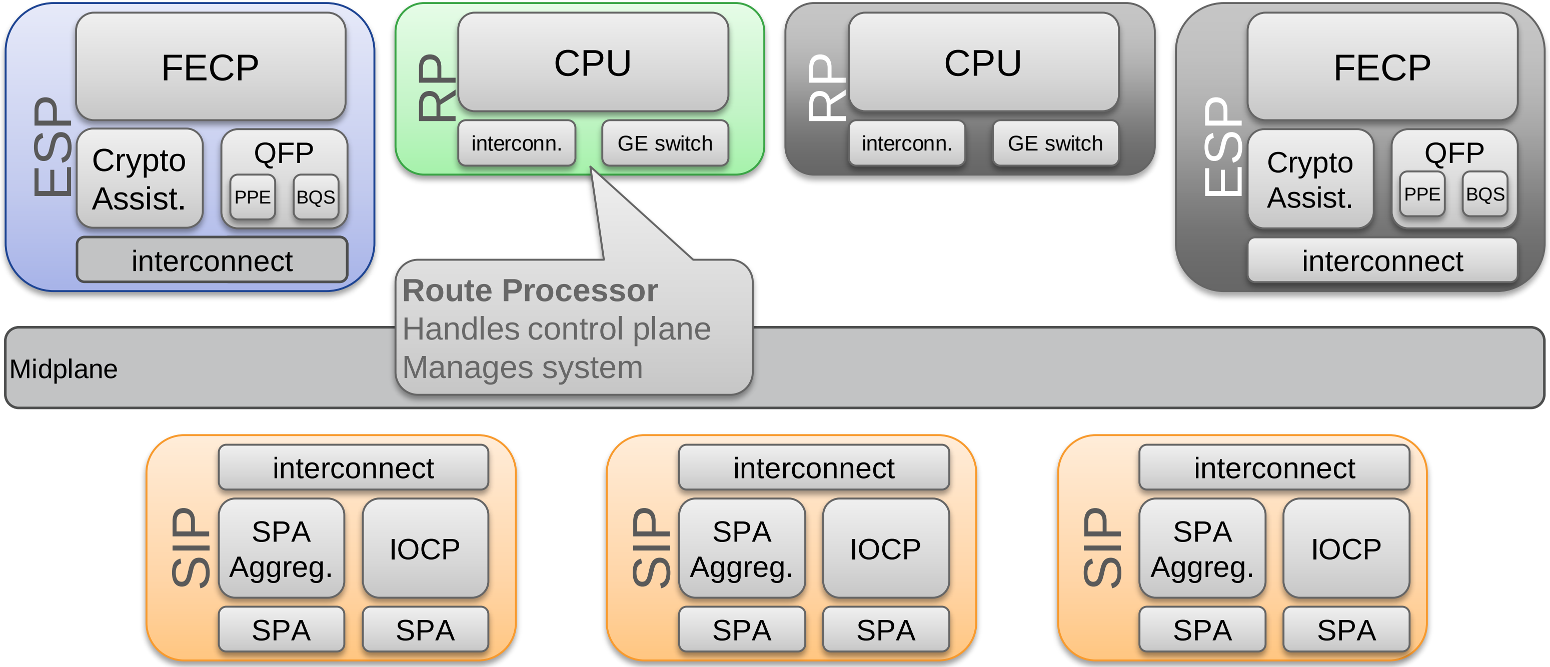
One IOS-XE Feature Set

ASR 1001-X	ASR 1002-X	ASR 1001-HX	ASR 1002-HX	ASR 1004	ASR 1006 ASR1006-X	ASR 1009-X	ASR 1013
2.5 - 20 Gbps	2.5 - 36 Gbps	60 Gbps	100 Gbps	20 - 40 Gbps	20 - 100 Gbps	100-200+ Gbps	40 - 200+ Gbps

ASR1000 building blocks

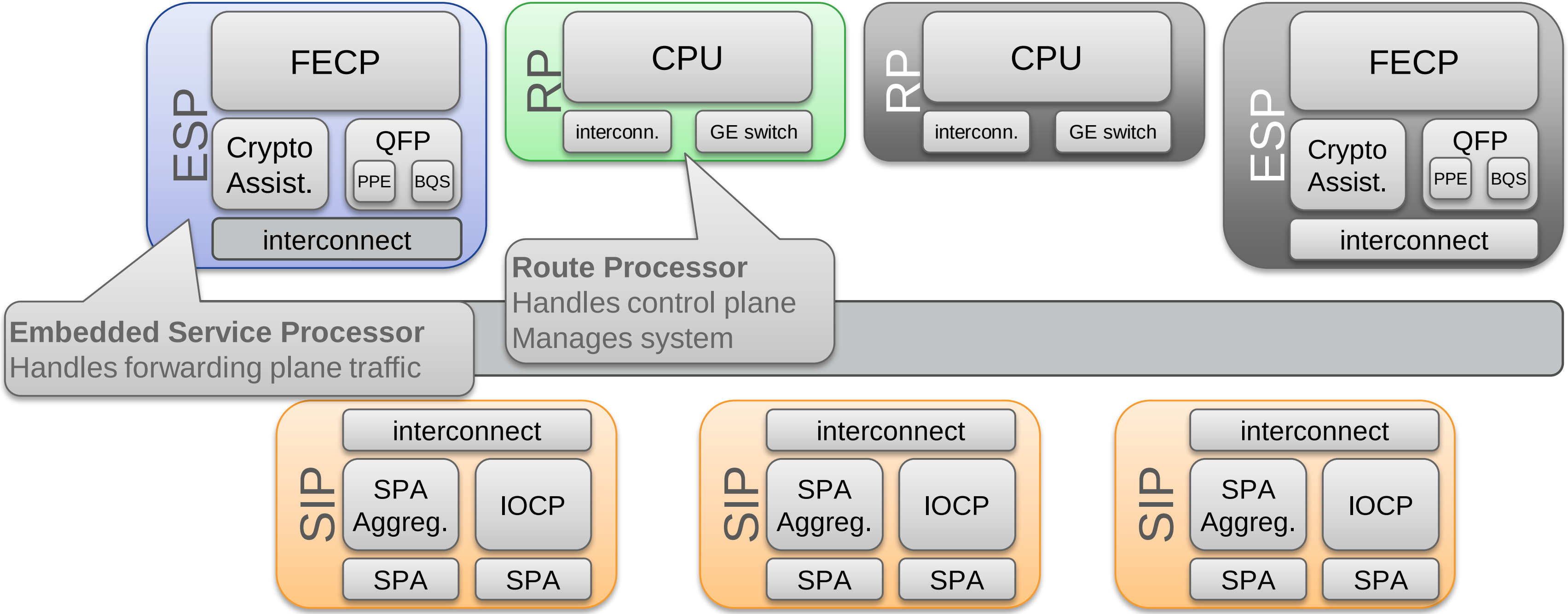


ASR1000 building blocks



Route Processor
Handles control plane
Manages system

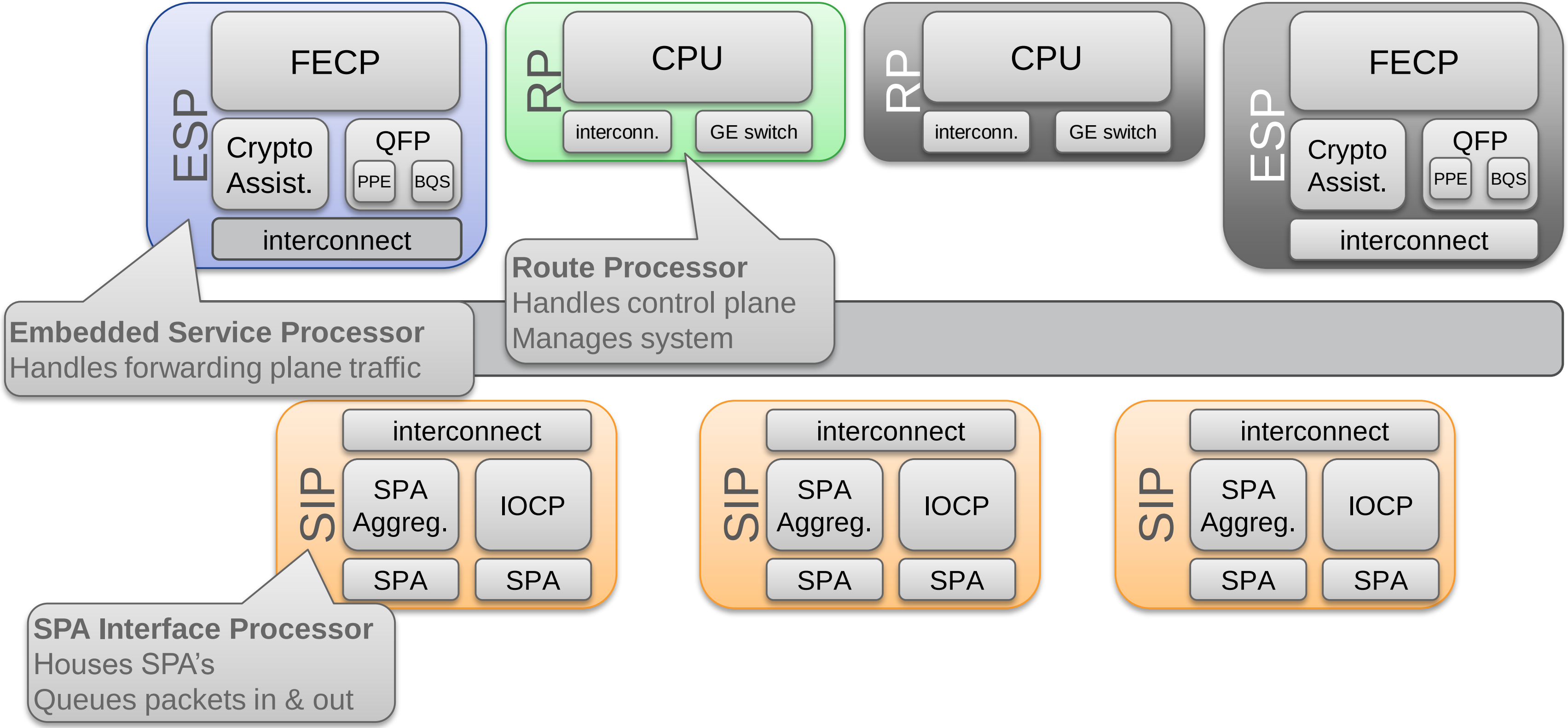
ASR1000 building blocks



Route Processor
Handles control plane
Manages system

Embedded Service Processor
Handles forwarding plane traffic

ASR1000 building blocks

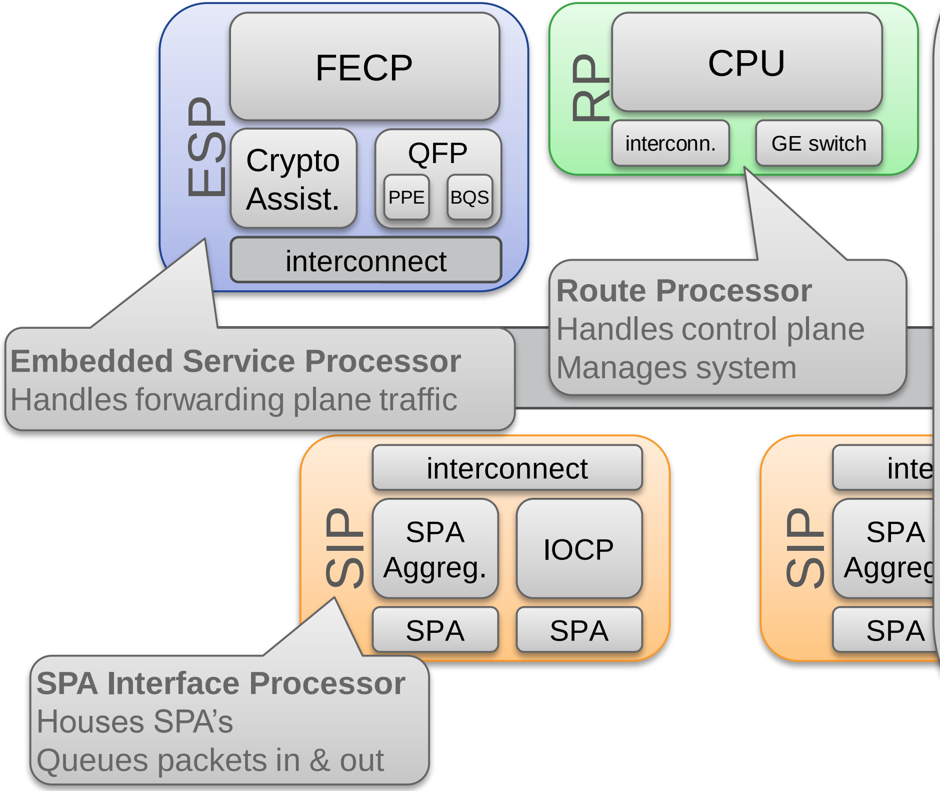


Embedded Service Processor
Handles forwarding plane traffic

Route Processor
Handles control plane
Manages system

SPA Interface Processor
Houses SPA's
Queues packets in & out

ASR1000 building blocks



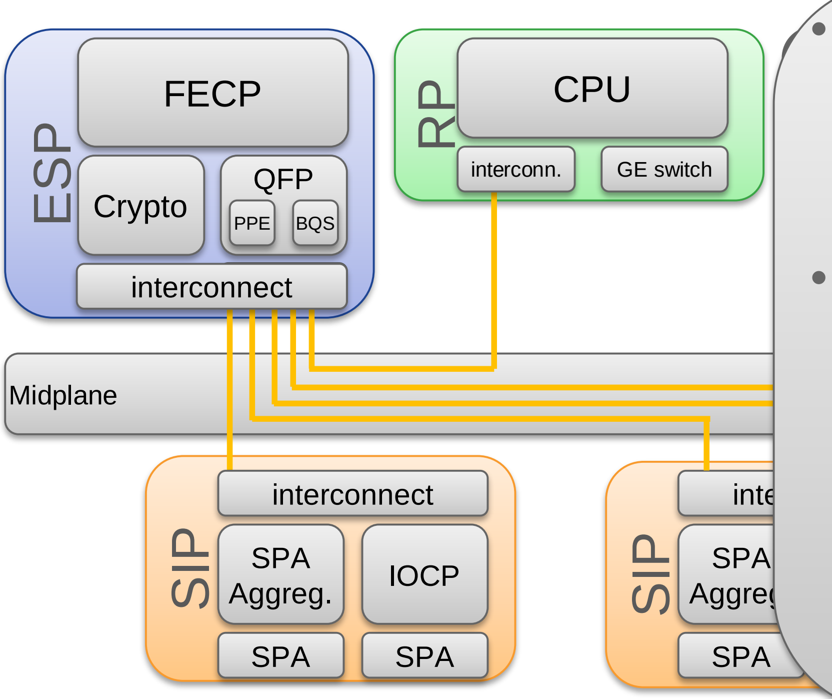
Embedded Service Processor
Handles forwarding plane traffic

Route Processor
Handles control plane
Manages system

SPA Interface Processor
Houses SPA’s
Queues packets in & out

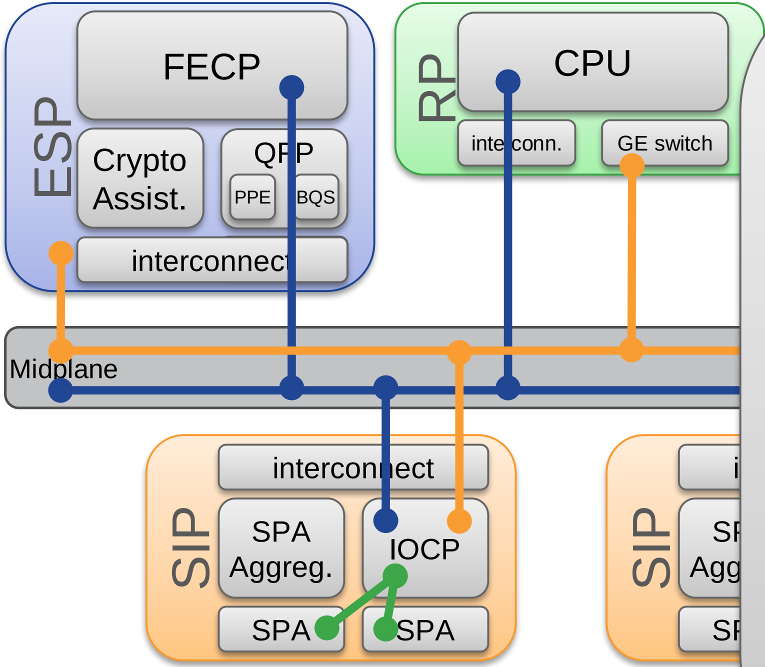
- Route Processor (RP)
  - Handles **control plane** traffic
  - Manages system
- Embedded Service Processor (ESP)
  - Handles data plane traffic
- SPA Interface Processor (SIP)
  - Shared Port Adapters provide interface connectivity
- Centralized Forwarding Architecture
  - All traffic flows through the active ESP, standby is synchronized with all flow state with a dedicated 10-Gbps link
- Distributed Control Architecture
  - All major system components have a powerful control processor dedicated for control and management planes

ASR1000 data plane architecture



- Enhanced SerDes Interconnect (ESI)
  - serial communication via midplane
  - can run at 11.5Gbps or 23Gbps
- Provides data packet communication
  - data packets between ESPs and other linecards punt/inject traffic to/from RP
  - state synchronization between ESPs
  - two ESI links between each ESP and all linecards
  - Additional full set of ESI links to standby ESP CRC protection of packet contents

ASR1000 control plane architecture



Ethernet out-of-band channel (EOBC)

- indication if cards are installed and ready loading images, stats collection
- messages to program QFP

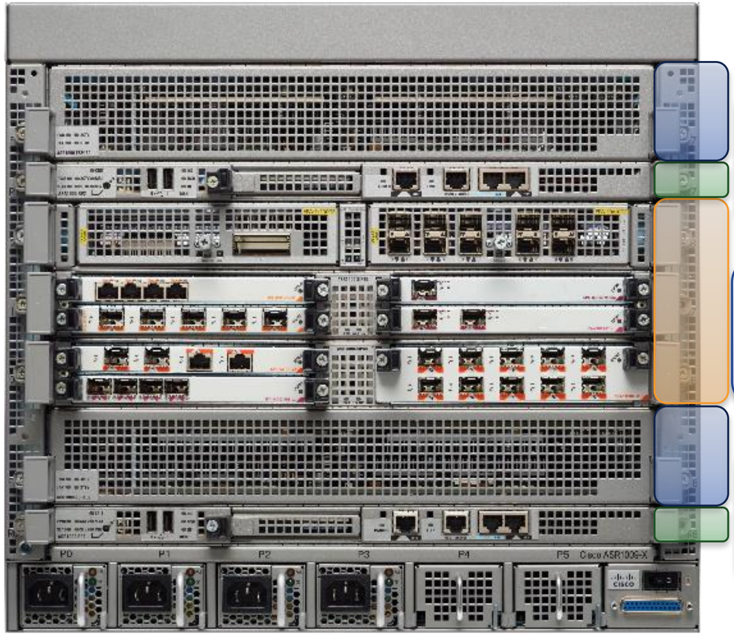
Inter-Integrated Circuit (I^{2}C)

- monitor health of hardware components
- control resets
- communicate active/standby
- real time presence and ready indicators
- control the other RP (reset, power-down,etc.)
- report power-supply status
- EEPROM access

SPA control links

- detect SPA OIR
- reset SPAs (via I^{2}C)
- power-control SPAs (via I^{2}C)
- read EEPROMs

ASR1000 chassis configuration



CC0, CC1 and CC2
SPA interface access

FP0 and FP1
data plane processing

RP0 and RP
control plane processing

ASR1004 supports redundant control planes via dual IOS process redundancy on a single RP card.

ASR1006, ASR1006-X, ASR1009-X,and ASR1013 support redundant control and data planes via active/standby hardware.

ASR1000 – power supplies

ASR1002

ASR1004

3x multispeed fan per PEM

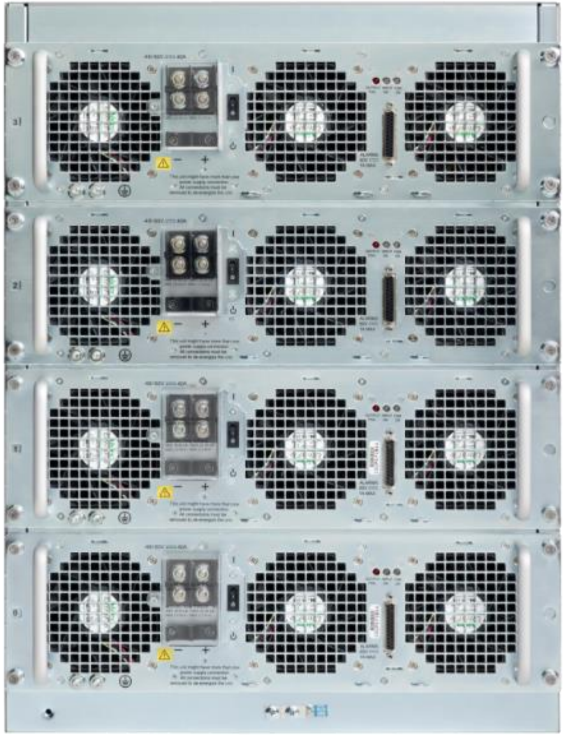
2 PEM total

ASR1006

3x multispeed fan per PEM

2 PEM total

ASR1001-X



ASR1013

3x multispeed fan per PEM

4 PEMs total

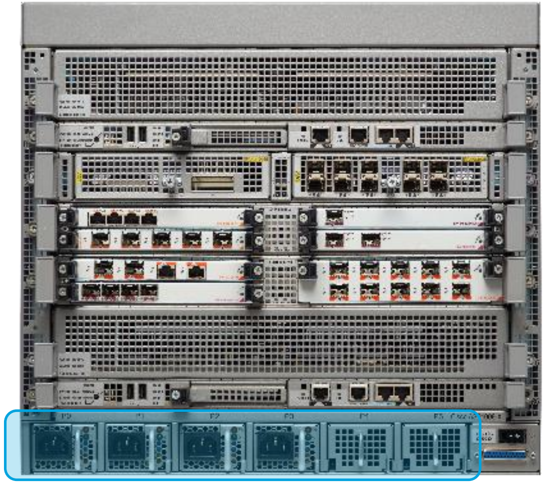
ASR1000-X – power supplies in new chassis

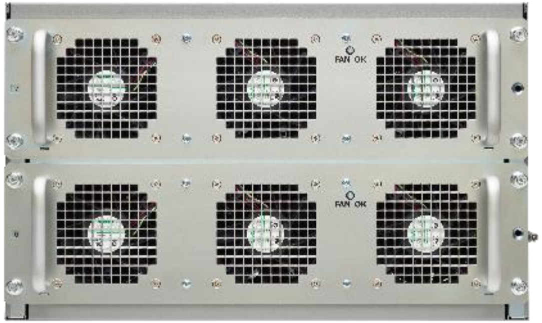
ASR1009-X

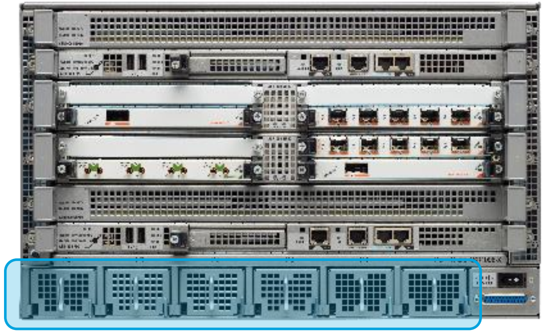
3x multispeed fans
Up to 6 power supplies
(2 included by default)

ASR1006-X

3x multispeed fans
Up to 6 power supplies
(2 included by default)







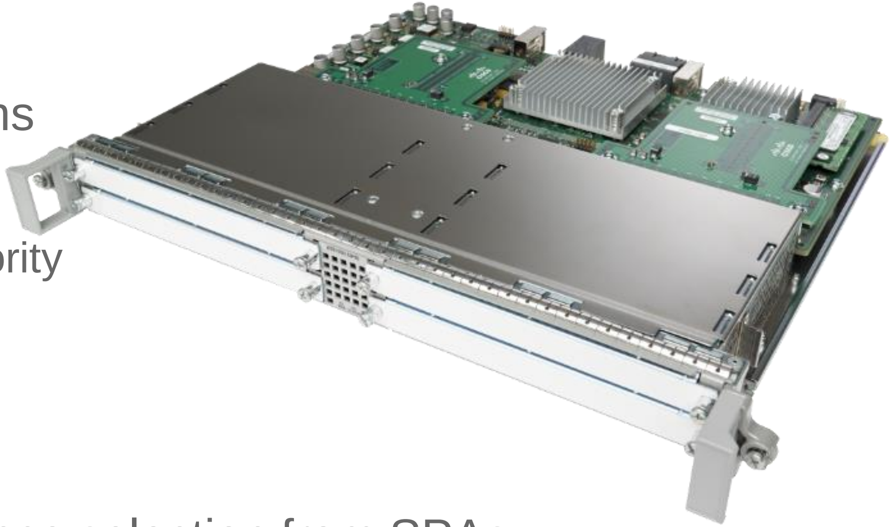
ASR1000 systems

	ASR 1001-X	ASR 1002-X	ASR 1001-HX	ASR 1002-HX	ASR 1004	ASR 1006	ASR 1006-X	ASR 1009-X	ASR 1013
Expansion slots	1 SPA 1 NIM	3 SPA	None	1 EPA	8 SPA 2 EPA	12 SPA 6 EPA	8 SPA 4 EPA	12 SPA 6 EPA	24 SPA 12 EPA
RP Slots	Integrated	Integrated	Integrated	Integrated	1	2	2	2	2
ESP Slots	Integrated	Integrated	Integrated	Integrated	1	2	2	2	2
SIP / MIP Slots	Integrated	Integrated	Integrated	Integrated	2 SIP	3 SIP	3 MIP / SIP	2 MIP / SIP	6
IOS Redundancy	Software	Software	Software	Software	Software	Hardware	Hardware	Hardware	Hardware
Built-In Ethernet	6 GE + 2 TenGE	6 GE	8 GE + 4 TenGE + 4 Flex GE	8 GE + 8 TenGE	N/A	N/A	N/A	N/A	N/A
Bandwidth Gbps	2.5 to 20	5 to 36	60	100	20 to 40	20 to 100	40 to 100	40 to 200	40-200+

ASR1000 systems

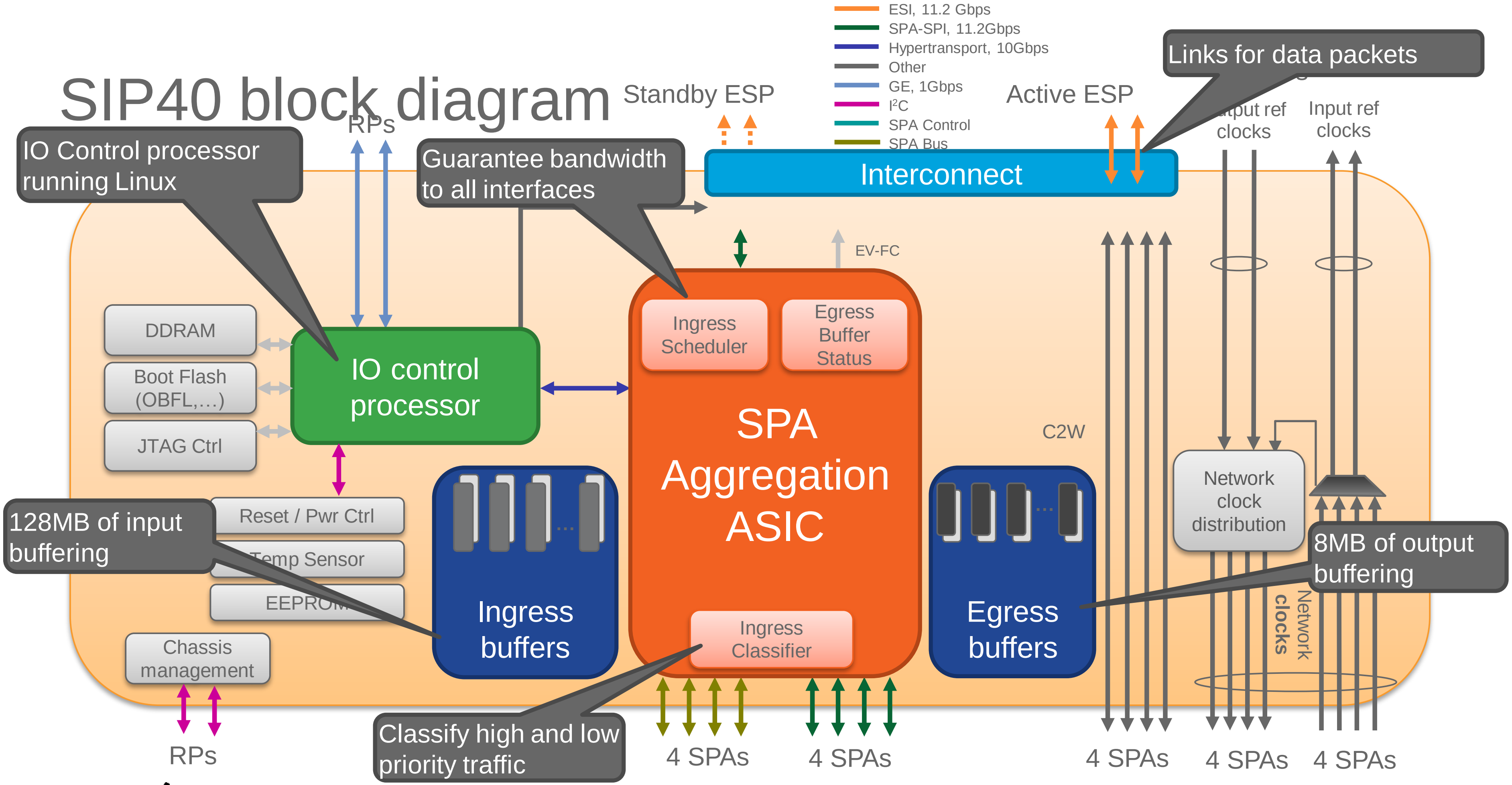
	ASR 1001-X	ASR 1002-X	ASR 1001-HX	ASR 1002-HX	ASR 1004	ASR 1006	ASR 1006-X	ASR 1009-X	ASR 1013
Height	1.75” (1RU)	3.5” (2RU)	6 GE	6 GE	7” (4RU)	10.5” (6RU)	10.5” (6RU)	15.75” (9 RU)	22.7” (13RU)
Max AC Output Power	250W	470W	360W	500W	765W	1275W	4030W	4575W	3390W
Airflow	Front to back	Front to back	Front to back	Front to back	Front to back	Front to back	Front to back	Front to back	Front to back

ASR1000 SPA interface processor (aka SIP)



- Supports up to 4 SPAs, full OIR support
- Does not participate in forwarding decisions
- Preliminary QoS
  - Ingress packet classification – high & low priority
  - Ingress over-subscription buffering
  - 128MB of ingress oversubscription buffering
- Capture stats on dropped packets
- Network clock distribution to SPAs, reference selection from SPAs
- IOCP manages midplane links, SPA OIR, SPA drivers
- SIP40 supports minimally disruptive restart for ISSU (MDR)
  - SIP reboot times of 25 seconds or less
  - SPA reboot times of 10 seconds or less

SIP40 block diagram



IO Control processor running Linux

Guarantee bandwidth to all interfaces

Links for data packets

128MB of input buffering

8MB of output buffering

Classify high and low priority traffic

For your reference

Supported SPAs and SFPs

WAN optics	Ethernet Optics	POS SPAs	Serial SPAs	Ethernet SPAs
SFP-OC3-MM	SFP-GE-S /	SPA-2XOC3-POS	SPA-4XT-Serial	SPA-4X1FE-TX-V2
SFP-OC3-SR	GLC-SX-MMD	SPA-4XOC3-POS	SPA-8XCHT1/E1	SPA-8X1FE-TX-V2
SFP-OC3-IR1	SFP-GE-L /	SPA-8XOC3-POS	SPA-4XCT3/DS0	SPA-2X1GE-V2
SFP-OC3-LR1	GLC-LH-SMD	SPA-1XOC12-POS	SPA-2XCT3/DS0	SPA-5X1GE-V2
SFP-OC3-LR2	SFP-GE-Z	SPA-2XOC12-POS	SPA-1XCHSTM1/OC3	SPA-8X1GE-V2
SFP-OC12-MM	SFP-GE-T	SPA-4XOC12-POS	SPA-1XCHOC12/DS0	SPA-10X1GE-V2
SFP-OC12-SR	CWDM	SPA-8XOC12-POS	SPA-2XT3/E3	SPA-1X10GE-L-V2
SFP-OC12-IR1	XFP-10GLR-OC192SR /	SPA-1XOC48-POS/RPR	SPA-4xT3/E3	SPA-1X10GE-WL-V2
SFP-OC12-LR1	XFP10GLR-192SR-L	SPA-2XOC48POS/RPR		SPA-2X1GE-SYNCE
SFP-OC12-LR2	XFP-10GER-192IR+ /	SPA-4XOC48POS/RPR		
SFP-OC48-SR	XFP10GER-192IR-L	SPA-OC192POS-XFP		
SFP-OC48-IR1	XFP-10GZR-OC192LR			
SFP-OC48-LR2	XFP-10G-MM-SR	**ATM SPAs**	**Service SPAs**	**CEOPs SPAs**
	DWDM-XFP			
XFP-10GLR-OC192SR	(32 fixed channels)	SPA-1XOC3-ATM-V2	SPA-WMA-K9	SPA-1CHOC3-CE-ATM
XFP-10GER-OC192IR		SPA-3XOC3-ATM-V2	SPA-DSP	SPA-24CHT1-CE-ATM
XFP-10GZR-OC192LR	GLC-GE-100FX	SPA-1XOC12-ATM-V2		
	GLC-BX-U	SPA-2CHT3-CE-ATM		
	GLC-BX-D			

For your reference

ASR1000 End-of-Sale platform hardware

End of Sale			Replacement
PID	EoS Announce	EoS Date	PID
ASR1000-ESP5	31-Mar-2015	29-Apr-2016	ASR1000-ESP20 ASR1002-X
ASR1000-ESP10	31-Mar-2015	29-Apr-2016	ASR1000-ESP20 ASR1002-X
ASR1000-RP1	31-Mar-2015	29-Apr-2016	ASR1000-RP2
ASR1000-SIP10	31-Mar-2015	29-Apr-2016	ASR1000-SIP40
ASR1001	31-Mar-2015	29-Apr-2016	ASR1001-X
ASR1002	31-Mar-2015	29-Apr-2016	ASR1002-X

ASR1000 MIP-100 interface card (aka MIP)

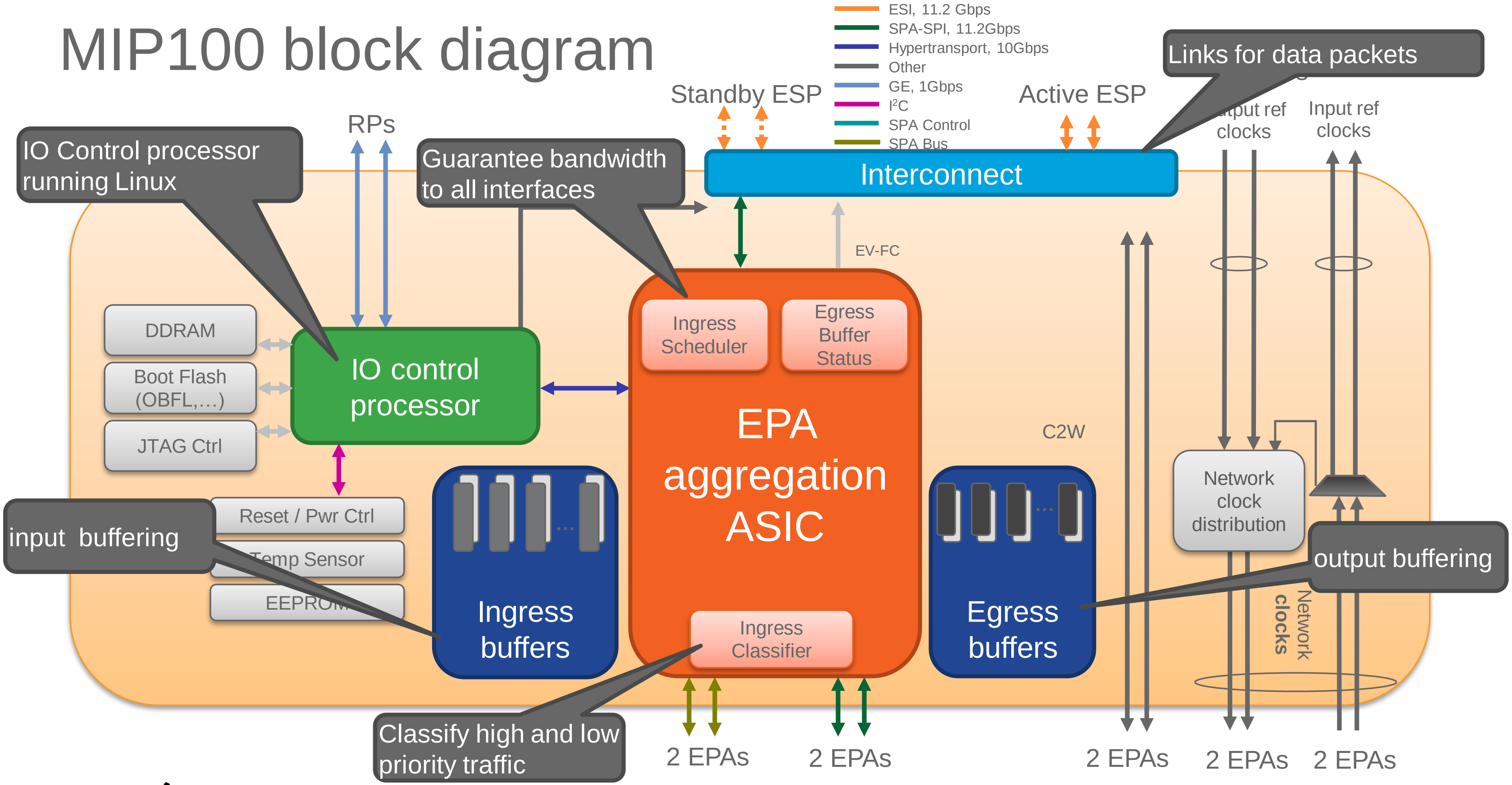
- Supports up to 2 EPAs, full OIR support
- Does not participate in forwarding decisions
- Preliminary QoS
  - Ingress packet classification – high & low priority
  - Ingress over-subscription buffering
- Capture stats on dropped packets
- Network clock distribution to SPAs, reference selection from SPAs
- IOCP manages midplane links, SPA OIR, SPA drivers
- MIP100 supports minimally disruptive restart for ISSU (MDR)

EPAs for the MIP-100

- **Ethernet Port Adapters** support Ethernet only
- Feature parity with existing Ethernet interfaces
- SyncE, Y.1731 (CFM)
- MACSEC
- Currently four models
  - EPA-18X1GE
  - EPA-10X10GE
  - EPA-CPAK-2X40GE
  - EPA-1X100GE

EPA	Optics	
EPA-1X100GE	CPAK-100G-SR10	CPAK-100G-LR4
EPA-CPAK-2X40GE	CPAK-100G-SR10 with required breakout cable CAB-MPO24-2XMPO12	
EPA-10X10GE	SFP-10G-SR SFP-10G-SR-X SFP-10G-LR	SFP-10G-LRM SFP-10G-LR-X SFP-10G-ER
EPA-18X1GE	GLC-GE-100FX GLC-SX-MMD GLC-LH-SMD SFP-GE-T GLC-BX-U GLC-BX-D GLC-TE	GLC-SX-MM GLC-LH-SM GLC-EX-SMD GLC-ZX-SMD DWDM-SFP CWDM-SFP

MIP100 block diagram



IO Control processor running Linux

Guarantee bandwidth to all interfaces

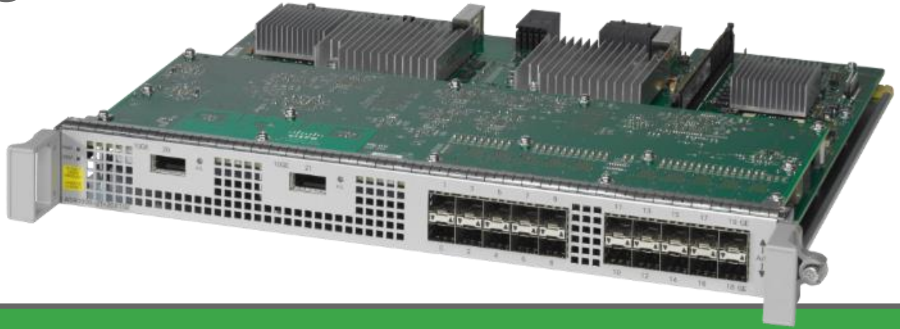
Links for data packets

input buffering

output buffering

Classify high and low priority traffic

Fixed configuration Ethernet linecards



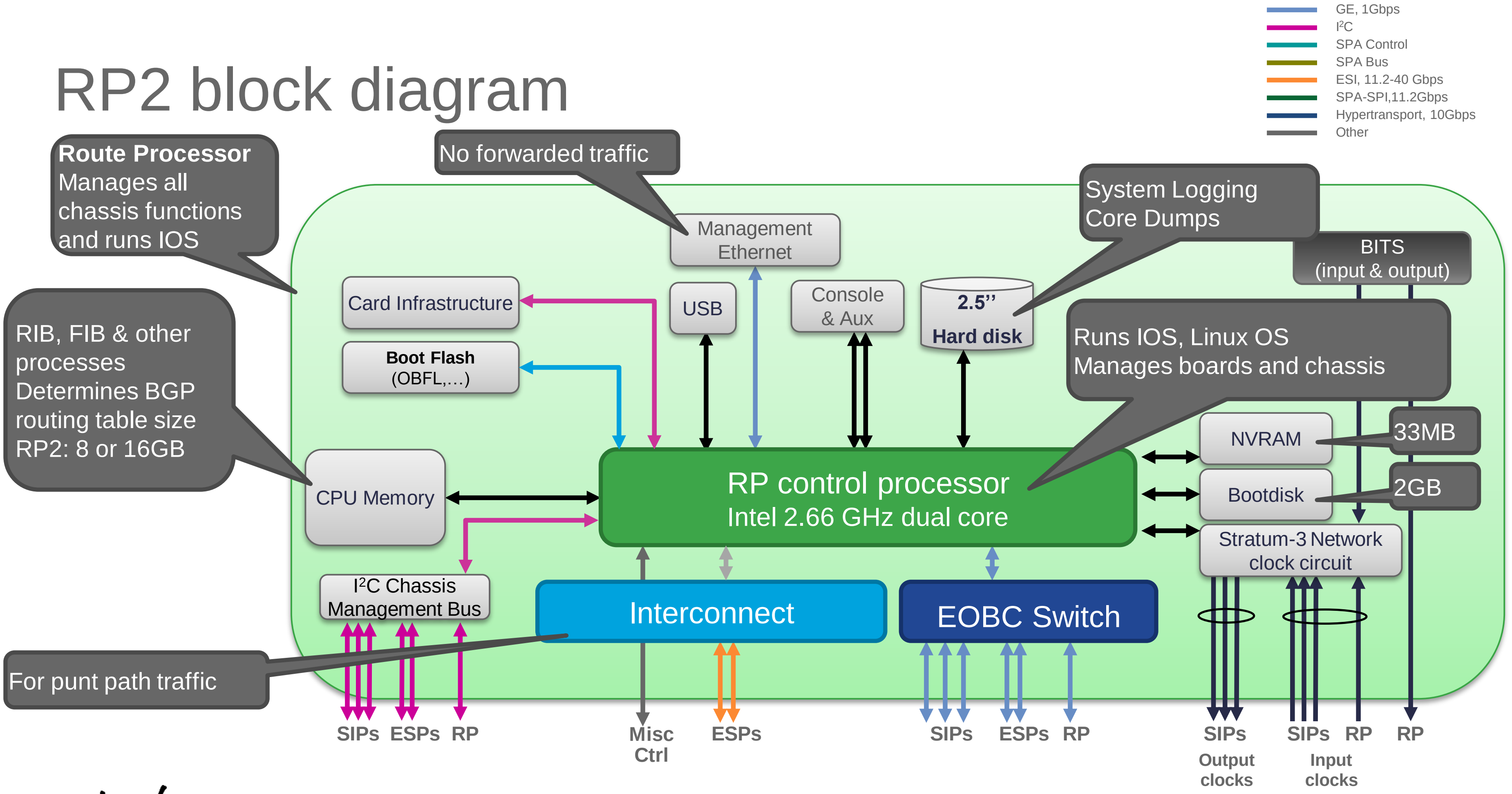
Support	Key features and advantages
ASR1000-2T+20X1GE – XE3.10	Full feature parity with existing Ethernet interfaces
ASR1000-6TGE – XE3.12	SyncE, Y.1731 (CFM)
Requires modular chassis (1004, 1006, 1006-X, 1009-X, or 1013)	Significant price savings versus fully populated SIP40 with corresponding SPAs
Requires RP2 with minimum of ESP40	Line rate performance for all interfaces concurrently

Modular route processors: RP2 and RP3

- RP2
  - 2.66Ghz Intel dual-core architecture
  - 64-bit IOS XE
  - Up to 16GB IOS memory
  - 2GB Bootflash (eUSB)
  - Hot swappable 80GB hard drive

- RP3
  - 2.66Ghz Intel quad-core architecture
  - 64-bit IOS XE
  - Up to 64GB IOS memory
  - Crypto co-processor to aid in crypto session setup
  - 2GB Bootflash (eUSB)
  - Hot swappable 100GB solid state hard drive

RP2 block diagram



Route Processor
Manages all chassis functions and runs IOS

No forwarded traffic

System Logging
Core Dumps

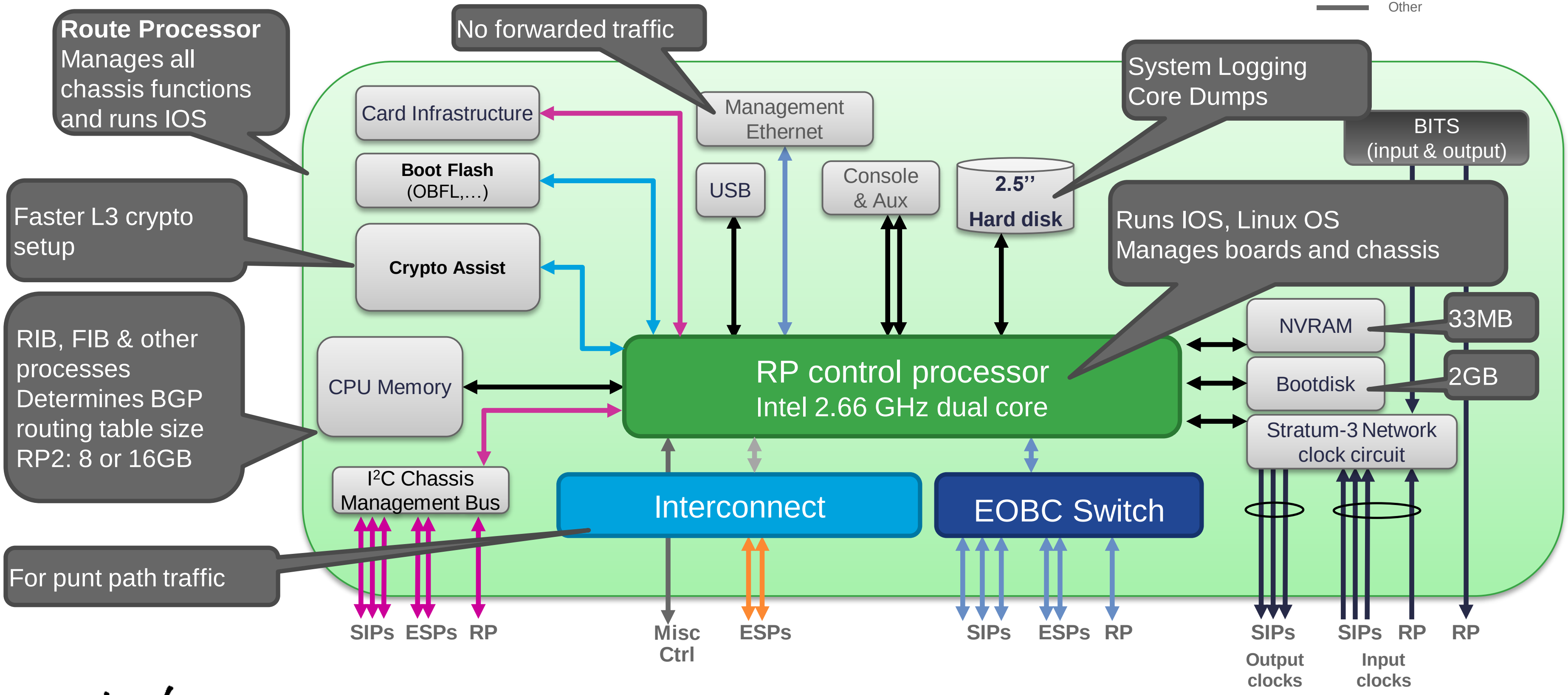
RIB, FIB & other processes
Determines BGP routing table size
RP2: 8 or 16GB

Runs IOS, Linux OS
Manages boards and chassis

For punt path traffic

RP3 block diagram

🟦	GE, 1Gbps
🟪	I²C
🟦	SPA Control
🟩	SPA Bus
🟧	ESI, 11.2-40 Gbps
🟩	SPA-SPI,11.2Gbps
🟦	Hypertransport, 10Gbps
⬛	Other



Route Processor Manages all chassis functions and runs IOS

No forwarded traffic

System Logging Core Dumps

Faster L3 crypto setup

Runs IOS, Linux OS Manages boards and chassis

RIB, FIB & other processes Determines BGP routing table size RP2: 8 or 16GB

For punt path traffic

Cisco live!

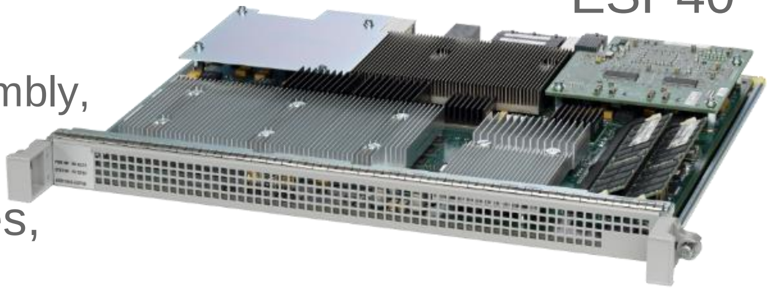
Route processor overview

	ASR1001-X	ASR1002-X	ASR1001-HX	ASR1002-HX	RP2	RP3
CPU	quad core, 2.0GHz	quad core, 2.13GHz	quad core, 2.5Ghz	quad core, 2.5Ghz	dual core, 2.66GHz	quad core, 2.66GHz
Default memory	8GB (4x2GB)	4GB	8GB	16GB	8GB, 4x2GB	16GB, 4x4GB
Memory upgrade options	16GB (4x4GB)	16GB (4x4GB)	16GB	32GB	16GB (4x4GB)	64GB (4x16GB)
Built-In eUSB Bootflash	8GB	8GB	8GB	8GB	2GB	2GB
Storage	optional 160 GB HDD external USB	optional 160GB HDD external USB	external USB	NIM Module HDD external USB	80GB HDD external USB	100GB solid state HDD external USB
IOS XE OS	64 bit	64 bit	64 bit	64 bit	64 bit	64 bit
Chassis Support	integrated	integrated	integrated	integrated	ASR1004, ASR1006 ASR1006-X, ASR1009-	ASR1006-X, ASR1009-X, ASR1013

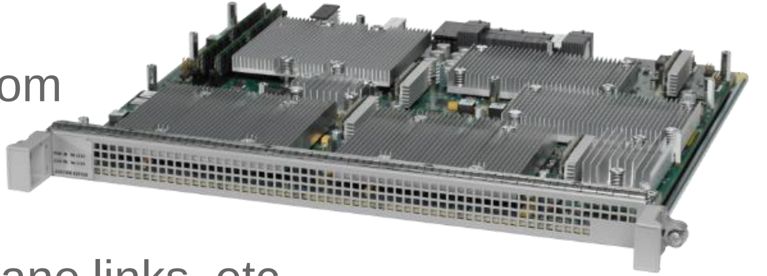
ASR1000 Embedded Services Processor

- Centralized, programmable forwarding engine providing full-packet processing
  - Packet Buffering and Queuing/Scheduling (BQS)
  - For output traffic to carrier cards/SPAs
  - For special features such as traffic shaping, reassembly, replication, punt to RP, cryptography, etc.
- 5 levels of HQoS scheduling, up to 464K Queues, Priority Propagation
- Dedicated crypto co-processor
- Interconnect providing data path links (ESI) to/from other cards over midplane
  - Input scheduler for allocating QFP BW among ESIs
- FECP CPU manages QFP, crypto device, midplane links, etc.

ESP40

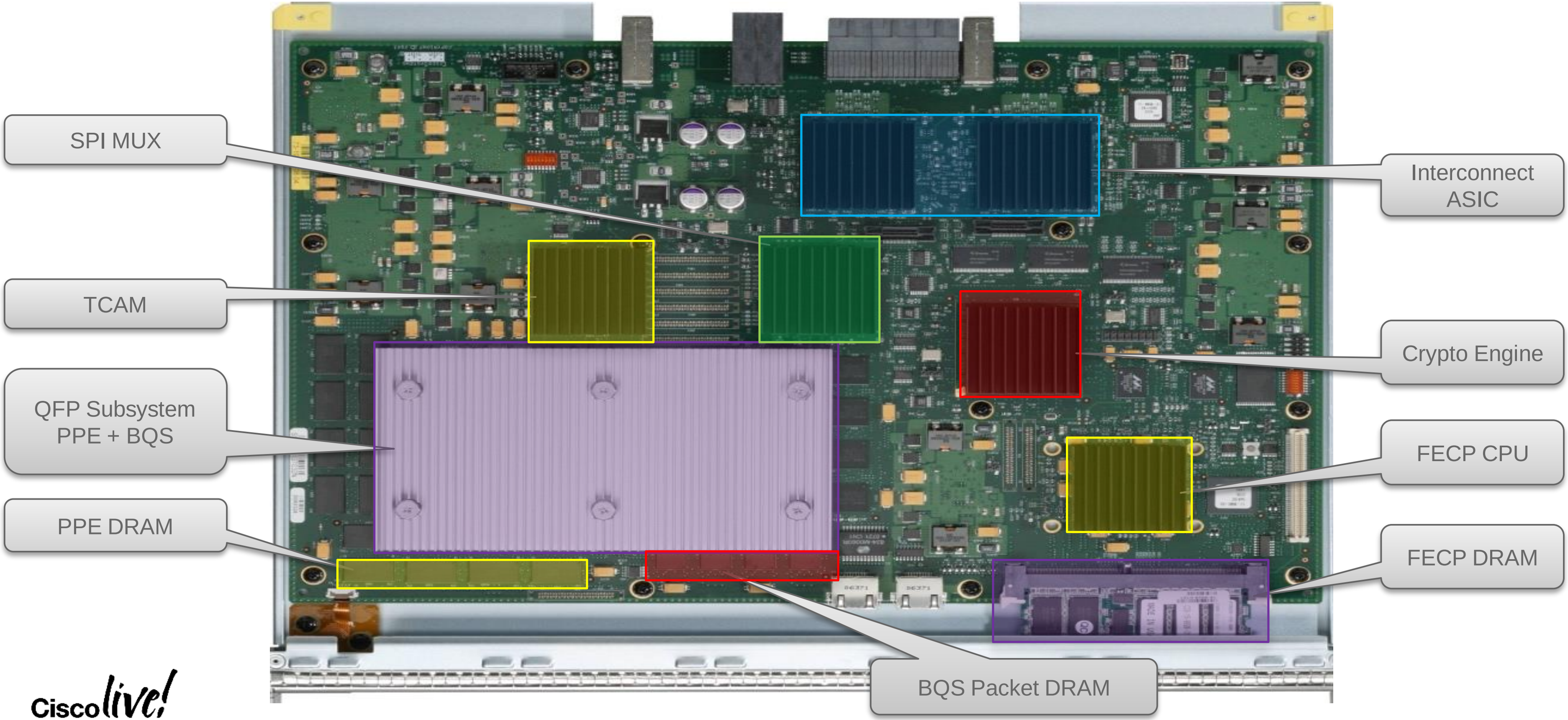


ESP100

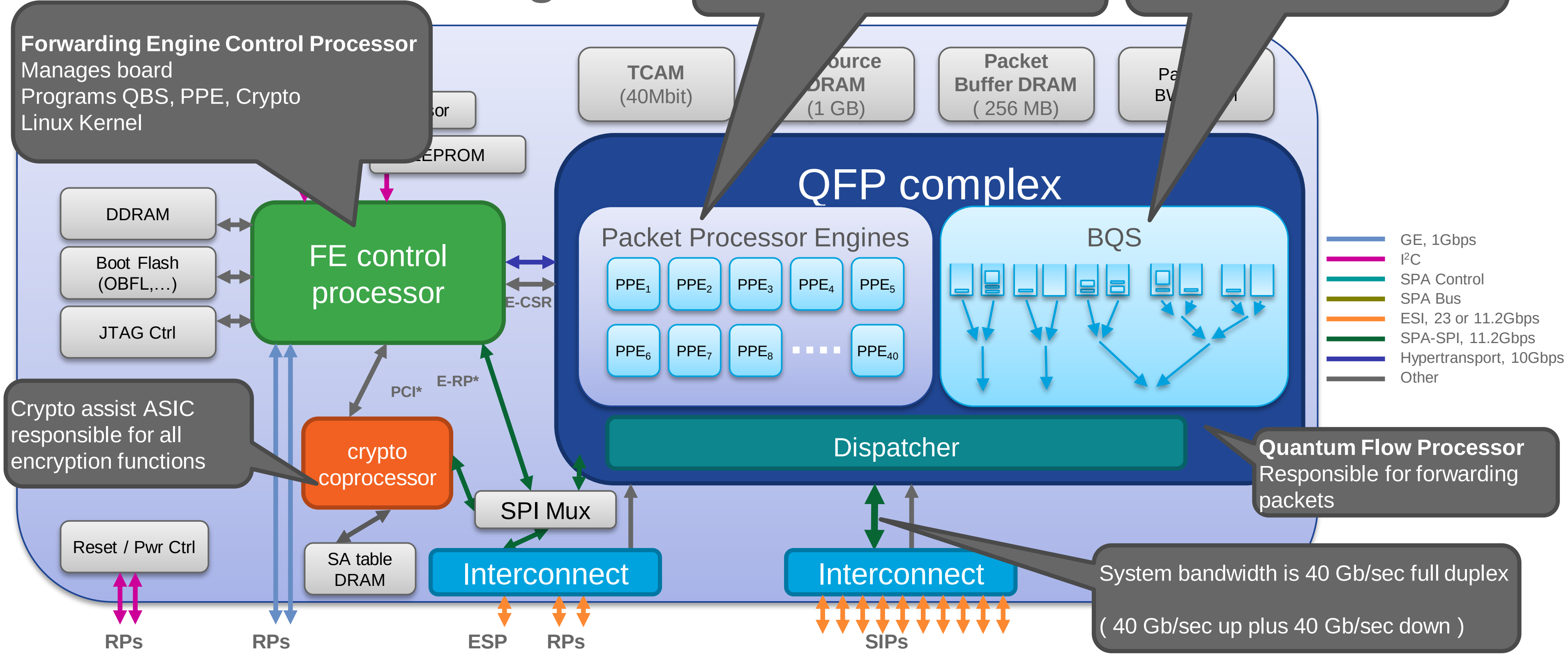


For your reference

ASR1000 Embedded Services Processor



ESP40 block diagram



PPE engines
responsible for all feature implementation

Buffering Queuing & Scheduling
Executes complex QoS scheduling
Queues and schedules packets

Forwarding Engine Control Processor
Manages board
Programs QBS, PPE, Crypto
Linux Kernel

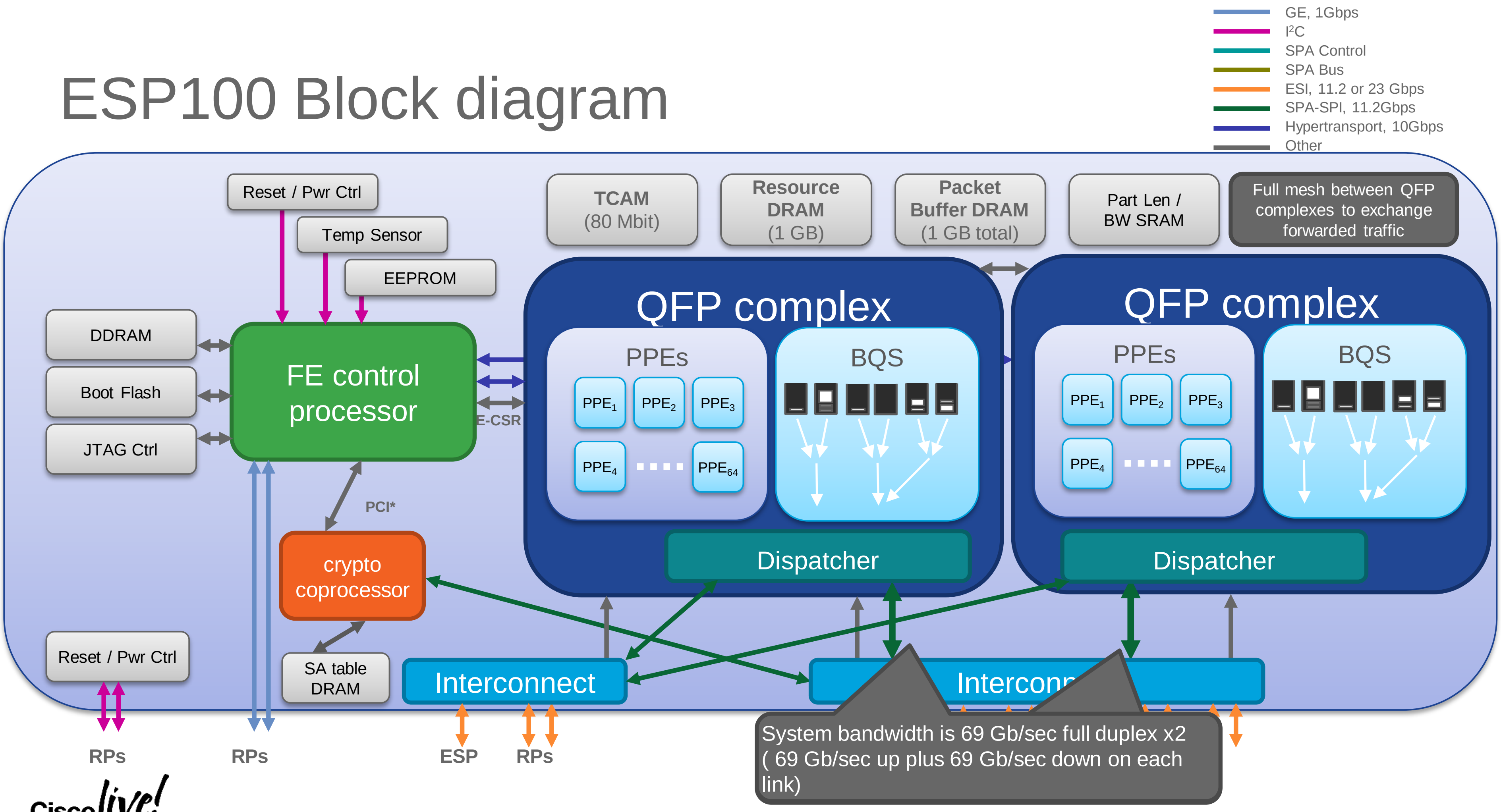
Crypto assist ASIC responsible for all encryption functions

Quantum Flow Processor
Responsible for forwarding packets

System bandwidth is 40 Gb/sec full duplex

(40 Gb/sec up plus 40 Gb/sec down)

ESP100 Block diagram



System bandwidth is 69 Gb/sec full duplex x2 (69 Gb/sec up plus 69 Gb/sec down on each link)

For your reference

ASR1000 ESP reference

	ASR 1001-X	ASR 1002-X	ASR 1001-HX	ASR 1002-HX	ESP20	ESP40	ESP100	ESP200
System bandwidth	2.5/5/10/20 Gpbs	5/10/20/36 Gbps	44 – 60 Gbps port based	44 – 100 Gbps port based	20 Gbps	40 Gbps	100 Gbps	200 Gpbs
Performance	17 Mpps	30 Mpps	59 Mpps	58 Mpps	24 Mpps	24 Mpps	59 Mpps	113 Mpps
QFP Cores	31	64	128	128	40	40	128	256
Crypto BW (1400B)	8 Gpbs	4 Gbps	16 Gbps	25 Gpbs	8.5 Gbps	11 Gbps	29 Gbps	78 Gbps
QFP Resource Mem	4GB (unified)	1GB	2 GB / QFP 4GB Total	4GB	1GB	1GB	2 GB / QFP 4GB Total	2 GB / QFP 8GB total
Packet Buffer	512MB (unified)	512MB	1GB	1GB	256MB	256MB	1GB	2GB
Control CPU	Quad core 2.00 GHz	Quad core 2.13 GHz	Quad core 2.5 GHz	Quad core 2.5 GHz	Single core 1.2 GHz	Dual core 1.8 GHz	Dual core 1.73 GHz	Dual core 1.73 GHz
Control Memory	shared	shared	shared	shared	4 GB	8 GB	16 GB	32 GB
TCAM	10 Mb	40 Mb	40 Mb	80 Mb	40 Mb	40 Mb	80 Mb	2 x 80 Mb
Chassis Support	Integrated	Integrated	Integrated	Integrated	1004, 1006	1004, 1006, 1013, 1006-X, 1009-X	1006, 1013, 1006-X,1009-X	1009-X, 1013

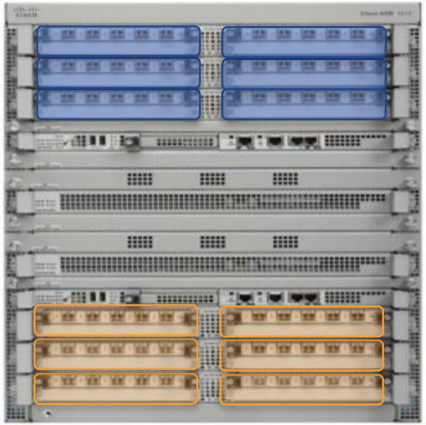
For your reference

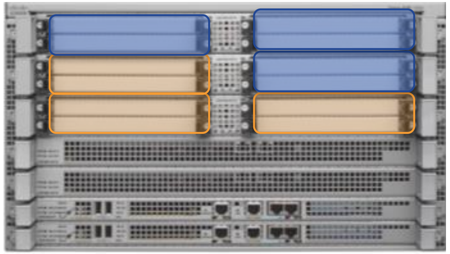
ASR1000 computational reference

FRU	RP2	RP3	ASR 1001-X	ASR 1002-X	ASR 1001-HX	ASR 1002-HX	ESP20	ESP40	ESP100	ESP200
Control plane cores	2	4	4	4	4	4	1	1	2	2
Control plane clocking	2.66 GHz	2.2 GHz	2.00 GHz	2.13 GHz	2.5 GHz	2.5 GHz	800 MHz	1.80 GHz	1.73 GHz	1.73 GHz
Data plane cores			31	62	62	128	40	40	128	256
Data plane clocking			1.5 GHz	1.2 GHz	1.5 GHz	1.5 GHz	1.2 GHz	1.2 GHz	1.5 GHz	1.5 GHz
Control plane SDRAM	8/16 GB	8/16/32/64 GB	8/16 GB	4/8/16 GB	8/16 GB	16/32 GB	4 GB	8 GB	16 GB	32 GB
Bootflash	2 GB	8 GB	8 GB	8 GB	32 GB	32 GB				
NVRAM	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
QFP memory			4 GB	1 GB	1 GB	4 GB	1 GB	1 GB	4 GB	8 GB
Packet buffer memory			512 MB	512 MB	512 MB	1 GB	256 MB	256 MB	1 GB	2 GB
TCAM			10 Mbit	40 Mbit	40 Mbit	80 Mbit	40 Mbit	40 Mbit	80 Mbit	2x80 Mbit

ESP-100 and QFP Responsibilities

- Each ESP 100 uses 2 QFP-NG ASICs to achieve performance
- Each QFP-NG is associated with a subset of the SPA-bays and interfaces
- Should be taken into account for
  - QoS
  - Multicast
  - NAT



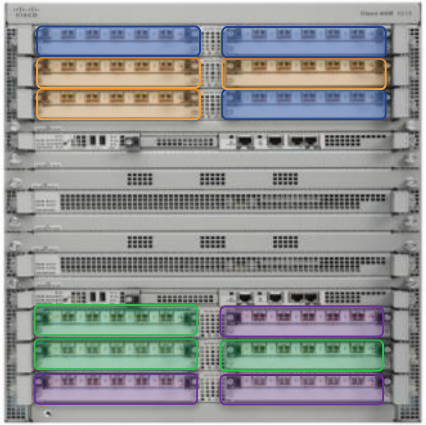


🟦 Egress queuing for interfaces handled by QFP0

🟧 Egress queuing for interfaces handled by QFP1

ESP-200 and QFP Responsibilities

- Each ESP200 uses 4 QFP ASICs to achieve performance
- Each QFP is associated with a subset of the SPA-bays and interfaces
- SIP40 linecards may be split amongst multiple QFPs
  - SIP10 linecards will be serviced entirely by the leftmost QFP indicated per slot
- Should be taken into account for
  - QoS
  - Multicast
  - NAT



Egress queuing for interfaces handled by

- QFP0
- QFP1
- QFP2
- QFP3

ESP100 / ESP200 – In-service HW upgrade

- Support for ESP40 to ESP 100 upgrade provided the following criteria are met
  - Old image already supports ESP100 (XE 3.7 or later)
  - ESP40 config has less than 4000 QoS schedules per hierarchy
  - ESP40 config has less than 116K QoS queues on those interfaces associated with a single QFP on ESP100 or ESP200
    - Typically only ever a risk in Broadband configurations
    - Can be mitigated by spreading the queues across multiple interfaces distributed in the chassis
- Downgrading
  - Need to ensure that the ESP100 config does not exceed the scaling limits of ESP40 in any respect

For your reference

TCAM Uses

Definition	Ternary Content-Addressable Memory is designed for rapid, hardware-based table lookups of Layer 3 and Layer 4 information. In the TCAM, a single lookup provides all Layer 2 and Layer 3 forwarding information.
Which ASR 1000 features use TCAM?	• Security Access Control Lists (ACL) • Firewall • IPSec • Ethernet Flow Point for Ethernet Virtual Circuits • Flexible Packet Matching • Lawful Intercept • Local Packet Transport Services (LPTS) • Multi Topology Routing • NAT • Policy Based Routing • QoS • NBAR / SCEASR • Web Cache Control Protocol • Edge Switching Services • Event Monitoring

SIP and ESP combinations in modular chassis

Available ESI links

ESP-200

100	40	40
100	40	40
100	40	40

ESP-100

100	40	40
100	40	40
100	40	40

40 = 20 + 20

Any slot that is 40 compatible actually has two 20 links. One serves the right side SPA and the other serves the left side SPAs.

ESP-200

ESP-100

ASR1009-X

40	or	100	200
40	or	100	200
40	or	100	200

Future support

ASR1006-X

40	or	100	200
40	or	100	200

ESP-200

ESP-100

Max speed per slot in modular chassis

Available ESI links

ESP-200

100	40	40
100	40	40
100	40	40

ESP-40

40
40
40
40
40
40

ESP-100

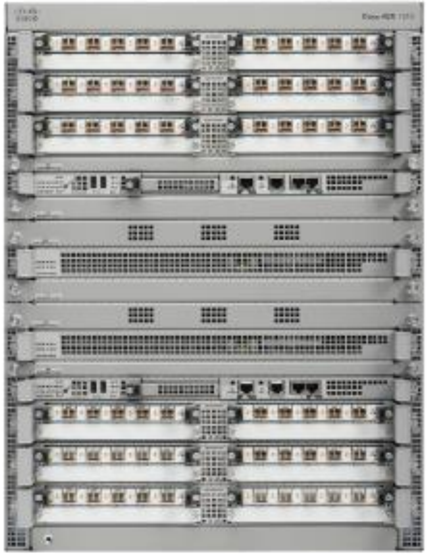
100	40	40
100	40	40
100	40	40

ESP-20

ESP-20	40	40	40

Any slot that is 40 compatible will also support 10 with the SIP-10 card which has been EoS. Only for non-X chassis though. No 10 support in -X chassis

ASR1013



40

40

40 or 100

ESP-200

ESP-100

ESP-40

40 or 100

40

40

ASR1006

40

40

40

ESP-100

ESP-40

ESP-20

ASR1004

40

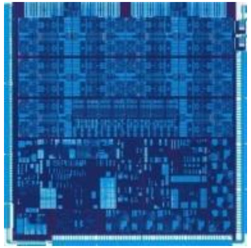
40

ESP-40

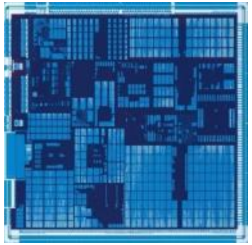
ESP-20

Quantum Flow Processor – ASR 1000 innovation

- Five year design and continued evolution – developing the 3rd generation
- Massively parallel: 64 cores, 4 threads per core for 256 packets in flight
- QFP Architecture designed to scale to beyond 100Gbit/sec
- High-priority traffic path throughout forwarding processing
- Packet replication capabilities for Lawful Intercept
- Full visibility of entire L2 frame
- Latency: tens of microseconds with features enabled
- Interfaces on-chip for external cryptographic engine
- 2nd generation QFP is capable of 70Gbit/sec, 32Mpps processing
- Can cascade 1, 2 or 4 chips to build higher capacity ESPs



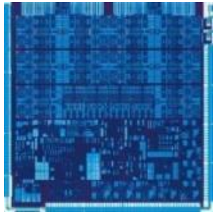
Cisco QFP
Packet Processor

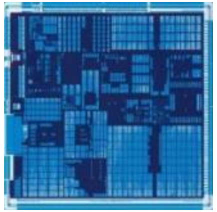


Cisco QFP
Traffic Manager

Cisco Quantum Flow Processor

custom versus off the shelf



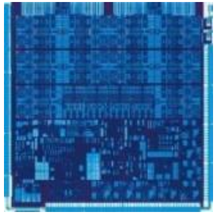


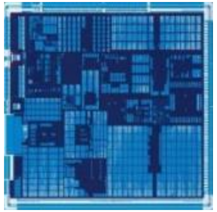
- Custom design needed for next-gen Network Integrated Services
  - Existing CPUs do not offer forwarding power required
  - Architecture of general purpose CPUs relies on larger memory caches (64B/128B) which is inefficient for network features
- QFP uses 16 byte memory access
  - minimizes wasted memory reads and increases memory access
  - for the same raw memory bandwidth, a 16B read allows 4-8 times the number of memory accesses/sec as a CPU using 64/128B accesses

- Preserves C-language programming support
  - Including stacking for nested procedures
  - Differentiator as compared to NPUs
  - Key to feature velocity
  - Support for portable, large-scale development
- Add hardware assists to further boost performance
  - TCAM, PLU, HMR…
  - Trade-off power requirement vs. board space

Cisco Quantum Flow Processor

2nd generation details





- Used in ASR1002-X, ESP100 & ESP200
- 2nd gen QFP integrates both the PPE engine and the Traffic manager into a single ASIC
- 64 PPEs per 2nd gen QFP
  40 PPE for 1st gen QFP
- 116K queues per 2nd gen QFP
  128K queues for 1st gen QFP
- Can be used in a matrix of 2 or 4
  - ESP100 has 232K queues
  - ESP200 has 464K queues

- 1st and 2nd gen QFPs run the same code
- Maintains identical feature behavior between QFP hardware releases
  - Full configuration consistency
  - Identical feature behavior (NAT, FW, etc)
- In-service hardware upgrade from ESP40 to ESP100 supported
- Differences
  - Minor behavioral show-command differences
  - Deployment differences in deployments with large number of BQS schedules

For your reference

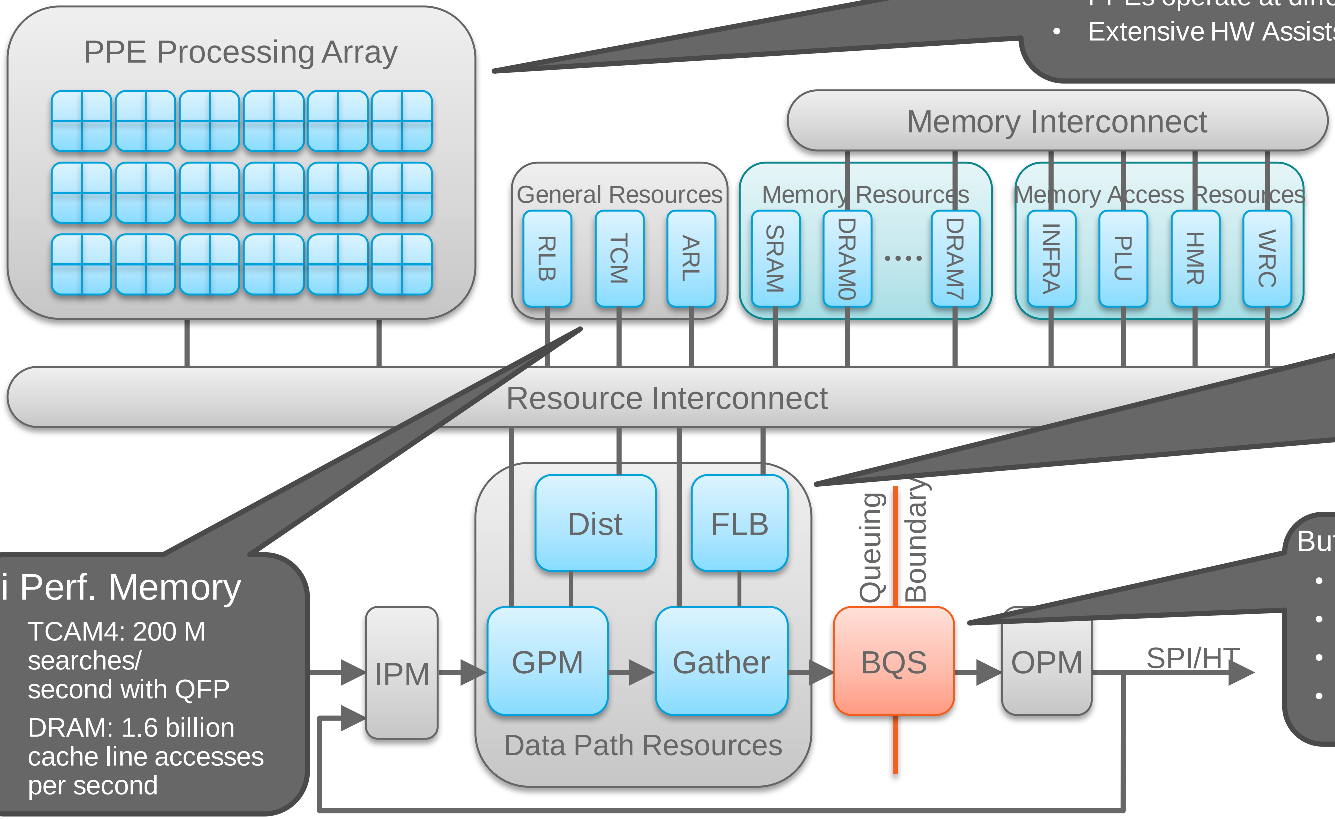
Quantum Flow Processor Video

http://www.cisco.com/cdc_content_elements/flash/netsol/sp/quantum_flow/demo.html

AR1000 QFP Architecture

40 PPEs (1st Gen); 64 PPEs (2nd Gen)

- Tensilica (MIPS-like) instruction set architecture
- Data cache (1KB per thread, 16B cache line)
- Four HW threads per PPE
- PPEs operate at different speeds on various ESPs
- Extensive HW Assists: ACL, TBM-lookup, WRED, Flow Locks



Distributor Assigns Each Packet to a PPE/Context

- QFP is not doing flow-based load-balancing among processors
- Distribution is to any eligible PPE/Context
- Hardware locks for ordering and mutual exclusion

Buffering, Queuing, & Scheduling (BQS)

- HQF/MQC compatible
- 128K queues
- Flexible allocation of schedule resources
- 5+ levels of scheduling hierarchy

Hi Perf. Memory

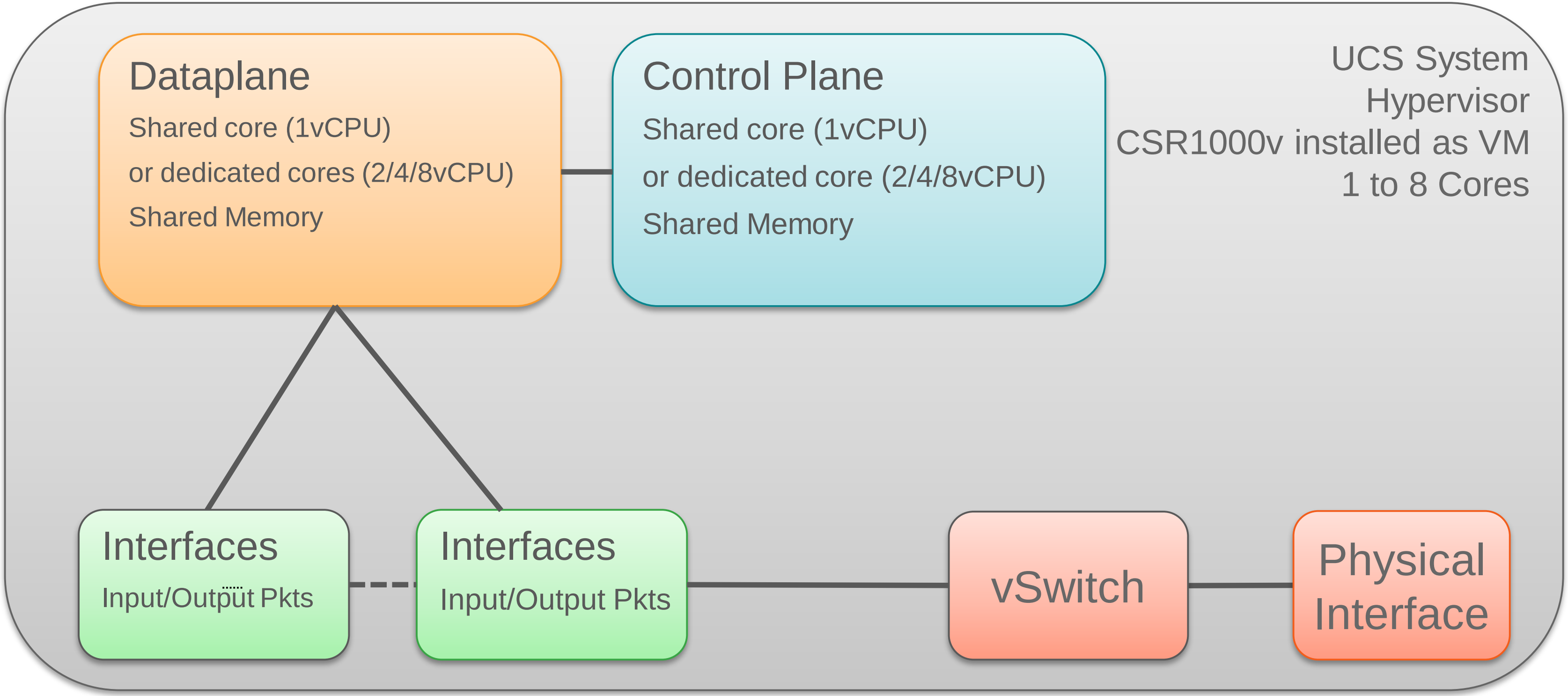
- TCAM4: 200 M searches/ second with QFP
- DRAM: 1.6 billion cache line accesses per second

QFP Hardware Assists

- RLB = Regular Lock Block
- TCM = TCAM Controller
- ARL = ACL Range Lookup
- INFRA = DMA Engine, HT access, CSR access
- PLU = Pointer Lookup Unit (Tree Bitmap lookup)
- HMR = Hash Mod Read
- WRC = Weighted RED Controller
- Gather = gathers fragments
- FLB = Flow lock block
  - Packets are given internal ID based on source / destination interface, packet header fields etc
  - ID then used internally to ensure packet sequencing

System Architecture – CSR 1000V

CSR1000V



Dataplane

Shared core (1vCPU)

or dedicated cores (2/4/8vCPU)

Shared Memory

Control Plane

Shared core (1vCPU)

or dedicated core (2/4/8vCPU)

Shared Memory

UCS System
Hypervisor
CSR1000v installed as VM
1 to 8 Cores

Interfaces

Input/Output Pkts

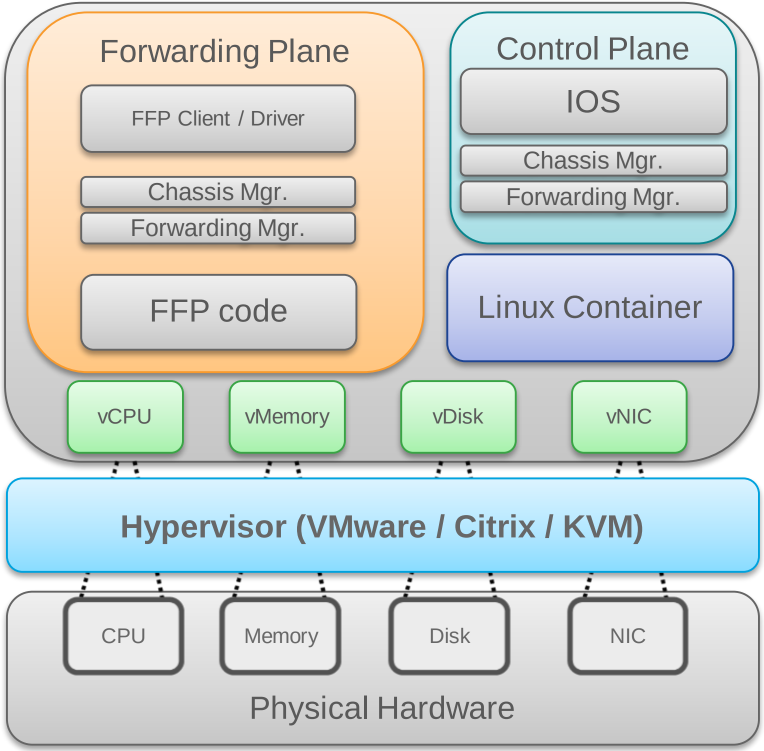
Interfaces

Input/Output Pkts

vSwitch

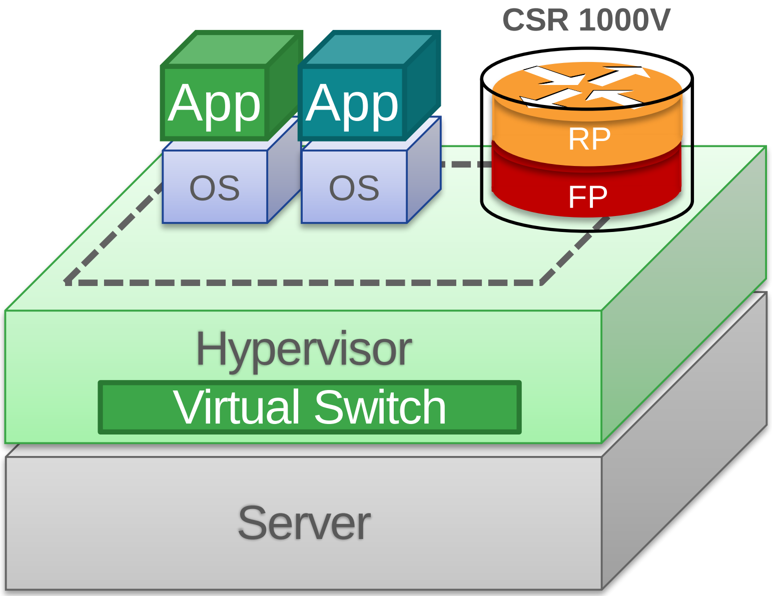
Physical Interface

CSR 1000V - virtualized IOS XE



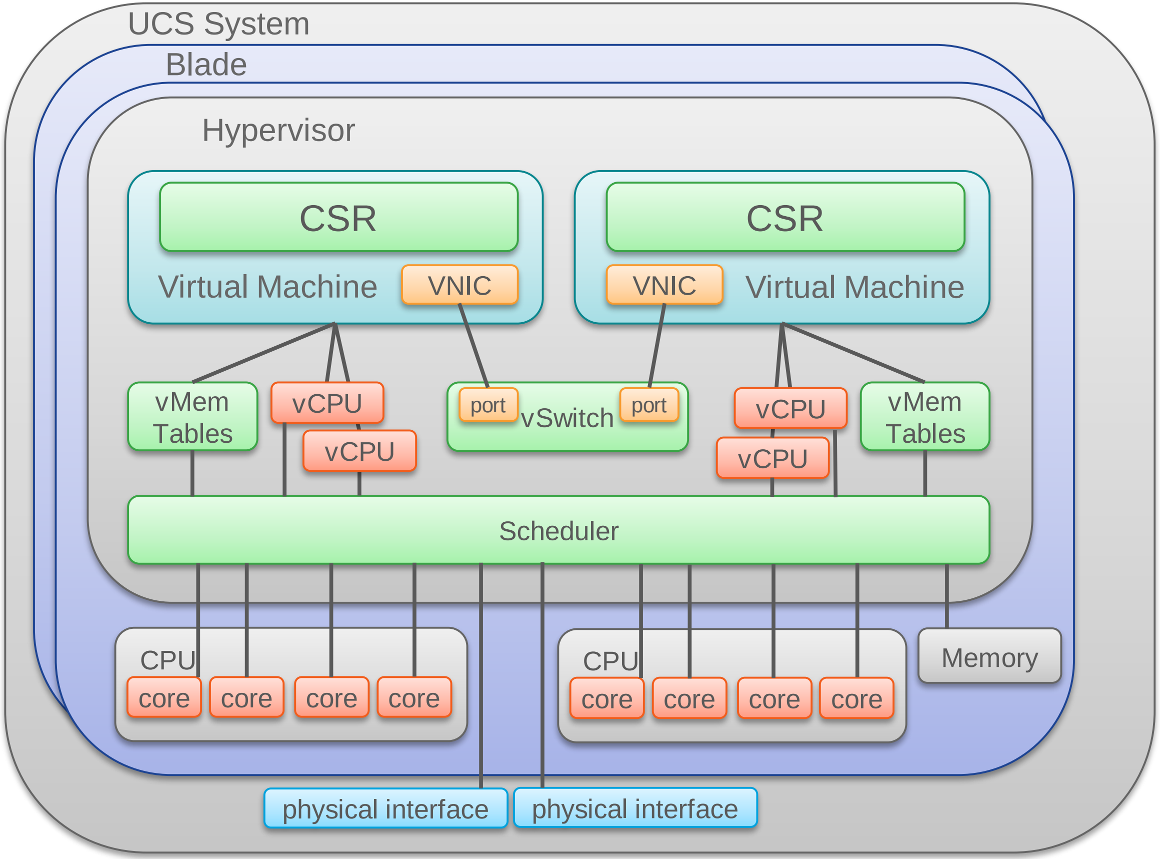
- Virtualized IOS XE
  - Generalized to work on any x86 system
  - Hardware specifics abstracted through a virtualization layer
  - Forwarding (ESP) and Control (RP) mapped to vCPUs
  - Bootflash: NVRAM: are mapped into memory from hard disk
  - Boot loader functions implemented by GRUB
- Limitations
  - No dedicated crypto engine – we leverage the Intel AES-NI instruction set to provide hardware crypto assist.
  - No QFP – lower forwarding performance
  - No HW Accelerators – Less efficient feature processing

CSR 1000V - virtualized IOS XE



- IOS XE Cloud Edition
  - Select IOS XE Features based on use case
- Infrastructure Agnostic
  - Supports any x86 server or vSwitch
  - Runs on ESXi, KVM, Hyper-V, Xen, Amazon AWS, Microsoft Azure
- Throughput Elasticity
  - Delivers 10Mbps to 20Gbps performance
- Multiple Licensing Models
  - Term, Perpetual, Usage
- Programmability
  - RESTful API for automated management

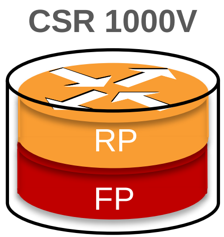
CSR 1000V – Hypervisor Interactions



- Hypervisor abstracts and shares physical hardware resources across multiple VMs
- Scheduling of vCPU onto physical cores can create non-deterministic behavior
- Scheduling of vNICs onto physical ports can lead to packet loss and jitter
- ESXi scheduler spreads the load across all physical cores intelligently according to a proportional share-based algorithm

CSR 1000V – vCPU Allocation

	Control Plane	Data Plane
# vCPUs	Virtual Route Processor	Virtual Forwarding Processor
1	1	1
2	1	2
3	1	2-3
4	1	2-4
5	1	2-5
6	1	2-6
7	1	2-7
8	1	2-8



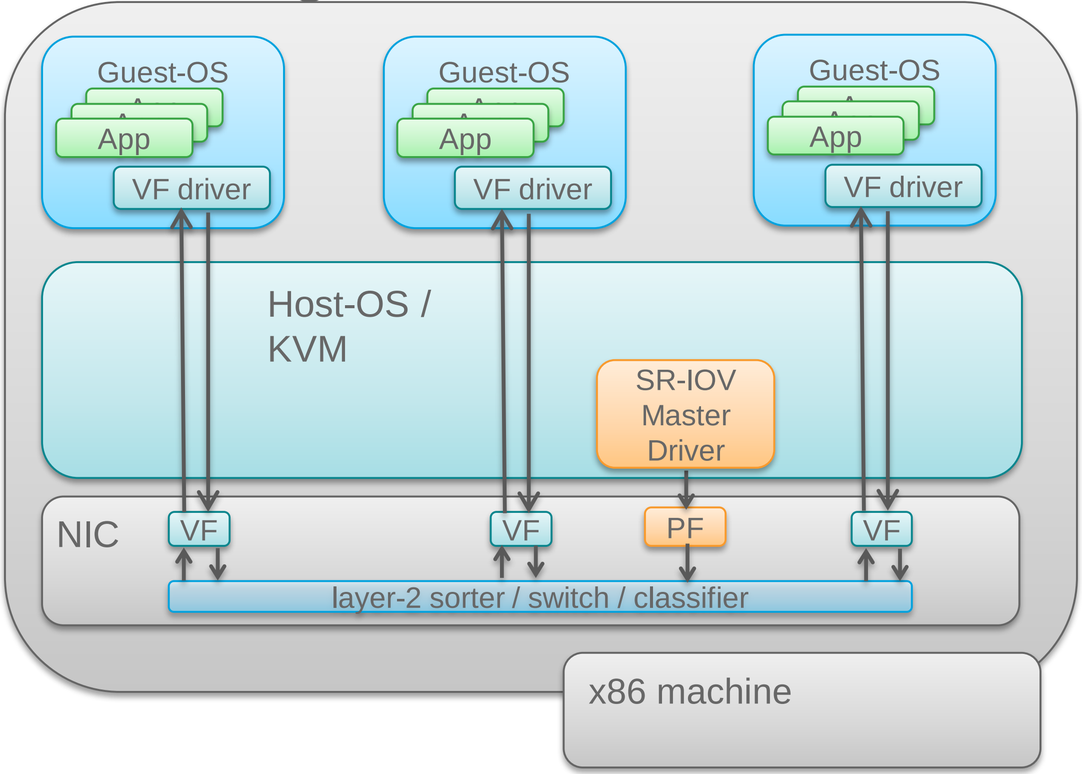
- Separation of control-plane and data-plane
- vCPU allocation is static and done during bootup

CSR 1000V – Network I/O

Method	Driver	Performance	Pros/Cons	Supported
Emulated	E1000	Low	• Wide compatibility • Worst Performance	NO
Para-virtualized	VMXNET3 VirtIO	Excellent	• Virtualization Aware • High degree of interaction between guest OS and hypervisor	Yes (Default)
Pass-through	NIC Dependent	Best	• Direct access to HW – High I/O • Lose virtualization features such as vMotion	Yes – only Intel NICs (ixgbevf / ixgbe drivers)

CSR 1000V – Network I/O Optimization SR-IOV with PCIe Pass-Through

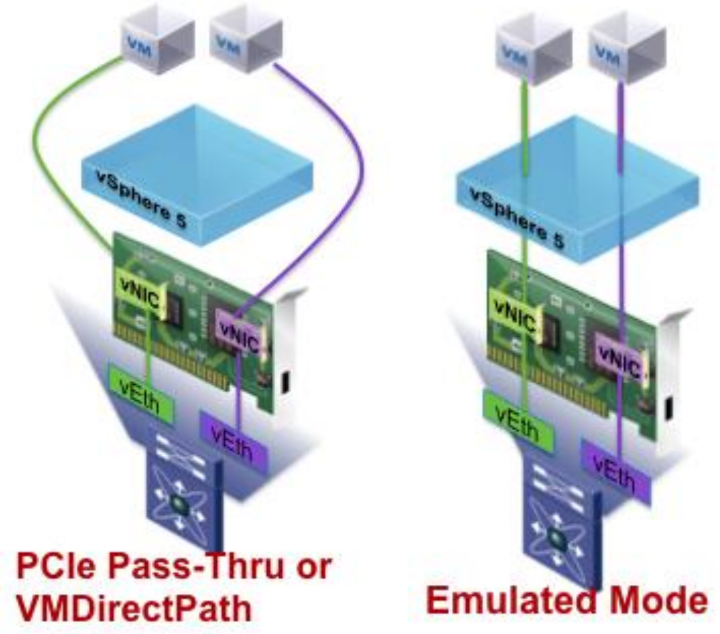
- Allows a single PCIe device to appear to be multiple separate devices (NIC Supports Virtualization)
- Network traffic bypasses software switch layers
- Creates physical and virtual functions:
  - PF: Controls sorter
  - VF: Passes packets
- Requires support in BIOS/Hypervisor
  - Intel VT-D / AMD IOMMU
- Only supported on Intel NICs



CSR 1000V – Network I/O Optimization

UCS VM-FEX

- UCS VM-FEX provides dedicated hardware resources to each VM
- vSwitch and hypervisor virtualization layers are bypassed
- Virtualization performed in hardware
- Supports DirectPath or Emulated mode
- Support for vMotion
- Requires dedicated cards (eg. VIC1280)



System Architecture – ISR 4000

Introducing the Cisco ISR 4000 Family

Enabling Branch Services for the 21st Century Network

Delivering the Ultimate Application Experience Over Any Connection

Revolutionary Architecture

- 4-10 times faster, at the same price
- Deterministic performance with services
- Pay as you grow
- Virtualized network function

Service Innovation

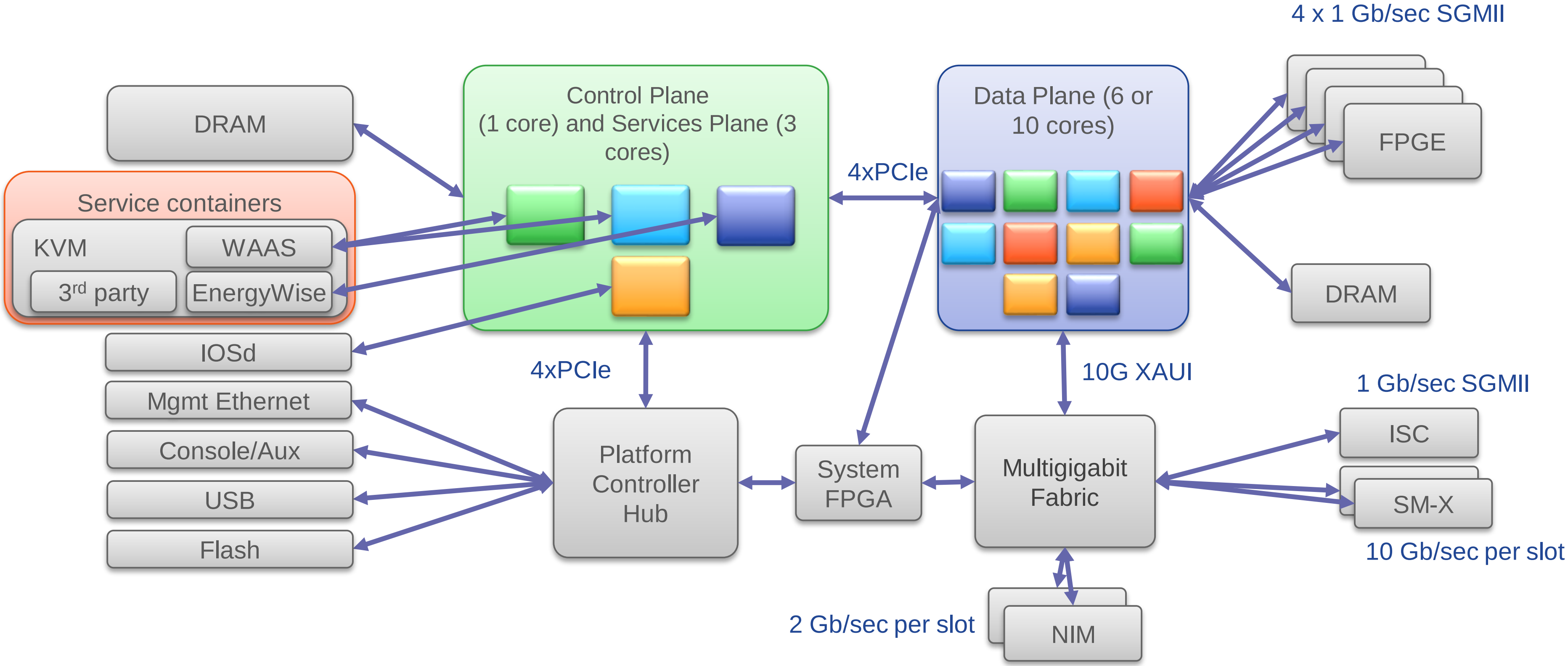
- Native Layer 2 – 7 services
- Converged network, compute, storage
- Simple, scalable WAN path control
- Best-of-breed security: Sourcefire® IDS

ISR4000 system specification

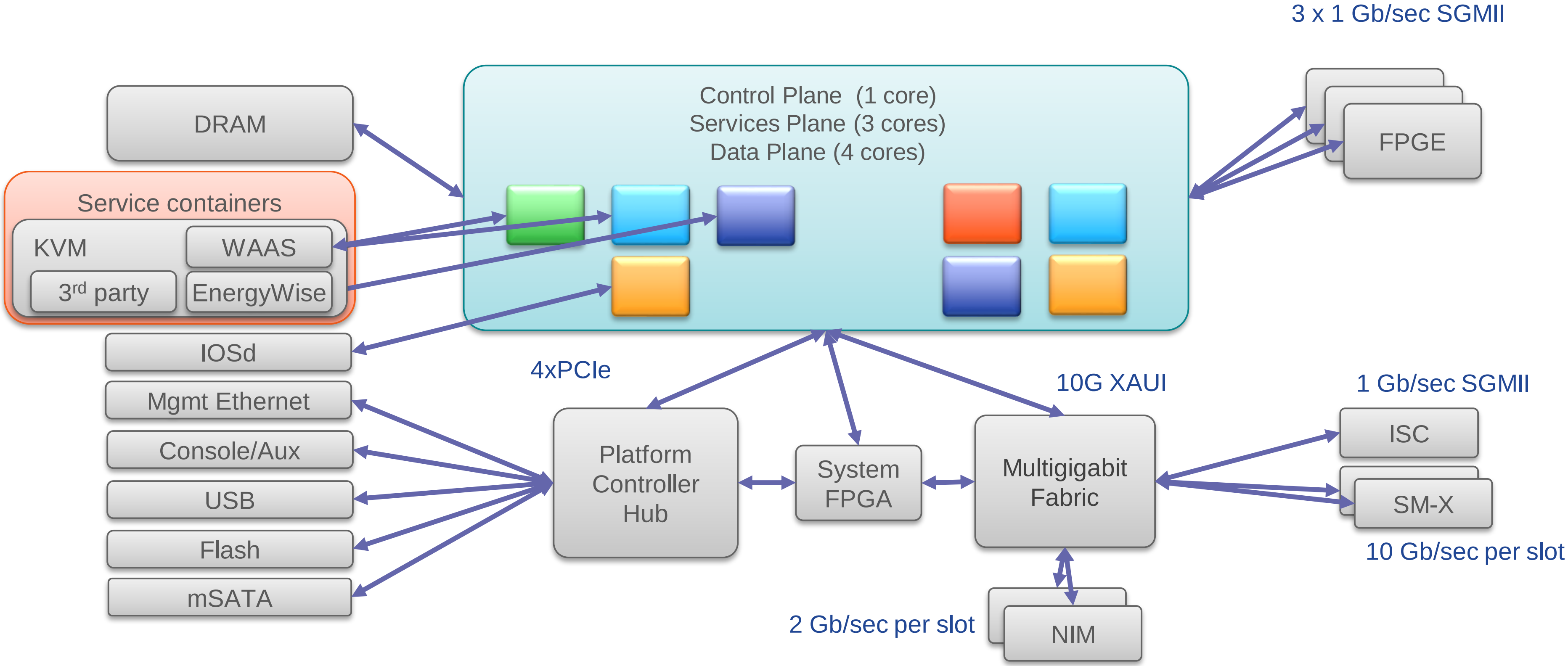
	4221	4321	4331	4351	4431	4451
CPU architecture	2 Core	4 Core	8 Core	8 Core	4 Core CP/SP 6 Core DP	4 Core CP/SP 10 Core DP
NIM slots	2	2	2	3	3	3
SM-X slots	0	0	1	2	0	2
Front-Panel Ethernet	1 * dual Phy 1 RJ45	1 * dual Phy 1 RJ45	1 * dual Phy 1 RJ45 1 SFP	3 dual Phy	4 * dual Phy	4 * dual Phy
Performance (default / max)	35 / 75 Mbit/s	50 / 100 Mbit/s	100 / 300 Mbit/s	200 / 400 Mbit/s	500 / 1000 Mbit/s	1000 / 2000 Mbit/s
Power Supplies	One External AC	One External AC	One Internal AC	One Internal AC/DC	Dual Internal AC or DC	Dual Internal AC or DC
Default / maximum DRAM	4 GB shared	4 / 8 GB shared	4 / 16 GB shared	4 / 16 GB shared	4 / 16 GB for CP/SP 2 GB for DP	4 / 16 GB for CP/SP 2 GB for DP
Default / maximum Flash	8 GB	4 / 8 GB	4 / 16 GB	4 / 16 GB	8 / 32 GB	8 / 32 GB
Management Ethernet	1 Gbit/s	1 Gbit/s	1 Gbit/s	1 Gbit/s	1 Gbit/s	1 Gbit/s

CP= Control Plane, DP = Data Plane, SP = Services Plane

Cisco ISR 4400 Architecture



Cisco ISR 4200 and 4300 Architecture



Note: 4321 uses 2DP, 1CP & 1SP cores
4221 uses 1DP and 1CP cores

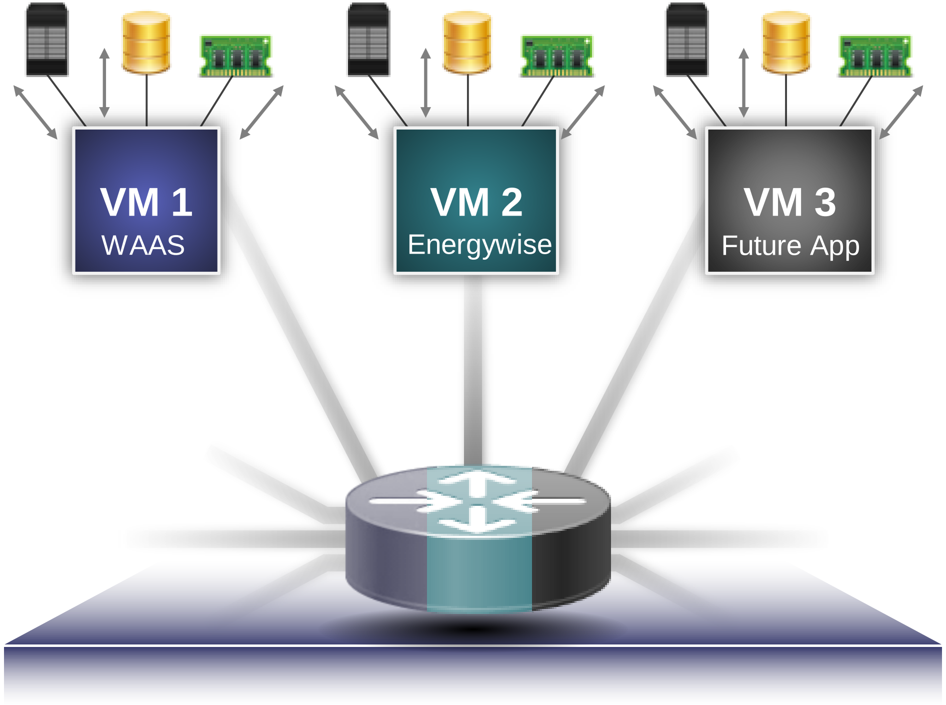
Service Virtualization for networking

Service Containers

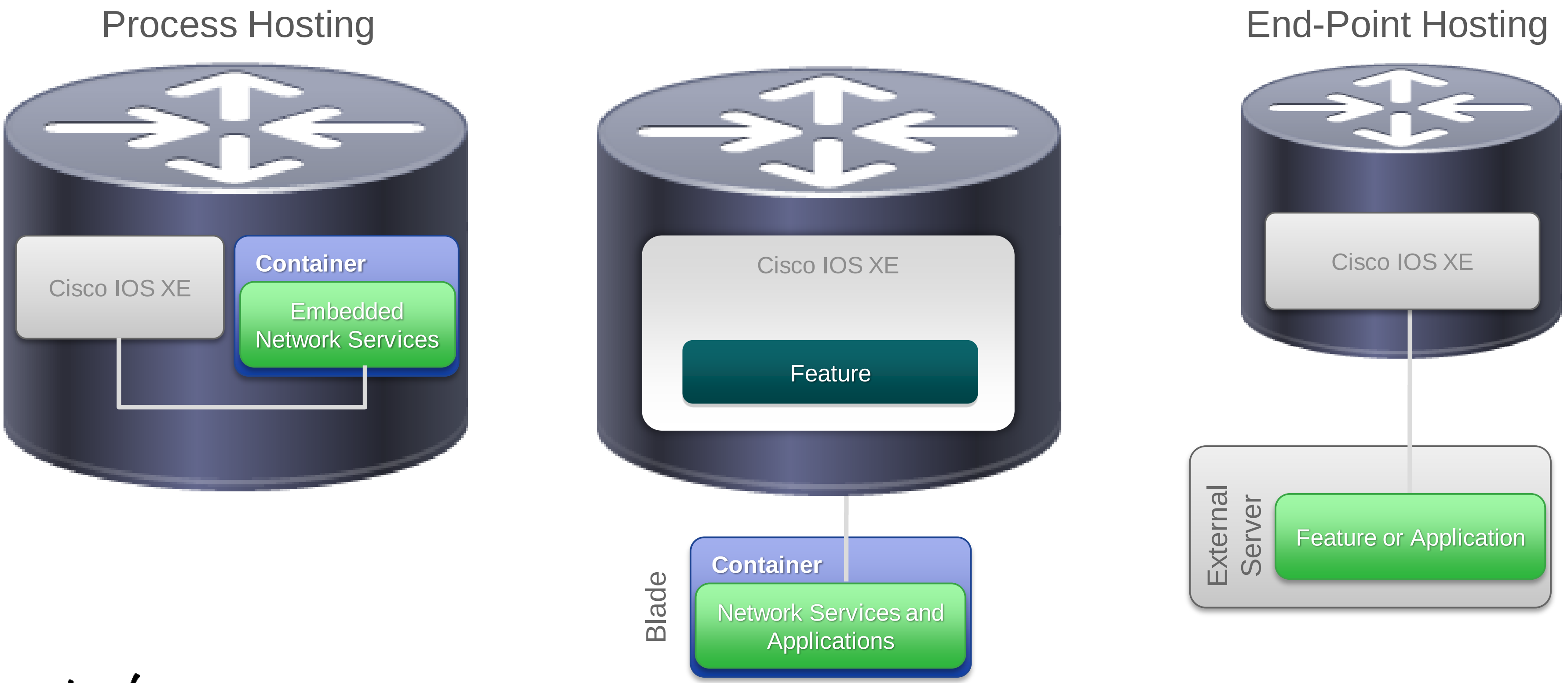
- Dedicated virtualized compute resources
- CPU, disk, memory for each service
- Easily repurpose resources
- Industry-standard hypervisor

Benefits

- Better performing network services
- Ease of deployment with zero footprint; no truck roll
- Greater security through fault isolation
- High reliability
- Flexibility to upgrade network services independent of router IOS® Software



Future service virtualization



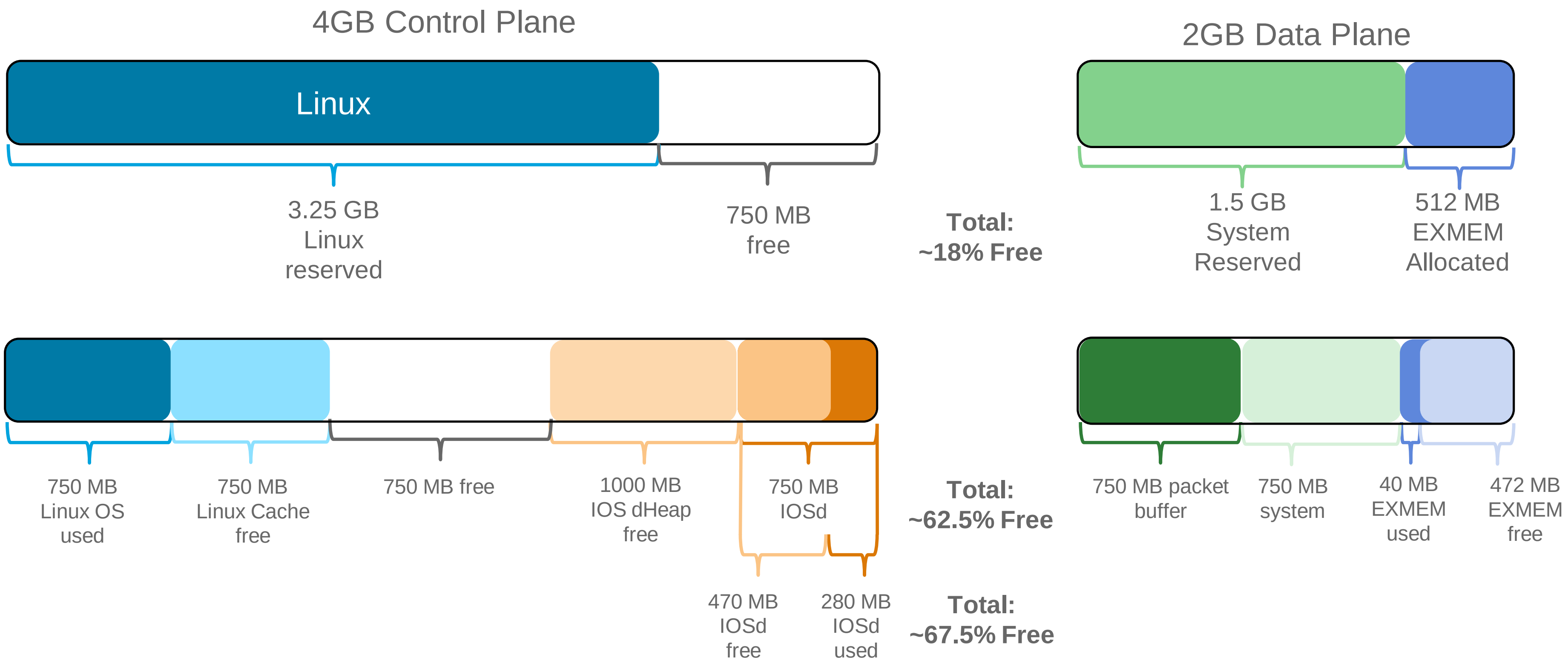
Maximum interface termination

	ISR4221	ISR4321	ISR4331	ISR4351	ISR4431	ISR4451	Comment
SM-X (single width)	0	0	1	2	0	2	
NIMs (single width)	2	2	2 (3)	3 (5)	3	3 (5)	With SM-X-NIM-ADPTR each (SM-X = NIM)
10GE Routed	0	0	1	2	0	2	With SM-X-4X1G-1X10G
1 GE Routed	2+4 = 6	2+4 = 6	3+10 = 13	3+18 = 21	4+6 = 10	4+18 = 22	With onboard + SM-X-6X1G and/or NIM-2GE-CU-SFP
1 GE Switched	16	16	40	72	24	72	With NIM-ES2-8-P & SM-X-ES3-24-P
T3/E3 Clear Channel	0	0	1	2	0	2	With SM-X-1T3/E3
T1/E1 Clear Channel	8	8	24	40	24	40	With NIM-8MFT-T1/E1 or NIM-4MFT-T1/E1 (4321)
T1/E1 Channelized	4	4	24	40	24	40	With NIM-8CE1T1-PR or NIM-2CE1T1-PR (4321)
FXS	-	8	12	20	12	20	With NIM-4FXS
FXO	-	8	12	20	12	20	With NIM-4FXO
Serial	4	4	12	20	12	20	With NIM-4T or NIM-2T (4321)
VA DSL	2	2	3	5	3	5	With NIM-VAB-A or NIM-VA-B or NIM-VAB-M

DRAM demystified

ISR 4400, 4GB CP, 2GB DP, IOS-XE 3.13.1

ISR 4400 – Memory allocation



4GB Control Plane

Linux

3.25 GB Linux reserved

750 MB free

Total: ~18% Free

750 MB Linux OS used

750 MB Linux Cache free

750 MB free

1000 MB IOS dHeap free

750 MB IOSd

470 MB IOSd free

280 MB IOSd used

Total: ~62.5% Free

Total: ~67.5% Free

2GB Data Plane

1.5 GB System Reserved

512 MB EXMEM Allocated

750 MB packet buffer

750 MB system

40 MB EXMEM used

472 MB EXMEM free

What is DP and CP memory used for?

- Control Plane Memory
  - Used for IOS daemon
    - This daemon holds the IOS system as well Control Plane Tables (i.e. Routing Information Base)
  - Used for Linux
    - This manages the entire device and also allocates memory to service containers
    - The linux portion grows when IOS is growing due to information replication into other processes

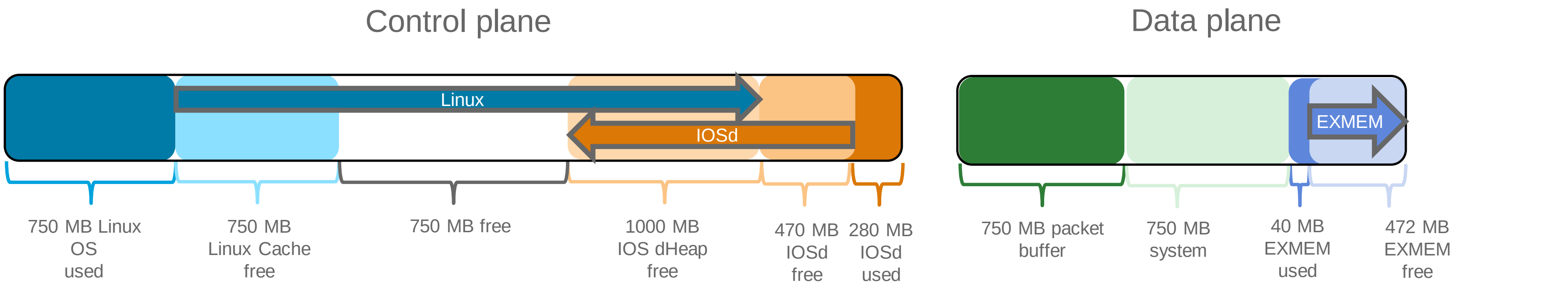
- Data Plane Memory:
  - Used exclusively by IOS for data plane services
    - Packet Buffering
    - System internal processes
    - EX Memory, this grows when scalable features are used (Forwarding Information Base, NAT Table etc.)

    - * Allocation will vary by IOS-XE release

ISR 4400, 4GB CP, 2GB DP, IOS-XE 3.13.1

ISR 4400 – Memory allocation

ISR 4400, 4GB CP, 2 GB DP, IOS-XE 3.13.1



ISR 4400, 4GB CP, 2GB DP, IOS-XE 3.13.1

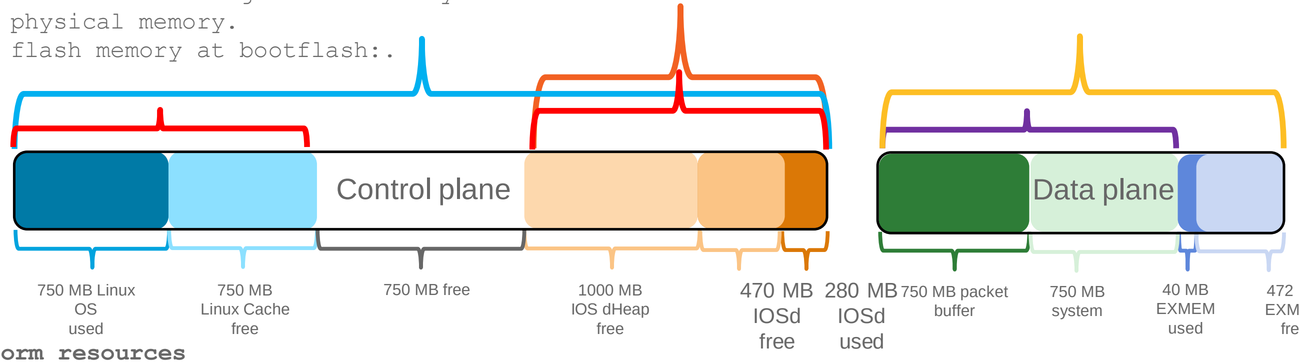
ISR 4400 – How to monitor CP and DP

```
ISR4451#show version
<snip>
System image file is "bootflash:/isr4400-universalk9.03.13.01.S.154-3.S1-ext.SPA.bin"
<snip>
cisco ISR4451-X/K9 (2RU) processor with 1687854K/6147K bytes of memory.
Processor board ID FGL165210MU
4 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
4194304K bytes of physical memory.
7393215K bytes of flash memory at bootflash:.
```

Reserved IOS Memory

Total CP Memory

Total Flash Memory



```
ISR4451#show platform resources
**State Acronym: H - Healthy, W - Warning, C - Critical
Resource                 Usage              Max            Warning        Critical       State
-----------------------------------------------------------------------------------------------
RP0 (ok, active)                                                                         H
 Control Processor       2.40%              100%           90%            95%            H
  DRAM                   3180MB(82%)        3878MB         90%            95%            H
ESP0(ok, active)                                                                         H
 QFP                                                                                     H
  DRAM                   1589776KB(75%)     2097152KB      80%            90%            H
  IRAM                   0KB(0%)            0KB            80%            90%            H
```

Reserved CP Memory

Usable CP Memory

Reserved DP Memory

Total DP Memory

ISR 4400, 4GB CP, 2GB DP, IOS-XE 3.13.1

ISR 4400 – How to monitor CP

```
ISR4451#show memory
                    Head      Total(b)      Used(b)       Free(b)   Lowest(b)   Largest(b)
 Address           Bytes          Prev     Next Ref         PrevF       NextF     Alloc PC
Processor  7F4A5B545010    1728363504    284041616    1444321888   679710664   1048575908
 lsmpi_io  7F4A5AE431A8       6295128      6294304           824         824          412
Dynamic heap limit(MB) 1000           Use(MB)  0
```

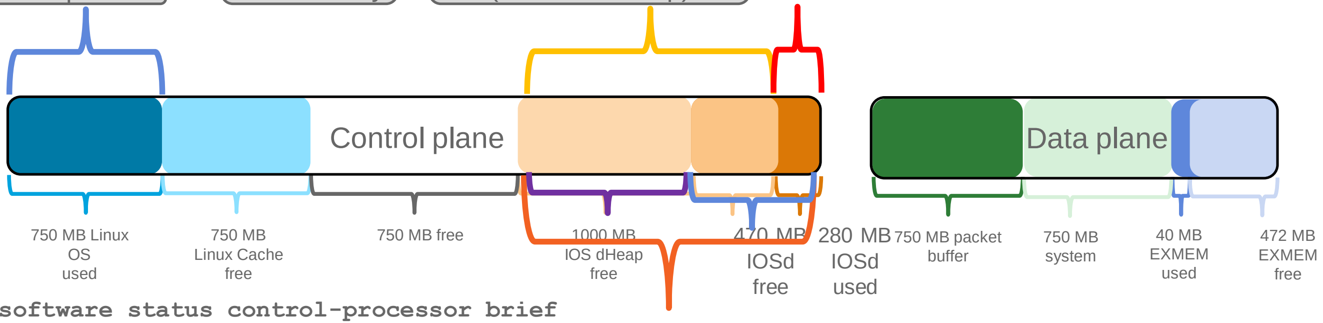
Total available dHeap

Total reserved IOS Memory (includes dHeap)

dHeap used

Total used IOS Memory

Total free IOS Memory (includes dHeap)



```
ISR4451#show platform software status control-processor brief
Load Average
 Slot  Status  1-Min  5-Min 15-Min
  RP0 Healthy   0.00   0.04   0.06

Memory (kB)
 Slot  Status    Total     Used (Pct)     Free (Pct) Committed (Pct)
  RP0 Healthy  3972052  3259444 (82%)   712608 (18%)   1506452 (38%)

CPU Utilization
 Slot  CPU   User System   Nice   Idle    IRQ   SIRQ IOwait
  RP0    0   2.39   0.39   0.00  97.00   0.09   0.09   0.00
```

Total used Memory (excludes Cache & dHeap, includes full 750 MB IOS)

ISR 4400, 4GB CP, 2GB DP, IOS-XE 3.13.1

ISR 4400 – How to monitor DP

```
ISR4451#show platform hardware qfp active infrastructure exmem statistics
QFP exmem statistics

Type: Name: DRAM, QFP: 0
  Total: 2147483648
  InUse: 1648148480
  Free:   499335168
  Lowest free water mark: 1[illegible]488448
<snip>
```

Total Physical DP Memory

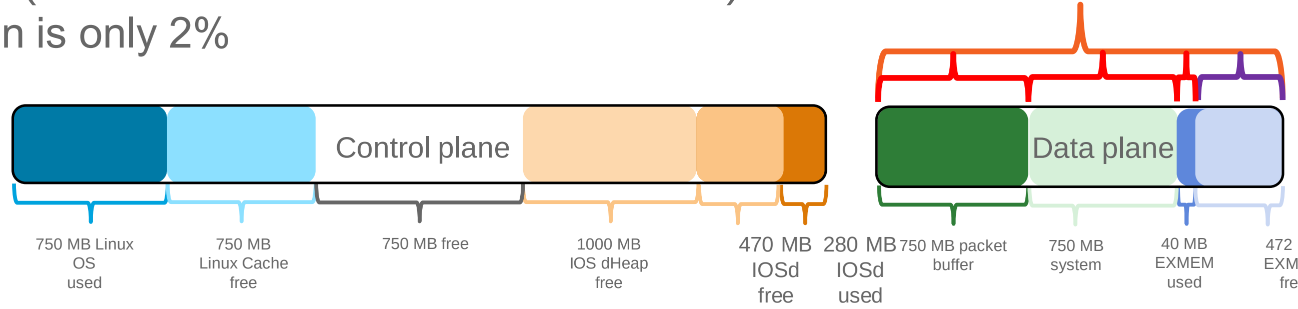
DP Memory used by System (750 MB), Buffer (756MB) and EX (~20MB)

Free DP memory (used by EX only!)

75% of memory appear to be used

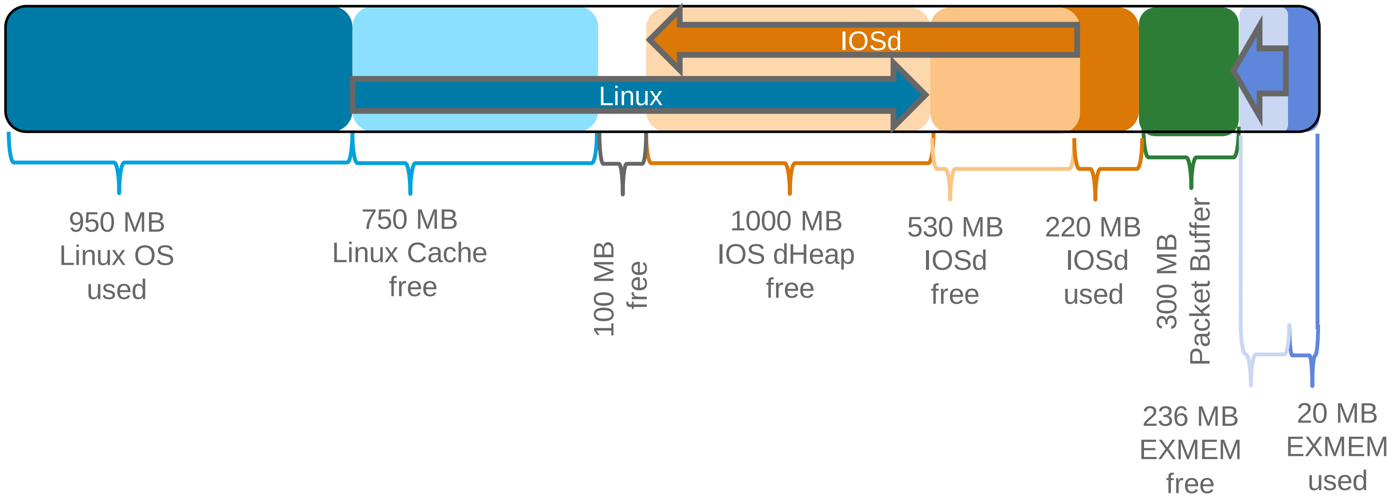
These are reserved for packet buffers and system internals

The EX part (that scales with features like the RIB) has 499 MB free out of 512 MB, hence it’s utilization is only 2%



ISR 4200 / 4300 – Memory allocation

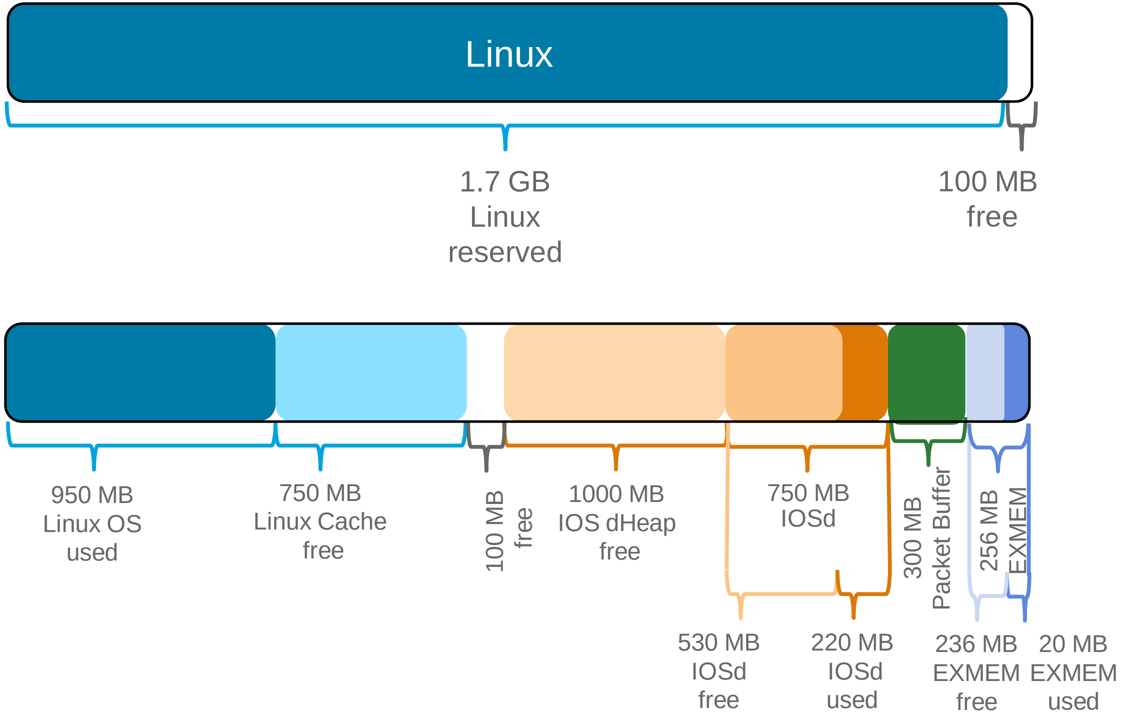
ISR4300, 4GB CP & DP, IOS-XE 3.13.1



ISR 4300, 4GB CP & DP, IOS-XE 3.13.1

ISR 4200 / 4300 – Memory allocation

4GB Control & Data Plane



Linux

1.7 GB Linux reserved

100 MB free

950 MB Linux OS used

750 MB Linux Cache free

100 MB free

1000 MB IOS dHeap free

750 MB IOSd

300 MB Packet Buffer

256 MB EXMEM

530 MB IOSd free

220 MB IOSd used

236 MB EXMEM free

20 MB EXMEM used

Total: ~2% Free

Total: ~42% Free

Total: ~ 65% Free

Cisco live!

ASR 1002-X, 16GB CP, 2GB DP, IOS-XE 3.13.1

ASR1002-X – How to monitor CP

```
ASR1002-X#show memory
                Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor  7F0615A74010   4840049792   555705528   4284344264   4284328728   4284338796
 lsmpi_io  7F06152711A8      6295128     6294304          824          824          412
```

Total reserved IOS Memory

Total used IOS Memory

Total free IOS Memory (includes dHeap)

```
ASR1002-X#show platform software status control-processor brief
Load Average
 Slot  Status  1-Min  5-Min 15-Min
  RP0 Healthy   0.00   0.00   0.00

Memory (kB)
 Slot  Status    Total     Used (Pct)     Free (Pct) Committed (Pct)
  RP0 Healthy 16337120  3547568 (22%) 12789552 (78%)   6015204 (37%)

CPU Utilization
 Slot  CPU   User System   Nice   Idle    IRQ   SIRQ IOwait
  RP0    0   2.29   3.19   0.00  94.40   0.00   0.09   0.00
         1   0.80   0.90   0.00  98.30   0.00   0.00   0.00
         2   0.20   0.50   0.00  99.30   0.00   0.00   0.00
         3   0.19   0.39   0.00  99.40   0.00   0.00   0.00
```

Total used Memory (*includes*Cache & dHeap, includes full 750 MB IOS)

The “`show memory`” command is executed inside IOSd, therefore it will only show what is available to the IOSd process.

ASR 1002-X, 16GB CP, 2GB DP, IOS-XE 3.13.1

ASR1002-X – How to monitor DP

```
ASR1002-X# show platform hardware qfp active infrastructure exmem statistics
QFP exmem statistics

Type: Name: DRAM, QFP: 0
  Total: 1073741824
  InUse: 205749248
  Free: 867992576
  Lowest free water mark: 867992576
<snip>
```

Total Physical DP Memory

DP Memory used

Free DP memory

Differences between Platforms

ISR4200 / ISR4300

- CP IOS monitoring:
  - IOS can grow into dHeap
- CP memory monitoring:
  - “used” shows allocated memory
  - “committed” shows used memory
- DP memory monitoring:
  - DP memory shows only EXMEM

ISR4400

- CP IOS monitoring:
  - IOS can grow into dHeap
- CP memory monitoring:
  - “used” shows allocated memory
  - “committed” shows used memory
- DP memory monitoring:
  - DP memory includes system, buffer and EXMEM

ASR1000

- CP IOS monitoring:
  - IOS has fixed allocation at boot, no growth into dHeap
- CP memory monitoring:
  - “used” shows used memory
  - “committed” shows allocated memory
- DP memory monitoring:
  - DP memory shows only EXMEM

ISR 4400, 4GB CP, 2GB DP, IOS-XE 3.13.1

Routing Scale Test

IPv4 BGP Routes	show platform resources		show memory			show platform software status control-processor brief			show plat hardware qfp active infra exmem statistics	
	Reserved CP	Reserved DP	Total used	Total Free	Heap Used	used	free	committed	InUse	Free
0	3233MB(83%)	1591MB(75%)	290MB	1411MB	0MB	3312MB (83%)	659MB (17%)	1506MB (38%)	1648MB	499MB
100000	3523MB(90%)	1617MB(77%)	431MB	1296MB	0MB	3603MB (91%)	368MB (9%)	1661MB (42%)	1656MB	490MB
200000	3819MB(98%)	1627MB(77%)	569MB	1158MB	0MB	3907MB (98%)	64MB (2%)	1813MB (46%)	1667MB	480MB
300000	3854MB(99%)	1636MB(78%)	707MB	1020MB	48MB	3945MB (99%)	26MB (1%)	1998MB (50%)	1675MB	472MB
400000	3779MB(97%)	1646MB(78%)	845MB	882MB	160MB	3870MB (97%)	101MB (3%)	2282MB (57%)	1685MB	461MB
500000	3851MB(99%)	1654MB(78%)	984MB	744MB	304MB	3943MB (99%)	28MB (1%)	2580MB (65%)	1694MB	453MB
600000	3853MB(99%)	1664MB(79%)	1122MB	606MB	448MB	3946MB (99%)	25MB (1%)	2882MB (73%)	1704MB	442MB
700000	3851MB(99%)	1674MB(79%)	1260MB	467MB	576MB	3943MB (99%)	28MB (1%)	3165MB (80%)	1713MB	434MB
800000	3850MB(99%)	1683MB(80%)	1398MB	330MB	688MB	3942MB (99%)	29MB (1%)	3430MB (86%)	1723MB	423MB
900000	Unsupported									
1000000										

Free Memory and Committed Memory should be monitored closely.

Other data is misleading due to the inclusion of heap and cache.

ISR 4300, 4GB CP & DP, IOS-XE 3.13.1

Routing Scale Test

IPv4 BGP Routes	show platform resources		show memory			show platform software status control-processor brief			show platform hardware qfp active infrastructure exmem statistics	
	Reserved CP	Reserved DP	Total used	Total Free	Heap Used	used	free	committed	InUse	Free
0	3773MB(97%)	22MB(8%)	229MB	1498MB	0MB	3888MB (98%)	61MB (2%)	2302MB (58%)	23MB	244MB
100000	3830MB(99%)	49MB(18%)	366MB	1362MB	0MB	3920MB (99%)	29MB (1%)	2457MB (62%)	50MB	218MB
200000	3830MB(99%)	59MB(22%)	507MB	1220MB	0MB	3920MB (99%)	29MB (1%)	2609MB (66%)	60MB	207MB
300000	3830MB(99%)	67MB(25%)	641MB	1087MB	0MB	3920MB (99%)	29MB (1%)	2762MB (70%)	69MB	199MB
400000	3829MB(99%)	77MB(29%)	782MB	946MB	112MB	3920MB (99%)	29MB (1%)	3030MB (77%)	79MB	188MB
500000	3828MB(99%)	86MB(33%)	919MB	808MB	240MB	3921MB (99%)	29MB (1%)	3313MB (84%)	88MB	179MB
600000	3828MB(99%)	96MB(36%)	1056MB	671MB	368MB	3921MB (99%)	29MB (1%)	3598604 (91%)	98MB	170MB
700000	Unsupported									
800000										
900000										

In comparison to 4400 the IOSd memory limit was probably not reached on this 4300.

The overall memory consumption identified by “committed memory” is the limitation.

Conclusion

- There are 3 possible memory bottlenecks:
- 1. IOSd Memory
  - Even including dHeap there is a limit to how big IOSd can grow
- 2. Overall Linux Memory
  - Because Linux grows at about the same rate as IOSd and reduces it’s cache constantly this absence of cache eventually becomes an issue
- 3. EXMEM (Data Plane)
  - This is unrelated to the control-plane memory but still can pose a limitation, especially as it can’t be increased as of current software

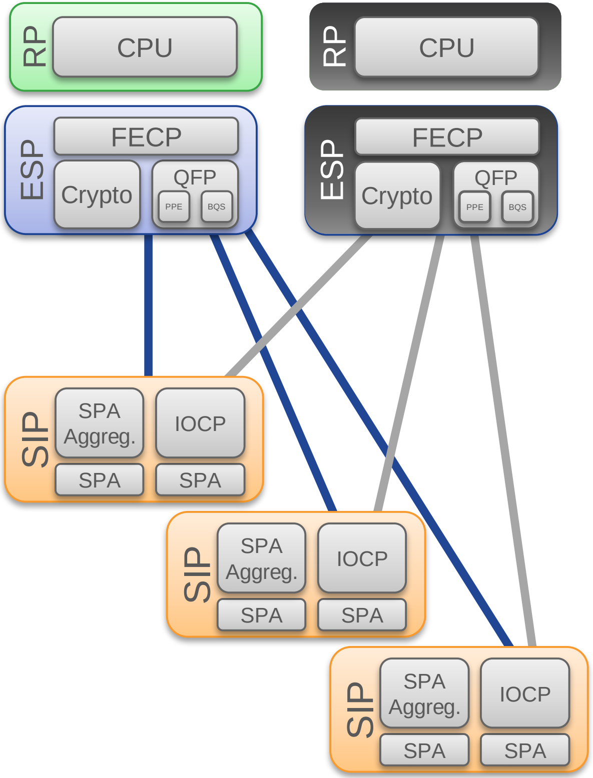
Scaling up with bigger memory (IOS-XE 3.13.1)

Platform	CP & DP Memory	Linux	IOS dHeap	IOS static	EXMEM	Service Containers
4400	4GB, 2GB	2.25 GB	1 GB	750 MB	512 MB	0 GB
4400	8GB, 2GB	4.25 GB	3 GB	750 MB	512 MB	4 GB
4400	16GB, 2GB	8.25 GB	7 GB	750 MB	512 MB	8 GB
4300	4GB	1.6 GB	1 GB	750 MB	256 MB	0 GB
4300	8GB	3.6 GB	3 GB	750 MB	256 MB	4 GB
4300	16GB	7.6 GB	7 GB	750 MB	256 MB	8 GB
ASR1002-X	16GB, 2GB	12 GB	part of fixed Linux memory	4GB	1024MB	0 GB

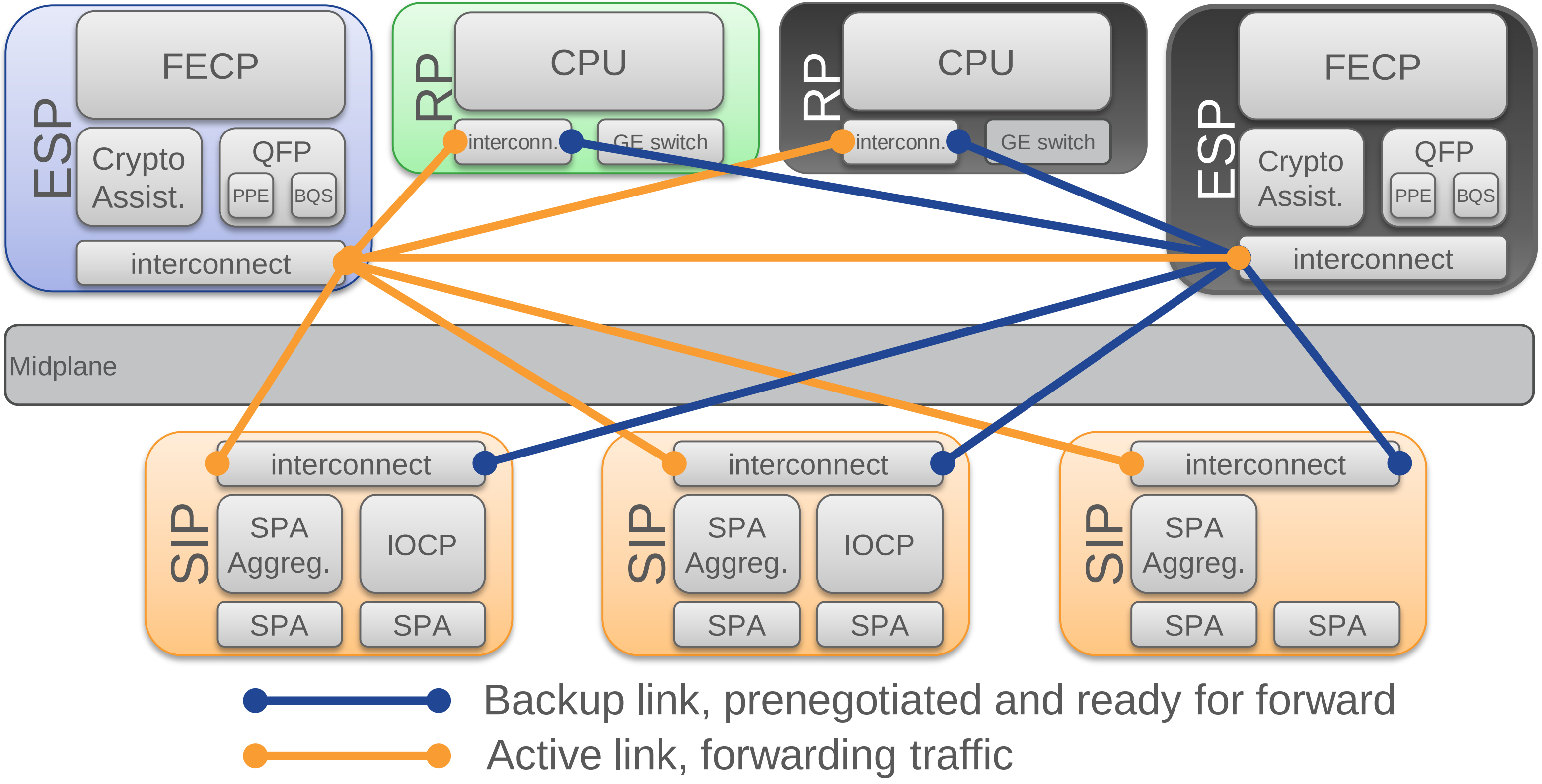
High Availability

High-Availability on the ASR 1000

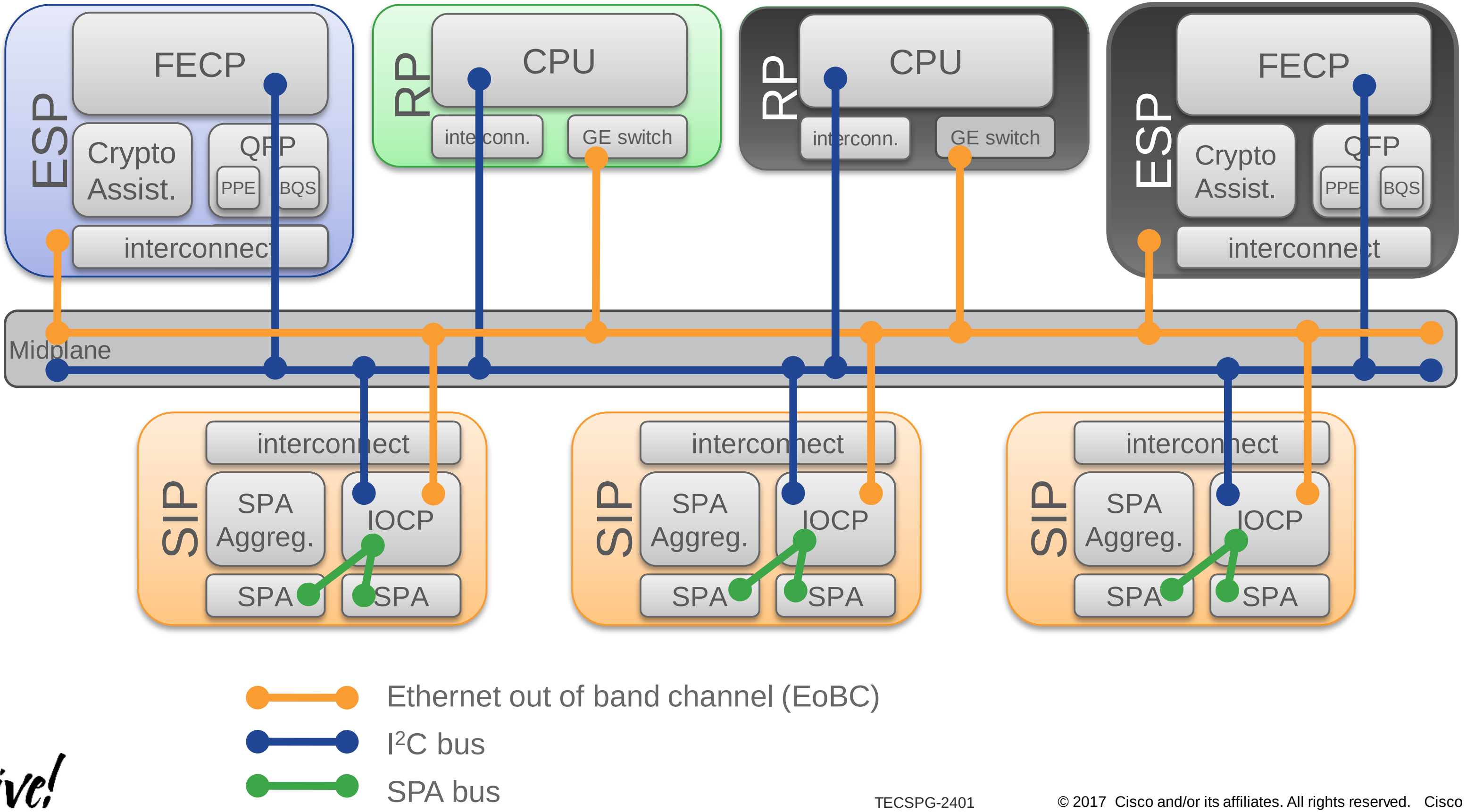
- Redundant ESP / RP on ASR 1006, 1006-X, 1009-X & 1013
- Software Redundancy on ASR 1001-X, 1002-X & 1004
- Max 50ms loss for ESP fail-over
- Zero packet loss on RP fail-over
- Intra-chassis Stateful Switchover (SSO)
  - Stateful features: PPPoX, AAA, DHCP, IPSec, NAT, Firewall
- IOS XE also provides full support for Network Resiliency
  - NSF/GR for BGP, OSPFv2/v3, IS-IS, EIGRP, LDP
  - IP Event Dampening; BFD (BGP, IS-IS, OSPF)
  - first hop redundancy protocols: GLBP, HSRP, VRRP
- Support for ISSU super and sub-package upgrades
- Stateful inter-chassis redundancy available for NAT, SBC, Firewall



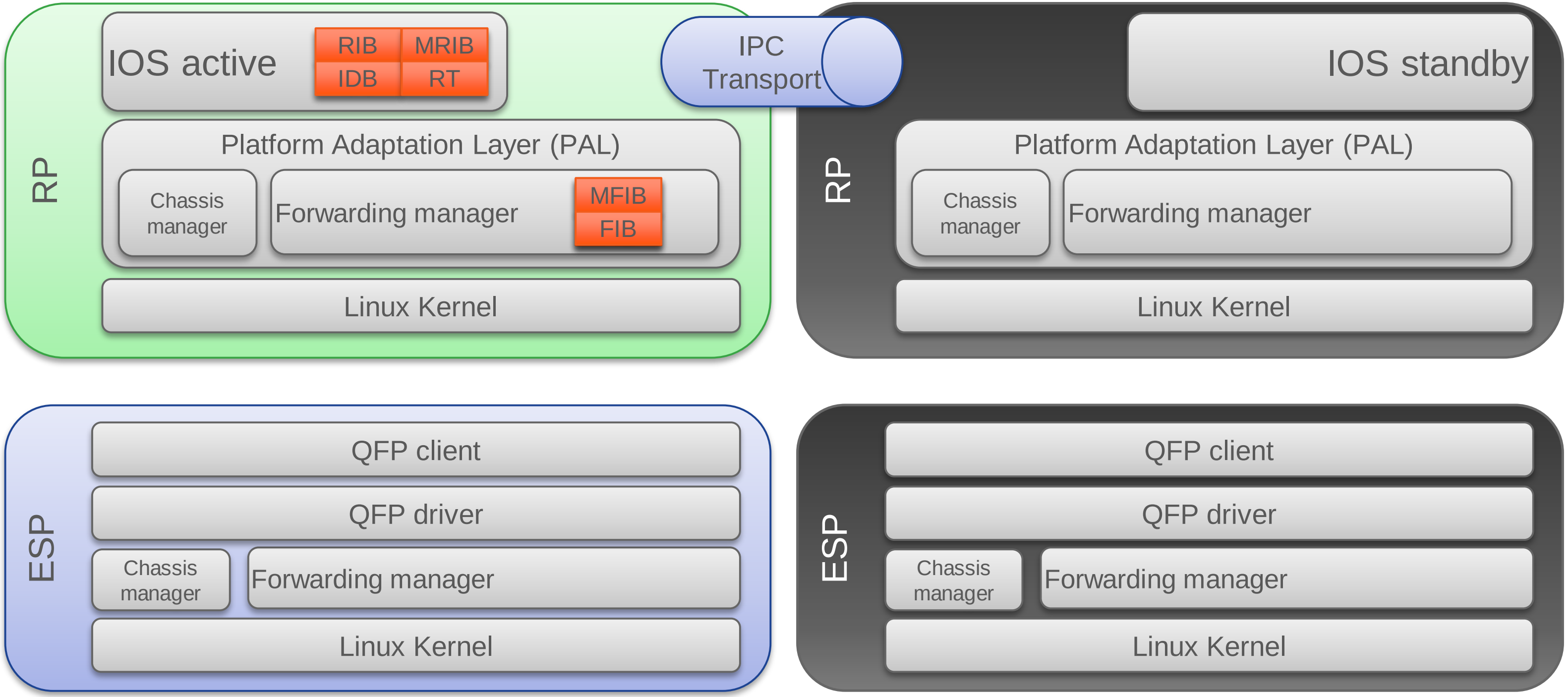
ASR1000 data plane redundancy



ASR1000 control plane architecture



ASR 1006 High Availability Infrastructure

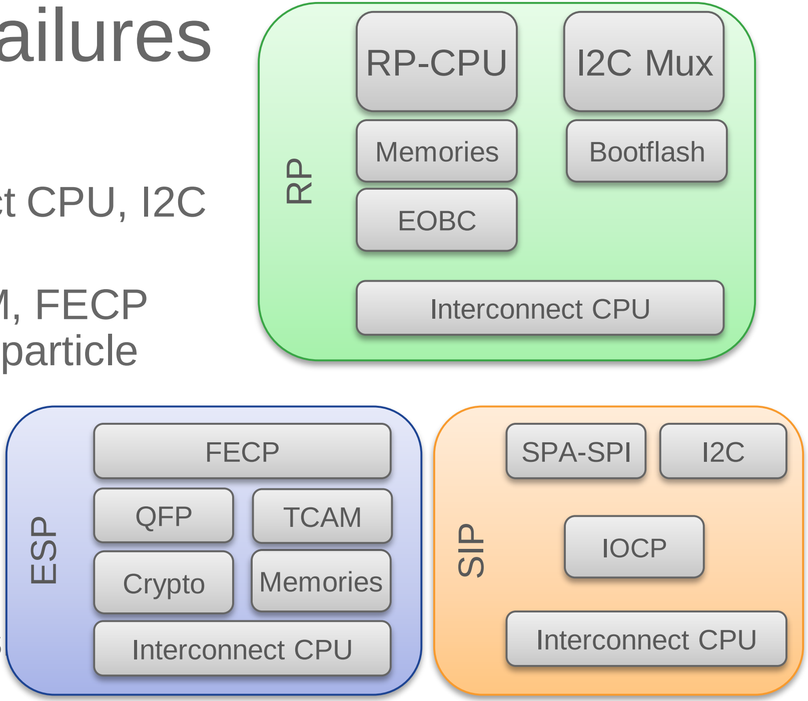


Which Events Trigger Failovers?

- The following events may trigger failovers on the RP/ESP:
  - Hardware component failures
  - Software component failures
  - Online Insertion and Removal (OIR)
  - CLI-initiated failover (e.g. reload command, force-switchover command)

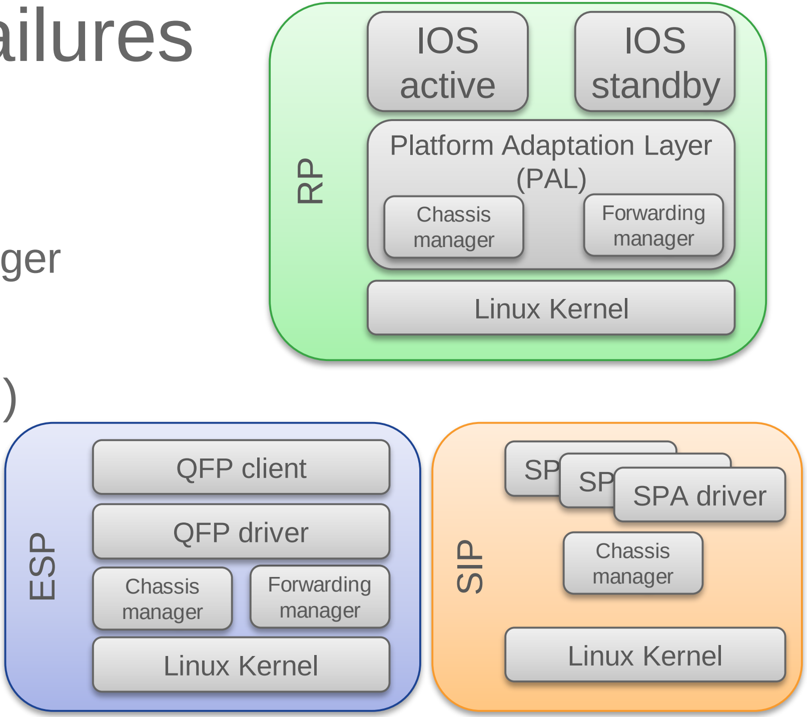
Failover Triggers: Hardware Failures

- What hardware failures?
  - CPUs: RP-CPU, QFP, FECP, IOCP, interconnect CPU, I2C Mux, ESP Crypto Chip, heat sinks, …
  - Memory: NVRAM, TCAM, Bootflash, RP SDRAM, FECP SDRAM, resource DRAM, Packet buffer DRAM, particle length DRAM, IOCP SDRAM, …
  - Interconnects: ESI Links, I2C links, EOBC Links, SPA-SPI bus, local RP bus, local FP bus
- Detected using
  - Software running on the failed hardware will cras software crashes
  - Watchdog timers: low level watchdogs to monitor for failures



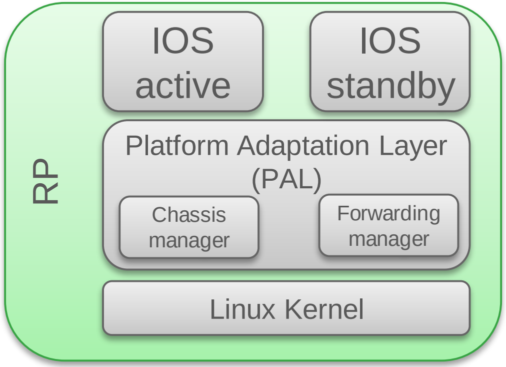
Failover Triggers: Software Failures

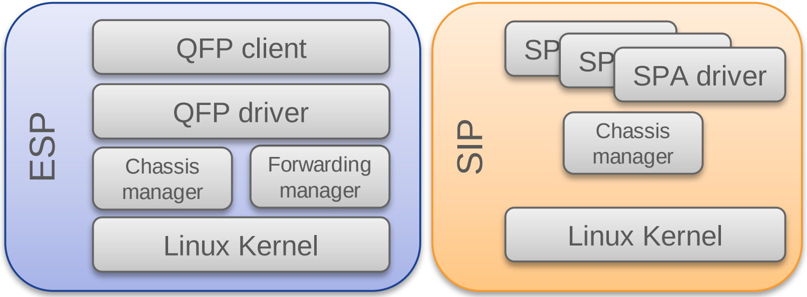
- What software Failures?
  - Kernel: Linux on RP / ESP / SIP
  - Middleware: chassis manager, forwarding manager
  - IOS, SPA drivers
- Detected using the process manager (PMAN)
  - PMAN: every software process has a corresponding PMAN process to check its health
    - if software process crashes, PMAN will detect via a signal from the kernel
  - IPC: between 2 IOS (and only for IOS)
  - Hardware watchdog timers supervise Linux and software stack



Failover Triggers: Software Failures

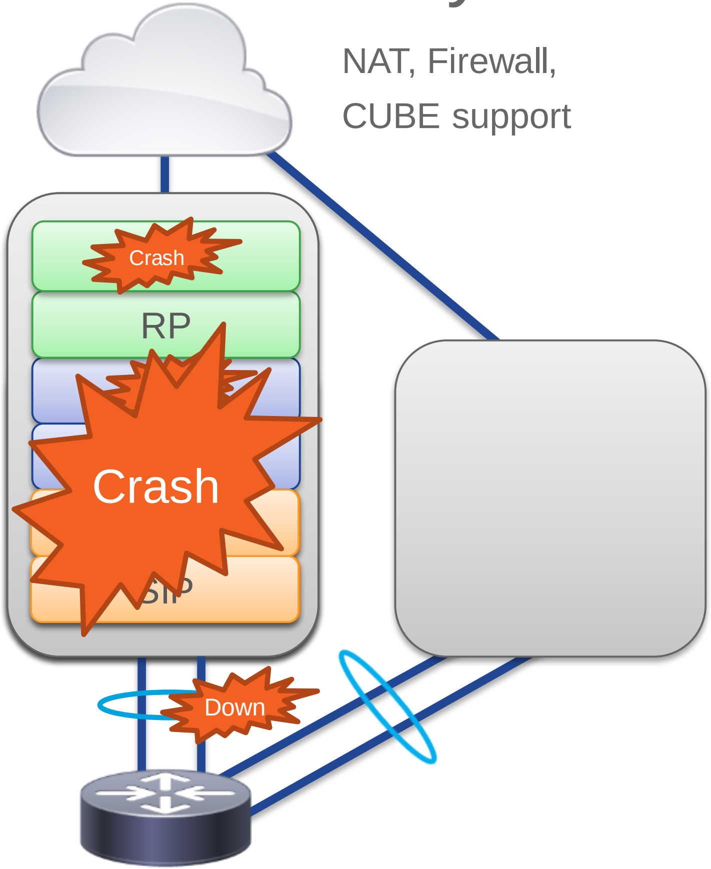
- Kernel will take the module down in a controlled manner
  - IOS, CMESP, CMSIP, FMESP, QFP Driver/Client are not re-startable
  - PMAN-initiated failover using CPLD register bits for ESP or RP
    (failover within 3ms)
- some processes are re-startable (CMRP, FMRP, SSH, telnet, …)
  - Kernel will try to re-start the processes in this case
  - If unsuccessful, then the process will be held down and console message logged





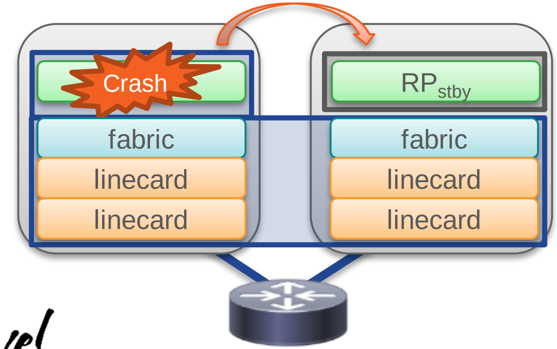
Stateful Application Inter-Chassis Redundancy

- Current Intra-chassis HA typically protects against
  - Control Plane (RP) Failures
  - Forwarding Plane (ESP) failures
  - Interface failures can be mitigated using link bundling (e.g. GEC)
- Any other failures may result in extended recovery times
- Inter-chassis redundancy provides additional resilience against
  - Interface Failures
  - System failures
  - Site failures (allowing for geographic redundancy)



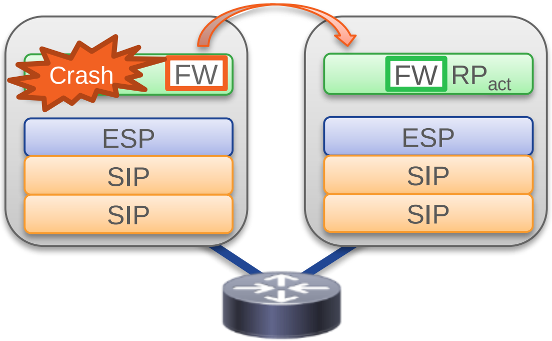
System level redundancy

- VSS, nV
- Failover Granularity at the System Level
- Control-plane active-standby
    - Active RP considers 'remote' linecards under its control
- Forwarding-plane active-active
- No application granularity for failover
    - Need to ensure all features are SSO capable



RG Infra

- Failover granularity at the application level (NAT, Firewall, SBC etc)
- Control plane active-active
    - Each RP only considers its own linecards, but synchronizes application state
- Forwarding-plane active-active
- Can have one set of firewall services resilient, and other set of firewall services non-resilient

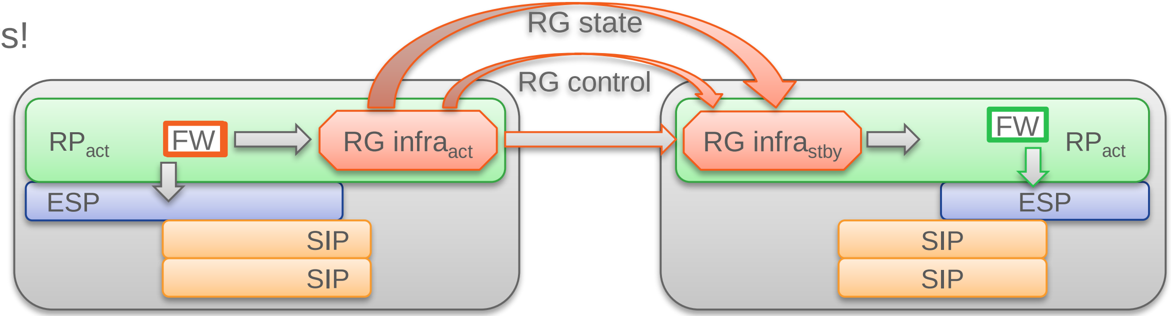


Introduction to RG-Infra

- RG Infra is the IOS Redundancy Group Infrastructure to enable the synchronization of application state data between different physical systems
  - Does the job of RF/CF between chassis
- Infrastructure provides the functions to
  - Pair two instances of RG configured on different chassis for application redundancy purposes
  - Determine active/standby state of each RG instance
  - Exchange application state data (e.g. for NAT/Firewall)
  - Detect failures in the local system
  - Initiate & manage failover (based on RG priorities, allows for pre-emption)
- Assumptions
  - Application state has to be supported by RG infra (ASR 1000 currently supports NAT, Firewall, SBC)
  - Connectivity redundancy solved at the architectural level (need to ‘externalize’ the redundant ESI links of the intra-chassis redundancy solution)

Redundancy Groups Functions

- Registers applications as clients
- Registers (sub)interfaces / {SA/DA}-tuplets in case of firewall
- Determines if traffic needs to be processed or not
- Communicates control information between RGs using a redundancy group protocol
  - Advertisement of RGs and RG state
  - Determination of peer IP address
  - Determination of presence of active RG
- Synchronizes application state data using a transport protocol
- Manages Failovers!



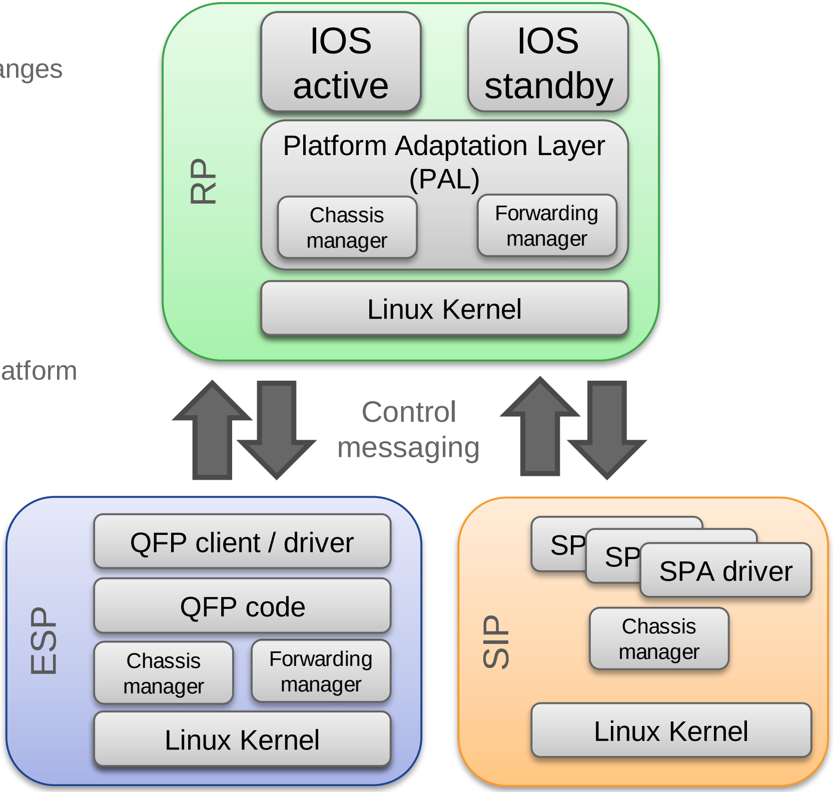
ISSU - In Service Software Upgrade

IOS XE Software packaging - terminology

- IOS XE software for ASR 1000 is released every 4 months, 3 times a year
- Software that is posted in cisco.com is called ‘Consolidated Package’
- Consolidated Package contains several ‘sub-packages’ which are extracted from the Consolidated Package
- The sub-packages can be used to individually upgrade a specific software component of the ASR 1000

Software Sub-packages

- RPBase: RP OS
  - Upgrading of the OS will require reload to the RP and expect minimal changes
- RPIOS: IOS
  - Facilitates Software Redundancy feature
- RPAccess (K9 & non-K9):
  - Software required for Router access; 2 versions will be available. One that contains open SSH & SSL and one without
  - To facilitate software packaging for export-restricted countries
- RPControl :
  - Control Plane processes that interface between IOS and the rest of the platform
  - IOS XE Middleware
- ESPBase:
  - ESP OS + Control processes + QFP client/driver/ucode:
  - Any software upgrade of the ESP requires reload of the ESP
- SIPBase:
  - SIP OS + Control processes
  - OS upgrade requires reload of the SIP
- SIPSPA:
  - SPA drivers and FPD (SPA FPGA image)
  - Facilitates SPA driver upgrade of specific SPA slots



Universal software images

- In IOS XE 3.X software, only the ASR1001-X and the ASR1002-X platforms were running universal software images
- As of 16.x software, all ASR1000 platforms are running universal images
- With universal images, licensing commands entered on the CLI determine the feature set

	Product ID in Cisco IOS XE 3.x	**Product ID in Cisco IOS XE 16.3**	**Description in Cisco IOS XE 16.3**	**License in Cisco IOS XE 16.3**
Cisco ASR 1000 Series RP2 ADVANCED IP SERVICES	SASR1R2- AISK9-316S	SASR1KRPUK9-163	Cisco ASR 1000 Series RPX86 UNIVERSAL	AIS
Cisco ASR 1000 Series RP2 ADVANCED IP SERVICES W/O LI	SASR1R2AI S9NLI316S	SASR1KRPUNLIK9-163	Cisco ASR 1000 Series RPX86 UNIVERSAL W/O LI	AIS
Cisco ASR 1000 Series RP2 ADV ENT SERVICES	SASR1R2- AESK9-316S	SASR1KRPUK9-163	Cisco ASR 1000 Series RPX86 UNIVERSAL	AES
Cisco ASR 1000 Series RP2 ADV ENT SERVICES W/O LI	SASR1R2AE S9NLI316S	SASR1KRPUNLIK9-163	Cisco ASR 1000 Series RPX86 UNIVERSAL W/O LI	AES
Cisco ASR 1000 Series RP2 IP BASE	SASR1R2- IPBK9-316S	SASR1KRPUK9-163	Cisco ASR 1000 Series RPX86 UNIVERSAL	None

Cisco IOS XE Images content

IP Base		
ACL	BGP	EIGRP
ISIS	OSPF	RIP
EEM	ERSPAN	ISSU
HSRP	VRRP	GLBP
Multicast	NAT	NBAR
Netflow	PPPoE client	SNMP
TACACS	All intf	IPSLA
IPv6 parity to IPv4 features		LI
K9 images:	SSH	SSL

Advanced IP services	
IP Base features	
BFD	
Broadband (BNG / ISG)	
CUBE (SP)	CUBE (Ent)
Firewall	L2 & L3 VPN
MPLS	OTV
PfR	LISP
IPSec	EVC/BDI
E-OAM	

Advanced enterprise services
Advanced IP services features
DECNet V
IPX

Some of the features require Feature Licenses in addition to the software image

Data current to IOS XE3.13. Always check Cisco Feature Navigator for the most up to date information regarding features included in releases and feature sets.

ISSU Support Criteria

- When a new IOS XE feature release or rebuild is released, a compatibility matrix will be published identifying all the previous releases and rebuilds that release has been tested for ISSU compatibility.
  - The matrix will be available as part of the cisco.com documentation – ASR1000 Configuration Guide
- Compatibility matrix will refer to IOS XE releases and rebuilds using 'Consolidated Packages' only.
  - Heterogeneous packages are not used for ISSU compatibility testing.

ASR 1000 ISSU

- Ability to perform software upgrade of the IOS image on the single-engine systems
- Support for in-service software downgrade
- “In Service” component upgrades (SIP-Base, SIP-SPA, ESP-Base) without reboot to the system
- Hitless upgrade of some of the software packages in a single engine system
- Hitless upgrade of some software packages in the active RP of a redundant engine system
- Pre-provisioning Capability
- RP Portability - installing & configuring hardware that are physically not present in the chassis
    - Allows configuration of RP in one system (i.e. a 1004) and then move it to another system (i.e. a fully populated 1006)

To / From	3.16.0	3.16.1	3.16.2	3.17.0	3.17.1
3.16.0	N/A	Tested	SSO	Tested	SSO
3.16.1	Tested	N/A	Tested	Tested	SSO
3.16.2	SSO	Tested	N/A	Tested	Tested
3.17.0	Tested	Tested	SSO	N/A	Tested
3.17.1	SSO	SSO	Tested	Tested	N/A

ISSU Compatibility Summary

- ISSU compatibility is determined by the CONTENT of what went into a release, not the type of release (ie release, rebuild, etc)
  - ISSU supported: Across IOS XE rebuilds (Example: 3.16.1 to 3.16.2)
  - ISSU goal: ISSU to work across IOS XE Feature releases (Example 3.15.3 to 3.16.2)
- Compatibility is both forward and backward, if applicable (assuming configuration compatibility)
- Skipping of releases will be allowed, if the 2 releases are ISSU deemed compatible and stated as such in the Compatibility Matrix
- Compatibility is only supported between like IOS XE images. Both images need to have the same feature set of the RP-IOS sub-package. For example:
  - From IPBase-K9 To IPBase-K9
  - From AIS-non-K9 To AIS-non-K9
  - Non-universal to universal ISSU upgrade not supported

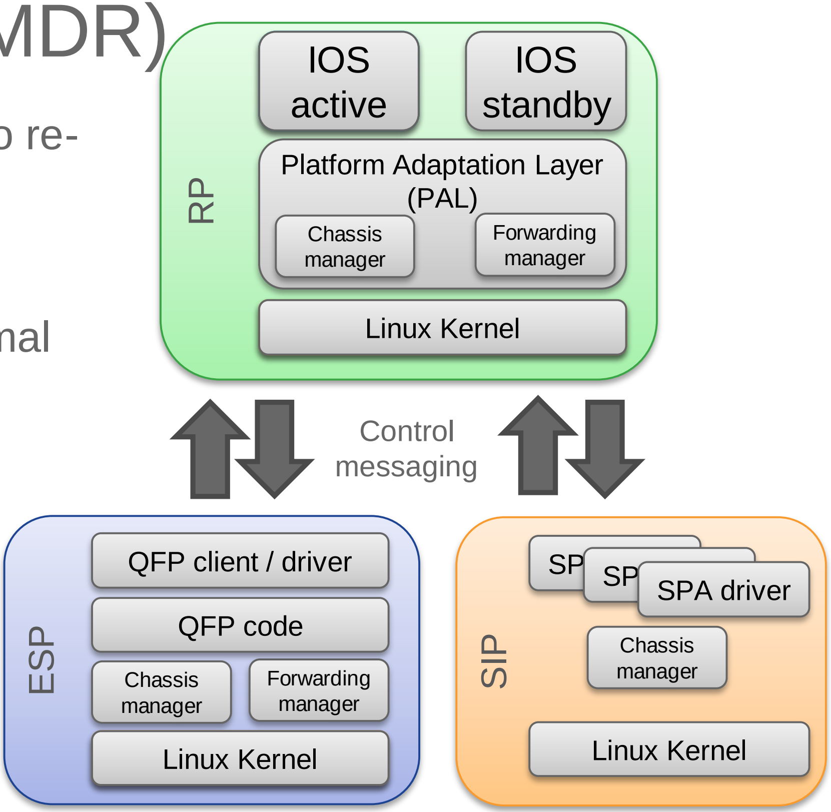
One shot ISSU procedure

- Existing ISSU procedure is a multiple step process. This enhancement greatly simplifies the ISSU process by a single CLI which will execute the multiple steps
- CLI: `request platform software package install node file <filename> sip-delay <1-172800>`
  - Sip-delay will allow delay for each SIP upgrade in the sub-package mode
- When this command is executed, it will automatically be adapted to ‘consolidated mode’ or ‘sub-package mode’ running in the system
- In sub-package mode, this CLI will execute the step-by-step procedure documented in cisco.com
- This table summarizes the support matrix of one shot ISSU in terms of ASR 1000 platform and package mode running in the system

Platform	Consolidated package one shot	Sub-packages one shot
ASR 1013, ASR1009-X	Support	Support
ASR 1006, ASR1006-X	Support	Support
ASR 1004	N/A	Not Supported
ASR 1002-X	N/A	Not Supported
ASR 1001-X	N/A	Not Supported

Minimum Disruptive Restart (MDR)

- Non-MDR upgrade causes 100s packet loss due to re-boot of SIP/SPAs
- MDR reboot time is 25s for SIP, and 10s for SPAs
- SIP/SPA software upgrade can be done with minimal interruption packet flow
- During MDR period, some functions are disabled
  - OIR (SPA or transceiver), APS, interface configuration changes, line alarms
- Requirements / Caveats
  - Hardware (RP, ESP) redundancy
  - Supported for SIP40 (SIP10 does not support MDR)
  - CPLD or FPGA upgrades require full reload of SPA
  - ‘from’ and ‘to’ software versions must support MDR
  - Statistics counters will be re-set after the software upgrade

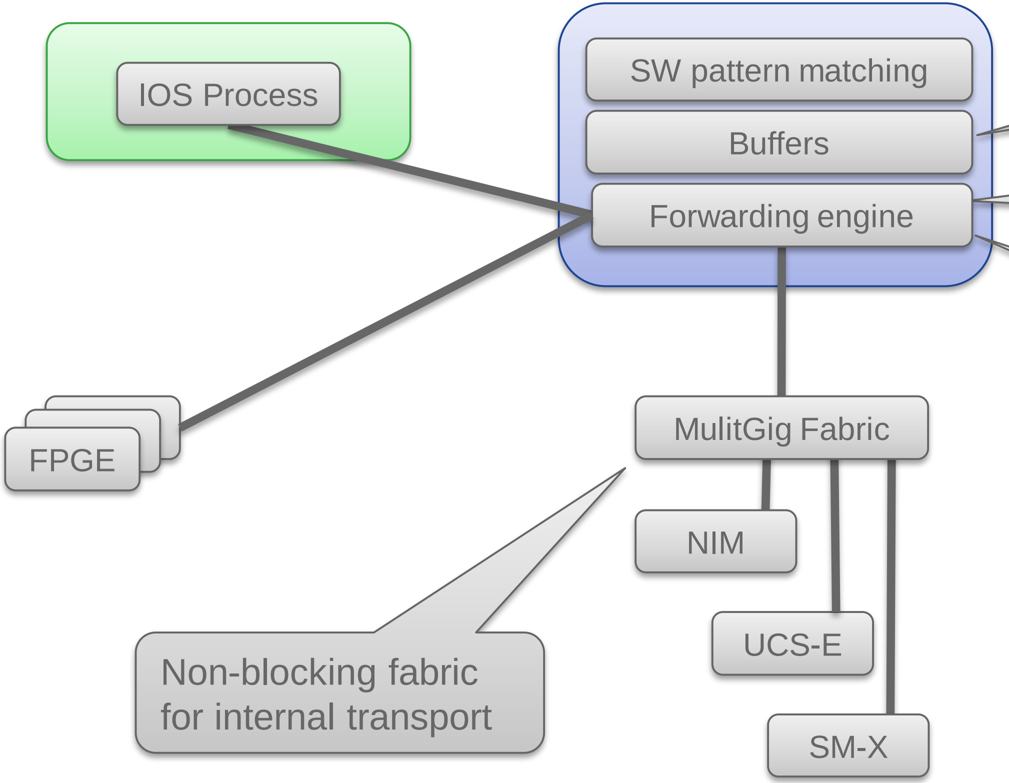


Break – 15 minutes

Quality of Service

ISR4000 only

ISR4000 overall forwarding path with QoS



Packet buffers used by forwarding engine

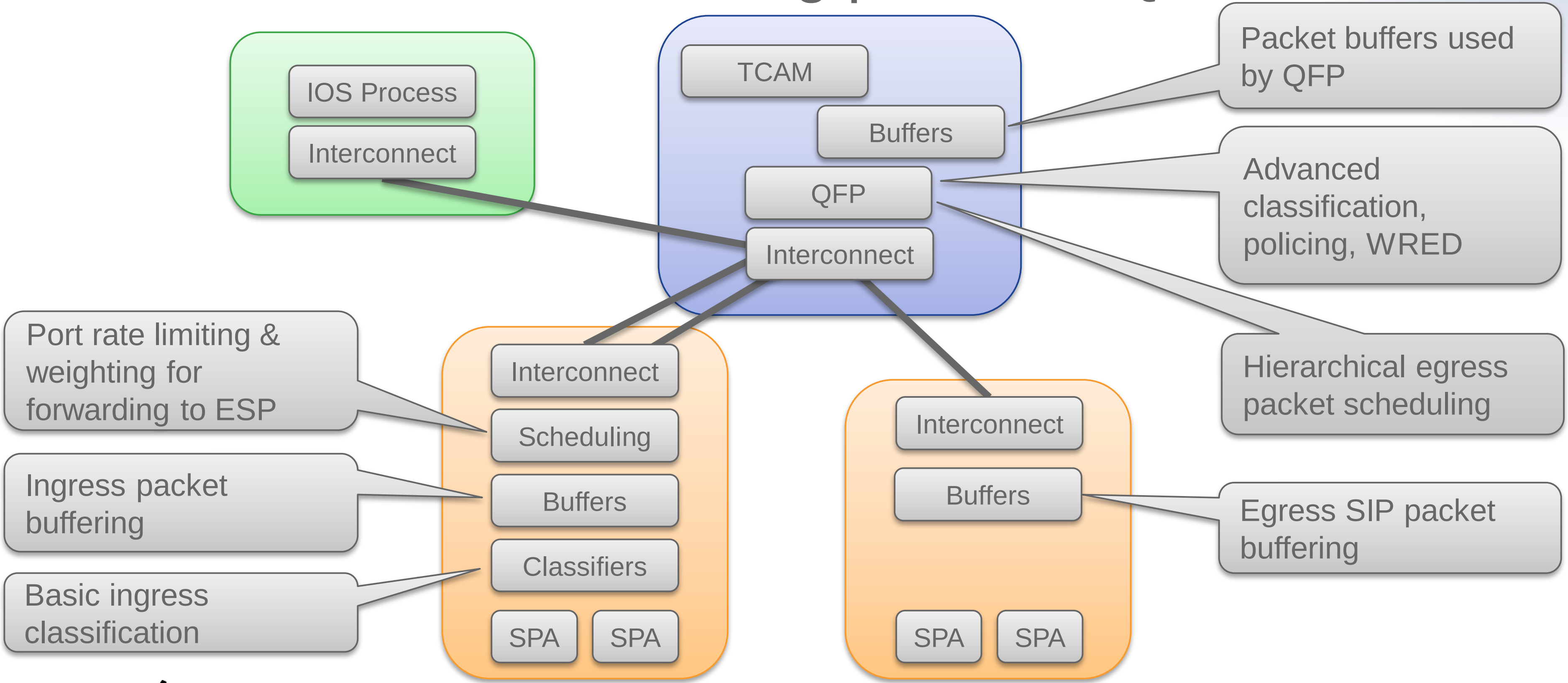
Advanced classification, policing, WRED

Hierarchical egress packet scheduling

Non-blocking fabric for internal transport

ASR1000 only

ASR1000 overall forwarding path with QoS



Packet buffers used by QFP

Advanced classification, policing, WRED

Hierarchical egress packet scheduling

Egress SIP packet buffering

Port rate limiting & weighting for forwarding to ESP

Ingress packet buffering

Basic ingress classification

ASR1000 only

ASR 1000 QoS – SIP ingress path

- Ingress packet priority classification

  Classification based on:
  802.1p, IPv4 TOS, IPv6 TC, MPLS EXP

  Configurable per port or VLAN

- Ingress SIP buffering

  128 MB input buffer

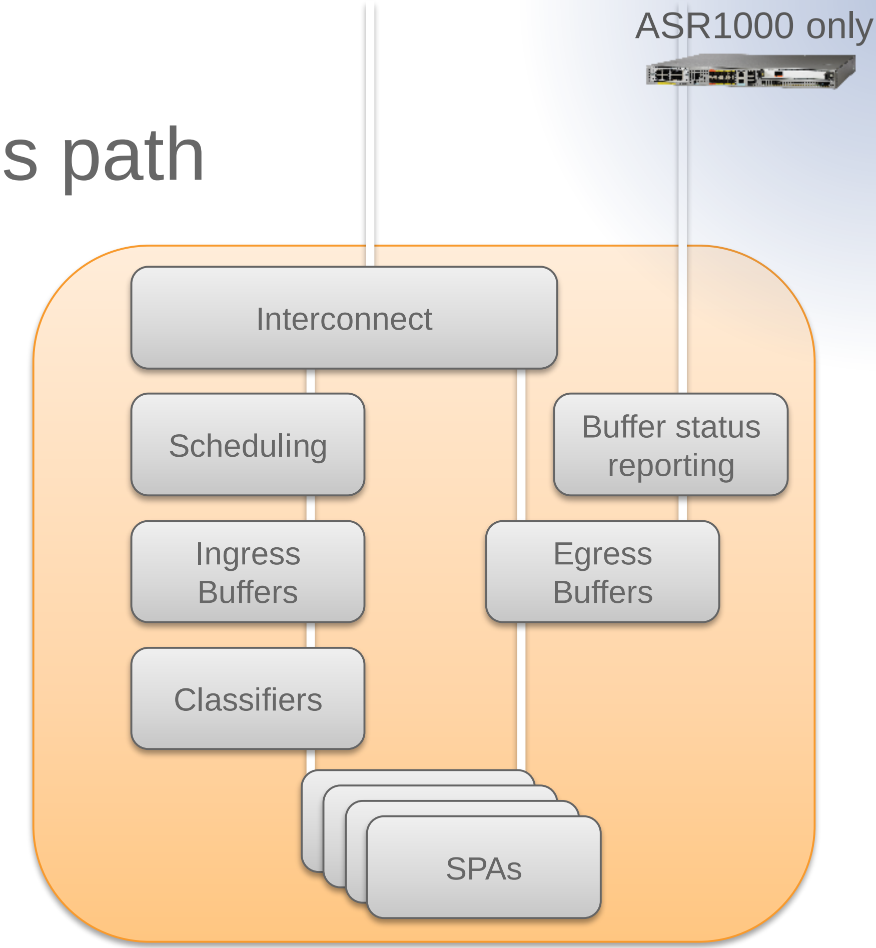
  2 queues, high & low per port

- Ingress SIP scheduler

  Defaults to weighted fair amongst ingress ports

  Excess bandwidth is shared

  Excess weight per port is configurable



ASR1000 only

ASR 1000 SIP ingress path QoS config

For your reference

- `plim qos input policer bandwidth X strict-priority`
  - Limits the amount of high-priority traffic accepted on an interface.
  - X is expressed in kilobits per second.
- `plim qos input queue [0 | strict-priority ] pause enable`
  - Enables the generation of Ethernet pause frames when low / high priority packet depth hits a certain threshold
- `plim qos input queue [0 | strict-priority ] pause threshold X`
  - Defines the threshold of when to generate an ethernet pause frame back to the remote device.
  - X is expressed in percent of queue limit.
- `plim qos input weight X`
  - Defines the weight of the ingress interfaces traffic when scheduling traffic to be sent from the SIP to the ESP for forwarding

ASR1000 only

ASR 1000 SIP ingress path QoS config

For your reference

- `plim qos input map [ ip | ipv6 | mpls | cos ] … queue [ 0 | strict-priority ]`
  - Access to CLI to maps specific IPv4 TOS, IPv6 traffic class , MPLS EXP, or 802.1p values to high or low priority queues.
  - It is possible to classify the various encapsulations simultaneously.
  - cos option is only available on Ethernet subinterfaces.
  - Enabling cos matching will override any main interface matching for traffic on that specific vlan(s).
- By default the following traffic classes are considered high priority
  - IPv4: precedences 6 & 7, DSCP values cs6, cs7
  - IPv6: traffic class ef (46)
  - MPLS: EXP values 6 & 7
  - 802.1p: values 6 & 7

ASR1000 only

ASR 1000 SIP ingress path QoS stats

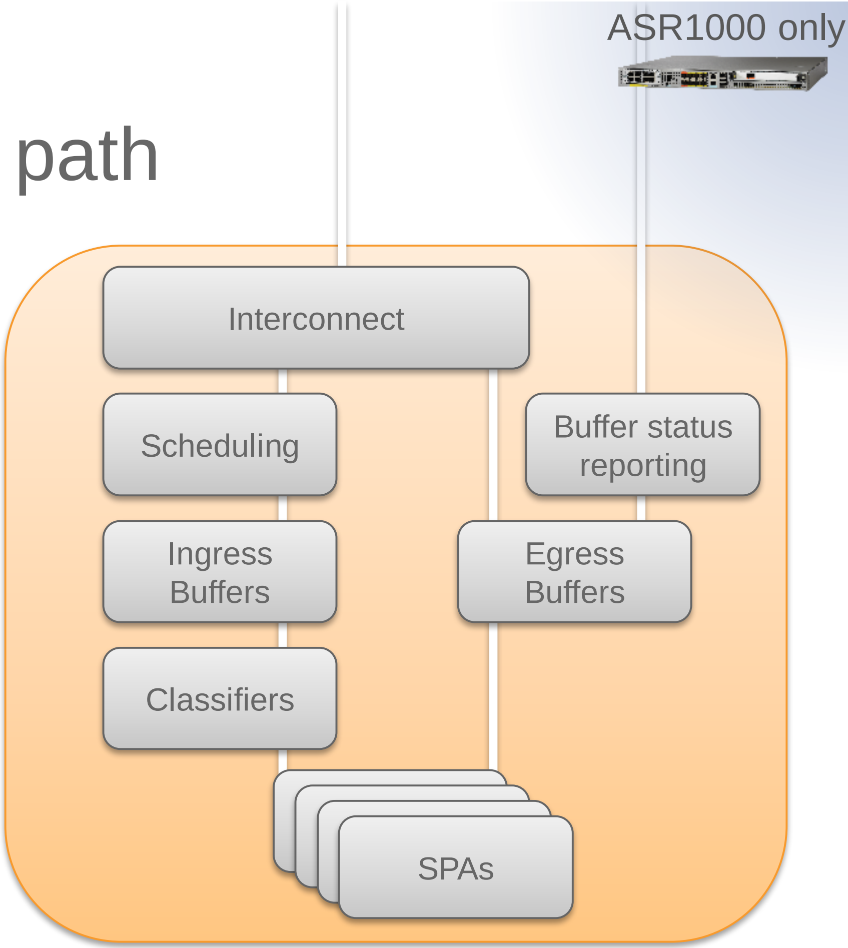
For your reference

- `show platform hard port x/y/z plim qos …`
  - Provides details on QoS configuration for SIP forwarding to ESP.
- `show platform hard port x/y/z plim buffer settings detail`
  - Provides details on SIP buffer utilization in transmit and receive directions.
- `show platform hard interface A x/y/z plim qos input map`
  - Provides details on packet classification for high and low precedence ingress queues on the SIP.

ASR1000 only

ASR 1000 QoS – SIP egress path

- 2 Mbyte of egress buffering per SIP card
- No need for additional SIP based classification or queuing.
- Heavy lifting already done by QFP engine.
- Egress SIP has high and low priority buffers in case there is backpressure from a SPA



For your reference

ASR1000 SIP egress path

- Egress path on SIP has two queues per interface, high and low priorities
- All packets in high priority queue for an interface must be drained before any low priority packets will be sent to the SPA for egress
- `show platform hard slot X plim buffer settings detail`
  - Provides details on egress buffer utilization on SIP. These parameters are not user configurable.

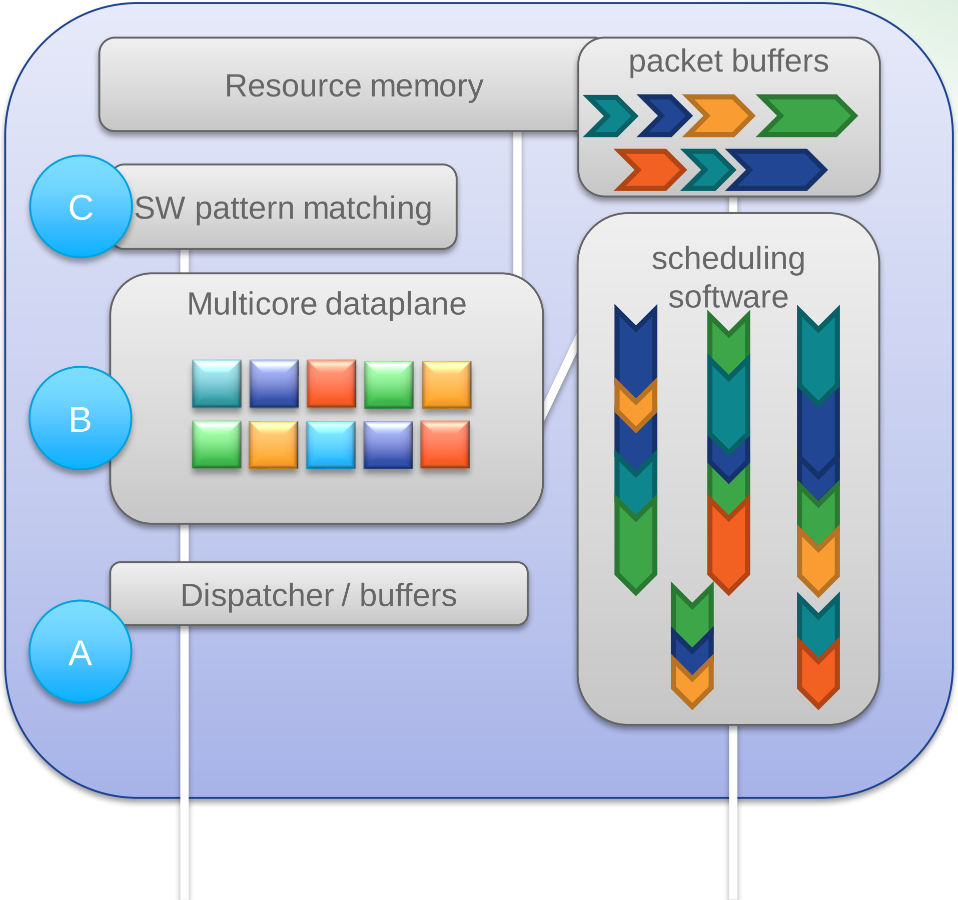
ISR4000 only

ISR4000 QoS forwarding path

A. Ingress packets arrive from MGF and FPGE and are temporarily stored in small internal packet buffer until processed

B. Available core is allocated for a packet and begins processing (security ACLs, ingress QoS, etc)

C. Dataplane cores use SW pattern matching to perform lookups for features enabled for this packet, update statistics, update state for stateful features, forward to crypto engine, find egress interface, etc.



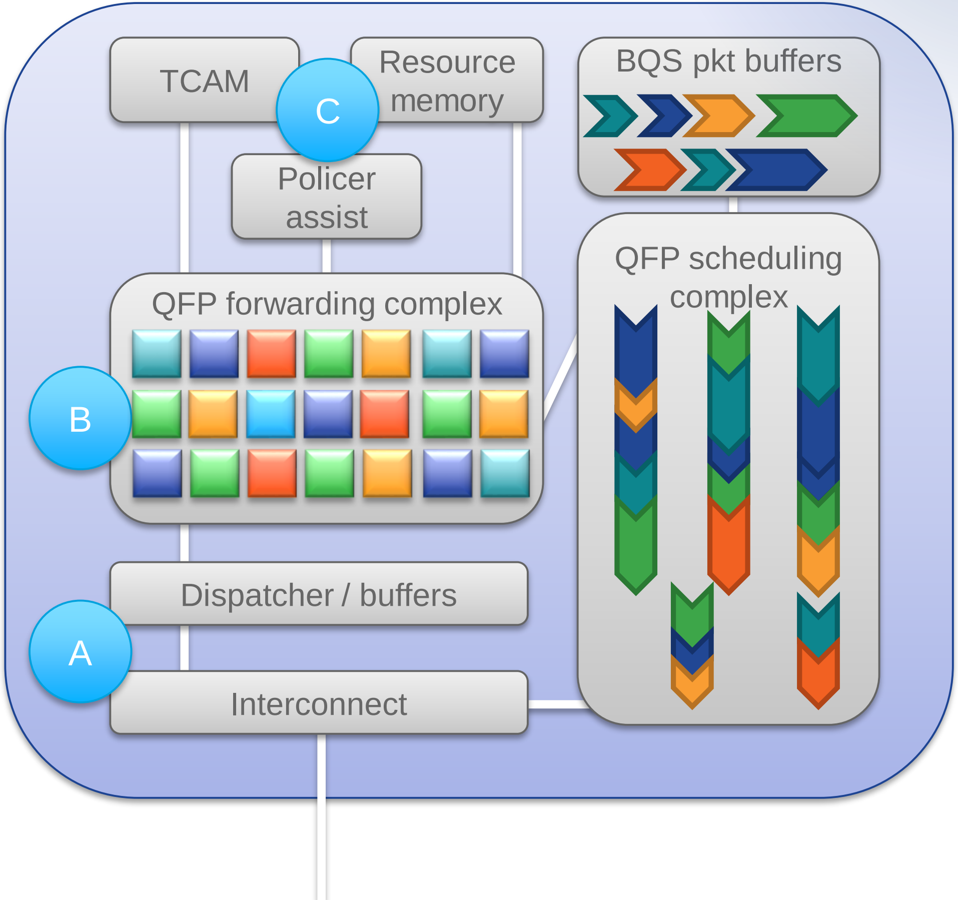
ASR1000 only

ASR1000 QoS ESP forwarding path

A. Ingress packets arrive through the interconnect and are temporarily stored in small internal packet buffer until processed

B. Available QFP PPE is allocated for a packet and begins processing (security ACLs, ingress QoS, etc)

C. QFP PPEs use DRAM and TCAM to perform lookups for features enabled for this packet, update statistics, update state for stateful features, forward to crypto engine, find egress interface, etc.



ASR1000 only

ASR 1000 ESP Interconnect scheduling

- ESP Interconnect scheduling ensures fair access by each SIP to the Cisco QFP
- By default each SIP is allocated:
  - a minimum of approximately 50 Mb/sec of high priority traffic to the Cisco QFP
  - an equal weight for any excess bandwidth beyond the guaranteed minimum
- All high priority traffic from all SIPs is processed before low priority traffic is handed to the Cisco QFP
- These parameters are not user-configurable.

ASR1000 only

ASR1000 ESP interconnect status

For your reference

- `show plat hard slot X serdes qos`
  - Where X is `F0` or `F1`
  - This shows how minimum bandwidth is allocated on the ESP forwarding card for incoming traffic for various linecards in the chassis.

IOS XE Packet Processing Engines for QoS

- Packets are accepted into the forwarding engine and allocated a free core to handle the packet
- Multiple packets are handled simultaneously in the forwarding engine
- The following QoS functions are handled by forwarding engine:
  - Classification
  - Marking
  - Policing
  - WRED
- After all the above QoS functions (along with other packet forwarding features such as NAT, Netflow, etc.) are handled the packet is put in packet buffer memory handed off to the Traffic Manager

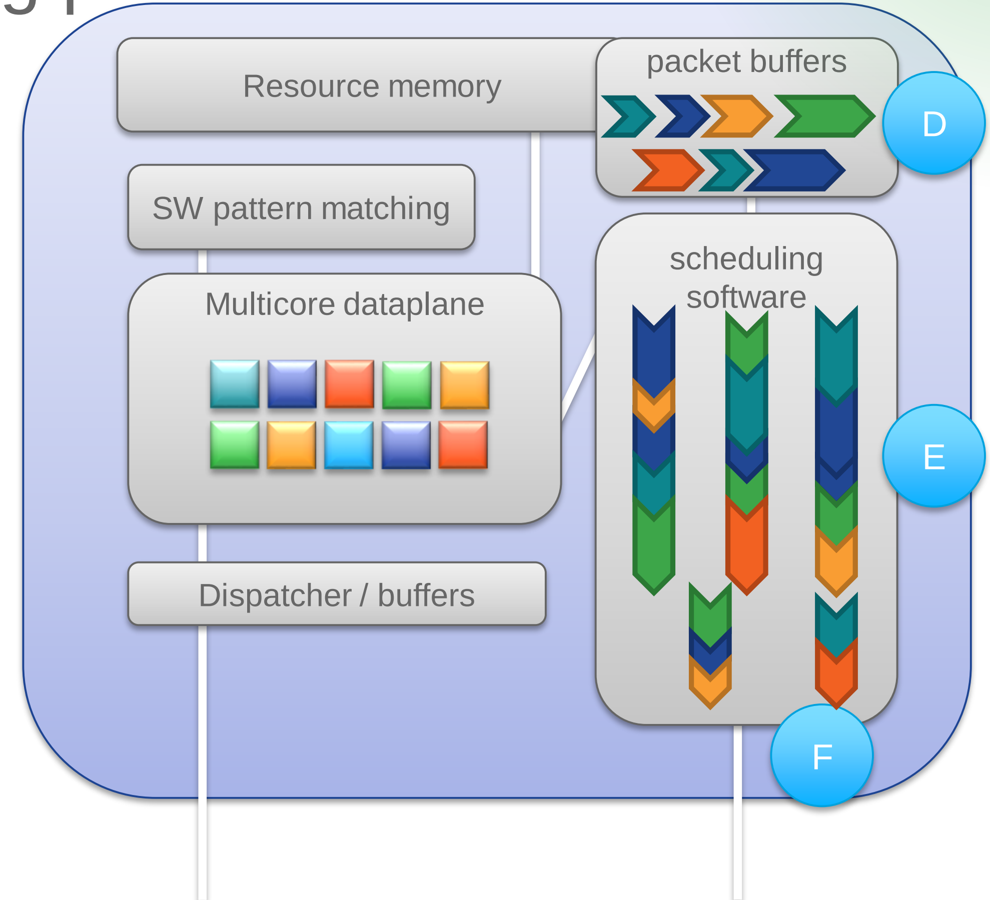
ISR4000 only

ISR4000 QoS forwarding path

D. Once packet processing is complete and packet has been modified, packet is given to the scheduler.

E. Based on default and user configurations, packets are scheduled for transmission based on the egress physical interface.

F. After the packet is release for egress, it is sent to the MGF or PCIe bus for transmission out the physical interface.



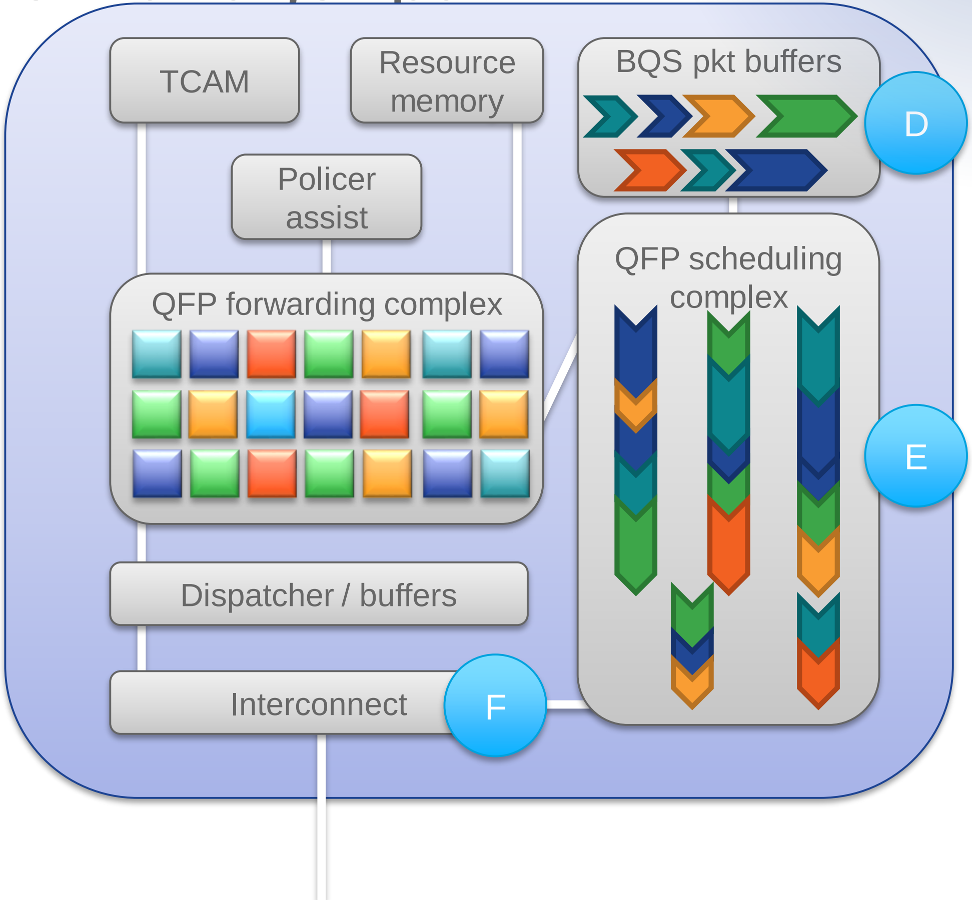
ASR1000 only

ASR1000 QoS ESP traffic manager path

D. Once packet processing is complete and packet has been modified, packet is given to the scheduler.

E. Based on default and user configurations, packets are scheduled for transmission based on the egress physical interface.

F. After the packet is release for egress, it is sent to the interconnect then to the SIP card for egress from a physical interface.



Cisco live!

Traffic Manager processing

- The Traffic Manager performs all packet scheduling decisions.
- Packets move through the QoS hierarchy even if MQC QoS is not configured.
- Traffic Manager implements a 3 parameter scheduler which gives advanced flexibility
  - Minimum - `bandwidth`
  - Excess - `bandwidth remaining`
  - Maximum - `shape`
- Priority propagation (via minimum) ensures that high priority packets are forwarded first without loss

For your reference

Traffic Manager statistics

- `show plat hard qfp active stat drop all | inc BqsOor`
  - This gives a counter which shows if any packets have been dropped because of packet buffer memory exhaustion.
- `show plat hard qfp active infra bqs status`
  - Gives metrics on how many active queues and schedules are in use. Also gives statistics on QFP QoS hierarchies that are under transition.
- `show plat hard qfp active bqs 0 packet-buffer util`
  - Gives metrics on current utilization of packet buffer memory

For your reference

Forwarding engine specifications

Card	Packet memory	Maximum queues	TCAM
ASR 1001-X	512 MB	16,000	10 Mb
ASR 1002-X	512MB	116,000	40 Mb
ASR 1001-HX	512MB	116,000	40 Mb
ASR 1002-HX	1GB	232,000	80 Mb
ESP-20	256 MB	128,000	40 Mb
ESP-40	256 MB	128,000	40 Mb
ESP-100	1GB	232,000	80 Mb
ESP-200	2GB	464,000	80 Mb x 2
ISR 4400 series	750MB	limited by available memory	n/a
ISR 4300 series	300MB		n/a

IOS XE – non-queuing highlights

- Classification
  - IPv4 precedence/DSCP, IPv6 precedence/DSCP, MPLS EXP, FR-DE, ACL, packet-length, ATM CLP, COS, inner/outer COS (QinQ), vlan, input-interface, qos-group, discard-class
  - QFP is assisted in hardware by TCAM on ASR1000, optimized software matching on ISR4000
- Marking
  - IPv4 precedence/DSCP, IPv6 precedence/DSCP, MPLS EXP, FR-DE, discard-class, qos-group, ATM CLP, COS, inner/outer COS
- Detailed match and marker stats may be enabled with a global configuration option
  - `platform qos marker-statistics`
  - `platform qos match-statistics per-filter`
  - `platform qos match-statistics per-ace`
  - Detailed statistics will show per line match statistics in class-maps. For marking, the detailed stats show the number of packets marked per action.

IOS XE QoS – non-queuing

- Policing
  - 1R2C – 1 rate 2 color
  - 1R3C – 1 rate 3 color
  - 2R2C – 2 rate 2 color
  - 2R3C – 2 rate 3 color
  - color blind and aware in XE 3.2 and higher software
    - supports RFC 2697 and RFC 2698
  - explicit rate and percent based configuration
  - dedicated policer block in QFP hardware on ASR1000
  - Policing order of operation (not configurable)
    - XE 3.1 and earlier software evaluates from the parent down to the child
    - XE 3.2 and later software evaluates from the child up through to the parent (the same as queuing functions)

IOS XE QoS – non-queuing

- WRED
  - precedence (implicit MPLS EXP), dscp, and discard-class based
  - ECN marking
  - byte, packet, and time based CLI on ASR1000
  - packet based only on ISR4000
  - packet based configurations limited to exponential constant values 1 through 6 on ASR1000
  - dedicated WRED block in QFP hardware on ASR1000

IOS XE QoS – queuing

- Up to 3 layers of queuing configured with MQC QoS
- Two levels of priority traffic (1 and 2), followed by non-priority traffic
- Strict and conditional priority rate limiting
- 3 parameter scheduler (minimum, maximum, & excess)
- Priority propagation (via minimum) to ensure no loss priority forwarding via minimum parameter
- burst parameters are accepted but not used by scheduler
- Backpressure mechanism between hardware components to deal with external flow control
- `fair-queue` consumes 16 queues for each class configured with it
  - Allows configuration of aggregate queue depth and per-flow queue depth

IOS XE QoS – queuing

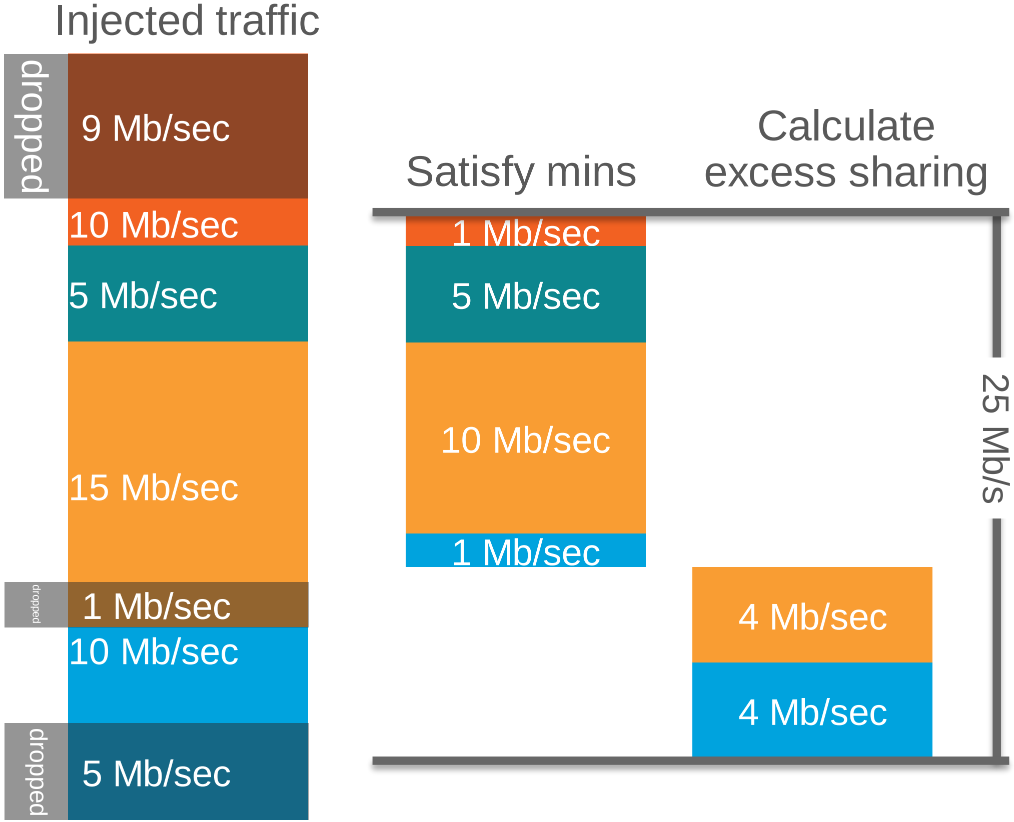
- Queue-limit may be manually configured with various units on ASR1000
  - packets, time, or bytes (packets only on ISR4000)
- Within a policy-map, all classes must use the same type of units for all features
- Using packets based queue-limit deals well with bursts of variable size packets while providing a maximum limit to introduced latency when all packets are MTU sized
- Using time or byte based queue-limits provides more exact control over maximum latency but will hold a variable number of packets based on the size of the packets enqueued
  - Simplifies use of the same policy-map on interfaces of different speeds
  - Time based configuration results in bytes programmed in hardware when policy-map is attached to egress interface

IOS XE QoS – 3 parameter scheduler

- ASR 1000 QFP provides an advanced 3 parameter scheduler
  - Minimum - `bandwidth`
  - Excess - `bandwidth remaining`
  - Maximum - `shape`
- 3 parameter schedulers share excess bandwidth equally in default configuration
  - versus 2 parameter schedules that share excess bandwidth proportional to the minimum configuration
- `bandwidth` and `bandwidth remaining` may not be configured in the same policy-map or class in current software

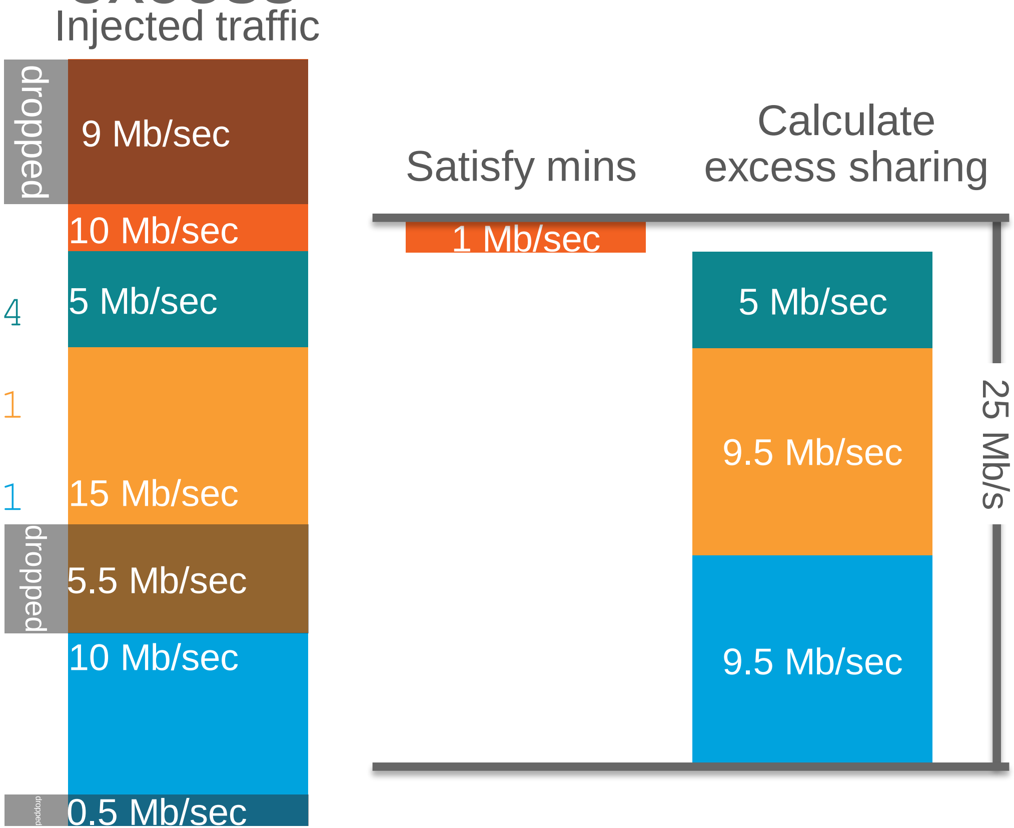
3 parameter scheduler – minimum

```
policy-map child
  class voice
    priority level 1
    police cir 1000000  (bit/sec)
  class critical_services
    bandwidth 10000  (kbit/sec)
  class internal_services
    bandwidth 10000  (kbit/sec)
  class class-default
    bandwidth 1000  (kbit/sec)
!
policy-map parent
  class class-default
    shape average 25000000
    service-policy child
```



3 parameter scheduler – excess

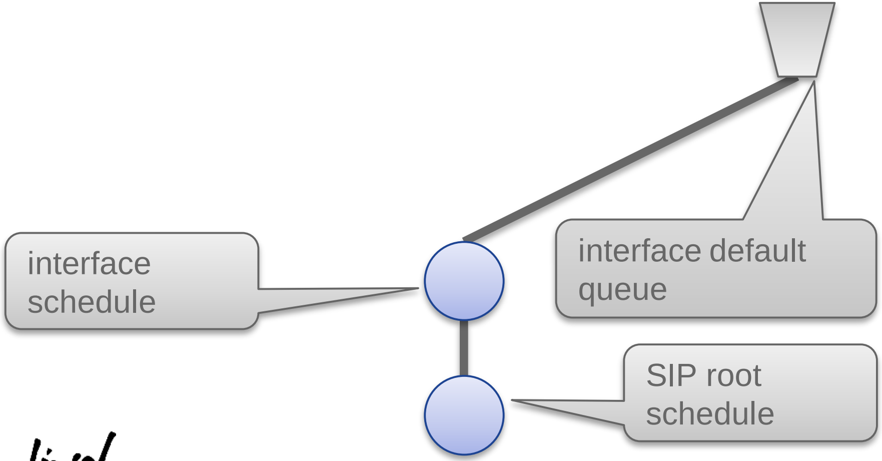
```
policy-map child
  class voice
    priority level 1
    police cir 1000000 (bit/sec)
  class critical_services
    bandwidth remaining ratio 4
  class internal_services
    bandwidth remaining ratio 1
  class class-default
    bandwidth remaining ratio 1
!
policy-map parent
  class class-default
    shape average 25000000
    service-policy child
```



IOS XE QoS hierarchies

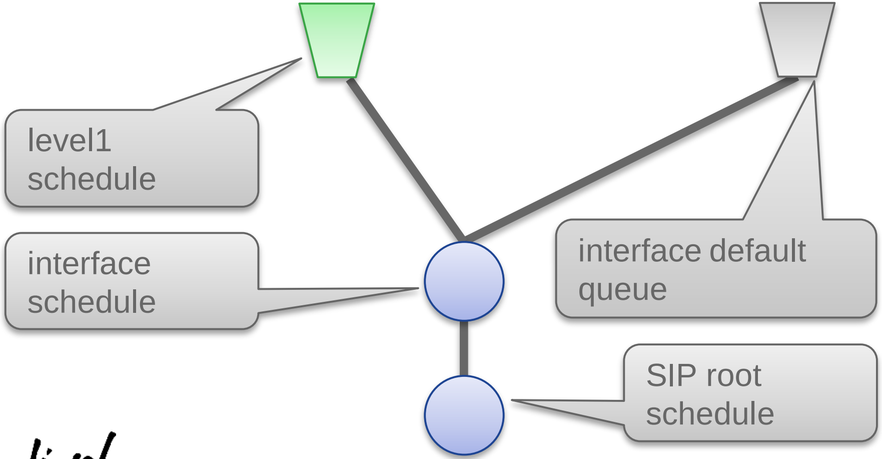
- Generally, MQC based policy-maps with queuing functions may be attached to a physical interface or sub-interface
- It is possible to attach a non-queuing policy-map to one location and then a queuing policy-map to the other
- Some scenarios are supported with 2 level hierarchical policy-maps on tunnels and a class-default shaper on the physical interface
- Broadband applications have their own set of supported scenarios which support queuing policy-maps on sub-interfaces and then on the dynamically created sessions which traverse that sub-interface
- Innovative hierarchies which move beyond strict parent-child hierarchies can be built using service-fragment CLI

IOS XE QoS hierarchy



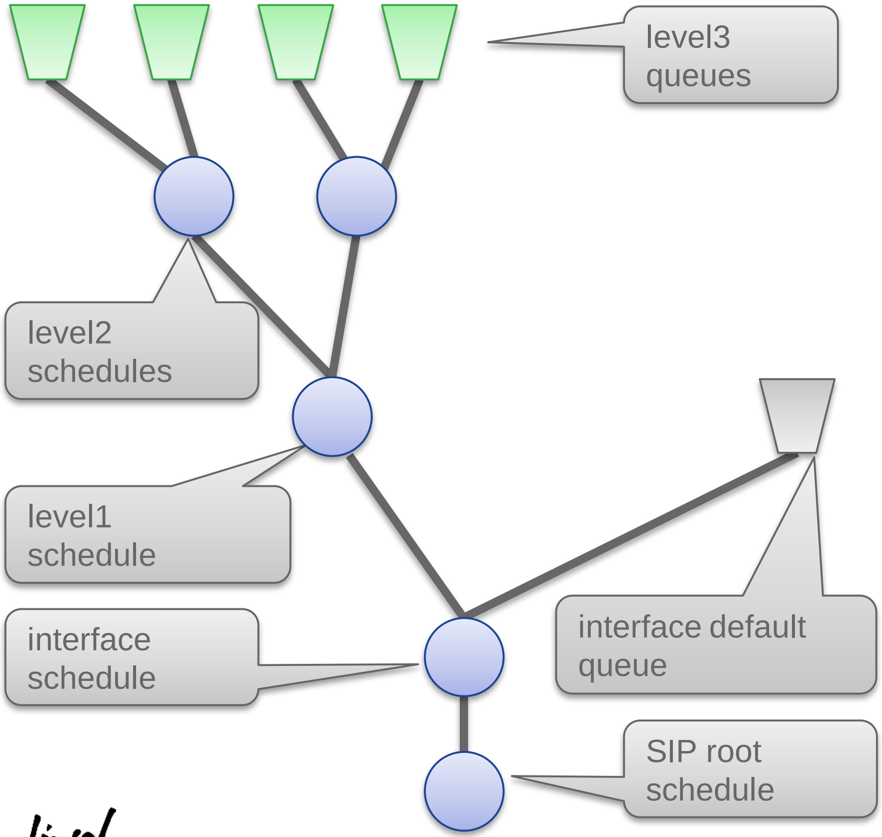
```
policy-map level1
  class class-default
    shape average 100000000
!
!
policy-map level2
  class user1
    shape average 60000000
    service-policy level3
  class class-default
    shape average 60000000
    service-policy level3
!
policy-map level3
  class prec0
    priority
    police cir 10000000
  class class-default
!
interface gig0/0/0.2
!
!
interface gig0/0/0.3
!
```

IOS XE QoS hierarchy



```
policy-map level1
  class class-default
    shape average 100000000
!
!
policy-map level2
  class user1
    shape average 60000000
    service-policy level3
  class class-default
    shape average 60000000
    service-policy level3
!
policy-map level3
  class prec0
    priority
    police cir 10000000
  class class-default
!
interface gig0/0/0.2
  service-policy out level1
!
interface gig0/0/0.3
!
```

IOS XE QoS hierarchy



```
policy-map level1
  class class-default
    shape average 100000000
    service-policy level2
!
policy-map level2
  class user1
    shape average 60000000
    service-policy level3
  class class-default
    shape average 60000000
    service-policy level3
!
policy-map level3
  class prec0
    priority
    police cir 10000000
  class class-default
!
interface gig0/0/0.2
  service-policy out level1
!
interface gig0/0/0.3
!
```

IOS XE observed Packet and frame sizes

- Traffic manager calculates ethernet packet size on everything between the MAC L2 header and the end of payload
  - IFG, preamble, and FCS are not included
  - For queuing features, the packet size can be adjusted manually

```
shape average 1000000 account user-defined -4
```

  - atm cell overhead compensation is available so that ATM L2 links downstream are not overdriven

```
shape average 1000000 account user-defined 24 atm
```

- Traffic manager includes the 4 byte CRC in packet sizes for frame-relay
- MQC based QoS for ATM performs L3 shaping and only compensates for atm cell overhead with the above atm directive in shaped classes
  - ATM vc rate configurations are ATM L2 based shaping

IOS XE Etherchannel QoS support

- With VLAN based load balancing
  1. Egress MQC queuing configuration on Port-channel sub-interfaces
  2. Egress MQC queuing configuration on Port-channel member link
  3. Policy Aggregation – Egress MQC queuing on sub-interface
  4. Ingress policing and marking on Port-channel sub-interface
  5. Egress policing and marking on Port-channel member link
  6. Policy Aggregation – Egress MQC queuing on main-interface (XE2.6 and higher)
- Active/standby with LACP (1+1)
  7. Egress MQC queuing configuration on Port-channel member link (XE2.4 and higher)
  9. Egress MQC queuing configuration on PPPoE sessions, model D.2 (XE3.7 and higher)
  10. Egress MQC queuing configuration on PPPoE sessions, model F (XE3.8 and higher)
- Etherchannel with LACP and load balancing (active/active)
  8. Egress MQC queuing configuration supported on Port-channel member link (XE2.5 and higher)
  11. General MQC QoS support on Port-channel main-interface (XE3.12 and higher)

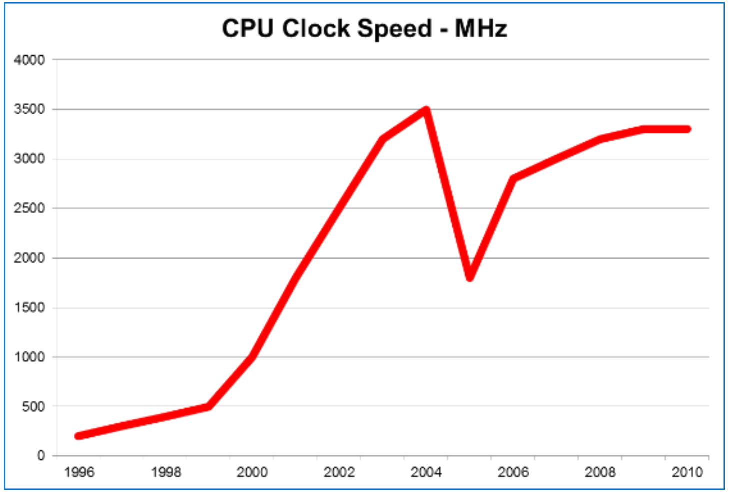
Aggregate Etherchannel QoS

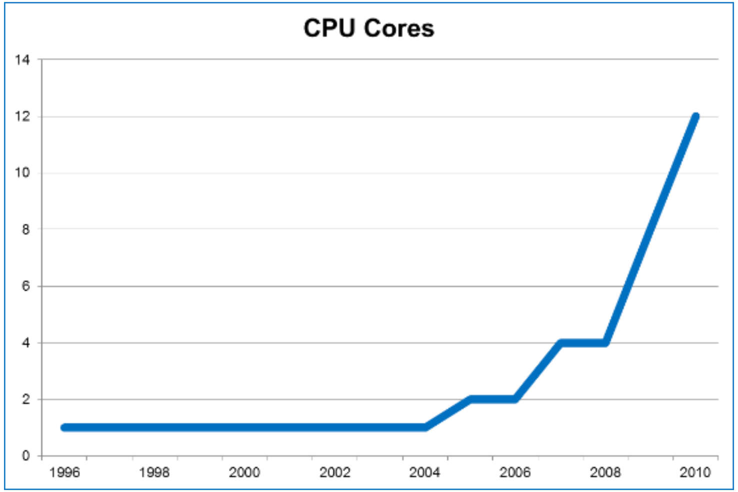
with LACP and flow based load balancing

- Requires that aggregate Port-channel interfaces be identified before creation with the `platform qos port-channel-aggregate X` command
- Up to four member links per aggregate Port-channel are supported
- FastEthernet and GigabitEthernet and TenGigabitEthernet interfaces are supported
  - All member links in a port-channel must be the same speed
- Policy-maps may be applied to the aggregate Port-channel main interface, sub-interface, or service-groups
  - No member link QoS
- For vlan based QoS, a policy-map using VLAN classification should be applied to the aggregate Port-channel main-interface
- Supports 3 levels of hierarchical policy-maps
  - Including 3 levels of policers and/or queuing

Performance

CPU Evolution





What happened in 2005?
What about Moore’s Law?

Multi-core, great for the general user.
How does IOS deal with this?

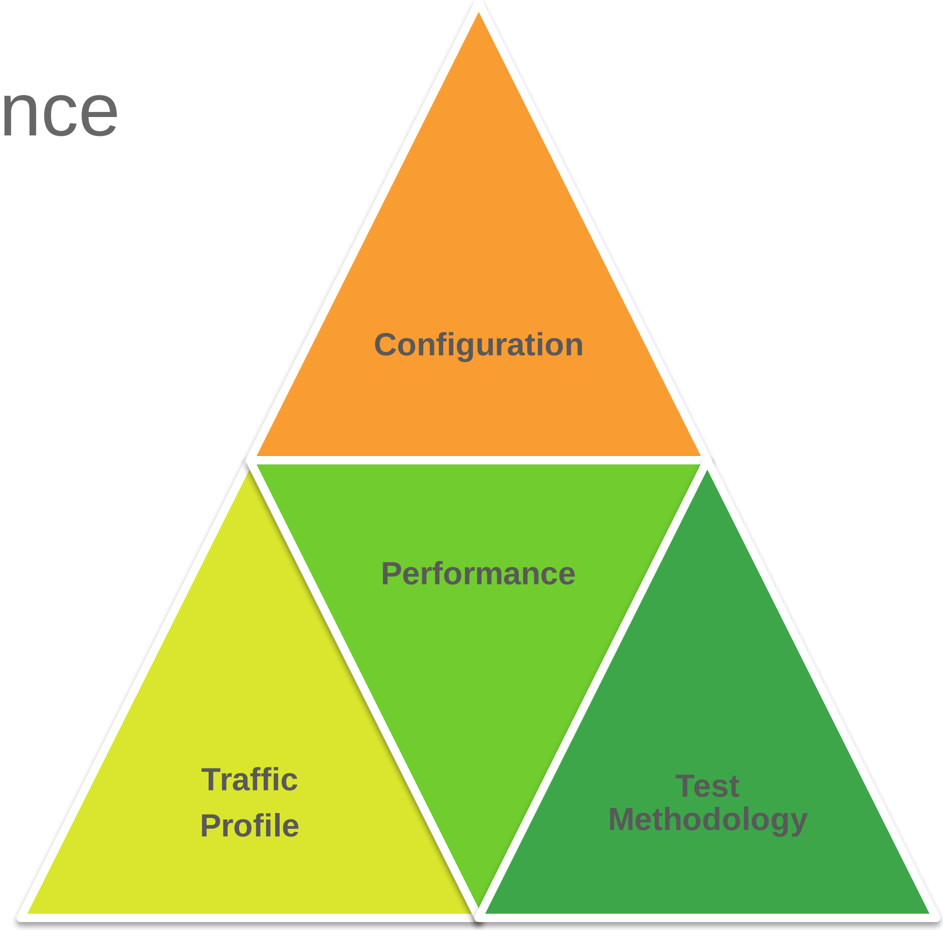
What makes up performance

System aspects:

- Available purpose-built processors / chips
- BW of physical interfaces
- Platform Internal Architecture (i.e. MGF)
- Operative System architecture

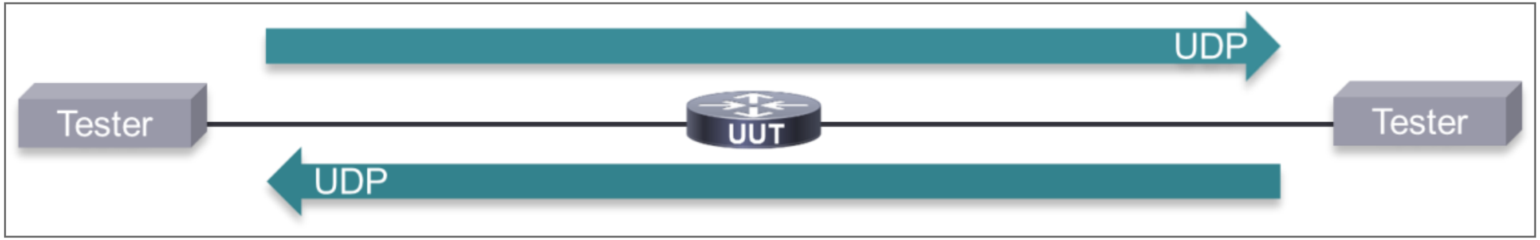
Test aspects:

- Traffic profile (frame size & traffic type)
- Enabled Features
- Test Methodology (NDR)

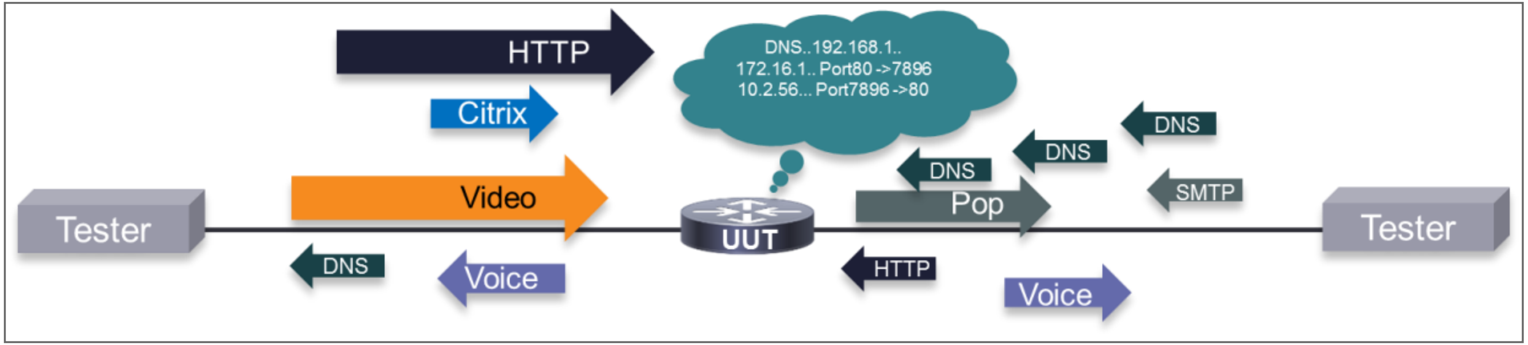


Traffic Profile Overview

Stateless

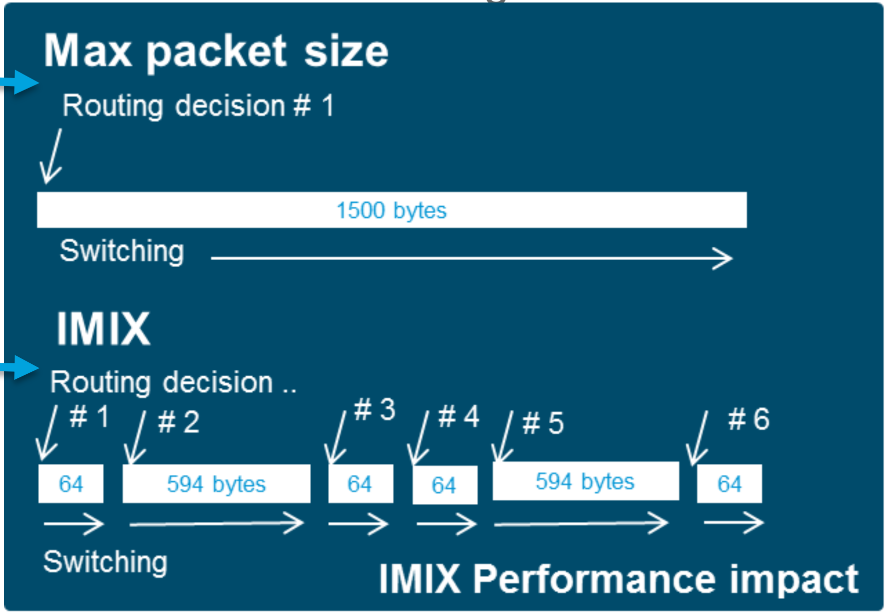


Stateful



Impact of Packet size

- One route decision = One packet served
- Routing capacity = Number of packets per second served for a given service.
- Big packets
  - Many bits switched for each route decision
  - = Higher Mbps number
- Small packets
  - Few bits switched for each route decision.
  - = Lower Mbps number



Mbps or PPS?

Stateless FW	Mbps			PPS		
Platform	64	IMIX	1518	64	IMIX	1518
1941	19.0	108.5	450.8	37,201	37,493	37,120

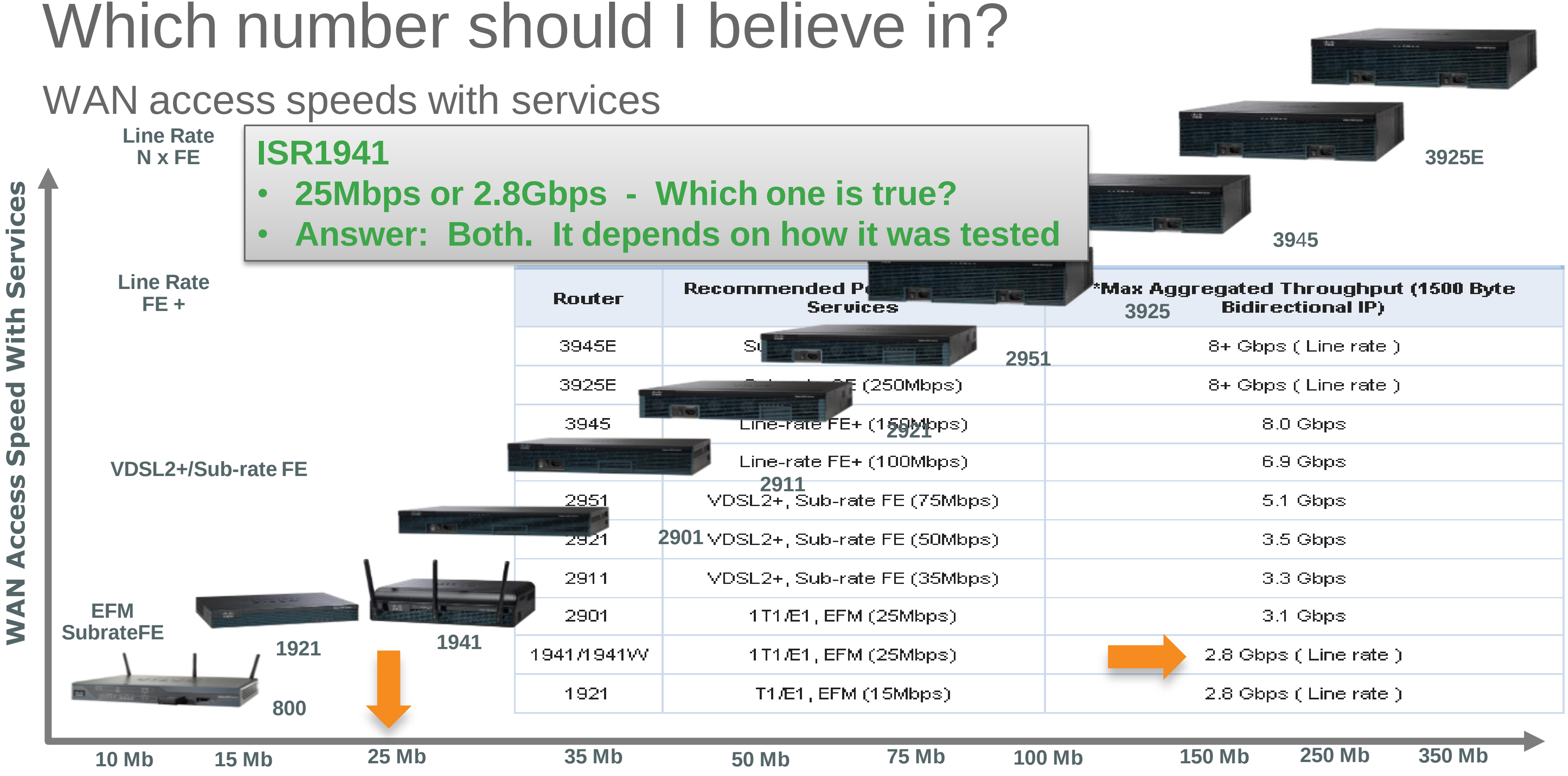
- Example: 1941 with Firewall configuration and different frame sizes
- Min. Frame Size (64 byte) has 23 times less Mbit/s than Max. Frame Size (1518)
- Across different Frame Sizes the pps is contant, even though Mbps varies
- Packet per second = The true routing capacity and hard to skew
- Only applies until interface or performance license limit is reached

Which number should I believe in?

WAN access speeds with services

ISR1941

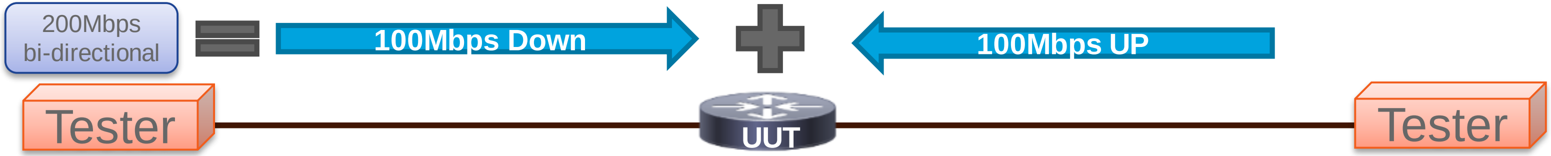
- **25Mbps or 2.8Gbps - Which one is true?**
- **Answer: Both. It depends on how it was tested**



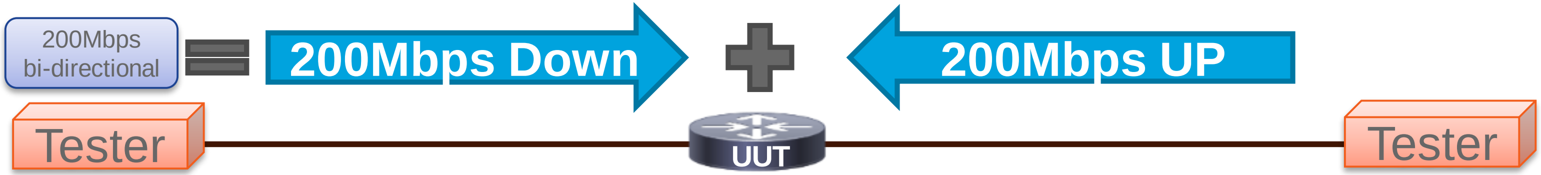
Router	Recommended P[illegible] Services	*Max Aggregated Throughput (1500 Byte Bidirectional IP)
3945E	Su	8+ Gbps (Line rate)
3925E	Sub-rate GE (250Mbps)	8+ Gbps (Line rate)
3945	Line-rate FE+ (150Mbps)	8.0 Gbps
3925	Line-rate FE+ (100Mbps)	6.9 Gbps
2951	VDSL2+, Sub-rate FE (75Mbps)	5.1 Gbps
2921	VDSL2+, Sub-rate FE (50Mbps)	3.5 Gbps
2911	VDSL2+, Sub-rate FE (35Mbps)	3.3 Gbps
2901	1 T1/E1, EFM (25Mbps)	3.1 Gbps
1941/1941W	1 T1/E1, EFM (25Mbps)	2.8 Gbps (Line rate)
1921	T1/E1, EFM (15Mbps)	2.8 Gbps (Line rate)

How to report the result?

- Performance data is usually referred to as either **”Uni-directional”** or **”Bi-directional”**
  - Uni-directional: A traffic flow going to or from a device, not in both directions
  - Bi-directional: Can mean one of two things depending on who’s using the term.
    1. The sum flows in both directions, hence the term Bi-directional



2. Bandwidth expected in both directions, hence the term Bi-directional



# 2 is thus twice as high as #1

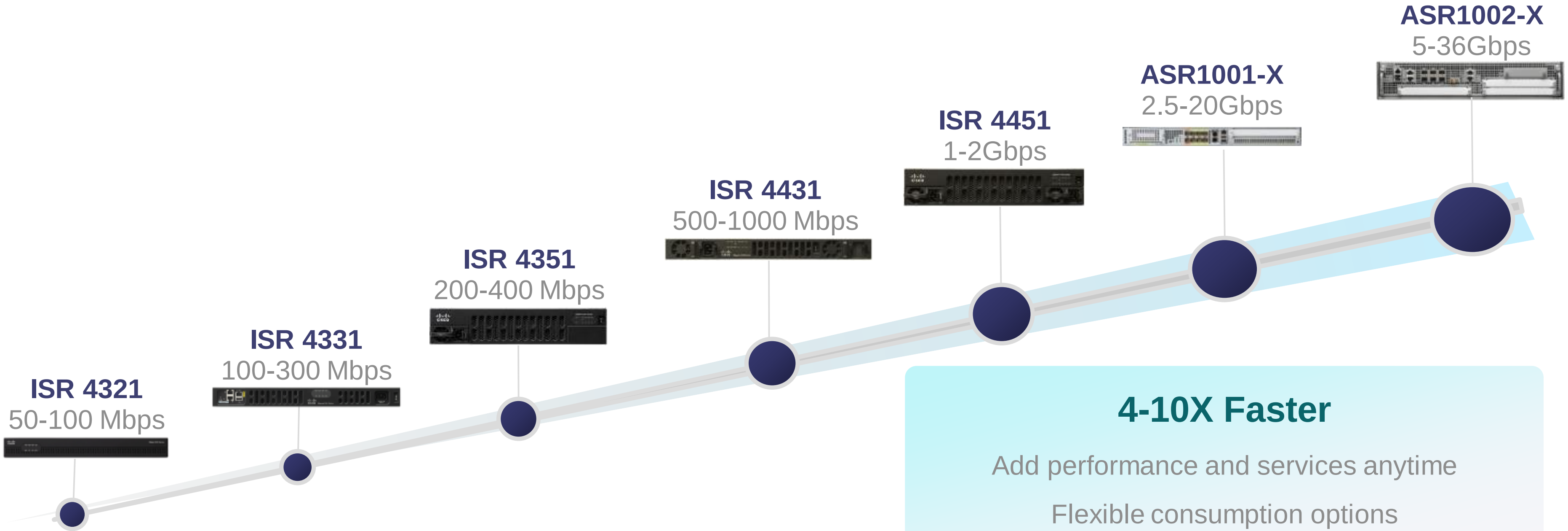
Reporting results unambiguously

- ”**Aggregate**” = The sum of all traffic flows going to and from a device

- Why we report in aggregate numbers:
  - represents total performance capacity
  - the router’s CPU doesn’t care which way a packet is going
  - the traffic generator aggregates all traffic it detects in all flows
  - whether it’s a ratio of 90% down and 10% up, or a perfect 50/50 split doesn’t matter

Pay-As-You-Grow performance with ISRs & ASRs

Investment Protection Without Oversubscription



4-10X Faster

Add performance and services anytime

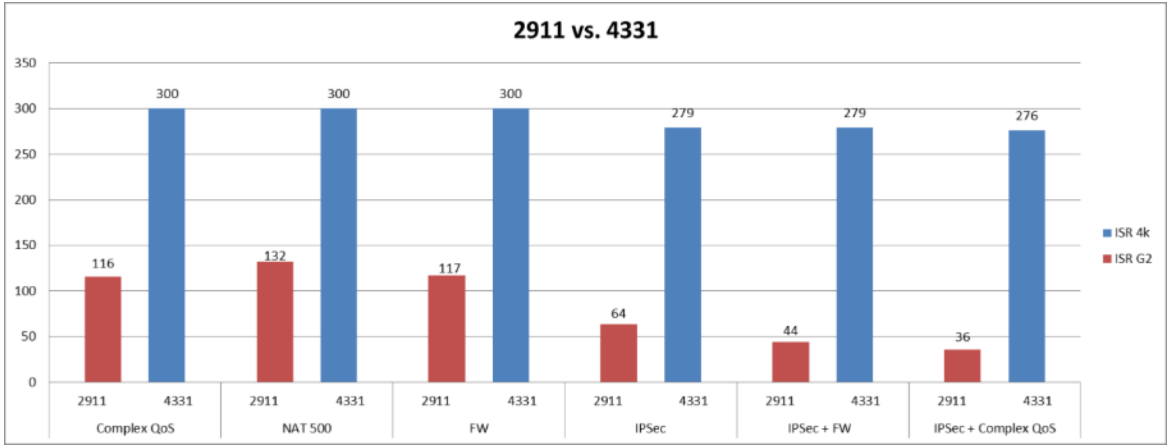
Flexible consumption options

Performance license limit – ISR4000 example

- Notice that many of the results are at the exact licensed max limit.
- This means router hit shaper before bottoming out
- How much CPU is then left?

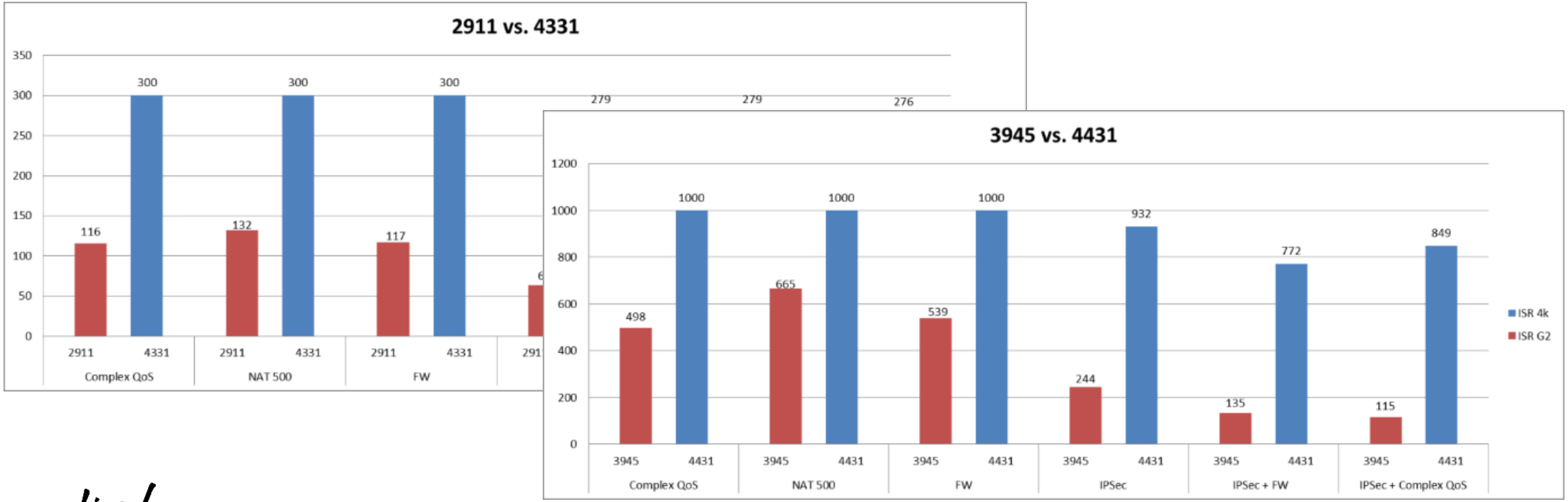
Performance license limit – ISR4000 example

- Notice that many of the results are at the exact licensed max limit.
- This means router hit shaper before bottoming out
- How much CPU is then left?



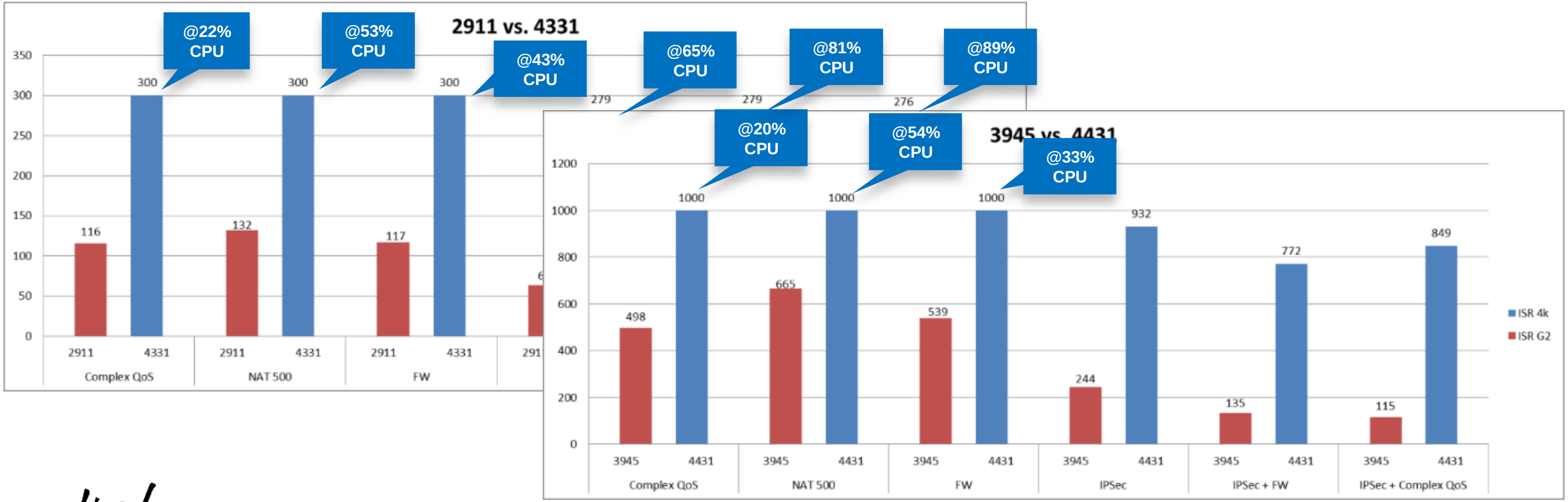
Performance license limit – ISR4000 example

- Notice that many of the results are at the exact licensed max limit.
- This means router hit shaper before bottoming out
- How much CPU is then left?



Performance license limit – ISR4000 example

- Notice that many of the results are at the exact licensed max limit.
- This means router hit shaper before bottoming out
- How much CPU is then left?



@22% CPU

@53% CPU

@43% CPU

@65% CPU

@81% CPU

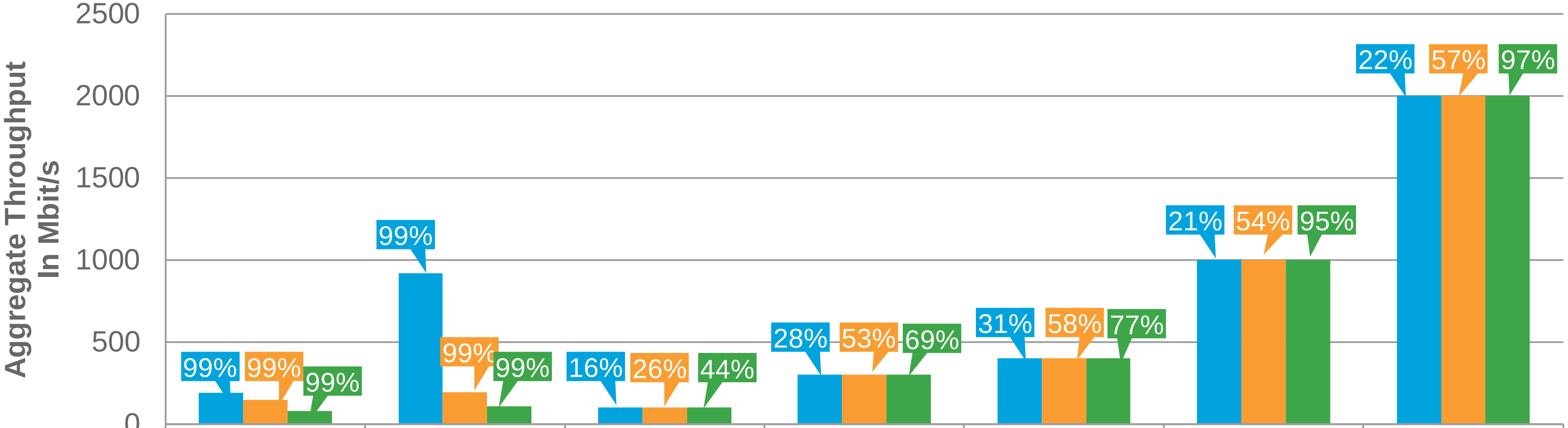
@89% CPU

@20% CPU

@54% CPU

@33% CPU

ISR Portfolio Performance Overview



	c880/c819	c890	4321	4331	4351	4431	4451
🟦 CEF only	190	920	100	300	400	1000	2000
🟧 NAT	148	192	100	300	400	1000	2000
🟩 IPSEC	80	106	100	300	400	1000	2000

*XX% CPU Utilization

Ideal as “CPE Lite”

Ideal as Service-Rich CPE

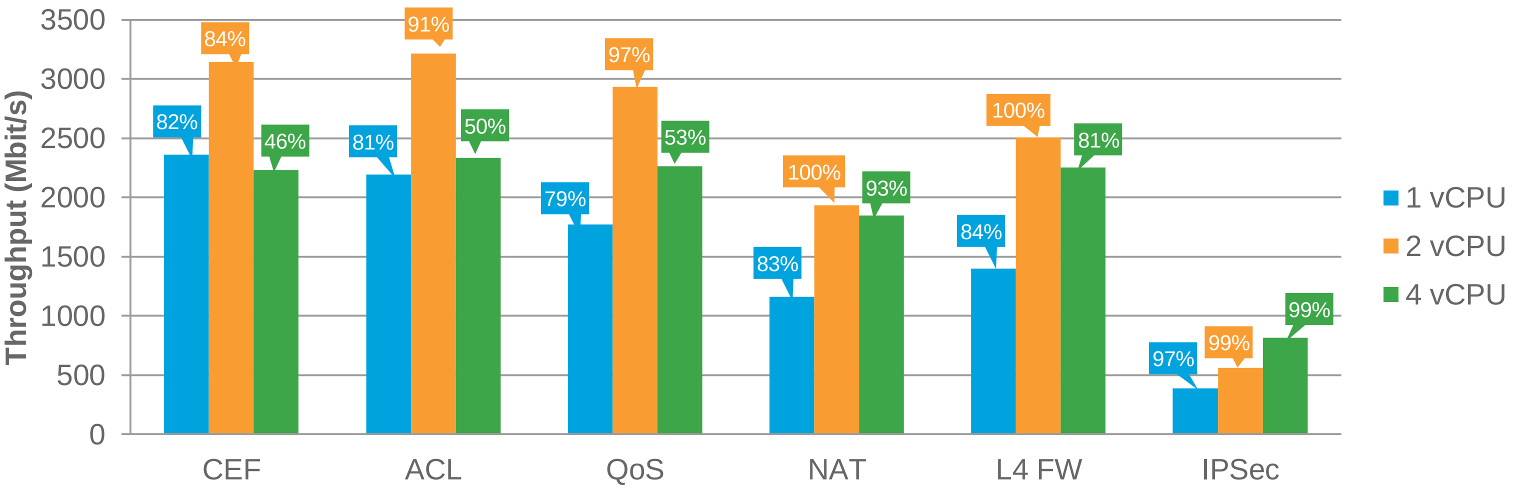
For your reference

How many Advanced Services can we pile on?

		FW + NAT + QoS + IPSEC		AVC		AVC + NBAR + QoS + IPSEC	
Platform	License	Mbps	CPU %	Mbps	CPU %	Mbps	CPU %
ISR 4321	50	42	31	45	32	42	54
	100	85	62	67	98	77	99
ISR 4331	100	86	53	97	54	91	86
	300	216	95	238	98	138	99
ISR 4351	200	175	87	200	91	124	99
	400	272	98	282	98	164	99
ISR 4431	500	322	99	279	99	174	99
	1000	545	99	482	99	302	99
ISR 4451	1000	545	99	491	99	305	99
	2000	959	99	850	99	540	99

CSR 1000V Performance

Single feature tests

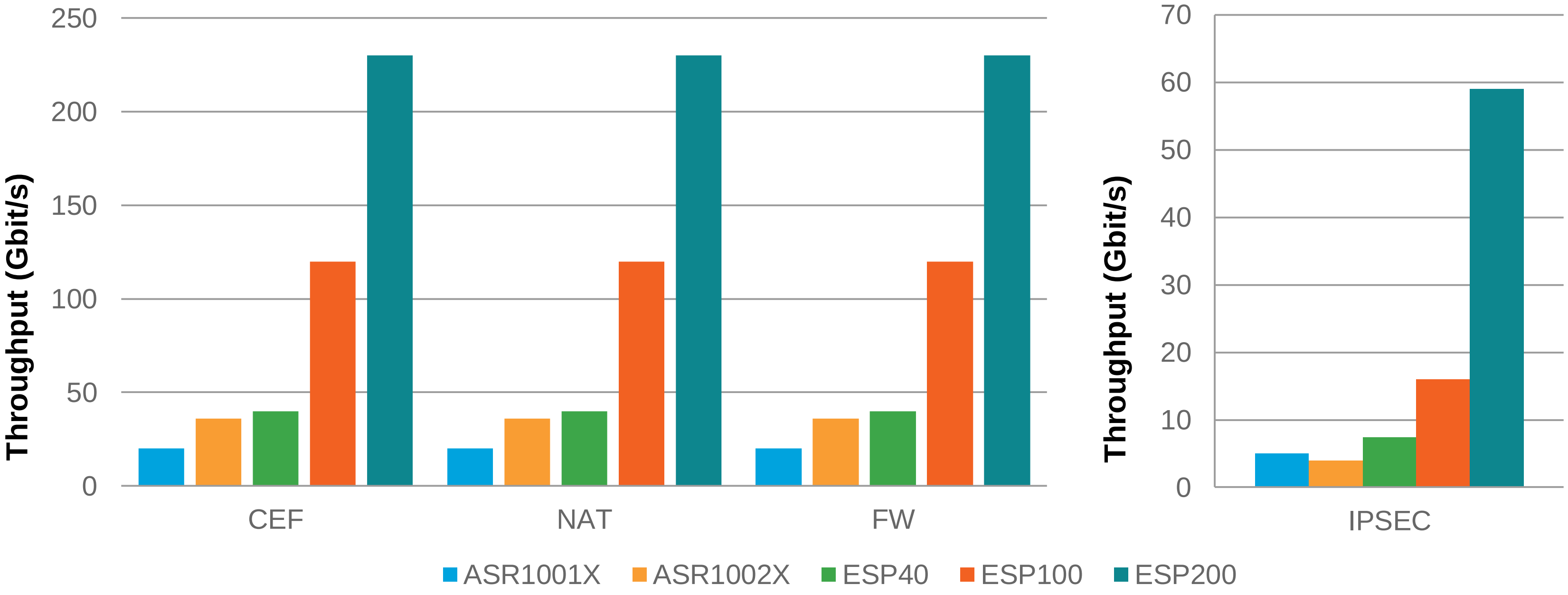


Challenge: single features don't load-balance well across multiple CPUs

Testing parameters:

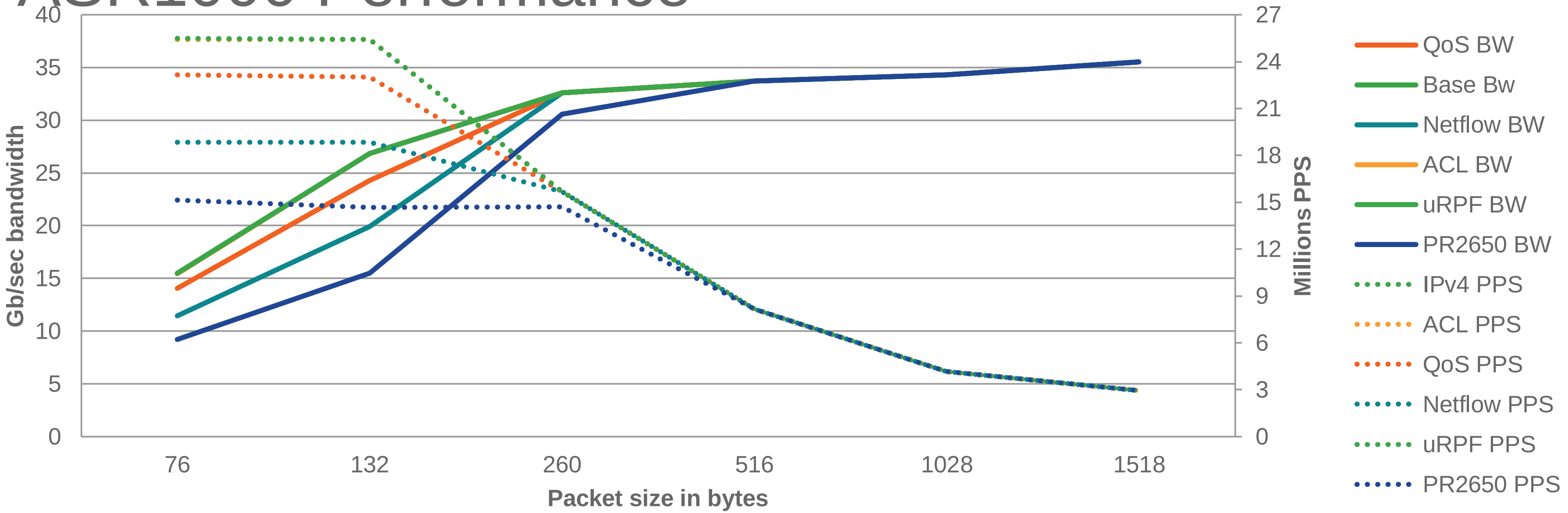
- IMIX traffic at 0.01% Drop Rate
- IOS-XE Image 3.14
- Platform: UCSC-C240-M3S with Intel Xeon E5-2643 v2 running ESXi 5.5
- VM-FEX results are on average 17% higher

ASR1000 Performance



ASR1000 performs to the advertised limits in all single feature tests except IPSEC

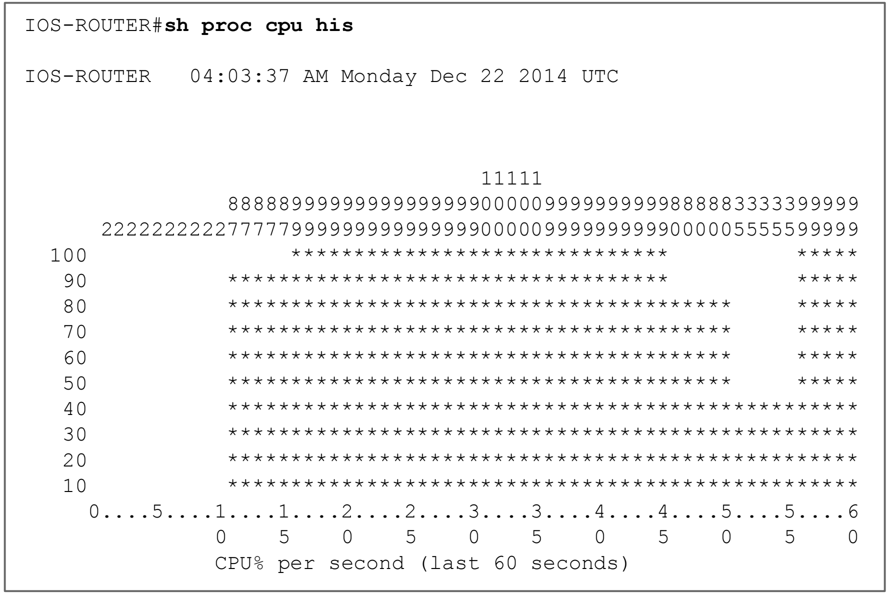
ASR1000 Performance



- Individual features have small impact with small packet sizes (76B and 132B)
- Individual features have very little impact at large packet sizes (above 260B)
- QFP has excellent behavior even with combined features for larger packet sizes!

How to verify current CPU Load

IOS Router with single CPU



```
IOS-ROUTER#sh proc cpu his

IOS-ROUTER   04:03:37 AM Monday Dec 22 2014 UTC



                                    11111
                88888999999999999999000009999999999888883333399999
      222222222277777999999999999999000009999999999000005555599999
  100                ******************************          *****
   90           ***********************************          *****
   80           ****************************************     *****
   70           ****************************************     *****
   60           ****************************************     *****
   50           ****************************************     *****
   40           **************************************************
   30           **************************************************
   20           **************************************************
   10           **************************************************
     0....5....1....1....2....2....3....3....4....4....5....5....6
               0    5    0    5    0    5    0    5    0    5    0
               CPU% per second (last 60 seconds)
```

Simple command showing overall CPU utilization

How to verify current CPU Load

IOS-XE Router with dedicated CP/SP and DP CPUs (ISR4400 & ASR1000 Series)

Classic command shows average CP/SP utilization

```
ISR4451#sh proc cpu his


                                       1111111111111111111111111
                                  555550000000000000000000000000
         55555               88888222220000000000000000000000000
100                                    ***********************
 90                                    ***********************
 80                                    ***********************
 70                                    ***********************
 60                                    ***********************
 50                               ****************************
 40                               ****************************
 30                               ****************************
 20                               ****************************
 10      *****               *********************************
   0....5....1....1....2....2....3....3....4....4....5....5....6
             0    5    0    5    0    5    0    5    0    5    0
             CPU% per second (last 60 seconds)
```

IOS-XE command shows per core utilization

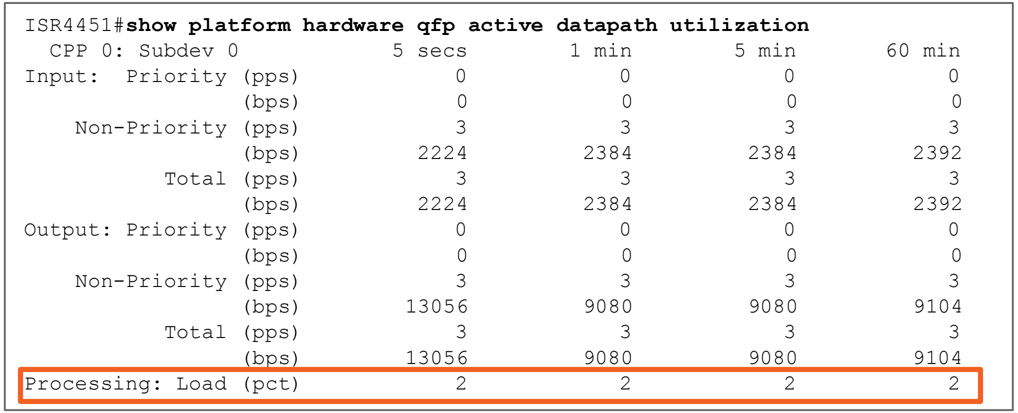
```
ISR4451#show platform software status control-processor brief
Load Average
 Slot  Status  1-Min  5-Min 15-Min
  RP0 Healthy   0.02   0.25   0.15

Memory (kB)
 Slot  Status    Total     Used (Pct)     Free (Pct) Committed (Pct)
  RP0 Healthy  3972052  3928184 (99%)    43868 ( 1%)   2584140 (65%)

CPU Utilization
 Slot  CPU   User System   Nice   Idle    IRQ   SIRQ IOwait
  RP0    0   1.10   1.10   0.00  97.70   0.00   0.10   0.00
         1   0.70   3.50   0.00  95.80   0.00   0.00   0.00
         2   0.30   1.70   0.00  98.00   0.00   0.00   0.00
         3   0.30   0.70   0.00  98.99   0.00   0.00   0.00
         4   0.50   0.30   0.00  99.20   0.00   0.00   0.00
         5   3.10   1.90   0.00  95.00   0.00   0.00   0.00
```

This is the **Control-Plane** utilization!

`show platform software status control-processor brief`

How to verify current CPU Load

IOS-XE Router with dedicated CP/SP and DP CPUs (ISR4400 & ASR1000 Series)



```
ISR4451#show platform hardware qfp active datapath utilization
  CPP 0: Subdev 0                5 secs        1 min        5 min       60 min
Input:  Priority (pps)                0            0            0            0
                 (bps)                0            0            0            0
    Non-Priority (pps)                3            3            3            3
                 (bps)             2224         2384         2384         2392
           Total (pps)                3            3            3            3
                 (bps)             2224         2384         2384         2392
Output: Priority (pps)                0            0            0            0
                 (bps)                0            0            0            0
    Non-Priority (pps)                3            3            3            3
                 (bps)            13056         9080         9080         9104
           Total (pps)                3            3            3            3
                 (bps)            13056         9080         9080         9104
Processing: Load (pct)                2            2            2            2
```

ISR4400 and ASR1000 have a second command to monitor data plane Cores:

`show platform hardware qfp active datapath utilization`

We only see average DP utilization, no breakdown per core.

Uni-dimentional scale for select features

For your reference

	ASR1001-X	ASR1002-X	ASR1001-HX	ASR1002-HX	RP2/ ESP20	RP2/ ESP40	RP2/ ESP100	RP2/ ESP200
VLANs (per port/SPA/system)	4K/8K/16K	4K/8K/16K	4K/8K/16K	4K/32K/ 32K	4K/32K/ 64K	4K/32K/ 64K	4K/32K/ 64K	4K/32K/ 64K
IPv4 routes	1M	3.5M	3.5M	1.0M	4M	4M	4M	4M
IPv6 routes	1M	3M	3M	0.5M	4M	4M	4M	4M
Sessions	not avail	29K	29K	24K	32K	64K	58K	58K
L2TP tunnels	4K	4K	4K	12K	16K	16K	16K	16K
Session setup rate (PTA/L2TP) in cps	150/100	150/100	150/100	100/50	150/100	150/100	150/100	150/100
BGP neighbors	8K	8K	8K	4K	8K	8K	8K	8K
OSPF neighbors	1K	2K	2K	1K	2K	2K	2K	2K
Unique QoS policy- / class-maps	1K/4K	4K/4K	4K/4K	1K/4K	4K/4K	4K/4K	4K/4K	4K/4K
ACL/ACE	4K/50K	4K/120K	4K/120K	4K/50K	4K/100K	4K/100K	4K/400K	4K/400K
Multicast groups	2000	4000	4000	1000	4000	4000	44K	44K
IPv4/IPv6 mroutes	64K	64K	64K	64K	100K	100K	100K	100K
Firewall sessions	2M	2M	2M	1M	2M	2M	6M	6M
NAT + firewall sessions	2M	1M	1M	500K	1M	1M	6M	6M
Netflow cache entries	2M	2M	2M	1M	2M	2M	2M	2M
VRFs	4K	8K	8K	1K	8K	8K	8K	8K
BFD sessions (offloaded)	4095	4095	4095	2047	4095	4095	4095	4095
AVC throughput (Mpps/Gbps)	not avail	6/20	6/20	2.5/10	3/20	3.4/20	3.6/40	not avail

Configuration specifics

Management Ethernet

- ASR1000 and ISR4000 have dedicated GigE Management Ethernet
  - Not usable for ‘normal’ traffic
  - Supports only basic ACLs
  - Most forwarding features do not work on this port (traffic not processed by QFP)
  - Intended for out of band router access—has SW support for rate limiting but that takes CPU cycles to drop packets
- Don’t connect to the ‘outside’ world
- Always configured in dedicated VRF
  - VRF cannot be removed from interface

TFTP Package to the RP from ROMMON

- SET the following variables within the ROMMON
  - RP does not have full RxBoot environment
    ROMMON is basically beefed up to support TFTP

```
rommon 2 > set
IP_SUBNET_MASK=255.255.0.0
TFTP_SERVER=2.8.54.2
TFTP_FILE=mcpude_12_18.bin
DEFAULT_GATEWAY=2.1.0.1
IP_ADDRESS=2.1.35.52
```

- Connect the GE management port on the RP to your management VLAN
- access the TFTP server where the “consolidated” package is located
- Issue the following command at ROMMON:

```
boot tftp:
```

- Image will be transferred directly to the RP DRAM for execution

Initial RP config in IOS for normal operation

- First thing that you will notice here is the default definition of “Mgmt-intf” VRF (case-sensitive), which includes RP Mgmt. Gi0 port

```
Router#sh ip vrf interfaces
Interface      IP-Address      VRF              Protocol
Gi0            unassigned      Mgmt-intf        up
```

- Assign the Gi0 interface an IP address, and set the default route in the VRF

```
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 <gateway_ip_address>
```

- Set the TFTP source interface to Gi0 for file transfers:

```
ip tftp source-interface gigabitEthernet 0
```

- Multiple options for file storage and booting when transferring images to the RP
  - `bootflash:` 1-8GB — recommended, larger on systems without harddisk:
  - `harddisk:` 40-80GB — not on all platforms

Configuring Management Ethernet

```
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
ip domain name vrf Mgmt-intf cisco.com
ip name-server vrf Mgmt-intf 171.70.168.183
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 172.27.55.129
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 ip address 172.27.55.210 255.255.255.128
 speed auto
 duplex auto
 negotiation auto
```

Filesystem Specifics

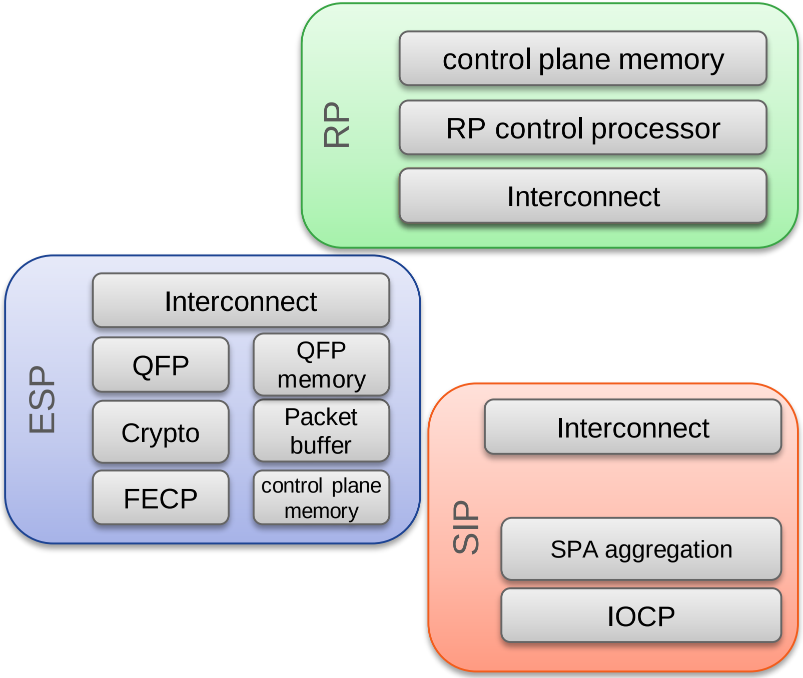
- All media shows up as type ‘disk’ regardless of type of media (SATA disk, USB flash, etc)
- `harddisk:` and `bootflash:` always formatted as ext2
- External `usb0:`, `usb1:` can be formatted as FAT16, FAT32, or ext2
- No support for multiple partitions at this time, only first partition on each device is visible
- `fsck` supported for all file system types; `/automatic` is implicit
- IOS does not control these devices directly
  - no flash driver in IOS
  - no SATA driver in IOS
  - Linux has the drivers, does the mount/umount under the covers

Core dumps, crashinfo

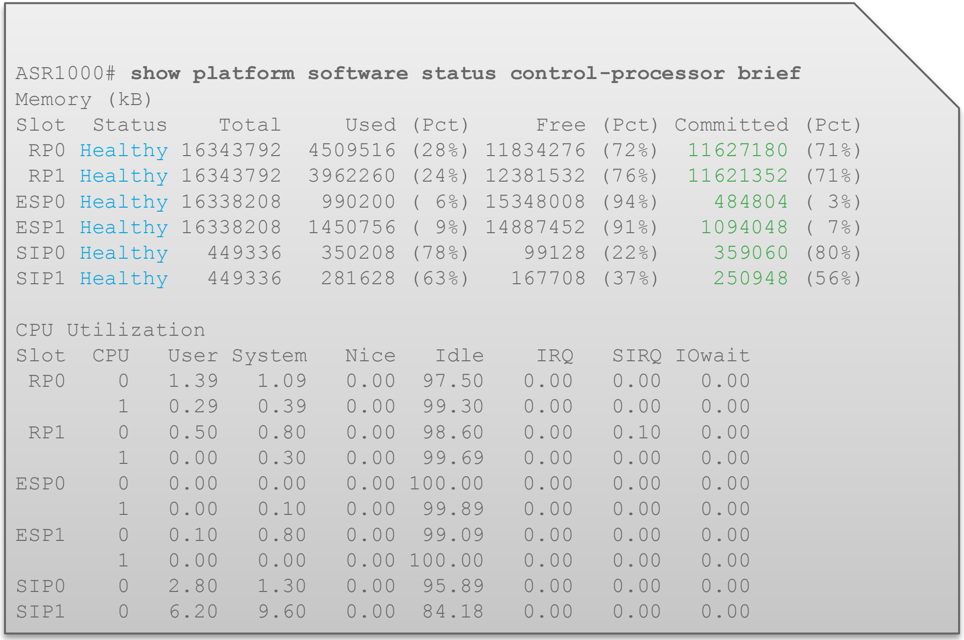
- Core dumps for all processes (IOS, cmand, fman_rp, …) and kernel all get written to
  - `harddisk:core/`
  - `or bootflash:/core when no harddisk is present.`
- File name pattern:
  `<hostname>_<FRU type>_<unit>_<process>_<time>.core.gz`
- IOSd generates crashinfo files into `bootflash:` when it crashes—like other IOS based platforms

IOS XE System Health Monitoring

- standard IOS CPU utilization and memory usage, e.g., “show process cpu” are not sufficient to determine ASR1000 health
- Monitoring the CPU and memory utilization of the following system elements is strongly recommended
  - RP CPU and Memory Utilization
  - ESP CPU and Memory Utilization
  - QFP Utilization
  - NOTE: On fixed configuration platforms it is critical to understand that the RP/ESP/SIP are actually sharing the same CPU and memory. Therefore checking the RP values reports for all three.
- Relevant MIBs:
  - CISCO-PROCESS-MIB
  - CISCO-ENTITY-QFP-MIB



IOS XE Control-Processor



```
ASR1000# show platform software status control-processor brief
Memory (kB)
Slot  Status    Total     Used (Pct)     Free (Pct) Committed (Pct)
 RP0 Healthy 16343792  4509516 (28%) 11834276 (72%)  11627180 (71%)
 RP1 Healthy 16343792  3962260 (24%) 12381532 (76%)  11621352 (71%)
ESP0 Healthy 16338208   990200 ( 6%) 15348008 (94%)    484804 ( 3%)
ESP1 Healthy 16338208  1450756 ( 9%) 14887452 (91%)   1094048 ( 7%)
SIP0 Healthy   449336   350208 (78%)    99128 (22%)    359060 (80%)
SIP1 Healthy   449336   281628 (63%)   167708 (37%)    250948 (56%)

CPU Utilization
Slot  CPU   User System   Nice   Idle    IRQ   SIRQ IOwait
 RP0    0   1.39   1.09   0.00  97.50   0.00   0.00   0.00
        1   0.29   0.39   0.00  99.30   0.00   0.00   0.00
 RP1    0   0.50   0.80   0.00  98.60   0.00   0.10   0.00
        1   0.00   0.30   0.00  99.69   0.00   0.00   0.00
ESP0    0   0.00   0.00   0.00 100.00   0.00   0.00   0.00
        1   0.00   0.10   0.00  99.89   0.00   0.00   0.00
ESP1    0   0.10   0.80   0.00  99.09   0.00   0.00   0.00
        1   0.00   0.00   0.00 100.00   0.00   0.00   0.00
SIP0    0   2.80   1.30   0.00  95.89   0.00   0.00   0.00
SIP1    0   6.20   9.60   0.00  84.18   0.00   0.00   0.00
```

- Key data to monitor for BB/ISG deployments:
  - RP/ESP Load Averages
  - Committed Memory
  - RP/ESP CPU Utilization
- All key data is retrievable via SNMP

Due to the Linux cache mechanism, Used and Free memory % are not always accurate. The cache gets counted as used when it is really potentially free. The critical item to view from this output is “Healthy” status. To see accurate Used/Free/Cache usage use the ‘`monitor`’ cmd on the next slide.

Committed Value on ASR differs from ISR platform.
ISR – Represents actual memory in use.
ASR – Represent max potential memory usage.

IOS XE Linux top

- Captures actual ‘`top`’ output from RP/ESP/SIP. This can be used to determine Used/Free/ Cache memory usage and CPU usage. More accurate than the ‘status control-processor brief’ command but obviously more in-depth.

```
ASR# show platform software process slot RP active monitor cycles 2 interval 10
<snip>
top - 14:19:18 up 1 day, 22:09,  0 users,  load average: 0.80, 0.53, 0.42
Tasks: 227 total,   2 running, 225 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.7%us,  0.6%sy,  0.0%ni, 97.7%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   8067244k total,  2697464k used,  5369780k free,   169760k buffers
Swap:        0k total,        0k used,        0k free,  1394588k cached
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
29060 root      20   0 99848  91m 5656 S    2  1.2   0:10.87 smand
29638 root      20   0 4820m 641m 206m S    0  8.1  11:08.68 linux_iosd-imag
 9055 root      20   0  178m  53m  31m R    0  0.7   2:56.29 mcpcc-lc-ms
<snip>
```

- It is IMPORTANT to use 2 cycles at a reasonable interval (5-10sec) and IGNORE the CPU values from the first output. The first output averages over a very small timeframe and the CPU reports are invalid. Only the 2nd cycle output averages over the desired interval and provides accurate results. This is a linux limitation, not and IOS issue.

- In this example, Used Mem = 2.7G, but 1.4G of this is cached. Therefore available memory = Free 5.4G + Cache 1.4G = 6.8G

IOS XE BQS queue and schedule count

```
ASR1001# show platform hardware qfp active infrastructure bqs status
BQS-RM Status :
=============================================
Current SW Memory Size:              4000
Object Counts:
  Recycle Object Count:                91
  Recycle Schedule Count:              15
  Recycle Queue Count:                 52
  # of Active Queues:                 144
  # of Active Schedules:               56
  # of Active Roots:                   14
<snip>
```

- This command has a large amount of output related to QoS actions and events.
- The elements to look for are in the summary table listing the number of active queues and schedules in the system.

IOS XE QoS sorter memory

```
ASR1001# show platform hardware qfp active infrastructure bqs sorter memory available
Level:Class       Total    Available   Remaining
=============    ======    =========   =========
ROOT:ONCHIP          64           64        100%
ROOT:COS_L2         448          448        100%
ROOT:NORMAL           0            0          0%
BRANCH:ONCHIP       128          122         95%
BRANCH:COS_L2       384          384        100%
BRANCH:NORMAL     15872        15872        100%
STEM:ONCHIP         992          877         88%
STEM:COS_L2        1024         1024        100%
STEM:NORMAL      260064       259934         99%
```

- Show memory utilization by all the active elements in the BQS system, primarily used for QoS.
- The last line “`STEM:NORMAL`” is the primary element to monitor. Keeping the % Remaining at a reasonable level (> 10%) for dynamic system events.
- Note: This command is dependent on an actual BQS ASIC being present and as such is not operational on ISR or CSR platforms.

IOS XE QFP memory statistics

- This command shows the specific QFP memory usage.
- The SRAM memory is fixed and should never change.
- The DRAM memory is the main memory used, when this reaches near 100% the IRAM memory will increase to handle the extra requirements.
- The IRAM should be monitored for a reasonable free available to handle dynamic events. (20-30% free)
- This memory is used for ALL the features that are processed by the QFP.

```
ASR1001# show platform hardware qfp active infrastructure exmem statistics
QFP exmem statistics
Type: Name: DRAM, QFP: 0
  Total: 268435456
  InUse: 96961536
  Free: 171473920
  Lowest free water mark: 171438080
Type: Name: SRAM, QFP: 0
  Total: 32768
  InUse: 14880
  Free: 17888
  Lowest free water mark: 17888
Type: Name: IRAM, QFP: 0
  Total: 134217728
  InUse: 7027712
  Free: 127190016
  Lowest free water mark: 127190016
```

IOS XE datapath utilization

- This output shows the actual processing load at the QFP from all interfaces.
- The Input/Output Priority/ Non-Priority pps and bps counts should be the aggregate from all interfaces.
- The Processing Load (pct) needs to be monitored. A consistent load below 90% is expected. Once the load goes above 95% there can be ingress packet drops due to processing backpressure (Flow-Control).
- Note that the QFP also has to process all the inter-chassis control packets also which adds to the processing load independent of actual traffic.

```
ASR1001# show platform hardware qfp active datapath utilization
  CPP 0: Subdev 0            5 secs        1 min        5 min       60 min
Input:  Priority (pps)            1            1            1            1
                 (bps)          680         1160         1144         1152
    Non-Priority (pps)            1            4            4            4
                 (bps)          584         3040         2992         3000
           Total (pps)            2            5            5            5
                 (bps)         1264         4200         4136         4152
Output: Priority (pps)            0            1            1            1
                 (bps)          496          864          856          856
    Non-Priority (pps)            1            4            4            4
                 (bps)         3184         9168         9064         9200
           Total (pps)            1            5            5            5
                 (bps)         3680        10032         9920        10056
Processing: Load (pct)            0            0            0            0
```

IOS XE datapath utilization summary

```
ASR1001# show platform hardware qfp active datapath utilization summary
   CPP 0:                    5 secs       1 min       5 min      60 min
Input:     Total (pps)        7262        7264        7264        2875
                 (bps)    59458736    59462160 29265240496  4891165824
Output:    Total (pps)           4           5           5           2
                 (bps)        8600       15536       15840        6168
Processing: Load (pct)           1           1           1           1
```

- This output shows the same details as the previous command but combines multiple CPP subdev and priority/non-priority details into a shorter version. Mainly useful on ESP100, ESP200 where multiple QFP ASICs are used.

IOS XE datapath drops

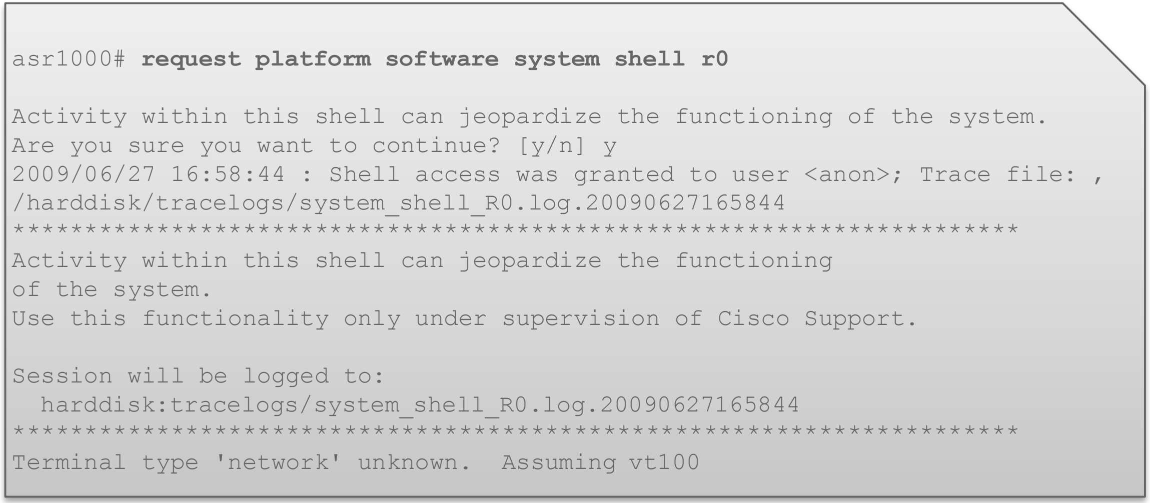
```
ASR1001# show platform hardware qfp active statistics drop
-----------------------------------------------------------------------
Global Drop Stats                         Packets                 Octets
-----------------------------------------------------------------------
Ipv4NoRoute                                    10                    644
Wred                                           19                   1392
```

- This output shows the reason for any drops in the QFP complex. There are many reasons for drops but he output command only shows non-zero statistics (use `all` keyword to see all reasons)

- If there are drops outside the QFP they will show up in other places
  - “`show interface`” output
    - queue overload
  - “`show controller`” output
    - due to flow-control because of the ingress overflow

IOS XE platform shell

- Used when there is not enough information from the IOS CLI
- Fully functional shell as ‘root’
  - you can see/break everything from here
- Shell session is recorded and send to syslog when done



```
asr1000# request platform software system shell r0

Activity within this shell can jeopardize the functioning of the system.
Are you sure you want to continue? [y/n] y
2009/06/27 16:58:44 : Shell access was granted to user <anon>; Trace file: ,
/harddisk/tracelogs/system_shell_R0.log.20090627165844
**********************************************************************
Activity within this shell can jeopardize the functioning
of the system.
Use this functionality only under supervision of Cisco Support.

Session will be logged to:
  harddisk:tracelogs/system_shell_R0.log.20090627165844
**********************************************************************
Terminal type 'network' unknown.  Assuming vt100
```

- “`service internal`” and “`platform shell`” are required on all platforms and some may also require a license to be installed.
- Remember that here be dragons and you taste good with ketchup.

Command List

- Summary of RP/ESP/SIP CPU and Memory

```
show platform software status control-processor brief
```

- Linux level RP/ESP/SIP CPU,Memory and Process list (top command)

```
show platform software process slot RP active monitor cycles 2 interval 10
```

- QoS Queue/Scheduler counts

```
show platform hardware qfp active infrastructure bqs status
```

- QoS Resource usage (only on systems with BQS ASIC, ie ASR1000)

```
show platform hardware qfp active infrastructure bqs sorter memory available
```

- QFP Memory Usage

```
show platform hardware qfp active infrastructure exmem statistics
```

- QFP Datapath Processing

```
show platform hardware qfp active datapath utilization <summary>
```

ESP 100/200 show command differences

- `show platform hardware qfp active infrastructure exmem statistics`
  - On ESP 100 the SRAM reports 0 values (no SRAM)
- `show platform hardware qfp active datapath utilization`
  - must be executed multiple times on ESP100/200, once for each 2nd gen QFP
  - Use the summary option to see mulitple QFP ASIC details compressed into one output.
- `show platform hardware qfp active infrastructure bqs sorter memory [active, free, available, utilization]`
  - different output due to two 2nd gen QFP but same fundamental info for active, free, available
  - Utilization is not implemented for 2nd gen QFP
- `show platform hardware qfp active infrastructure bqs status`
  - slightly different, ESP100/200 does not report memory size

IOS XE packet tracing

- Introduced in XE3.10 as part of the IOS-XE serviceability initiative.
- Pactrac provides visibility into the treatment of packets of an IOS-XE platform with simple to use commands. It is intended to be used externally (TAC, customers) and internally (DE, DT) to troubleshoot, diagnose or gain a deeper understanding of the actions taken on a packet during packet processing.
- Pactrac limits its inspection to the packets matched by the debug platform condition statements making it a viable option even under heavy traffic situations seen in customer environments.
- Three specific levels of inspection are provided by pactrac: accounting, per packet summary and per packet path data. Each level adds a deeper look into the packet processing at the expense of some packet processing capability.

Packet-Trace: Configuration Example

- The following shows how one would trace the first 128 packets entering GigabitEthernet0/0/0 including FIA trace and a copy of up to the first 2048 octets of the input packet.

```
debug platform condition interface g0/0/0 ingress
debug platform packet-trace enable
debug platform packet-trace packet 128 fia-trace
debug platform packet-trace copy packet input size 2048
debug platform condition start
```

Packet-Trace: Configuration Highlights

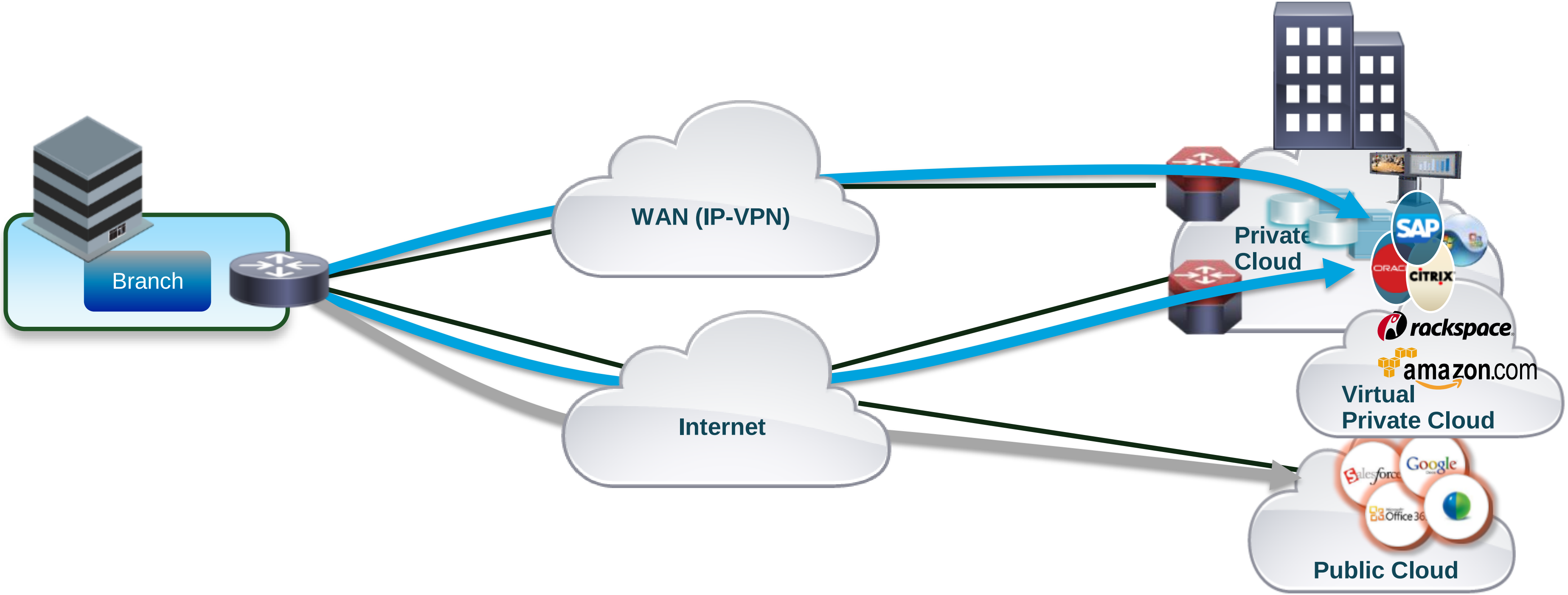
- Be mindful of how much QFP DRAM memory a config needs and how much memory is available
- memory needed = (stats overhead) + num pkts * (summary size + path data size + copy size)
- Stats overhead and summary size are fixed and about 2KB and 128B respectively
- Path data size and copy size (in/out/both) are user configurable
- Configure as much detail as you want…more detail…more performance impact for matched packets (reading/writing memory costs!)
- Each config change temporarily disables pactrac and clears counts/buffers
- "Cheap" way of '`debug plat cond stop`', '`clear plat pack stats`' and '`debug plat cond start`'
- Some configs require a 'stop' in order to display summary or per packet data
- Currently circular and drop tracing
- REMINDER: Conditions define where and when filters are applied to a packet

Packet-Trace: Show Commands

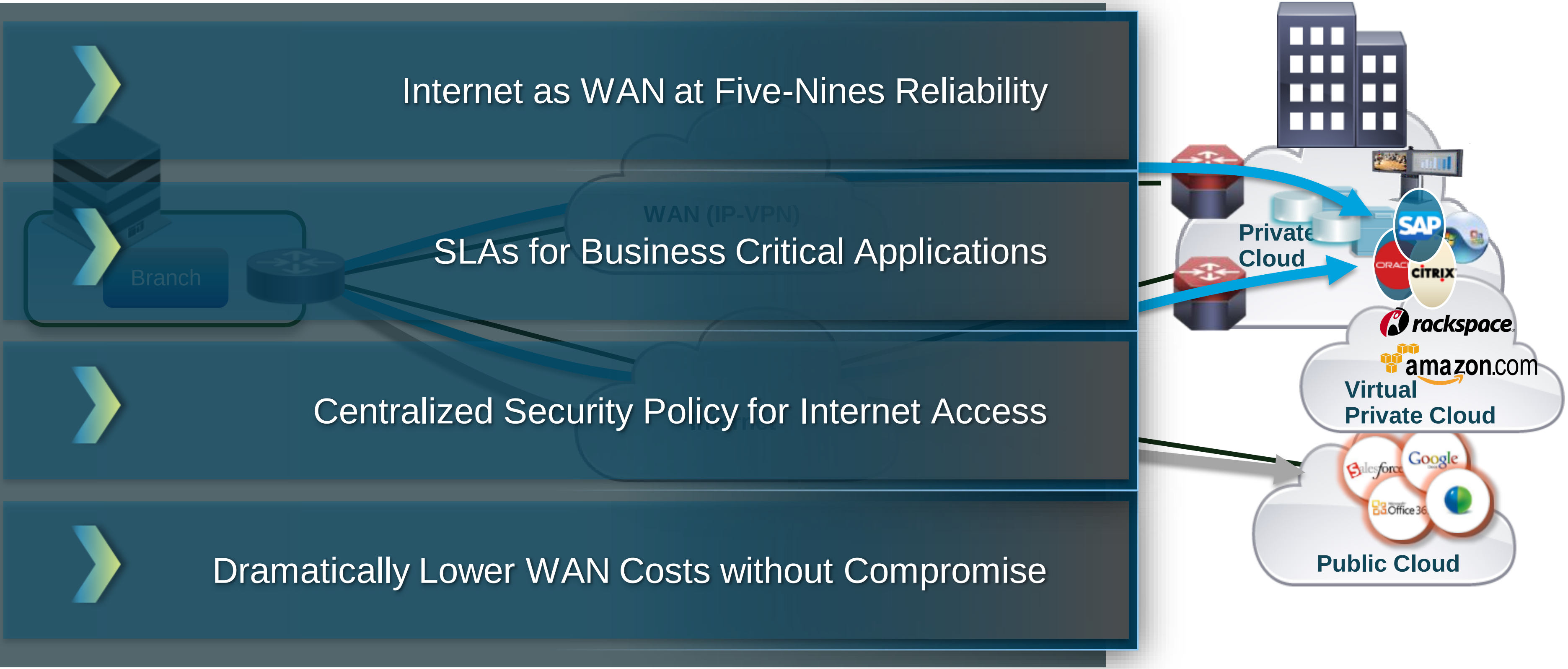
- Show commands are used to display pactrac configuration and each level of data:
- `show platform packet-trace configuration`
  - Displays packet-trace configuration including any defaults
- `show platform packet-trace statistics`
  - Displays accounting data for all pactrac packets
- `show platform packet-trace summary`
  - Displays summary data for the number of packets specified by debug platform packet-trace packet
- `show platform packet-trace packet { all | <pkt-num>} [decode]*`
  - Displays all path data for all packets or the packet specified
- Decode attempts to display packets captured by debug platform packet-trace copy in user friendly way
- Only a few protocol headers are supported initially (ARPA, IP, TCP, UDP, ICMP)
  - decode was introduced in XE3.11

Use cases

IOS XE for Intelligent WAN

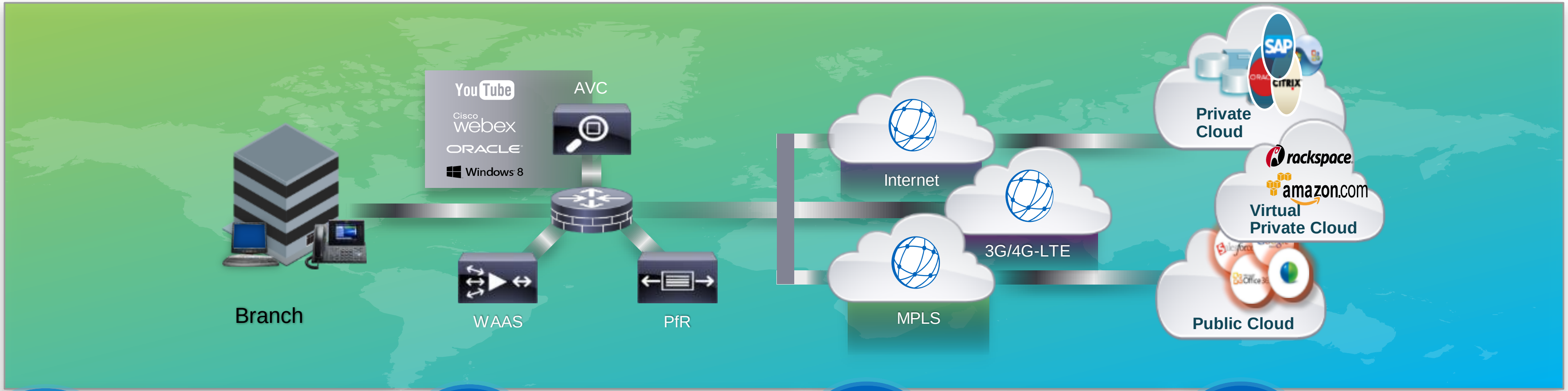


Intelligent WAN – Leveraging the Internet



- Internet as WAN at Five-Nines Reliability
- SLAs for Business Critical Applications
- Centralized Security Policy for Internet Access
- Dramatically Lower WAN Costs without Compromise

Intelligent WAN Solution Components



Transport Independent

- Consistent operational model
- Simple Provider migrations
- Scalable and Modular design
- **DMVPN** IPsec overlay design

Intelligent Path Control

- Application best path based on delay, loss, jitter, path preference
- Load Balancing for full utilization of all bandwidth
- Improved network availability
- **Performance Routing (PfR)**

Application Optimization

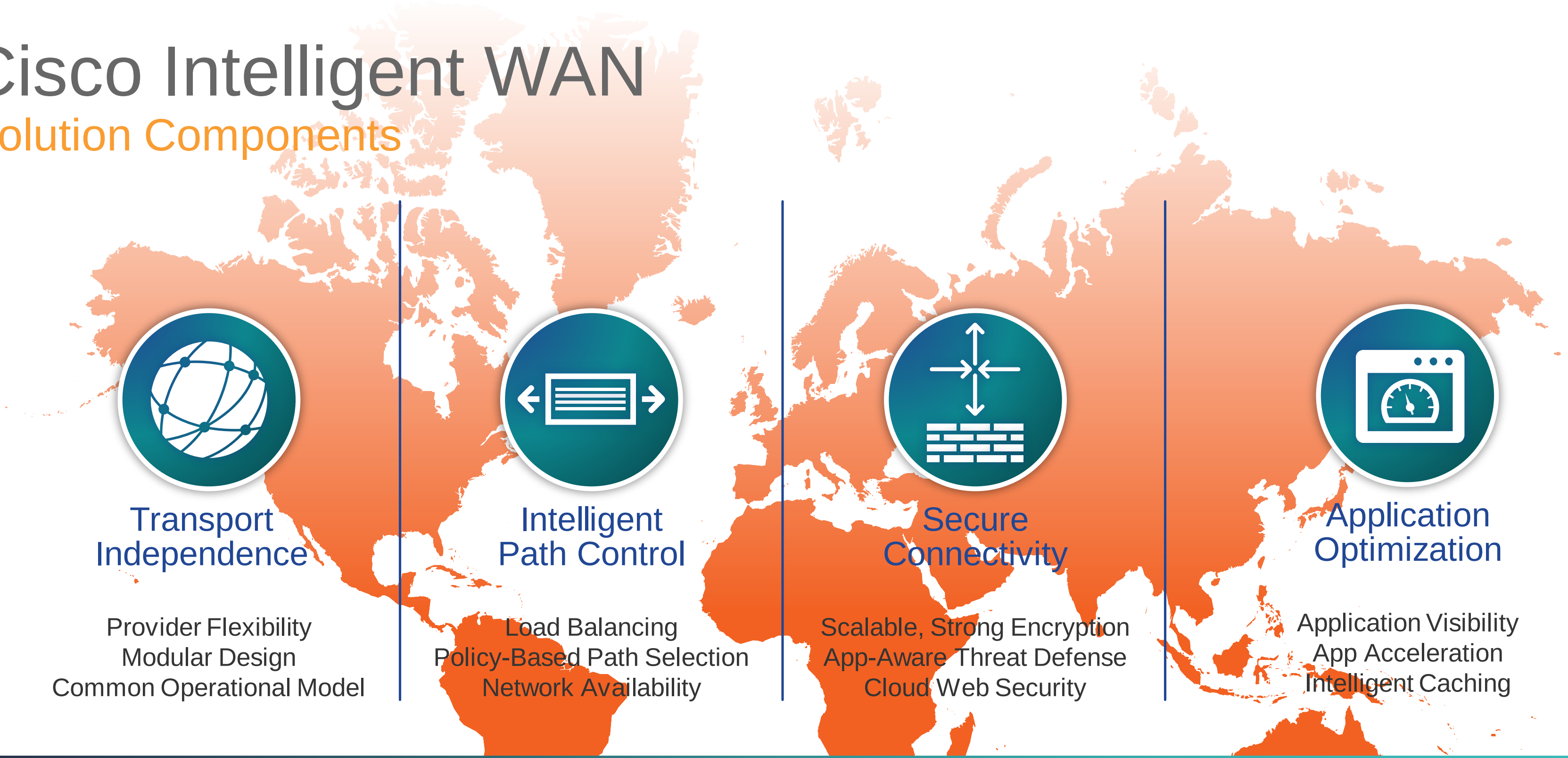
- Application monitoring with **Application Visibility & Control (AVC)**
- Application Acceleration and bandwidth savings with **WAAS**

Secure Connectivity

- **Certified** strong encryption
- Comprehensive threat defense with **ASA & IOS Firewall/IPS**
- **Cloud Web Security (CWS)** for scalable secure direct Internet access

Cisco Intelligent WAN

Solution Components



Transport Independence

Provider Flexibility
Modular Design
Common Operational Model

Intelligent Path Control

Load Balancing
Policy-Based Path Selection
Network Availability

Secure Connectivity

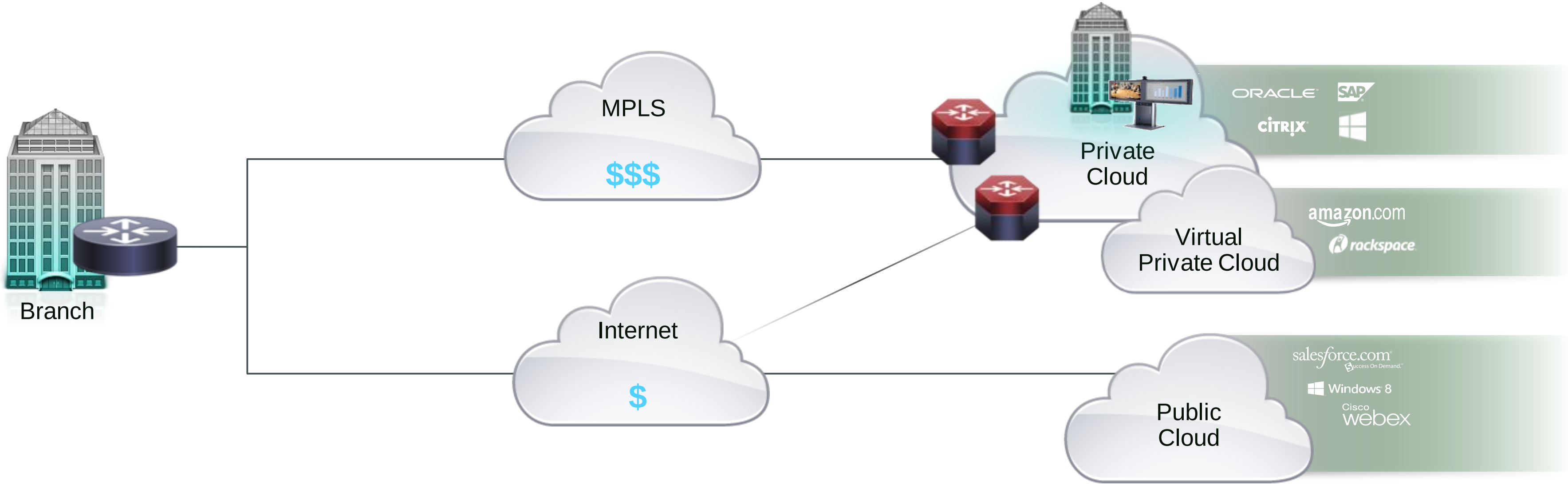
Scalable, Strong Encryption
App-Aware Threat Defense
Cloud Web Security

Application Optimization

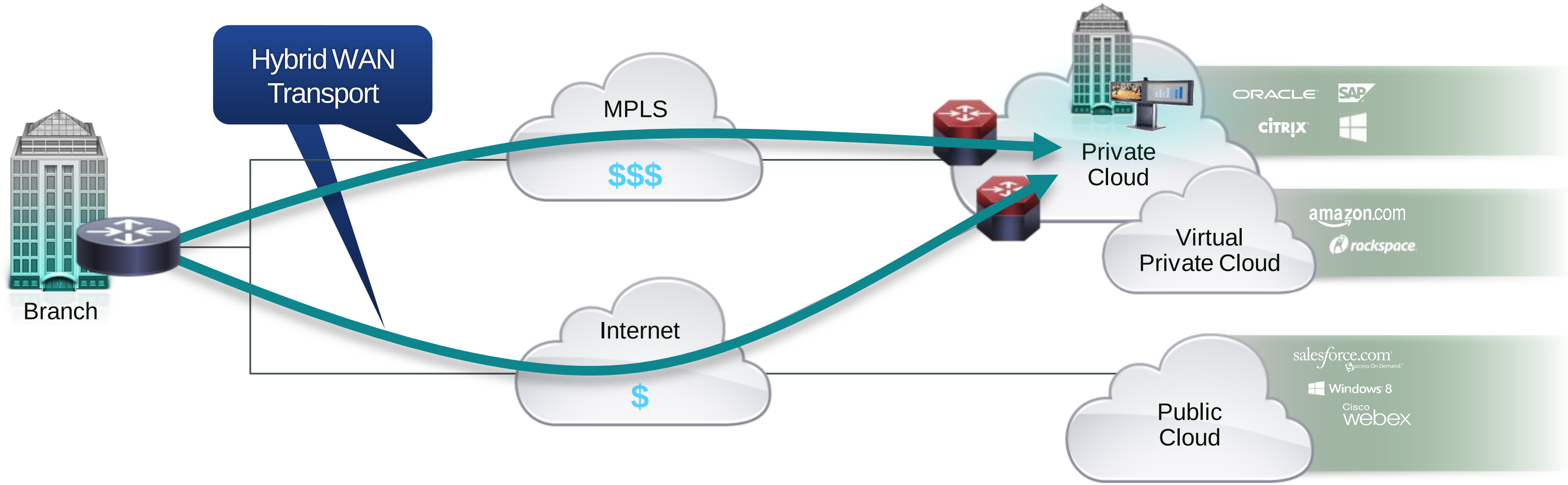
Application Visibility
App Acceleration
Intelligent Caching

Application Experience / IT Simplicity / Lower WAN Costs

SD WAN (IWAN)

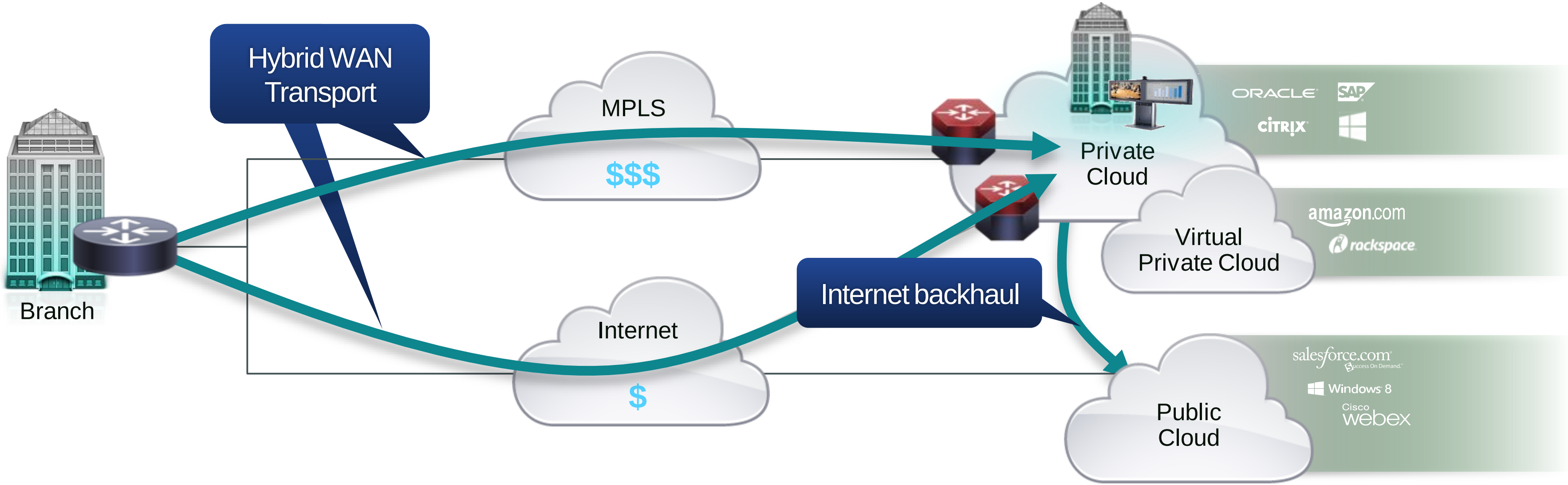

MPLS
$$$
Internet
$
Branch
Private Cloud
Virtual Private Cloud
Public Cloud
ORACLE
SAP
CITRIX
amazon.com
rackspace
salesforce.com
Success On Demand.
Windows 8
Cisco webex

SD WAN (IWAN)



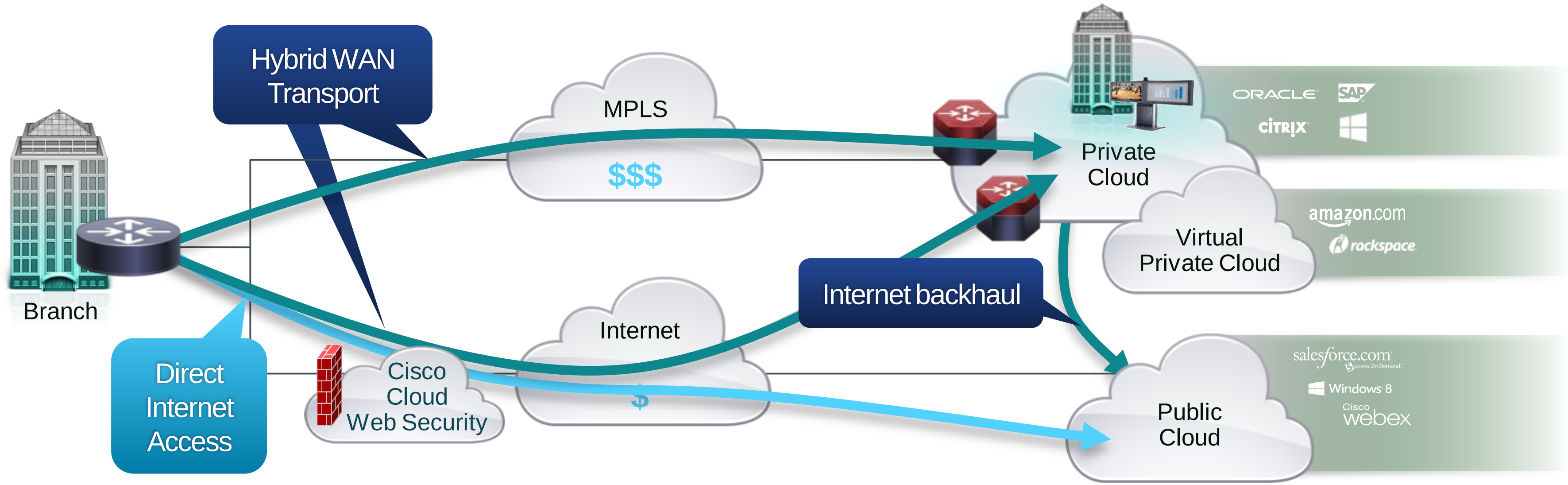
- ✓ Secure **WAN transport** across MPLS and/or Internet for private cloud / DC access

SD WAN (IWAN)



✓ Secure **WAN transport** across MPLS and/or Internet for private cloud / DC access

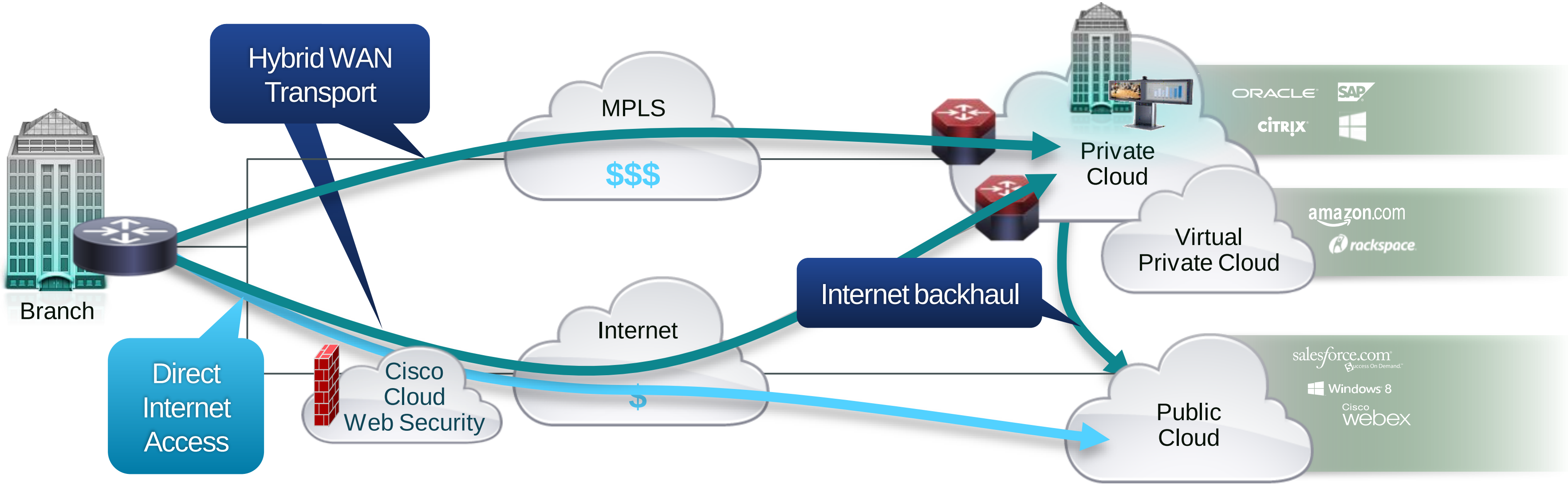
SD WAN (IWAN)



✓ Secure **WAN transport** across MPLS and/or Internet for private cloud / DC access

✓ Leverage **local Internet** path for public cloud and Internet access

SD WAN (IWAN)



✓ Secure **WAN transport** across MPLS and/or Internet for private cloud / DC

✓ Leverage **local Internet** path for public cloud and Internet access

Increase WAN Capacity

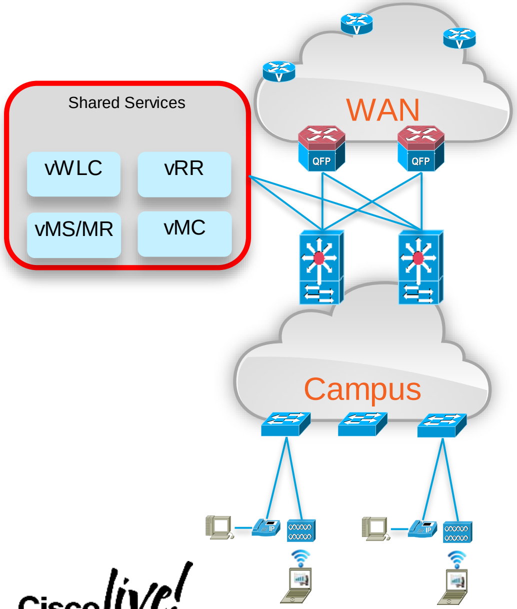
Improve App Performance

Scale Security at the Branch

Currently Deployed Virtualization Solutions

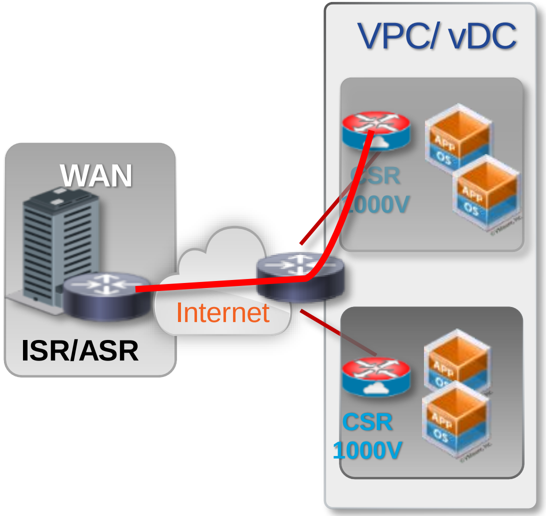
Control Plane

- RR, LISP MS/MR..

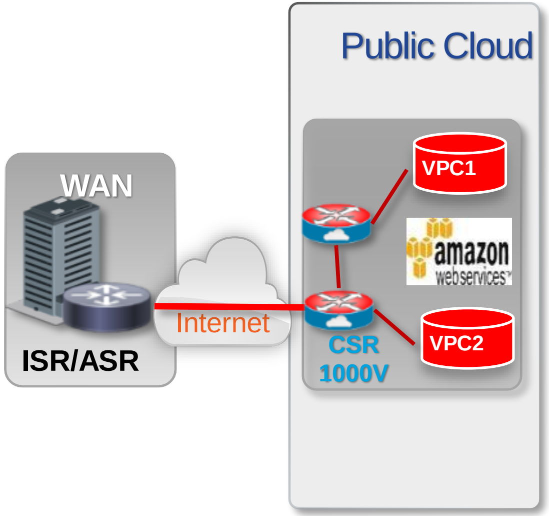


(Virtual) Private Cloud / DC

- CE/PE Functionality



- Public Cloud



ASR 1000 as Services PE

Functions

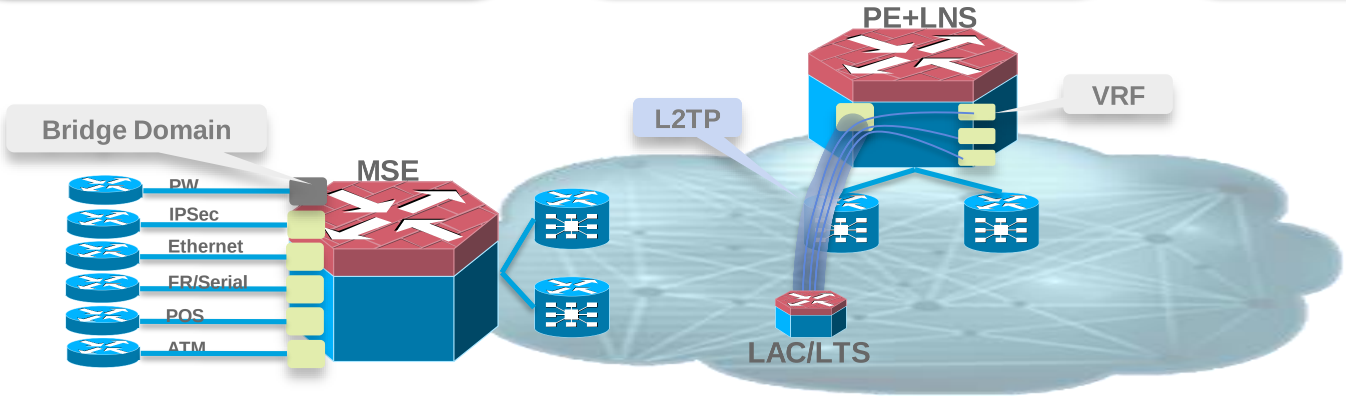
- ASR 1006 / 1013 as MSE
  L3VPN / L2VPN / VPLS
  CsC, Extranet
- ASR 1002 –X as PE+LNS
- Hierarchical QoS
- High-Availability / ISSU
  FRR, Fast Convergence
  PE-CE BFD + NSR

Services

- Dual-stack
- Multicast / mVPN
- EVC
- RA-MPLS with MLP
- Firewall / CGN / NAT64
- IPSec
- Routed PW into VRF
- EOAM / SLA

Scalability*

- 2.5 – 100 Gbps
- 4M Routes
- 8000 VRF
- 16000 PW / L2TP
- 8000 PPP / 1000 MLP
- 2000 eBGP + NSR



* uni-dimensional. Scalability may vary with feature combinations

ASR 1000 as Internet Edge

Functions

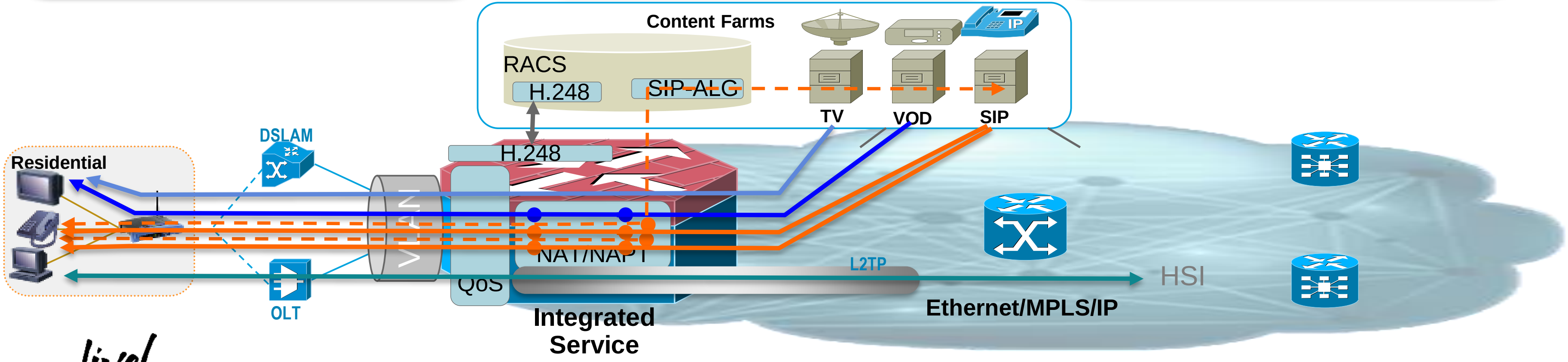
- ASR 1006 RP1 / ESP10 providing IP Edge functions
  - Over 3000 systems deployed
- 3-play IP Edge
  - No MPLS
- Extreme focus on HA & ISSU
  - Multiple ISSU upgrades executed

Services

- HSI with LAC model
  - 1/2/4 Gbps services
- VoIP and VVoIP using distributed SBC
- Multicast with 3 levels of QoS
- GEC VLAN loadbalancing

Scalability*

- Up to 10K residential subs
- 64000 SIP sessions
- 1000 MLD / IGMP
- 100K mroutes
- Oversubscribed System
- Redundant links



ASR 1000 as Enterprise Edge

Functions

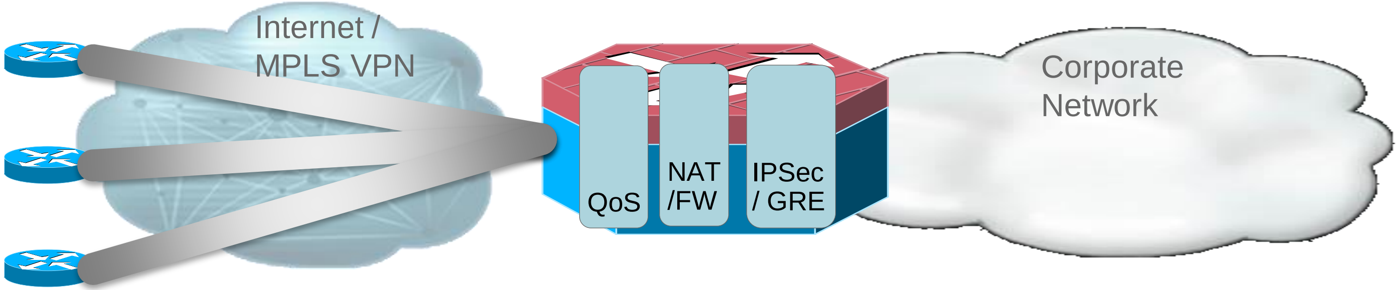
- ASR 1001 / 1002 / 1006 as
  WAN Edge
  Secure VPN functions
  Internet Edge
- H-QoS
- IPSec: S2S, DMVPN, GETVPN

Services

- FNF
- PFR
- Multilink FR / PPP
- VRF-aware PBR
- NAT / Firewall
  With inter-chassis redundancy
- USGv6
- Trustsec
- Application Visibility & Control

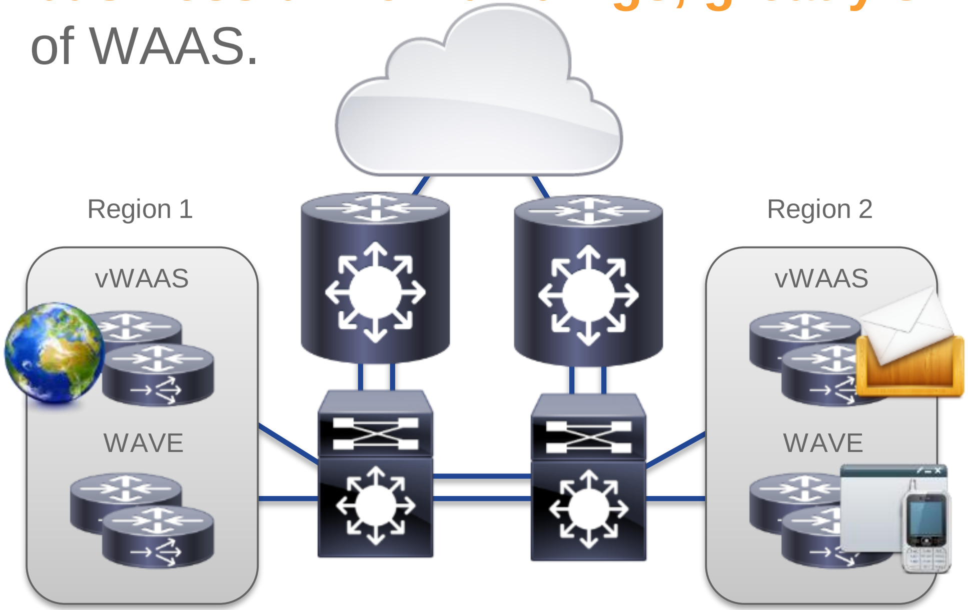
Scalability

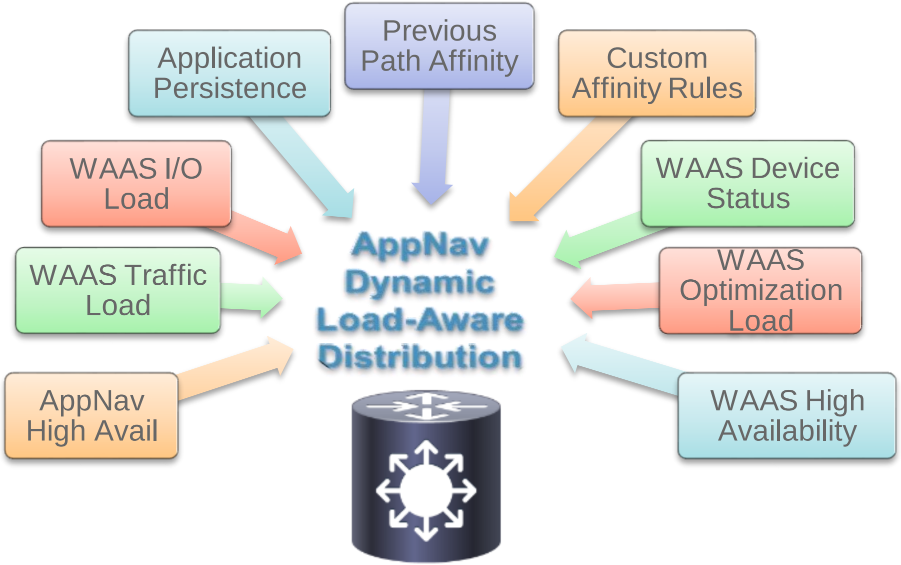
- 4000+ IPSec tunnels
- 60K ACEs in 4K ACLs
- 4K policy maps in 4K class maps
- 4K GRE



ASR1000 with AppNav-XE

Virtualize WAN optimization resources into **pools of elastic resources** with **business driven bindings, greatly simplify deployment** and management of WAAS.





Application Persistence

Previous Path Affinity

Custom Affinity Rules

WAAS I/O Load

WAAS Device Status

WAAS Traffic Load

WAAS Optimization Load

AppNav High Avail

WAAS High Availability

AppNav Dynamic Load-Aware Distribution

Summary

IOS XE summary

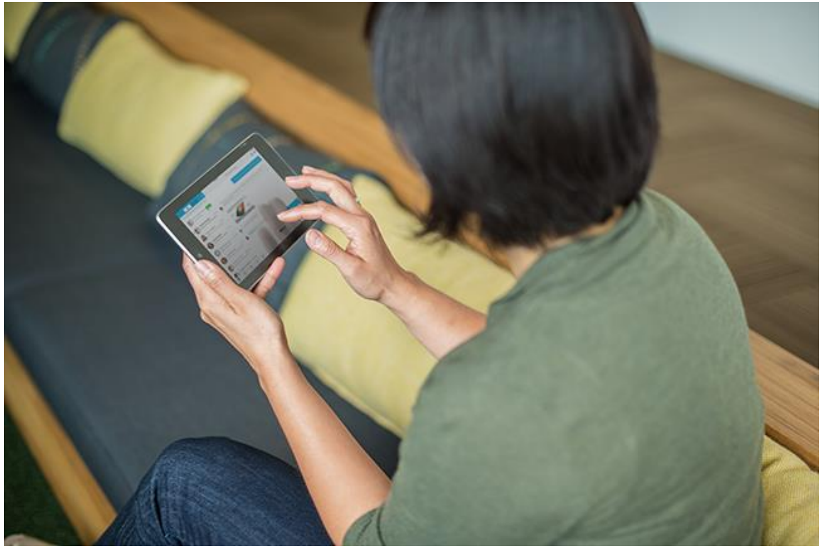
- IOS XE is an evolution of IOS
  - provides operational continuity
  - configurations move forward
  - IOS protocol troubleshooting moves forward
- Data / control / service plane separation
  - Functionality isolation, DOS protection
  - Improved and predictable performance
  - Cost efficiencies

IOS XE summary

- Operational excellence
  - QoS, High Availability, easy service enablement
- Platform management
  - Multiple processors, memories, busses to be monitored
- Common code and feature sets across multiple locations in the network
  - Eases deployments, decreases incompatibilities

Complete Your Online Session Evaluation

- Please complete your Online Session Evaluations after each session
- Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt
- All surveys can be completed via the Cisco Live Mobile App or the Communication Stations



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

Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Lunch & Learn
- Meet the Engineer 1:1 meetings
- Related sessions

Thank You

Cisco live!

February 20 - 24, 2017 · Berlin

Your Time Is Now