# Cisco Nexus 5000 and 2000 Hardware Architecture

BRKARC-3452

# Session Goal

- This session presents an in-depth study of the architecture of the Nexus 5000 Data Center switch and the Nexus 2000 Fabric Extender. Topics include internal architecture of the Nexus 5000 and 2000, the architecture of VN-Link port extension as implemented in the Nexus 2000, Unified I/O, and 10G cut-thru Ethernet
- Related sessions:

  BRKARC-3470 - Cisco Nexus 7000 Switch Architecture

  BRKARC-3471 - Cisco NXOS Software Architecture

  BRKDCT-2079 - The Evolution of Data Center Networks

  BRKDCT-2023 Evolution of the Data Centre Access Architecture

  BRKSAN-2047 – FCoE Design, Operations and Management Best Practices

# Agenda

- Data Center Virtualized Access—Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch- Nexus 5000 and Nexus 2000

# Nexus 5000 and 2000

## Virtualized Data Center Access

### Nexus 5010

**20 Fixed Ports 10G/FCoE/IEEE DCB**
**Line-rate, Non-blocking 10G**
**1 Expansion Module Slot**
**Redundant Fans & Power Supplies**

### Nexus 2000 Fabric Extender

**48 Fixed Ports 1G Ethernet (1000 BASE-T)**
**4 Fixed Port 10G Uplink**
**Distributed Virtual Line Card**

### Nexus 5020

**40 Fixed Ports 10G/FCoE/IEEE DCB**
**Line-rate, Non-blocking 10G**
**2 Expansion Module Slots**
**Redundant Fans & Power Supplies**

### Ethernet

**6 Ports 10G/FCoE/IEEE DCB**

### Ethernet + Fibre Channel

**4 Ports 10G/FCoE/IEEE DCB**
**4 Ports 1/2/4G Fibre Channel**

### Fibre Channel

**8 Ports 1/2/4G Fibre Channel**

### 8G Fibre Channel

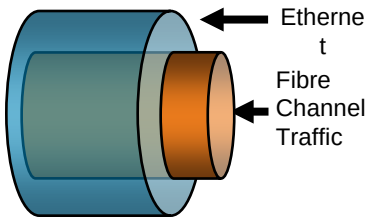
**6 Ports 2/4/8G Fibre Channel**

# Nexus 5000 and 2000

## Virtualized Data Center Access

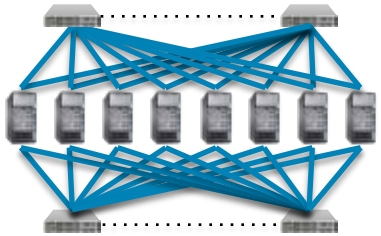
### Distributed Access Fabric

- Nexus 2000 and Fabric Extender Technology
- Centralized Control and Mgmt with Distributed Forwarding
- VN-Tag Enabled Port Extension
- Low Latency Cut-Thru Switching



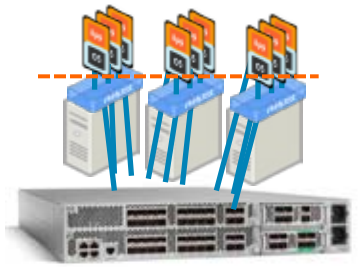

### Unified Fabric

- Native Fibre Channel supporting full SAN Fabric Capabilities
- FCoE enabled - T11 FC-BB-5
- Data Center Bridging Exchange Protocol (DCBX)
- Ingress Queuing, Virtual Output Queues and Lossless Fabric



### Ethernet Enhancements

- Virtual Port Channel (VPC)
- Layer 2 Multipath (TRILL & ETRILL)
- Priority Flow Control (PFC) IEEE 802.1Qbb
- Bandwidth Management IEEE 802.1Qaz

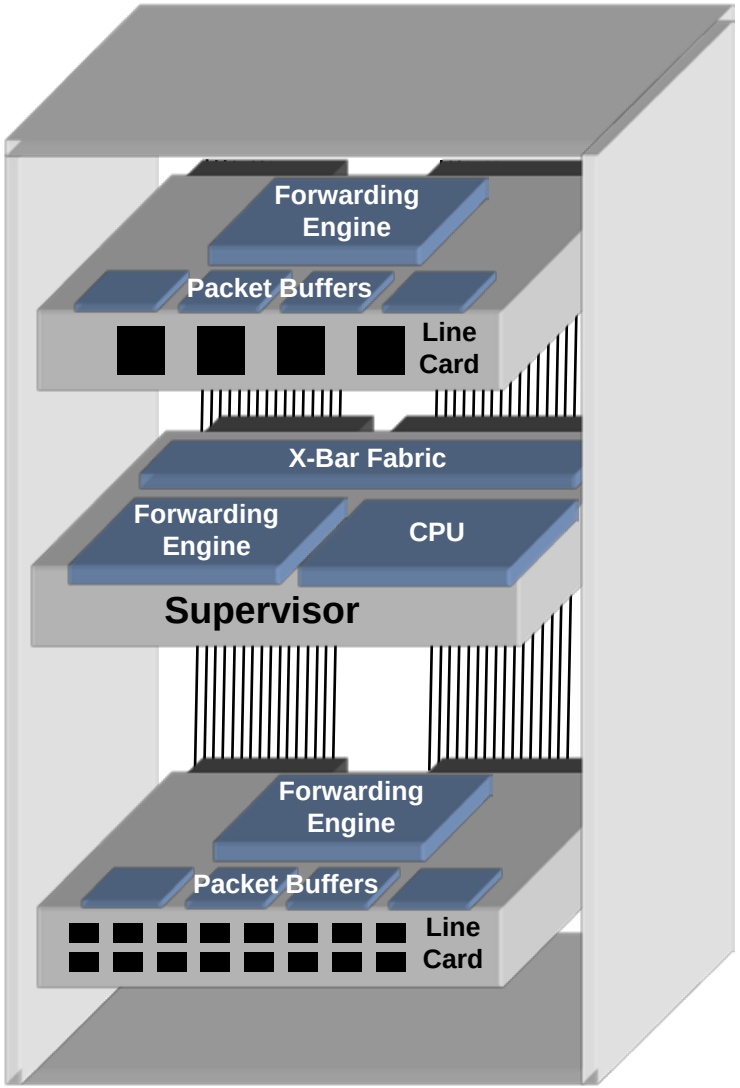


### VN-Link

- VN-Tag Port Extension+ SR-IOV
- Network Interface Virtualization (NIV)
- Port Profiles

# Nexus 5000 and 2000 Architecture
## Switch Morphology

- What's in a switch?
- Lookup/forwarding logic
  - L2/L3 forwarding, ACL, QoS TCAM
- CPU
  - Control and management
- Packet transport
  - Cross bar switching fabric
  - Component interconnects
- Ports (line cards)
  - Port buffers
  - Fabric buffers
- Not all switches are identical in architecture but have certain principles in common

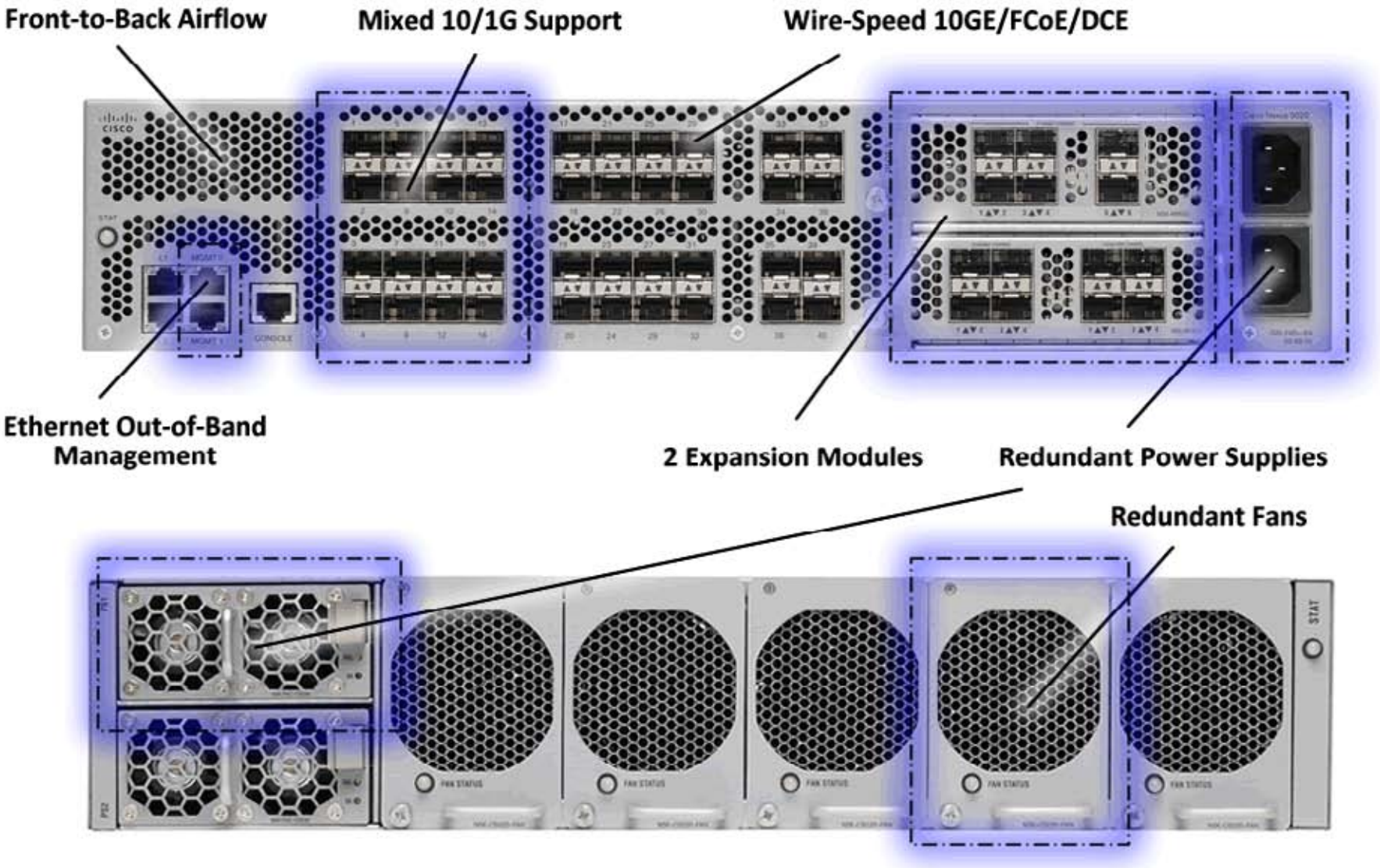


# Agenda

- Data Center Virtualized Access—Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

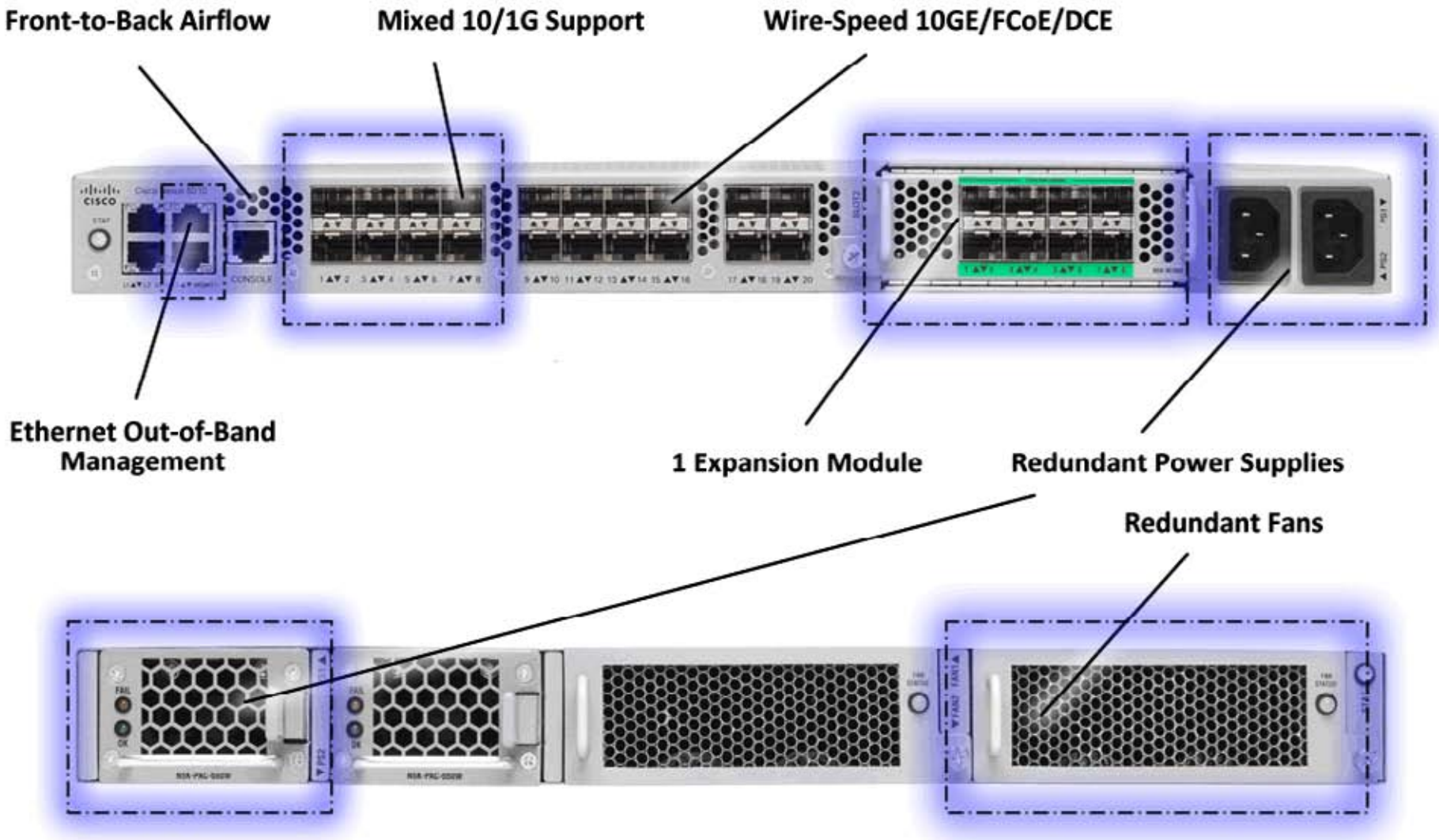
# Nexus 5000 and 2000 Architecture

## Nexus 5020



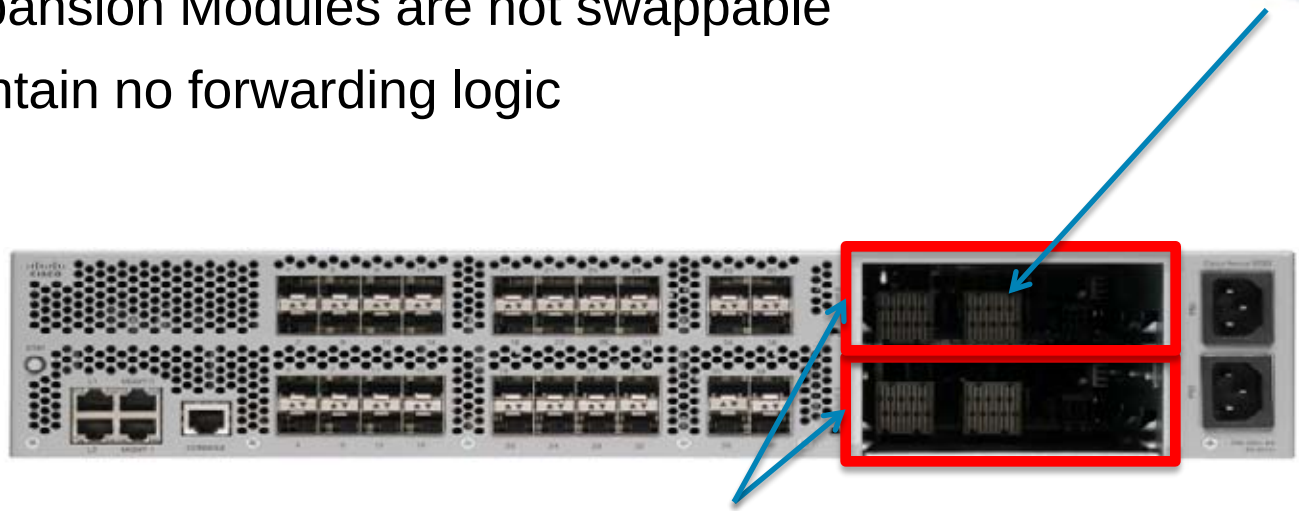
# Nexus 5000 and 2000 Architecture

## Nexus 5010



# Nexus 5000 and 2000 Architecture
## Expansion Modules

- Nexus 5000 utilizes expansion slots to provide flexibility of interface types
  - Additional 10GE DCB/FCoE compliant ports
  - 1/2/4/8G Fibre Channel ports
- Nexus 5020 has two expansion module slots
- Nexus 5010 has one expansion module slot
- Expansion Modules are hot swappable
- Contain no forwarding logic



# Nexus 5000 and 2000 Architecture

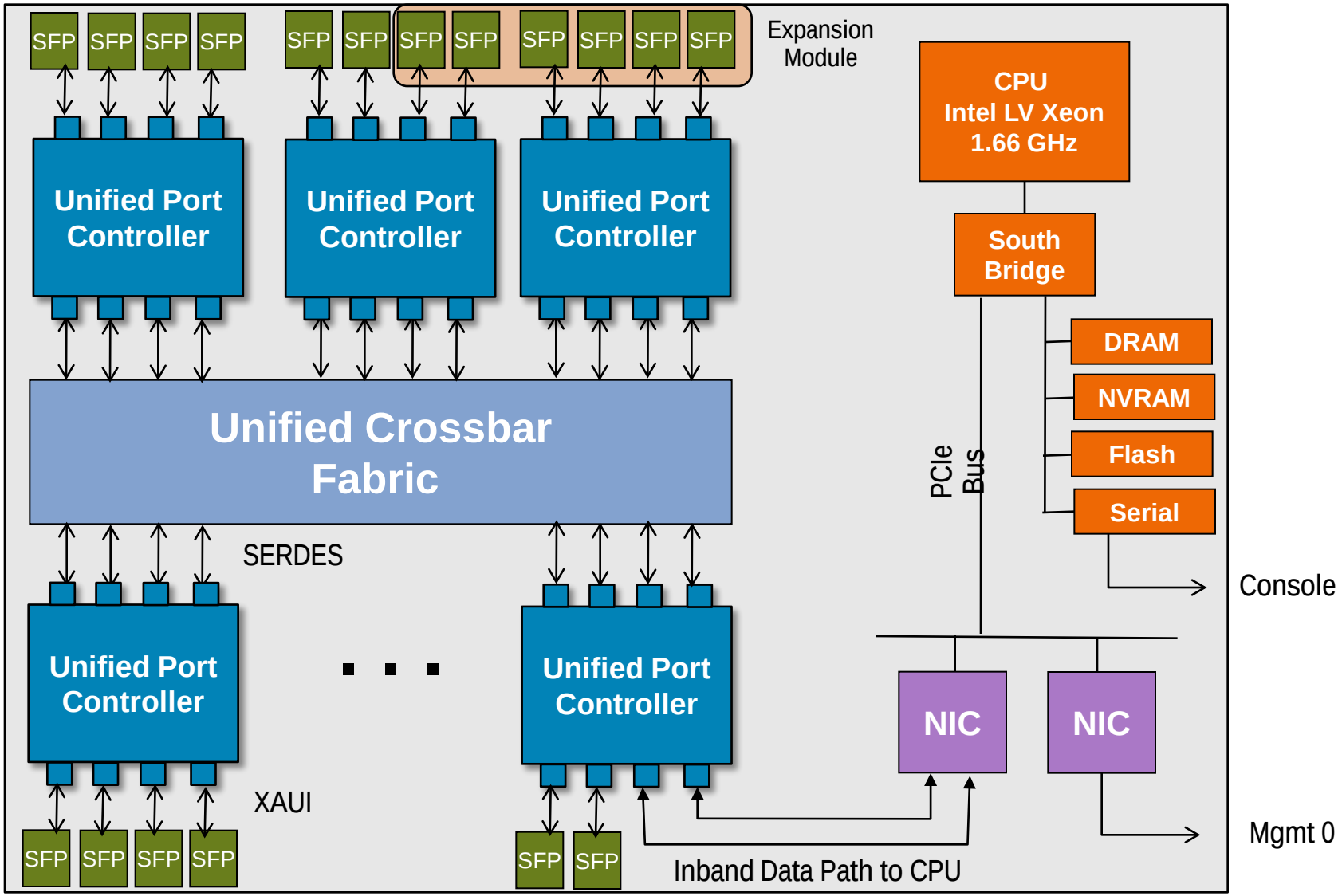
## Power Supplies

- Nexus 5020 power supplies
  - 1200 watt (N5K-PAC-1200W)
  - 750 watt (N5K-PAC-750W)
- Fully-loaded Nexus 5020 with 2 expansion modules and all links running at line rate only requires a ***single*** 750 watt power supply

```
dc11-5020-3# sh environment power

Power Supply:
Voltage: 12 Volts
-----------------------------------------------------
PS  Model                Power       Power     Status
                         (Watts)     (Amp)
-----------------------------------------------------
1   N5K-PAC-1200W        1200.00    100.00     ok
2   N5K-PAC-1200W        1200.00    100.00     ok

<snip>
Total Power Capacity                              2400.00 W

Power reserved for Supervisor(s)                   625.20 W
Power currently used by Modules                     72.00 W


                                                  -------------
Total Power Available                              1702.80 W
                                                  -------------
```

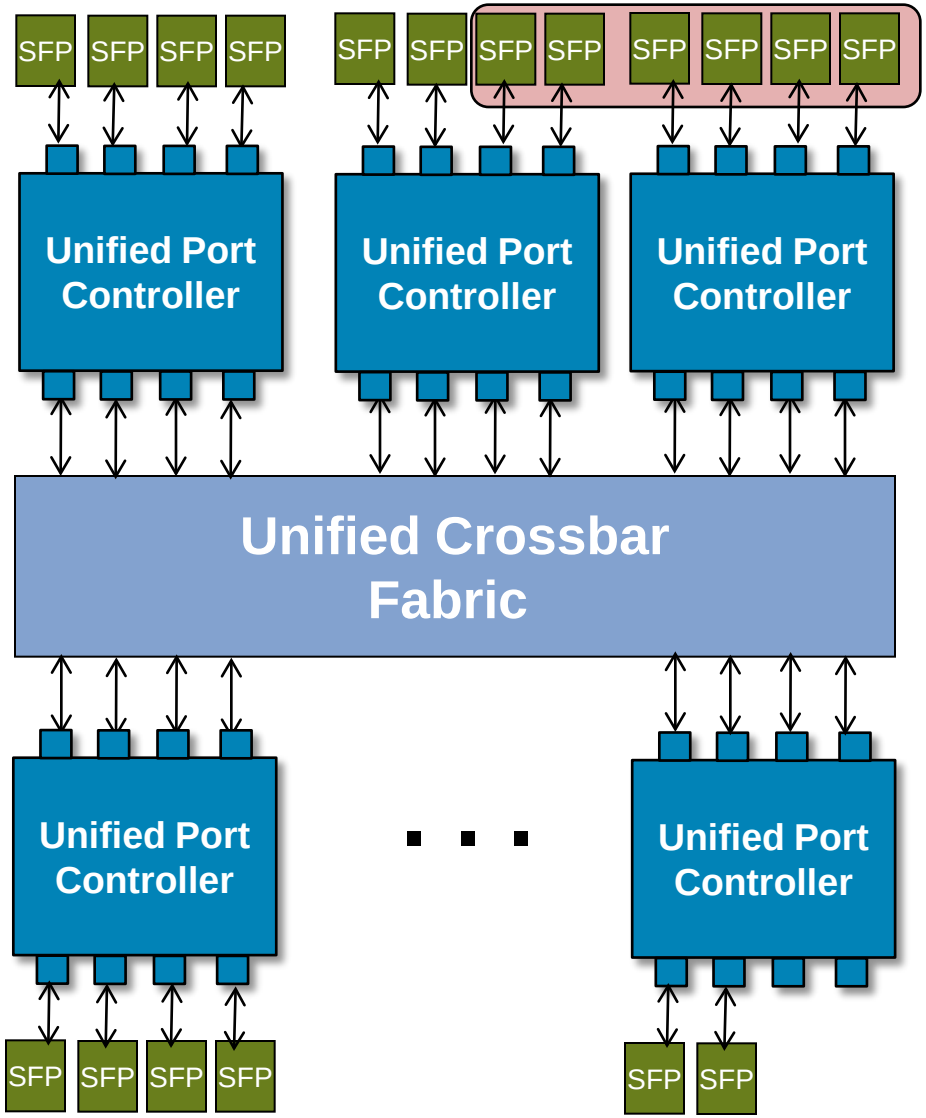
# Nexus 5000 Hardware Overview

## Data and Control Plane Elements



# Nexus 5000 Hardware Overview

## Data Plane Elements




- Nexus 5000 is a distributed forwarding architecture
- Cisco Nexus 5020: Layer 2 hardware forwarding at 1.04 Tbps or 773.8 million packets per second (mpps)
- Unified Port Controller (UPC) ASIC interconnected by a single stage Unified Crossbar Fabric (UCF)
- Unified Port Controllers provide distributed packet forwarding capabilities
- ***All*** port to port traffic passes through the UCF (Fabric)
- Four switch ports managed by each UPC
  - 14 UPC in Nexus 5020
  - 7 UPC in Nexus 5010

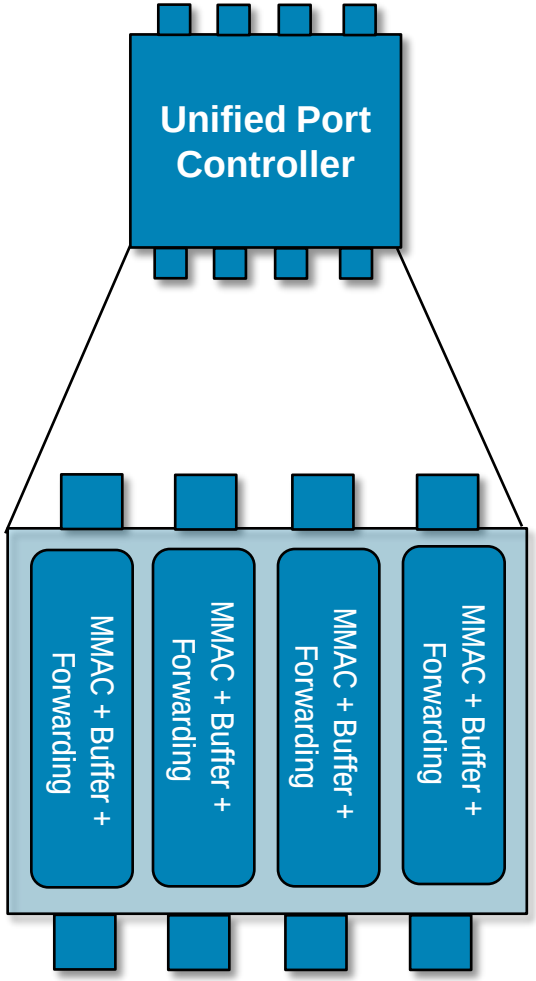
# Nexus 5000 Hardware Overview

## Unified Crossbar Fabric

- 58-port crossbar and scheduler
  - Three unicast and one multicast crosspoints
- Central tightly coupled scheduler
  - Request, propose, accept, grant, and acknowledge semantics
  - Packet enhanced iSLIP scheduler
- Distinct unicast and multicast schedulers
- Eight classes of service within the Fabric

# Nexus 5000 Hardware Overview
## Unified Port Controller

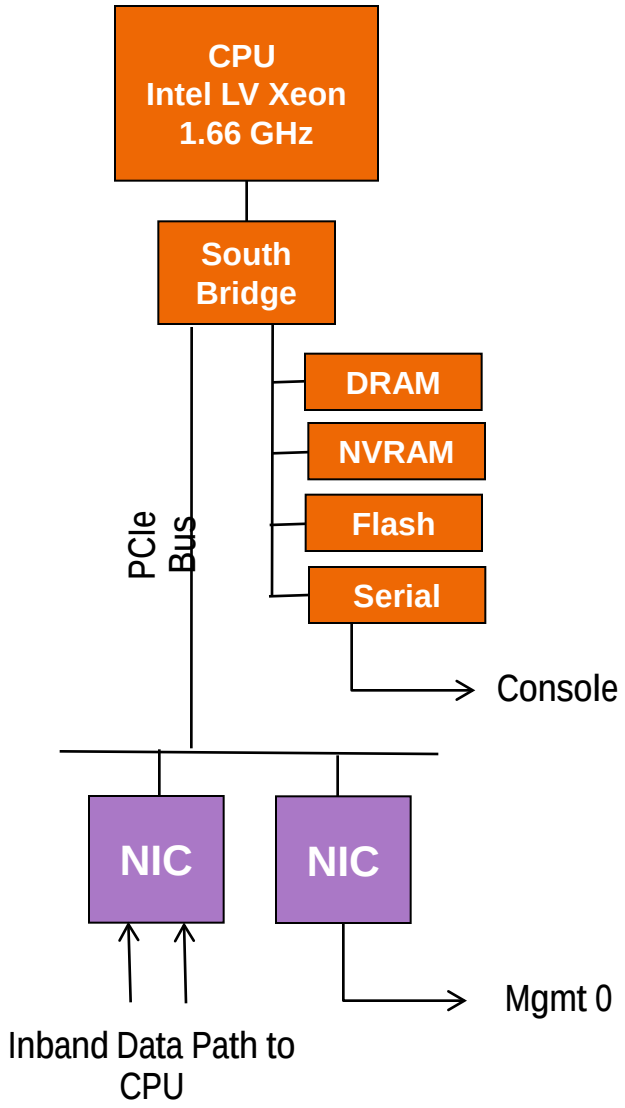
- Each UPC supports four ports and contains,
- Multimode Media access controllers (MAC)
  - Support 1/10 G Ethernet and 1/2/4 G Fibre Channel
  - (2/4/8 G Fibre Channel MAC is located on the Expansion Module)
- Packet buffering and queuing
  - 480 KB of buffering per port
- Forwarding controller
  - Ethernet and Fibre Channel Forwarding and Policy



# Nexus 5000 Hardware Overview

## Control Plane Elements

- CPU - 1.66 GHz Intel LV Xenon
- DRAM - 2 GB of DDR2 400 (PC2 3200) in two DIMM slots
- Program Store - 1 GB of USB-based (NAND) flash
- Boot/BIOS - 2 MB of EEPROM with locked recovery image
- On-Board Fault Log - 64 MB of flash for failure analysis
- NVRAM - 2 MB of SRAM: Syslog and licensing information
- Management Interfaces - RS-232 console port: console 0
- Mgmt 0 interface partitioned from in-band VLANs



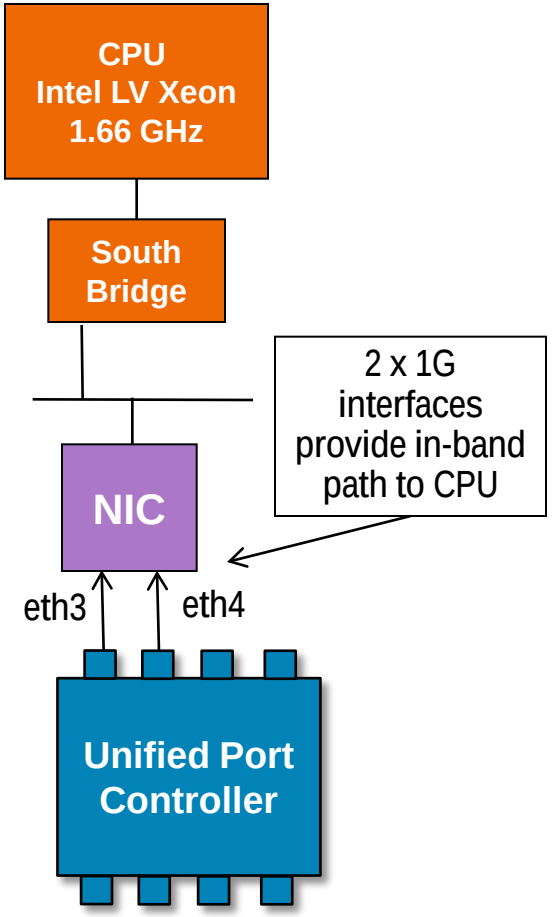
# Nexus 5000 Hardware Overview

## Control Plane Elements

- In-band traffic is identified by the UPC and punted to the CPU via two dedicated UPC interfaces, 5/0 and 5/1, which are in turn connected to eth3 and eth4 interfaces in the CPU complex
- Eth3 handles Rx and Tx of ***low*** priority control pkts

  IGMP, CDP, TCP/UDP/IP/ARP (for management purpose only)
- Eth4 handles Rx and Tx of ***high*** priority control pkts

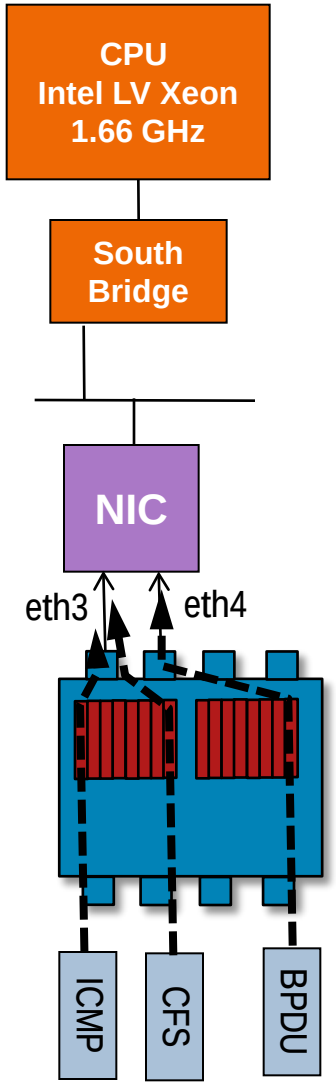
  STP, LACP, DCBX, FC and FCoE control frames (FC packets come to Switch CPU as FCoE packets)



# Nexus 5000 Hardware Overview

## Control Plane Elements

- CPU queuing structure provides strict protection and prioritization of inbound traffic
- Each of the two in-band ports has 8 queues and traffic is scheduled for those queues based on control plane priority (traffic CoS value)
- Prioritization of traffic between queues on each in-band interface
  - CLASS 7 is configured for strict priority scheduling (e.g. BPDU)
  - CLASS 6 is configured for DRR scheduling with 50% weight
  - Default classes (0 to 5) are configured for DRR scheduling with 10% weight
- Additionally each of the two in-band interfaces has a priority service order from the CPU
  - Eth 4 interface has high priority to service packets (no interrupt moderation)
  - Eth3 interface has low priority (interrupt moderation)



# Nexus 5000 Hardware Overview

## Control Plane Elements

- Monitoring of in-band traffic vis the NX-OS built-in ethanalyzer

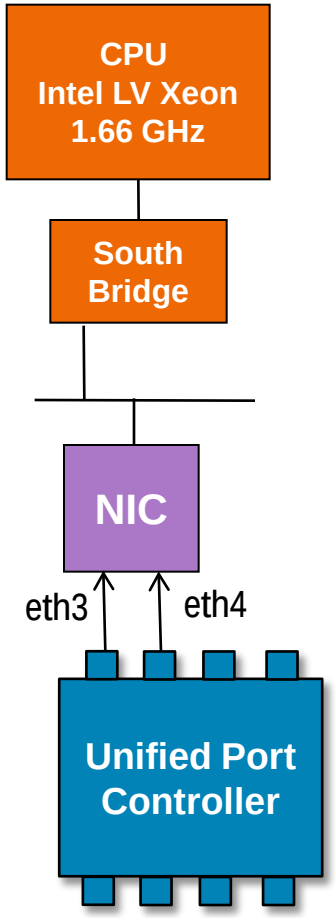
  Eth3 is equivalent to ‘inbound-lo’

  Eth4 is equivalent to ‘inbound-hi’

```
dc11-5020-3# ethanalyzer local sniff-interface ?
  inbound-hi   Inbound(high priority) interface
  inbound-low  Inbound(low priority) interface
  mgmt         Management interface
```

- CLI view of in-band control plane data

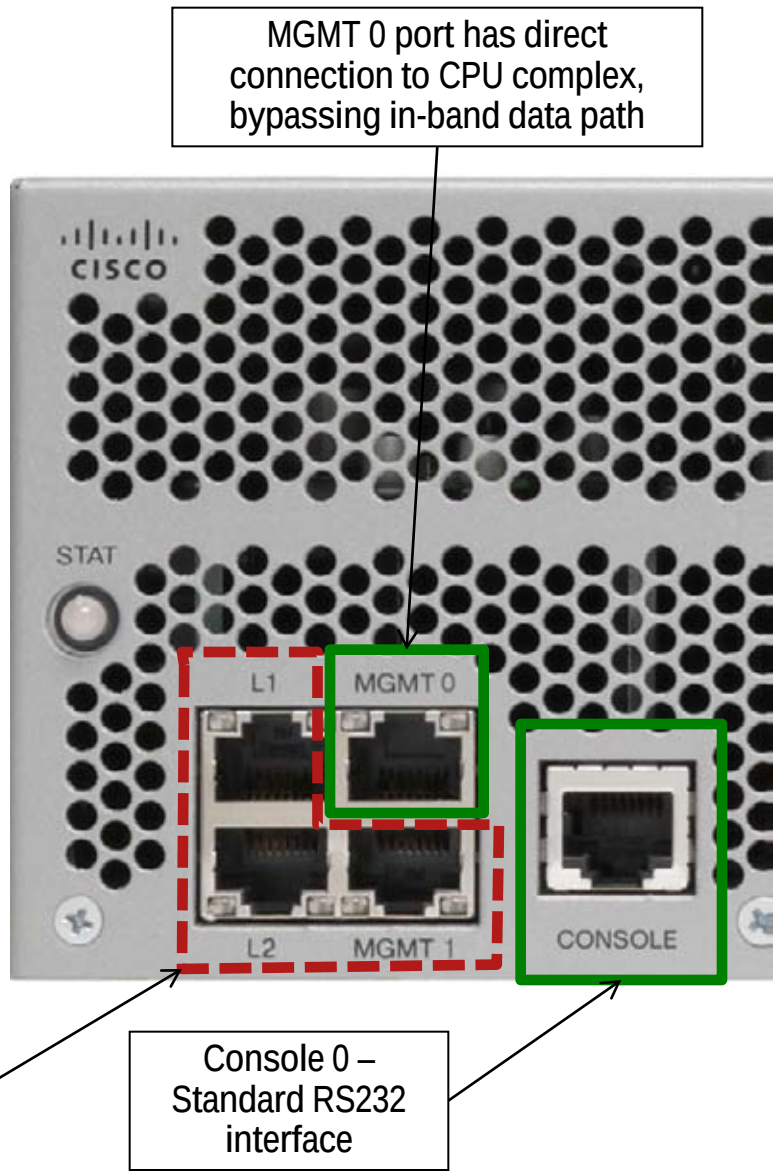
```
dc11-5020-4# sh hardware internal cpu-mac inband counters
eth3      Link encap:Ethernet  HWaddr 00:0D:EC:B2:0C:83
          UP BROADCAST RUNNING PROMISC ALLMULTI MULTICAST  MTU:2200  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:630 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:252 (252.0 b)  TX bytes:213773 (208.7 KiB)
          Base address:0x6020 Memory:fa4a0000-fa4c0000

eth4      Link encap:Ethernet  HWaddr 00:0D:EC:B2:0C:84
          UP BROADCAST RUNNING PROMISC ALLMULTI MULTICAST  MTU:2200  Metric:1
          RX packets:85379 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92039 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33960760 (32.3 MiB)  TX bytes:25825826 (24.6 MiB)
          Base address:0x6000 Memory:fa440000-fa460000
```



# Nexus 5000 Hardware Overview

## Mgmt and Console Ports

- Nexus 5000 only supports a single out of band management connection – “MGMT 0”
- MGMT 0 is isolated from any data plane traffic – dedicated VRF and bypassing in- band data plane(provides secure access to management plane)
- Ports L1 and L2 are reserved for use for ‘Unified Computing System’ Fabric Interconnects (Nexus 5000 and UCS 6100 share common HW platform)
- MGMT 1 is not currently enabled

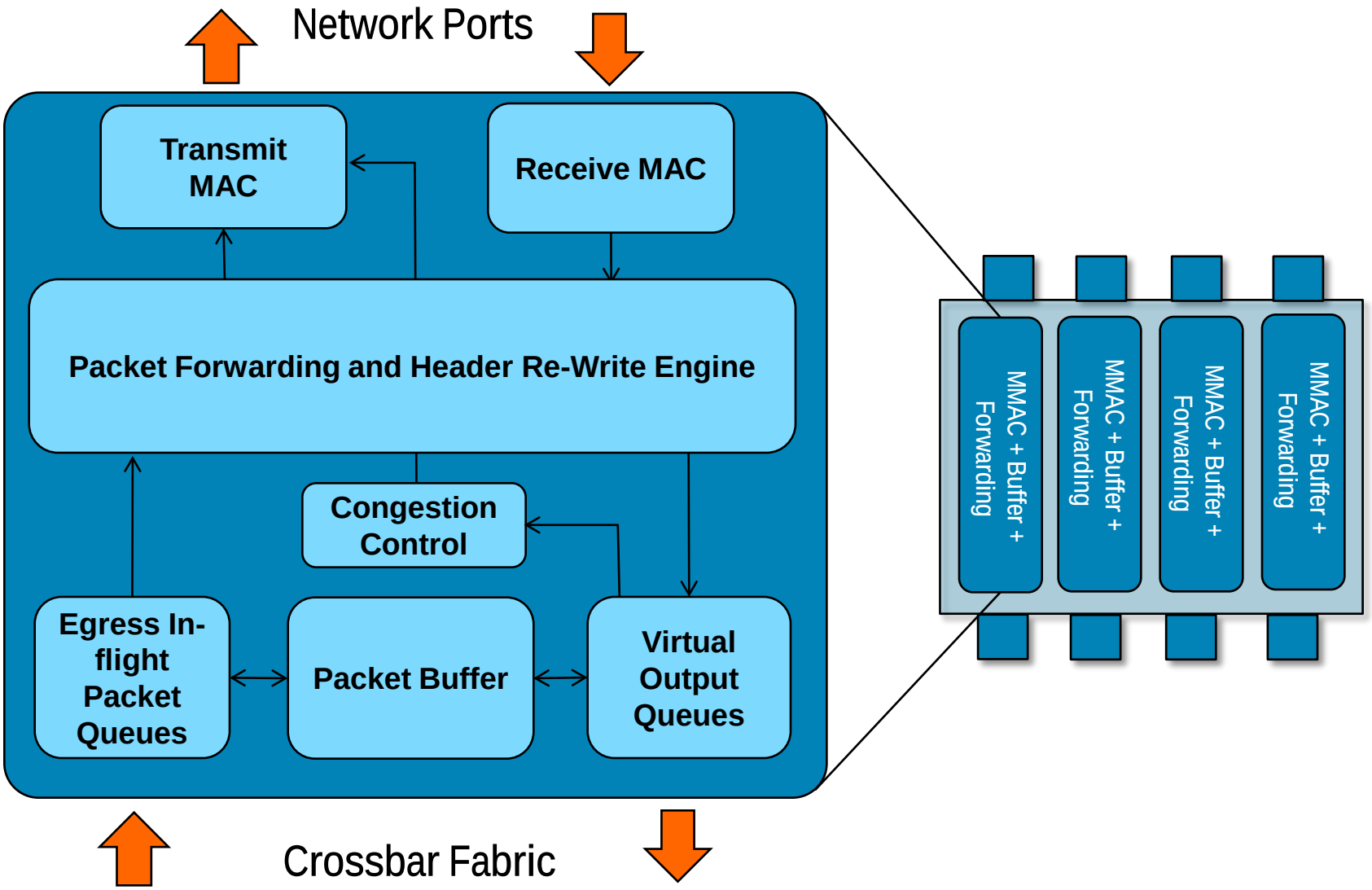


# Agenda

- Data Center Virtualized Access — Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

# Nexus 5000 Hardware Overview

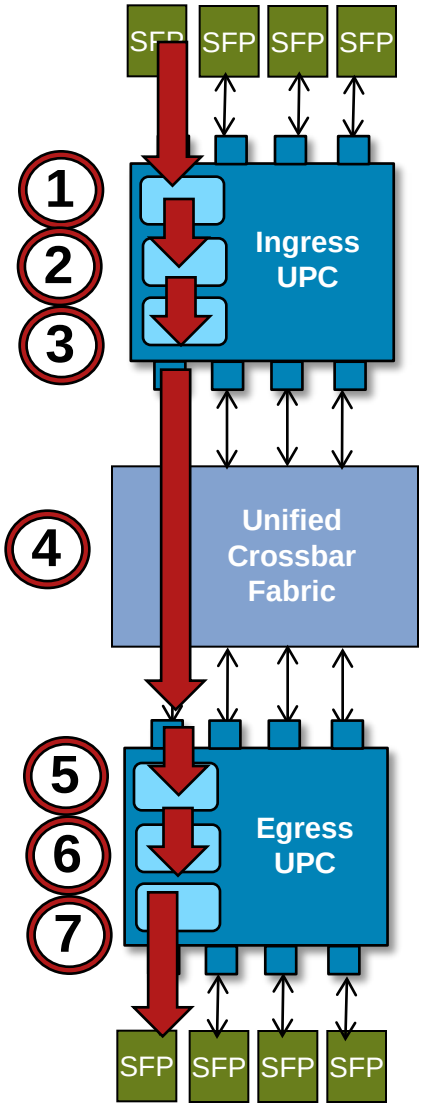
## UPC Details



# Nexus 5000 Hardware Overview

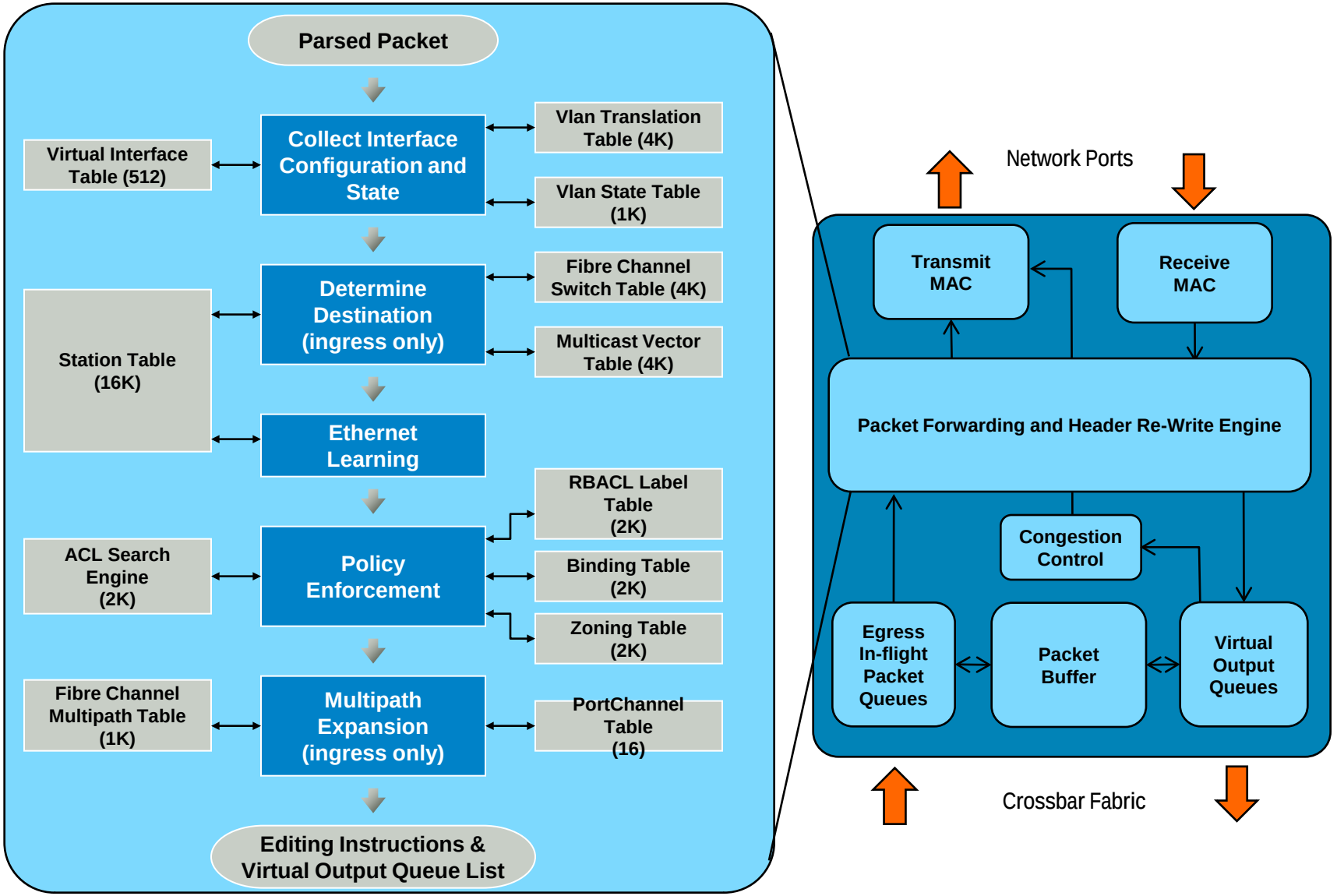
## Packet Forwarding Overview

1. Ingress MAC - MAC decoding, MACSEC processing (not supported currently), synchronize bytes
2. Ingress Forwarding Logic - Parse frame and perform forwarding and filtering searches, perform learning apply internal DCE header
3. Ingress Buffer (VoQ) - Queue frames, request service of fabric, dequeue frames to fabric and monitor queue usage to trigger congestion control
4. Cross Bar Fabric - Scheduler determines fairness of access to fabric and determines when frame is de-queued across the fabric
5. Egress Buffers - Landing spot for frames in flight when egress is paused
6. Egress Forwarding Logic - Parse, extract fields, learning and filtering searches, perform learning and finally convert to desired egress format
7. Egress MAC - MAC encoding, pack, synchronize bytes and transmit



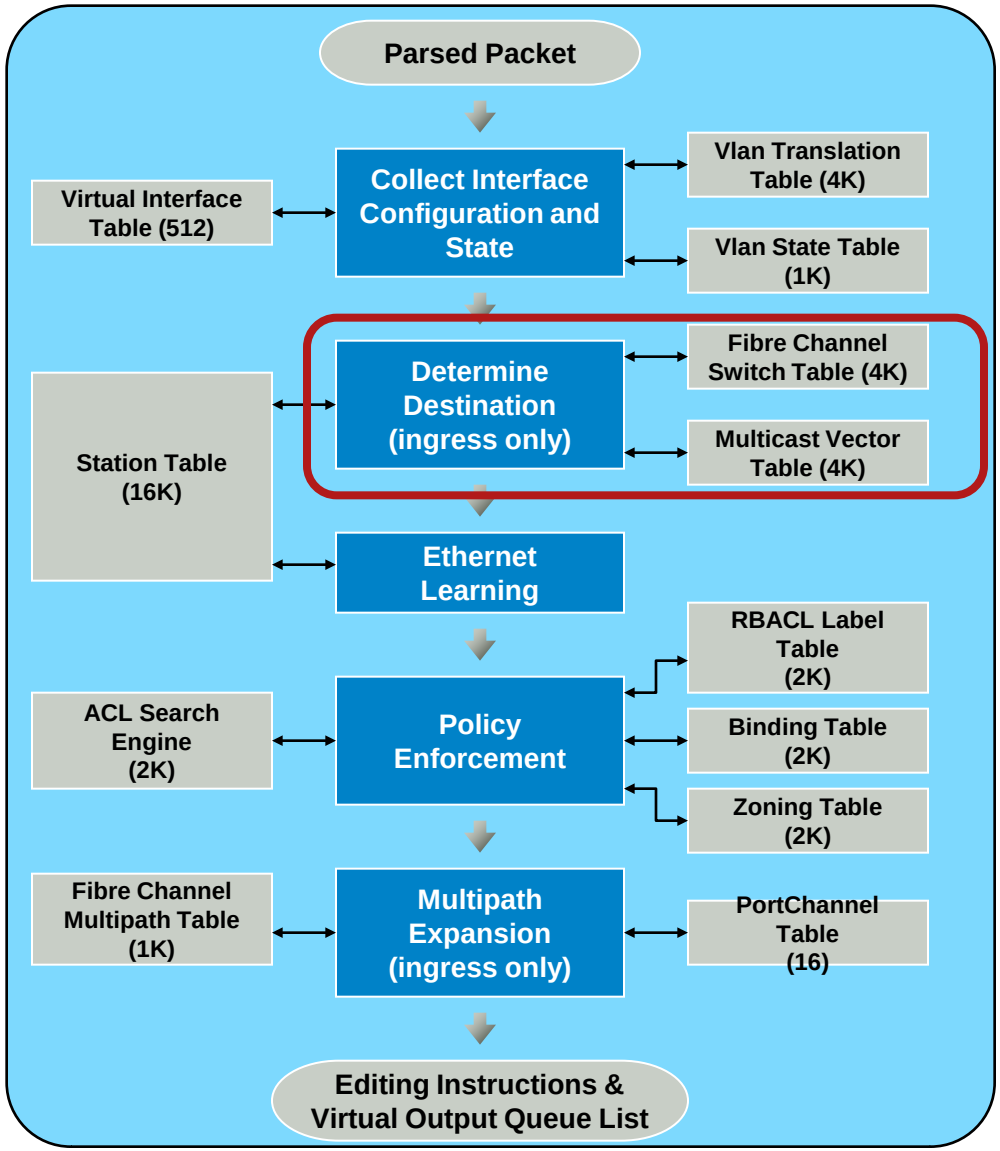
# Nexus 5000 Hardware Overview

## UPC Forwarding Details



# Nexus 5000 Hardware Overview

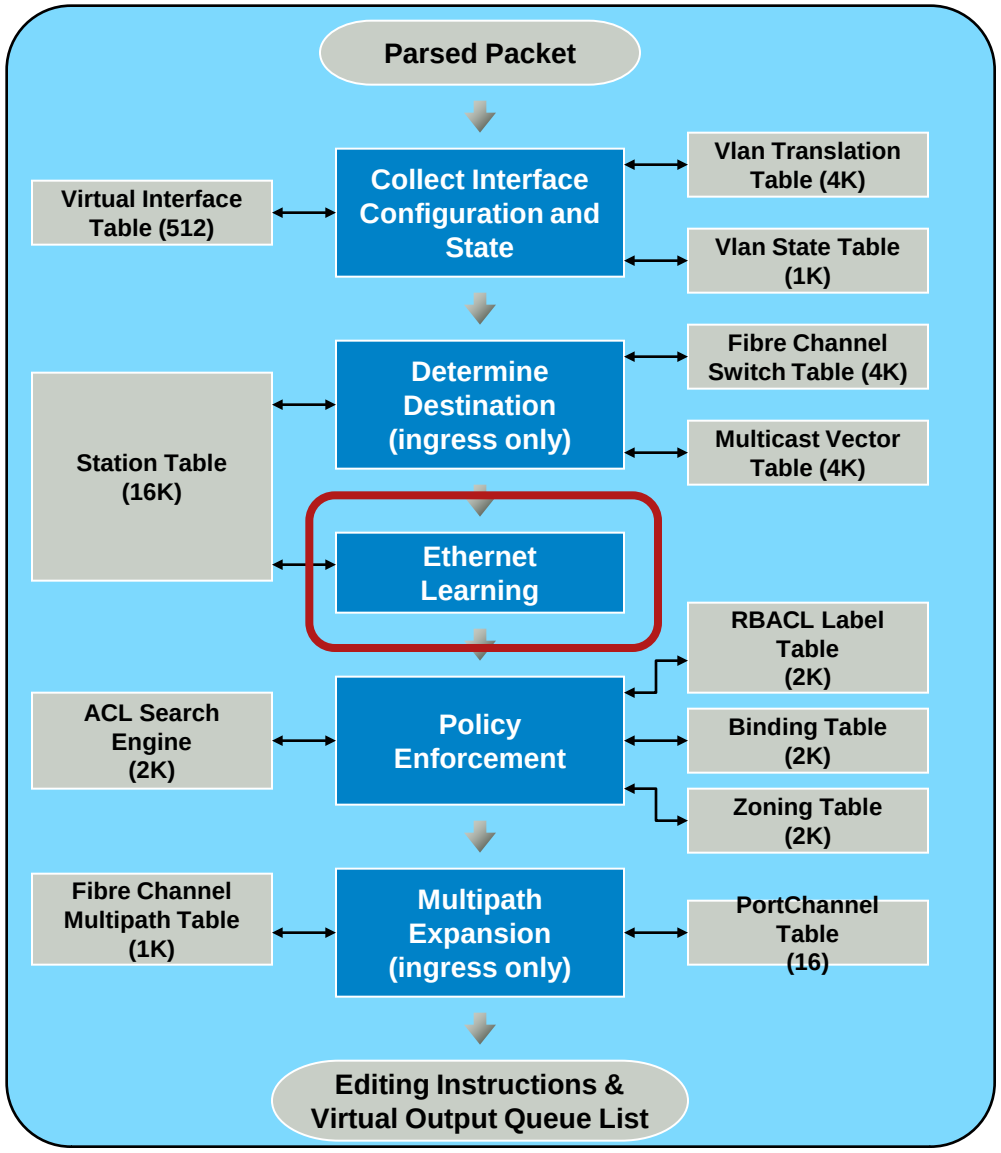
## UPC Ethernet and FC/FCoE Forwarding




- 16K Entry StationTable
  - Searched by {VLAN, destination address}
- Unknown addresses forwarded by VLAN multicast vectors
  - Unknown unicast
  - Unregistered multicast
  - Broadcast
- IP Multicast forwarded by MAC address
  - IP multicast groups registered by IGMP v1, v2, v3 snooping
  - Multicast vectors allocated dynamically based on destination membership
- Same mechanism forwards Fibre Channel in the local domain and N_Port Virtualizer

# Nexus 5000 Hardware Overview

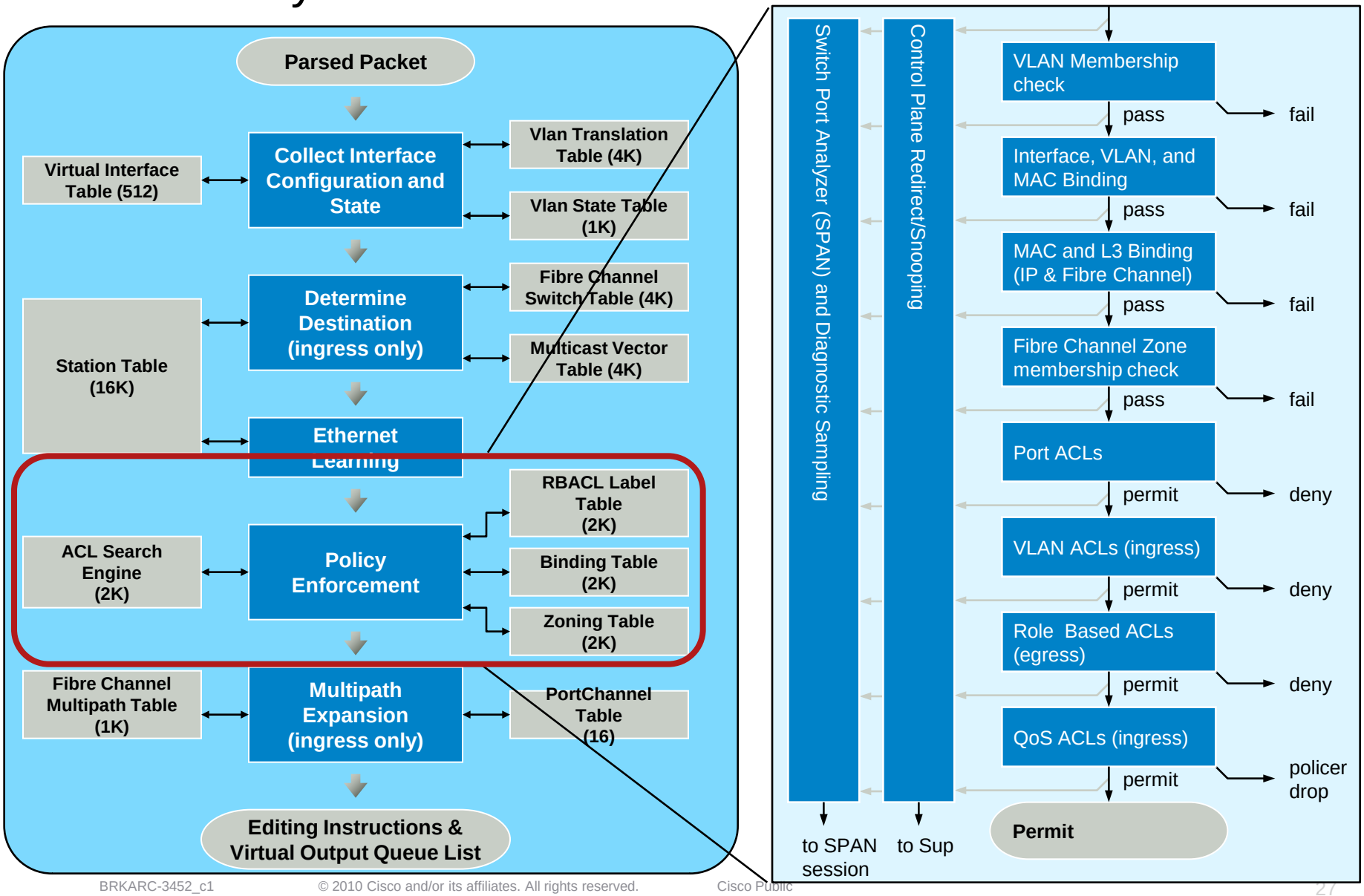
## UPC Ethernet Learning




- Ingress and Egress HW-based learning
  - Line rate on for all frames
  - Facilitates distributed table population
- Ingress UPC notifies supervisor to develop MAC database
- Supervisor pushes new addresses to all Unified Port Controllers (UPC)
  - Adds entries if missed
  - Re-enforces existing entries
- Supervisor queries tables to check for consistency
  - Maintains aging state
- CPU removes entries that are obsolete

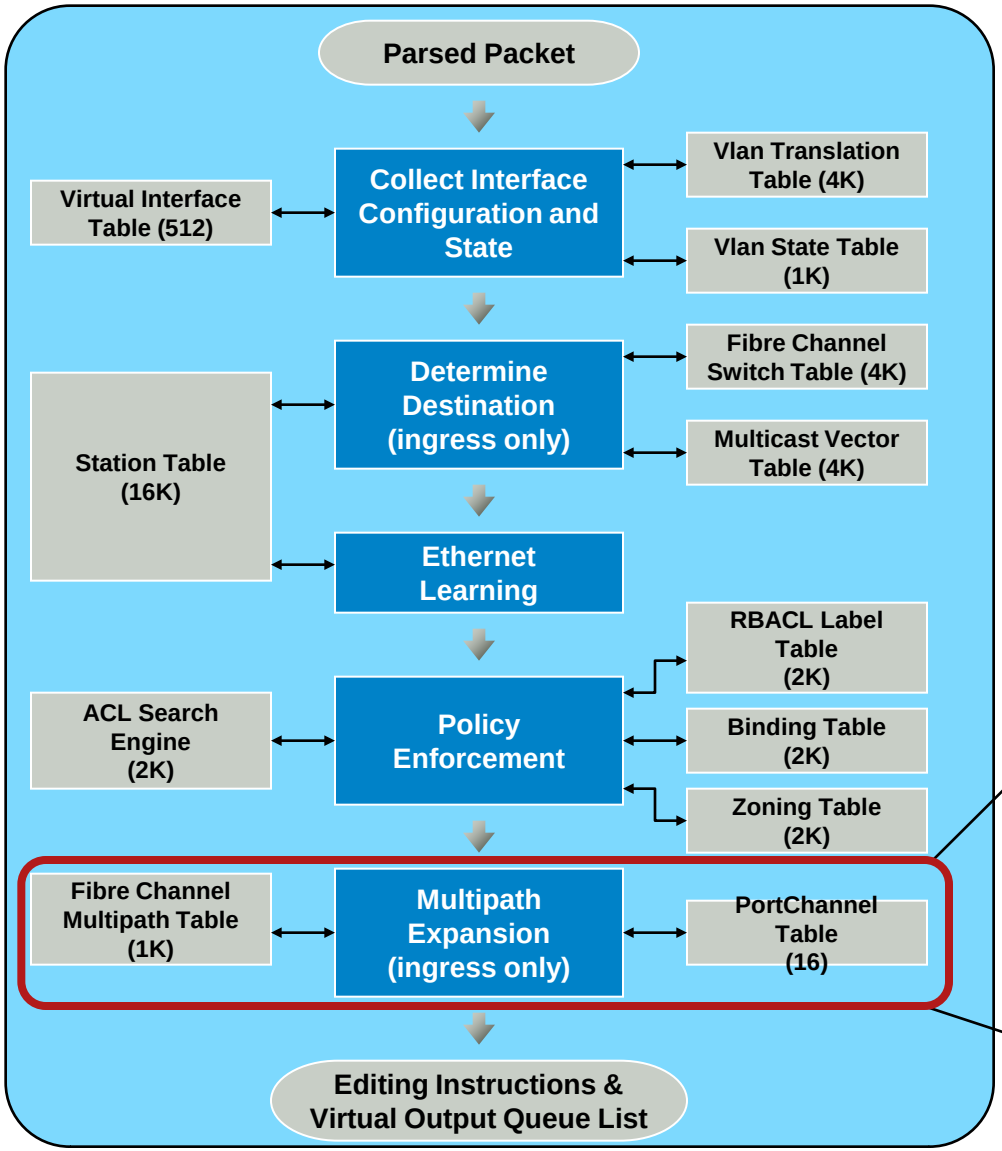
# Nexus 5000 Hardware Overview

## UPC Policy Enforcement

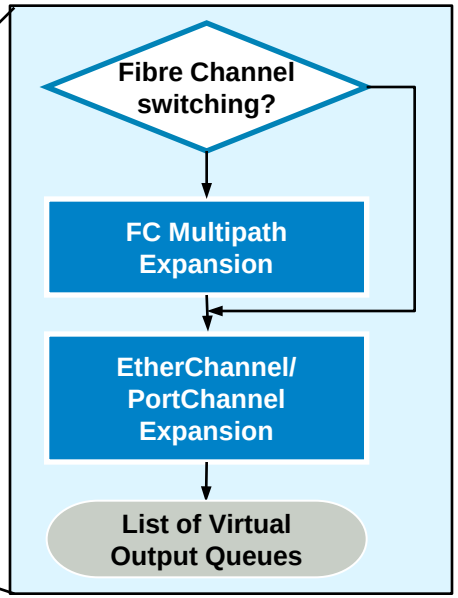


# Nexus 5000 Hardware Overview

## UPC Multipath Expansion

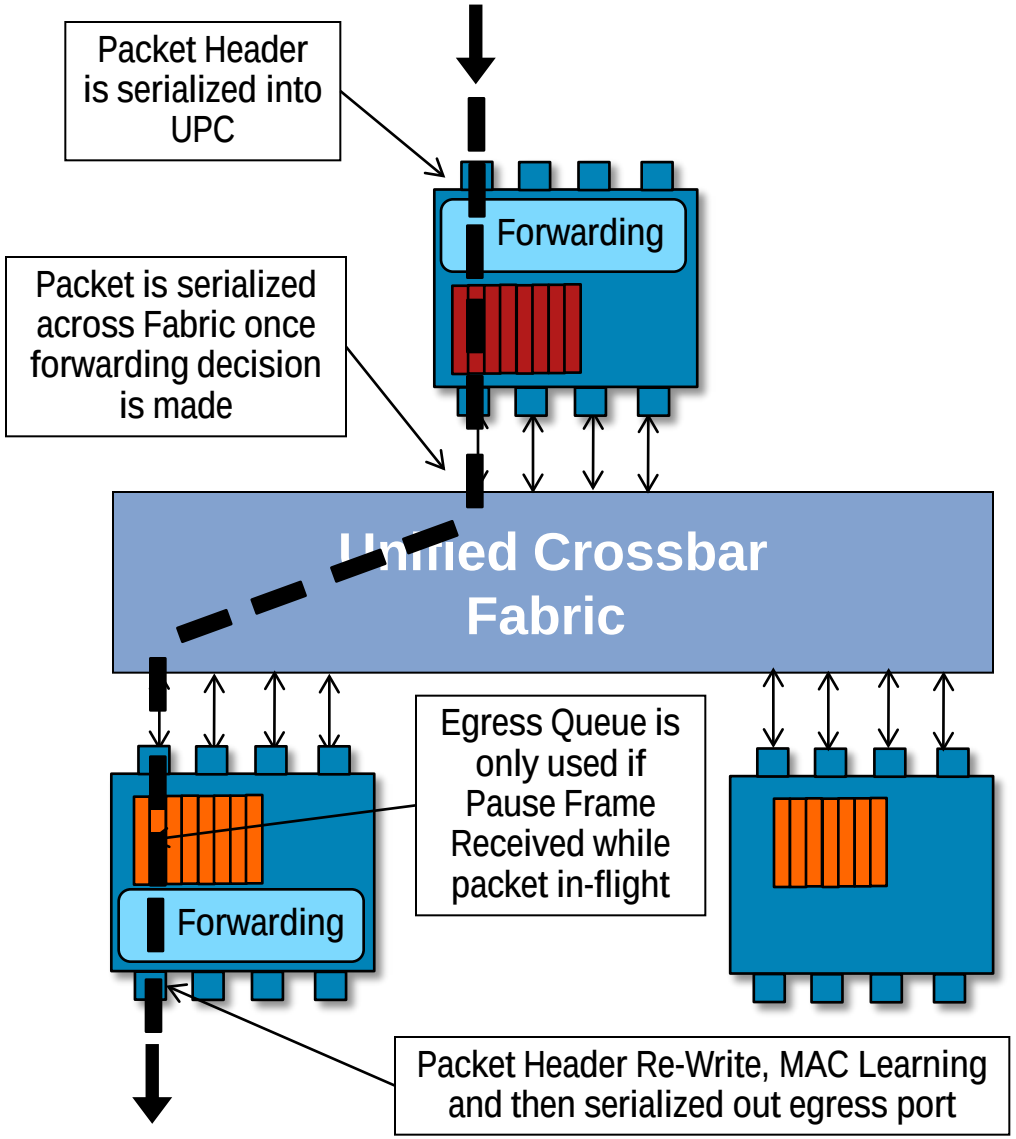



- Nexus 5000 utilizes a two stage multipath expansion
- Fibre Channel load shares via FSPF or NPV
- Secondary multipath hashing via Ethernet port channel or FC port channel
- 16 port channels of up to 16 ports each for either SAN or LAN



# Nexus 5000 Hardware Overview

## Packet Forwarding—Cut Thru Switching

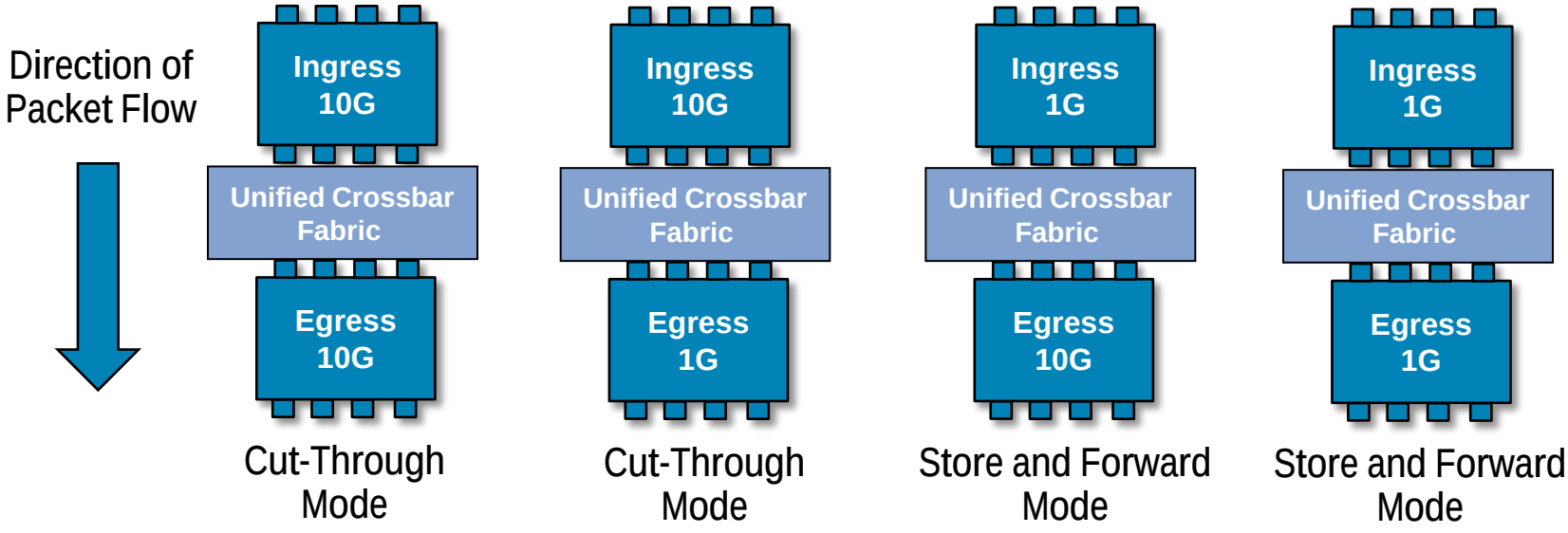



- Nexus 5000 utilizes a Cut Thru architecture when possible
- Bits are serialized in from the ingress port until enough of the packet header has been received to perform a forwarding and policy lookup
- Once a lookup decision has been made and the fabric has granted access to the egress port bits are forwarded through the fabric
- Egress port performs any header rewrite (e.g. CoS marking) and MAC begins serialization of bits out the egress port

# Nexus 5000 Hardware Overview

## Packet Forwarding—Cut-Through Switching

- Nexus 5000 utilizes both cut-through and store and forward switching
- Cut-through switching can only be performed when the ingress data rate is equivalent ***or*** faster than the egress data rate
- The X-bar fabric is designed to forward 10G packets in cut-through which requires that 1G to 1G switching also be performed in store and forward mode



# Nexus 5000 Hardware Overview

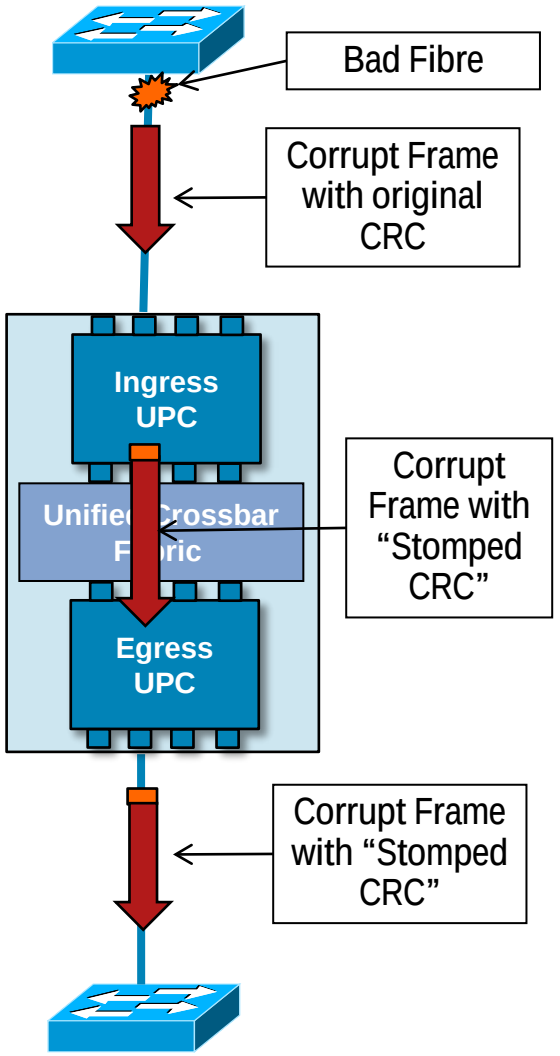
Forwarding Mode Behavior (Cut-Through or Store and Forward)

| Source Interface | Destination Interface | Switching Mode |
|---|---|---|
| 10 GigabitEthernet | 10 GigabitEthernet | Cut-Through |
| 10 GigabitEthernet | 1 GigabitEthernet | Cut-Through |
| 1 GigabitEthernet | 1 GigabitEthernet | Store-and-Forward |
| 1 GigabitEthernet | 10 GigabitEthernet | Store-and-Forward |
| FCoE | Fibre Channel | Cut-Through |
| Fibre Channel | FCoE | Store-and-Forward |
| Fibre Channel | Fibre Channel | Store-and-Forward |
| FCoE | FCoE | Cut-Through |

# Nexus 5000 Hardware Overview

## Packet Forwarding—Cut Through Switching

- In Cut-Through switching frames are not dropped due to bad CRC
- Nexus 5000 implements a CRC ‘stomp’ mechanism to identify frames that have been detected with a bad CRC upstream
- A packet with a bad CRC is “stomped”, by replacing the “bad” CRC with the original CRC exclusive-OR’d with the STOMP value ( a 1’s inverse operation on the CRC)
- In Cut Through switching frames with invalid MTU (frames with a larger MTU than allowed) are not dropped
- Frames with a “> MTU” length are truncated and have a stomped CRC included in the frame



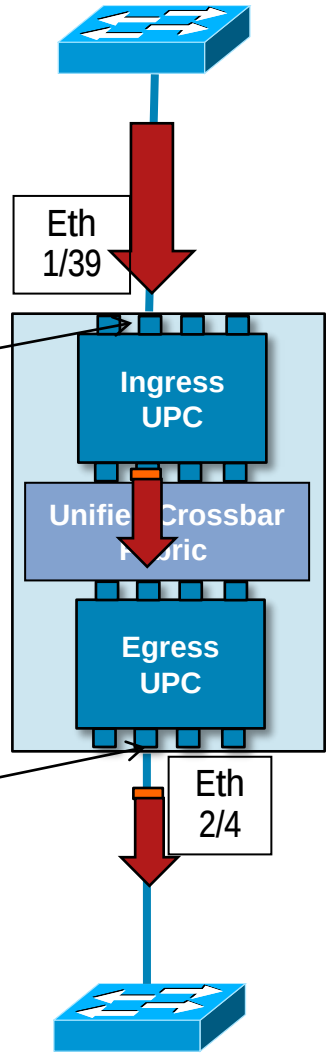
# Nexus 5000 Hardware Overview

## Packet Forwarding—Cut Through Switching

- Corrupt or Jumbo frames arriving inbound will count against the Rx Jumbo or CRC counters
- Corrupt or Jumbo frames will be identified via the Tx output error and Jumbo counters

```
dc11-5020-4# sh int eth 1/39
<snip>
 RX
    576 unicast packets  4813153 multicast packets  55273 broadcast packets
    4869002 input packets  313150983 bytes
    31 jumbo packets  0 storm suppression packets
    0 runts  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
```

```
dc11-5020-4# sh int eth 2/4
<snip>
 TX
    112 unicast packets  349327 multicast packets  56083 broadcast packets
    405553 output packets  53600658 bytes
    31 jumbo packets
    31 output errors  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble
    0 Tx pause
```



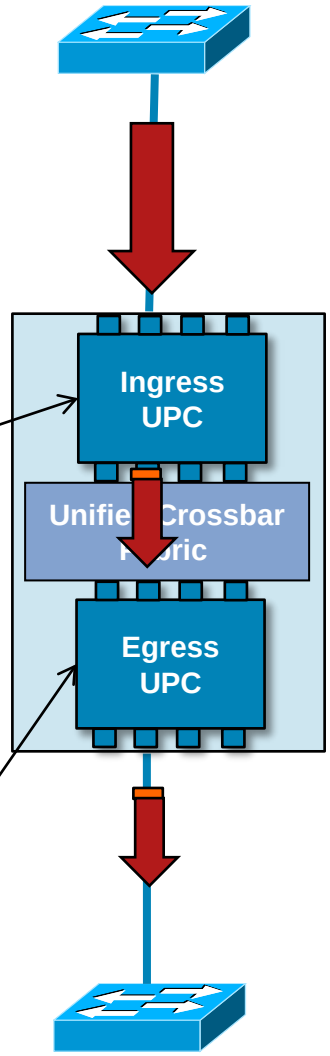
# Nexus 5000 Hardware Overview

## Packet Forwarding—Cut Thru Switching

- CRC and ‘stomped’ frames are tracked internally between ASIC’s within the switch as well as on the interface to determine internal HW errors are occurring

```
dc11-5020-4# sh hardware internal gatos asic 2 counters interrupt
<snip>
Gatos 2 interrupt statistics:
Interrupt name                                  |Count    |ThresRch|ThresCnt|Ivls
------------------------------------------------+--------+--------+--------+----
<snip>
gat_bm_port0_INT_err_ig_mtu_vio                 |1f       |0       |1f
<snip>
```

```
dc11-5020-4# sh hardware internal gatos asic 13 counters interrupt
<snip>
Gatos 13 interrupt statistics:
Interrupt name                                  |Count    |ThresRch|ThresCnt|Ivls
------------------------------------------------+--------+--------+--------+----
<snip>
gat_fw2_INT_eg_pkt_err_cb_bm_eof_err            |1f       |0       |1       |0
gat_fw2_INT_eg_pkt_err_eth_crc_stomp            |1f       |0       |1       |0
gat_fw2_INT_eg_pkt_err_ip_pyld_len_err          |1f       |0       |1       |0
gat_mm2_INT_rlp_tx_pkt_crc_err                  |1f       |0       |1       |0
<snip>
```




**NOTE:** At this point in your troubleshooting as you start to look at ASIC hardware counters you should be calling TAC ☺

# Nexus 5000 Hardware Overview

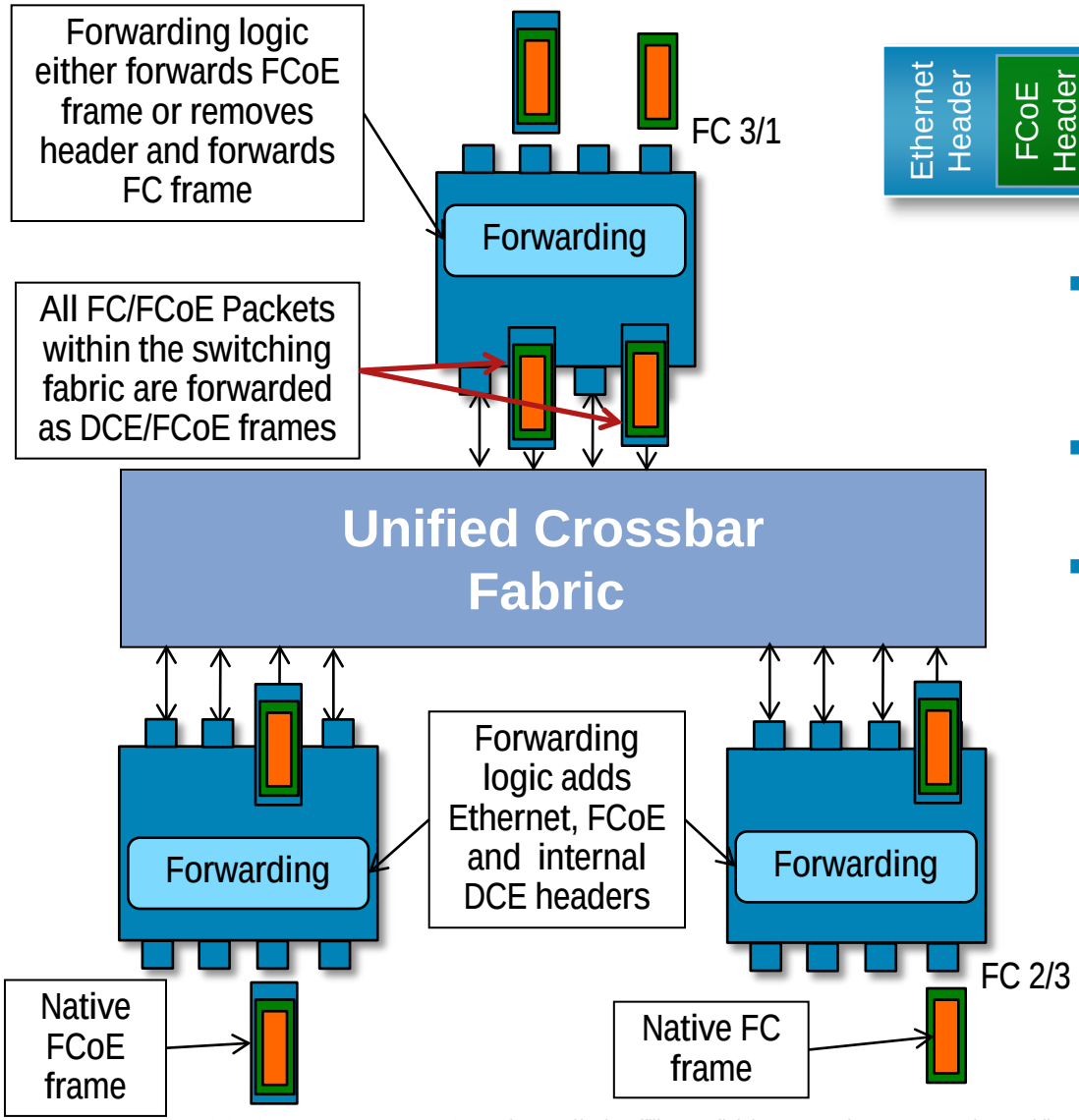
## CRC Behavior for Cut-Thru Frames

- The table below indicates the forwarding behavior for a corrupt packet (CRC error) arriving on a port operating in cut-through mode

| Source Interface Type | Destination Interface Type | Action |
|---|---|---|
| 10GE/DCE/FCoE | 10GE/DCE/FCoE | The CRC frame is transmitted as is |
| 10GE/DCE/FCoE | Native Fibre Channel | The FC CRC is stomped. Also the frame is transmitted with EOFa |
| Native Fibre Channel | Native Fibre Channel | The FC CRC is stomped. Also the frame is transmitted with EOFa |
| Native Fibre Channel | 10GE/DCE/FCoE | The FC CRC is stomped. Also the frame is transmitted with EOFa. Also the Ethernet CRC is stomped |

# Nexus 5000 Hardware Overview

## Packet Forwarding—Fibre Channel and FCoE

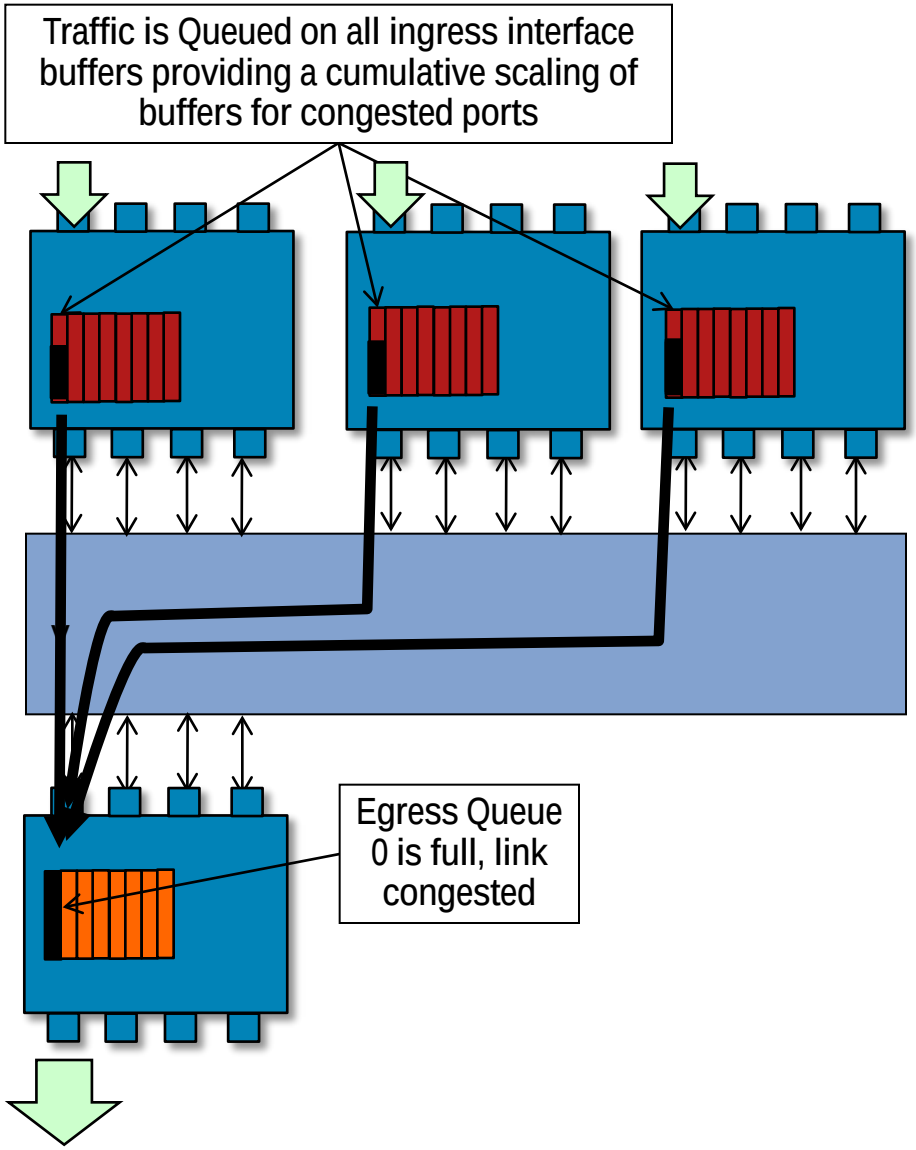




- Nexus 5000 operates as both an Ethernet switch and a Fibre Channel switch
- Supports native FC as well as FCoE interfaces
- Internally within the switching fabric all Fibre Channel frames are forwarded as DCE/FCoE frames
  - FC to FCoE
  - FC to FC
  - FCoE to FC
  - FCoE to FCoE

# Nexus 5000 Hardware Overview

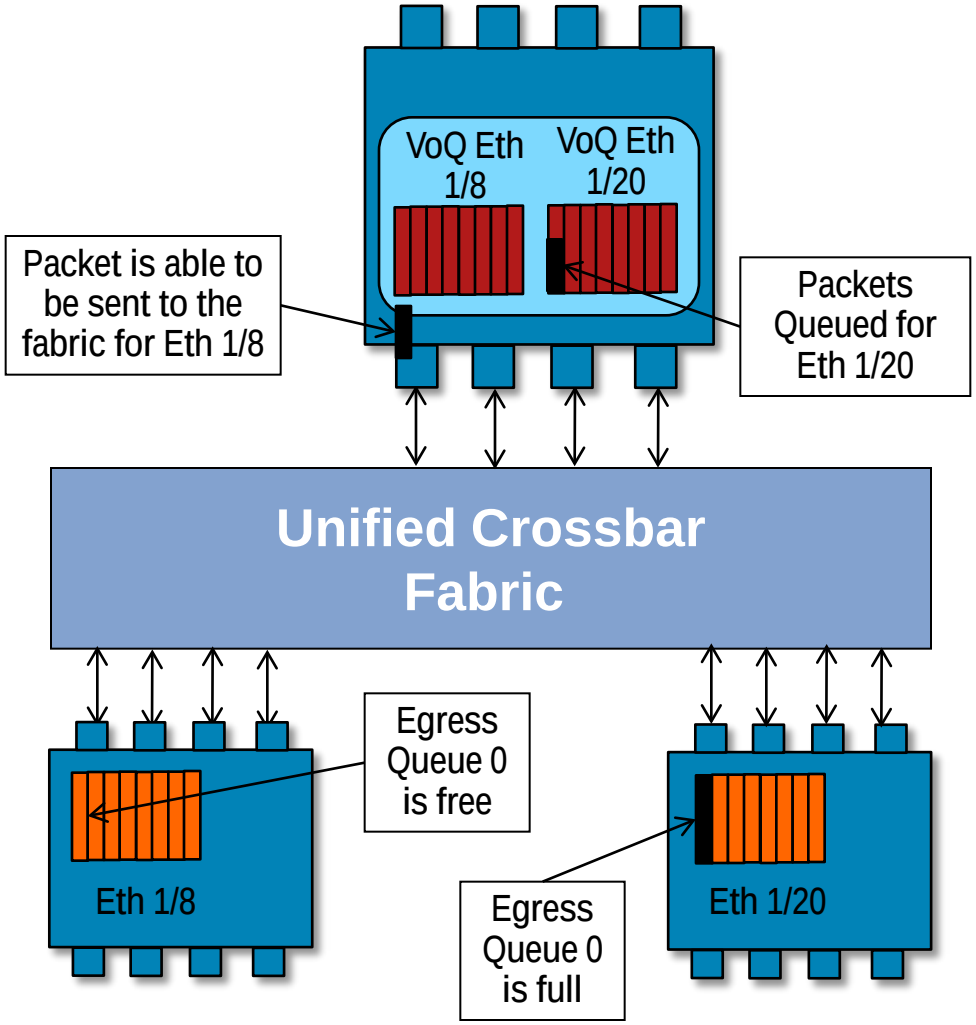
## Packet Forwarding—Ingress Queuing




- In typical Data Center access designs multiple ingress access ports transmit to a few uplink ports
- Nexus 5000 utilizes an ***Ingress*** Queuing architecture
- Packets are stored in ingress buffers until egress port is free to transmit
- Ingress queuing provides an additive effective
- *The total queue size available is equal to [number of ingress ports x queue depth per port]*
- Statistically ingress queuing provides the same advantages as shared buffer memory architectures

# Nexus 5000 Hardware Overview

## Packet Forwarding—Virtual Output Queues




- Nexus 5000 uses an 8 Queue QoS model for unicast traffic
- Traffic is Queued on the Ingress buffer until the egress port is free to transmit the packet
- To prevent Head of Line Blocking (HOLB) Nexus 5000 uses a Virtual Output Queue (VoQ) Model
- Each ingress port has a unique set of 8 virtual output queues for every egress port (8 x 58 = 464 unicast VoQ on every ingress port)
- If Queue 0 is congested for any port traffic in Queue 0 for all the other ports is still able to be transmitted
- Common shared buffer on ingress, VoQ are pointer lists and not physical buffers

# Nexus 5000 Hardware Overview

## UPC and 1G Ethernet

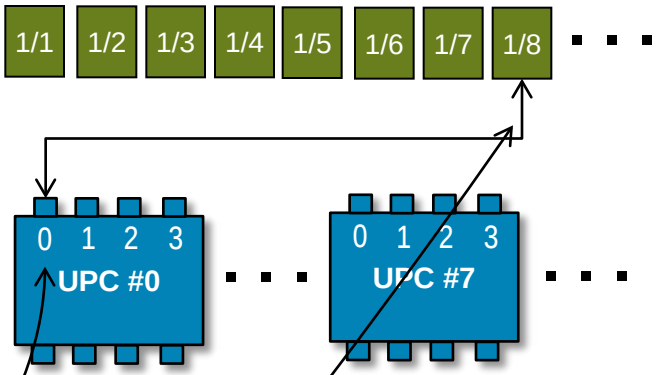
- Support for 1G speed on first 16 ports of Nexus 5020 and first eight ports of Nexus 5010
- Need to explicitly specify that the port runs at 1G speed
- Requires the use of a standard 1G SFP
  - GLC-T, GLC-SX-MM, GLC-LH-SM, SFP-GE-T, SFP-GE-S, SFP-GE-L (DOM capable SFP are supported)
- Supports for all features at 1G speed other than Unified I/O
  - No FCoE (no 1G Converged Network Adapters are shipping)
  - No Priority Flow Control (standard Pause is available)

```
interface Ethernet1/3
  switchport access vlan 800
  speed 1000
  channel-group 800
```

# Nexus 5000 Hardware Overview

## UPC and Port Mapping

- UPC interfaces are indirectly mapped to front panel ports
- Mapping of ports to UPC (Gatos) ASIC
  - The left column identifies the Ethernet interface identifier, xgb1/8 = e1/8
  - Column three and four reflect the UPC port that is associated with the physical Ethernet port



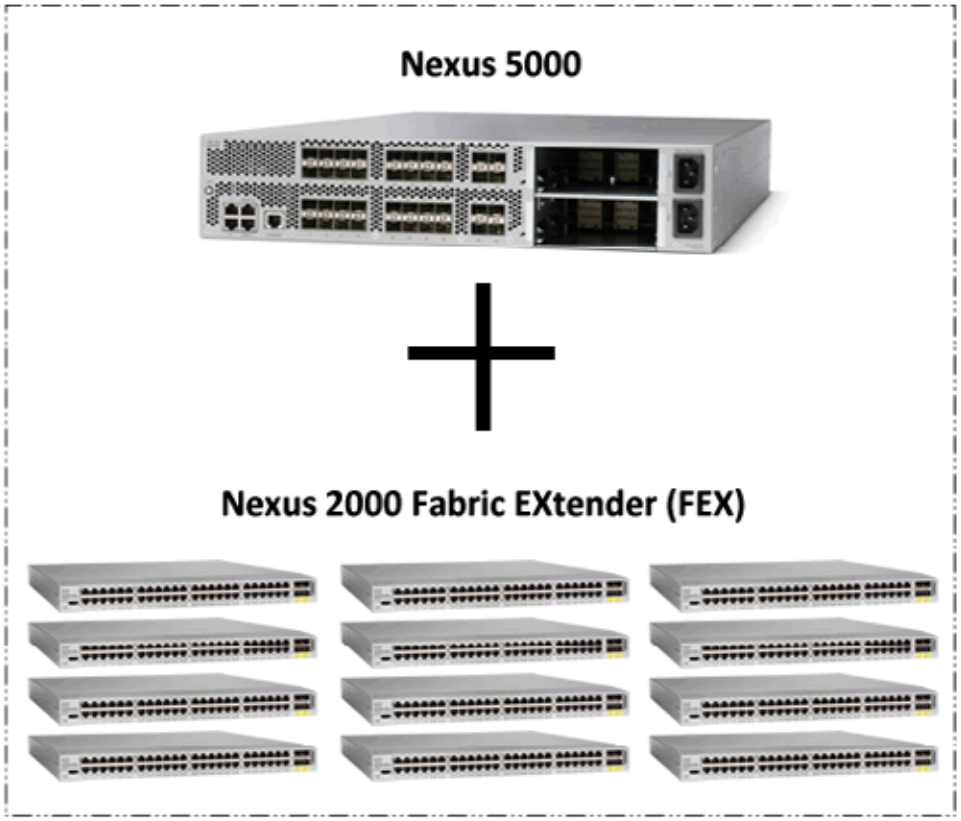

```
dc11-5020-3# show hardware internal gatos all-ports
<snip>

Gatos Port Info:
name   |log|gat|mac|flag|adm|opr|c:m:s:l|ipt|fab|xgat|xpt|if_index|diag
-------+---+---+---+----+---+---+-------+---+---+----+---+--------+----
1gb1/8  7  |0  |0  |b7  |en |up |0:0:0:0|0  |55 |0   |2  |1a007000|pass
1gb1/7 |6  |0  |1  |b7  |dis|dn |0:1:1:0|1  |54 |0   |0  |1a006000|pass
1gb1/3 |2  |0  |2  |b7  |en |up |1:2:2:0|2  |56 |0   |4  |1a002000|pass
xgb1/4 |3  |0  |3  |b7  |dis|dn |1:3:3:f|3  |57 |0   |6  |1a003000|pass
<snip>
xgb1/1 |0  |7  |2  |b7  |dis|dn |1:2:2:f|2  |6  |7   |4  |1a000000|pass
xgb1/2 |1  |7  |3  |b7  |dis|dn |1:3:3:f|3  |7  |7   |6  |1a001000|pass
```

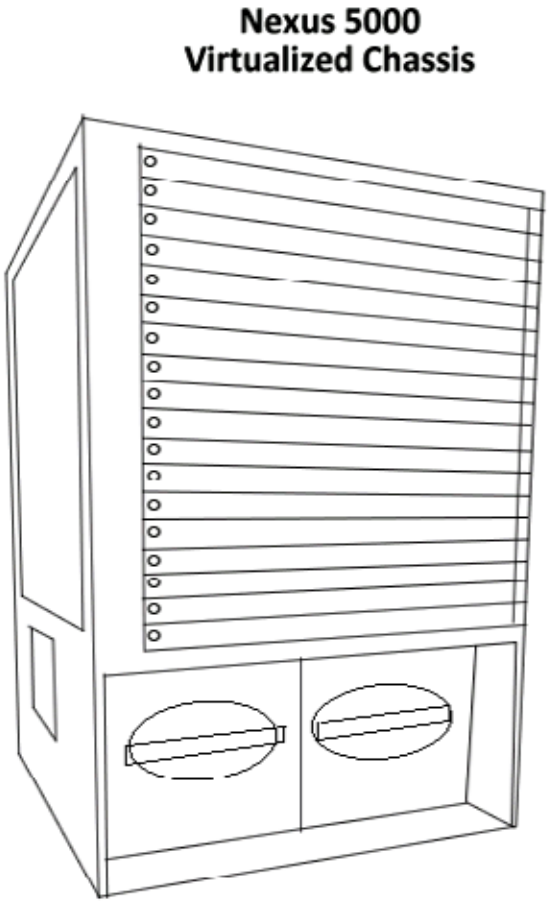
# Agenda

- Data Center Virtualized Access — Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

# Cisco Nexus 2000 Fabric Extender

## Virtualized Access Switch



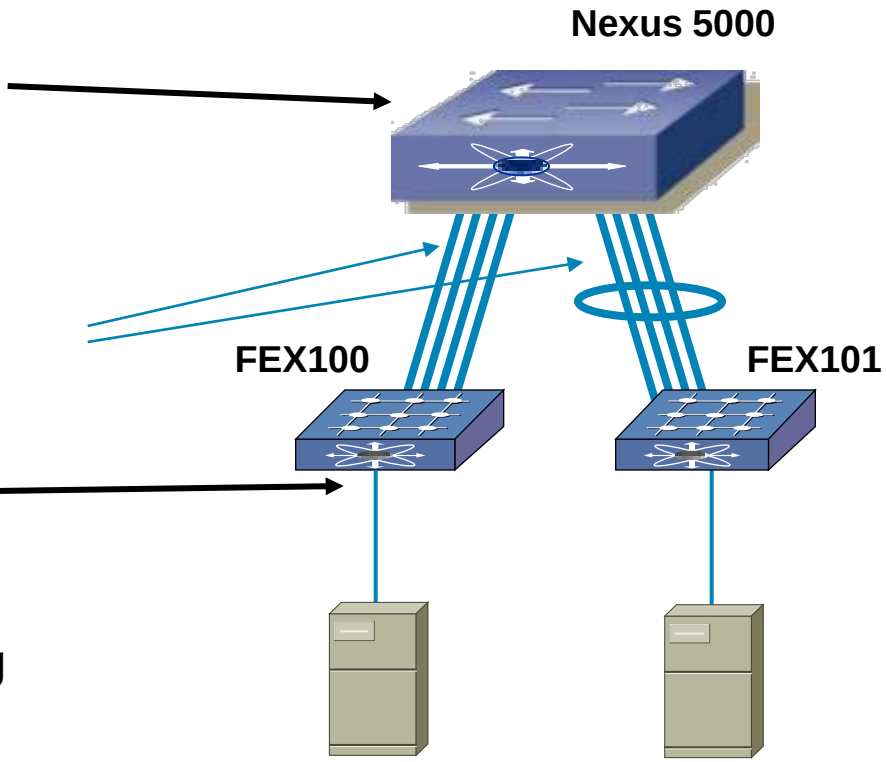




The Nexus 2000 Fabric Extender (FEX) acts as a remote line card (module) for the Nexus 5000, retaining all centralized management and configuration on the Nexus 5000, transforming it to a Virtualized Chassis

# Cisco Nexus 2000 Fabric Extender

## Fabric Extender Terminology

- **Parent Switch:** Acts as the combined Supervisor and Switching Fabric for the virtual switch
- **Fabric Links**: Extends the Switching Fabric to the remote line card (Connects Nexus 5000 to Fabric Extender)
- **Host Interfaces** (HIF)
- Fabric connectivity between Nexus 5000 and Nexus 2000 (FEX) can leverage either **pinning** or **port-channels**



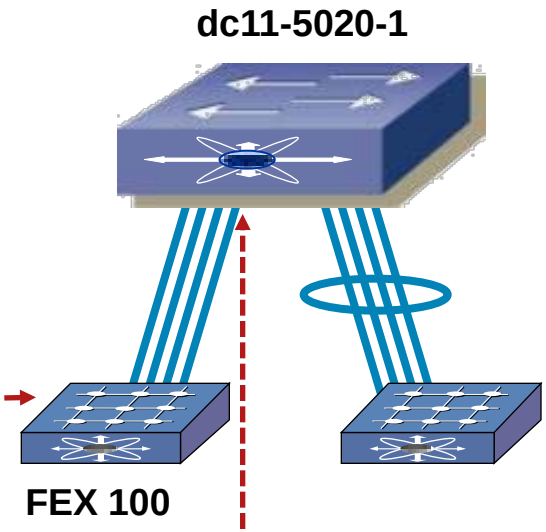

```
dc11-5020-1# show interface fex-fabric
     Fabric      Fabric       Fex                 FEX
Fex  Port      Port State    Uplink    Model         Serial
--------------------------------------------------------------
100   Eth1/17        Active     1    N2K-C2148T-1GE  JAF1311AFLL
100   Eth1/18        Active     2    N2K-C2148T-1GE  JAF1311AFLL
100   Eth1/19        Active     3    N2K-C2148T-1GE  JAF1311AFLL
100   Eth1/20        Active     4    N2K-C2148T-1GE  JAF1311AFLL
101   Eth1/21        Active     1    N2K-C2148T-1GE  JAF1311AFMT
101   Eth1/22        Active     2    N2K-C2148T-1GE  JAF1311AFMT
```

# Cisco Nexus 2000 Fabric Extender

## Configuring the Fabric

- Two step process
- Define the Fabric Extender (100–199) and the number of fabric uplinks to be used by that FEX (valid range: 1–4)

```
dc11-5020-1# switch# configure terminal
dc11-5020-1(config)# fex 100
dc11-5020-1(config-fex)# pinning max-links 4
```




- Configure Nexus 5000 ports as fabric ports and associate the desired FEX

```
dc11-5020-1# switch# switch# configure terminal
dc11-5020-1(config)# interface ethernet 1/1
dc11-5020-1(config-if)# switchport mode fex-fabric
dc11-5020-1(config-if)# fex associate 100
. . .
<repeat for all 4 interfaces used by this FEX>
```

# Cisco Nexus 2000 Fabric Extender

## Fabric Connectivity

Show the attached Fabric Extenders

```
dc11-5020-1# show fex
  FEX             FEX             FEX                 FEX
Number       Description         State      Model            Serial
------------------------------------------------------------------------
100        FEX0100        Online      N2K-C2148T-1GE   JAF1311AFLL
101        FEX0101        Online      N2K-C2148T-1GE   JAF1311AFMT
```

Show the status of fabric link 'port-channel 100'

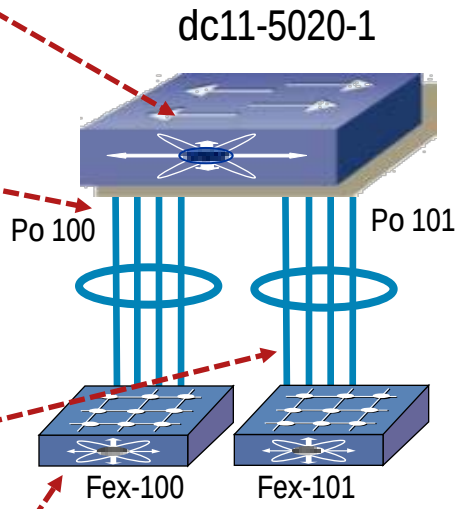
```
dc11-5020-1# show interface port-channel 100
port-channel100 is up
  Hardware: Port-Channel, address: 000d.eca4.5318 (bia 000d.eca4.5318)
<snip>
  Port mode is fex-fabric
```

Show the N2K interfaces carried over fabric link 'port-channel 101'

```
dc11-5020-1# sh int port-channel 101 fex-intf
Fabric           FEX
Interface        Interfaces
---------------------------------------------------
 Po101           Eth101/1/1    Eth101/1/2    Eth101/1/3    Eth101/1/4
                 Eth101/1/5    Eth101/1/6    Eth101/1/7    Eth101/1/8
<snip>
```

Show the interfaces themselves (N2K interfaces are N5K ports)

```
dc11-5020-1# sh int brief
<snip>
--------------------------------------------------------------------------------
Ethernet      VLAN    Type Mode   Status  Reason                   Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth100/1/1    1       eth  access up      none                       1000(D) --
Eth100/1/2    1       eth  access down    Link not connected         1000(D) --
Eth100/1/3    1       eth  access up      none                       1000(D) --
```

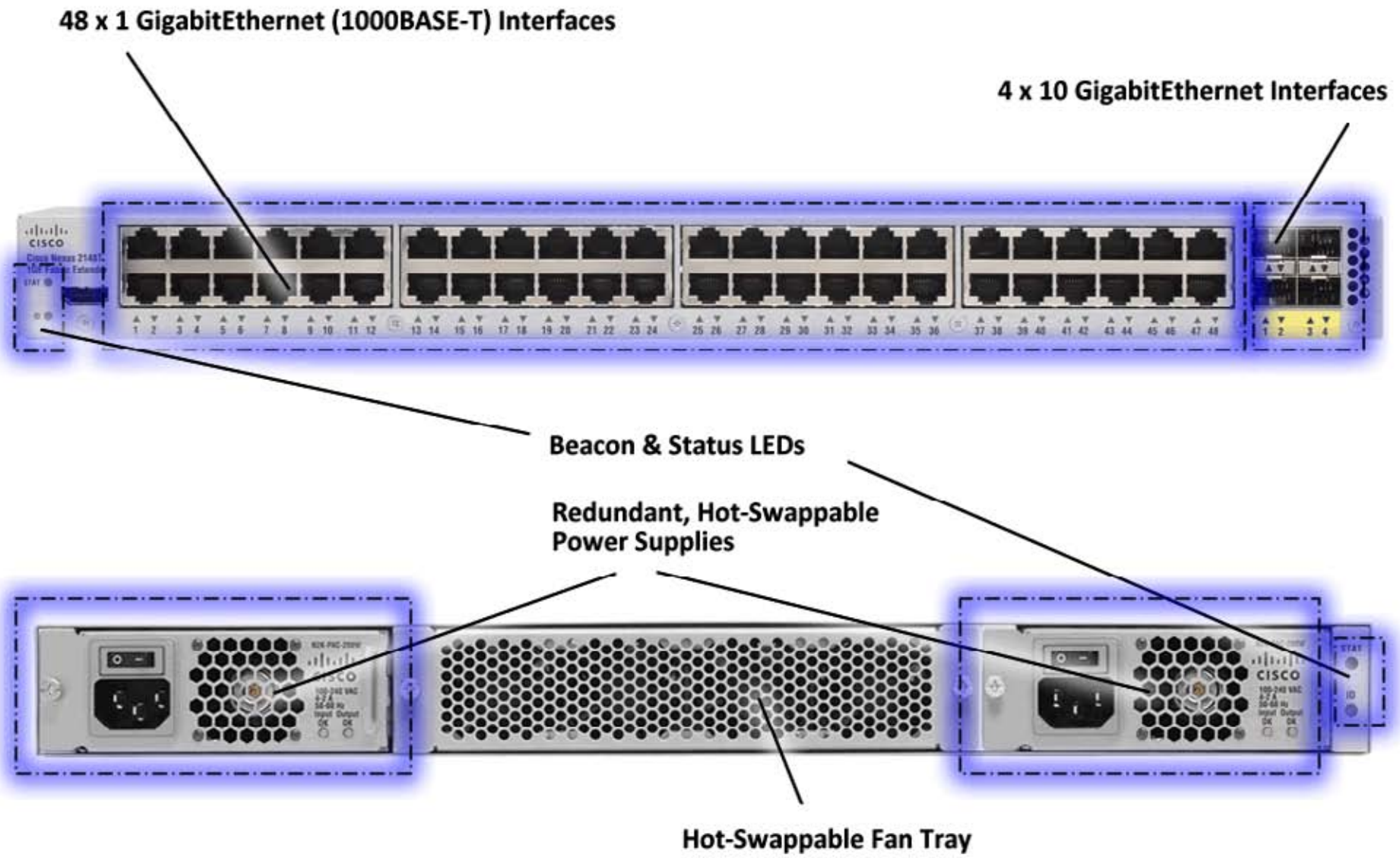


# Agenda

- Data Center Virtualized Access — Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

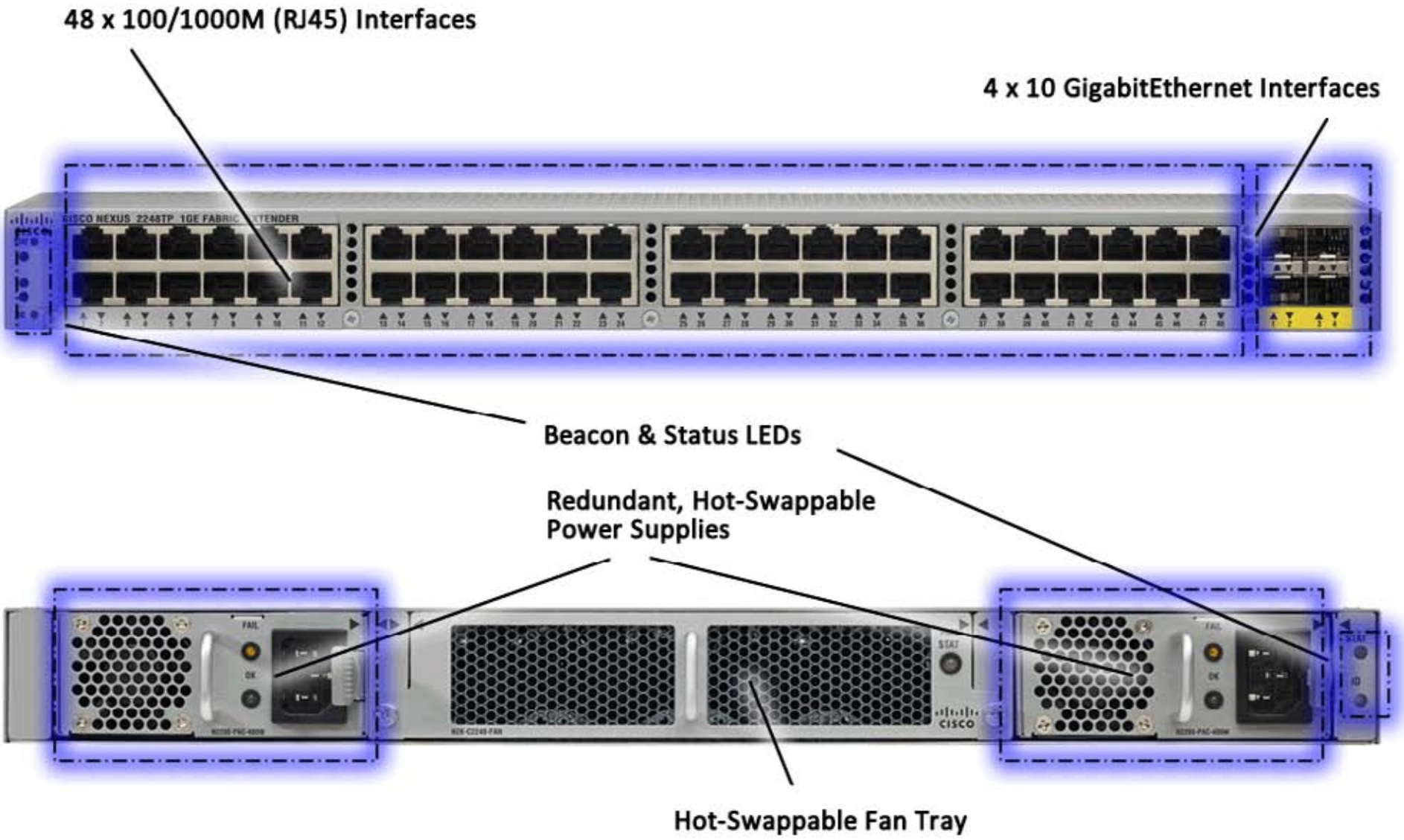
# Cisco Nexus 2148T Fabric Extender

## Overview



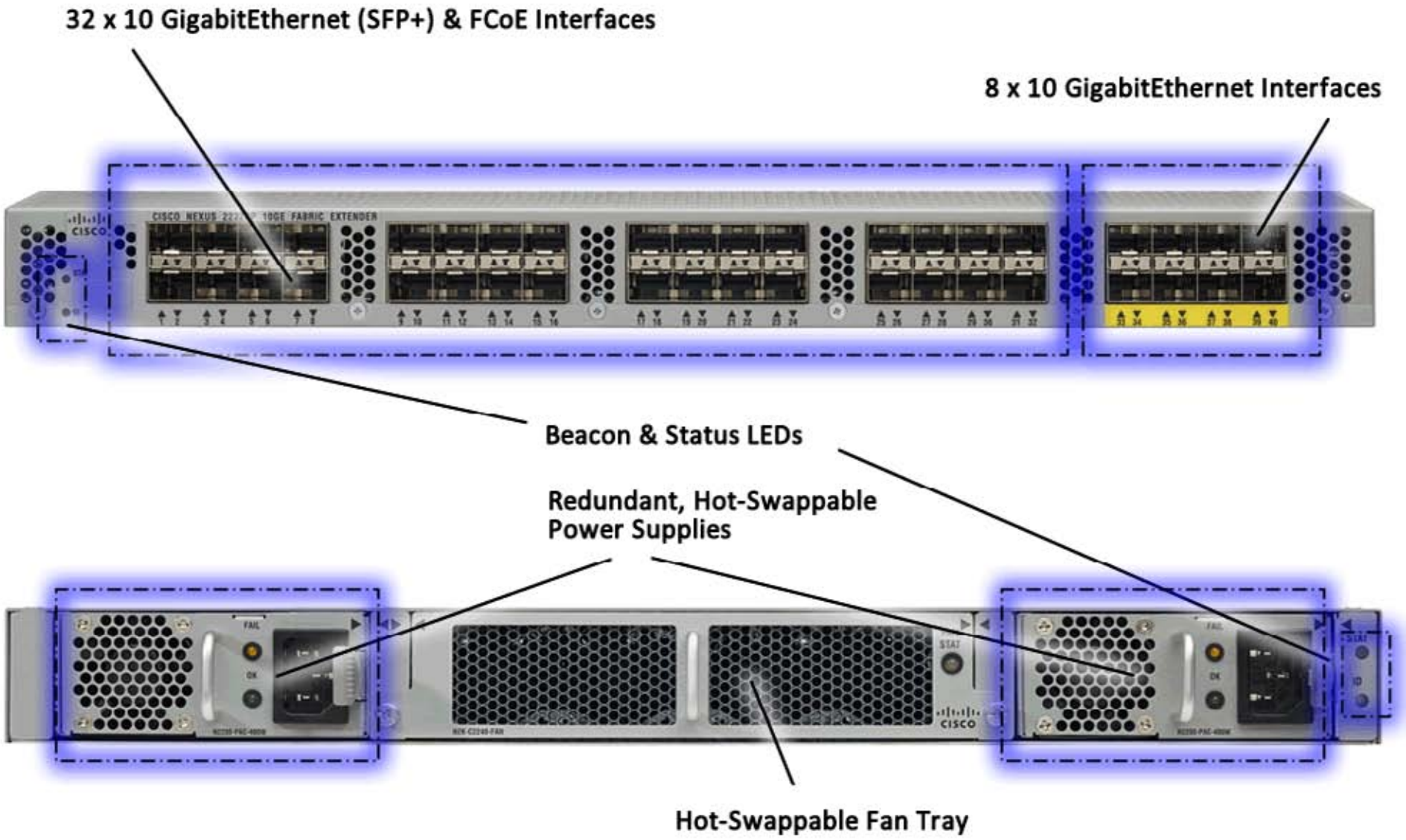
# Cisco Nexus 2248T Fabric Extender

## Overview



# Cisco Nexus 2232PP Fabric Extender

## Overview



# Nexus 2148T, 2232, 2248
## Capabilities

| Features | 2148 | 2248 | 2232 |
|---|---|---|---|
| Fabric Ports (NIF) | 4 | 4 | 8 |
| Fabric Link Port Speed (NIF) | 10Gbps | 10Gbps | 10Gbps |
| Port Channels on Fabric Links (NIF) | 1 x 4 ports maximum Hash L2/L3 fields | 1 x 4 ports maximum Hash L2/L3/L4 | 1 x 8 ports maximum Hash L2/L3/L4 |
| Host Ports (HIF) | 48 | 48 | 32 |
| Host Port Speeds (HIF) | 1 Gbps only | 100Mb/1Gbps | 1Gbps/10Gbps (No 1 Gbps at FCS, Q3CY10) |
| Local Port Channels on Host Ports (HIF) | Not Supported | Max 8 ports per port channel, Max of 24 port channels per 2248 Hash L2/L3/L4 | Max 8 ports per port channel, Max of 16 port channels per 2232 Hash L2/L3/L4 |
| FCoE/DCB Servers | No | No | Yes |
| Supported Parent Switches | Nexus 5010/5020 | Nexus 5010/5020 Nexus 7010/7018 * | Nexus 5010/5020 Nexus 7010/7018 * |

Requires N7K-M132XP-12 or N7K-M132XP-12L, Support for 2248 Target Is Q3CY'10, Support for 2232 Target Is Q4CY'10 (No DCB or FCoE Support at FCS for 2232 on Nexus 7000)

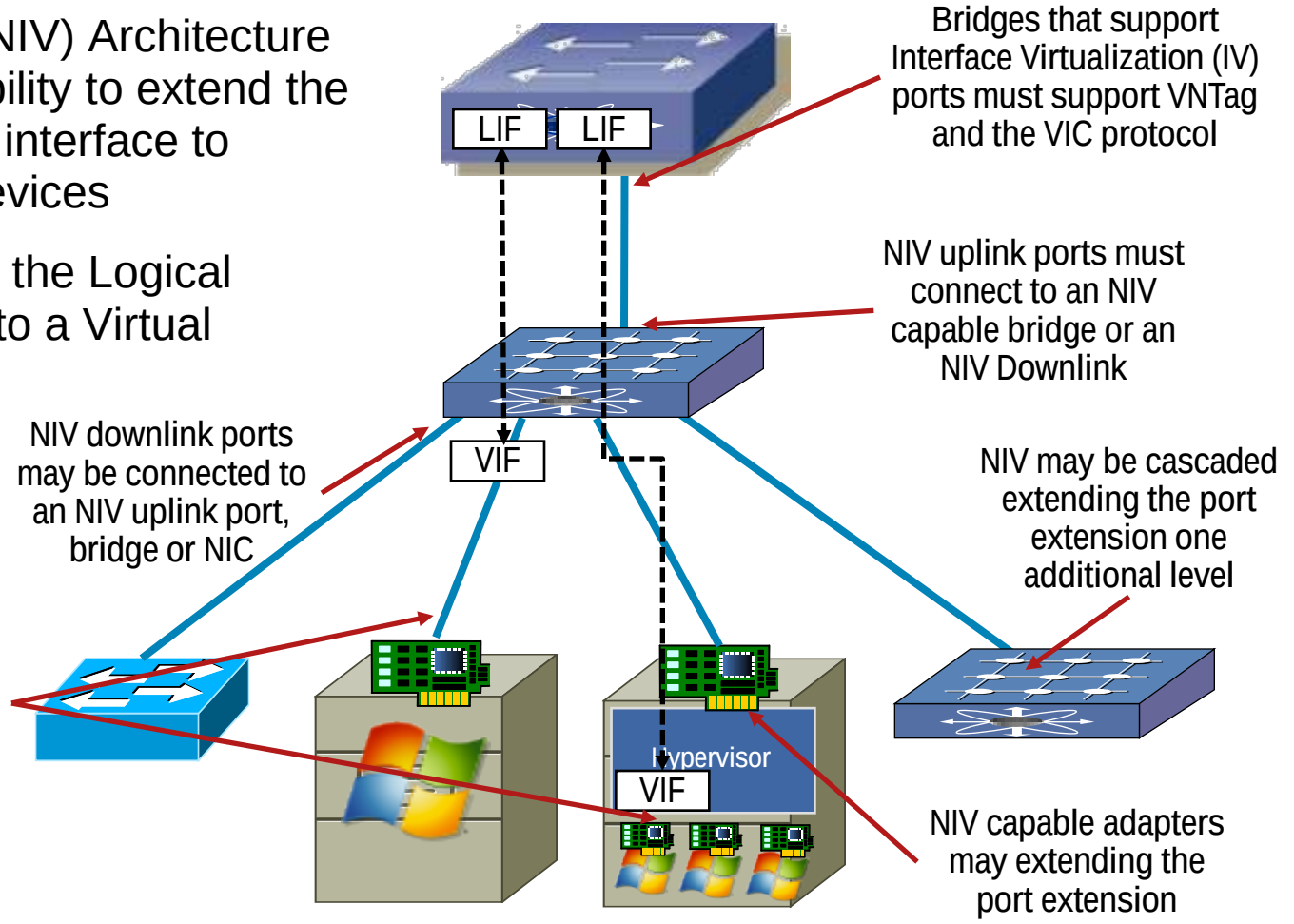
# Agenda

- Data Center Virtualized Access — Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

# Nexus 2000 Fabric Extender

## Network Interface Virtualization Architecture (NIV)

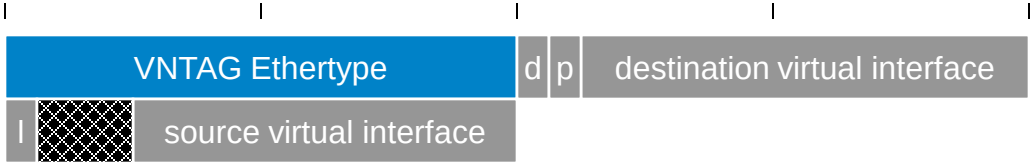
- The Network Interface Virtualization (NIV) Architecture provides the ability to extend the bridge (switch) interface to downstream devices
- NIV associates the Logical Interface (LIF) to a Virtual Interface (VIF)

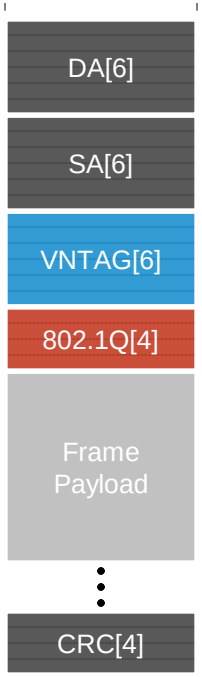



Note: Not All Designs Supported in the NIV Architecture Are Currently Implemented

# Nexus 2000 Fabric Extender

## VN-Tag



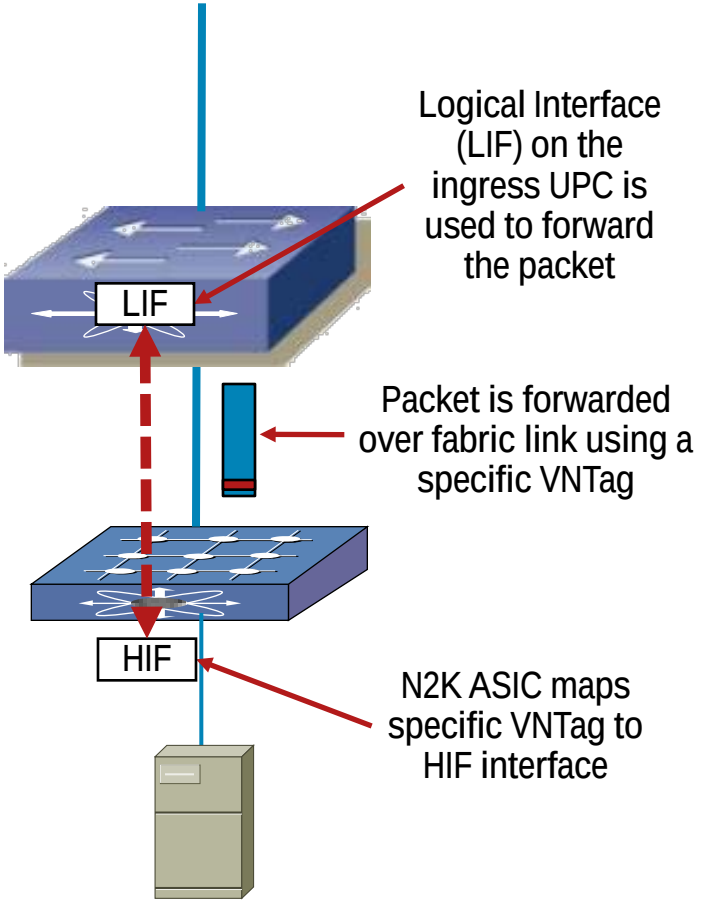




- direction indicates to/from adapter
- source virtual interface indicates frame source
  - looped indicates frame came back to source adapter
- destination virtual interface dictates forwarding
  - pointer helps pick specific destination vNIC or vNIC list
- Link local scope
  - Rooted at Virtual Interface Switch
  - 4096 virtual interfaces
  - 16,384 Virtual interface lists
- Coexists with VLAN (802.1Q) tag
  - 802.1Q tag is **mandatory** to signal data path priority

# Nexus 2000 Fabric Extender

## VN-Tag Port Extension

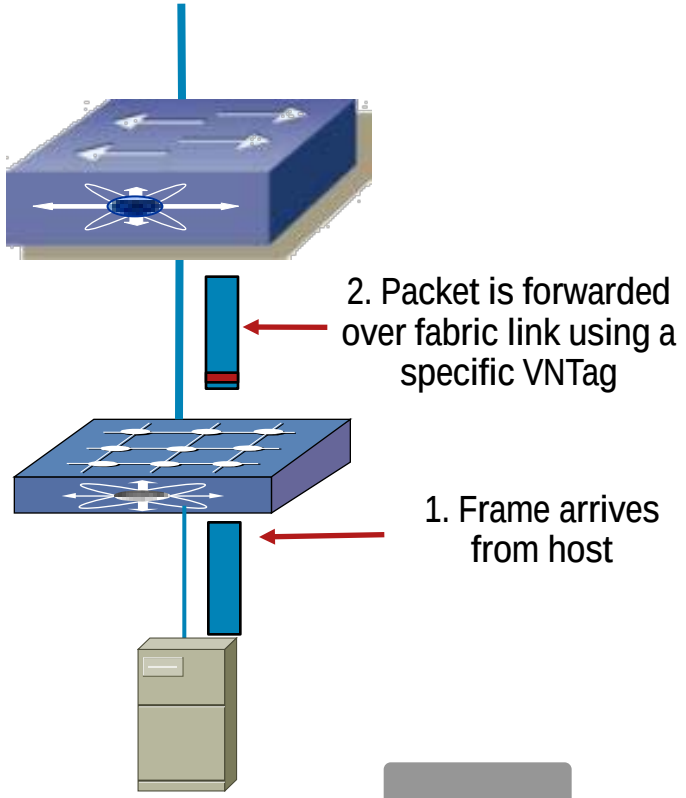
- Nexus 2000 Fabric Extender operates as a remote line card and does ***not*** support local switching
- All forwarding is performed on the Nexus 5000 UPC ASIC
- VNTag is a Network Interface Virtualization (NIV) technology that 'extends' the Nexus 5000 port down (Logical Interface = LIF) to the Nexus 2000 VIF referred to as a Host Interface (HIF)
  - VNTag is added to the packet between Fabric Extender and Nexus 5000
  - VNTag is stripped before the packet is sent to hosts
- VNTag allows the Fabric Extender to act as a data path of Nexus 5000 for all policy and forwarding

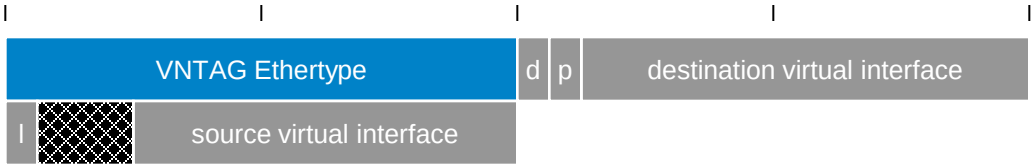


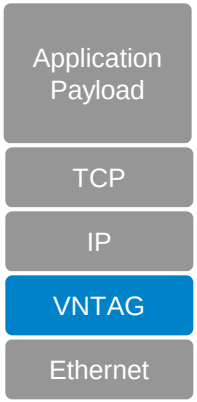
# Nexus 2000 Fabric Extender

## Host-to-Network Forwarding Part 1

- Nexus 2000 adds VNTAG
  - Unique VNTag for each Nexus 2000 Host Interface (HIF)
- VNTag field values
  - Direction bit is set to 0 indicating host to network forwarding
  - Source Virtual Interface is set based on the ingress HIF
  - p (pointer), l (looped), and destination virtual interface are undefined (0)
- Frame is unconditionally sent to the Nexus 5000



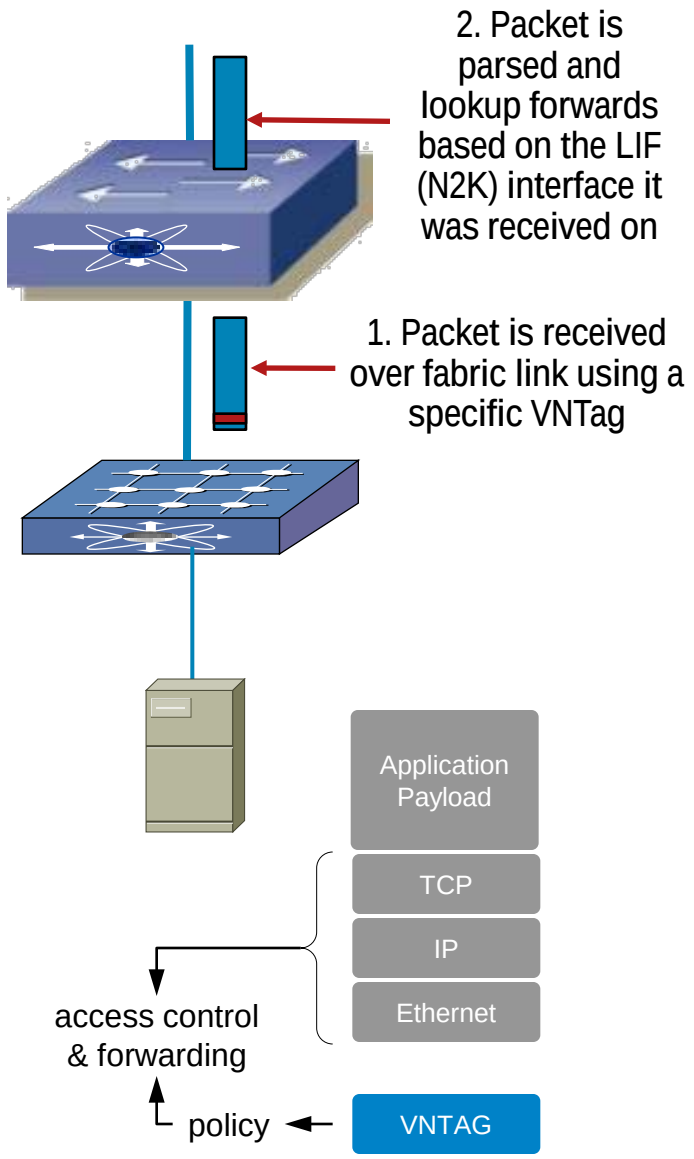



# Nexus 2000 Fabric Extender

## Host-to-Network Forwarding Part 2

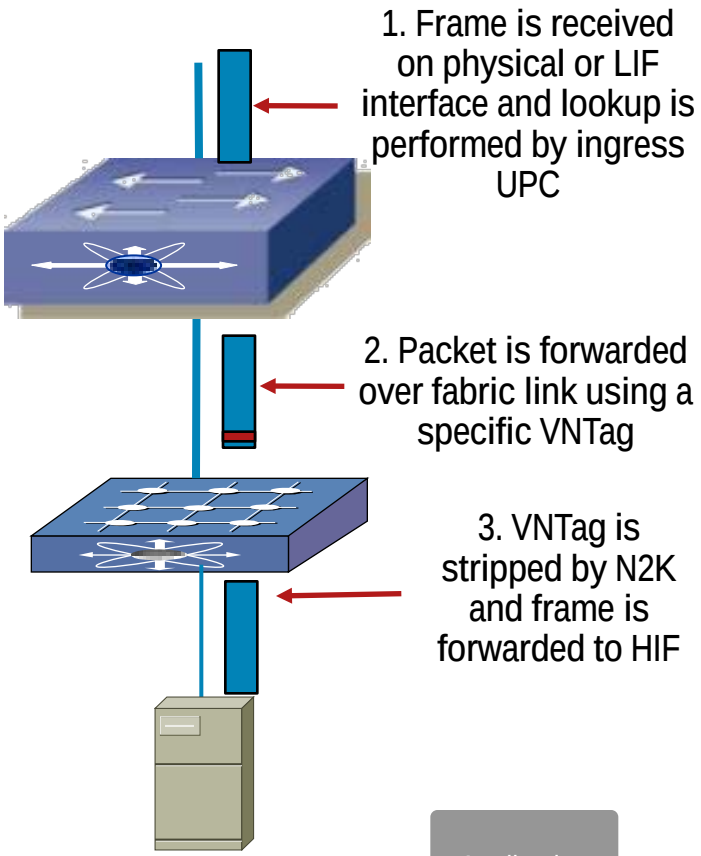
- Nexus 5000 ingress processing on fabric ports
- UPC extracts VNTAG which identifies the Logical Interface (LIF) corresponding to the physical HIF on the actual Nexus 2000
- Ingress policy based on physical Nexus 5000 port and LIF
  - Access control and forwarding based on frame fields and virtual interface (LIF) policy
  - Physical link level properties (e.g. MACSEC, …) are based on the Nexus 5000 port
- Forwarding selects destination port(s) and/or destination virtual interface(s)

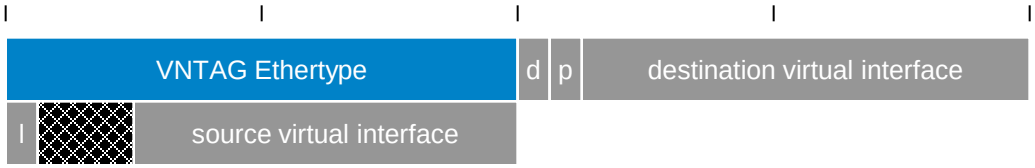


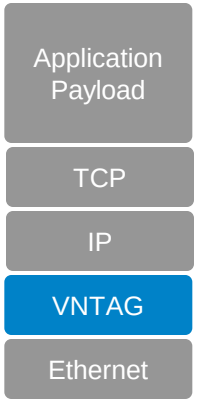
# Nexus 2000 Fabric Extender

## Network to Host Forwarding

- Nexus 5000 performs standard lookup and policy processing, when the egress port is determined to be an LIF (Nexus 2000) port
  - Insert VNTAG with direction is set to 1 (network to host)
  - Destination virtual interface is set to be the Nexus 2000 port VNTag
  - Source virtual interface is set if packet was sourced from an N2K port
  - L bit (looped) filter set if sending back to a source N2K
  - P bit is set if this is a multicast frame and requires egress replication

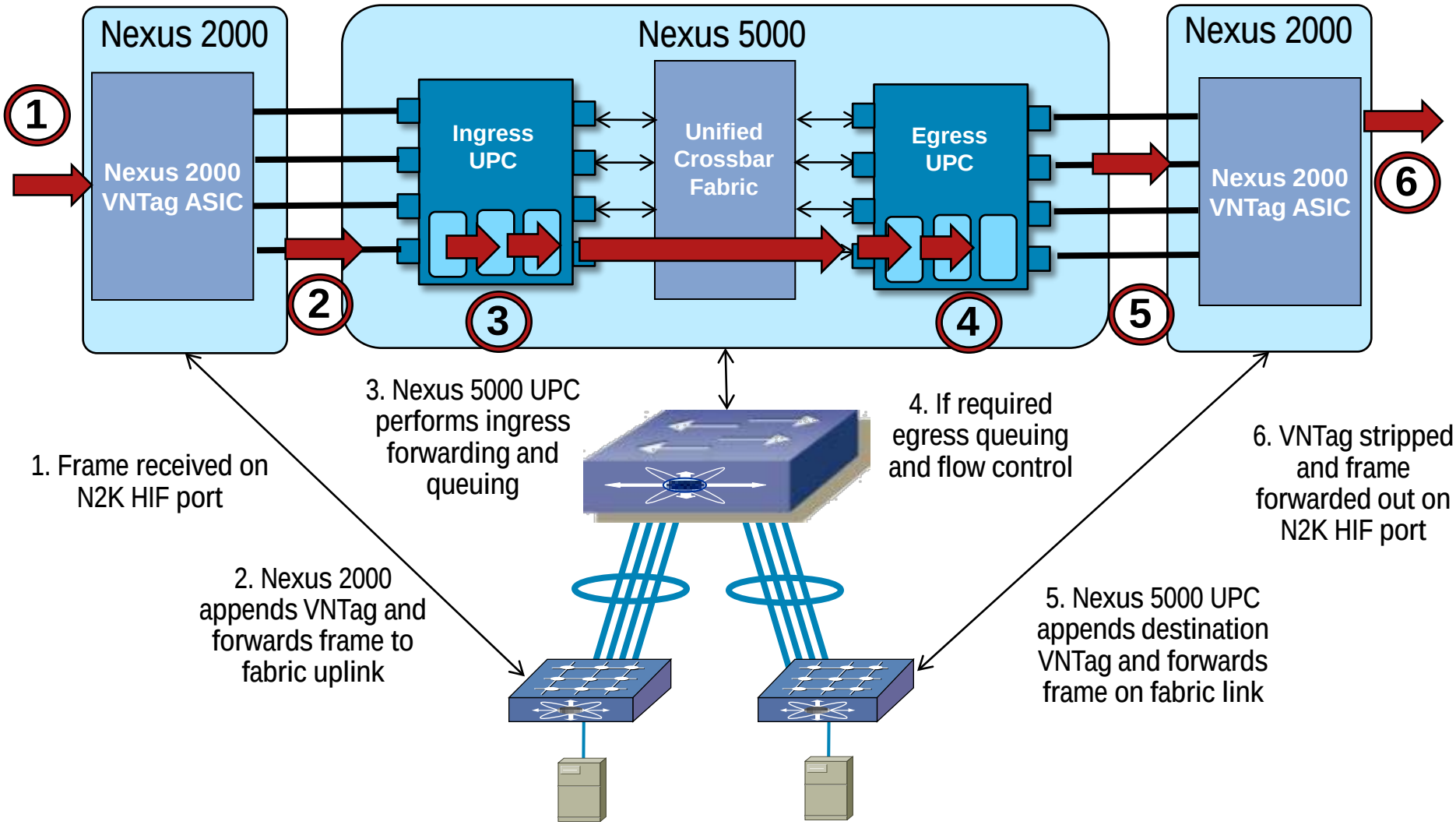


# Nexus 5000 and 2000

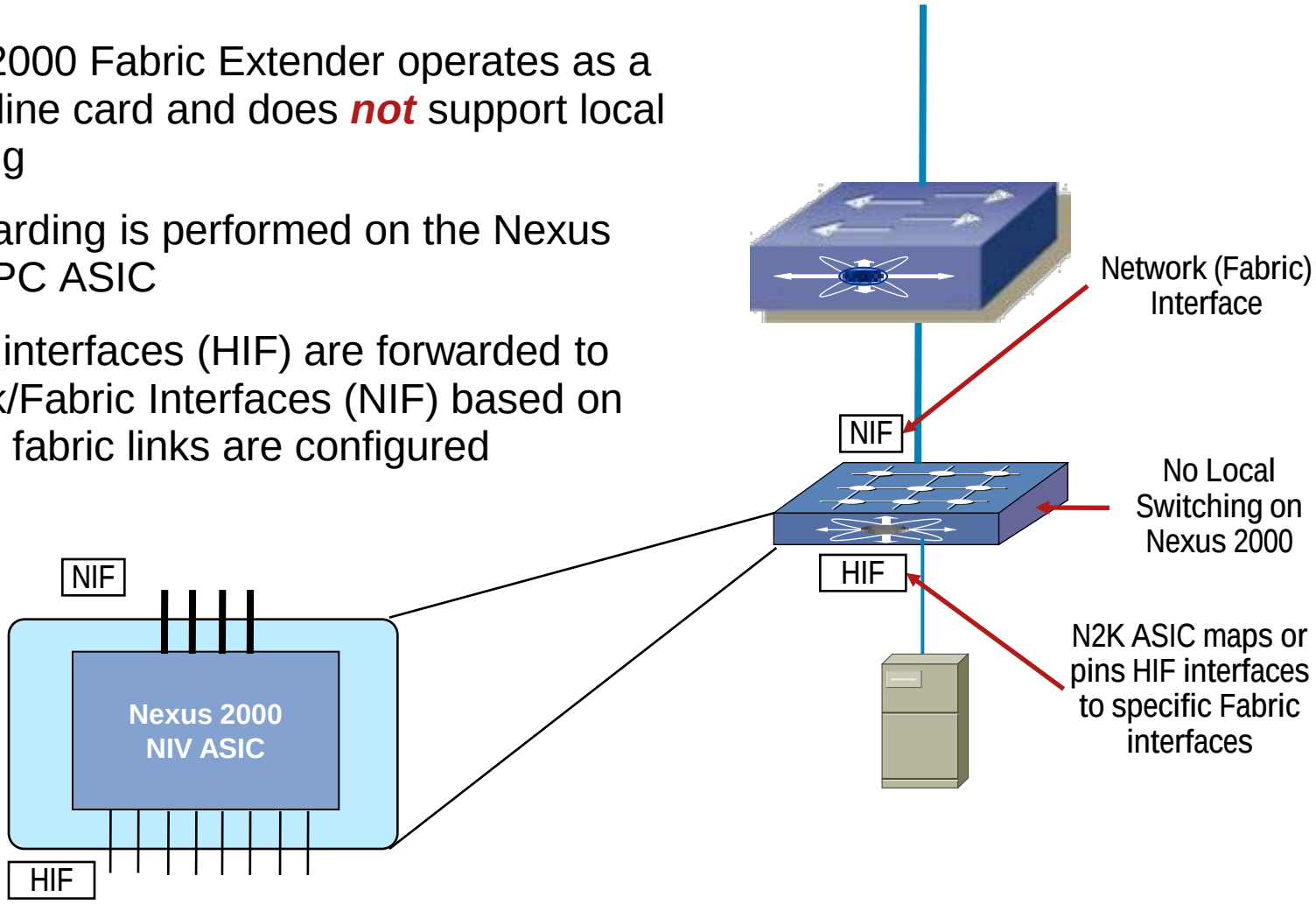
## Packet Forwarding Overview



# Nexus 2000 Fabric Extender

## Nexus 2000 Packet Forwarding

- Nexus 2000 Fabric Extender operates as a remote line card and does ***not*** support local switching
- All forwarding is performed on the Nexus 5000 UPC ASIC
- Ingress interfaces (HIF) are forwarded to Network/Fabric Interfaces (NIF) based on how the fabric links are configured



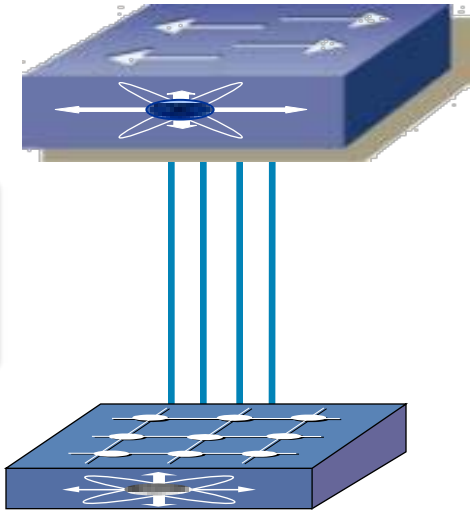
# Nexus 2000 Fabric Extender

## Fabric—Static Pinning

- Static Pinning associates (maps) specific server ports to specific fabric links
- Need to ensure that the ***same*** number of Ethernet ports are assigned as fex-fabric ports as defined in the ‘max-links’ parameter for that Fabric Extender

```
interface Ethernet1/1
  switchport mode fex-fabric
  fex associate 100

interface Ethernet1/2
  switchport mode fex-fabric
  fex associate 100

interface Ethernet1/3
  switchport mode fex-fabric
  fex associate 100

interface Ethernet1/4
  switchport mode fex-fabric
  fex associate 100

!
fex 100
  pinning max-links 4
  description Rack_100
```

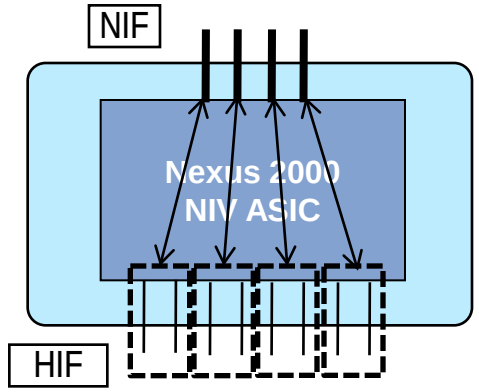
Ports Are Configured as Fabric and Associated with a Specific Fabric Extender



# Nexus 2000 Fabric Extender

## Fabric—Static Pinning

- Packets within the Nexus 2000 are ‘pinned’ or mapped from a specific ingress interface (HIF) to a specific fabric interface (NIF)
- When configured in ‘static pinning’ mode specific HIF are statically mapped to specific NIF
- Changing the number if fabric links requires the ASIC ‘pinning’ to be changed and is ***disruptive*** to traffic flows




```
dc11-5020-2# sh fex 150 detail
  FEX: 150 Description: FEX0150   state: Online
<snip>
  pinning-mode: static    Max-links: 2
  Fabric port for control traffic: Eth1/29
  Fabric interface state:
    Eth1/29 - Interface Up. State: Active
    Eth1/30 - Interface Up. State: Active
Fex Port        State  Fabric Port  Primary Fabric
       Eth150/1/1  Down     Eth1/29     Eth1/29
       Eth150/1/2  Down     Eth1/29     Eth1/29
<snip>
      Eth150/1/25  Down     Eth1/30     Eth1/30
      Eth150/1/26  Down     Eth1/30     Eth1/30
<snip>
```

Fabric Ports

Fabric Pinning

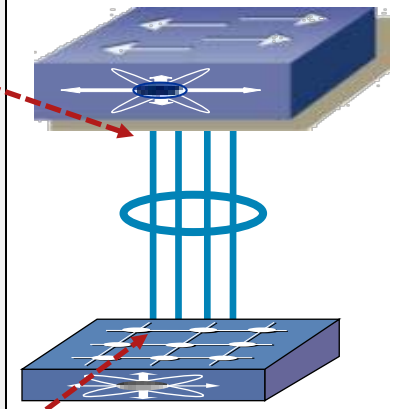
# Nexus 2000 Fabric Extender

## Fabric—Port Channel Configuration

```
interface port-channel1
  switchport mode fex-fabric
  description Fabric Extender 100
  fex associate 100

interface Ethernet1/1
  switchport mode fex-fabric
  description Fabric Extender 100 Etherchannel Link
  channel-group 1
  fex associate 100

interface Ethernet1/2
  switchport mode fex-fabric
  description Fabric Extender 100 Etherchannel Link
  channel-group 1
  fex associate 100

interface Ethernet1/3
  switchport mode fex-fabric
  description Fabric Extender 100 Etherchannel Link
  channel-group 1
  fex associate 100

interface Ethernet1/4
  switchport mode fex-fabric
  description Fabric Extender 100 Etherchannel Link
  channel-group 1
  fex associate 100

fex 100
  pinning max-links 1
  description Fabric Extender 100 – Using Etherchannel 1
```

Configure the Physical Ports as Members of the Fabric EtherChannel

Configure the Port Channel and Its Members to be Associated with a Specific Fabric Extender

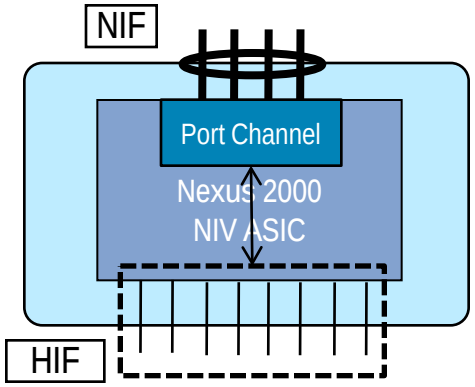


All server ports ‘pinned’ to a single logical fabric link

# Nexus 2000 Fabric Extender

## Fabric Port Channel Configuration

- In the fabric port channel configuration the internal forwarding within the Nexus 2000 ASIC is still ‘pinned’
- All HIF interfaces are pinned to an internal port channel NIF interface rather than to specific physical NIF interfaces
- Changing the number if fabric links does not require a changing in the internal forwarding mapping within the Nexus 2000 ASIC and is this ***‘non-disruptive’***




```
dc11-5020-1# sh fex 101 detail
  FEX: 101 Description: vPC-FEX   state: Online
<snip>
  pinning-mode: static    Max-links: 1
  Fabric port for control traffic: Eth1/21
  Fabric interface state:
    Po101 - Interface Up. State: Active
    Eth1/21 - Interface Up. State: Active
    Eth1/22 - Interface Up. State: Active
  Fex Port        State  Fabric Port  Primary Fabric
       Eth101/1/1    Up       Po101       Po101
       Eth101/1/2    Up       Po101       Po101
       Eth101/1/3  Down       Po101       Po101
<snip>
```

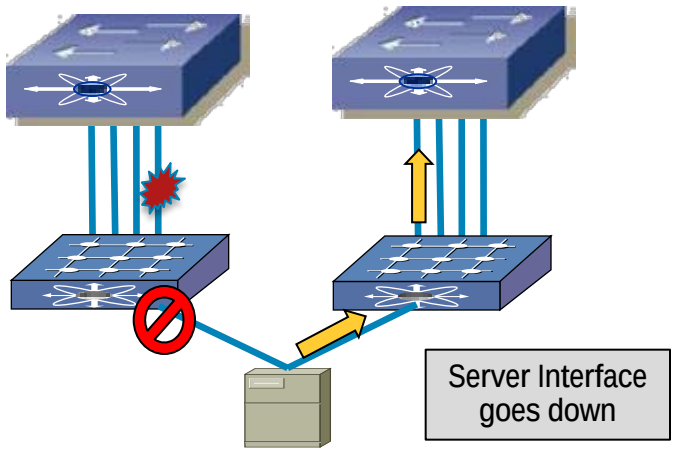
Fabric Ports

Fabric Pinning

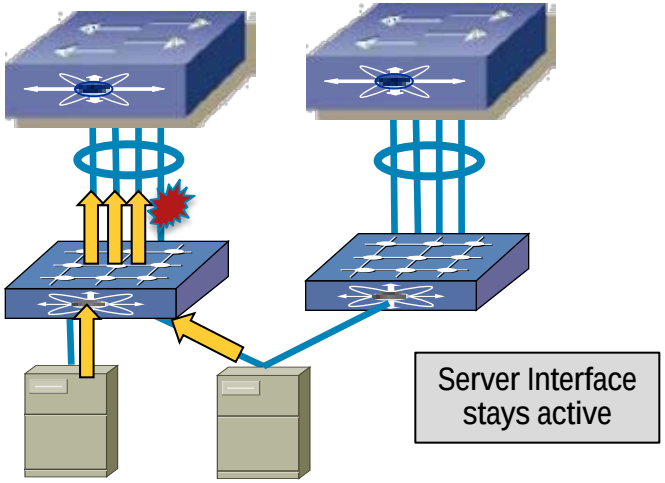
# Nexus 2000 Fabric Extender

## Nexus 2000 Packet Forwarding

- Fabric Extender associates (pins) a server side (1GE) port with a fabric uplink
- Server ports are either individually pinned to specific uplinks (static pinning) or all interfaces pinned to a single logical port channel
- Behavior on FEX uplink failure depends on the configuration
- *Static Pinning* – Server ports pinned to the specific uplink are brought down with the failure of the pinned uplink
- *Port Channel* – Server traffic is shifted to remaining uplinks based on port channel hash



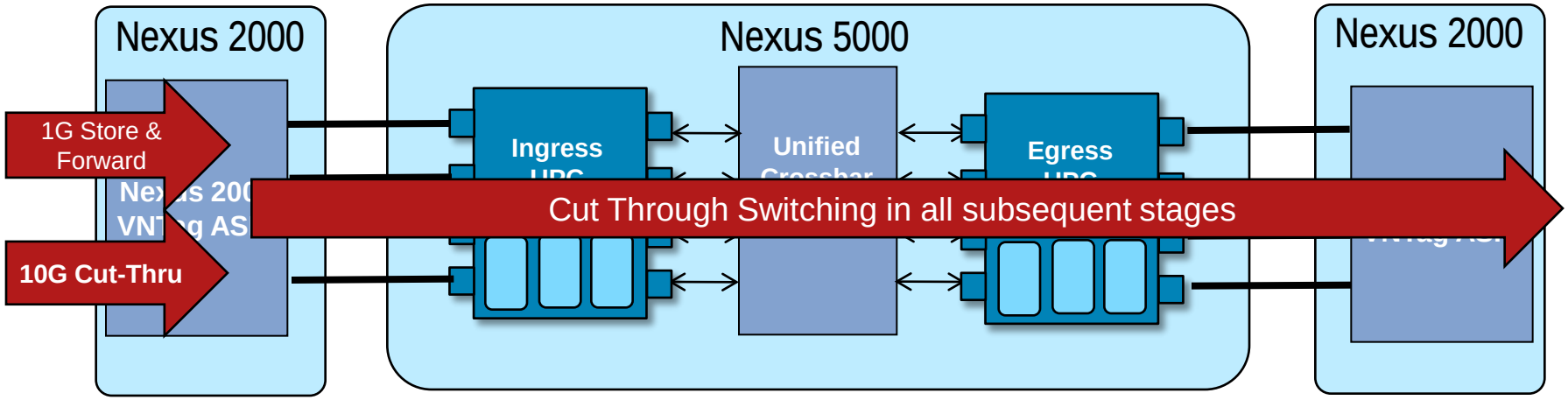

Static Pinning



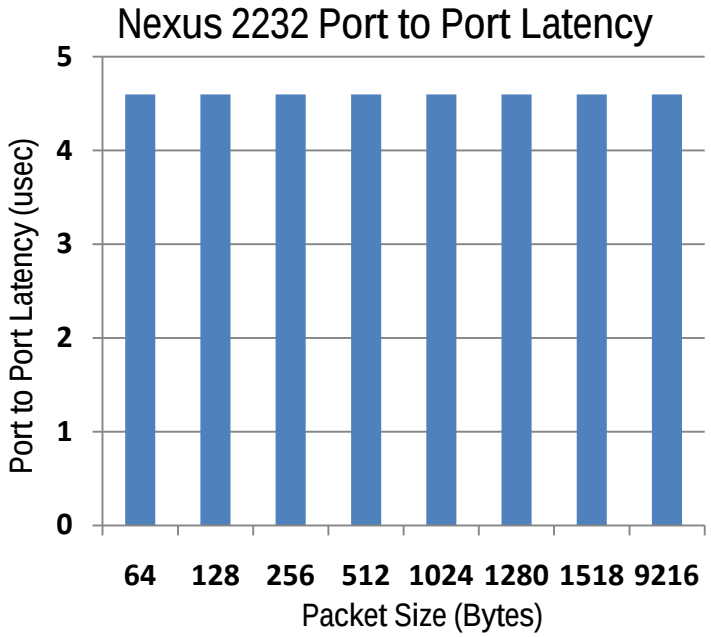

Port Channel

# Nexus 5000 and 2000 Virtual Switch

## Packet Forwarding Latency

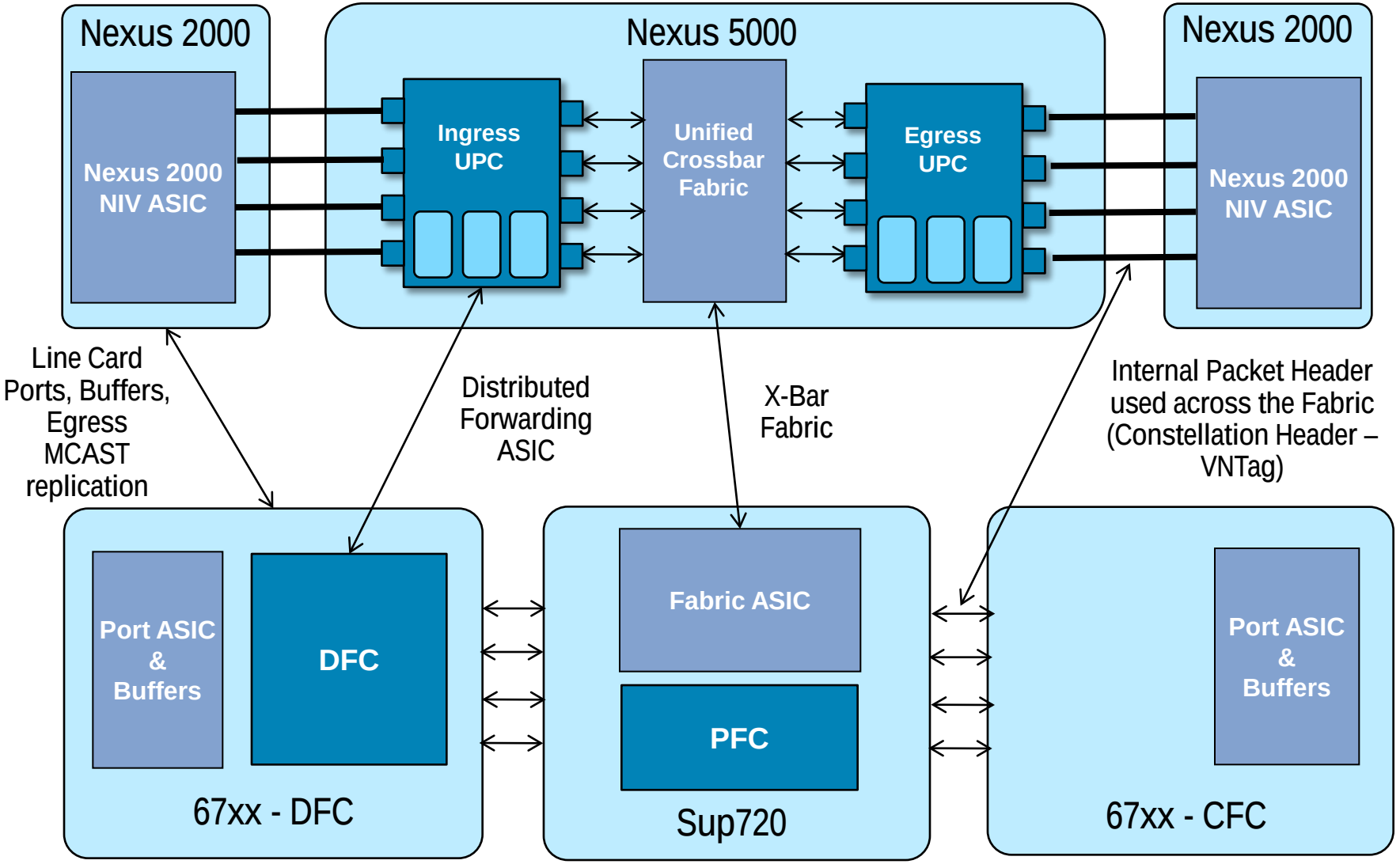



- Nexus 2000 also supports Cut -Through switching
  - 1GE to 10GE on first N2K ingress is store and forward
  - All other stages are Cut Through (10GE N2K port operates in end to end cut through)
- Port to Port latency is dependent on a single store and forward operation at most



# Nexus 5000 and 2000

## Switching Morphology—Is this Really Different?



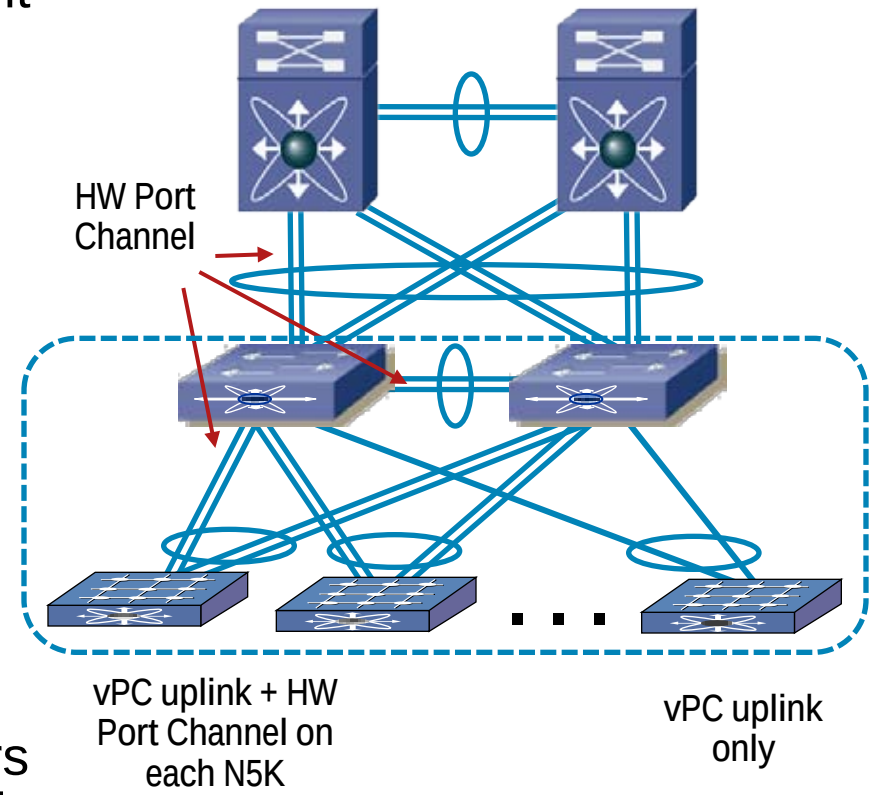
# Agenda

- Data Center Virtualized Access — Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

# Virtualized Access Switch

## Virtual Switch—How Many ???

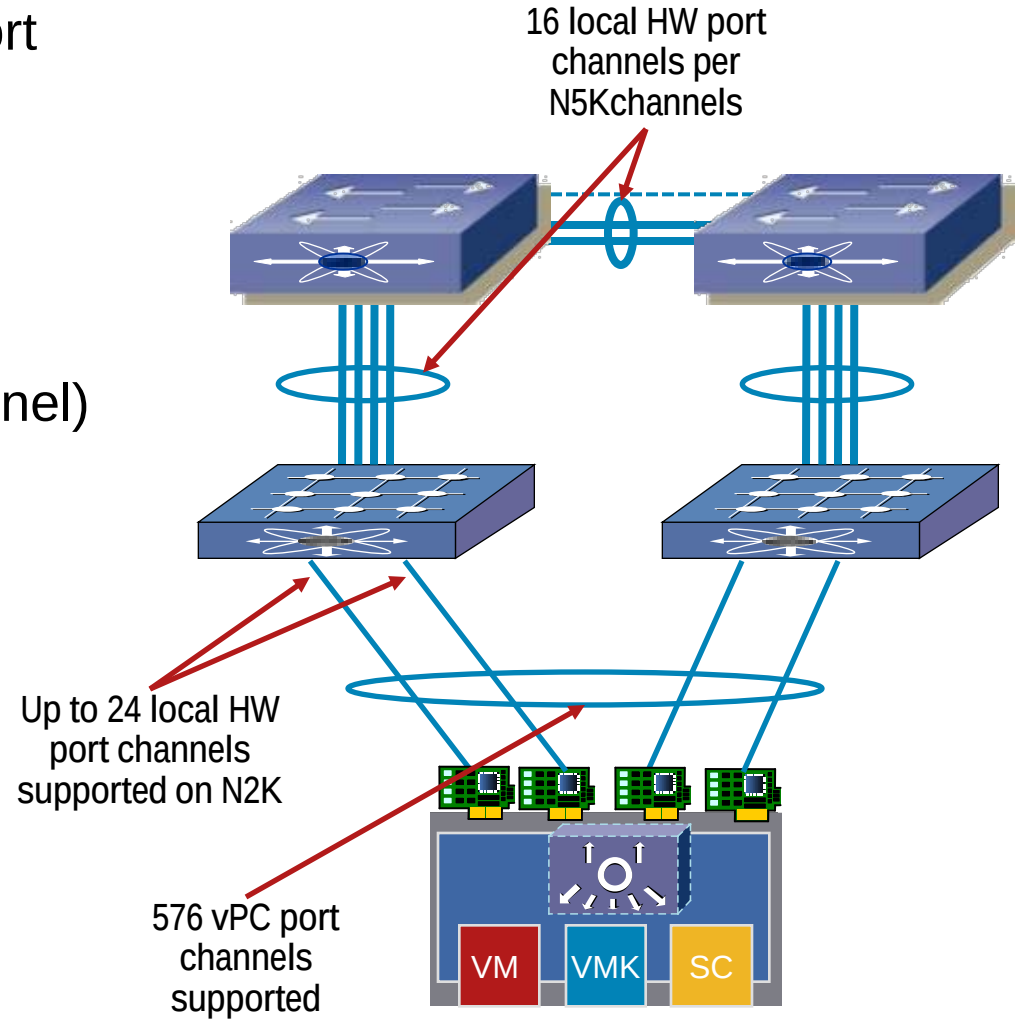
- The number of server ports, port channels, Fabric Extenders supported in a single virtual switch is dependent on two factors
- Control Plane Scalability
  - Number of Server Ports supported
  - Number of Fabric Extenders Supported
  - Number of vPC Port Channels Supported
- Physical System Scalability
  - Number of HW Port Channels
  - Number of Ports on the N5K
- The total number of Fabric Extenders that may be connected is dependent on the intersection of all of these factors



# Nexus Virtualized Access Switch

## Nexus 5000, 2148/2248/2232 and Port Channels

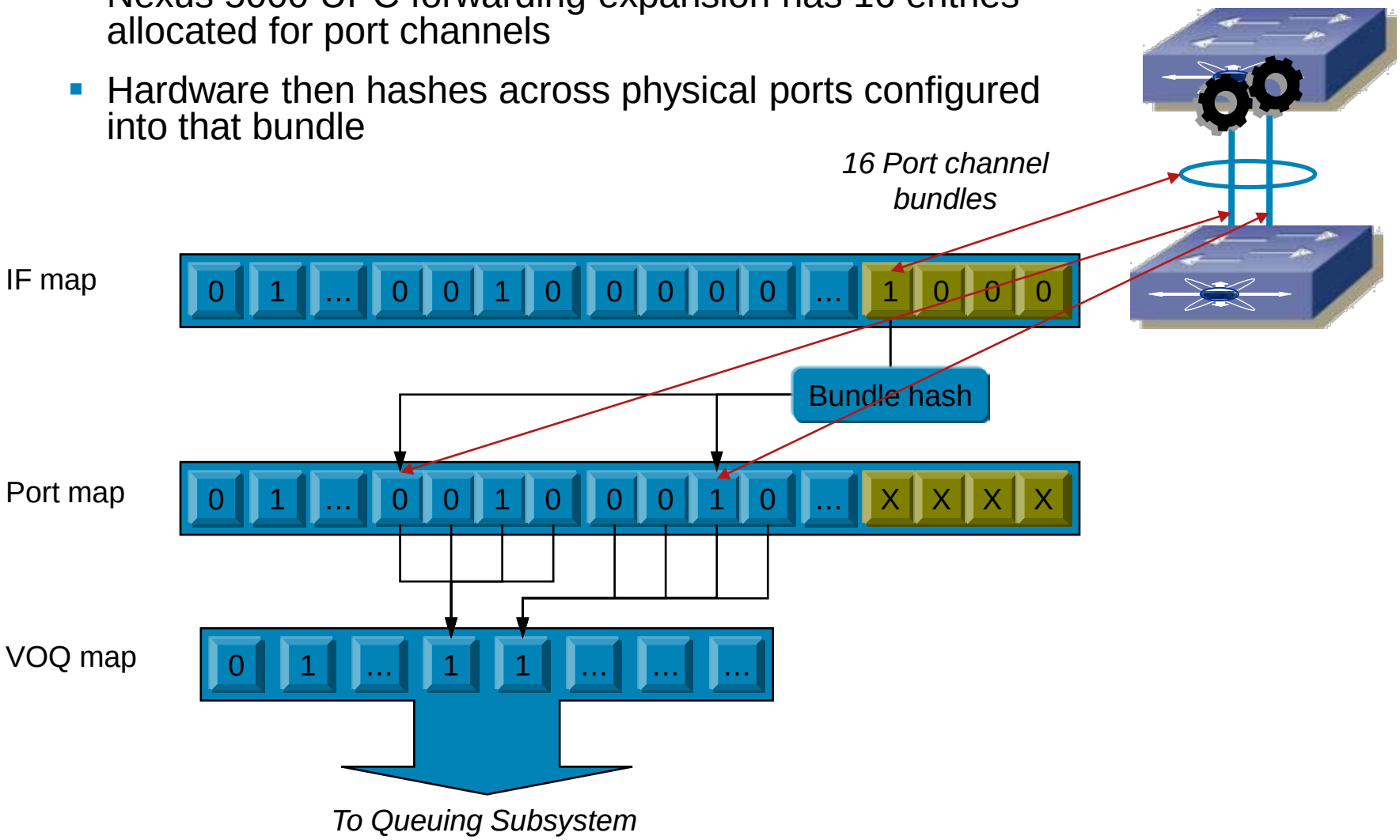
- Three distinct elements to port channel scaling
- Nexus 5000 Port Channels (ports physically attached to Nexus 5000)
  - 16 total (including both Ethernet and Fibre Channel)
- FEX Local Port Channels
  - 2148T – 0
  - 2248T – 24
  - 2232 - 16
- vPC Port Channels
  - 4.1(3) – 480
  - 4.2(1) - 576



# Nexus Virtualized Access Switch

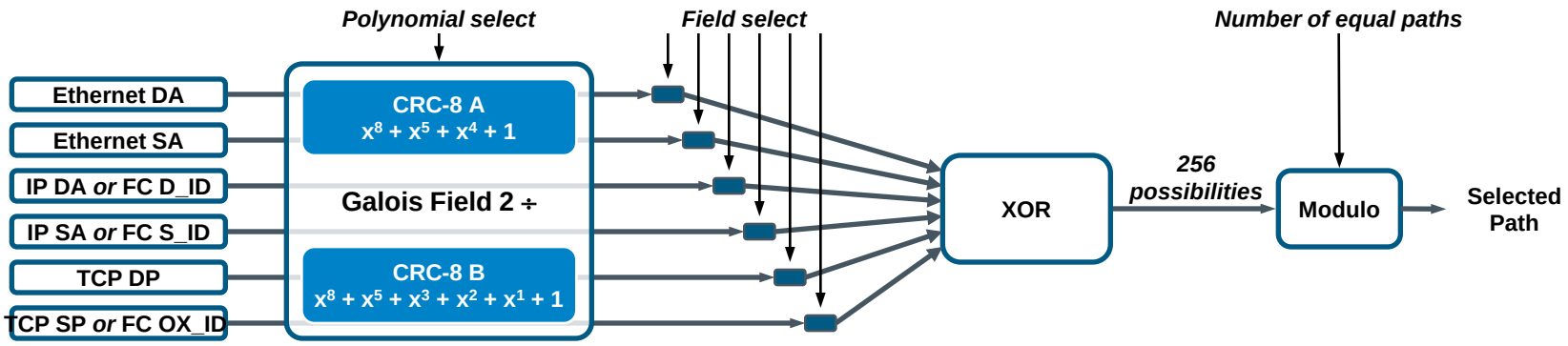
## Nexus 5000 Port Channels

- Nexus 5000 UPC forwarding expansion has 16 entries allocated for port channels
- Hardware then hashes across physical ports configured into that bundle



# Nexus Virtualized Access Switch

## Nexus 5000 Port Channel Expansion Algorithm

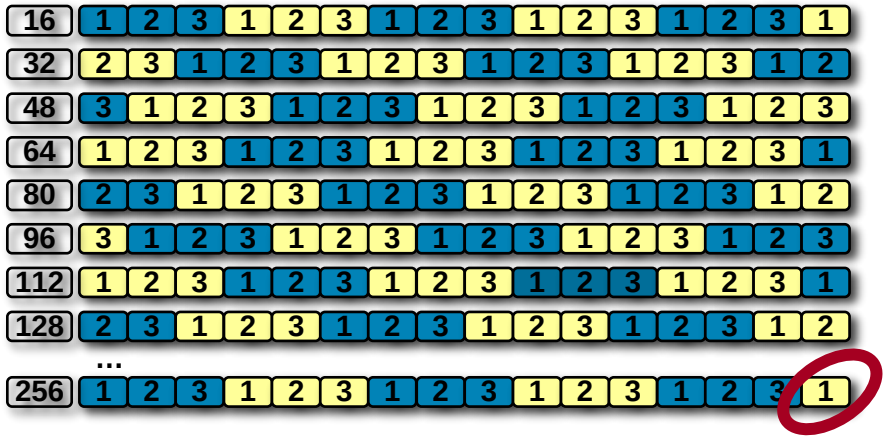



- Relevant frame fields
  - Ethernet Source Address and Destination Address always available
  - IP frames allows inclusion of IP v4/v6 Source and Destination Address
    - TCP/UDP frames can include source and destination ports
  - Fibre Channel frames can include D_ID and S_ID
    - OX_ID can also be included per VSAN
- Each field is divided by one of two CRC-8 polynomials
- Result of field CRC division is combined via bitwise XOR
- Result selected using modulo division by number of equal cost paths
  - 256 possibilities are reduced to avoid bias

# Nexus Virtualized Access Switch

## Nexus 5000 Port Channel Efficiency

- Prior generations of Etherchannel load sharing leveraged eight hash buckets
- Could lead to non optimal load sharing with an odd number of links
- Nexus 5000 and 2000 utilize 256 buckets
- Provides better load sharing in normal operation and avoids in-balancing of flows in any link failure cases

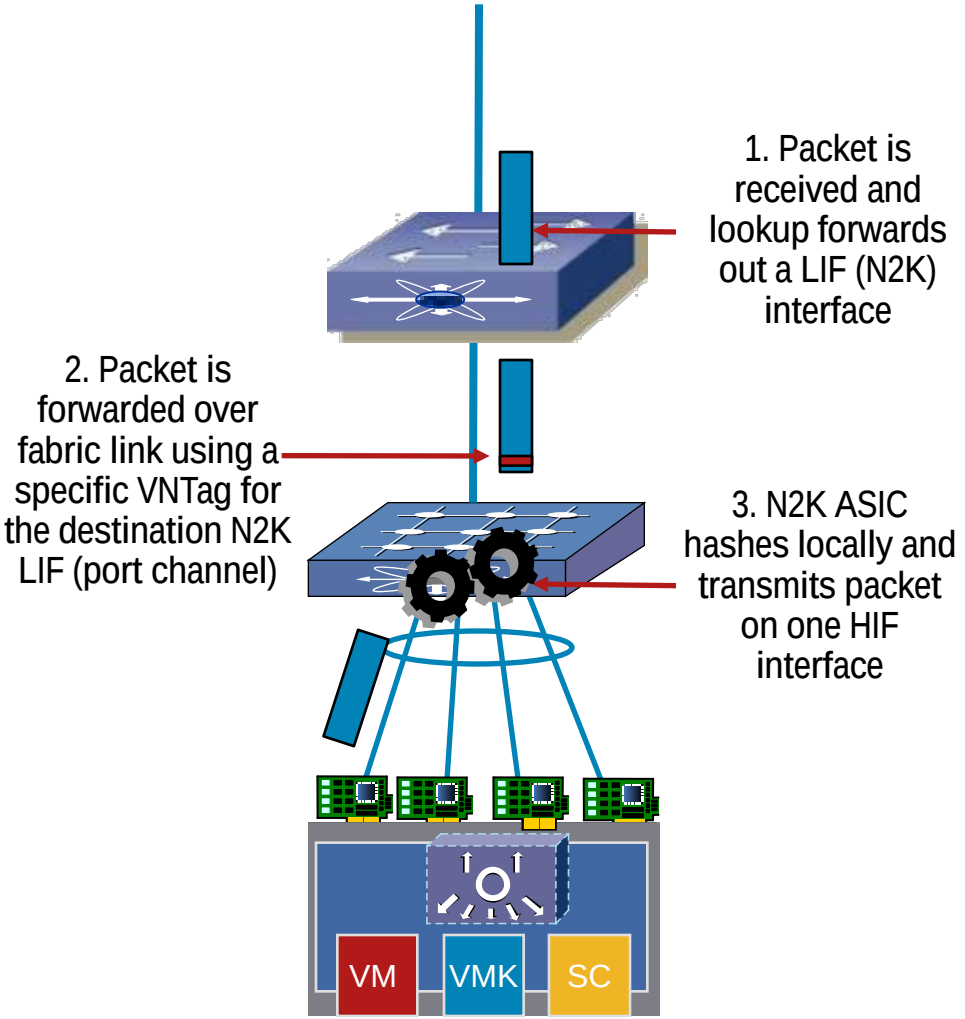




```
dc11-5020-3# sh port-channel load-balance forwarding-path interface port-channel 100
dst-ip 10.10.10.10 src-ip 11.11.11.11
Missing params will be substituted by 0's.
Load-balance Algorithm: source-dest-ip
crc8_hash: 24   Outgoing port id: Ethernet1/37
```

# Nexus Virtualized Access Switch

## Nexus 2148/2248/2232 Port Channels

- Nexus 2248/2232 FEX support local port channels
- All FEX ports are extended ports (Logical Interfaces = LIF)
- A local port channel on the N2K is still seen as a single extended port
- Extended ports are each mapped to a specific VNTag
- HW hashing occurs on the N2K ASIC
- Number of 'local' port channels on each N2K is based on the local ASIC
  - 2148T – 0
  - 2248T – 24
  - 2232 - 16



# Nexus Virtualized Access Switch

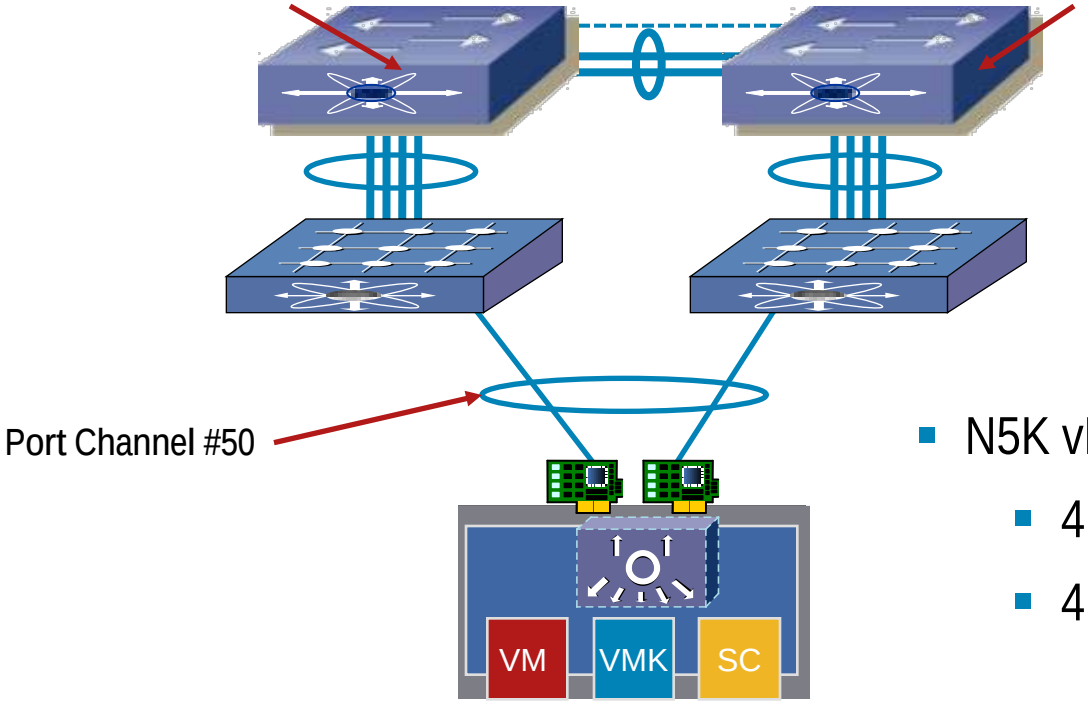
## Nexus Virtual Port Channels (vPC)

- A Virtual Port Channel (vPC) is synchronization of the forwarding behavior across two physical switches
- The number of vPC 'port channels' is based on control plane scaling

```
dc11-5020-3# sh mac-address-table int port-channel 50
VLAN      MAC Address       Type    Age       Port
---------+-----------------+-------+---------
200       001f.275e.2918    dynamic 0         Po50
200       001f.275e.7f98    dynamic 300       Po50
```

CFS

```
dc11-5020-4# sh mac-address-table in port-channel 50
VLAN      MAC Address       Type    Age       Port
---------+-----------------+-------+---------
200       001f.275e.2918    dynamic 300       Po50
200       001f.275e.7f98    dynamic 10        Po50
```

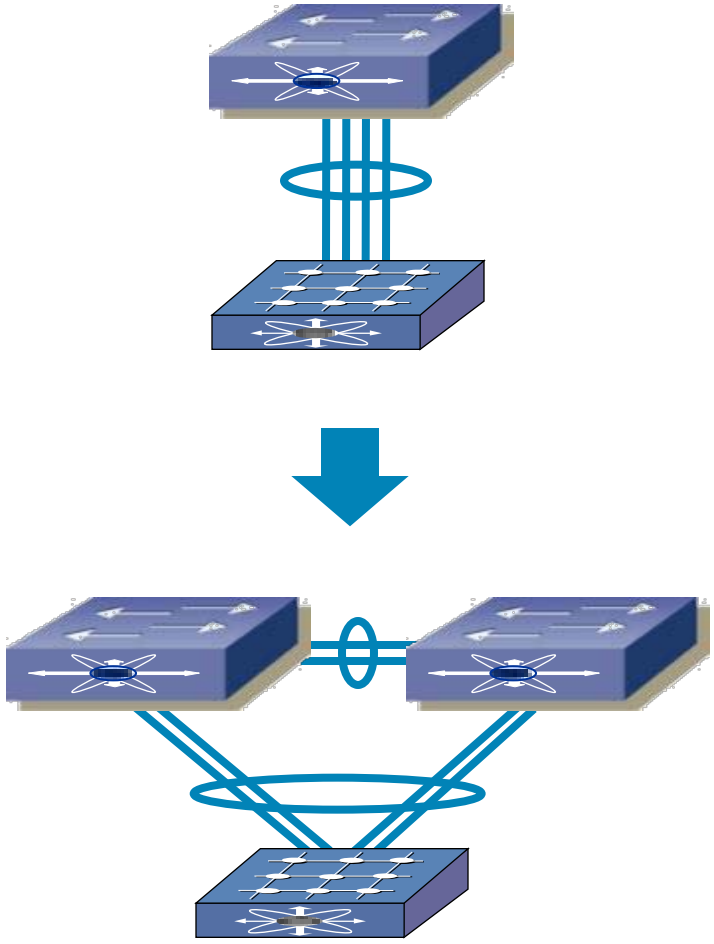



- N5K vPC Port Channels
  - 4.1(3) - 480
  - 4.2(1) - 576

# Nexus Virtualized Access Switch

## Dual Homed Nexus 2000 and vPC Host Ports

- NX-OS 4.1(3)N1 release supports vPC on the Nexus 5000 either for use to provide server NIC teaming or to provide dual supervisor configuration for the virtualized switch
- In the redundant supervisor mode the Etherchannel fabric uplink is split across two N5Ks
- Static pinning is not supported in a redundant supervisor mode
- Server ports appear on both N5K
- Currently configuration for all ports must be kept in sync manually on both N5Ks



# Nexus Virtualized Access Switch

## Fabric Link Virtual Port Channel Configuration

```
interface port-channel10
  switchport mode trunk
  switchport trunk allowed vlan 1,10
  vpc peer-link

interface port-channel50
  switchport mode fex-fabric
  vpc 50
  fex associate 100

interface Ethernet1/17
  switchport mode trunk
  switchport trunk allowed vlan 1,10
  channel-group 10 mode active

interface Ethernet1/18
  switchport mode trunk
  switchport trunk allowed vlan 1,10
  channel-group 10 mode active

interface Ethernet1/37
  switchport mode fex-fabric
  channel-group 50
  fex associate 100

interface Ethernet1/38
  switchport mode fex-fabric
  channel-group 50
  fex associate 100

fex 100
  pinning max-links 1
```

**Configure the vPC Peer Link (Full vPC Configuration not included in this example)**

**Configure the Physical Ports as Members of the Fabric EtherChannel**

**Configure the Port Channel and Its Members to be Associated with a Specific Fabric Extender**

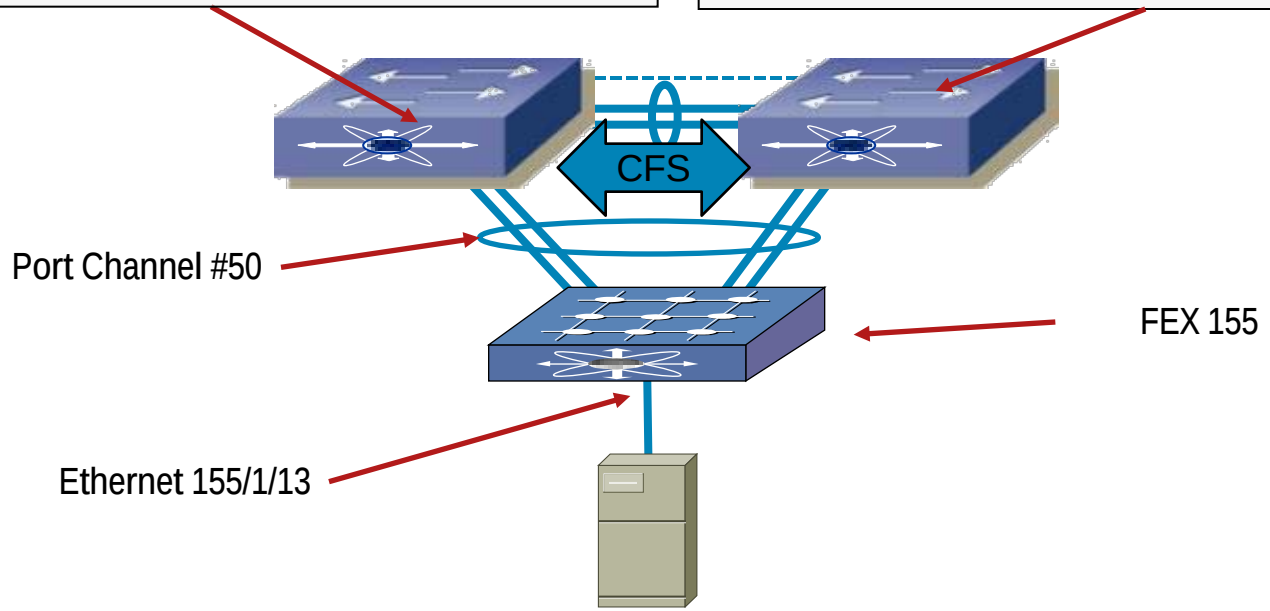
# Nexus Virtualized Access Switch

## Nexus 2000 vPC Host Ports

- A port on a dual homed Nexus 2000 is known as a vPC Host Port
- LIF port state is replicated and synchronized across both Nexus 5000 (CPU and memory load is replicated *not* distributed)

```
dc11-5020-3# sh vpc
<snip>
vPC status
----------------------------------------------------------------
id     Port        Status Consistency Reason    Active vlans
------ ----------- ------ ----------- --------- -----------
<snip>
157708 Eth155/1/13 up     success     success      105
```

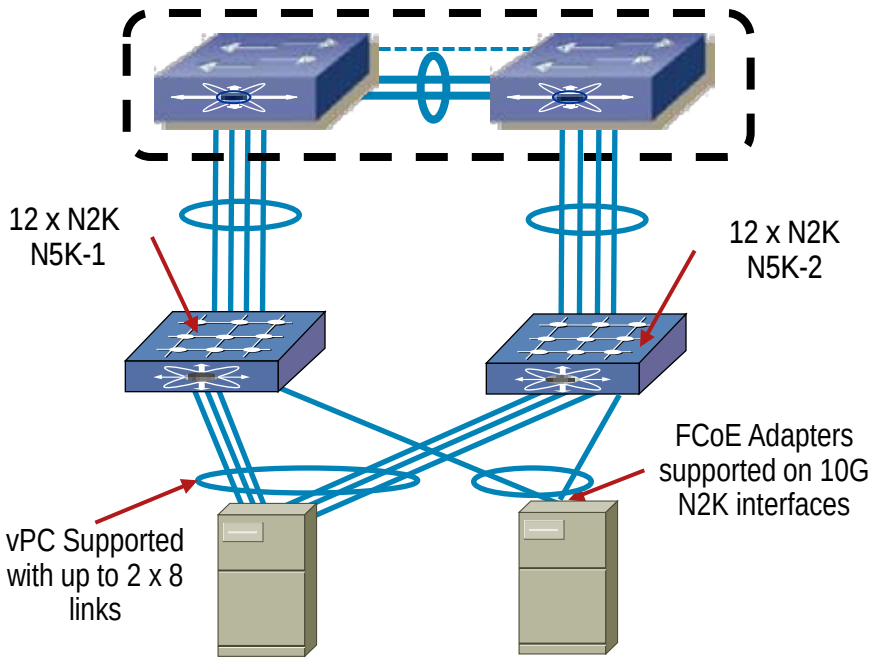
```
dc11-5020-4# sh vpc
<snip>
vPC status
----------------------------------------------------------------
id     Port        Status Consistency Reason    Active vlans
------ ----------- ------ ----------- --------- -----------
<snip>
157708 Eth155/1/13 up     success     success      105
```



# Nexus Virtualized Access Switch

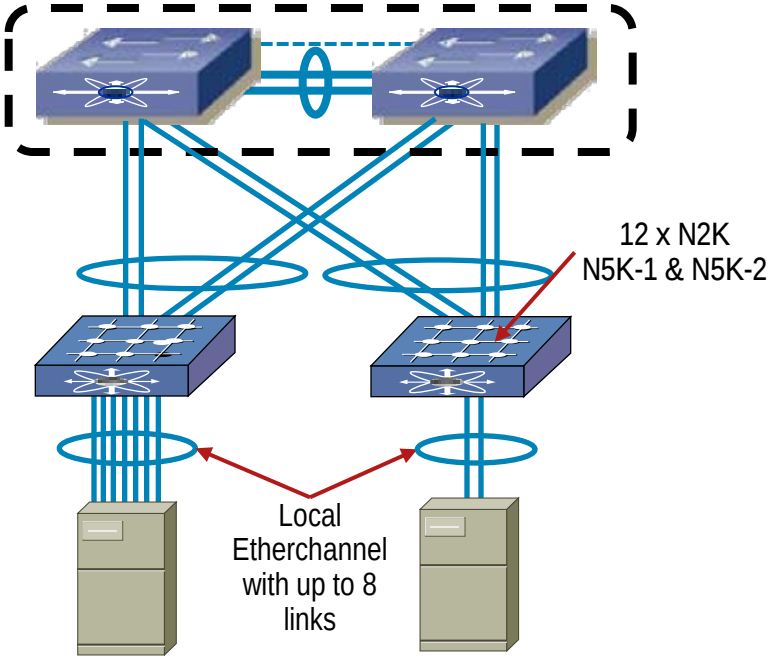
## Nexus 2248/2232 Port Channel Scaling

Single Homed (Straight Through)




- A maximum of 576 vPC port channels (max of 16 links per vPC port channel)
- Maximum of 12 x FEX connected to ***each*** Nexus 5000 (576 ports on each Nexus 5000 = 1152 total)

Dual Homed Nexus 2000

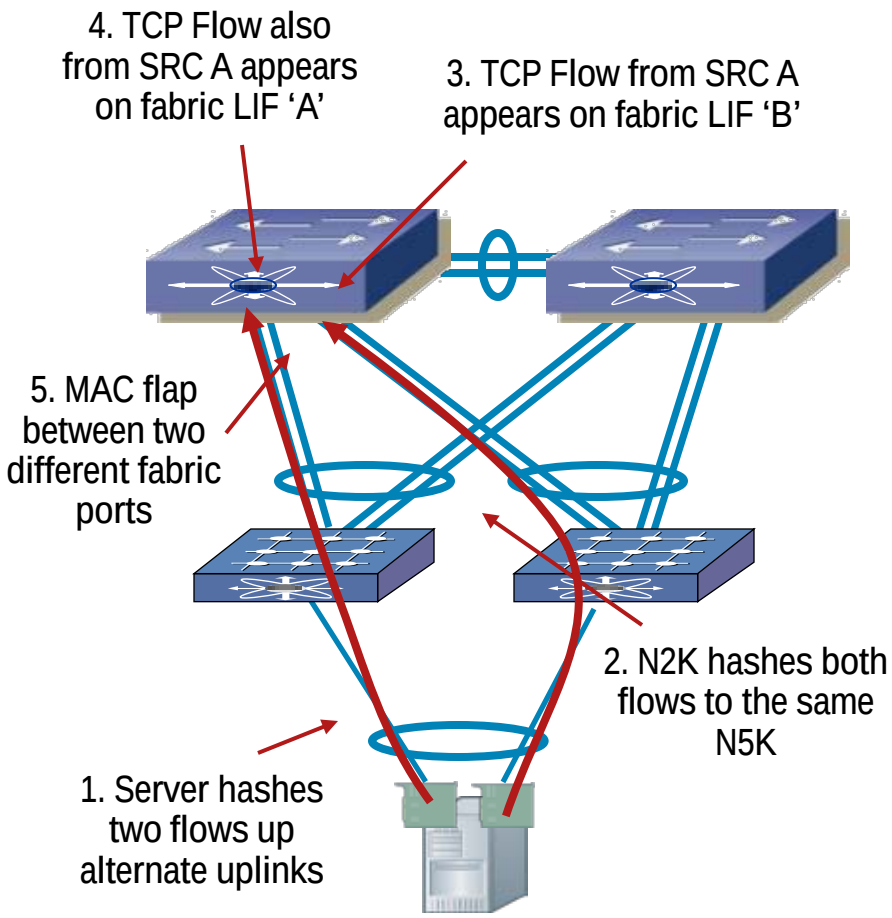



- Maximum of 288 HIF port channels (maximum of 8 links per port channel)
- Maximum of 12 x FEX connected to ***both*** Nexus 5000 (576 ports total)

# Nexus Virtualized Access Switch

## "Dual Tier vPC"—Port Channel of a Port Channel

- "Dual tiered" vPC is ***not*** currently supported
- vPC provides a logical port channel interface on the N5K
- In the *unsupported* configuration shown each N2K is attached with a port channel fabric interface and then carried over that interface is a second server port channel interface
- Two dependent layers of Etherchannel hashing (Server and N2K) for the same flows
- Two tiers of vPC as shown will result in MAC addresses flapping between two fabric ports (LIF)
- Timeframes to support this configuration is targeted for 1HCY11



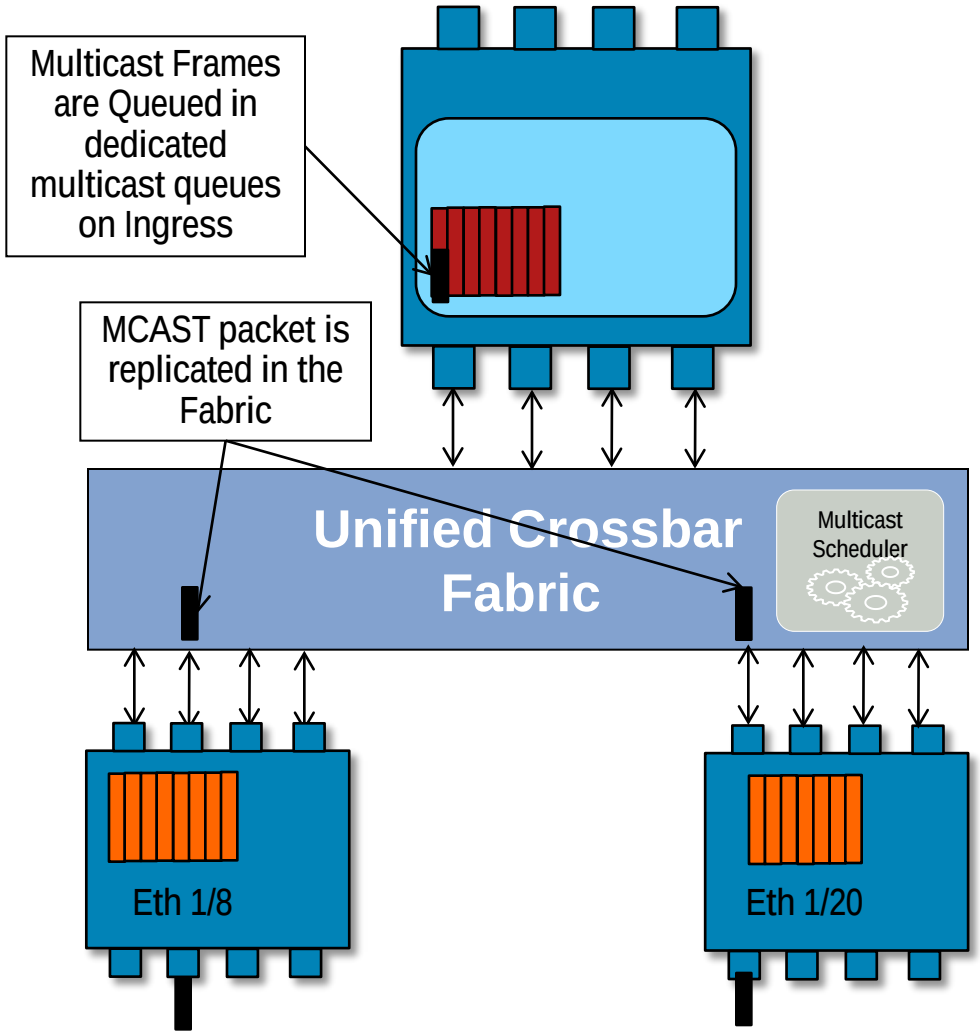

"Dual Tier vPC" as shown above is *not* currently supported

# Agenda

- Data Center Virtualized Access — Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

# Nexus 5000 Multicast Forwarding

## Fabric-Based Replication

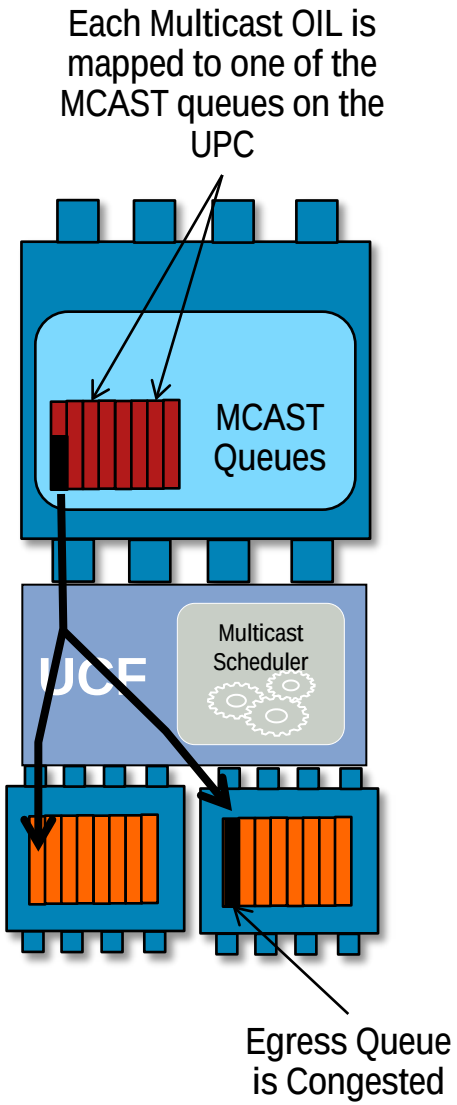



- Nexus 5000 uses fabric based egress replication
- Traffic is queued in the ingress UPC for each MCAST group
- When the scheduler permits the traffic if forwarded into the fabric and replicated to all egress ports
- When possible traffic is super-framed (multiple packets are sent with a single fabric scheduler grant) to improve throughput

# Nexus 5000 Multicast Forwarding

## Multicast Queues and Multicast Group Fan-Out

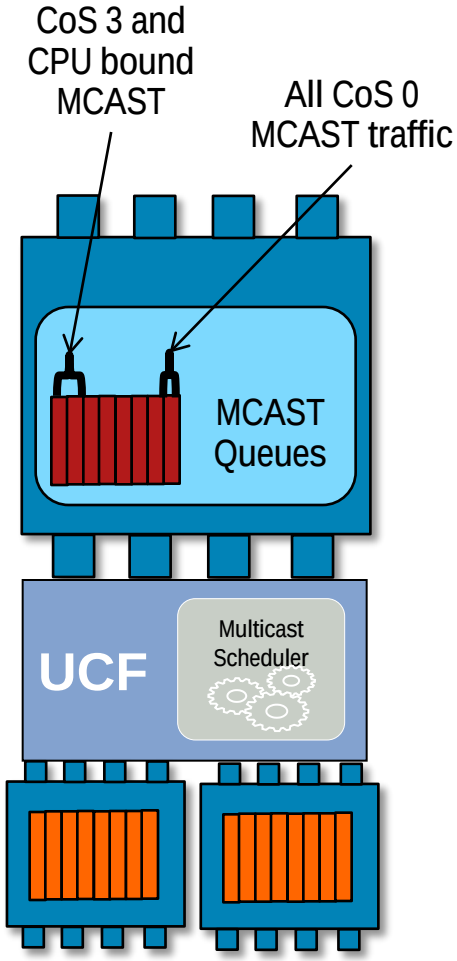
- A “FAN-OUT” = is an Output Interface List (OIL)
- The Nexus 5000 currently supports 1000 fan-outs and 4000 Multicast Groups
- The multicast groups need to be mapped to the 1000 fan-outs
- There are eight multicast queues per UPC forwarding engine (no VoQ for multicast)
- Hardware needs to map fan-outs to the eight queues
- Multicast scheduler waits until all egress queues are free to accept a frame before traffic in that queue is replicated across the fabric



# Nexus 5000 Multicast Forwarding

## Multicast Queues and Multicast Group Fan-Out

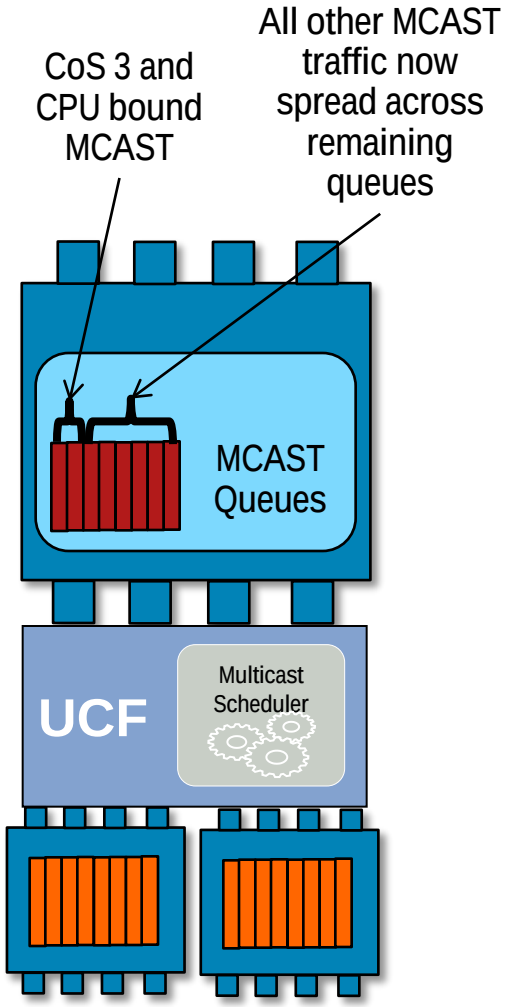
- Overlap of multicast groups to fan-outs to queues can result in contention for the fabric for a specific group
- Tuning of the multicast traffic and fan-out mapping to queues can be used to prioritize specific groups access to the fabric
- Of the eight queues available for multicast two are reserved (FCoE and sup-redirect multicast) leaving six for the remainder of the multicast traffic
- By default the switch uses the frame CoS to identify the multicast queue for a specific group
- If more groups are mapped to one CoS group than another the system queuing for multicast may be non-optimal



# Nexus 5000 Multicast Forwarding

## Multicast Queues and Multicast-optimization

- "Multicast-optimize" when enabled for a class of traffic assigns multicast fan-outs in that class to any unused CoS queues on a round robin basis
- With multicast optimization, you can assign these classes of traffic to the unused queues
  - One 'class of service' (CoS-based)
  - IP multicast (traffic-based)
  - All flood (traffic-based)

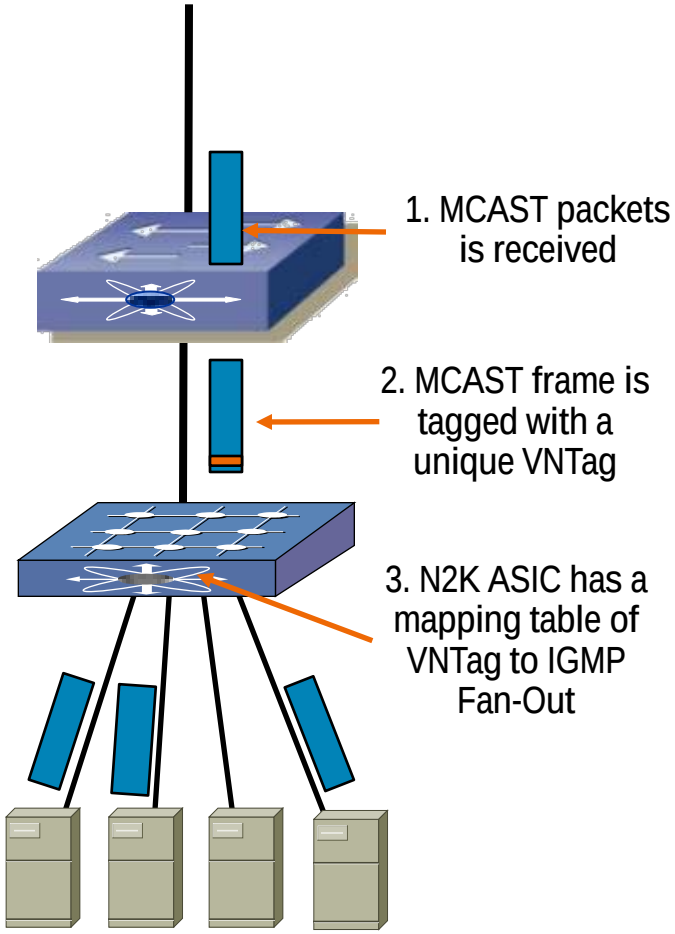



```
class-map type qos class-ip-multicast
policy-map type qos MULTICAST-OPTIMIZE
  class class-ip-multicast
    set qos-group 2
class-map type network-qos class-ip-multicast
  match qos-group 2
policy-map type network-qos MULTICAST-OPTIMIZE
  class type network-qos class-ip-multicast
    multicast-optimize
  class type network-qos class-default
system qos
  service-policy type qos input MULTICAST-OPTIMIZE
  service-policy type network-qos MULTICAST-OPTIMIZE
```

# Nexus Virtualized Access Switch

## Nexus 2000 Multicast Forwarding

- Nexus 2000 supports egress based Multicast replication
- Each fabric link has a list of VNTag's associated with each Multicast group
- A single copy of each multicast frame is sent down the fabric links to the Nexus 2000
- Extended Multicast VNTag has an associated flooding fan-out on the Nexus 2000 built via IGMP Snooping
- Nexus 2000 replicates and floods the multicast packet to the required interfaces
- Note: When the fabric links are configured using static pinning each fabric link needs a separate copy of the multicast packet (each pinned group on the Nexus 2000 replicates independently)
- Port Channel based fabric links only require a single copy of the multicast packet



# Agenda

- Data Center Virtualized Access — Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch—Nexus 5000 and Nexus 2000

# Nexus 5000 QoS

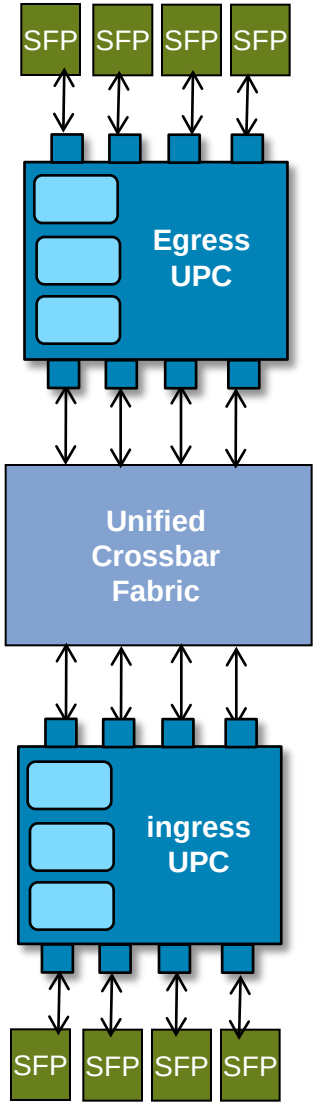
## QoS Capabilities and Configuration

- Nexus 5000 supports a new set of QoS capabilities designed to provide per system class based traffic control
  - Lossless Ethernet—Priority Flow Control (IEEE 802.1Qbb)
  - Traffic Protection—Bandwidth Management (IEEE 802.1Qaz)
  - Configuration signaling to end points—DCBX (part of IEEE 802.1Qaz)
- These new capabilities are added to and managed by the common Cisco MQC (Modular QoS CLI) which defines a three-step configuration model
  - Define matching criteria via a *class-map*
  - Associate action with each defined class via a *policy-map*
  - Apply policy to entire system or an interface via a *service-policy*
- Nexus 5000/7000 leverage the MQC qos-group capabilities to identify and define traffic in policy configuration

# Nexus 5000 QoS
## QoS Defaults

- QoS is enabled by default (not possible to turn it off)
- Four default class of services defined when system boots up
  - Two for control traffic (CoS 6 & 7)
  - One for FCoE traffic (class-fcoe – CoS 3)
  - Default Ethernet class (class-default – all others)
- Control traffic is treated as strict priority and serviced ahead of data traffic
- The two base user classes (class-fcoe and class-default) get 50% of guaranteed bandwidth by default

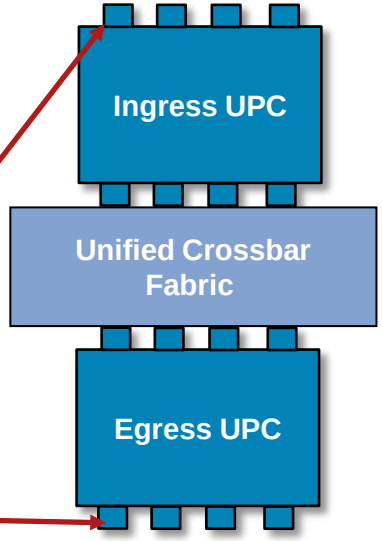
```
dc11-5020-2# sh policy-map system type qos input
<snip>
    Class-map (qos):   class-fcoe (match-any)
      Match: cos 3
      set qos-group 1

    Class-map (qos):   class-default (match-any)
      Match: any
      set qos-group 0
```



# Nexus 5000 QoS

## QoS Policy Types

- There are three QoS policy types used to define system behavior (qos, queuing, network-qos)
- There are three policy attachment points to apply these policies to
  - Ingress interface
  - System as a whole (defines global behavior)
  - Egress interface




| Policy Type | Function | Attach Point |
|---|---|---|
| qos | Define traffic classification rules | system qos<br>ingress Interface |
| queuing | Strict Priority queue<br>Deficit Weight Round Robin | system qos<br>egress Interface<br>ingress Interface |
| network-qos | System class characteristics (drop or no-drop, MTU), Buffer size, Marking | system qos |

# Nexus 5000 QoS

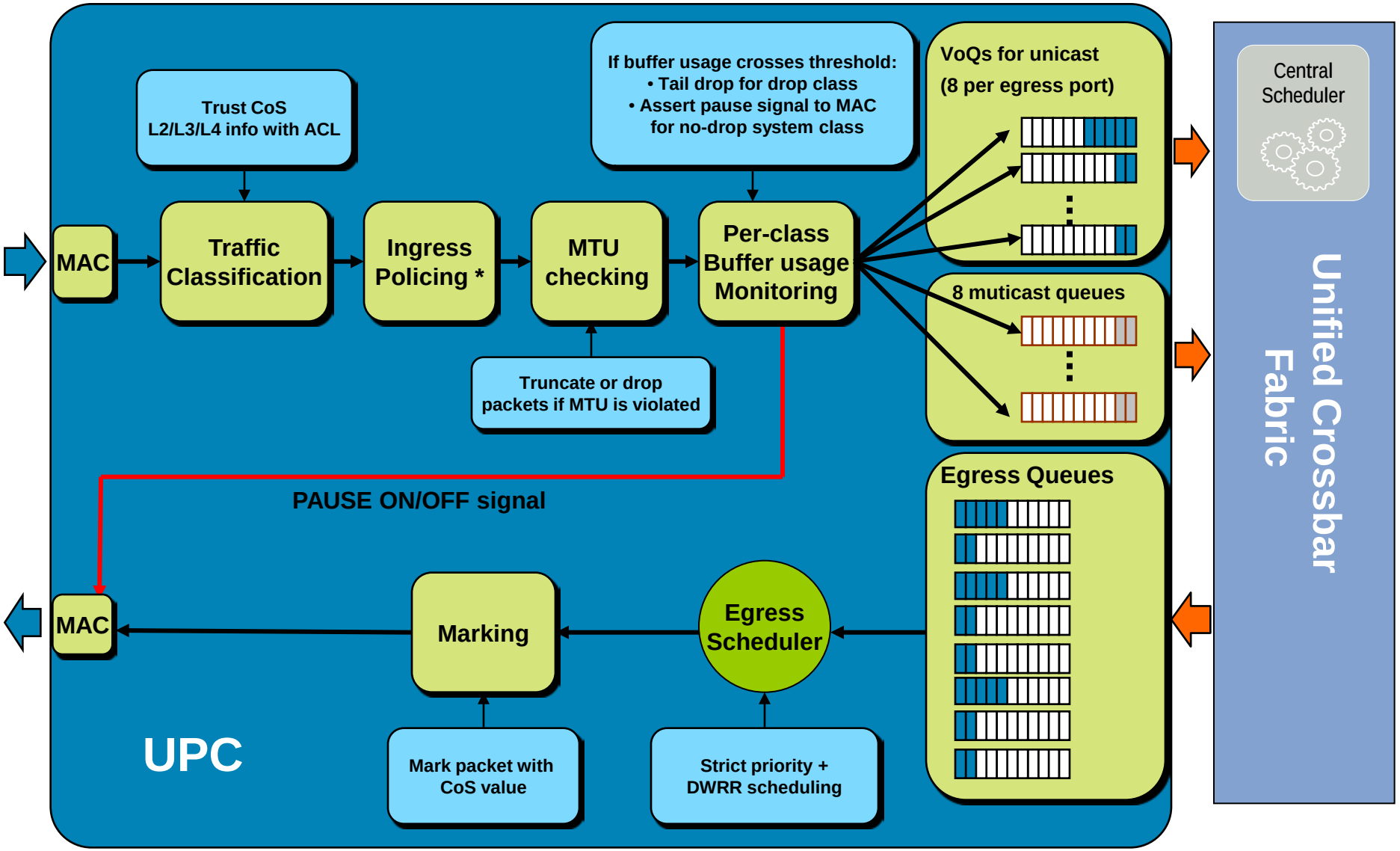
## UPC Buffering

- 480KB dedicated packet buffer per one 10GE port or per two FC ports
- Buffer is shared between ingress and egress with majority of buffer being allocated for ingress
  - Ingress buffering model
  - Buffer is allocated per system class
  - Egress buffer only for in flight packet absorption
- Buffer size of ingress queues for drop class can be adjusted using *network-qos* policy

| Class of Service | Ingress Buffer(KB) | Egress Buffer(KB) |
|---|---|---|
| Class-fcoe | 76.8 | 18.8 |
| User defined no-drop class of service with MTU<2240 | 76.8 | 18.8 |
| User defined no-drop class of service with MTU>2240 | 81.9 | 18.8 |
| Tail drop class of service | 20.4 | 18.8 |
| Class-default | All remaining buffer | 18.8 |

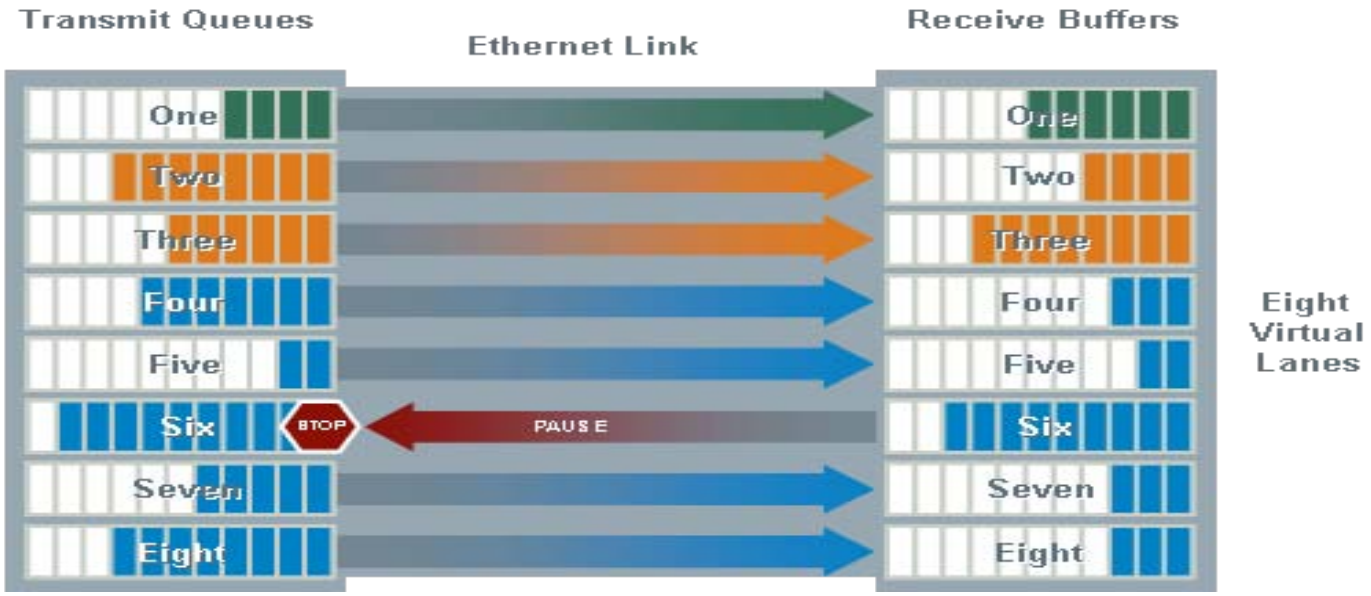
# Nexus 5000 QoS

## UPC QoS Capabilities (*Ingress Policing not Currently Supported)



# Nexus 5000 QoS

## Priority Flow Control and No-Drop Queues

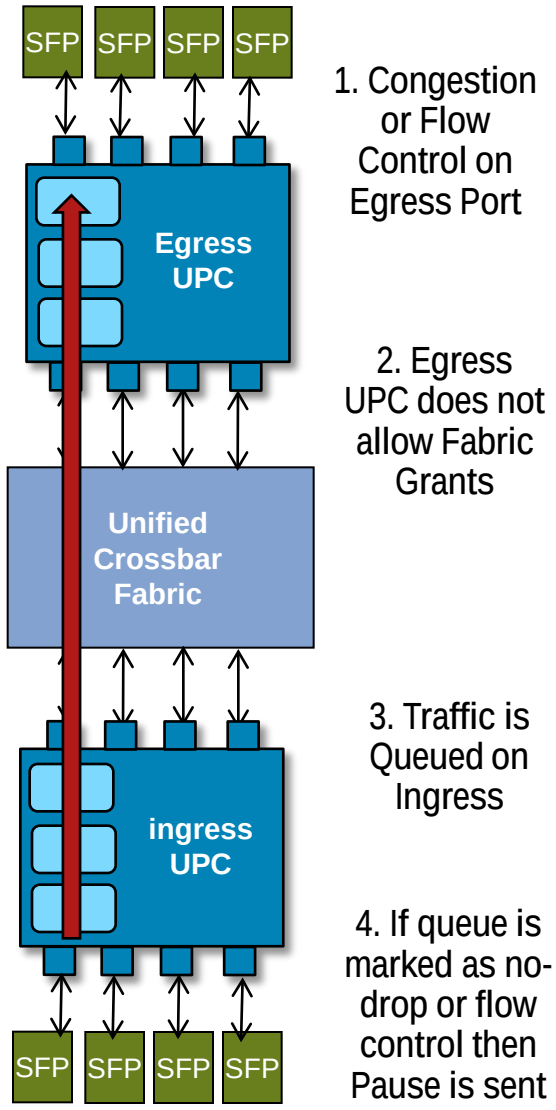



- Nexus 5000 supports a number of new QoS concepts and capabilities
- Priority Flow Control is an extension of standard 802.3x pause frames
- No-drop queues provide the ability to support loss-less Ethernet using PFC as a per queue congestion control signaling mechanism

# Nexus 5000 QoS

## Priority Flow Control and No-Drop Queues

- Actions when congestion occurs depending on policy configuration
  - PAUSE upstream transmitter for lossless traffic
  - Tail drop for regular traffic when buffer is exhausted
- Priority Flow Control (PFC) or 802.3X PAUSE can be deployed to ensure lossless for application that can't tolerate packet loss
- Buffer management module monitors buffer usage for no-drop class of service. It signals MAC to generate PFC (or link level PAUSE) when the buffer usage crosses threshold
- FCoE traffic is assigned to *class-fcoe*, which is a no-drop system class
- Other class of service by default have normal drop behavior (tail drop) but can be configured as no-drop

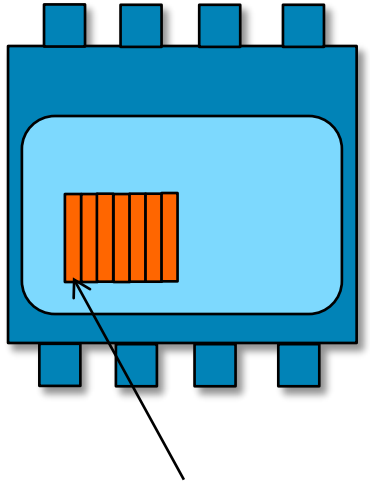


# Nexus 5000 QoS

## MTU per Class of Service (CoS Queue)

- MTU can be configured for each class of service (no interface level MTU)
- No fragmentation since Nexus 5000 is a L2 switch
- When forwarded using cut-through, frames are truncated if they are larger than MTU
- When forwarded using store-and-forward, frames are dropped if they are larger than MTU

```
class-map type qos iSCSI
  match cos 2
class-map type queuing iSCSI
match qos-group 2
policy-map type qos iSCSI
  class iSCSI
    set qos-group 2
class-map type network-qos iSCSI
  match qos-group 2
policy-map type network-qos iSCSI
  class type network-qos iSCSI
    mtu 9216
system qos
  service-policy type qos input iSCSI
  service-policy type network-qos iSCSI
```



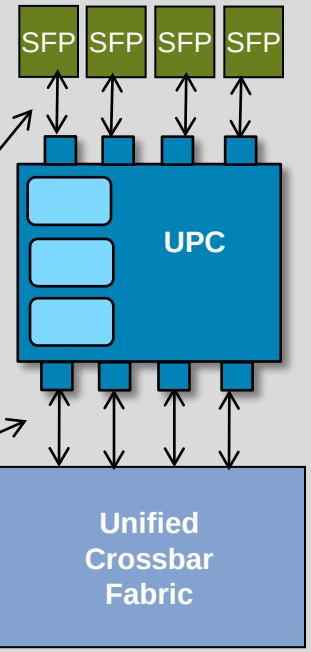
Each CoS queue on the Nexus 5000 supports a unique MTU

# Nexus 5000 QoS

## Mapping the Switch Architecture to ‘show queuing’

```
dc11-5020-4# sh queuing int eth 1/39

Interface Ethernet1/39 TX Queuing
qos-group  sched-type  oper-bandwidth
    0       WRR             50
    1       WRR             50

Interface Ethernet1/39 RX Queuing
qos-group  0
    q-size: 243200, HW MTU: 1600 (1500 configured)
    drop-type: drop, xon: 0, xoff: 1520
    Statistics:
        Pkts received over the port             : 85257
        Ucast pkts sent to the cross-bar        : 930
        Mcast pkts sent to the cross-bar        : 84327
        Ucast pkts received from the cross-bar  : 249
        Pkts sent to the port                   : 133878
        Pkts discarded on ingress               : 0
        Per-priority-pause status               : Rx (Inactive), Tx (Inactive)

<snip – other classes repeated>

Total Multicast crossbar statistics:
    Mcast pkts received from the cross-bar      : 283558
```

Egress (Tx) Queuing Configuration

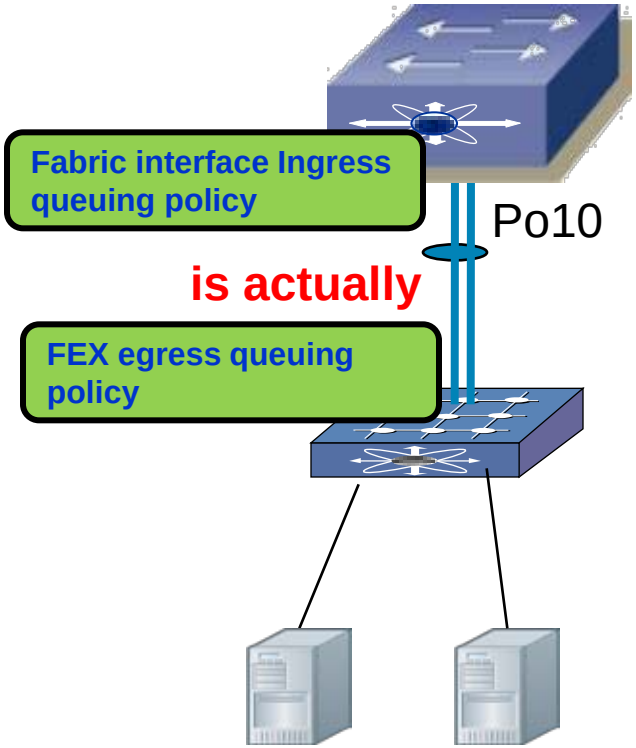



Packets Arriving on this port but dropped from ingress queue due to congestion on egress port

# Nexus Virtualized Access Switch

## Nexus 2000 QoS

- Nexus 2000 currently only supports CoS based traffic classification
- Nexus 2000 currently supports two classes of services and two queues
- The command *untagged cos* can be used to prioritize traffic from certain server interfaces
- Bandwidth allocation for traffic from FEX to Nexus 5000
  - Create new system class with *qos* and *network-qos* policy-map
  - Define *queuing* policy-map and apply it under fabric interface as INPUT queuing policy
- Bandwidth allocation for traffic from Nexus 5000 to Nexus 2000
  - Create new system class with *qos* and *network-qos* policy-map
  - Define *queuing* policy-map and apply it under *system qos* or fabric interface as OUTPUT queuing policy



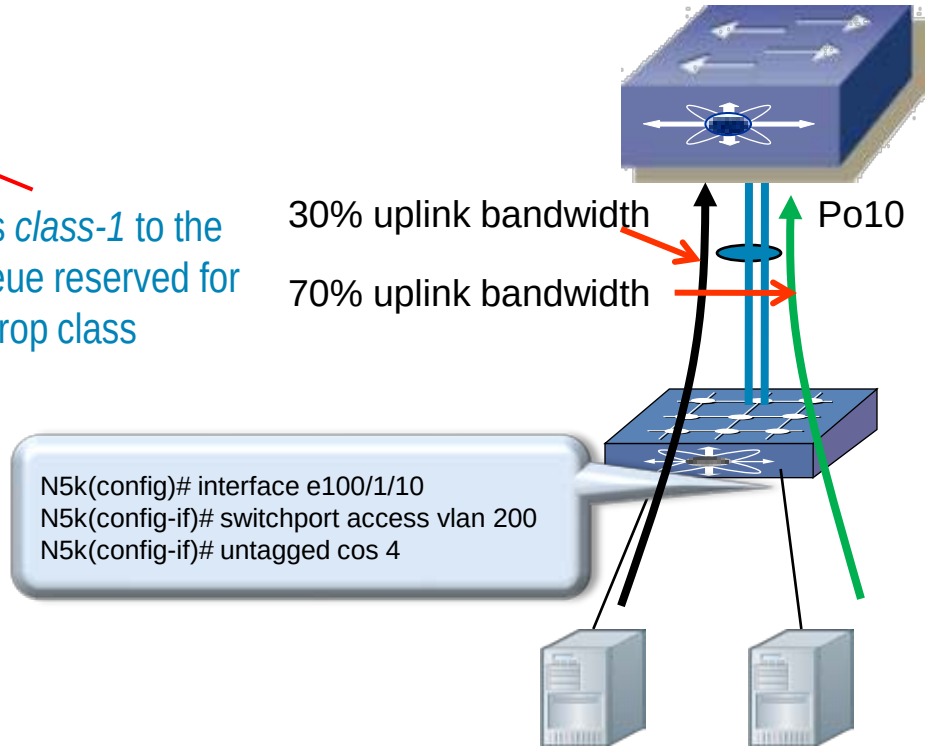
# Nexus Virtualized Access Switch

## Nexus 2000 QoS

```
N5k(config)# class-map type qos class-1
N5k(config-cmap-qos)# match cos 4
N5k(config-cmap-qos)#policy-map type qos policy-qos
N5k(config-pmap-qos)# class type qos class-1
N5k(config-pmap-c-qos)# set qos-group 2
N5k(config)# system qos
N5k(config-sys-qos)# service-policy type qos input policy-qos

N5k(config)# class-map type network-qos class-1
N5k(config-cmap-nq)# match qos-group 2
N5k(config)# policy-map type network-qos policy-nq
N5k(config-pmap-nq)# class type network-qos class-1
N5k(config-pmap-class)#pause no-drop
N5k(config-pmap-nq-c)# system qos
N5k(config-sys-qos)# service-policy type network-qos policy-nq

N5k(config-sys-qos)#class-map type queuing  class-1
N5k(config-cmap-que)# match qos-group 2

N5k(config-cmap-que)# policy-map type queuing policy-BW
N5k(config-pmap-que)# class type queuing class-fcoe
N5k(config-pmap-c-que)# bandwidth percent 0
N5k(config-pmap-c-que)# class type queuing class-default
N5k(config-pmap-c-que)# bandwidth percent 30
N5k(config-pmap-c-que)# class type queuing class-1
N5k(config-pmap-c-que)# bandwidth percent 70
N5k(config-pmap-c-que)#interface Po10
N5k(config-if)# service-policy type queuing input policy-BW

N5k(config-if)#interface e100/1/10
N5k(config-if)# switchport access vlan 200
N5k(config-if)# untagged cos 4
```

- Assign a CoS value for traffic received from high priority interfaces
- Create new system classes
- On Nexus 5000 allocate proper uplink bandwidth for new system class with input queuing policy

FEX maps *class-1* to the second queue reserved for no-drop class



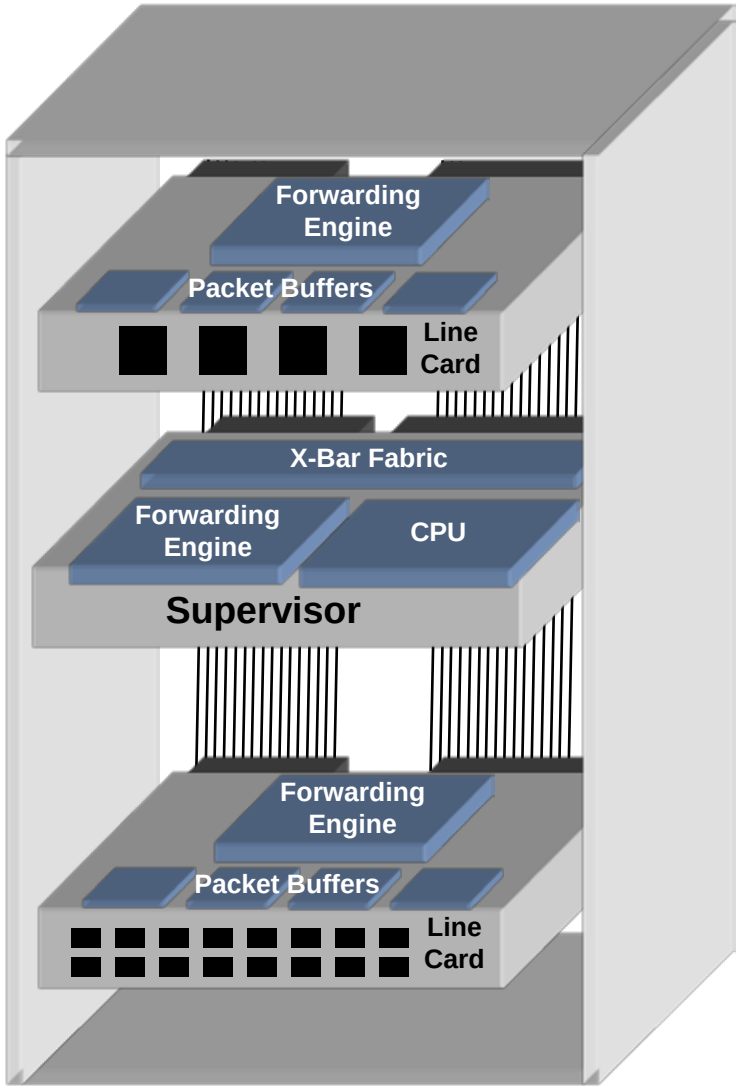
# Agenda

- Data Center Virtualized Access—Nexus 5000 and Nexus 2000
- Nexus 5000 (N5K)
  - Hardware Architecture
  - Day in the Life of a Packet
- Nexus 2000 (N2K)
  - Virtualized/Remote Line Card
  - N2K Hardware Architecture
  - Day in the Life of a Packet
- Virtualized Access Switch (N5K + N2K)
  - Port Channel and N2K Connectivity
  - Multicast
  - QoS
- Data Center Switch- Nexus 5000 and Nexus 2000

# Nexus 5000 and 2000 Architecture

## Data Center Switch

- The functional elements of the Nexus 5000 and 2000 are familiar
  - Distributed forwarding—L2/L3 forwarding, ACL, QoS TCAM
  - Protected management and control plane
  - Non-blocking cross bar switching fabric
  - Flexible connectivity through multiple line cards
- Some new capabilities and physical form factor
  - QoS - DCB, per class MTU, no-drop queues and VoQ
  - Multiprotocol—Ethernet and FC/FCoE forwarding
  - Remote Line Cards (FEX & VNTag)



# Conclusion

- You should now have a thorough understanding of the Nexus 5000 Data Center switch and the Nexus 2000 Fabric Extender packet flows, and key forwarding engine functions…
- **Any questions?**

# Q and A

# 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.

# Enter to Win a 12-Book Library of Your Choice from Cisco Press

Visit the Cisco Store in the World of Solutions, where you will be asked to enter this Session ID code


Check the **Recommended Reading** brochure for suggested products available at the Cisco Store


CISCO