Cisco live!

What You Make Possible

BUILT FOR THE HUMAN NETWORK

CISCO

Cisco Nexus 7000 Hardware Architecture

BRKARC-3470

Session Goal

- To provide you with a thorough understanding of the Cisco Nexus™ 7000 switching architecture, supervisor, fabric, and I/O module design, packet flows, and key forwarding engine functions
- This session will not examine NX-OS software architecture or other Nexus platform architectures
- Related sessions:
  - BRKDCT-2204 Nexus 7000/5000/2000/1000v Deployment Case Studies
  - BRKIPM-3062 Nexus Multicast Design Best Practices
  - BRKDCT-2121 VDC Design and Implementation
  - BRKDCT-2048 Deploying Virtual Port Channel in NX-OS
  - BRKARC-3472 NX-OS Routing & Layer 3 Switching
  - BRKDCT-2081 Cisco FabricPath Technology and Design
  - TECDCT-3297 Operating and Deploying NX-OS
  - BRKCRS-3144 Troubleshooting Cisco Nexus 7000 Series Switches
  - LTRCRT-5205 Configuring Nexus 7000 Virtualization Lab
  - LTRDCT-1142 FabricPath Deployment in the Data Center Lab

What Is Nexus 7000?

Data-center class Ethernet switch designed to deliver high-availability, system scale, usability, investment protection

I/O Modules

Supervisor Engine

Chassis

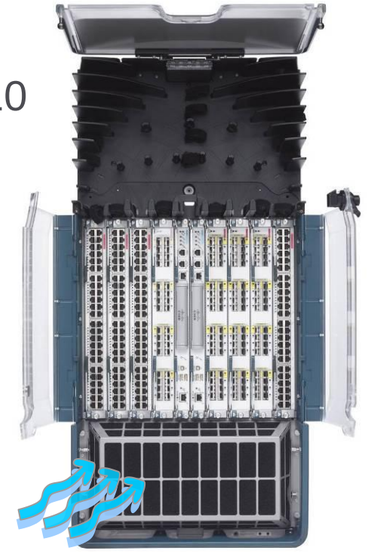
Fabrics

Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

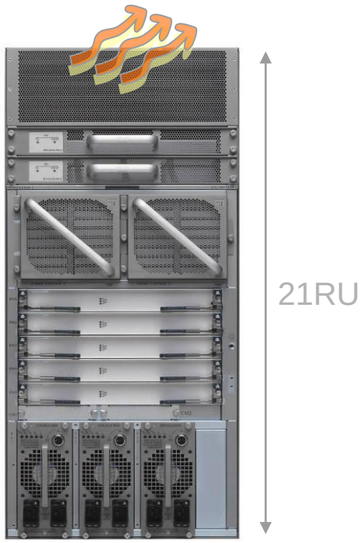
Nexus 7000 Chassis Family

Nexus 7010



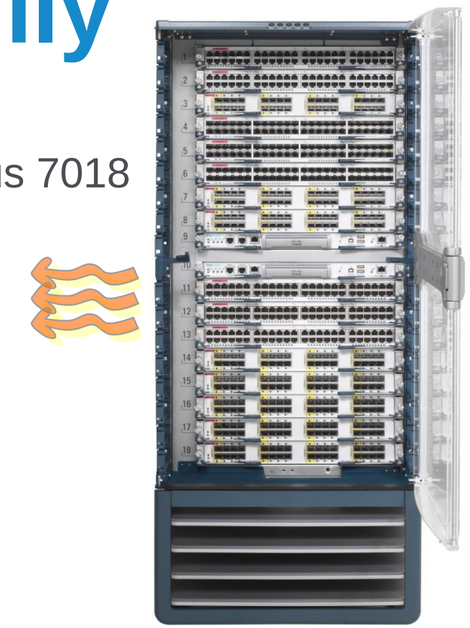
Front

N7K-C7010



Rear

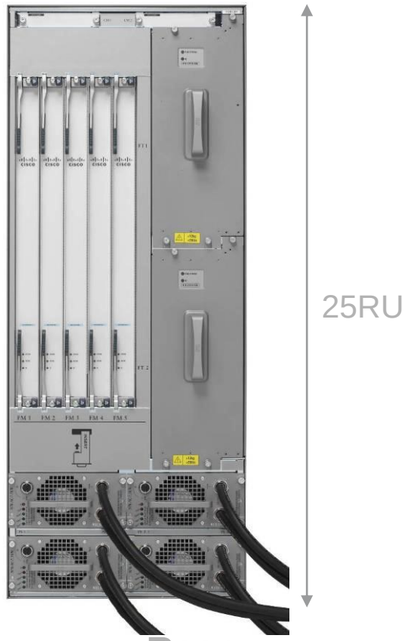
Nexus 7018



Front

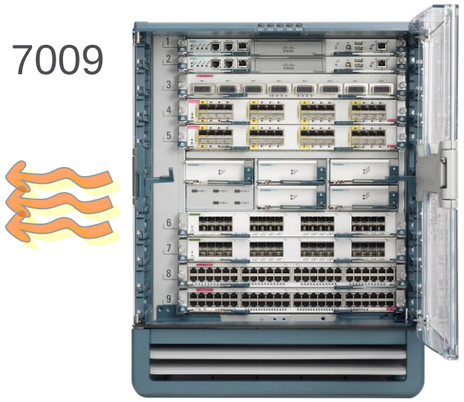
N7K-C7018

NX-OS 4.1(2) and later



Rear

Nexus 7009



Front

N7K-C7009

NX-OS 5.2(1) and later

Rear

Key Chassis Components

- Common components:
  - Supervisor Engines
  - I/O Modules
  - Power Supplies
- Chassis-specific components:
  - Fabric Modules
  - Fan Trays

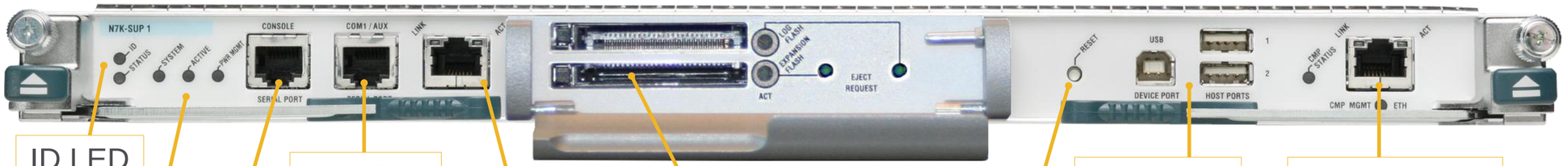
Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

Supervisor Engine 1

- Performs control plane and management functions
  - Dual-core 1.66GHz x86 processor with 8GB DRAM
  - 2MB NVRAM, 2GB internal bootdisk, compact flash slots, USB
- Console, aux, and out-of-band management interfaces
- Interfaces with I/O modules via 1G switched EOBC
- Houses dedicated central arbiter ASIC
  - Controls access to fabric bandwidth via dedicated arbitration path to I/O modules

N7K-SUP1



ID LED

Status LEDs

Console Port

AUX Port

Management Ethernet

Compact Flash Slots

Reset Button

USB Ports

CMP Ethernet

Nexus 7000 I/O Module Families

M Series and F Series

- M family – L2/L3/L4 with large forwarding tables and rich feature set

N7K-M132XP-12/
N7K-M132XP-12L

N7K-M108X2-12L

N7K-M148GT-11/N7K-M148GT-11L

N7K-M148GS-11/N7K-M148GS-11L

- F family – High performance, low latency, low power with streamlined feature set

N7K-F132XP-15

N7K-F248XP-25

Supported in NX-OS release 5.0(2a) and later

M Family

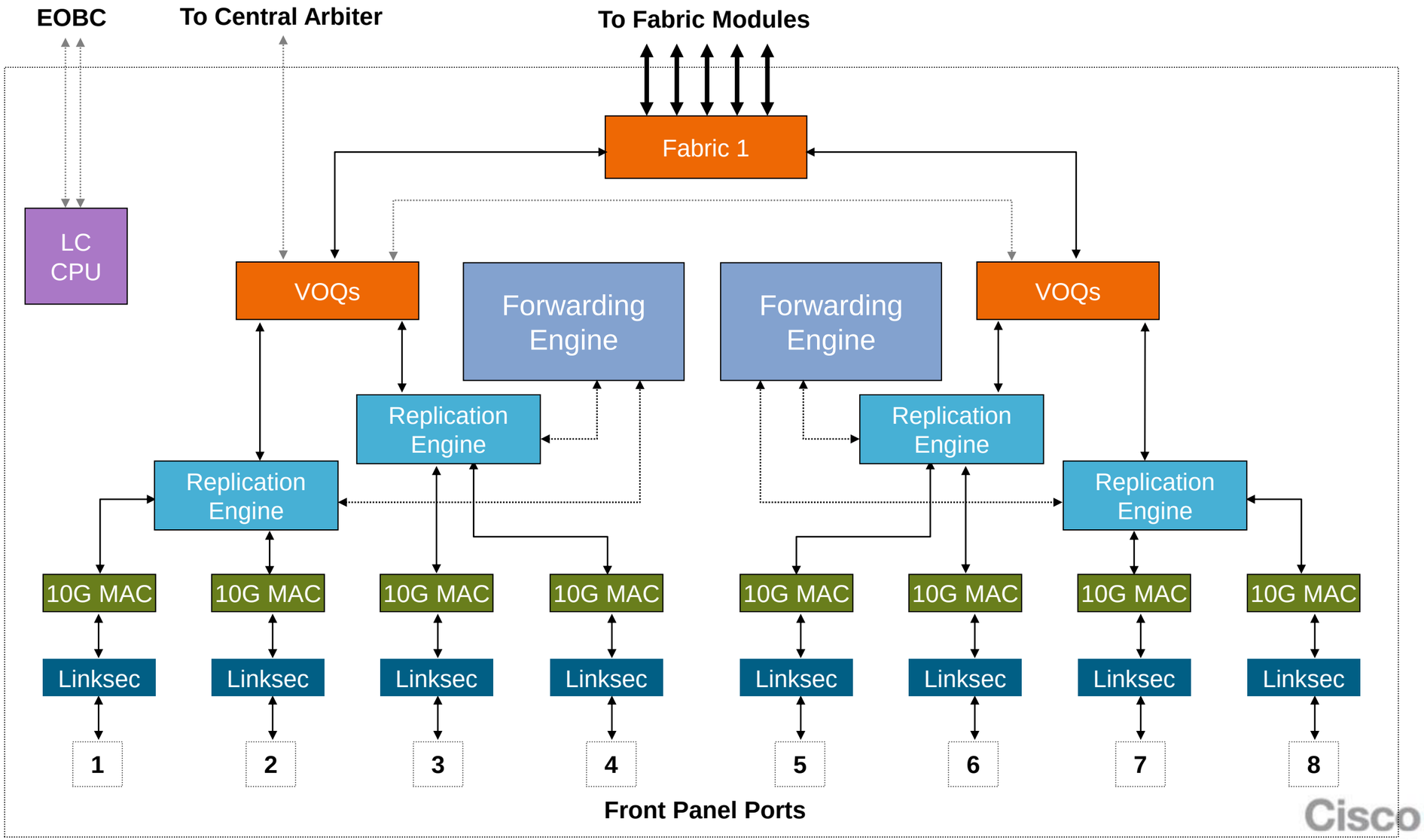
8-Port 10GE M1 I/O Module

N7K-M108X2-12L

- 8-port 10G with X2 transceivers
- 80G full-duplex fabric connectivity
- Two integrated forwarding engines (120Mpps)
  - Support for “XL” forwarding tables (licensed feature)
- Distributed L3 multicast replication
- 802.1AE LinkSec

8-Port 10G XL M1 I/O Module Architecture

N7K-M108X2-12L



N7K-M132XP-12 – Supported in all releases
N7K-M132XP-12L – Supported in NX-OS release 5.1(1) and later

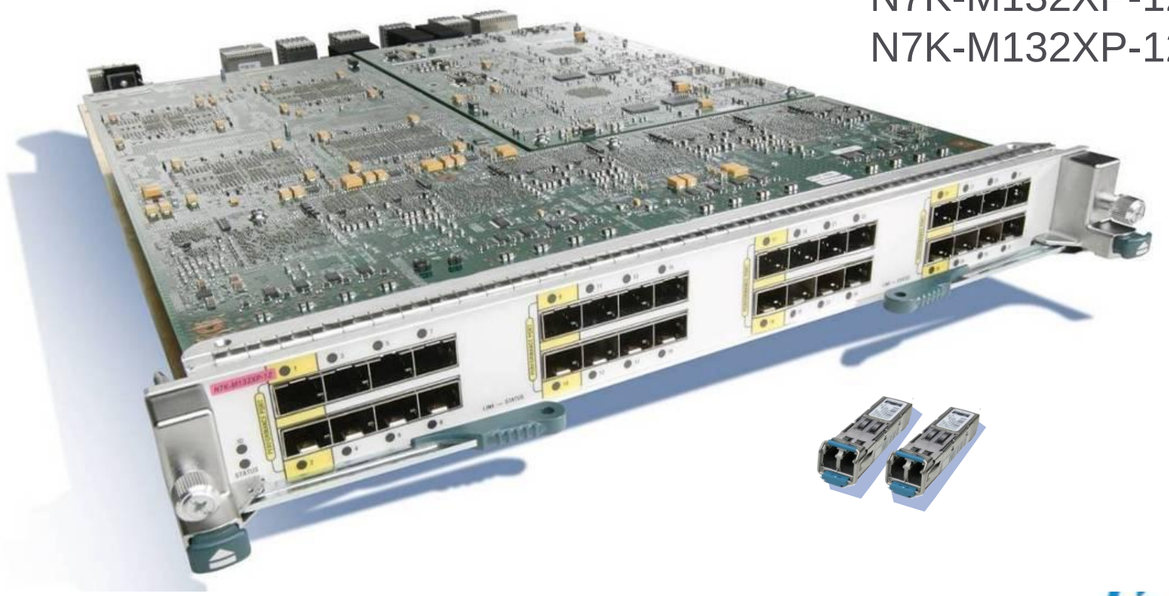
M Family

32-Port 10GE M1 I/O Modules

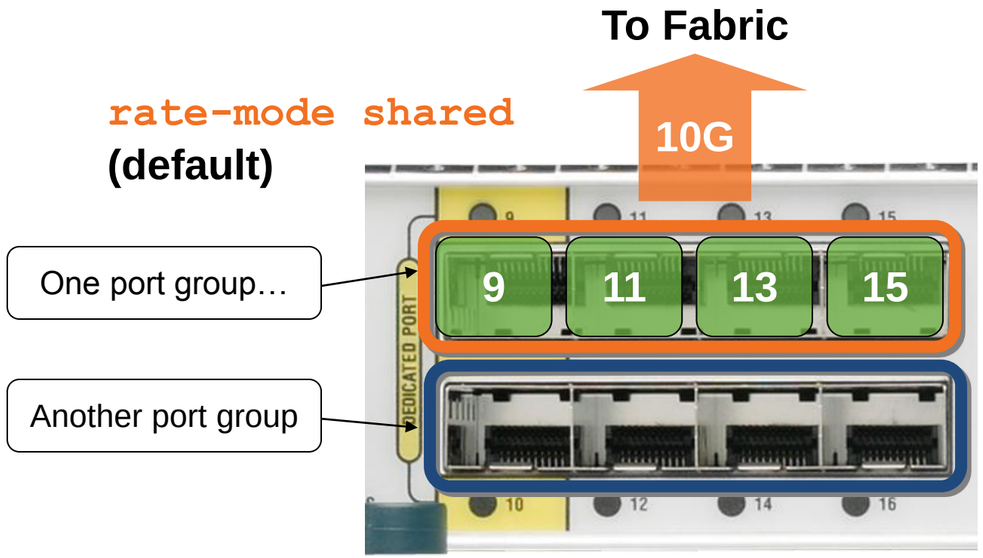
N7K-M132XP-12, N7K-M132XP-12L

- 32-port 10G with SFP+ transceivers
- 80G full-duplex fabric connectivity
- Integrated 60Mpps forwarding engine
  - XL forwarding engine on “L” version
- Oversubscription option for higher density (up to 4:1)
- Supports Nexus 2000 (FEX) connections
- Distributed L3 multicast replication
- LISP support
- 802.1AE LinkSec

N7K-M132XP-12/
N7K-M132XP-12L



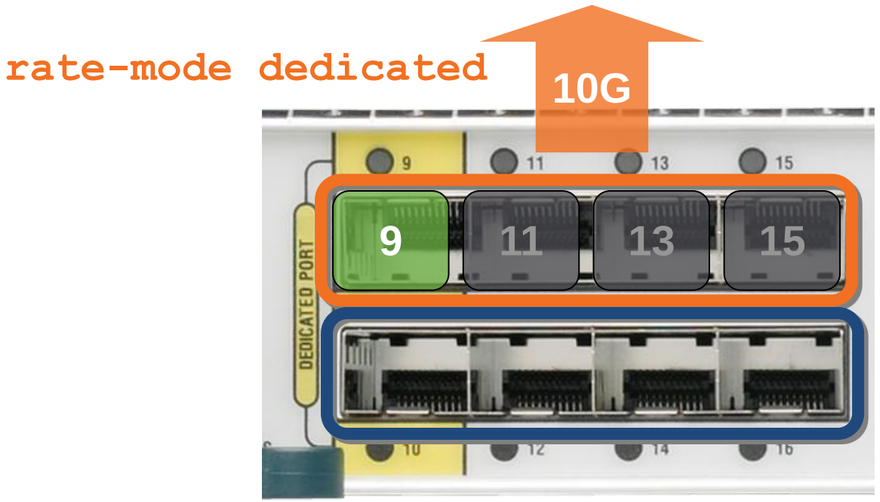
Shared vs. Dedicated Mode



Shared mode

- Four interfaces in port group share 10G bandwidth

“Port group”– group of contiguous even or odd ports that share 10G of bandwidth (e.g., ports 1,3,5,7)

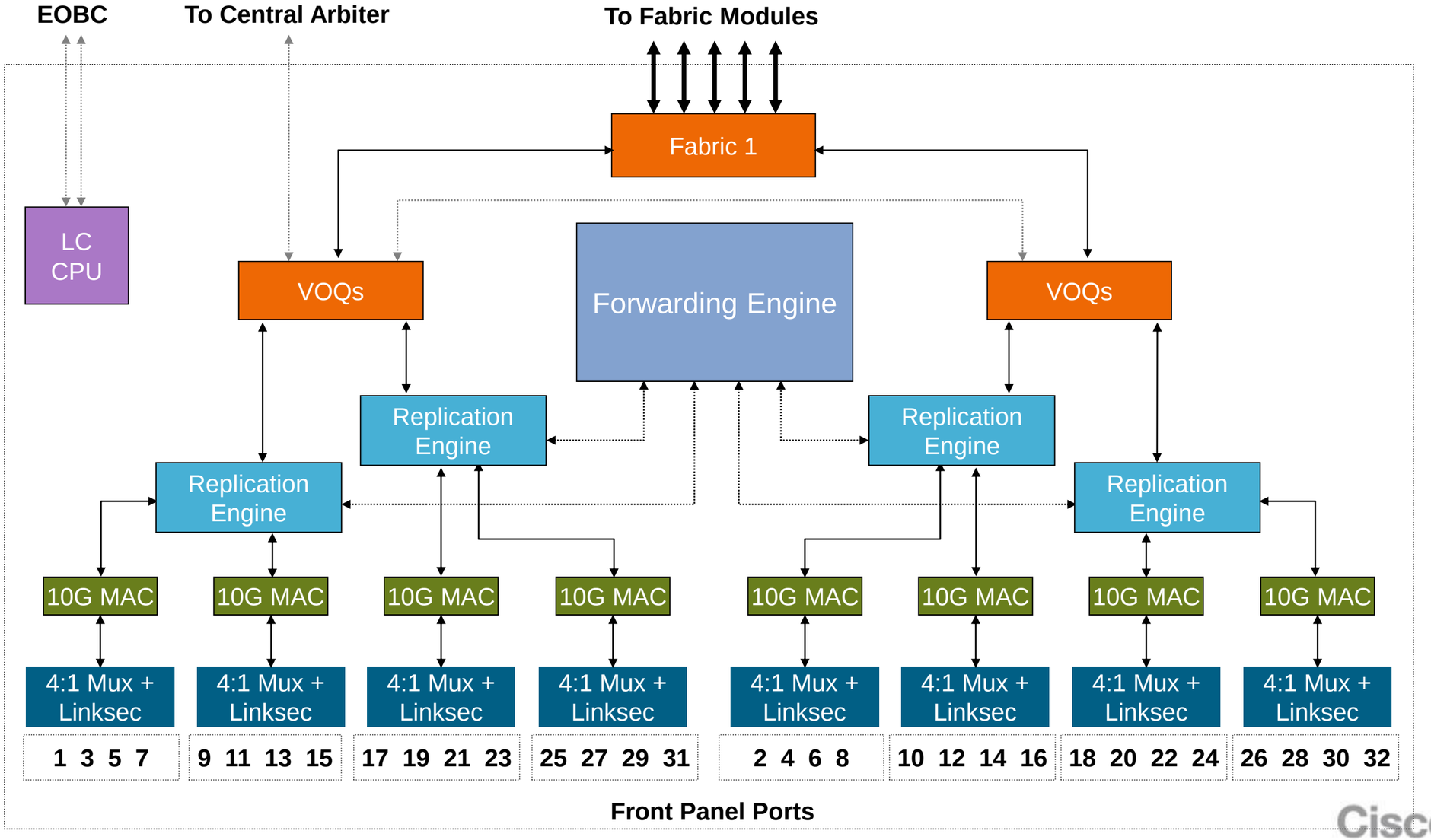


Dedicated mode

- First interface in port group gets 10G bandwidth
- Other three interfaces in port group disabled

32-Port 10G M1 I/O Module Architecture

N7K-M132XP-12, N7K-M132XP-12L



32-Port 1G/10GE F1 I/O Module

N7K-F132XP-15

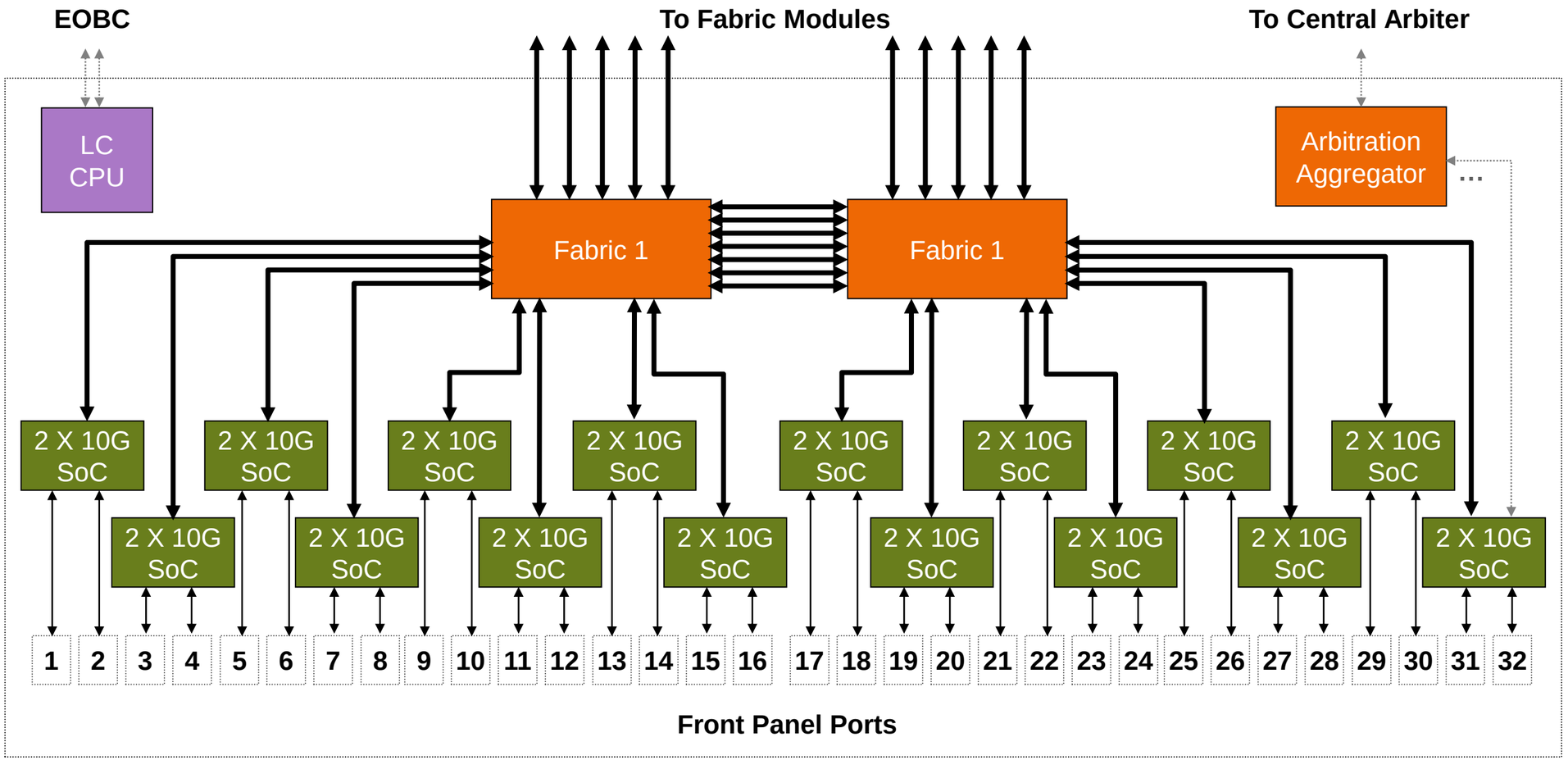
- 32-port 1G/10G with SFP/SFP+ transceivers
- 230G full-duplex fabric connectivity (320G local switching)
- System-on-chip (SoC)* forwarding engine design
  - 16 independent SoC ASICs
- Layer 2 forwarding with L3/L4 services (ACL/QoS)
- FabricPath-capable
- FCoE-capable

* sometimes called “switch-on-chip”

F Family

32-Port 1G/10G F1 I/O Module Architecture

N7K-F132XP-15



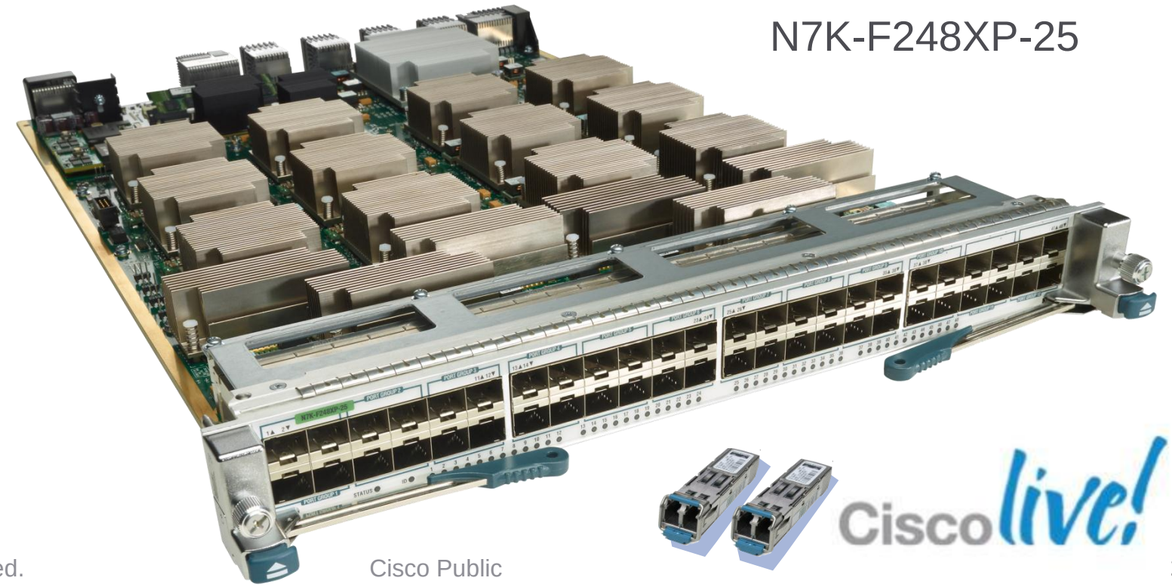
F Family

48-Port 1G/10GE F2 I/O Module

N7K-F248XP-25

- 48-port 1G/10G with SFP/SFP+ transceivers
- 480G full-duplex fabric connectivity
- System-on-chip (SoC)* forwarding engine design
  - 12 independent SoC ASICs
- Layer 2/Layer 3 forwarding with L3/L4 services (ACL/QoS)
- Supports Nexus 2000 (FEX) connections
- FabricPath-capable
- FCoE-ready

N7K-F248XP-25

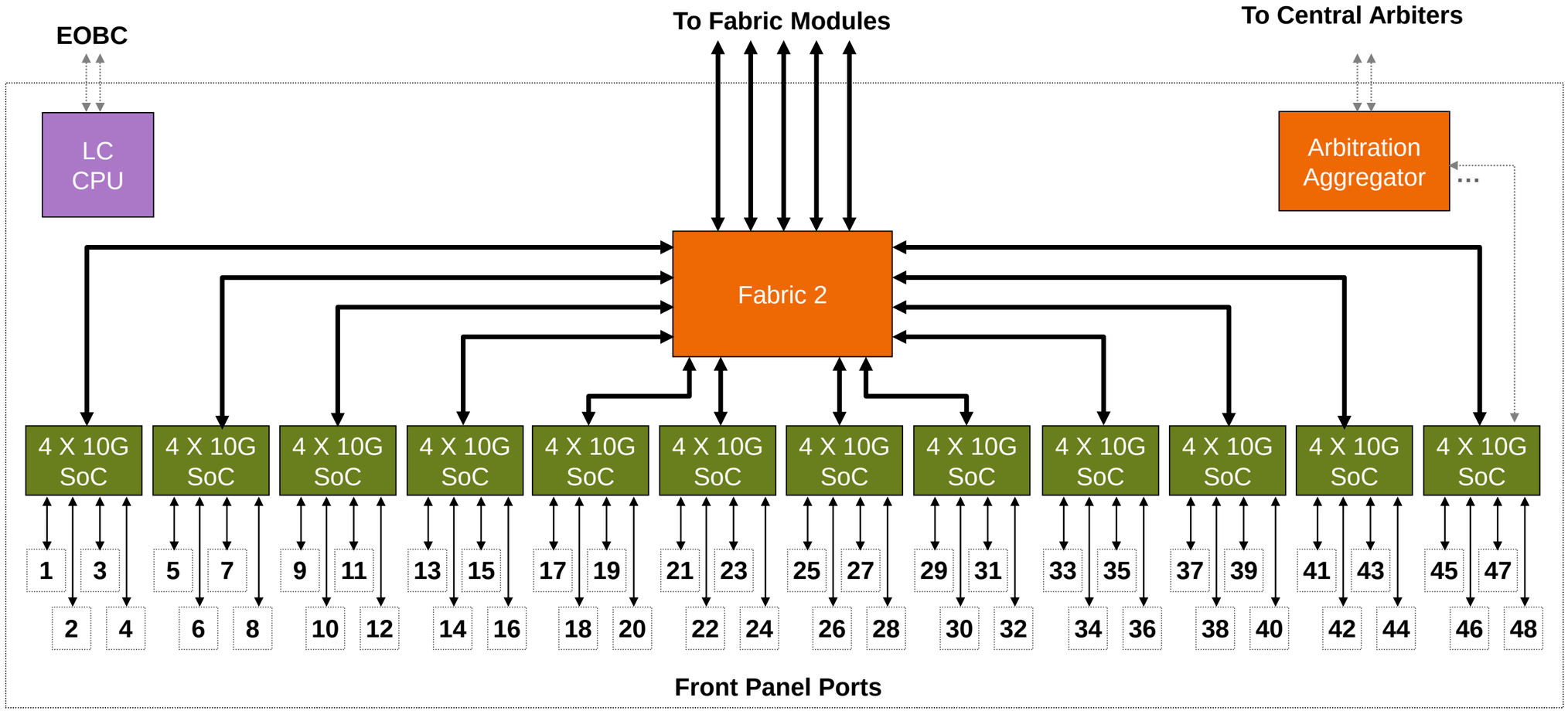


* sometimes called “switch-on-chip”

48-Port 1G/10G F2 I/O Module Architecture

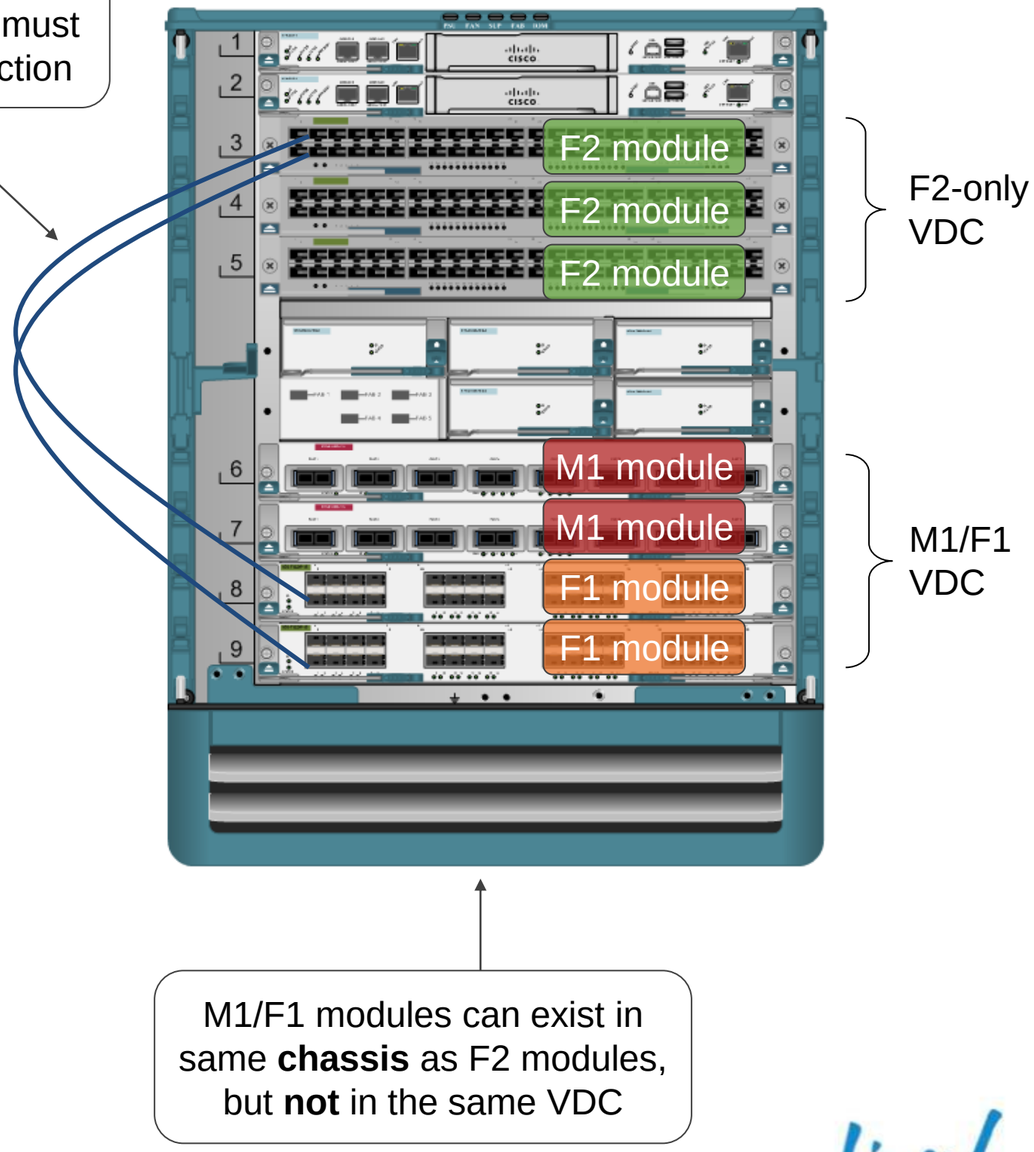
N7K-F248XP-25

F Family



F2-Only VDC

- F2 modules do **not** interoperate with other Nexus 7000 modules
- Must deploy in an “F2 only” VDC
- Can be default VDC, or any other VDC
  - Use the **limit-resource module-type f2** VDC configuration command
- System with only F2 modules and empty configuration boots with F2-only default VDC automatically



Agenda

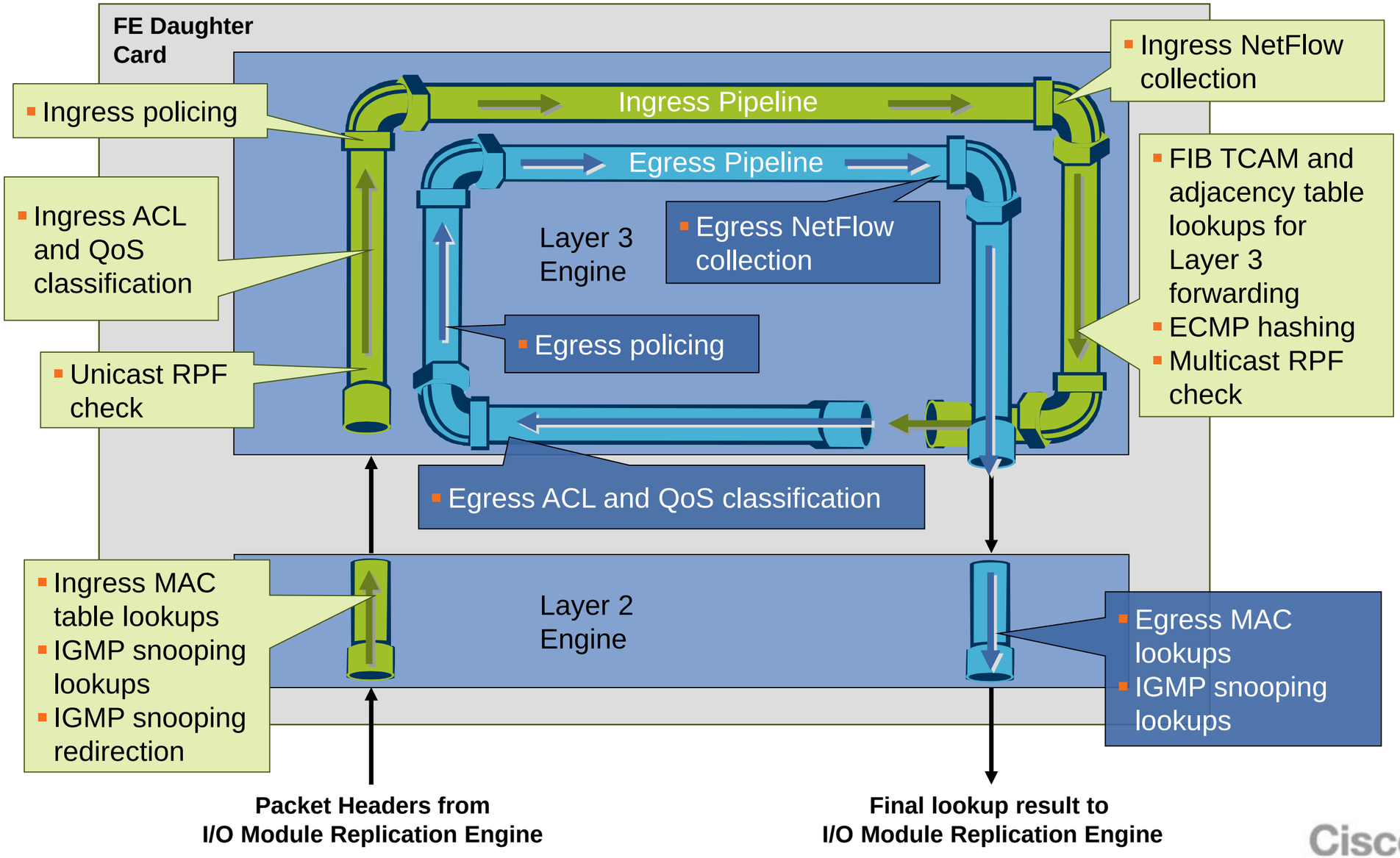
- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

M1 Forwarding Engine Hardware

- Hardware forwarding engine(s) integrated on every I/O module
- 60Mpps per forwarding engine Layer 2 bridging with hardware MAC learning
- 60Mpps per forwarding engine Layer 3 IPv4 and 30Mpps Layer 3 IPv6 unicast
- Layer 3 IPv4 and IPv6 multicast support (SM, SSM, bidir)
- MPLS
- OTV
- IGMP snooping
- RACL/VACL/PACL
- QoS remarking and policing policies
- Policy-based routing (PBR)
- Unicast RPF check and IP source guard
- Ingress and egress NetFlow (full and sampled)

Hardware Table	M1 Modules	M1-XL Modules without License	M1-XL Modules with License
FIB TCAM	128K	128K	**900K**
Classification TCAM (ACL/QoS)	64K	64K	**128K**
MAC Address Table	128K	128K	128K
NetFlow Table	512K	512K	512K

M1 Forwarding Engine Architecture



F2 Forwarding Engine Hardware

F Family

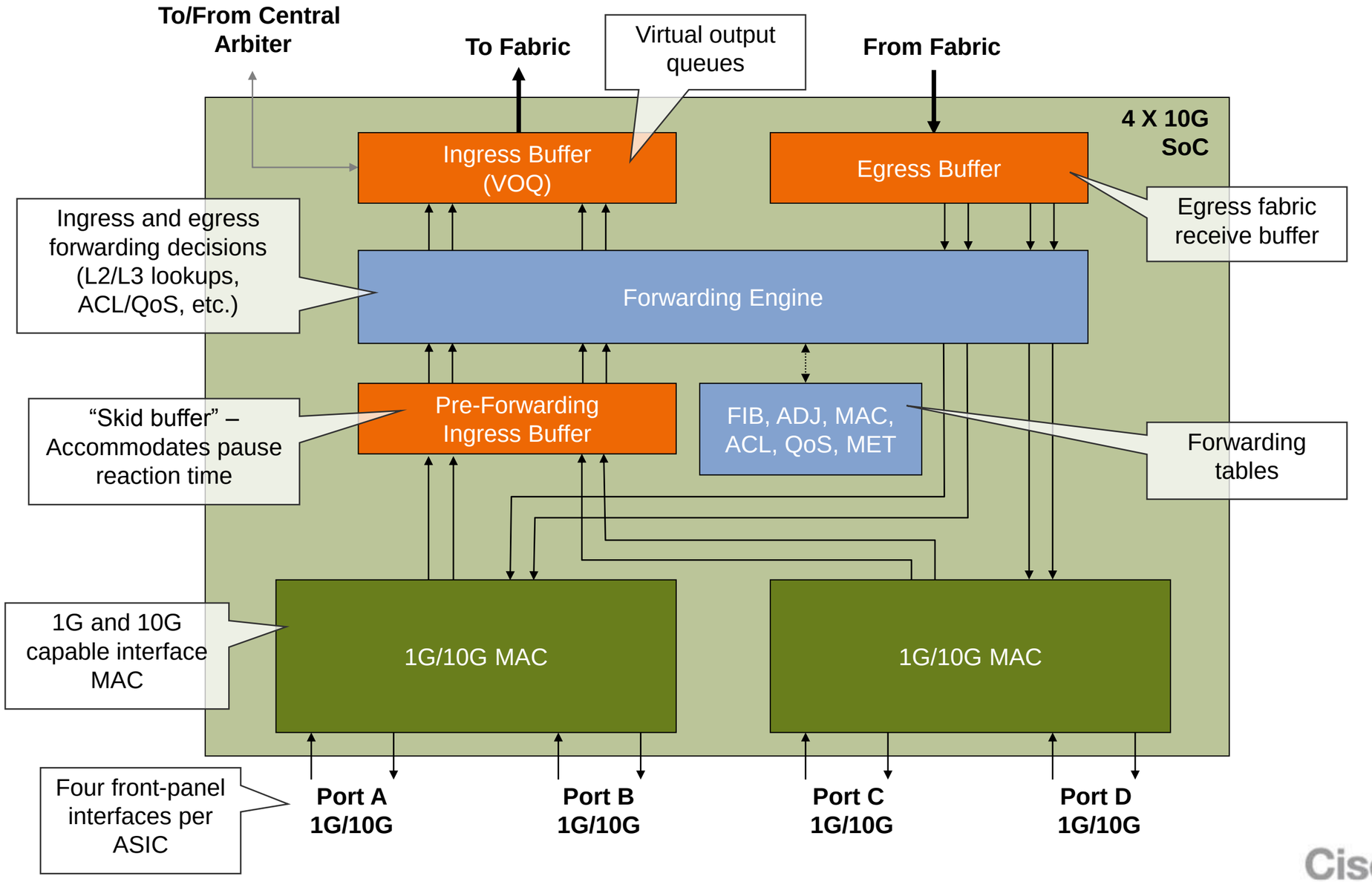
- Each SoC forwarding engine services 4 front-panel 10G ports (12 SoCs per module)
- 60Mpps per SoC Layer 2 bridging with hardware MAC learning
- 60Mpps per forwarding engine Layer 3 IPv4/ IPv6 unicast
- Layer 3 IPv4 and IPv6 multicast support (SM, SSM)
- IGMP snooping
- RACL/VACL/PACL
- QoS remarking and policing policies
- Policy-based routing (PBR)
- Unicast RPF check and IP source guard
- FabricPath forwarding
- Ingress sampled NetFlow (future)
- FCoE (future)

Hardware Table	Per F2 SoC	Per F2 Module
MAC Address Table	16K	256K*
FIB TCAM	32K IPv4/16K IPv6	32K IPv4/16K IPv6
Classification TCAM (ACL/QoS)	16K	192K*

* Assumes specific configuration to scale SoC resources

F Family

F2 Forwarding Engine Architecture



F1 Forwarding Engine Hardware

F Family

- Each SoC forwarding engine services 2 front-panel 10G ports (16 SoCs per module)
- 30Mpps per SoC Layer 2 bridging with hardware MAC learning
- IGMP snooping
- VACL/PACL
- QoS remarking policies
- FabricPath forwarding
- FCoE

Hardware Table	Per F1 SoC	Per F1 Module
MAC Address Table	16K	256K*
Classification TCAM (ACL/QoS)	1K in/1K out	16K in/16K out*

* Assumes specific configuration to scale SoC resources

Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

Crossbar Switch Fabric Modules

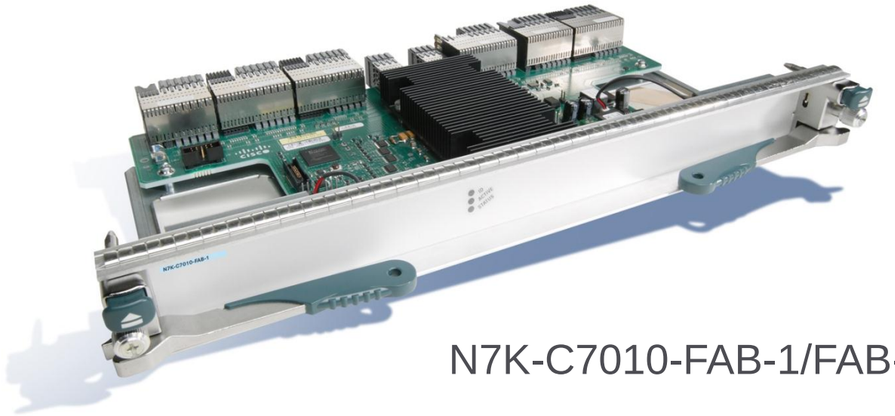
- Two fabric generations available – Fabric 1 and Fabric 2

Fabric	Per-fabric module bandwidth	Total bandwidth with 5 fabric modules
Fabric 1	46Gbps per slot	230Gbps per slot
Fabric 2	110Gbps per slot	550Gbps per slot

- Each installed fabric increases available per-payload slot bandwidth
- Different I/O modules leverage different amount of fabric bandwidth
- All I/O modules compatible with both Fabric 1 and Fabric 2
- Access to fabric bandwidth controlled using QoS-aware central arbitration with VOQ

N7K-C7018-FAB-1/FAB-2

N7K-C7009-FAB-2

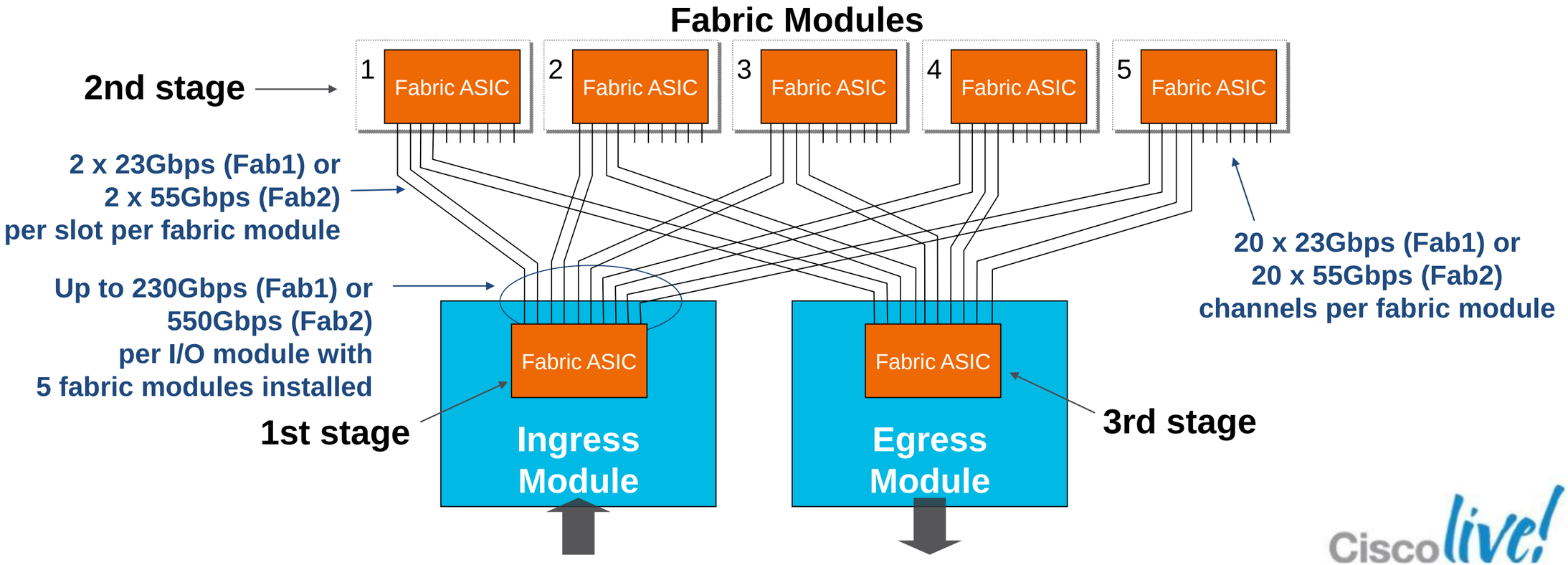


N7K-C7010-FAB-1/FAB-2

Multistage Crossbar

Nexus 7000 implements 3-stage crossbar switch fabric

- Stages 1 and 3 on I/O modules
- Stage 2 on fabric modules



I/O Module Capacity – Fabric 1

230Gbps

per slot bandwidth

<u>One fabric</u>

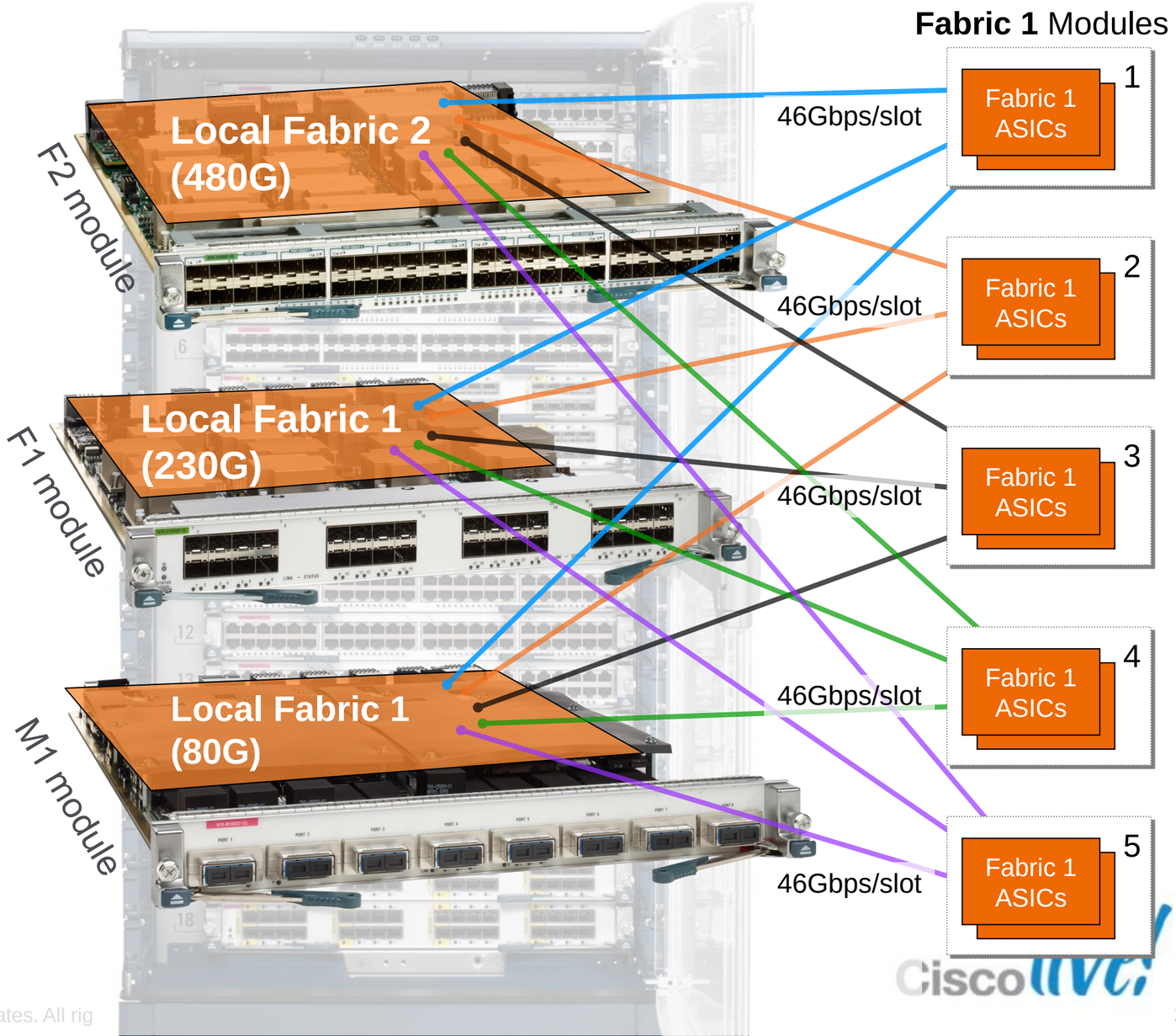
- **Any port** can pass traffic to **any other port** in system

<u>Two fabrics</u>

- 80G M1 module has full bandwidth

<u>Five fabrics</u>

- 230G F1 module has maximum bandwidth
- 480G F2 module limited to 230G per slot



I/O Module Capacity – Fabric 2

Fab2 does **NOT** make Fab1-based modules faster!!

550Gbps

per slot bandwidth

One fabric

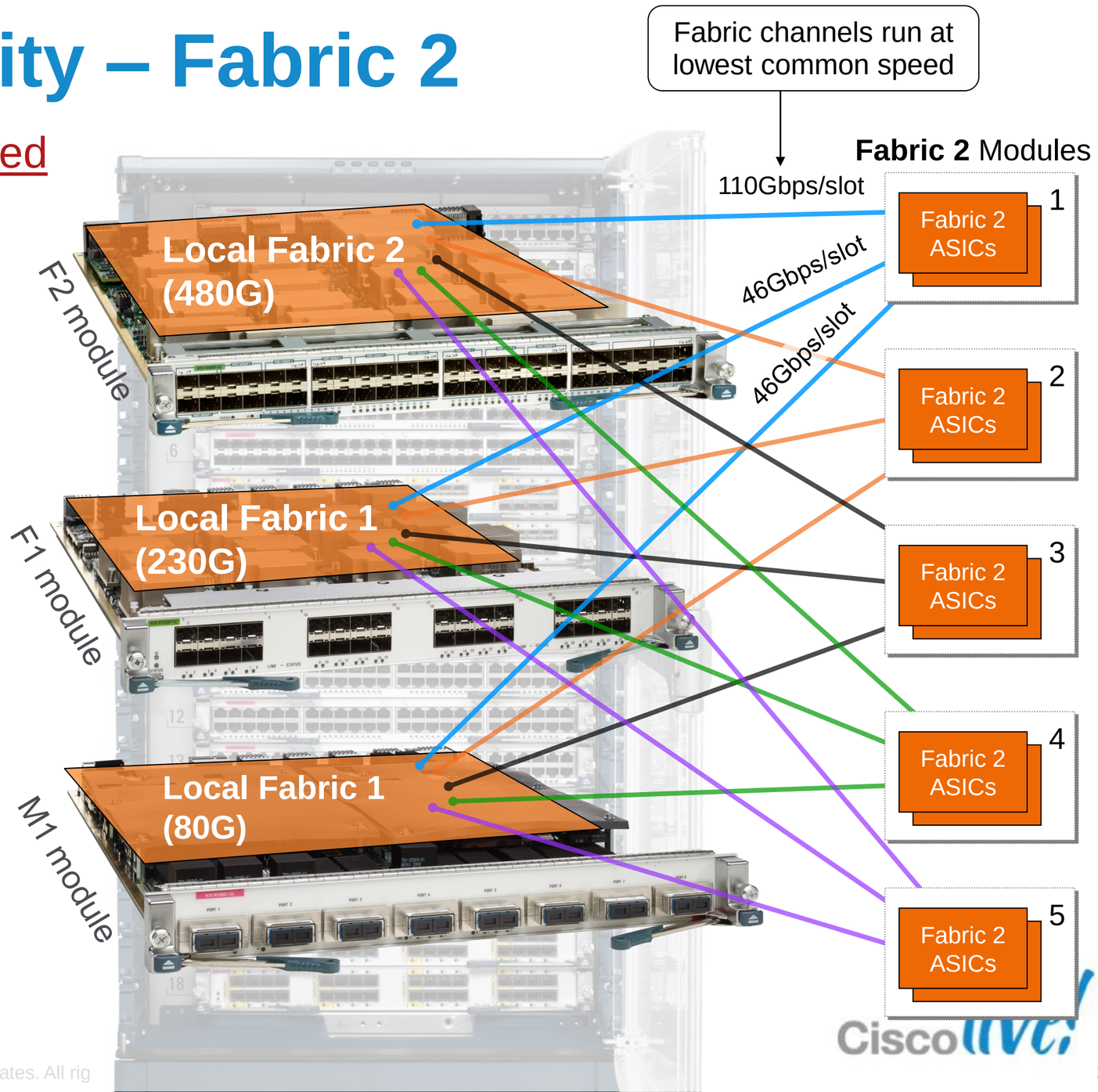
- **Any port** can pass traffic to **any other port** in system

Two fabrics

- 80G M1 module has full bandwidth

Five fabrics

- 230G F1 module has maximum bandwidth
- 480G F2 module has maximum bandwidth



Fabric 1 to Fabric 2 Migration

- Online, non-disruptive migration of Fabric 1 to Fabric 2 supported
- Upgrade to software release supporting Fabric 2
- Remove one Fabric 1 module at a time, replace with Fabric 2 module
  - Allow new Fabric 2 module to come completely online before removing next Fabric 1 module
- Mix of Fabric 1/Fabric 2 not recommended or supported for longer than duration of the migration
  - Within 12 hours of install of first Fabric 2 module, system syslogs warning to complete migration

http://www.cisco.com/en/US/docs/switches/datacenter/hw/nexus7000/installation/guide/n7k_replacing.html

Arbitration, VOQ, and Crossbar Fabric

- Arbitration, VOQ, and fabric combine to provide all necessary infrastructure for packet transport inside switch

- **Central arbitration** – Controls scheduling of traffic into fabric based on fairness, priority, and bandwidth availability at egress ports
- **Virtual Output Queues (VOQs)** – Provide buffering and queuing for ingress-buffered switch architecture
- **Crossbar fabric** – Provides dedicated, high-bandwidth interconnects between ingress and egress I/O modules

Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

Buffering, Queuing, and Scheduling

- Buffering – storing packets in memory
  - Needed to absorb bursts, manage congestion
- Queuing – buffering packets according to traffic class
  - Provides dedicated buffer for packets of different priority
- Scheduling – controlling the order of transmission of buffered packets
  - Ensures preferential treatment for packets of higher priority and fair treatment for packets of equal priority

- Nexus 7000 uses queuing policies and network-QoS policies to define buffering, queuing, and scheduling behavior
- Default queuing and network-QoS policies always in effect in absence of any user configuration

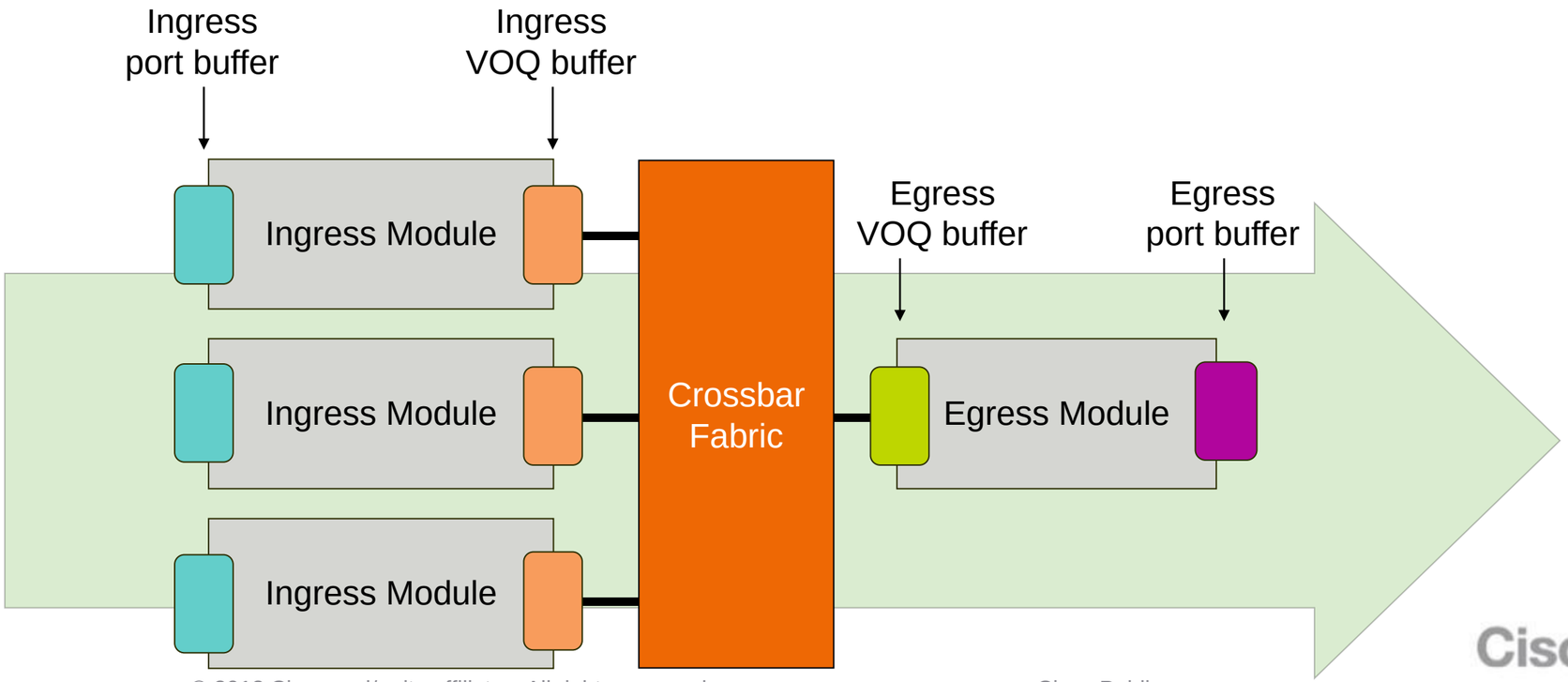
I/O Module Buffering Models

- Buffering model varies by I/O module family
  - **M1 modules**: hybrid model combining ingress VOQ-buffered architecture with egress port-buffered architecture
  - **F1/F2 modules**: pure ingress VOQ-buffered architecture
- All configuration through Modular QoS CLI (MQC)
  - Queuing parameters applied using class-maps/policy-maps/service-policies

Hybrid Ingress/Egress Buffered Model

M1 I/O Modules

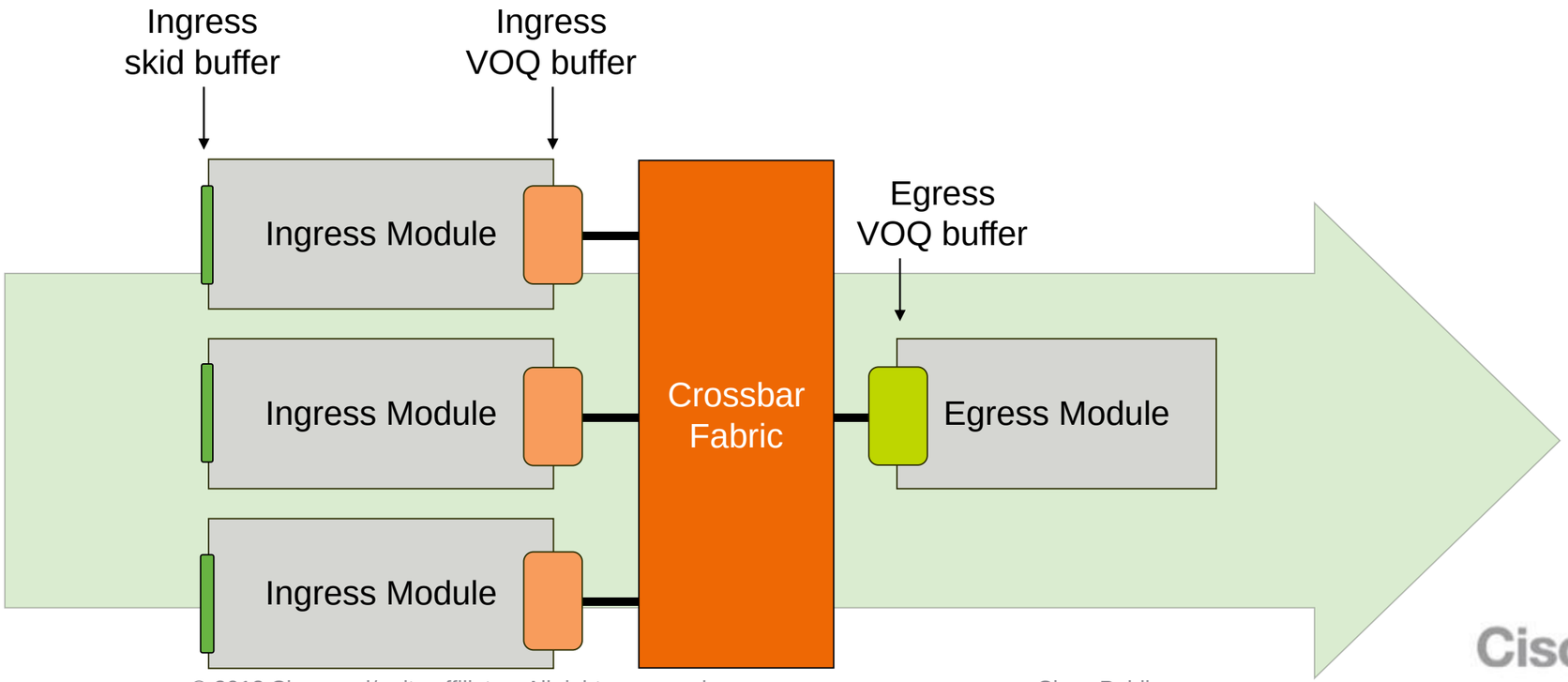
- Ingress port buffer – Manages congestion in ingress forwarding/replication engines only
- Ingress VOQ buffer – Manages congestion toward egress destinations over fabric
- Egress VOQ buffer – Receives frames from fabric; also buffers multidestination frames
- Egress port buffer – Manages congestion at egress interface



Ingress Buffered Model

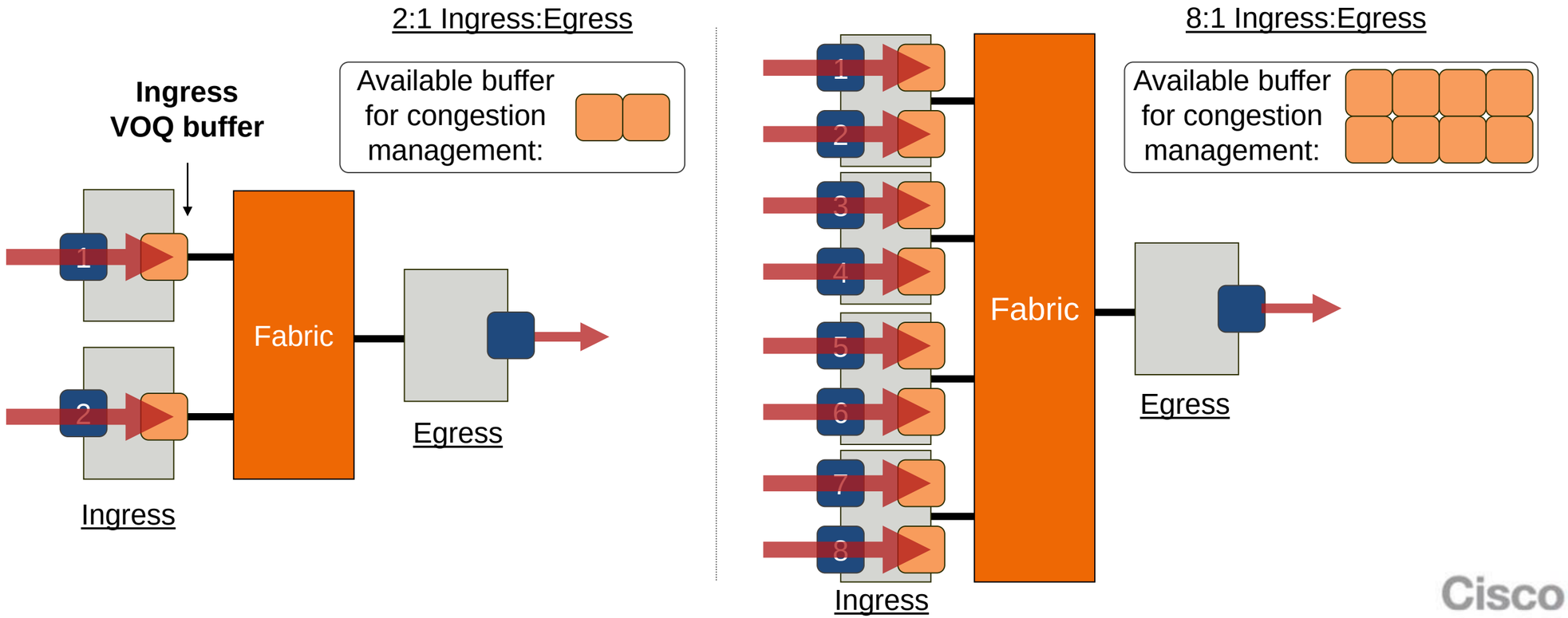
F1/F2 I/O Modules

- Ingress “skid” buffer – Absorbs packets in flight after external flow control asserted
- Ingress VOQ buffer – Manages congestion toward egress destinations over fabric
- Egress VOQ buffer – Receives frames from fabric; also buffers multidestination frames



Distributed Buffer Pool

- Ingress-buffered architecture implements large, distributed buffer pool to absorb congestion
- Absorbs congestion at every ingress port contributing to congestion, leveraging all per-port ingress buffer
- Excess traffic does not consume fabric bandwidth, only to be dropped at egress port



Agenda

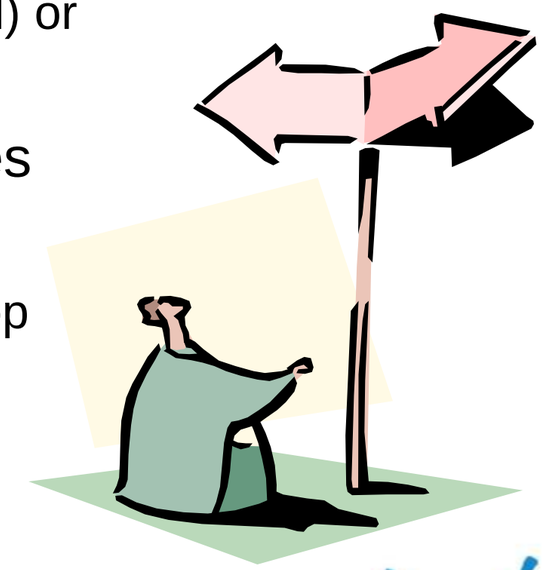
- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

Layer 2 Forwarding

- Layer 2 forwarding – traffic steering based on destination MAC address
- Hardware MAC learning
  - CPU not directly involved in learning
- Forwarding engine(s) on each module have copy of MAC table
  - New learns communicated to other forwarding engines via hardware “flood to fabric” mechanism
  - Software process ensures continuous MAC table sync
- Spanning tree (PVRST or MST), Virtual Port Channel (VPC), or FabricPath ensures loop-free Layer 2 topology

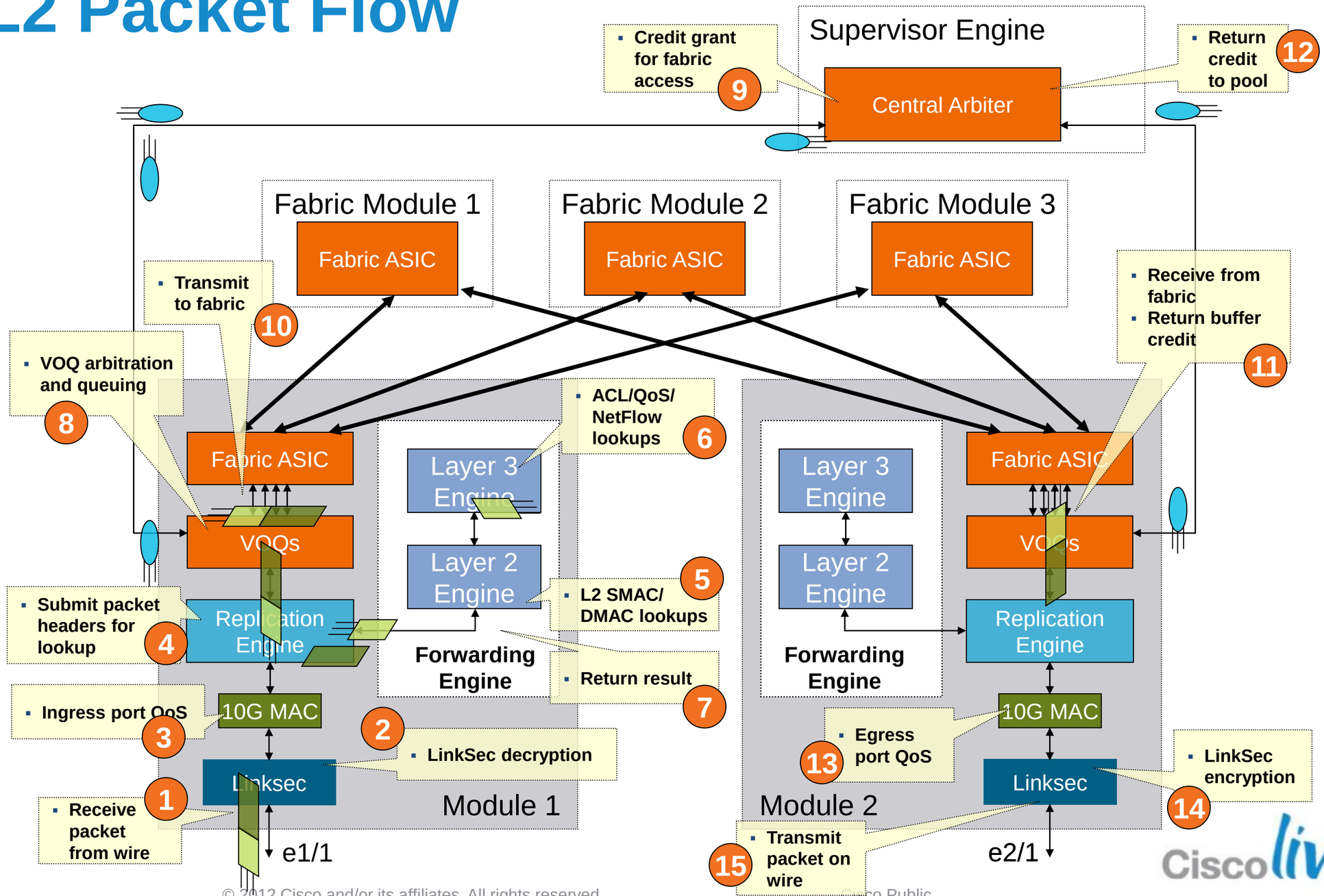
Hardware Layer 2 Forwarding Process

- In Classic Ethernet and FabricPath edge switches, MAC table lookup drives Layer 2 forwarding
  - –Source MAC and destination MAC lookups performed for each frame, based on {VLAN,MAC} pairs
  - –Source MAC lookup drives new learns and refreshes aging timers
  - –Destination MAC lookup dictates outgoing switchport (CE/FabricPath local) or destination Switch ID (FabricPath remote)
- In FabricPath core switches, Switch ID (routing) table lookup drives Layer 2 forwarding
  - –Destination SID lookup dictates outgoing FabricPath interface and next hop



HDR = Packet Headers DATA = Packet Data CTRL = Internal Signaling

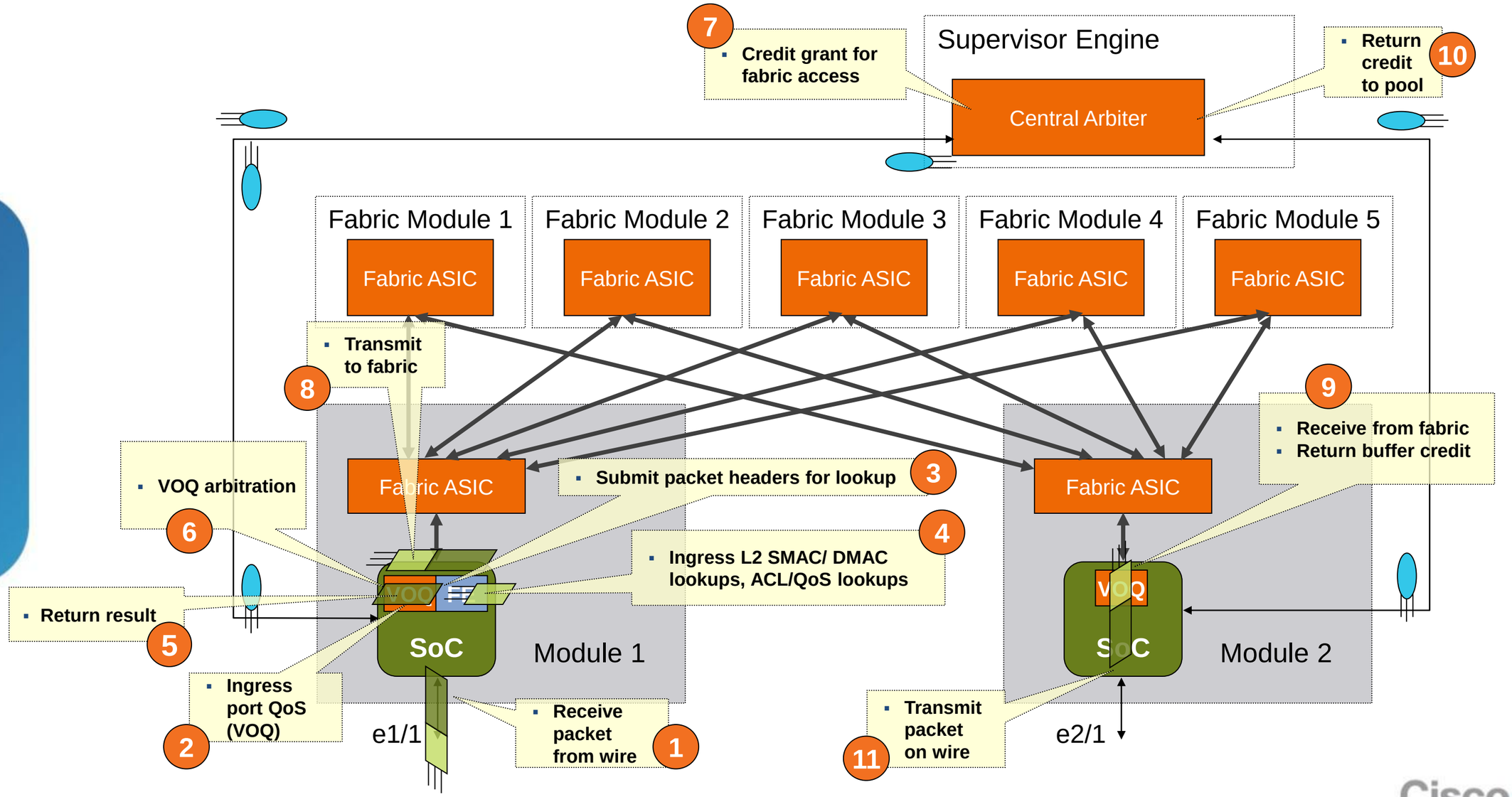
M1 L2 Packet Flow



HDR = Packet Headers DATA = Packet Data CTRL = Internal Signaling

F Family

F1/F2 L2 Packet Flow



Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

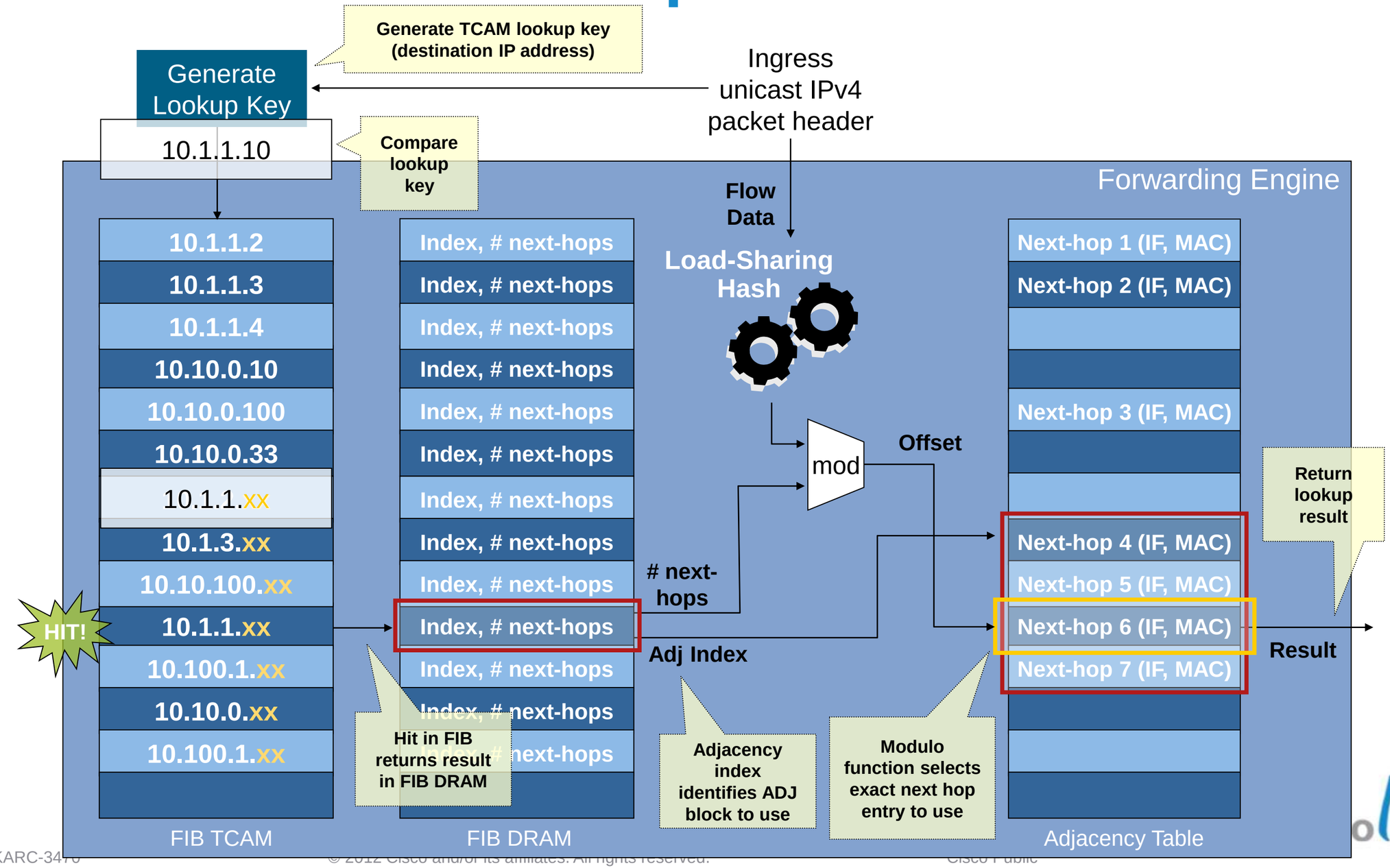
IP Forwarding

- Nexus 7000 decouples control plane and data plane
- Forwarding tables built on control plane using routing protocols or static configuration
  - –OSPF, EIGRP, IS-IS, RIP, BGP for dynamic routing
- Tables downloaded to forwarding engine hardware for data plane forwarding
  - –FIB TCAM contains IP prefixes
  - –Adjacency table contains next-hop information

Hardware IP Forwarding Process

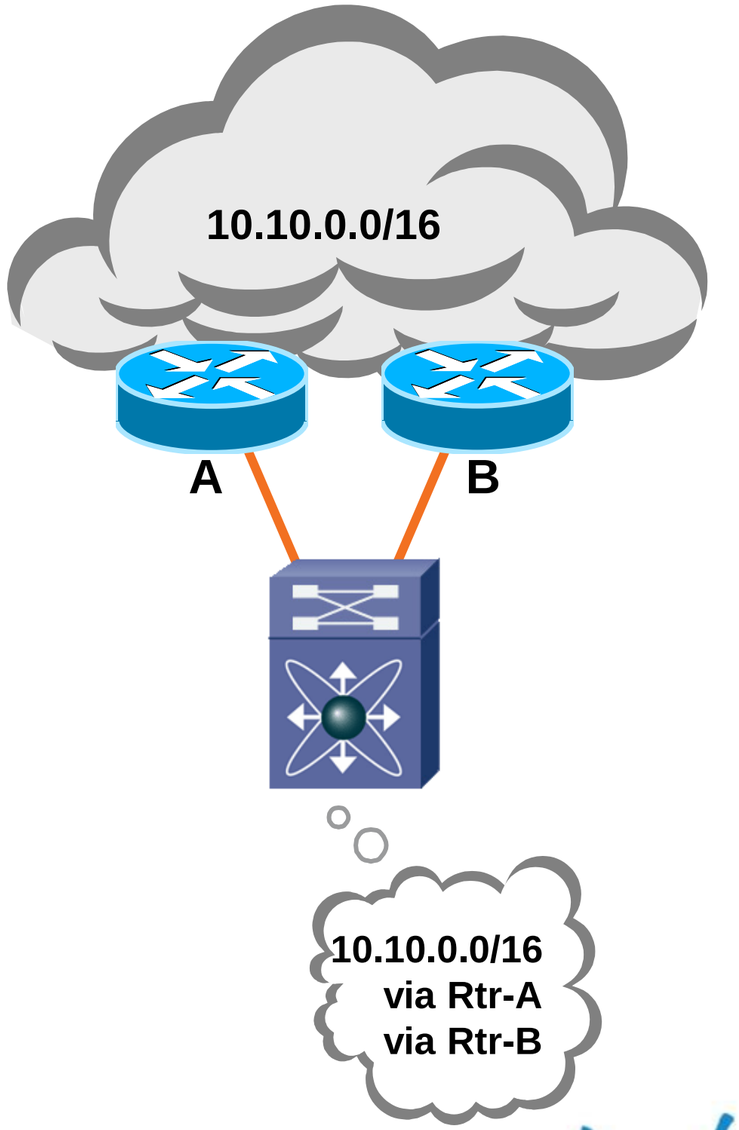
- FIB TCAM lookup based on destination prefix (longest-match)
- FIB “hit” returns adjacency, adjacency contains rewrite information (next-hop)
- Pipelined forwarding engine architecture also performs ACL, QoS, and NetFlow lookups, affecting final forwarding result

IPv4 FIB TCAM Lookup

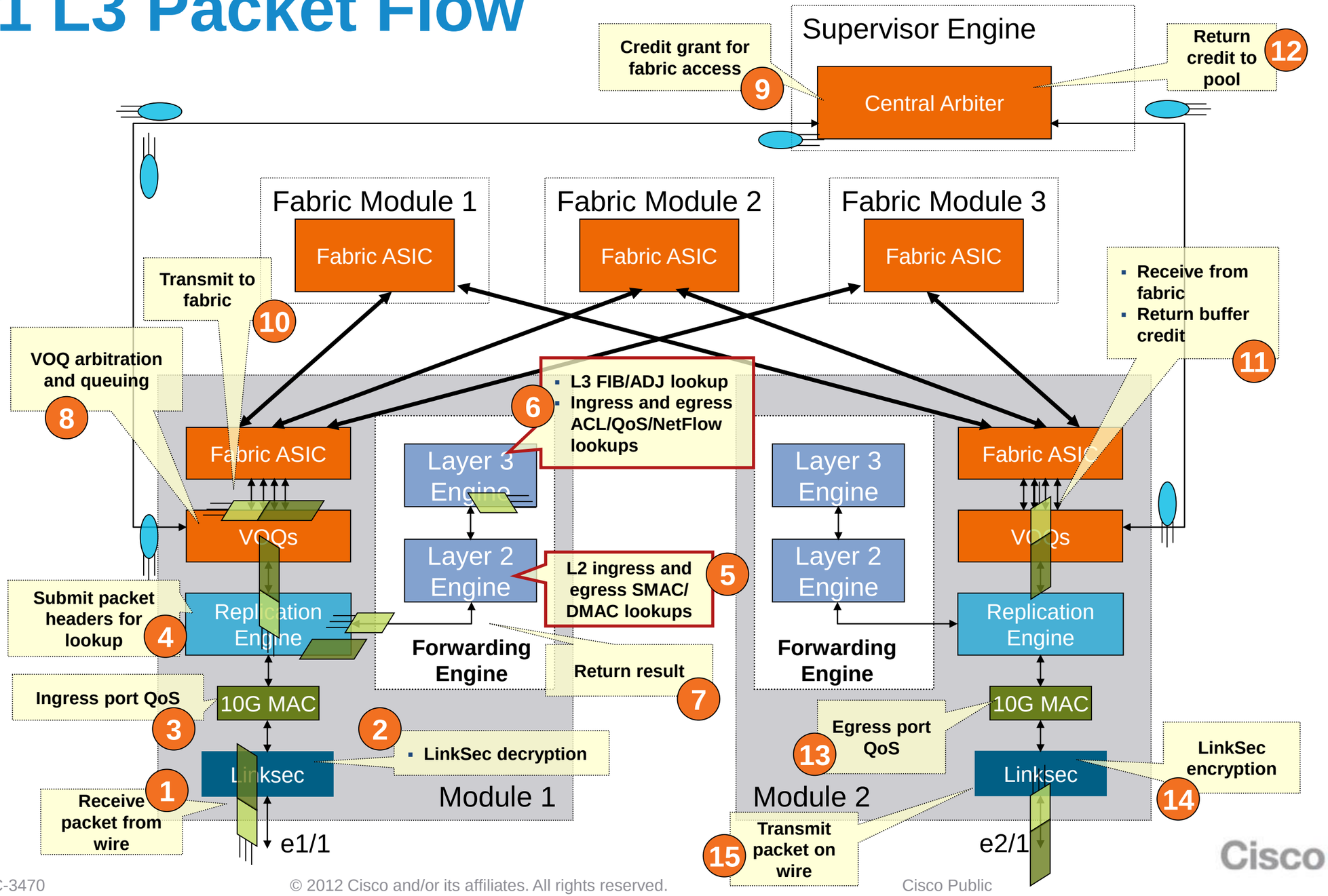


ECMP Load Sharing

- Up to 16 hardware load-sharing paths per prefix
- Use maximum-paths command in routing protocols to control number of load-sharing paths
- Load-sharing is per-IP flow
- Configure load-sharing hash options with global ip load-sharing command:
  - Source and Destination IP addresses
  - Source and Destination IP addresses plus L4 ports (default)
  - Destination IP address and L4 port
- Additional randomized number added to hash prevents polarization
  - Automatically generated or user configurable value



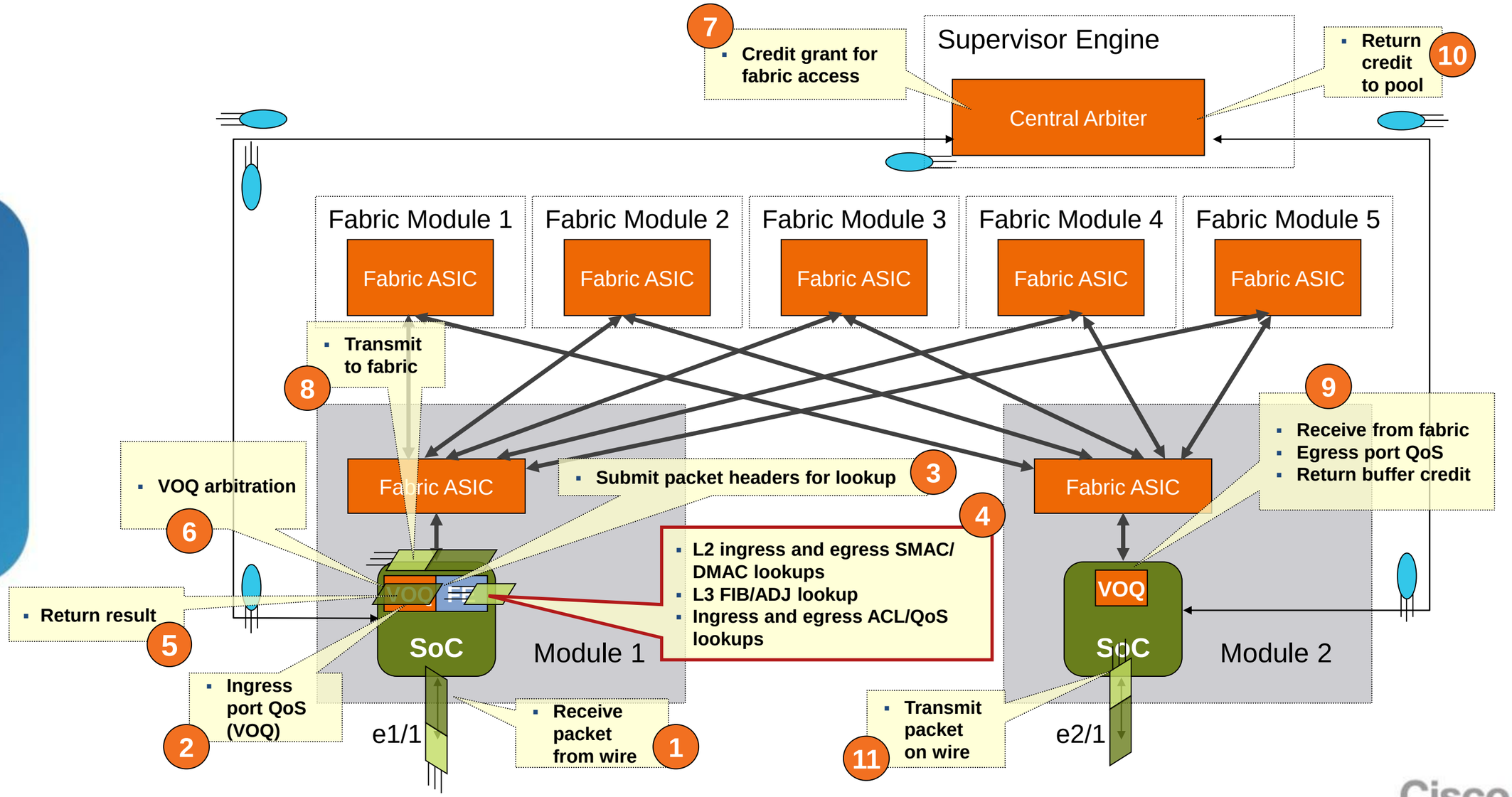
M1 L3 Packet Flow



HDR = Packet Headers DATA = Packet Data CTRL = Internal Signaling

F Family

F2 L3 Packet Flow

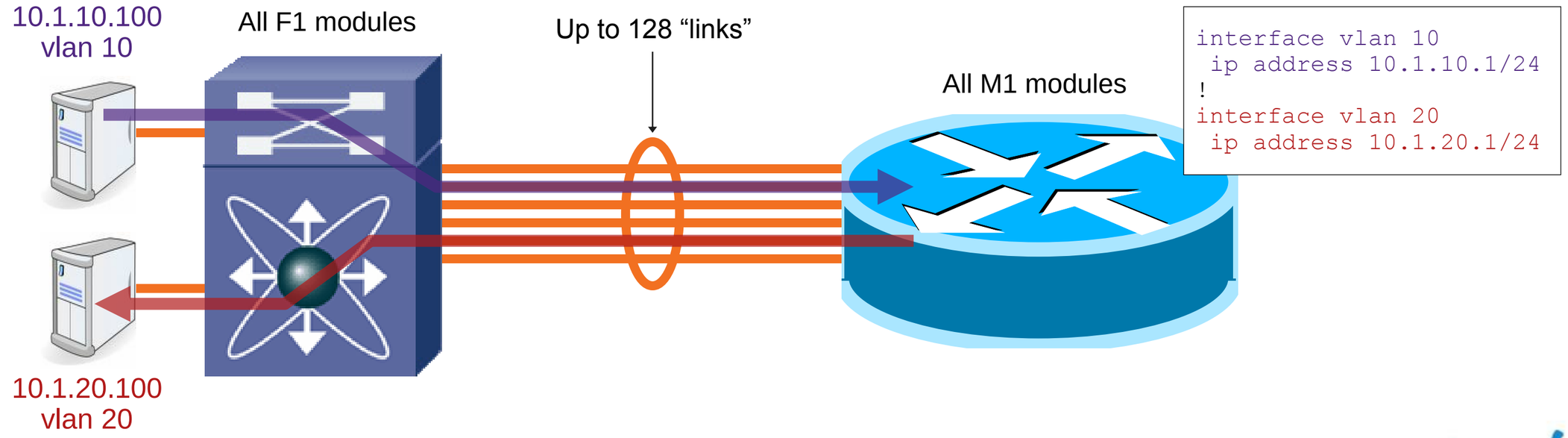


Layer 3 Forwarding with F1 I/O Modules

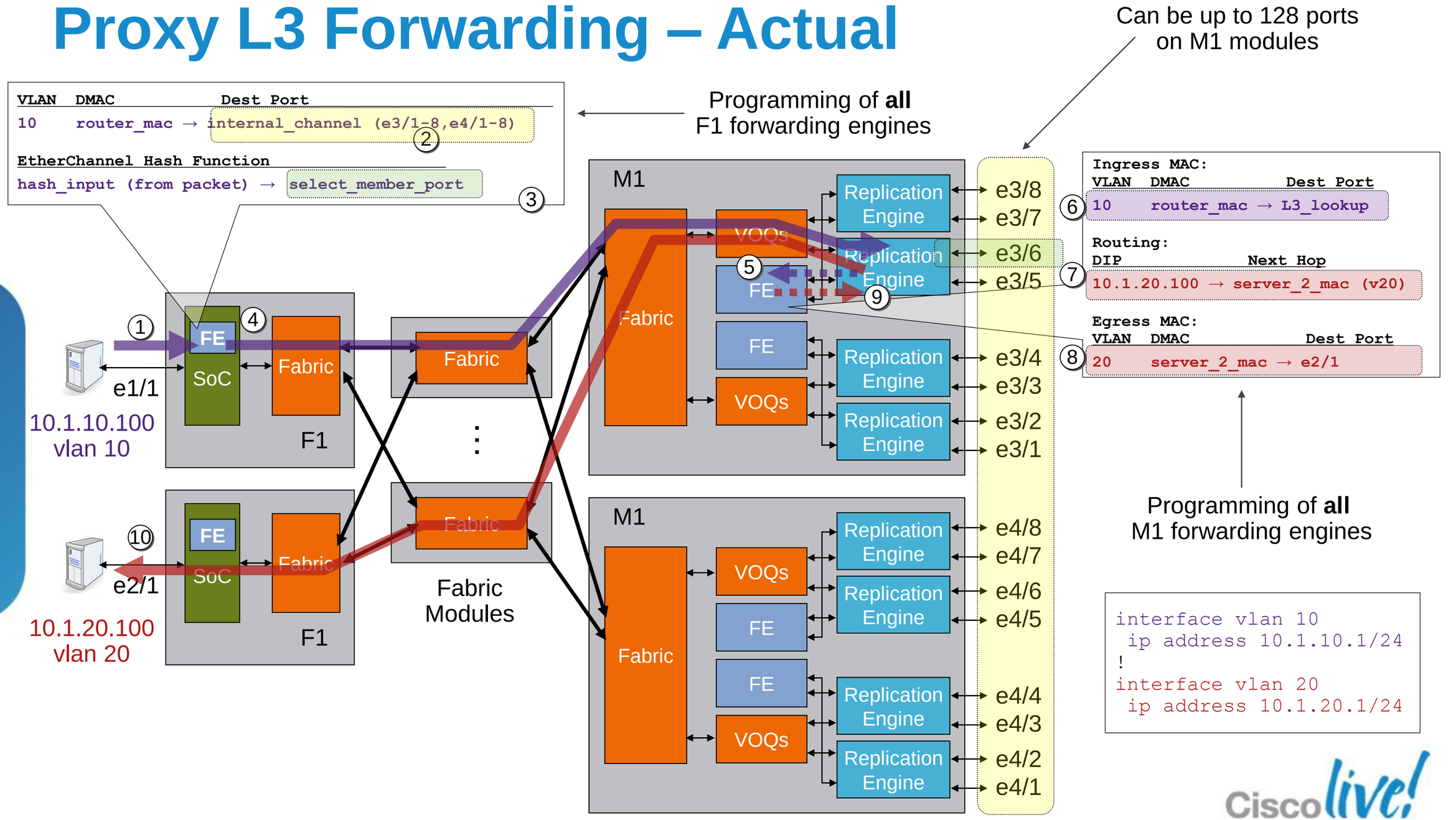
- F1 modules do not natively provide Layer 3 switching
  - Cannot inter-VLAN route on their own
- However, one or more M1/M1-XL modules can provide “proxy” Layer 3 services
  - M1 forwarding engines can proxy route for F1 modules
  - Proxy L3 forwarding enabled by default in M1/F1 VDC
- Packets destined to router MAC forwarded to M1 modules for Layer 3 via internal “Router Port-Channel”
  - Selection of which port on which M1 module based on EtherChannel hash function
  - Traffic requiring L3 from F1 modules traverses the fabric, “vectoring toward” M1 ports enabled for proxy L3
  - M1 module receiving such packets programmed to perform full ingress/egress L3 lookups

Proxy L3 Forwarding – Conceptual

- From F1 perspective, Router MAC reachable through giant port-channel
- All packets destined to Router MAC forwarded through fabric toward one "member port" in that channel



Proxy L3 Forwarding – Actual



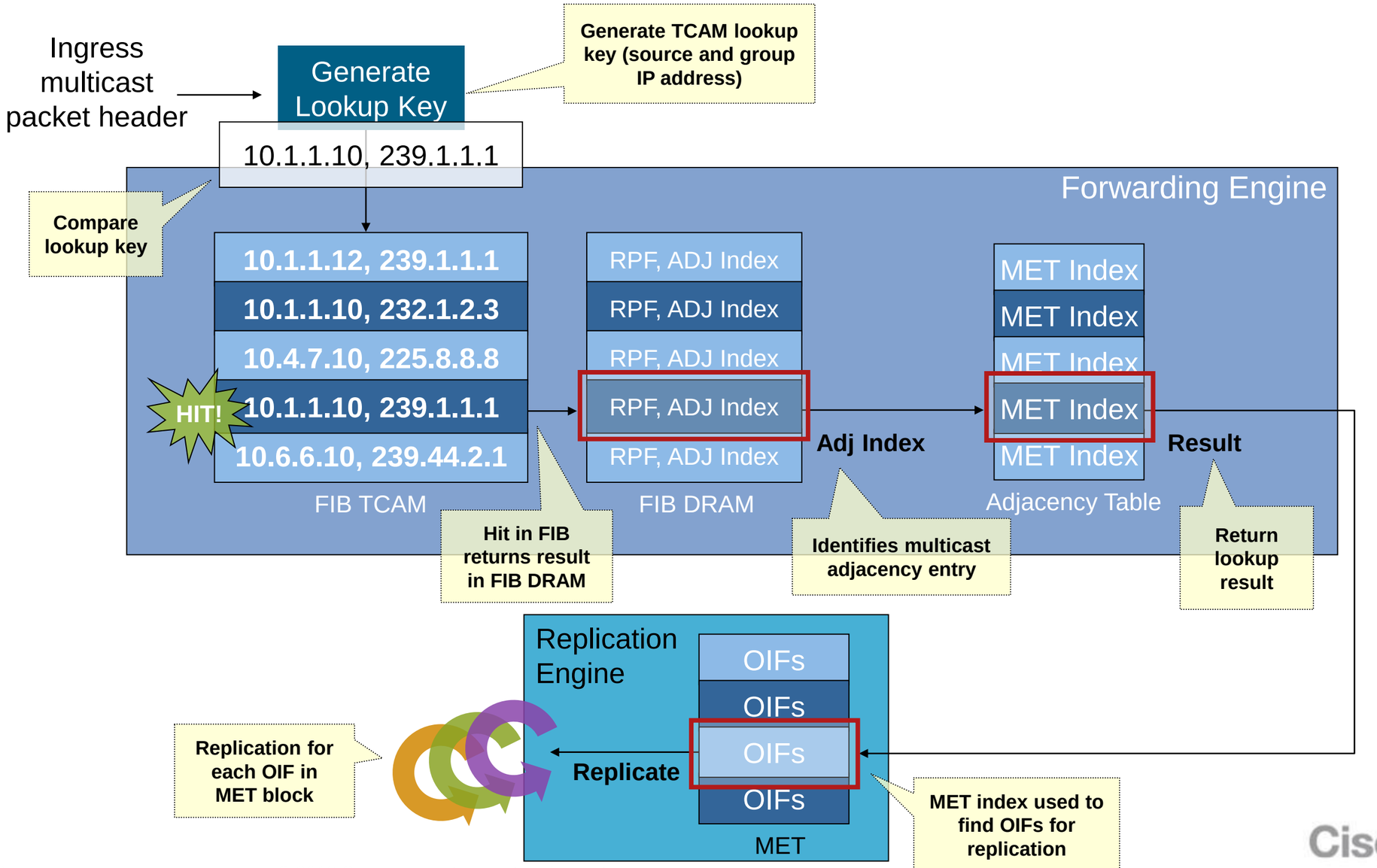
Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

IP Multicast Forwarding

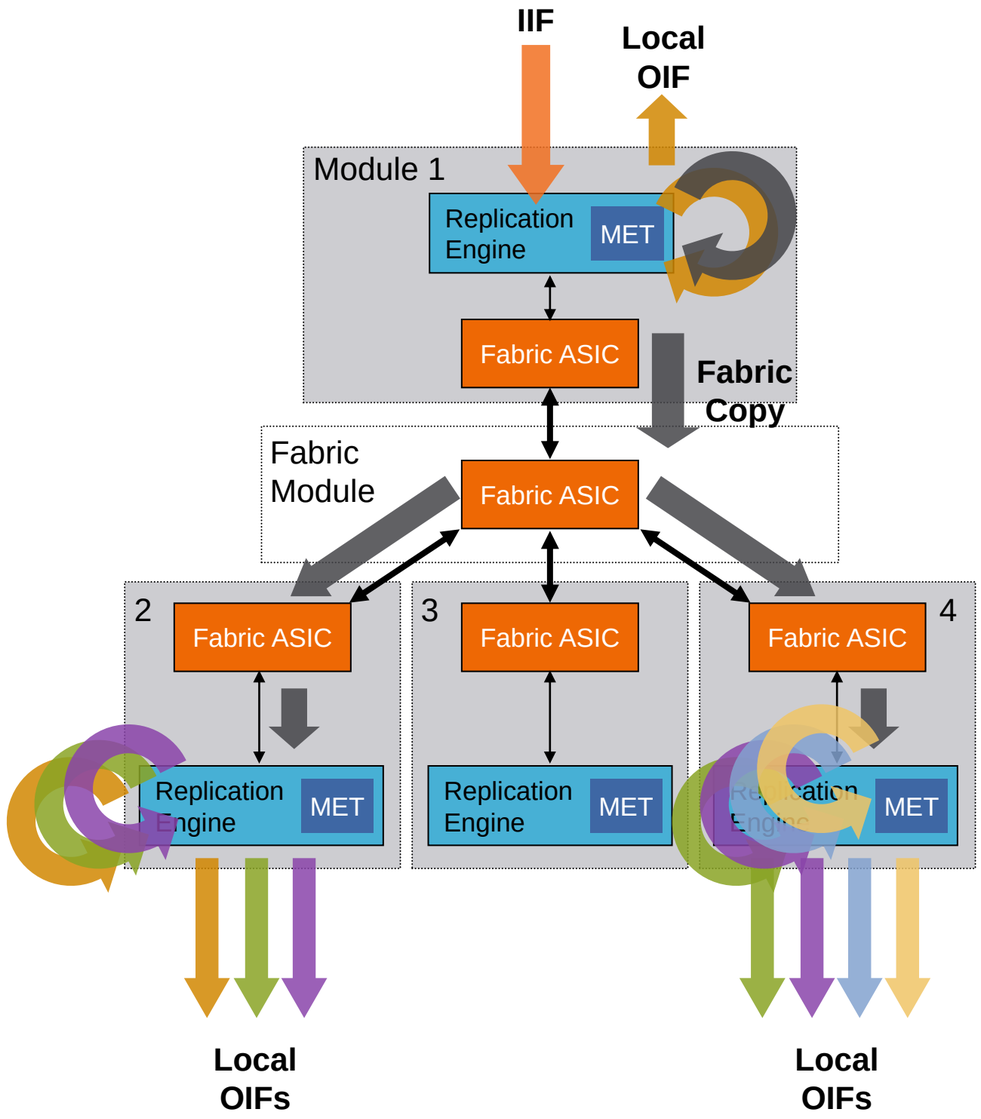
- Forwarding tables built on control plane using multicast protocols
  - PIM-SM, PIM-SSM, PIM-Bidir, IGMP, MLD
- Tables downloaded to:
  - Forwarding engine hardware for data plane forwarding (FIB/ADJ)
  - Replication engines for data plane packet replication (Multicast Expansion Table – MET)

IPv4 Multicast FIB TCAM Lookup



Egress Replication

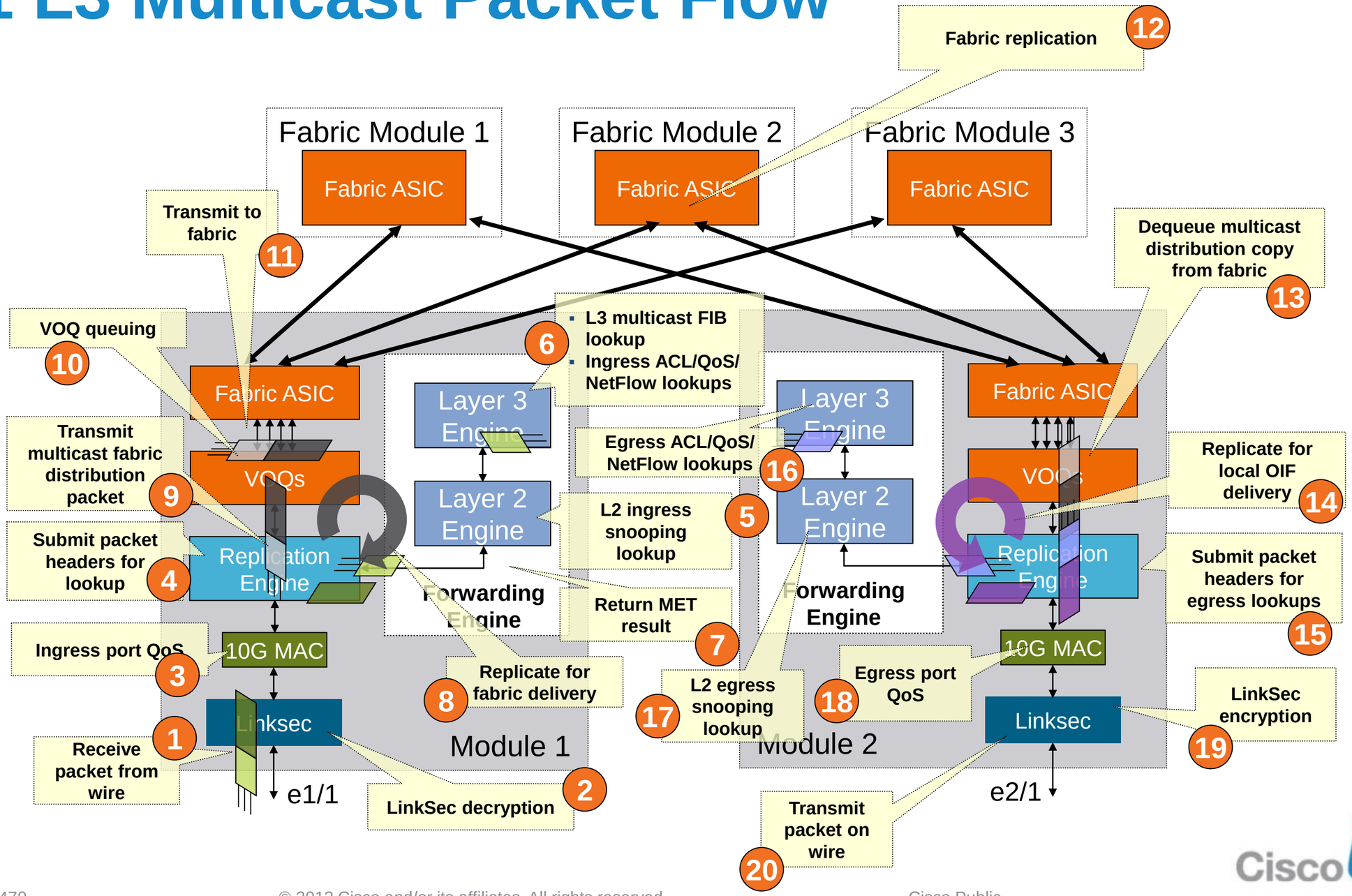
- Distributes multicast replication load among replication engines of all I/O modules with OIFs
- Input packets get lookup on ingress forwarding engine
- For OIFs on ingress module, ingress replication engine performs the replication
- For OIFs on other modules, ingress replication engine replicates a single copy of packet into fabric for those egress modules, fabric replicates as needed
- Each egress forwarding engine performs lookup to drive replication
- Replication engine on egress module performs replication for local OIFs



HDR = Packet Headers DATA = Packet Data

M Family

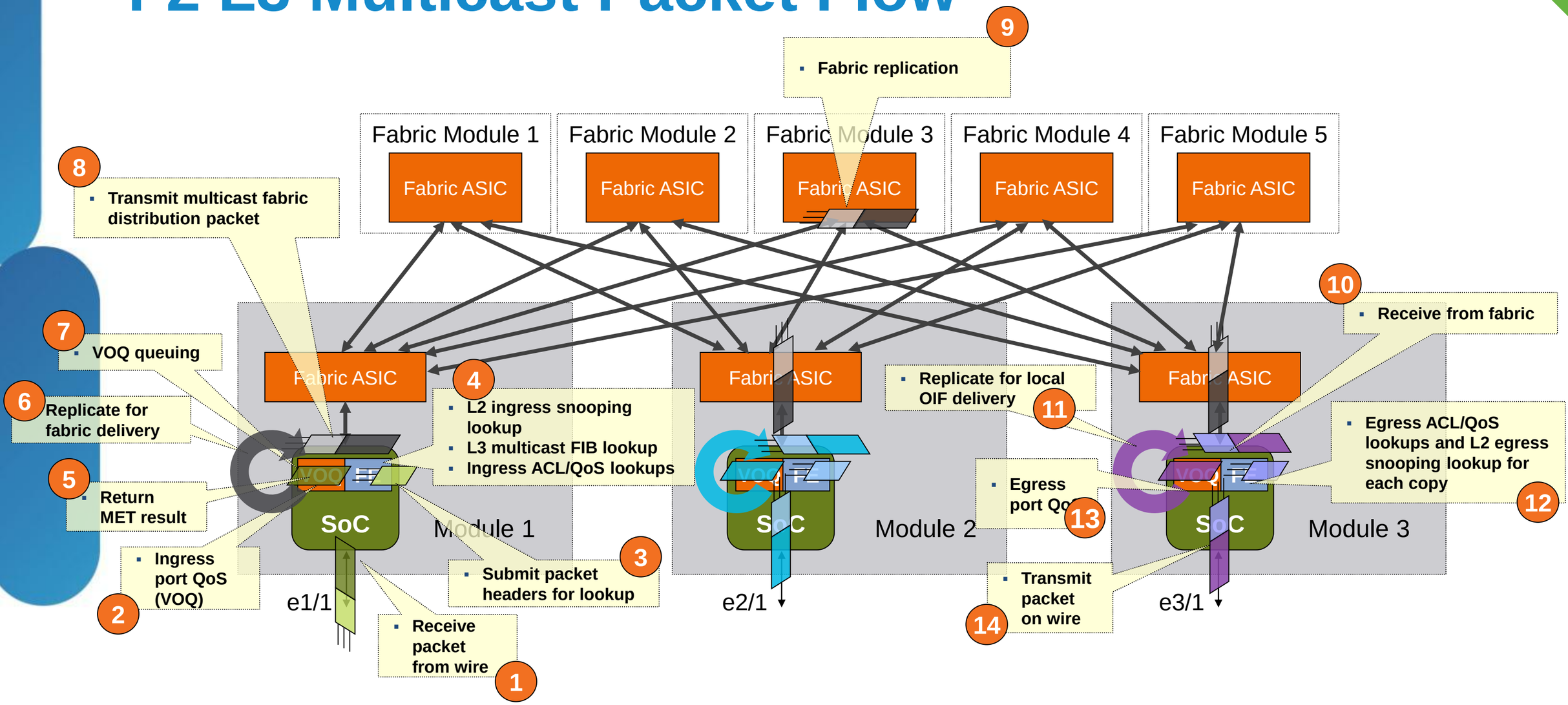
M1 L3 Multicast Packet Flow



HDR = Packet Headers DATA = Packet Data

F Family

F2 L3 Multicast Packet Flow



Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

What Is Classification?

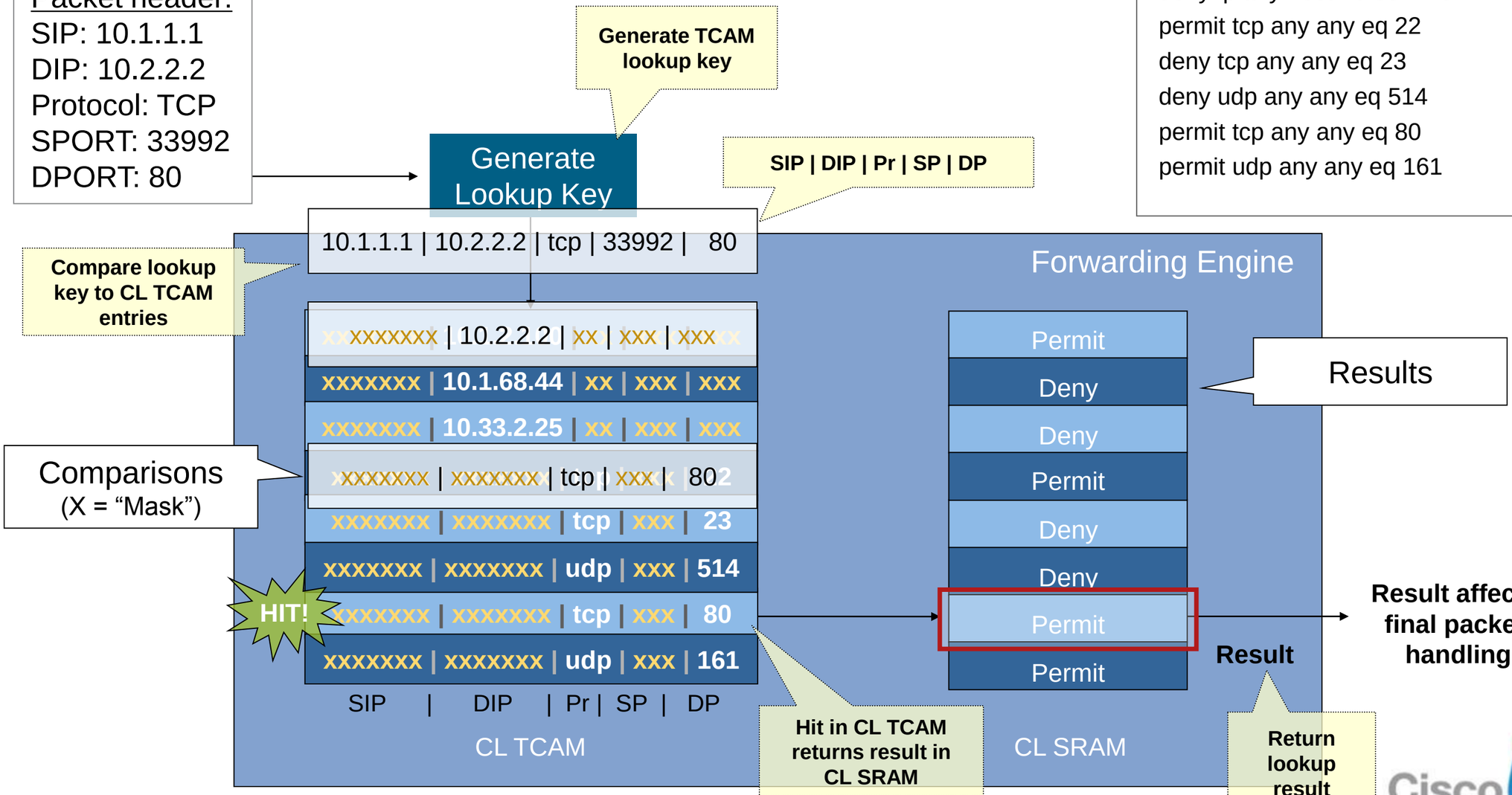
- Matching packets
  - Layer 2, Layer 3, and/or Layer 4 information
- Used to decide whether to apply a particular policy to a packet
  - Enforce security, QoS, or other policies
- Some examples:
  - Match TCP/UDP source/destination port numbers to enforce security policy
  - Match destination IP addresses to apply policy-based routing (PBR)
  - Match 5-tuple to apply marking policy
  - Match protocol-type to apply Control Plane Policing (CoPP)
  - etc.

CL TCAM Lookup – ACL

Security ACL

```
ip access-list example
 permit ip any host 10.1.2.100
 deny ip any host 10.1.68.44
 deny ip any host 10.33.2.25
 permit tcp any any eq 22
 deny tcp any any eq 23
 deny udp any any eq 514
 permit tcp any any eq 80
 permit udp any any eq 161
```

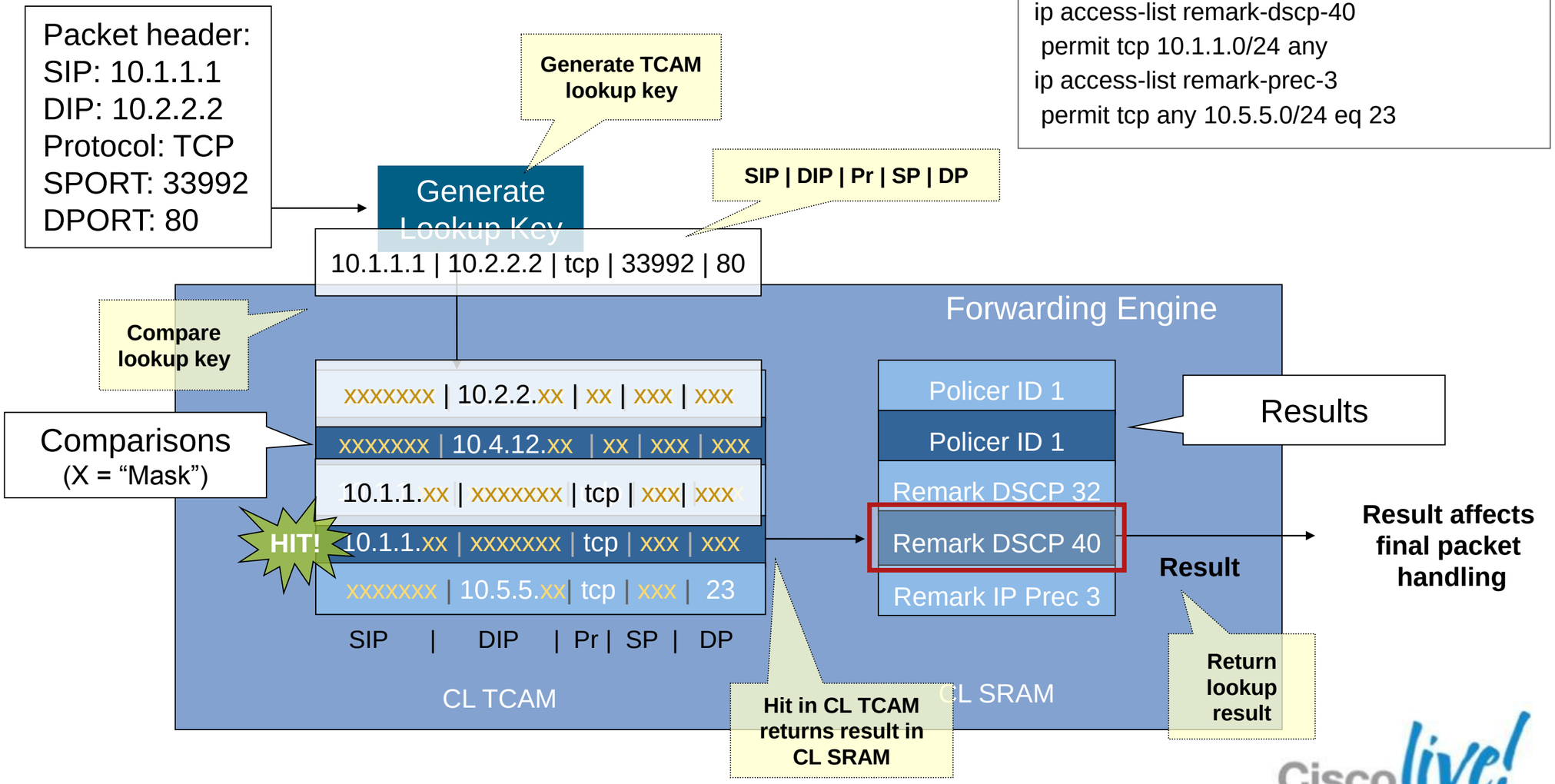
Packet header:
SIP: 10.1.1.1
DIP: 10.2.2.2
Protocol: TCP
SPORT: 33992
DPORT: 80



CL TCAM Lookup – QoS

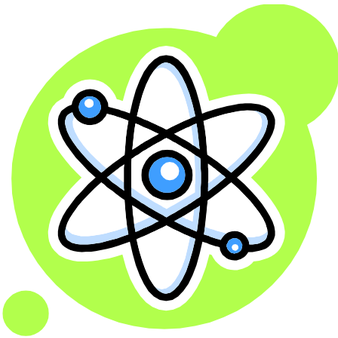
QoS Classification ACLs

```
ip access-list police
 permit ip any 10.3.3.0/24
 permit ip any 10.4.12.0/24
ip access-list remark-dscp-32
 permit udp 10.1.1.0/24 any
ip access-list remark-dscp-40
 permit tcp 10.1.1.0/24 any
ip access-list remark-prec-3
 permit tcp any 10.5.5.0/24 eq 23
```



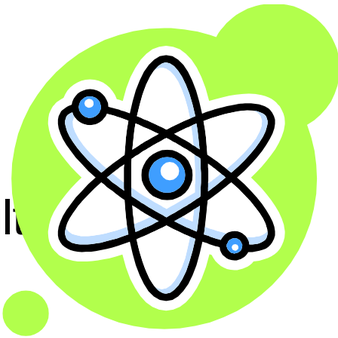
Atomic Policy Programming

- Avoids packet loss during policy updates
- Enabled by default
- Atomic programming process:
  - Program new policy in free/available CL TCAM entries
  - Enable new policy by swapping the ACL label on interface
  - Free CL TCAM resources used by previous policy



Atomic Policy Programming Cont.

- To support atomic programming, **software reserves 50% of available TCAM**
- If insufficient resources available, system returns an error and no modifications made in hardware
  - `Failed to complete Verification: Tcam will be over used, please turn off atomic update`
- Disable with **no platform access-list update atomic**
  - Disabling may be necessary for very large ACL configurations
  - Atomic programming attempted but not mandatory
- User can disable atomic programming and perform update non-atomically (assuming ACL fits in CL TCAM)
  - “Default” ACL result (deny by default) returned for duration of reprogramming
  - Use **[no] hardware access-list update default-result permit** to control default result



Classification Configuration Sessions

Two ways to configure ACL/QoS policies:

- Normal configuration mode (**config terminal**)
  - Configuration applied immediately line by line
  - Recommended only for small ACL/QoS configurations, or non-data-plane ACL configuration
- Session config mode (**config session**)
  - Configuration only applied after **commit** command issued
  - Recommended for large ACL/QoS configurations
- Config session mode also provides **verify** facility to “dry-run” the configuration against available system resources
  - No change to existing hardware configuration after verification (regardless of verification result)

Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

NetFlow on Nexus 7000

- NetFlow collects flow data for packets traversing forwarding engines
- Per-interface full and sampled NetFlow provided by M1 module hardware

	M1→M1	M1→F1	F1→M1	F1→F1	F2→F2
Bridged	Yes	Yes	No	No	No**
Routed	Yes	Yes	Yes*	Yes*	No**

* From release 5.2(1)
** Hardware supports ingress sampled NetFlow

- Each M1 module maintains independent NetFlow table
  - 512K hardware entries per forwarding engine
- Hardware NetFlow entry creation
  - CPU not involved in NetFlow entry creation/update

Full vs. Sampled NetFlow

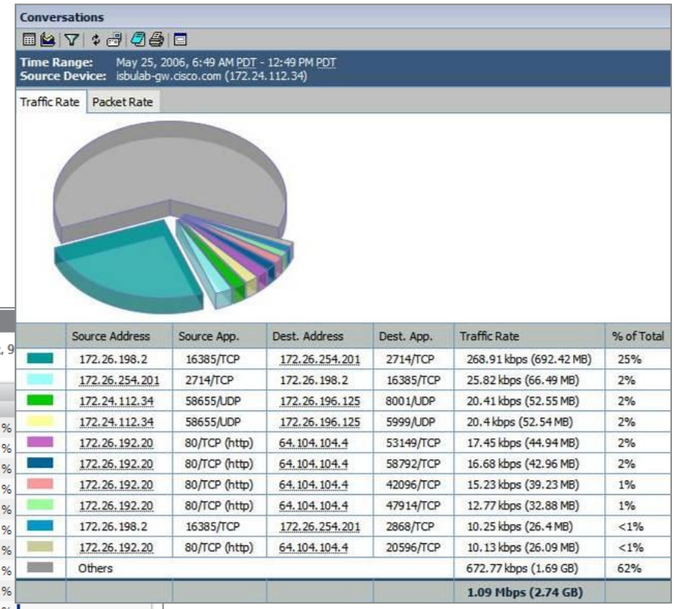
- NetFlow configured per-direction and per-interface
  - Ingress and/or egress on per-interface basis
- Each interface can collect full or sampled flow data
- Full NetFlow: Accounts for every packet of every flow on interface, up to capacity of NetFlow table
- Sampled NetFlow: Accounts for M in N packets on interface, up to capacity of NetFlow table

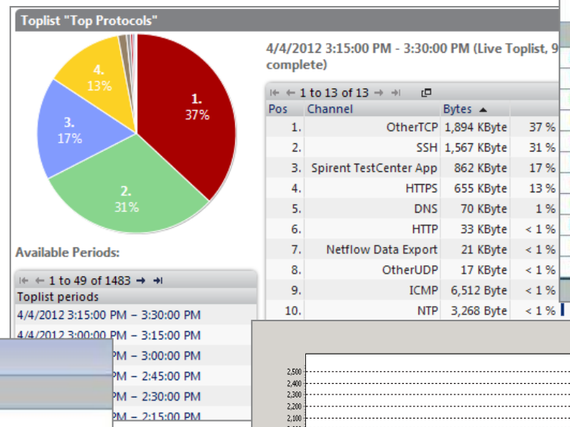
Sampled NetFlow Details

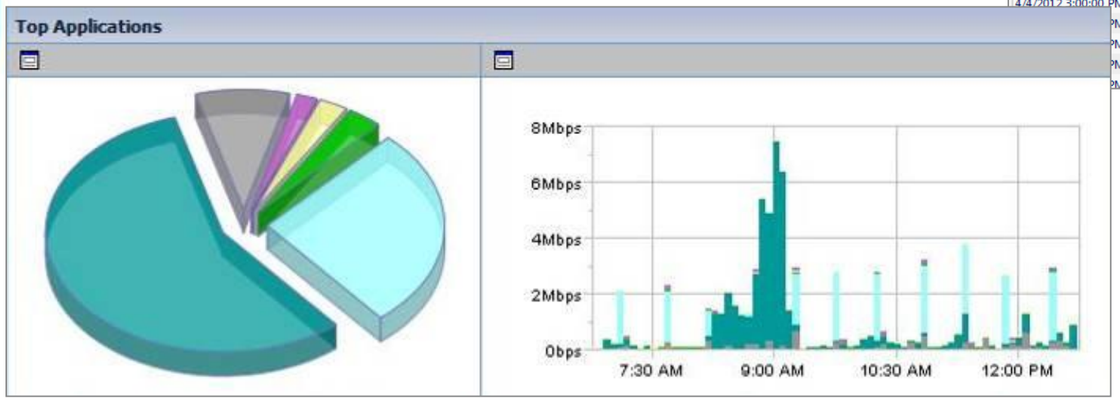
- Random packet-based sampling
- M:N sampling: Out of N consecutive packets, select M consecutive packets and account only for those flows in the hardware NetFlow table
- Sampled flows aged and exported from NetFlow table normally
- Advantages
  - Reduces NetFlow table utilization
  - Reduces CPU load on switch and collector
- Disadvantages
  - Some flows may not be accounted
  - Collector extrapolates total traffic load based on configured sampling rate

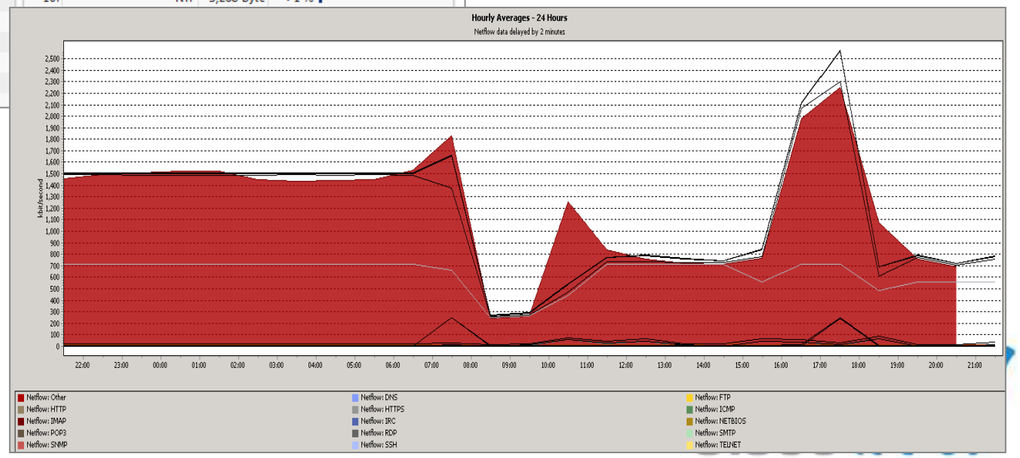
Netflow Data Export (NDE)

- Process of exporting statistics data from network devices to a “collector”
- Allows long-term baselining, trending, and analysis of NetFlow data
- Exported data sent via UDP
- Variety of export “formats” exist
  - Exported data and format of records varies from version to version

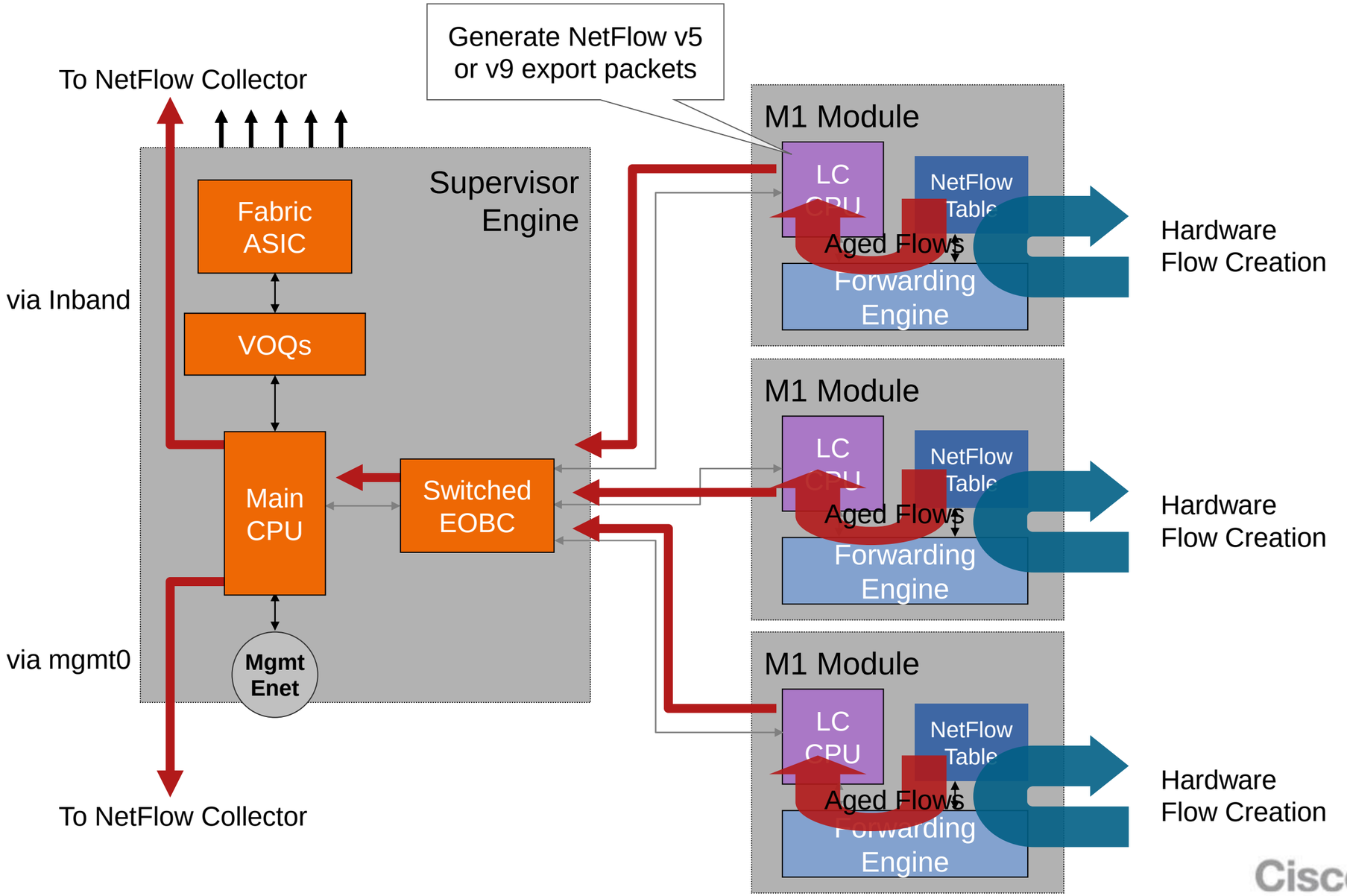








NetFlow Data Export Process



Agenda

- Chassis Architecture
- Supervisor Engine and I/O Module Architecture
- Forwarding Engine Architecture
- Fabric Architecture
- I/O Module Queuing
- Layer 2 Forwarding
- IP Forwarding
- IP Multicast Forwarding
- Classification
- NetFlow
- Conclusion

Nexus 7000 Architecture Summary

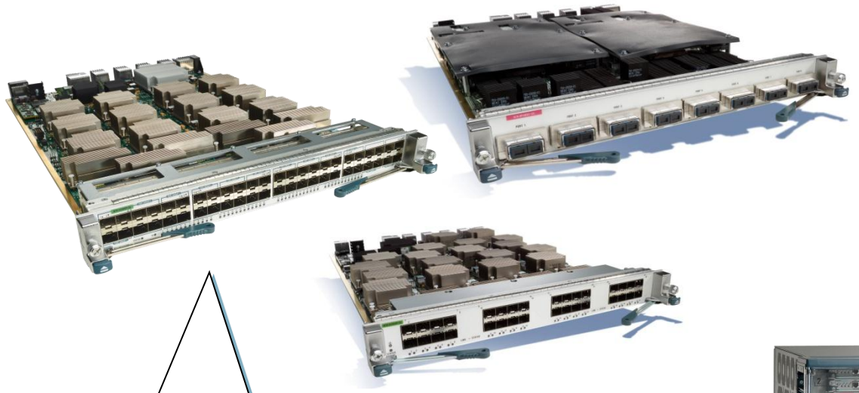
Control plane protocols, system and network management

Supervisor Engine

Future-proofed chassis designs with density and airflow options

Chassis

I/O Modules



Variety of front-panel interface and transceiver types with hardware-based forwarding and services, including unicast/multicast, bridging/routing, ACL/QoS classification, and NetFlow statistics

Fabrics

High-bandwidth fabric to interconnect I/O modules and provide investment protection

Conclusion

- You should now have a thorough understanding of the Nexus 7000 switching architecture, I/O module design, packet flows, and key forwarding engine functions…
- **Any questions?**

Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Passport points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

Don’t forget to activate your Cisco Live Virtual account for access to all session material, communities, and on-demand and live activities throughout the year. Activate your account at the Cisco booth in the World of Solutions or visit www.ciscolive.com

Final Thoughts

- Get hands-on experience with the Walk-in Labs located in World of Solutions, booth 1042
- Come see demos of many key solutions and products in the main Cisco booth 2924
- Visit www.ciscoLive365.com after the event for updated PDFs, on-demand session videos, networking, and more!
- Follow Cisco Live! using social media:
  - Facebook: https://www.facebook.com/ciscoliveus
  - Twitter: https://twitter.com/#!/CiscoLive
  - LinkedIn Group: http://linkd.in/CiscoLI

BUILT FOR
THE HUMAN
NETWORK