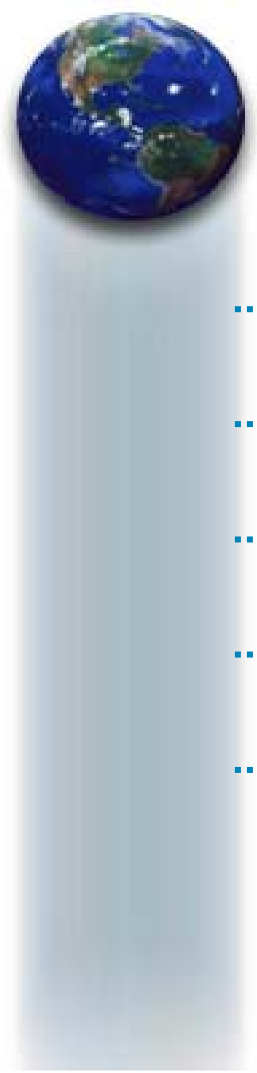
Cisco Catalyst 3750-E StackWise Plus

W. Brooke Frischemeier, brookexx@cisco.com

Agenda



StackWise Operation

Mixing StackWise Plus and StackWise

QoS

Hardware Detail

Packet Flow Detail

Port ASIC Detail

Stack Master and Members

- A stack is created by connecting switches using Cisco proprietary Stacking Cable
- During the formation of stack, a stack master is elected
- All switches have the ability to be stack master—no special hardware/software required
- The stack master can be selected by assigning a user-configurable priority 1 through 15, 15 being the highest
- An LED indicates stack master
- The master controls all centralized functions
- All non-master switches are called members

Functions of the Stack Master

- The stack master:
  - Builds and propagates the L3 FIB
  - Propagates the configuration to the stack
  - Controls of the console
  - Controls the CDP neighbor table
- The entire stack has single VLAN database
- On stack master failure, another switch in the stack takes over
- 1:N master redundancy
- Reconvergence times tested under heavy load:
  - Layer 1 failure is detected in several microseconds
  - Layer 2 failure ~ mseconds
  - Layer 3 link failure—sub 200 mseconds
  - Layer 3 member failure—sub 300 mseconds
  - Layer 3 master failure—up to eight seconds

Criteria for Stack Master Election

- When adding switches or merging stacks, the master will be chosen based on the rules below, in the order specified
- If the first rule does not apply, the second rule is tried, and so on, until an applicable rule is found:

1. The stack (or switch) whose master has the higher user configurable mastership priority
2. The stack (or switch) whose master is not using the default configuration
3. The stack (or switch) whose master has the higher software priority

   Cryptographic advanced IP services (IPv6)

   Noncryptographic advanced IP services (IPv6)

   Cryptographic IP services

   Noncryptographic IP services

   Cryptographic IP based

   Noncryptographic IP based
4. The stack (or switch) whose master has the longest uptime
5. The stack (or switch) whose master has the lowest MAC address

Switch Numbers

- Member switches, in a stack, are assigned switch numbers
- Valid switch numbers are 1 through 9

    Numbering does not reflect physical location of the stack members

- Switch numbers are “sticky”, i.e. they switch will keep the same switch number after reboot
- The user has the ability to renumber the switch through the CLI
- The switch number can be shown by using the “STACK” LED

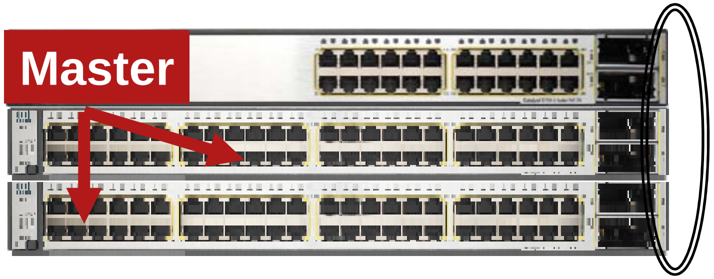
Centralized and Distributed Functions

- Centralized functions

  Those that are reside on the master node

  Those that are forwarded to the master node

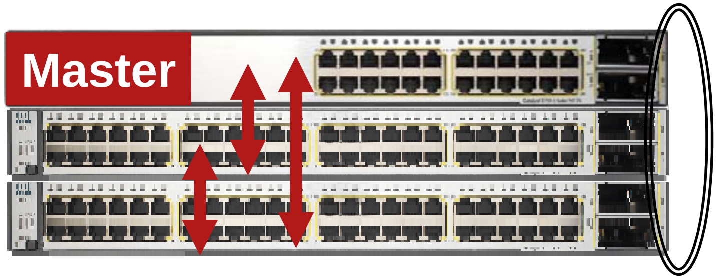
  Those that are controlled or synchronized by the master node



- Distributed functions

  Those that are performed locally by each node

  These functions are synchronized or updated between the nodes

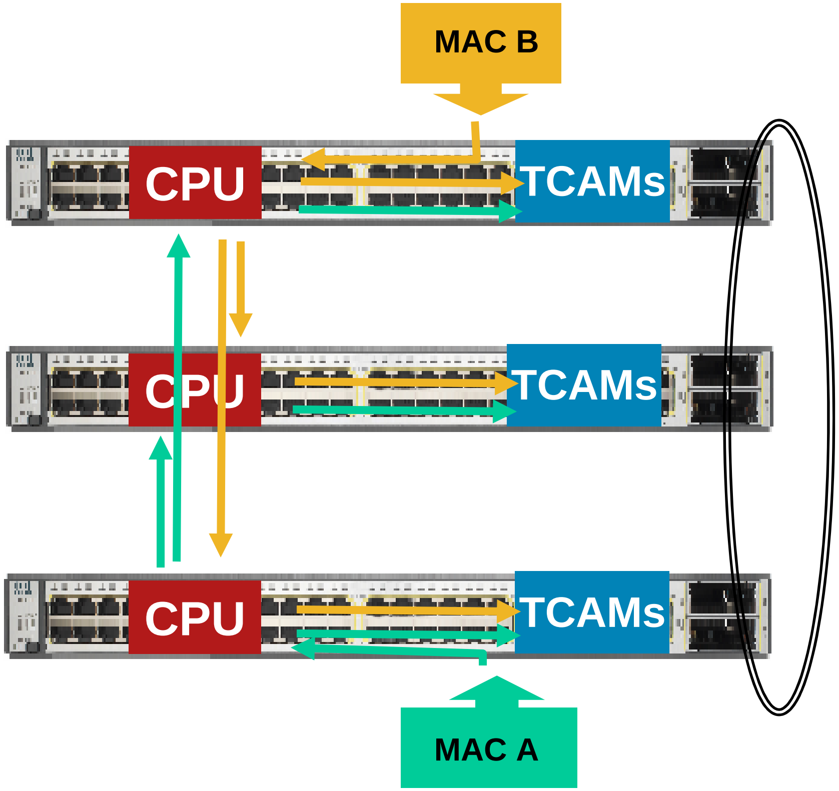


Distributed: MAC Address Management

- MAC address tables are synchronized across the stack
- How it is distributed:

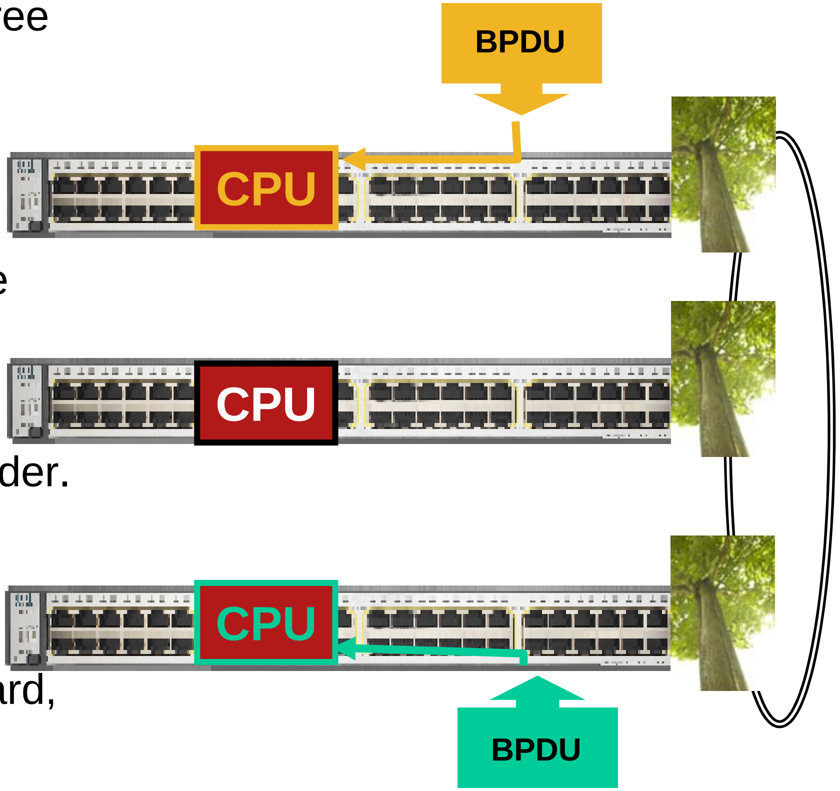
  A switch learns an address and sends a message to other switches in the stack

  Learning an address that was previously learned on a different port (either same or different switch) is considered as move



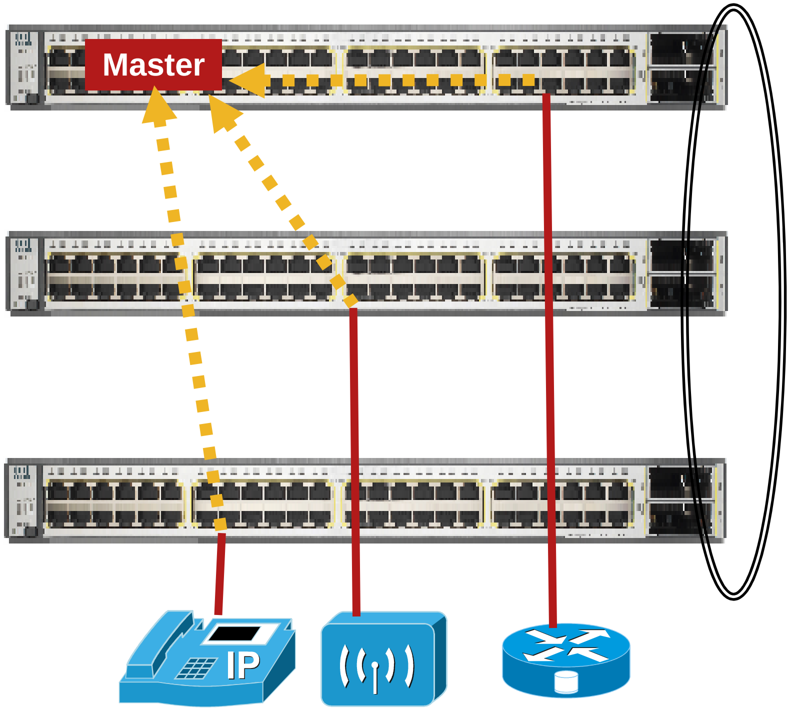
Distributed: STP

- Each switch in the stack runs its own spanning tree instance per VLAN
- Each switches will use the same bridge-id
- Each switch process its own BPDUs
- Show commands show spanning tree as a single entity
- Stacking ports are never blocked
- All packets on the ring have the internal ring header. Therefore, even broadcast packets are source stripped and do not continuously recirculate.
- Supports Cisco enhancements, like Uplink-fast, Backbone-fast, Port-fast, Root-guard, BPDU-guard, etc. are supported with no impact.
- There is support for 128 instances of STP per node/stack



Centralized: CDP

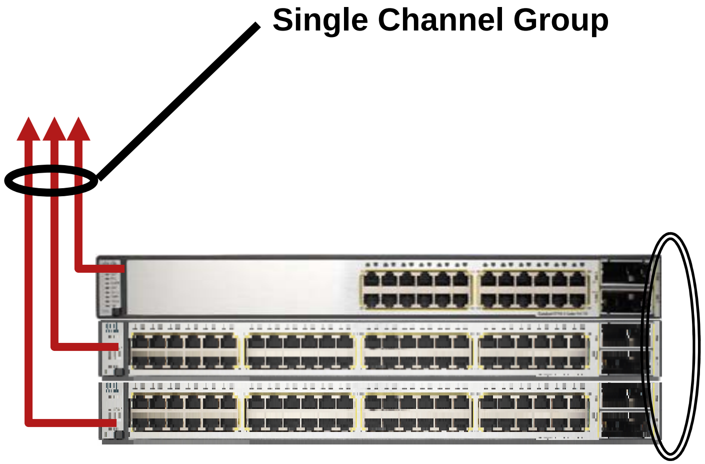
- CDP is implemented using centralized model
- The master will maintain CDP neighbor table and the neighbor tables will be empty on member nodes
- Upon a master switchover, a new master will build the CDP neighbor table



Centralized: Cross Stack Etherchannel/LACP

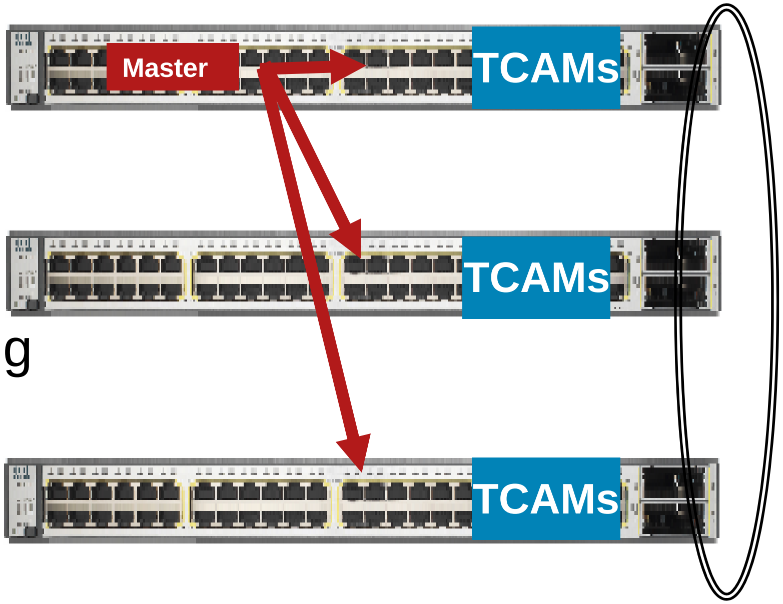
- An LACP-based Etherchannel can be formed with member ports from one or more switches in the stack
- Etherchannel control, not forwarding, is performed by the master node
- Benefits:

  In addition to port aggregation, load-balance and link redundancy and switch-level redundancy is provided



Centralized: VLAN Database

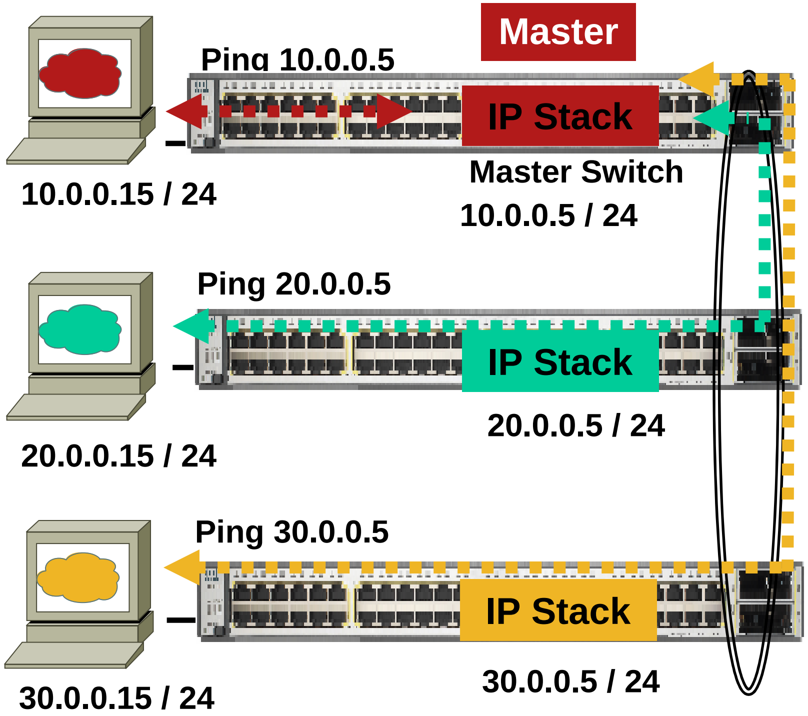
- All switches in the stack build from same VLAN database
- Members download VLAN database from master during initialization
- They are synchronized over the stack ports
- The stack supports all 3 VLAN Trunking Protocol (VTP) modes: server, client and transparent modes
- 1024 VLANs; 4K VLAN IDs are supported



Centralized: Cross Stack IP Host

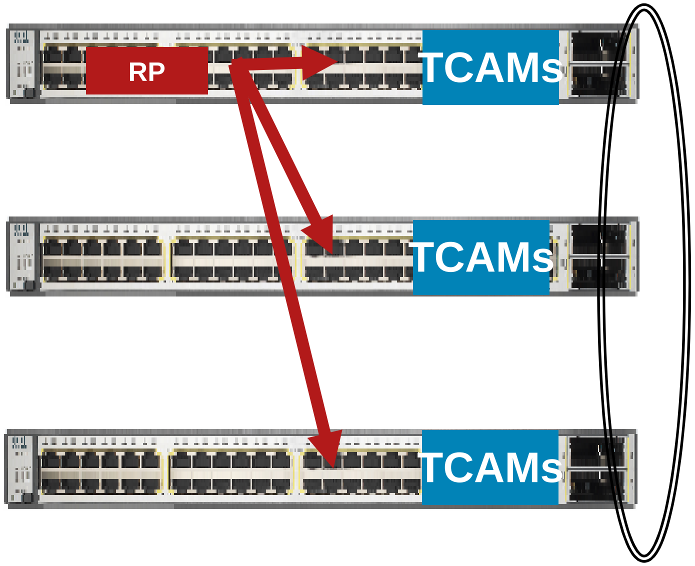
- The IP stack is active only on stack master

- All IP applications like ICMP, TFTP, FTP, HTTP, SNMP, etc are handled on the stack master irrespective of, which switch the L3 interface is connected to

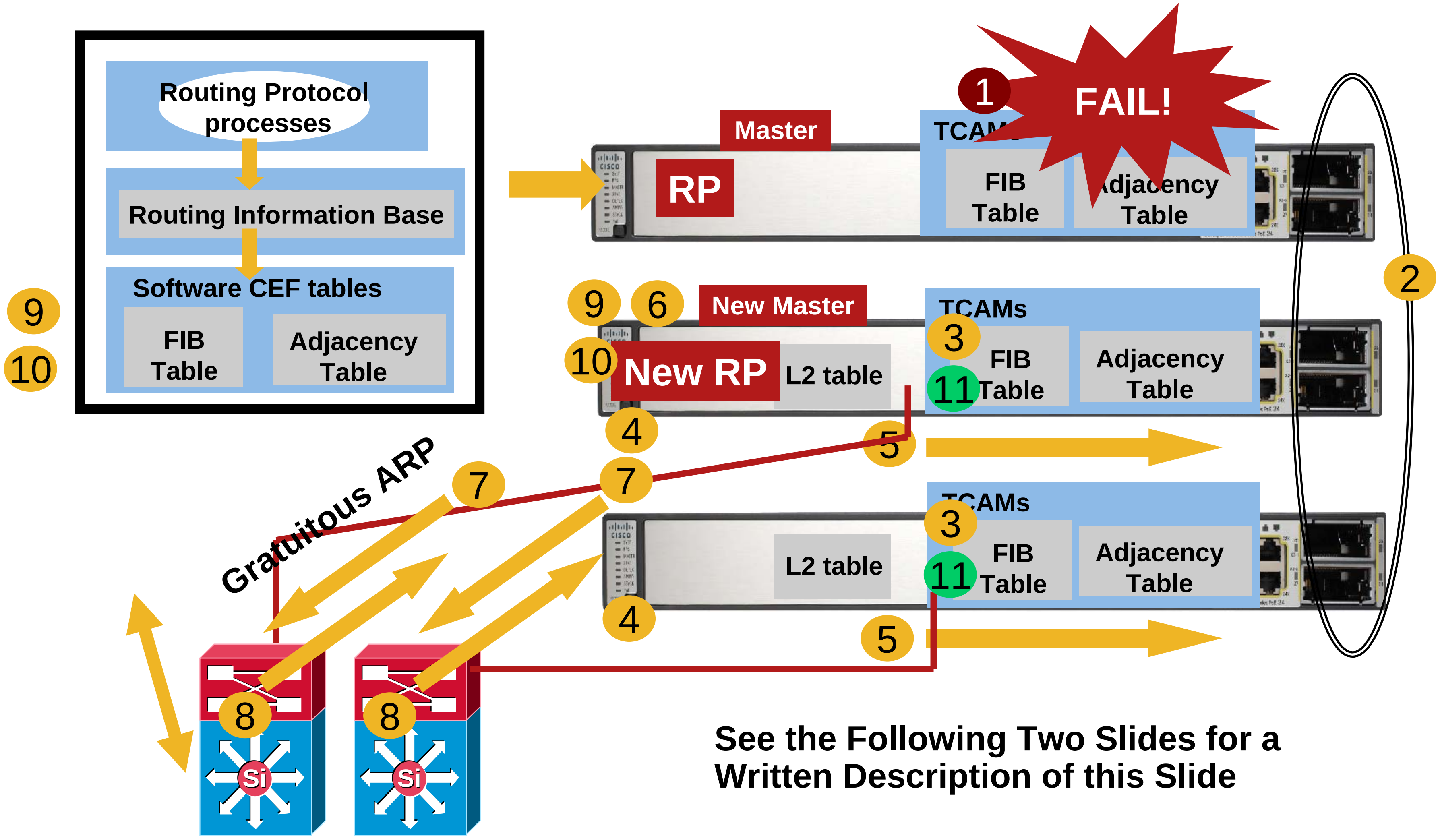


Centralized: L3 Routing Overview

- Routing protocols include Static, RIPv1, RIPv2, OSPF, IGRP, EIGRP, BGP, PIM-SM/DM, DVMRP, HSRP
- The Cisco Catalyst 3750 uses cross stack equal cost routing
- The Cisco Catalyst 3750 Stack appears as a single router to the world
- No HSRP peering among stack members, stack and external router are peers
- Policy Based Routing (PBR), IPv4 and IPv6 routing are supported in hardware
- Layer 3 link failure—sub 200 ms
- Layer 3 member failover—sub 300 ms
- Layer 3 master failover—up to eight seconds



Routing Master Failure—Recovery



See the Following Two Slides for a Written Description of this Slide

Routing Master Failure—
Recovery Event Sequence

1. The master switch fails/removed
2. The stack manager detects master removal, and performs new master election
3. All FIB/Adj marked stale and then new master RP is activated
4. All member switches join the new master
5. All this time, the switches forward traffic, using the stale FIB/Adj database
6. The new master brings up its L3 interfaces with the applied running config
7. Gratuitous ARPs sent out on each of the Up L3 interfaces, to update peers of new router MACs

Note: MAC address of all L3 interfaces are derived from the new master's MAC pool

Routing Master Failure—
Recovery Event Sequence (Cont.)

8. Peer routers/hosts continue sending traffic to the stack with the updated MAC address
9. Routing protocols if configured startup on new master and start exchanging protocol messages with the neighboring routers—thereby building their database and adjacencies
10. New routing table generated
11. Every update to the FIB/Adj, results in the stale flag being cleared, for the FIB/Adj entry updated
12. If a new member switch were to join the stack, the existing FIB/Adj database with stale entries is down loaded to them
13. After the routing protocols have converged, the only FIB/Adj entries left with stale flags are routes/next hops which no longer exist
14. After five minutes of new master election all stale routes are flushed

Configuration Management

- Master:

  Copies of the startup and running config files are kept on all members in the stack

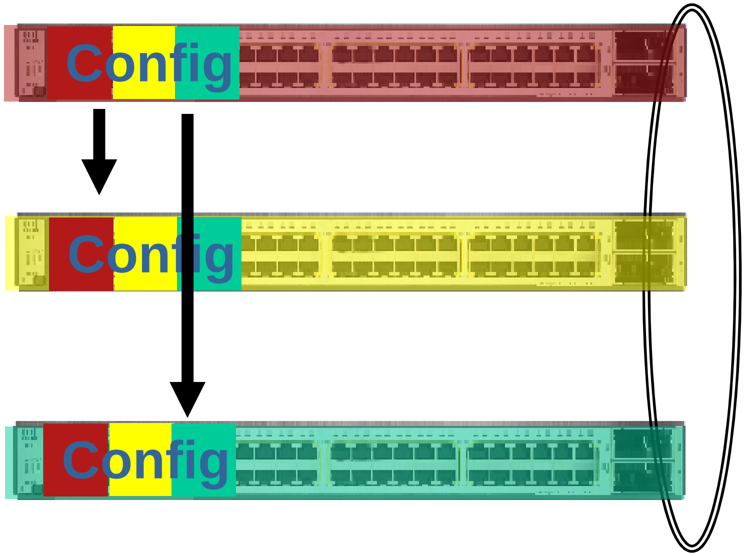
  The current running-config is synched from the master to all members

  On a switchover, the new master re-applies the running-config so that all switches are in sync

- Member:

  Keeps a copy of startup and running config at all times

  On boot-up waits for config file from master and parses it



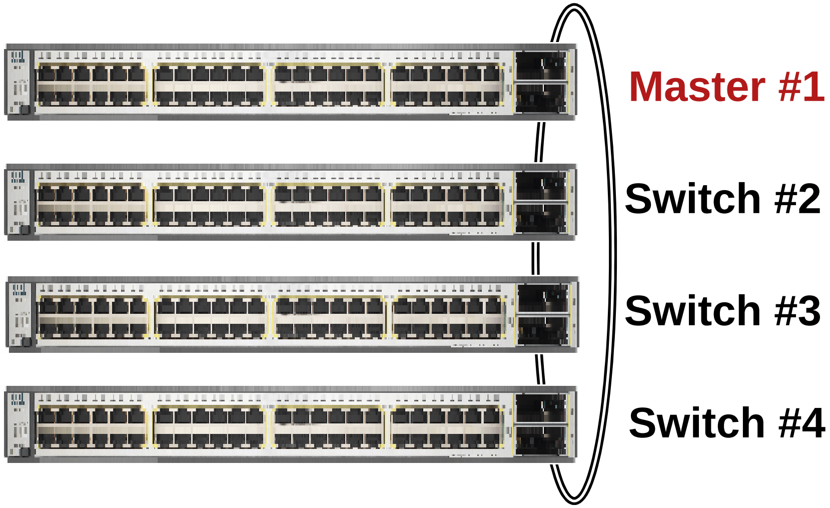
Automatic Software/Configuration Upgrade

The Master will:

- Transfer the same version of code to the new switch
- Assign the next available switch number to the switch if it does not already have one assign
- Transfer the global configuration

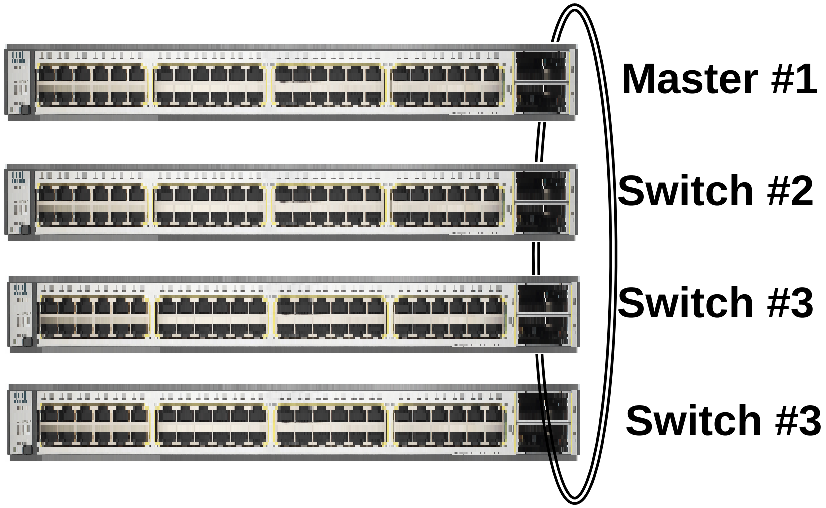
  Apply default configuration

  Apply preconfigured configuration



Switch Addition

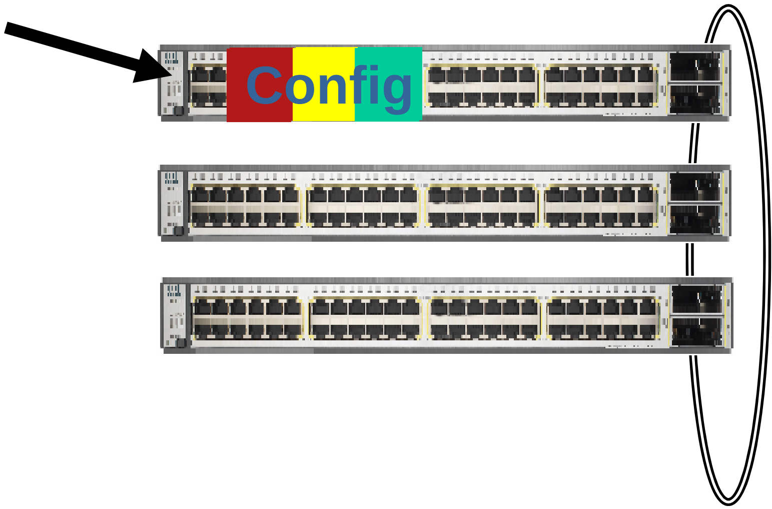
- The stack has three members—with numbers 1, 2, 3
- A new switch with an existing #3 is added to the stack
- The new switch detects a conflict, and loses
- It is assigned the #4 and reloads
- All configuration commands in the config file which apply to interfaces 4/0/* apply to the new switch



Switch Preprovisioning

Create a provision Switch #4 (Shadow).

Enter the port configuration of the New Switch.



Master #1

Switch #2

Switch #3

Set the Switch Number (#4)

Switch #4

Switch Removal

- The stack has three members—1, 2, 3
- Switch #3 is removed or powered down

  Neighbor loss is detected by Switch #1 and Switch #2

  Layer 2 and Layer 3 convergence may need to happen

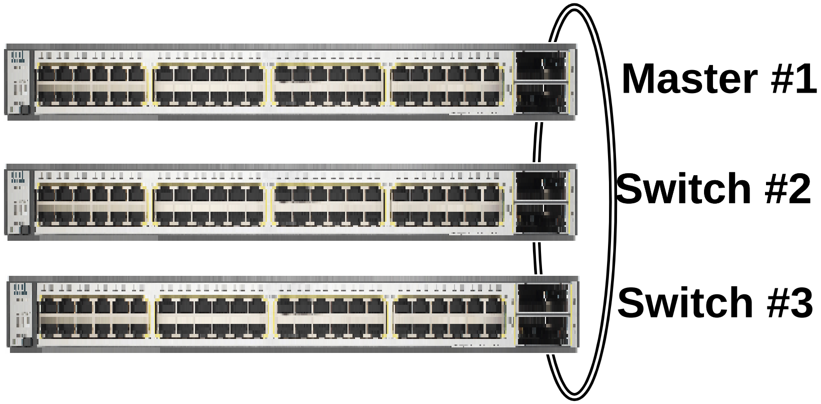
  Now there is a stack of two switches—Switch #1 and Switch #2

  Switch#1 is still the master
- Switch #1 is removed or powered down

  Switch #2 takes over as master

  Layer 2 and Layer 3 convergence may need to happen

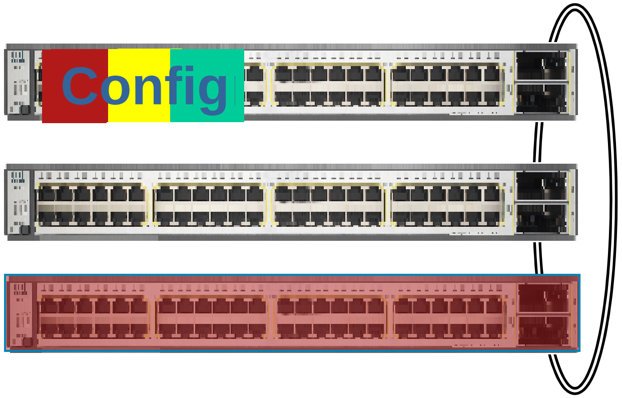
  Now there is a stack of one switch—#2 which is the master



Replacing a Switch

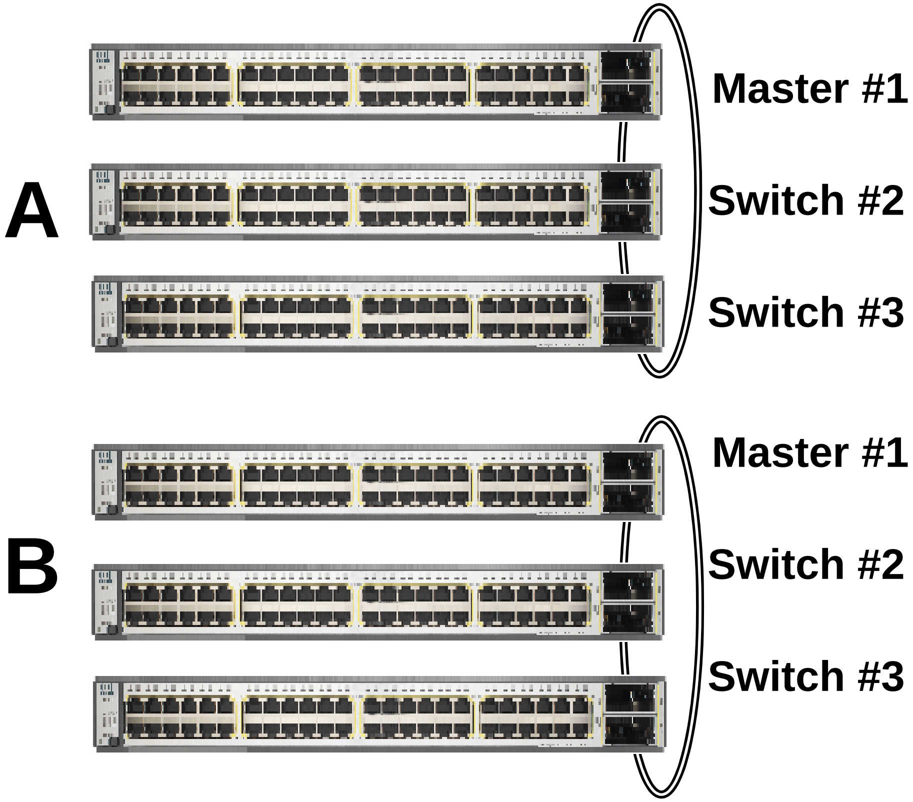
Replacing a Failed Switch:

- For example, the failed switch is a Cisco Catalyst 3750-24TS
- If replaced by another Cisco Catalyst 3750-24TS, the new switch will receive the port-level configuration of the original unit
- If replaced by a different switch, the original configuration is lost and the new switch receives all stack global configuration



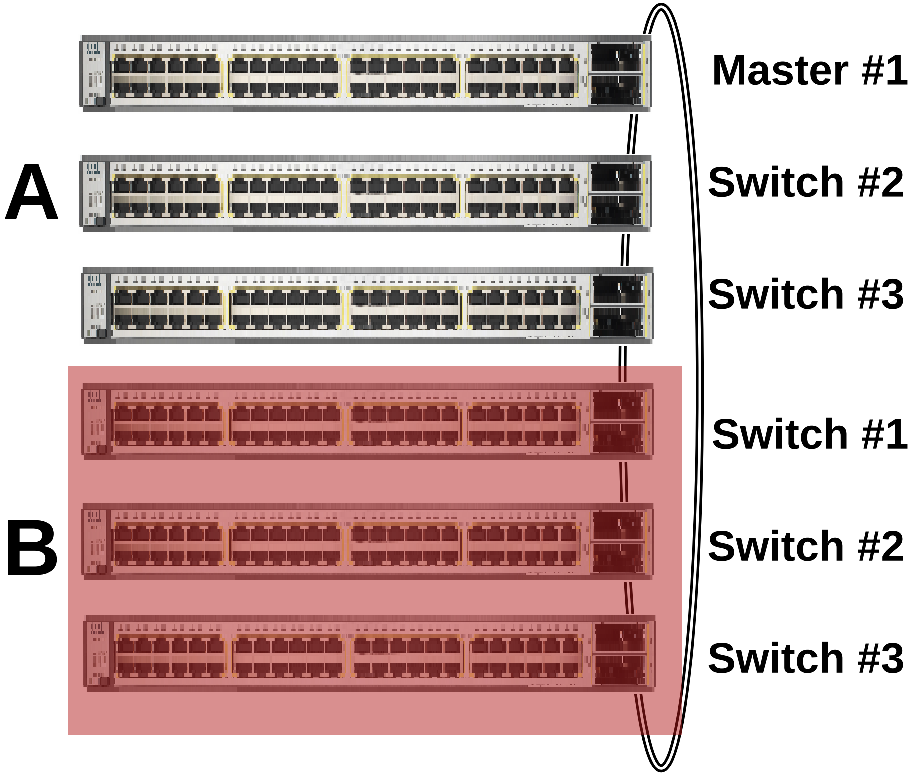
Stack Merge (Worst Case)

- Two stacks (A & B) both with switch numbers 1, 2, 3 (1 is master in both) are merged
- The stack masters go through a conflict



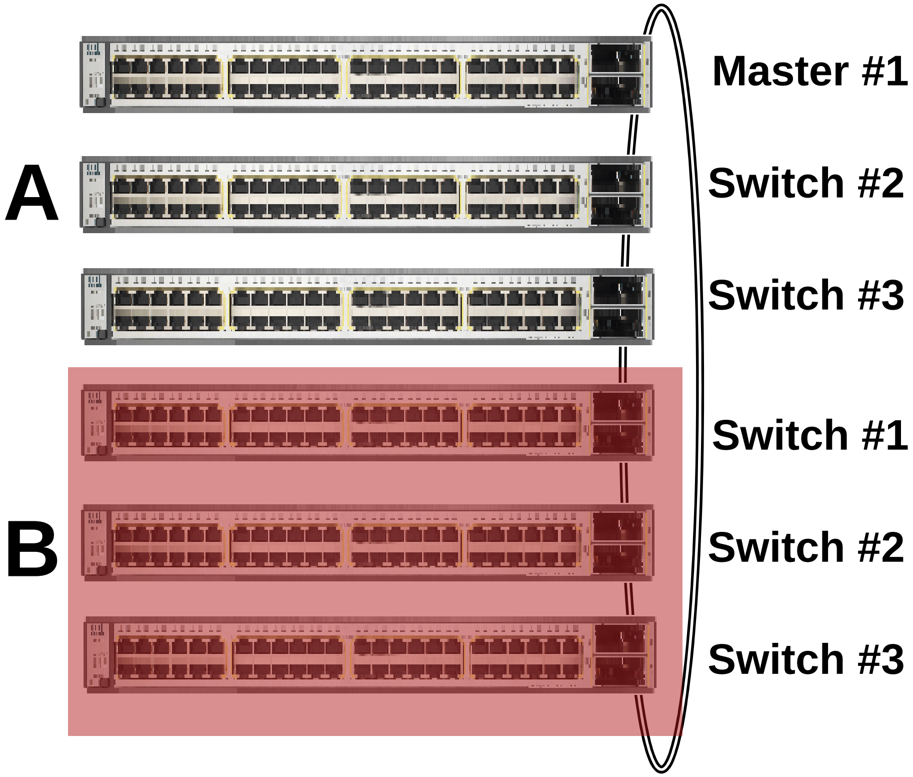
Stack Merge Cont.

- Suppose switch A1 wins the master conflict
- B1, B2, and B3 reset



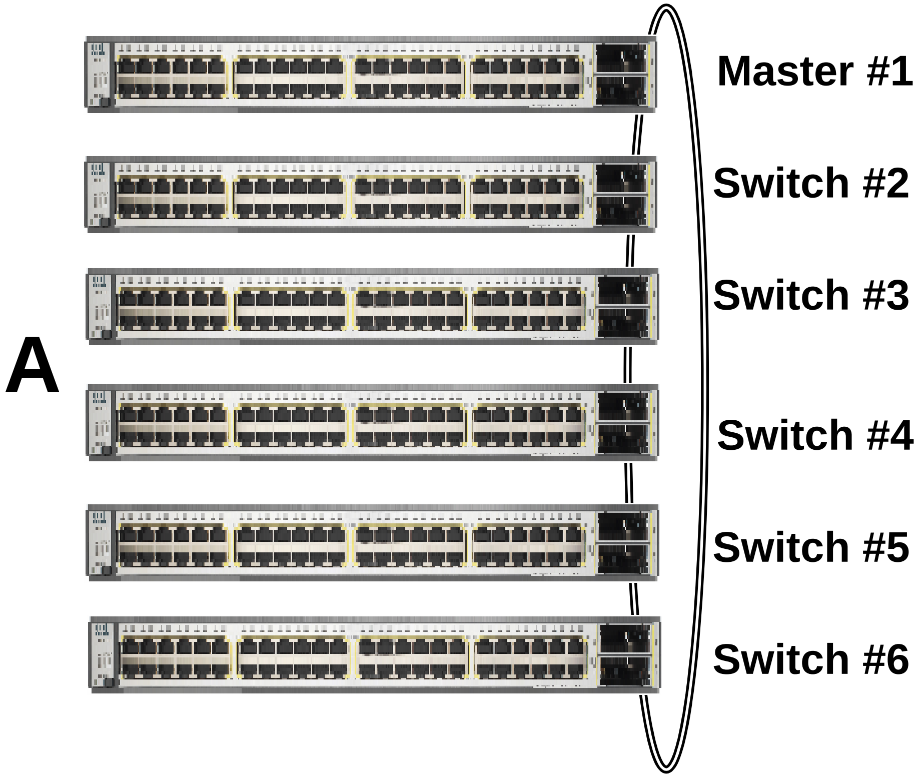
Stack Merge Cont.

- When switches B1, B2 and B3 reload, they have switch # conflicts with A1, A2 and A3
- They pick new numbers (say 4, 5 and 6)—reset

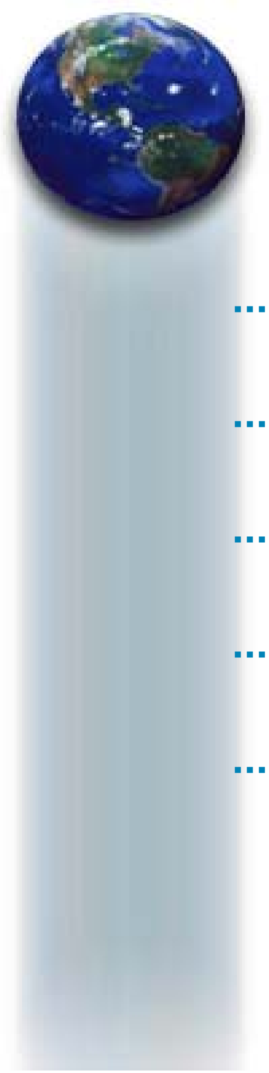


Stack Merge Cont.

- The config files on B1, B2 and B3 are rewritten with the config file from A1
- Any configuration in A1’s config file for boxes 4, 5, 6 now applies to new boxes
- Now there is one stack, stack A, with 6 boxes



Agenda



StackWise Operation

Mixing StackWise Plus and StackWise

QoS

Hardware Detail

Packet Flow Detail

Port ASIC Detail

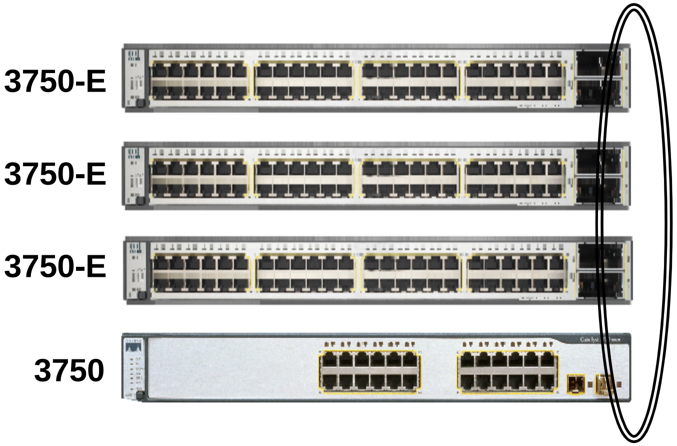
Differences Between StackWise Plus and StackWise

- StackWise Plus increases the effective throughput of StackWise beyond 32Gbps to over 64Gbps using spatial reuse
- StackWise Plus enables local switching so that traffic local to a switch does not traverse the stack
- StackWise Plus is compatible with StackWise, protecting customers investments in the Catalyst 3750 Series.
- Mixed StackWise and StackWise Plus stacks will autonegotiate and self configure

Mixed-Hardware Stack

Backward Compatibility

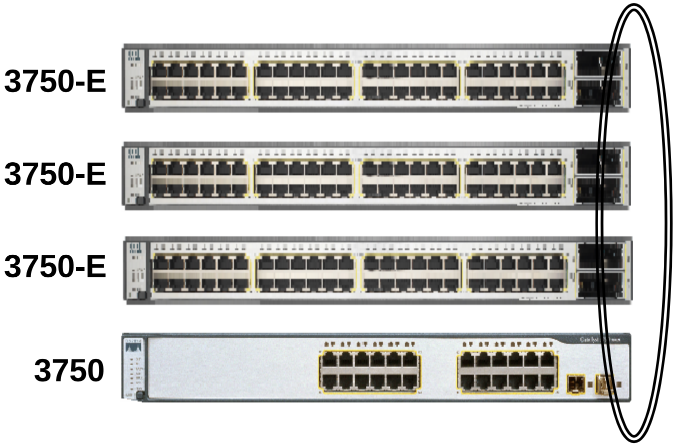
- The Catalyst 3750-E can be stacked with the Catalyst 3750
- The feature compatibility manger checks if a feature being configured should be rejected due to hardware incompatibility
- Catalyst 3750 can not join the stack with a Catalyst 3750-E when
  - It does not support the features already configured in running Configuration file
  - A feature being configured is rejected due to hardware incompatibility



Mixed-Hardware Stack

Backward Compatibility Cont'd.

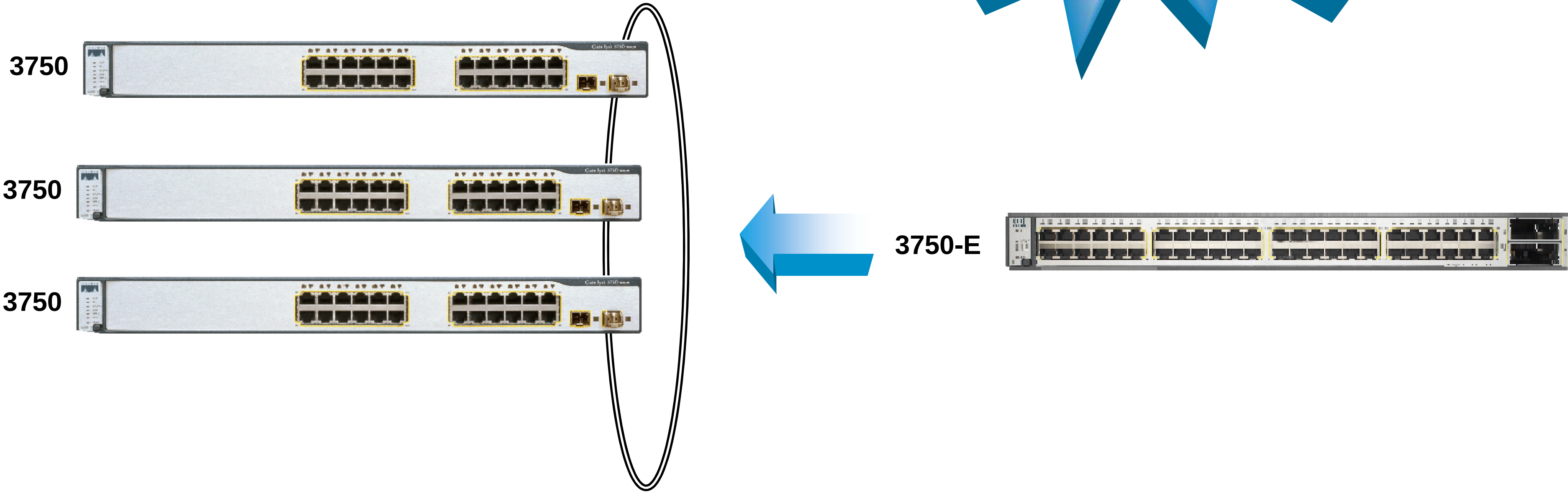
- A 3750-E port level feature does not affect processing on any Catalyst 3750 switch in the stack, but system level interdependent feature does
- A "Feature mismatch" state will occur, if at least one interdependent feature is configured in the exiting stack which is not supported by the new Catalyst 3750
- A Feature mismatch is calculated based on hardware version , number of interfaces, and the running features



3750-E Addition to 3750 Stack

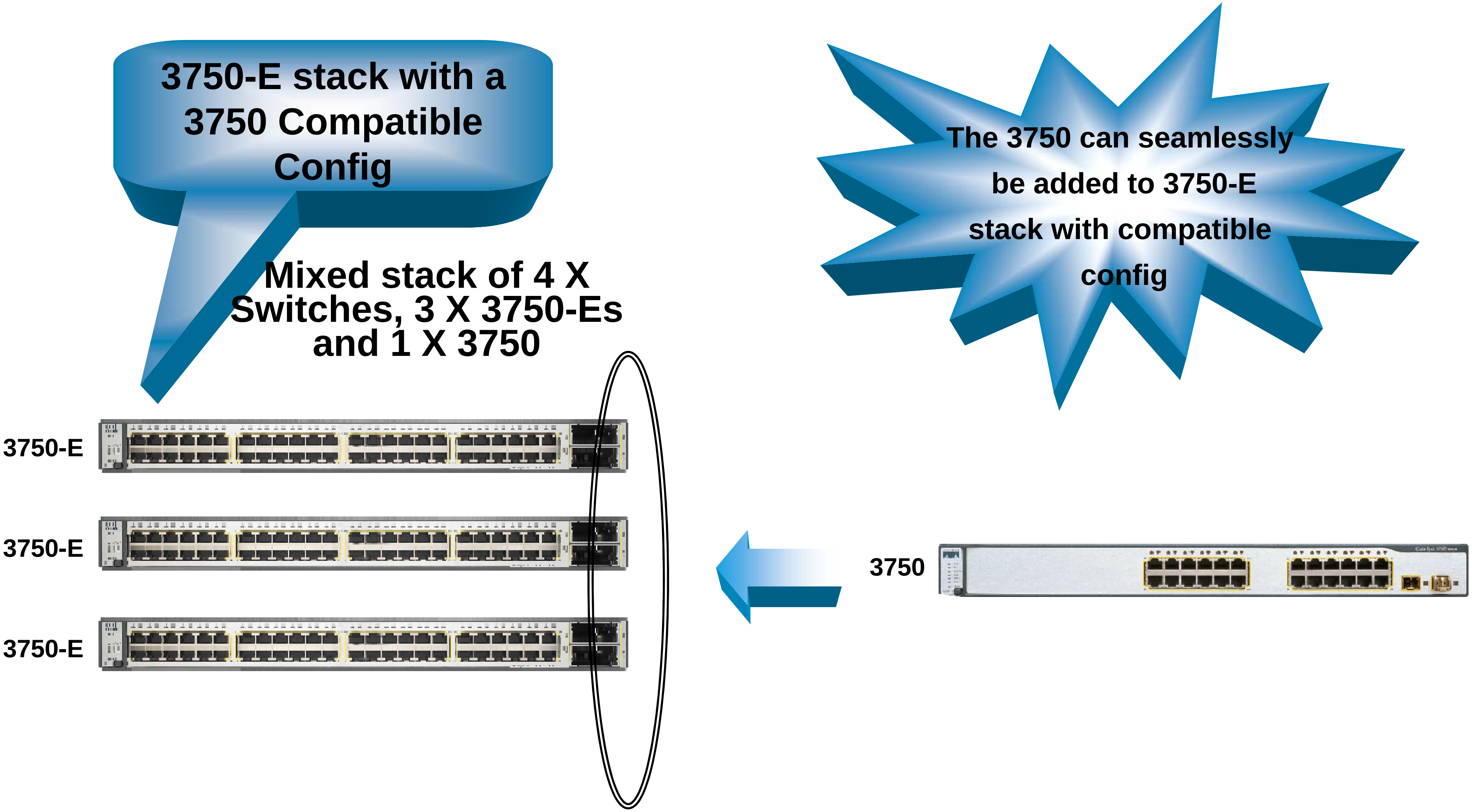
The 3750-E can be added to an existing 3750 stack seamlessly

Mixed stack of 4 X Switches, 3 X 3750s and 1 X 3750-E



3750 Addition to a 3750-E Stack

Compatible Configuration



3750-E stack with a 3750 Compatible Config

Mixed stack of 4 X Switches, 3 X 3750-Es and 1 X 3750

The 3750 can seamlessly be added to 3750-E stack with compatible config

3750-E

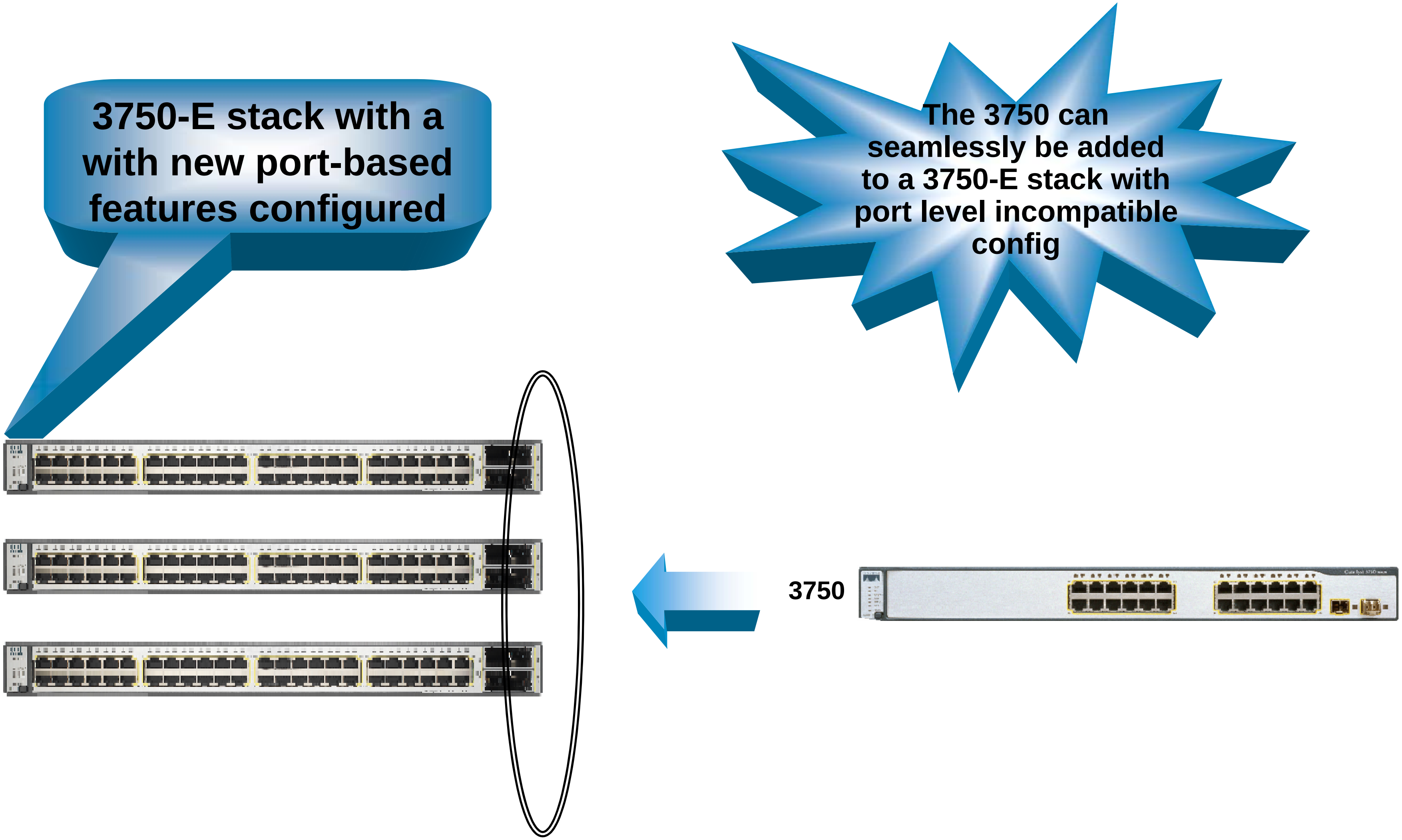
3750-E

3750-E

3750

3750 Addition to a 3750-E

With New 3750-E Port Features Enabled



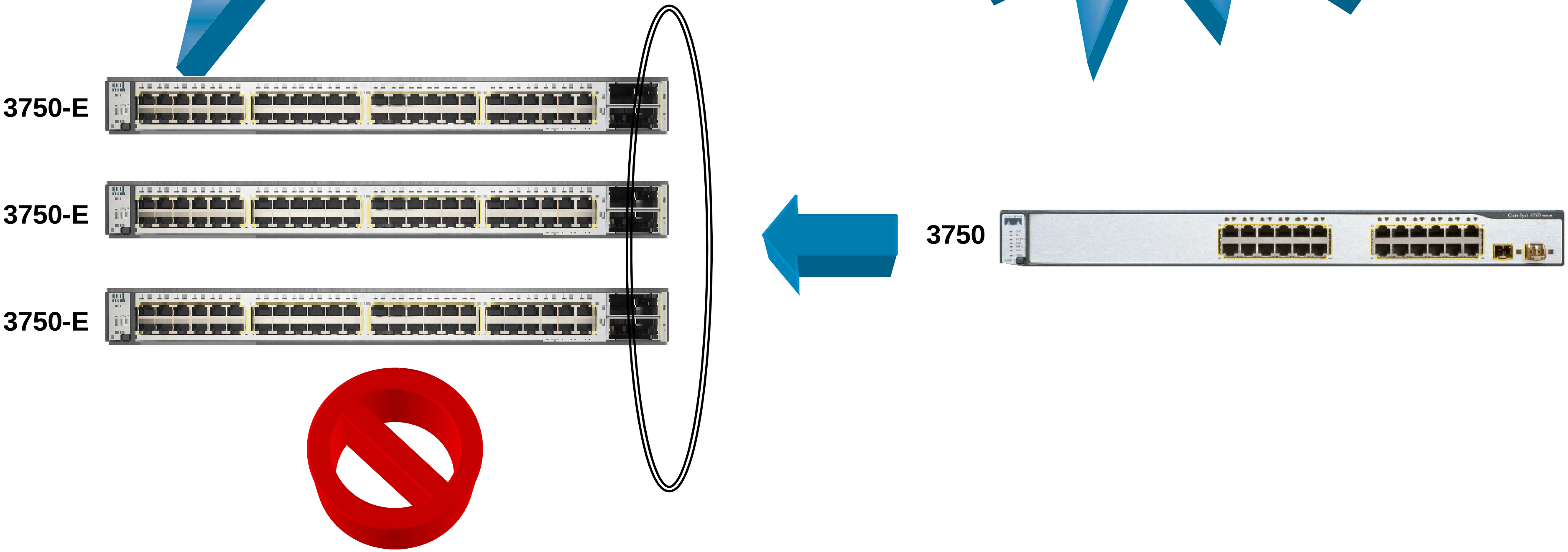
3750-E stack with a with new port-based features configured
The 3750 can seamlessly be added to a 3750-E stack with port level incompatible config
3750-E
3750-E
3750-E
3750

3750 Addition to a 3750-E

With New 3750-E Interdependant Features Enabled

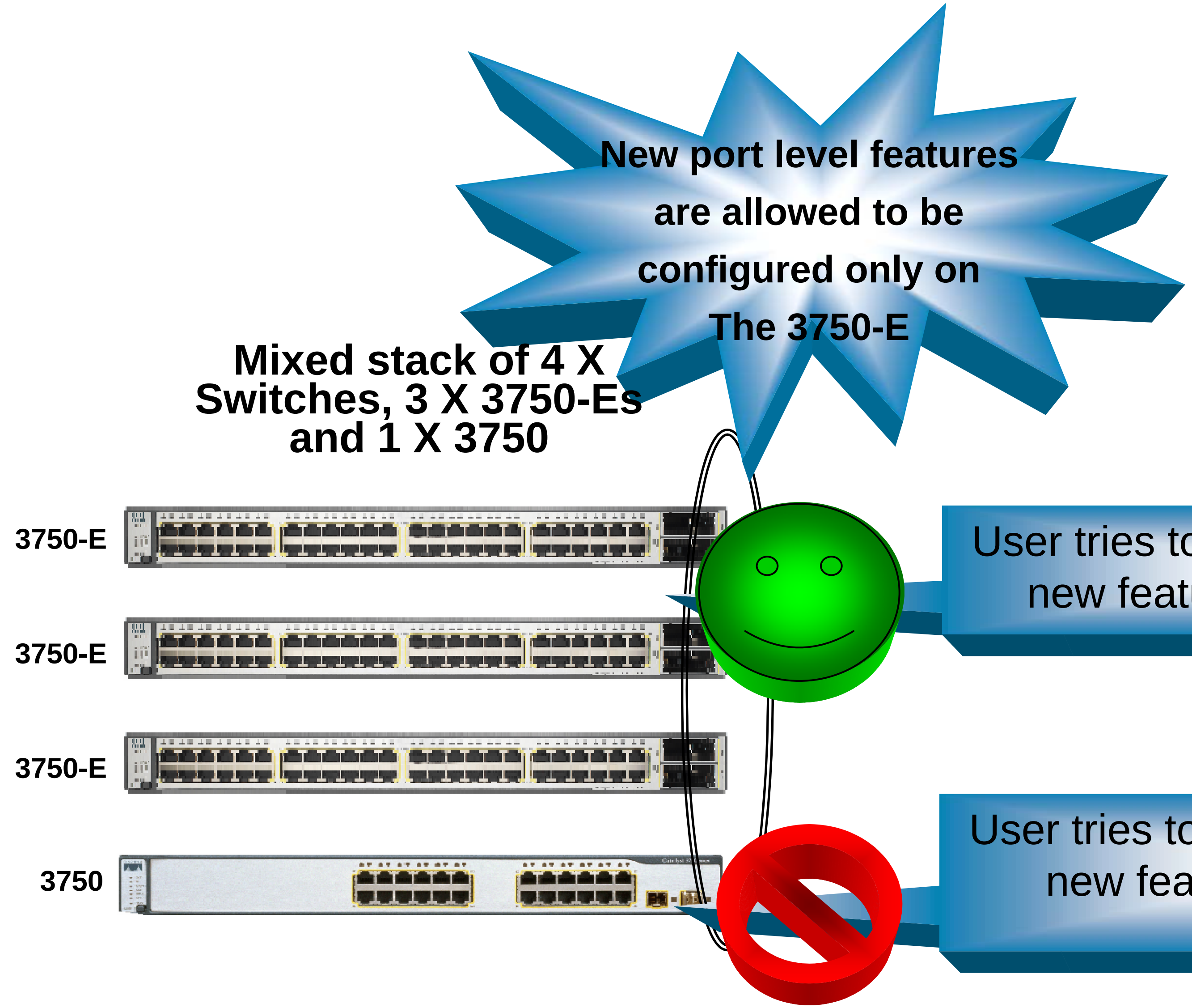
3750-E stack with a with new inter-dependant features

The 3750 is placed in feature mismatch mode and not allowed to stack with 3750-E stack



Mixed Stack: Incompatible Port Level

Independent Feature Configuration



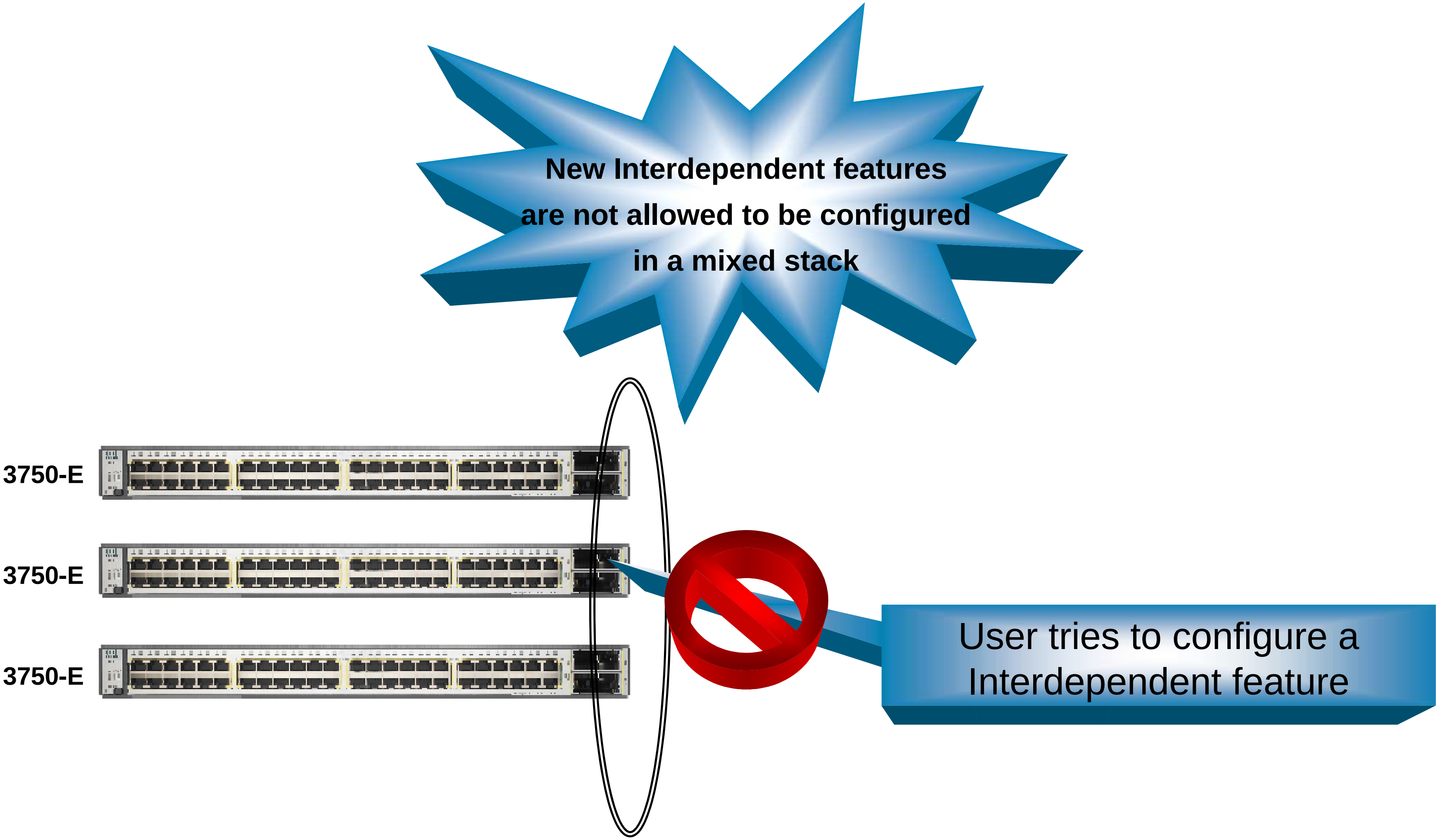
Mixed stack of 4 X Switches, 3 X 3750-Es and 1 X 3750

New port level features are allowed to be configured only on The 3750-E

User tries to configure a port based new feature on a 3750-E Port

User tries to configure a port based new feature on a 3750 Port

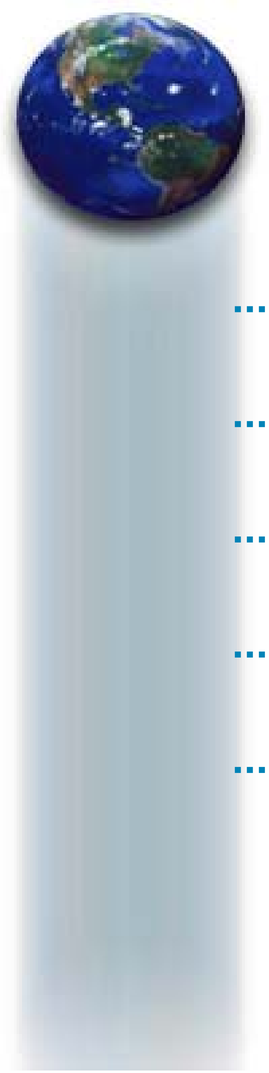
Mixed Stack: Incompatible Interdependent
Feature Configuration



New Interdependent features
are not allowed to be configured
in a mixed stack

User tries to configure a
Interdependent feature

Agenda



StackWise Operation

Mixing StackWise Plus and StackWise

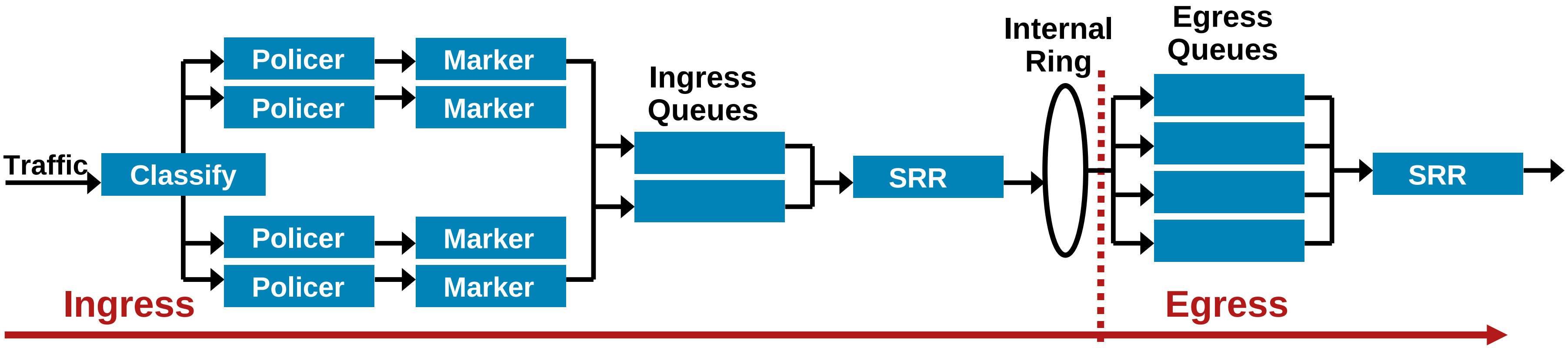
QoS

Hardware Detail

Packet Flow Detail

Port ASIC Detail

Cisco Catalyst 3750 QoS Model



Classification

- Inspect incoming packets
- Based on ACLs or configuration, determine classification label

Policing

- Ensure conformance to a specified rate
- On an aggregate or individual flow basis
- Up to 256 policers per Port ASIC
- Support for rate and burst

Marking

- Act on policer decision
- Reclass or drop out-of-profile

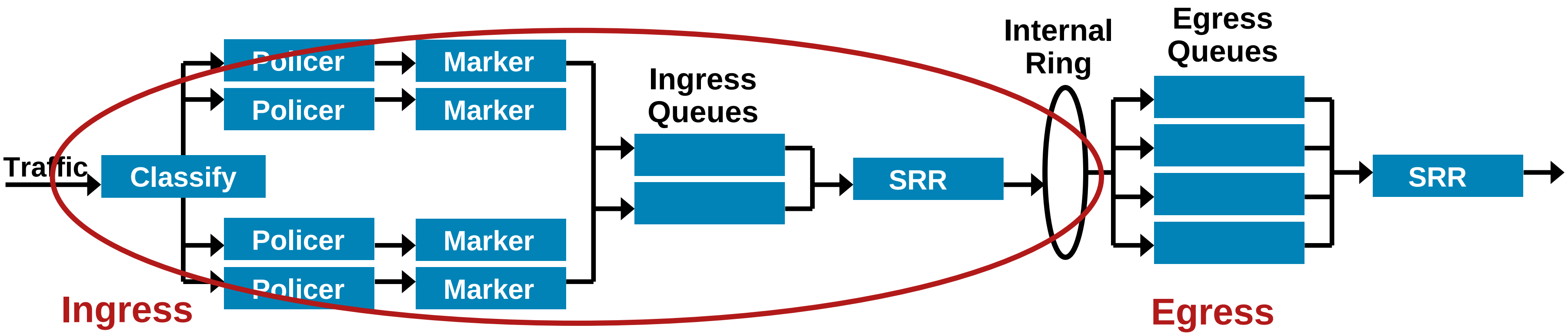
Ingress Queue/ Schedule Congestion Control

- Two queues/port ASIC shared servicing
- One queue is configurable for strict priority servicing
- WTD for congestion control (three thresholds per queue)
- SRR is performed

Egress Queue/ Schedule Congestion Control

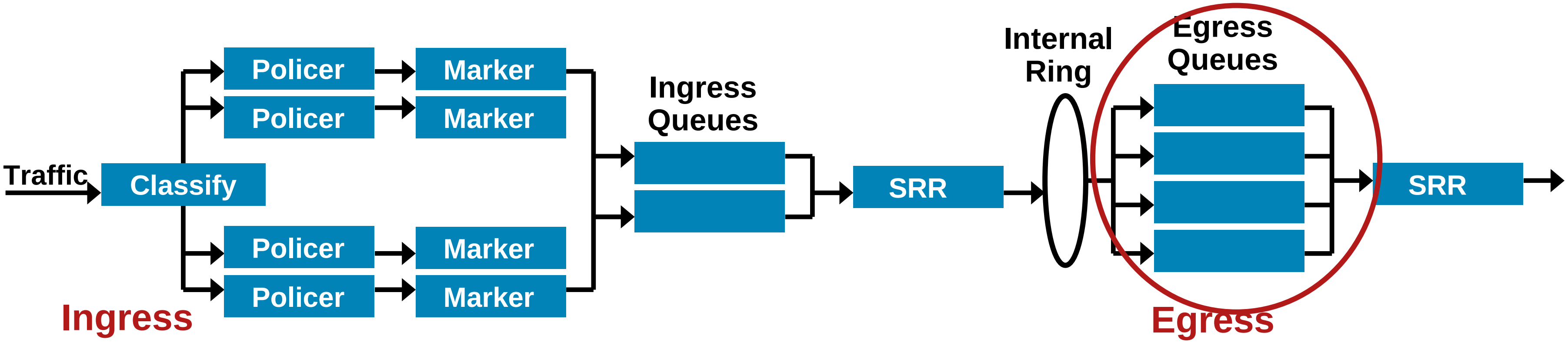
- Four SRR queues/port shared or shaped servicing
- One queue is configurable for strict priority servicing
- WTD for congestion control (three thresholds per queue)
- Egress queue shaping
- Egress port rate limiting

Ingress Policing and Queuing



- The Cisco Catalyst 3750 has two ingress queues, one of which can be configured to be a priority queue
- Ingress policing can be configured (DSCP, ToS, ACL, etc.)
- This can insure that, high priority and latency sensitive traffic is unimpeded when it is added to the ring
- These ingress queues, perform SRR in shared mode only
- Weighted Tail Drop (WTD) can also be performed

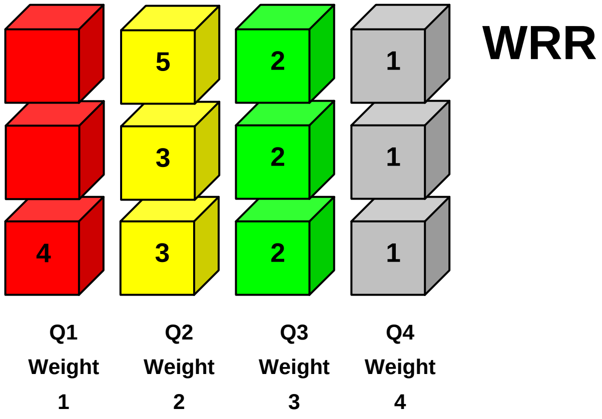
Egress Queuing

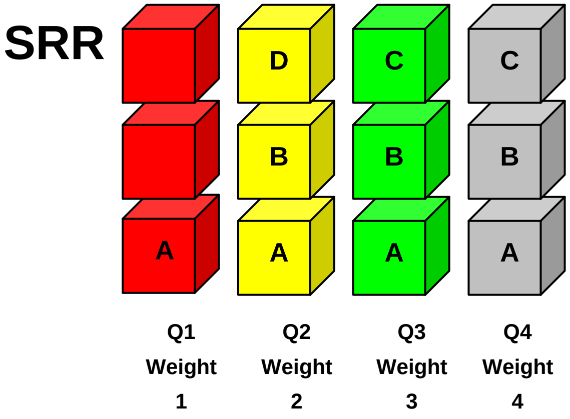


- The Cisco Catalyst 3750 has four egress queues, one of which is a priority queue
- Port-based bandwidth limiting can be configured from 10% to 90%
- These ingress queues, perform SRR in queue sharing and queue shaping mode
- Weighted Tail Drop (WTD) can also be performed

WRR vs. SRR

SRR is an evolution of WRR that protects against overwhelming buffers with huge bursts of traffic by using a smoother round-robin mechanism

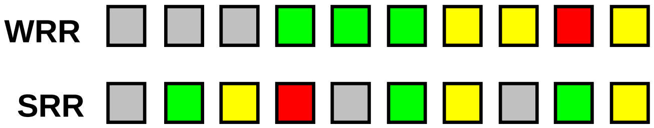




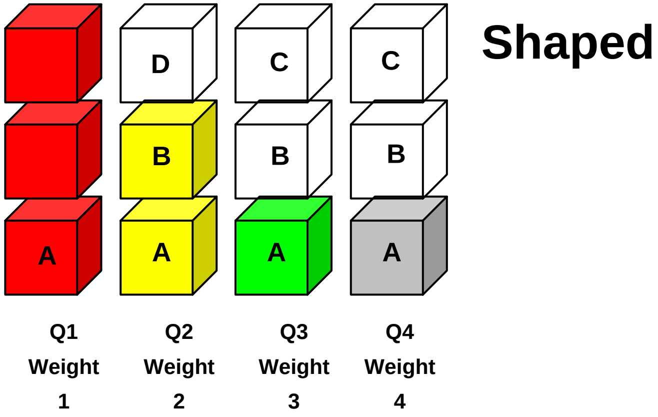
Each queue empties immediately as it is weighted

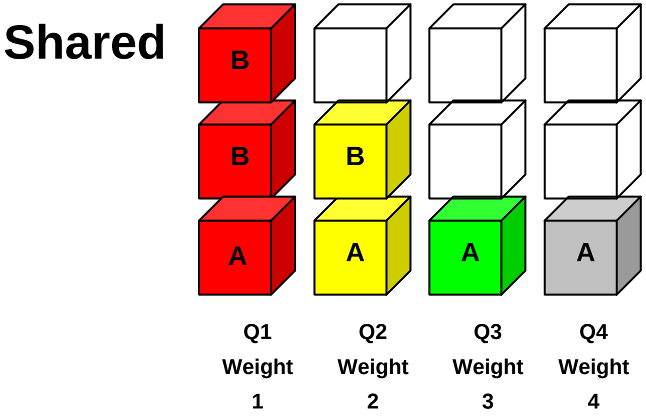
Each queue empties a weighted number of packets over a given period of time

Packet Order



Shaped SRR vs. Shared SRR





Lesser weight queues sit idle and wait to transmit, even if higher weight queues are empty

If higher weight queues are empty, lesser weight queues can continue to send while the higher weight queues are empty

Packet Order

SRR Non-shared ⬜ 🟩 🟨 🟥 **Wait** 🟨 **Wait** 🟥 **Wait** 🟥

SRR Shared ⬜ 🟩 🟨 🟥 🟨 🟥 🟥 **Room for more traffic, draining the buffers!**

Shared Queuing drains queues more efficiently!

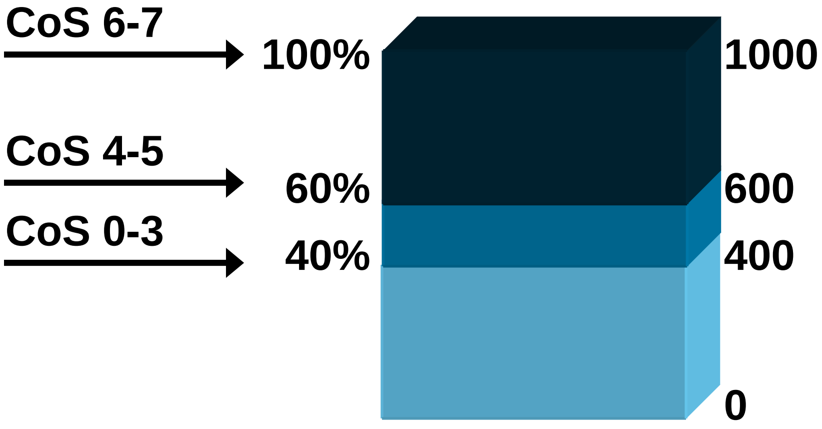
Note: WRR is not a shared servicing scheme and has gaps, as does Shaped SRR, however WRR still lacks SRRs even flows

Shaped SRR vs. Shared SRR and Traffic Shaping

- Neither Shaped SRR or Shared SRR is better
- Shared SRR is used when one wants to get the maximum efficiency out of a queuing system, because unused time slots can be reused by queues with excess traffic. This is not possible in a standard WRR.
- Shaped SRR is used when one wants to shape a queue or set a hard limit on how much bandwidth a queue can use
- When one uses Shaped SRR one can shape queues within a ports overall shaped rate

Cisco Catalyst 3750 Weighted Tail Drop

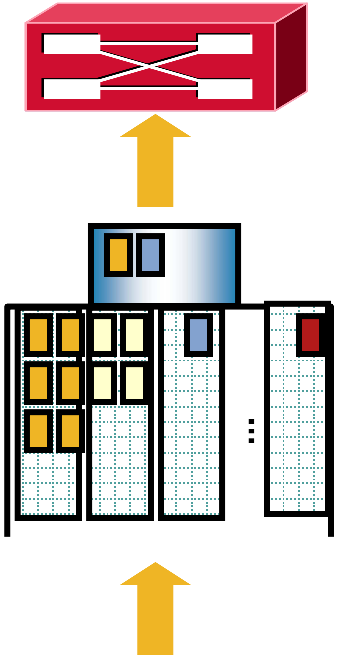
- WTD is a congestion-avoidance mechanism for managing the queue lengths and providing drop precedences for different traffic classifications
- WTD can be performed at either the Ingress Ring queues or the Egress queues
- User configurable thresholds determine when to drop certain types of packets
- As a queue fills up, lower priority packets are dropped first
- In this example, when the queue is 60% full, arriving packets marked with CoS 0-5 are dropped



One Is Displayed. All Queues Can Be Configured Independently

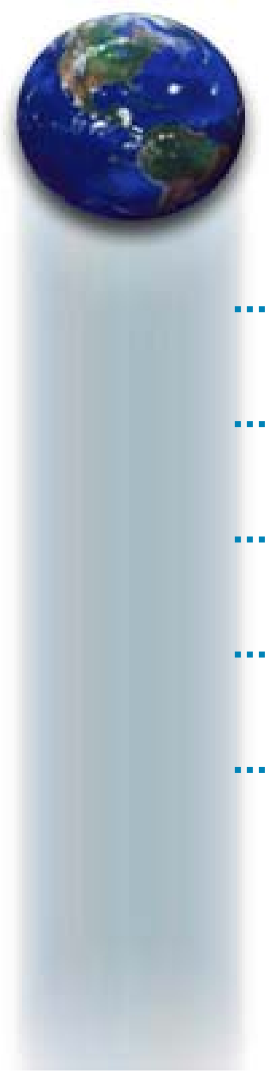
Catalyst 3750 Control Plane Protection
16 Processor Hardware Queues

- Each CPU has 16 queues for better traffic management.
- The workload is distributed to processors on each switch of the stack.
- The stack ring reserves bandwidth for priority traffic.
  - Bandwidth reservations on the ring ensure the CPU communication is not affected by data traffic.
- These 16 processor queues are not configurable.
  - STP, OSPF & inter-CPU packets on separate Queues



Traffic to the CPU

Agenda



StackWise Operation

Mixing StackWise Plus and StackWise

QoS

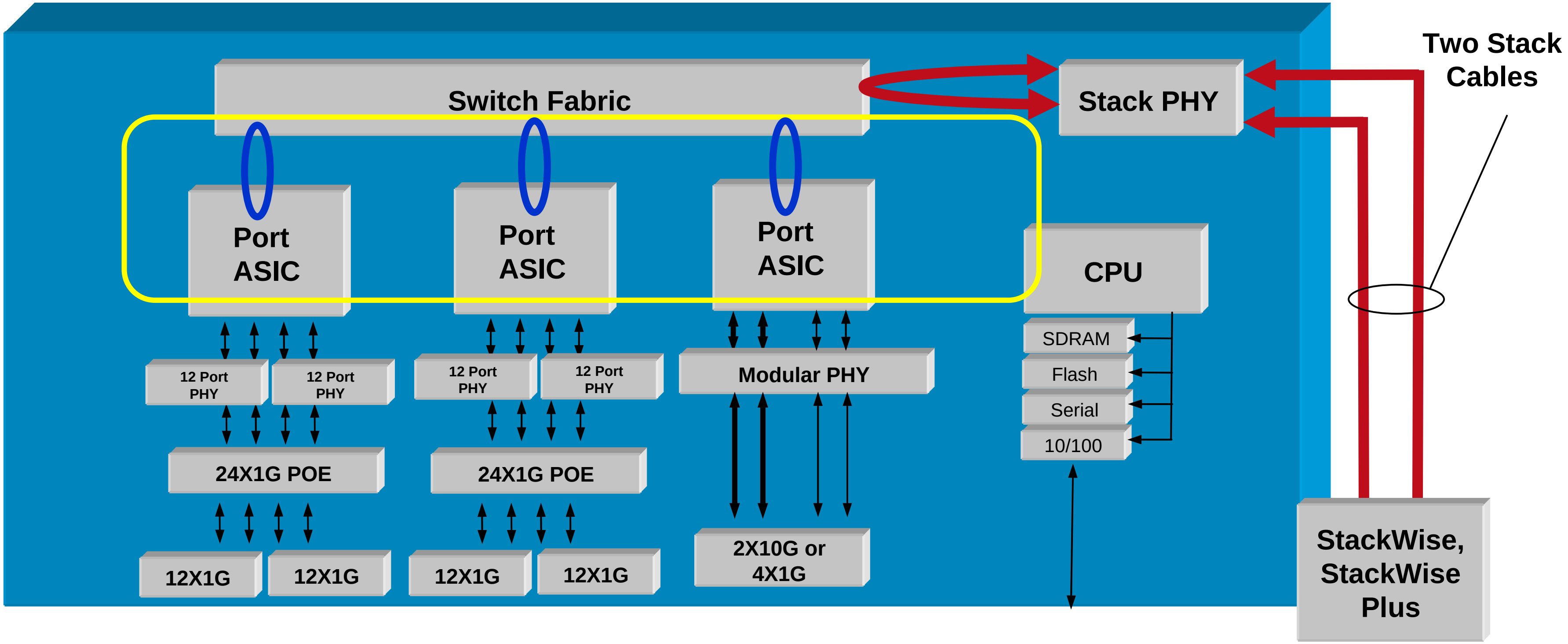
Hardware Detail

Packet Flow Detail

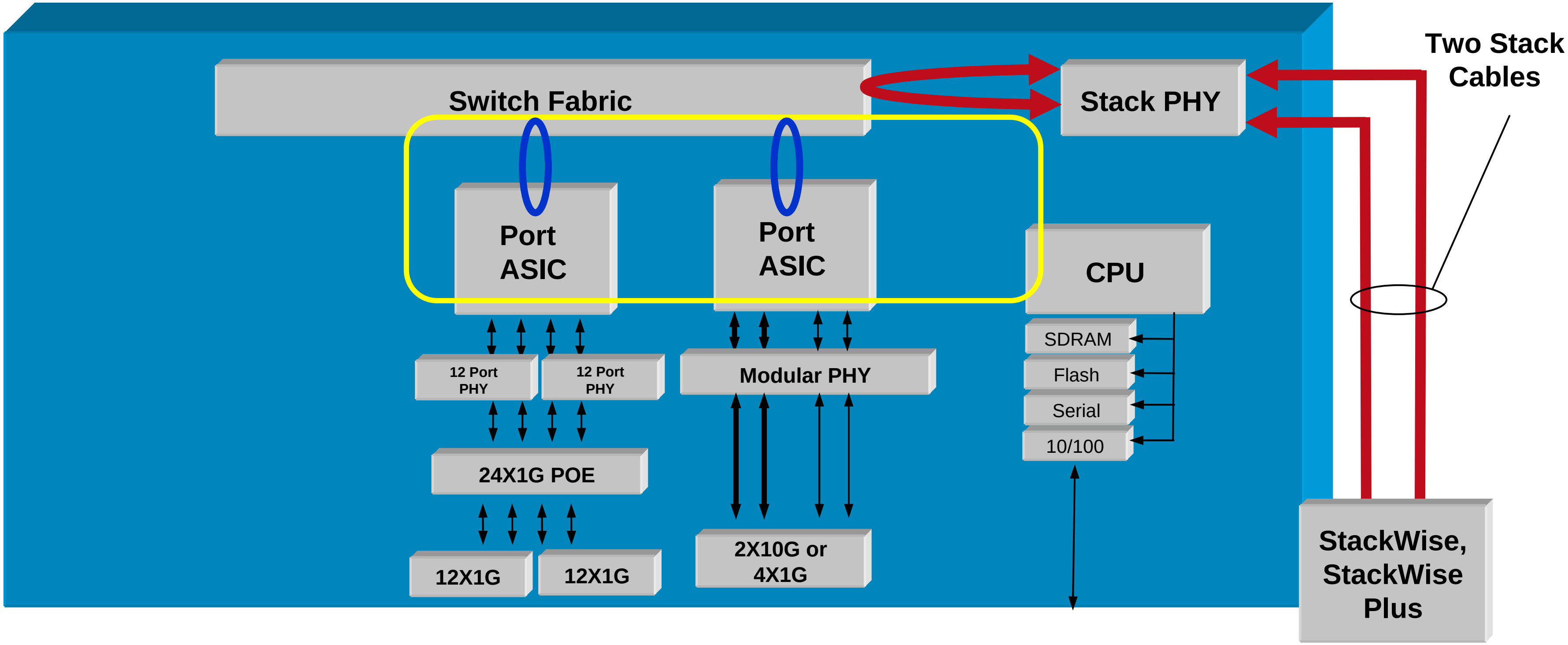
Port ASIC Detail

Catalyst 3750-E Hardware Block Diagram

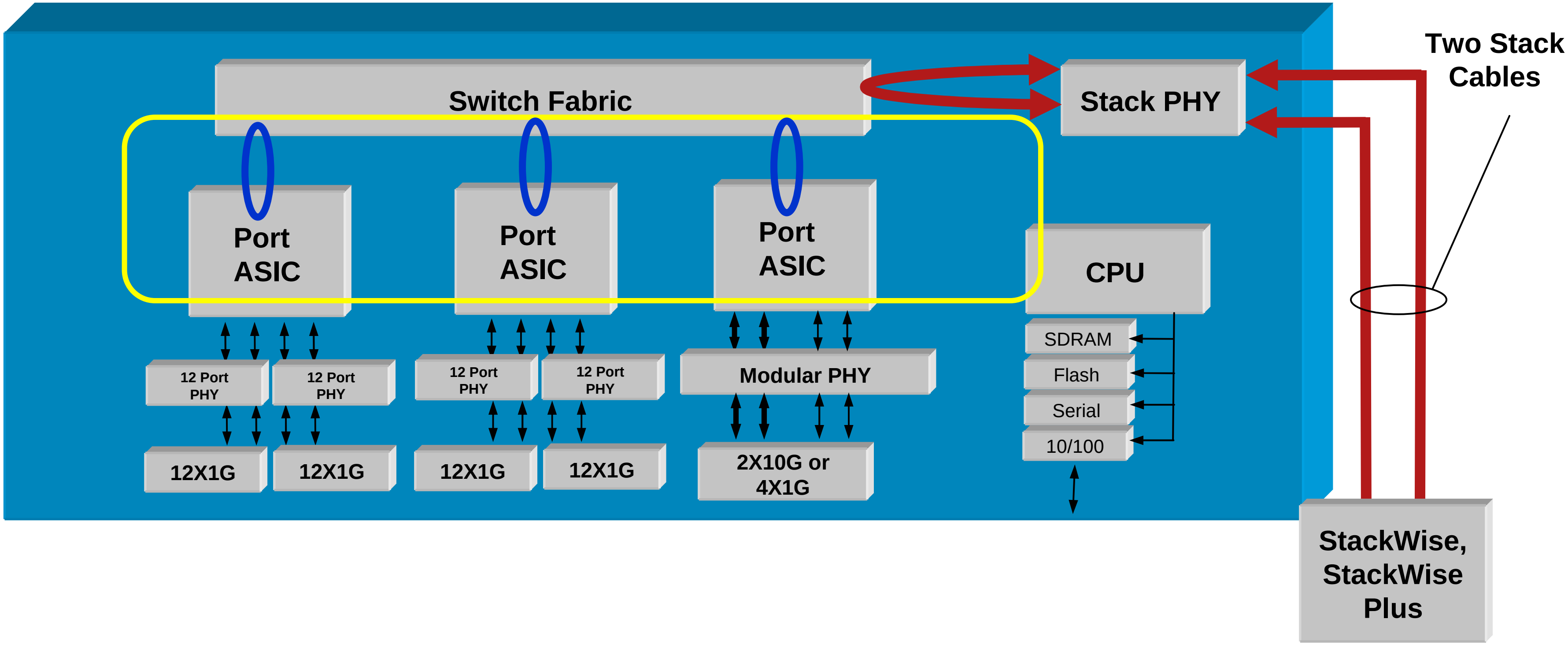
48port POE



Catalyst 3750-E Hardware Block Diagram
24port POE

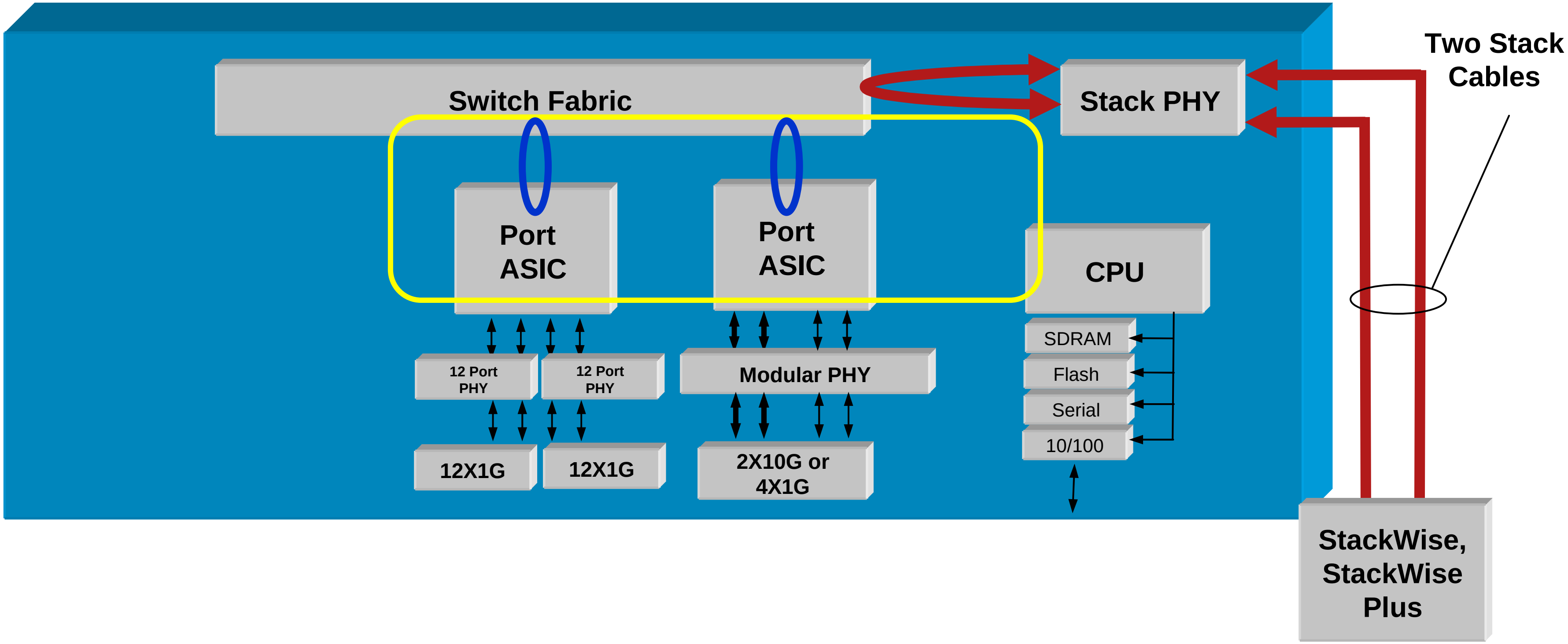


Catalyst 3750-E Hardware Block Diagram
48 port

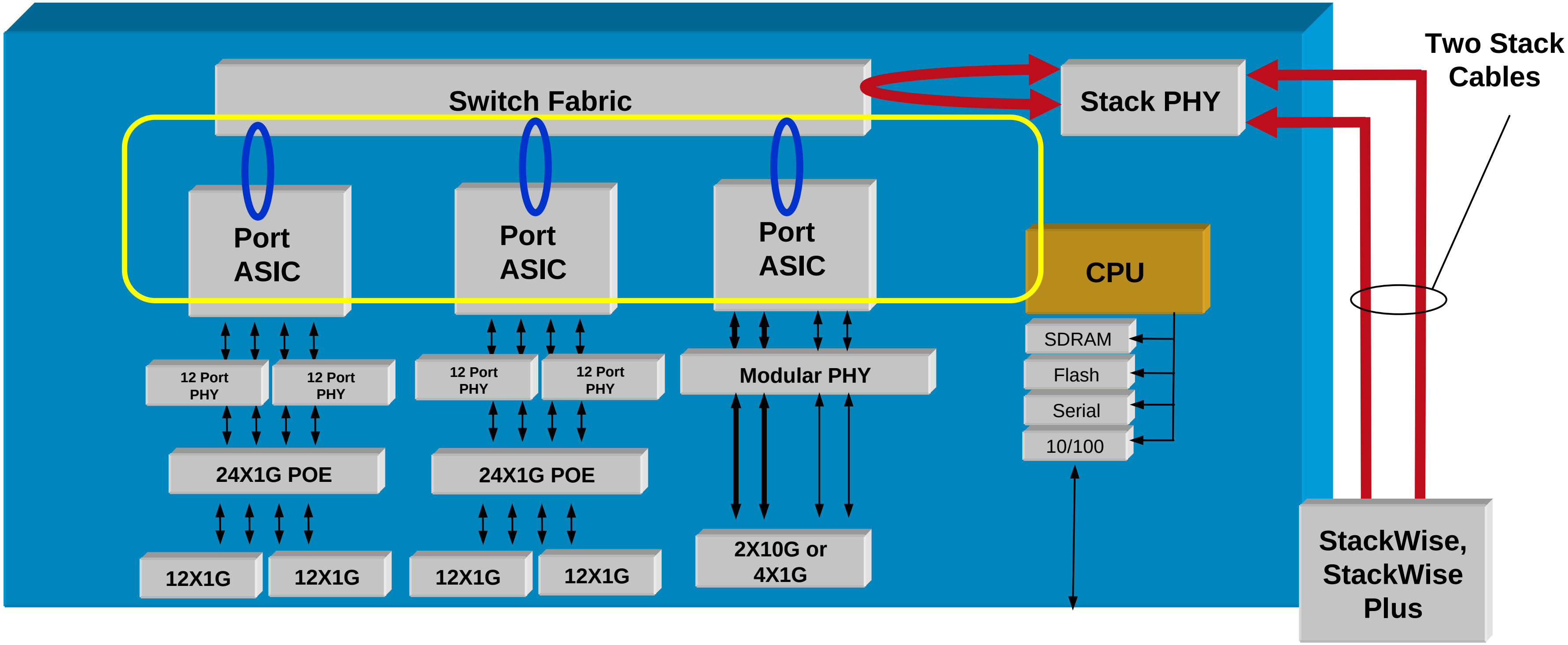


Catalyst 3750-E Hardware Block Diagram

24 port

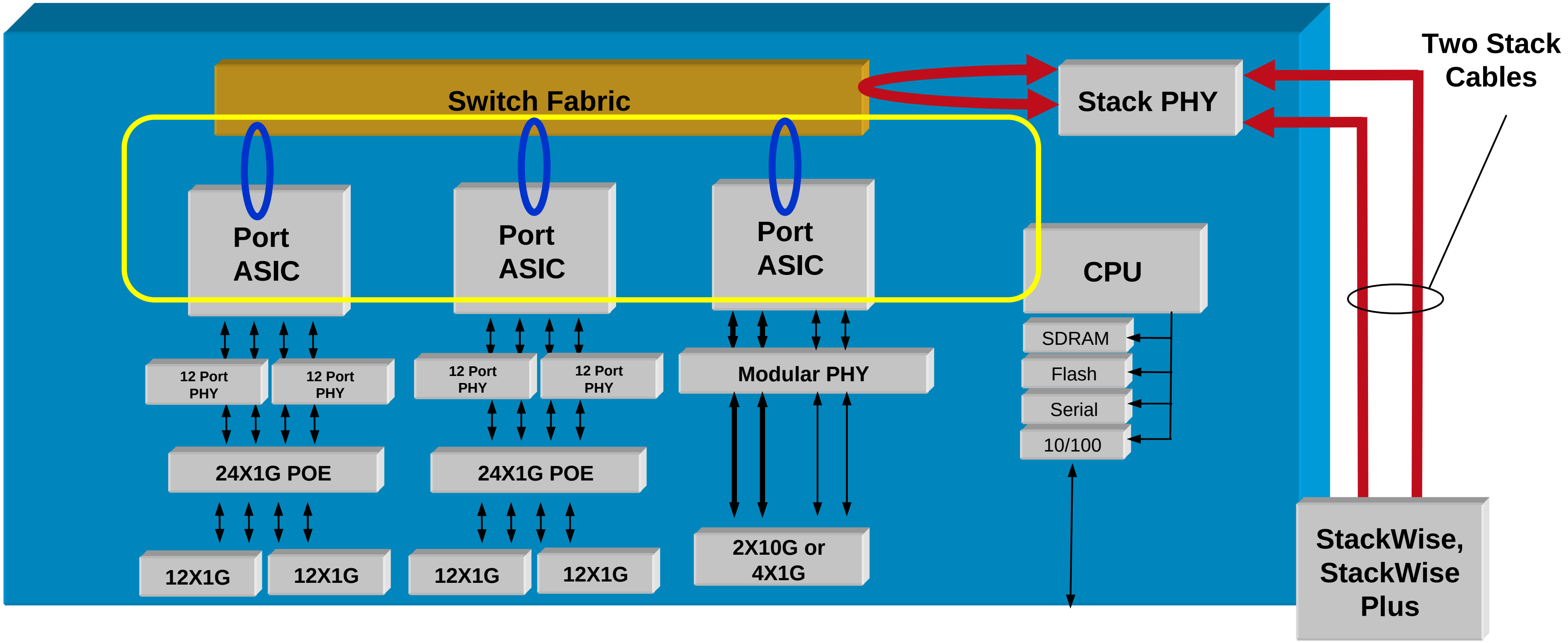


Architecture Overview: Processor



- Switch-to-Switch communication and synchronization
- Updates the MAC and Routing caches attached to each port ASIC
- Performs CPU-based slow-path forwarding when the TCAM is over its limits for MACs, Routes, ACL entries etc.
- The CPU communicates with the Port ASICs via a dedicated management 1G ring (the yellow ring in the diagram)

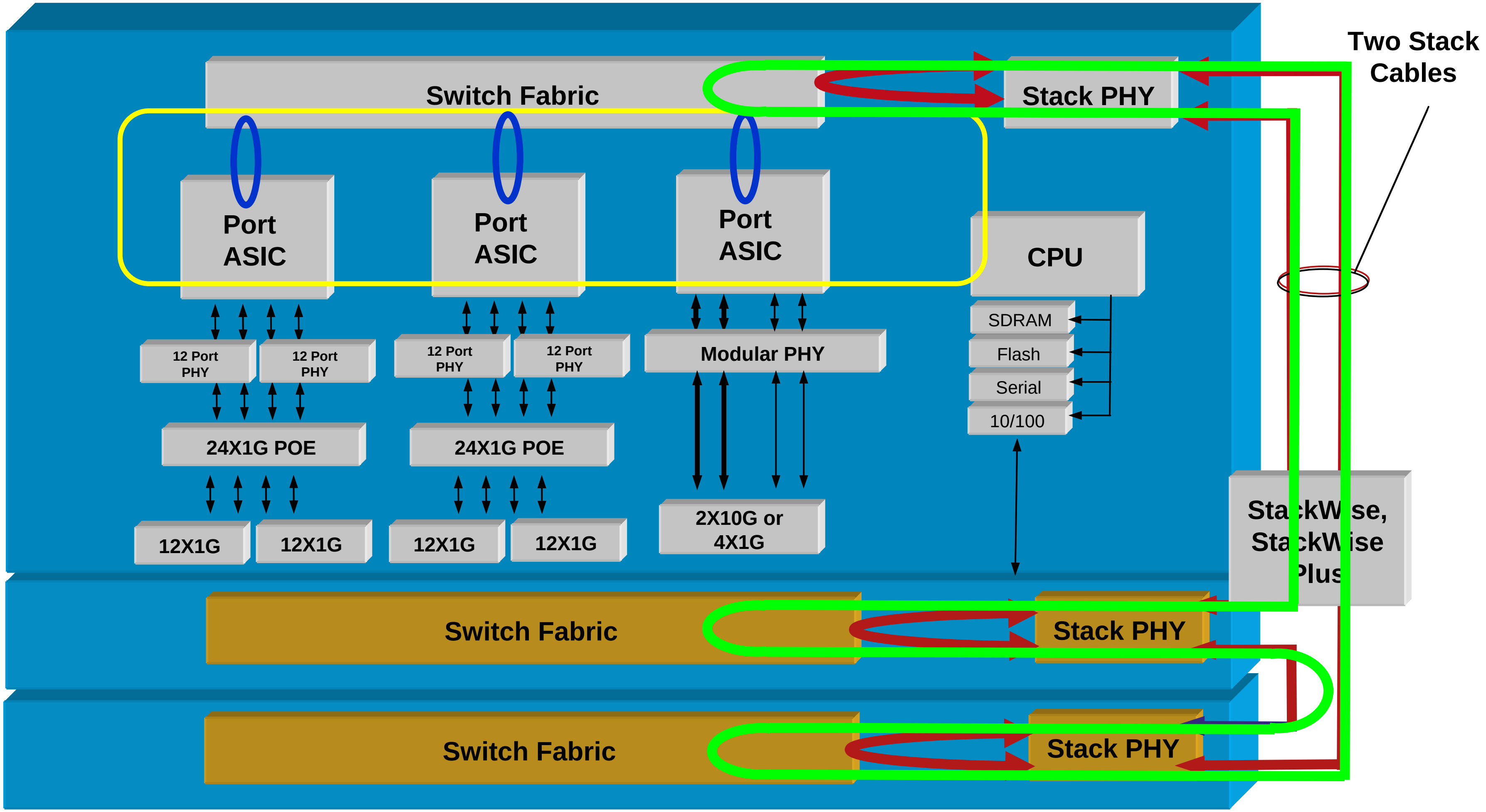
Architecture Overview: Switch Fabric



- 64 Gbps Ring Inter-connect data path to StackWises
- 1 Gbps Ring Inter-connect control path to the Port ASICs to the CPU
- 1 P2P, 24 Gbps ring connecting each Port ASIC
- Provides line rate local switching within a switch and stack connectivity

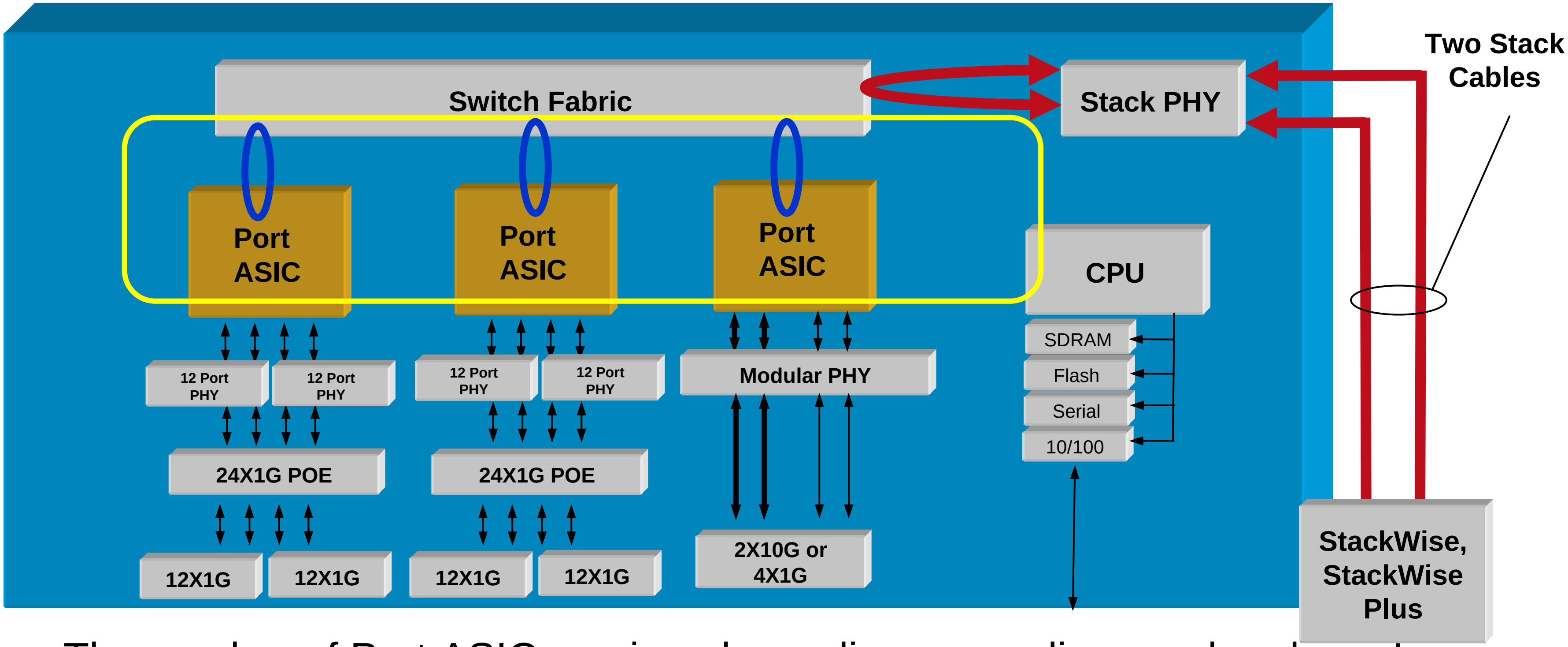
  48G + 2X10G traffic can be local switched (68G)
- Performs local switching between Port ASICs that are connected to it without using StackWise Plus resources
- Jumbo frame switching and routing

Ring View of the Switch Fabric



- Physically, the Ring Is a Series of Switch Fabrics Strung Together by Stack Cables
- The Switch Fabric performs token generation and ring control

Architecture Overview: Port ASIC



- The number of Port ASICs varies, depending on media speed and number of ports, maximum of 28 Gbps per Port ASIC
- The Port ASIC performs:

  Traffic forwarding

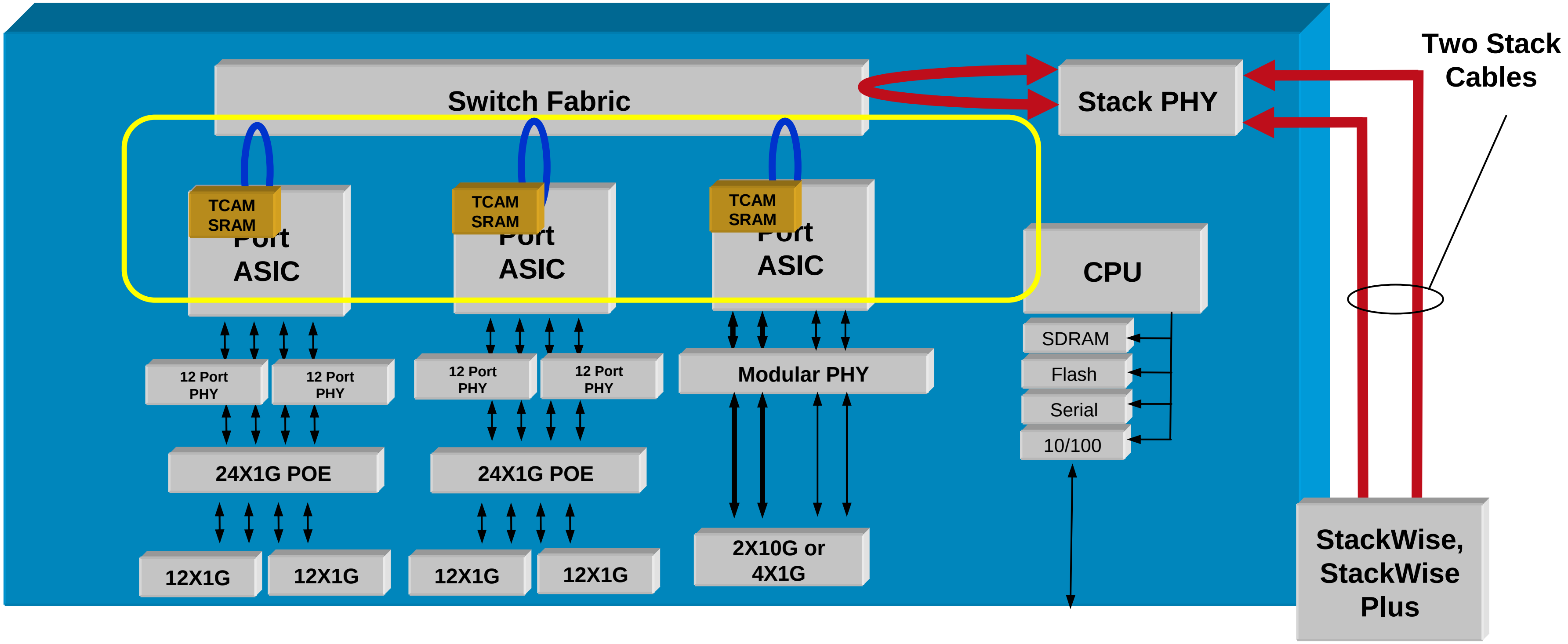
  QOS

  ACL lookup

  Route processing

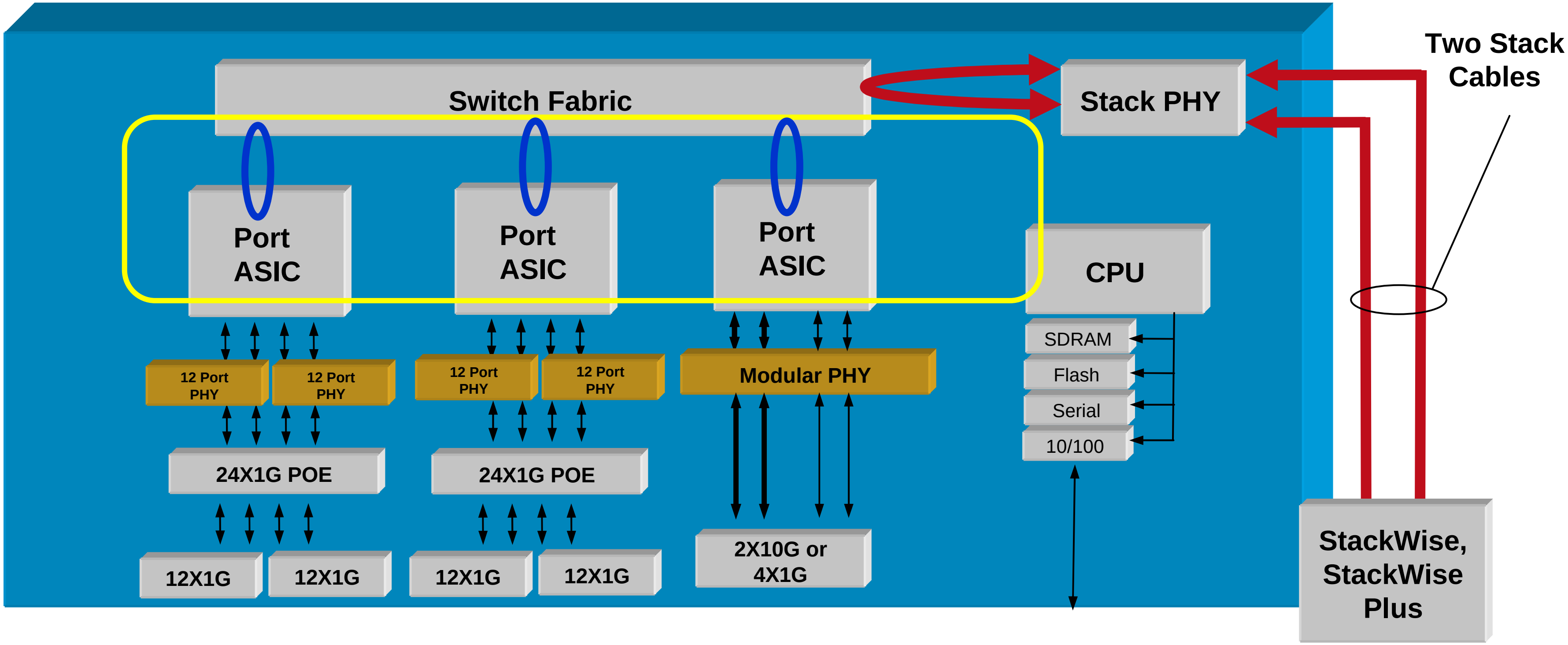
  STP

Architecture Overview: TCAM/SRAM



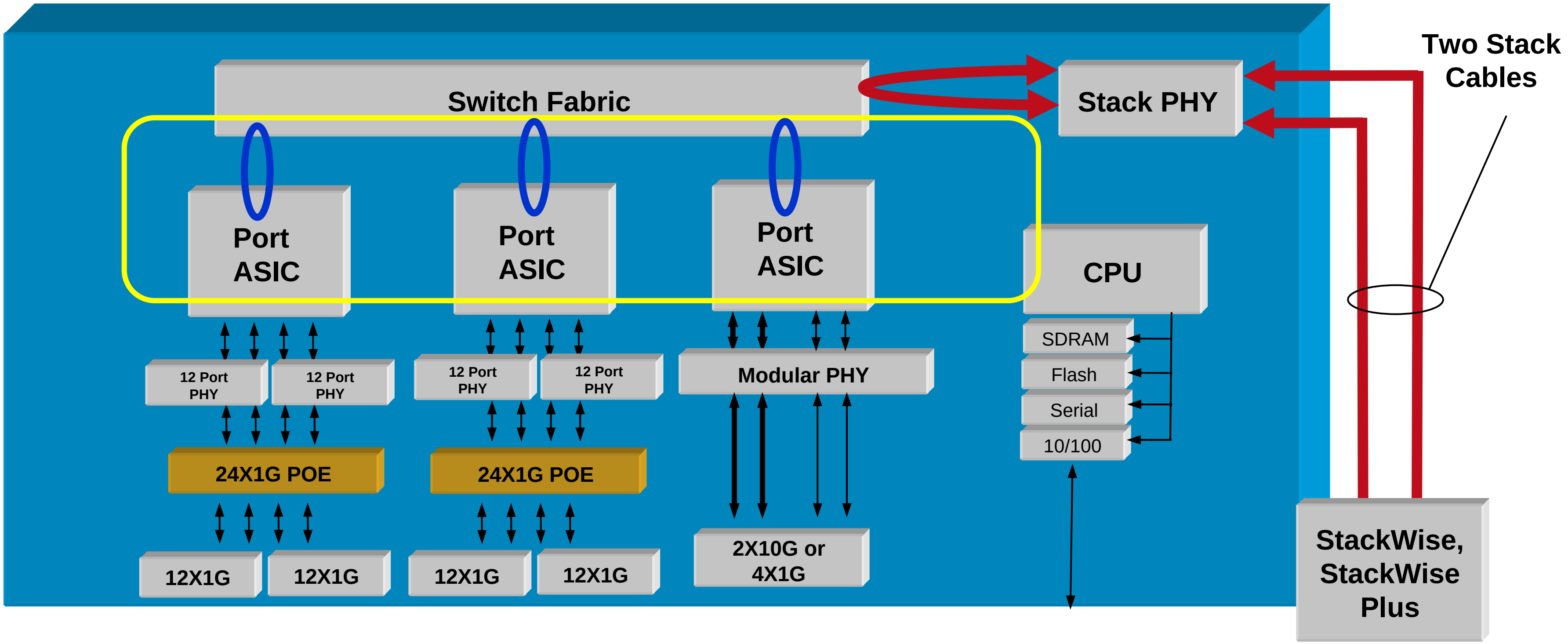
- Unlike the 3750, the 3750-E's TCAM/SRAM is incorporated into the Port ASIC
- The TCAM stores vital information including IPv4, IPv6 and MAC addresses
- The SRAM does use masking, it uses complete matches to forward
- SRAM tables have been sized to fit all existing Catalyst 3750 SDM templates
- SRAM forwarding supports L2, Multicast routing/bridging and unicast, and directly connected hosts for both IPV4/IPv6 and L3
- With the 3750-E one can now perform a simultaneous IP and MAC lookup with one ACE
- With the 3750-E it is now easier to configure the full 2K ACEs

Architecture Overview: PHY



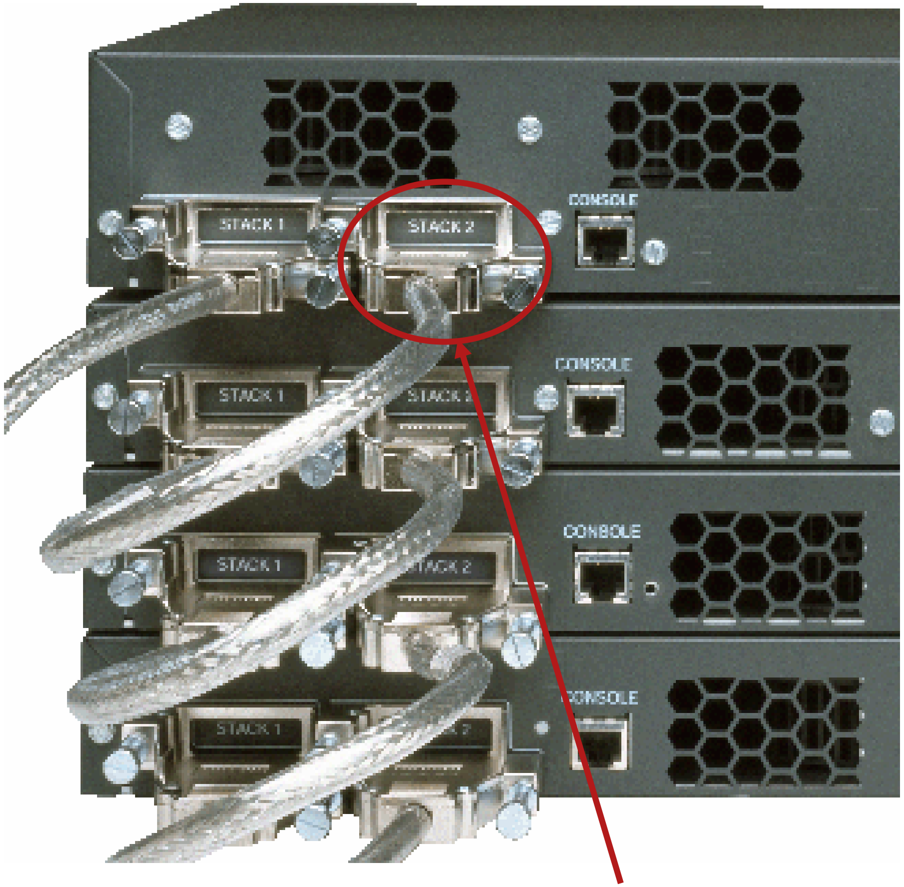
- All media conversion
- 10/100 Mbps
- 10/100/1000 Mbps
- 10 Gbps

Architecture Overview: POE



- 24 X 1G ports per POE
- Terminates all power to/from the PHY
- Performs per port auto-sensing and controls all POE

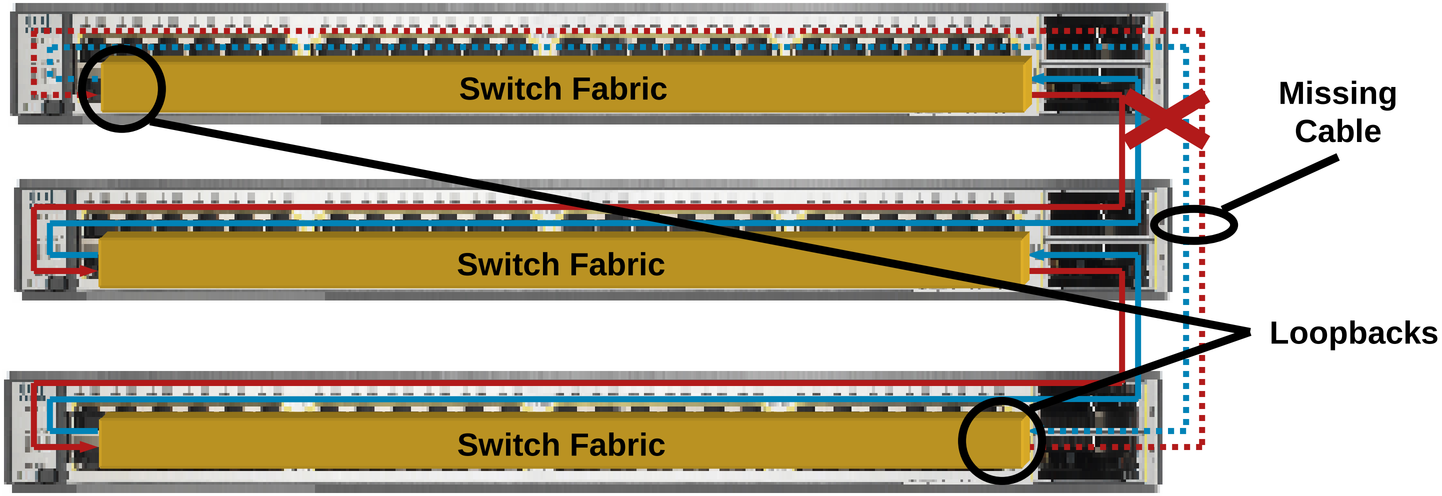
Stack—Cables



Heavily Braced Connector with Large Screws

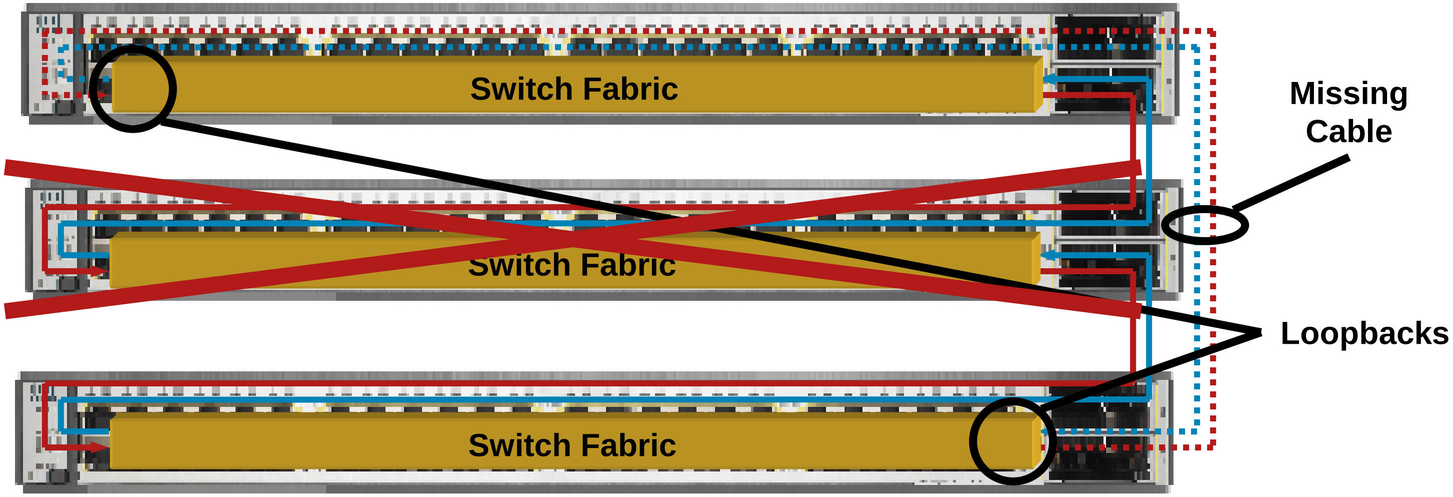
- Sizes: 50cm, 1m and 3m
- The Cisco Catalyst 3750-E/3750 has highly engineered stacking cables with patents pending on the stacking cable and connectors
- Heavy engineering and customer testing has shown that yanking and shaking this cable does not impact traffic flow and frames are not dropped
- Platforms using “recessed connectors” cable is exposed beyond the recess and can still easily be snagged. Thus, Cisco chose heavily reinforced non recessed connectors.
- Just pull on Cisco Catalyst 3750-E/3750 and you will see how strong the connector is
- Cisco has shown that a switch can even forward traffic when it is being swung around in the air…this is not a recommended deployment ☺

Healing a Missing Cable



- The Switch Fabric closest to cable detects link down
  - Criteria is coding violations in a period of time
  - Loss of at most one packet that was being transmitted when ring broke
  - Just microseconds for hardware to detect failure
- Each switch signals a bad link to stack its partner
- Both ends of the cable loop back on themselves
  - 32 Gbps backplane when cable is missing

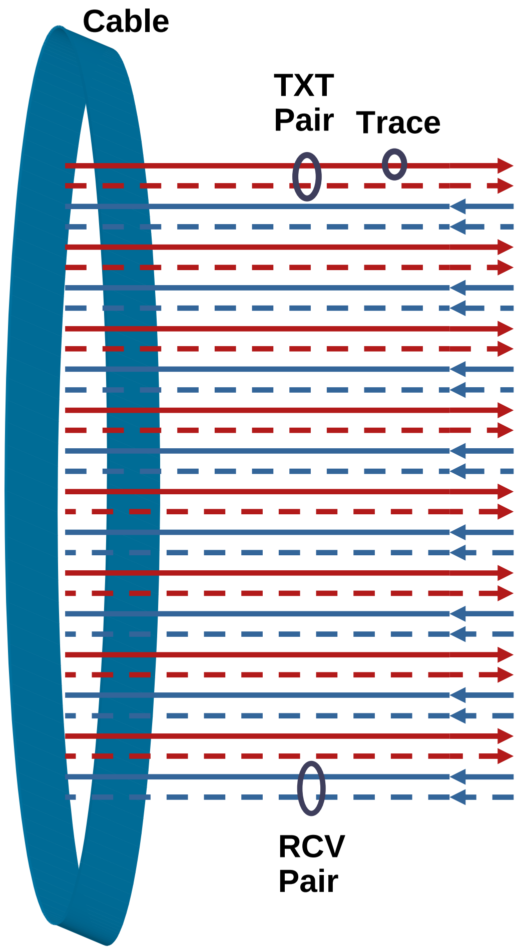
Healing a Failed System



- The Switch Fabric closest to cable detects link down
- The stack topology is rediscovered by proprietary Stack Discovery Protocol
- Both switches signal bad link to their stack partner
- Both of the links to the failed system loop back on themselves

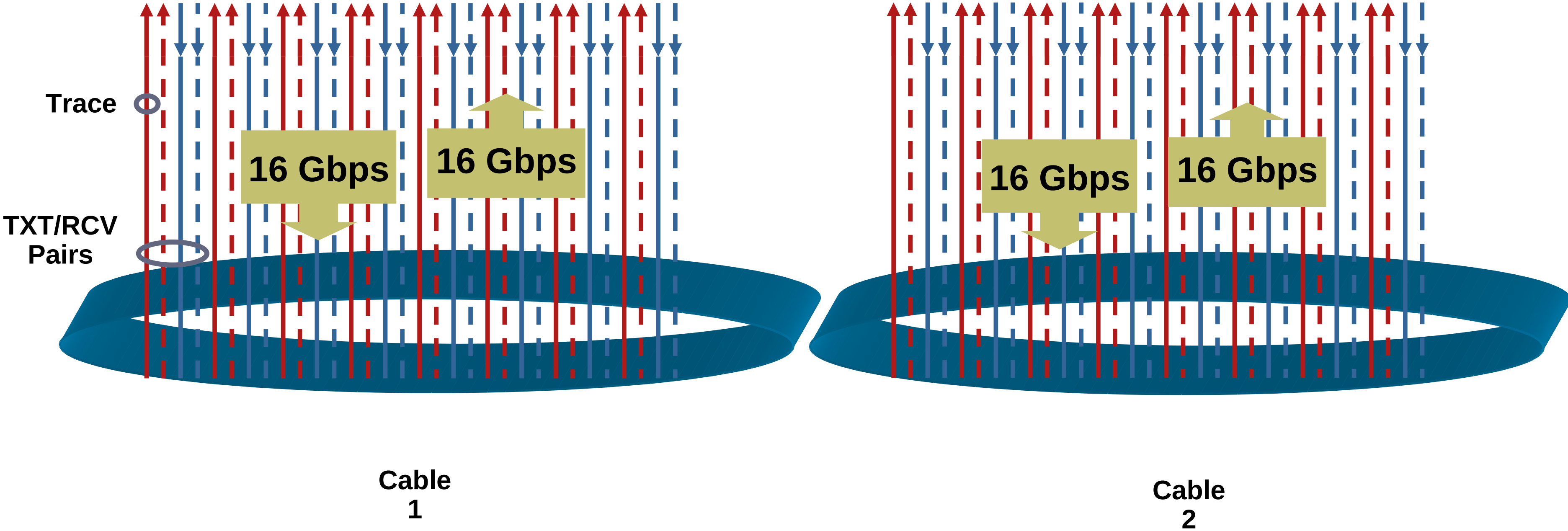
Understanding the Stack Cable

- Eight TXT/RCV pairs, that is 16 total pairs
- Each TXT/RCV pair has two traces that use differential signaling. That is 32 traces in total.
- Each TXT/RCV pair runs at 2.5 Gbps
- 8B/10B encoding is used. That is, for every ten bits sent, eight bits are user data and two bits are overhead.



Understanding the Stack Ring Speed

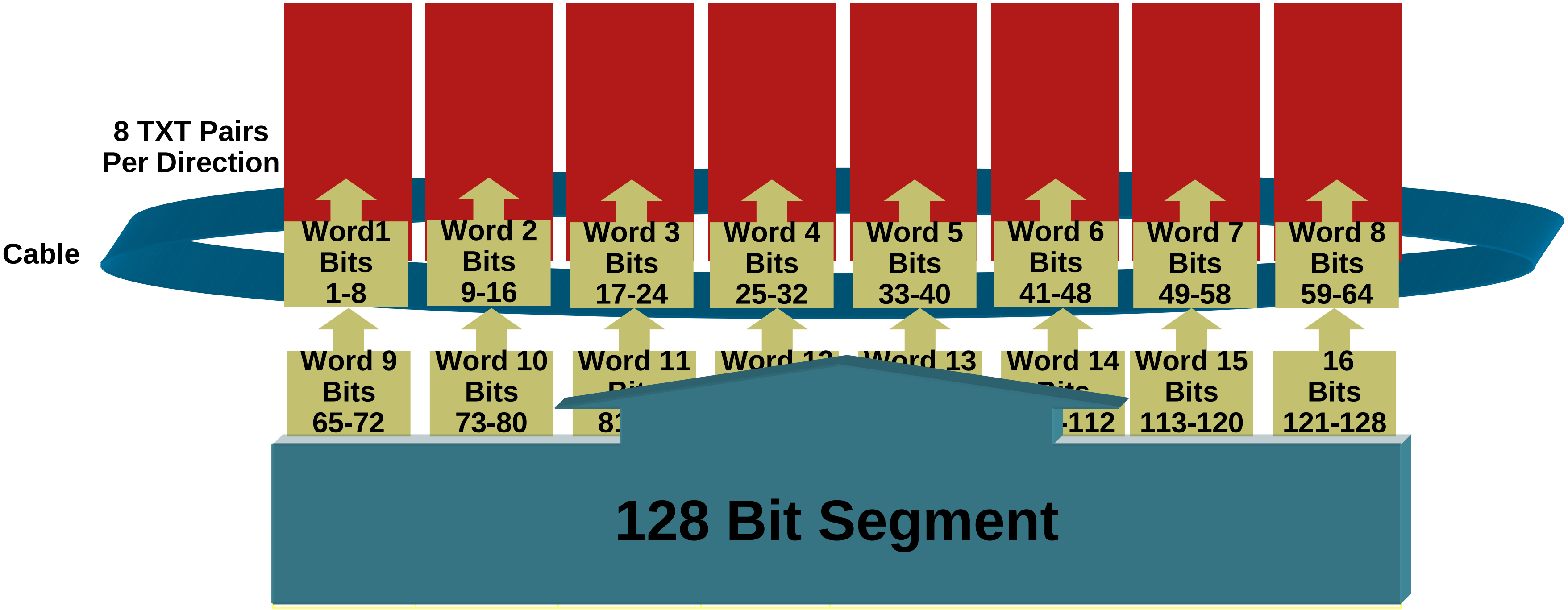
- Two Cable X 16 Pairs/Cable X 2.5 Gbps X 8B/10B = 64 Gbps*
- Or 32Gbps send and 32 Gbps receive (bidirectionally)*
- Or 16 Gbps per cable bidirectionally*



* This is physical bit rate not necessarily throughput

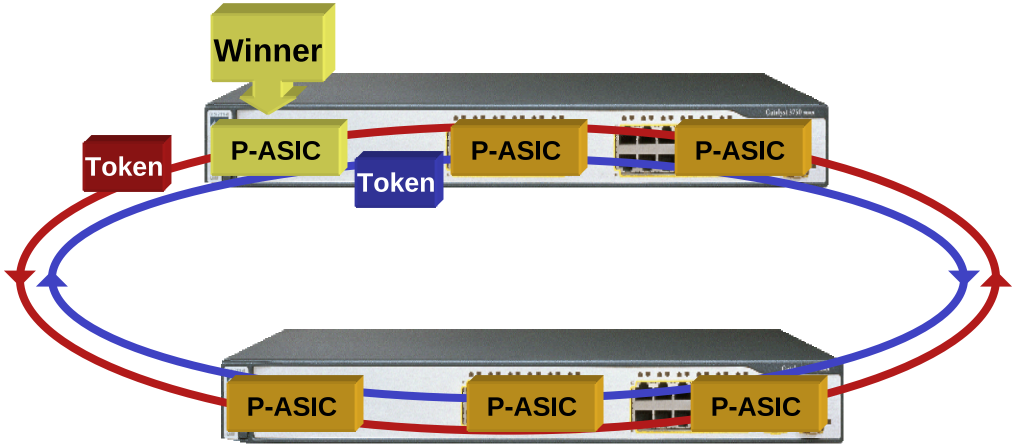
Writing onto the Stack Cable

- Each packet to be transmitted is broken into 128 bit segments
- Each segment broken up into 16x8 bit words
- The first eight words are then transmitted on the cable in parallel. Then the last eight are transmitted.
- Each cable transmits a true clear channel stream, not channelized



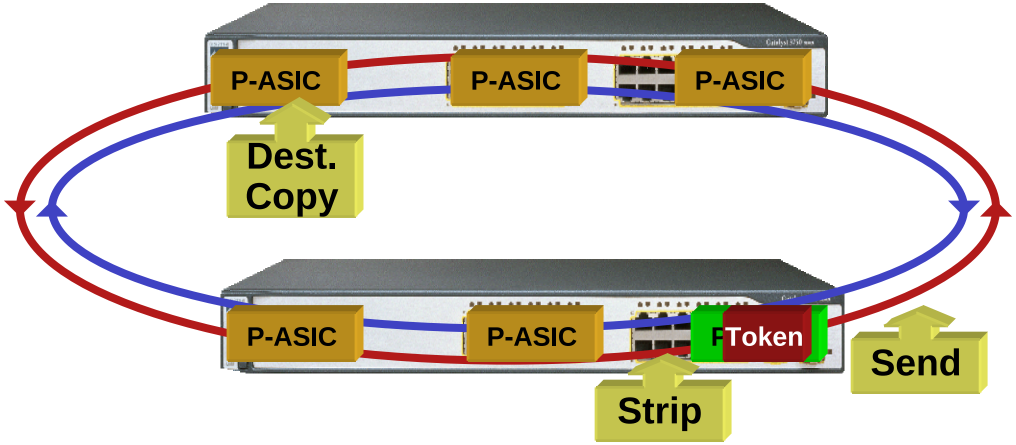
Ring Token Generation-StackWise

- At time = 0, each port ASIC on the ring negotiates to see who has the lowest ID
- The port ASIC with the lowest number generates the tokens (winner)
- There are two tokens, one for each direction around the ring



Ring Access: StackWise

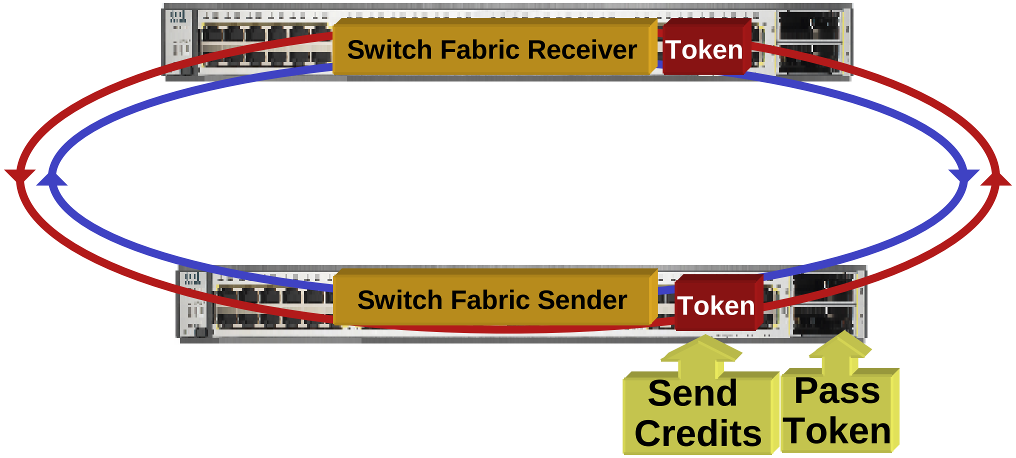
- When a Port ASIC receives a token it holds it while it transmits
- For each ring/direction the is a Target Rotation Time (TRT). This is the maximum amount of time they should have to wait after they pass the token.
- The destination port copies the packet
- Packets are source stripped, stripped by the port ASIC that transmits the packet
- Each port ASIC can only keep the token for a finite period of time. When that time is up or all of the packets are finished transmitting, it passes the token.
- Once a port ASIC chooses a ring (grabs a token) to transmit on, it is locked to that ring until all the outstanding frames it has placed on the ring return and are taken off the ring



Ring Access: StackWise Plus

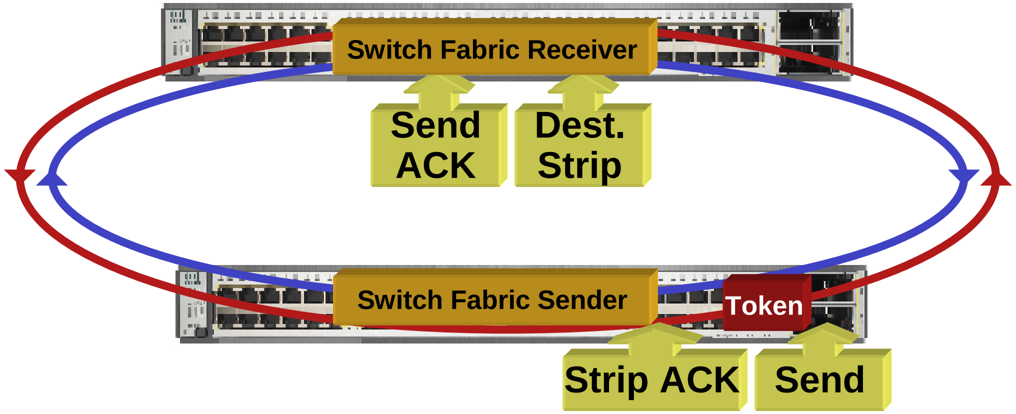
- The Switch Fabric generates and controls send credits for each Port ASIC
- A Port ASIC can send if the following conditions are true:
  - The Port ASIC has packets to send
  - The Port ASIC has “send credits”
  - When its upstream neighbor is not sending packets

- The sendinf Fabric:
  - Recieves the token,
  - Sets its “send credits”.
  - Passes the token and then
  - Sends its traffic according to the pervious 3 rules.
- Passing the token allows spatial reuse to occur.



Ring Access: StackWise Plus Cont'd.

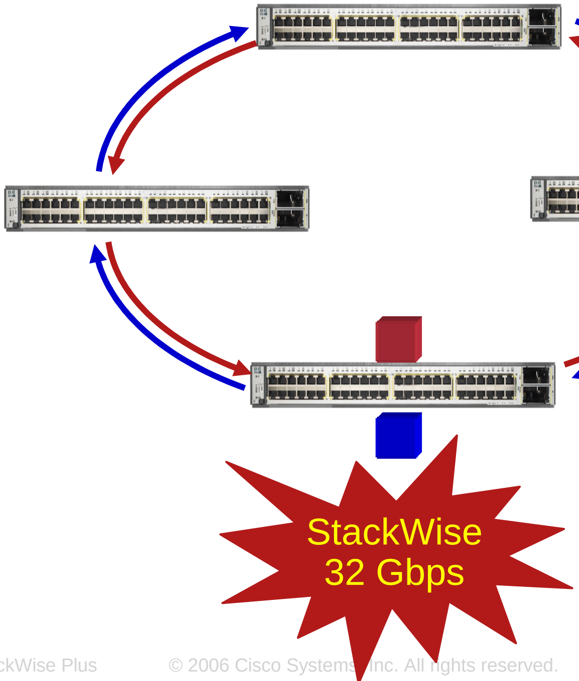
- The destination Switch Fabric strips the packet, i.e. destination stripped, multicast frames are source stripped.
- The destination Switch Fabric sends a 1 word ACK to the sender.
- The sender strips the ACK
- Once a Switch Fabric chooses a ring (grabs a token) to transmit, it is locked to that ring until all the outstanding frames it has placed on the ring return and are taken off the ring



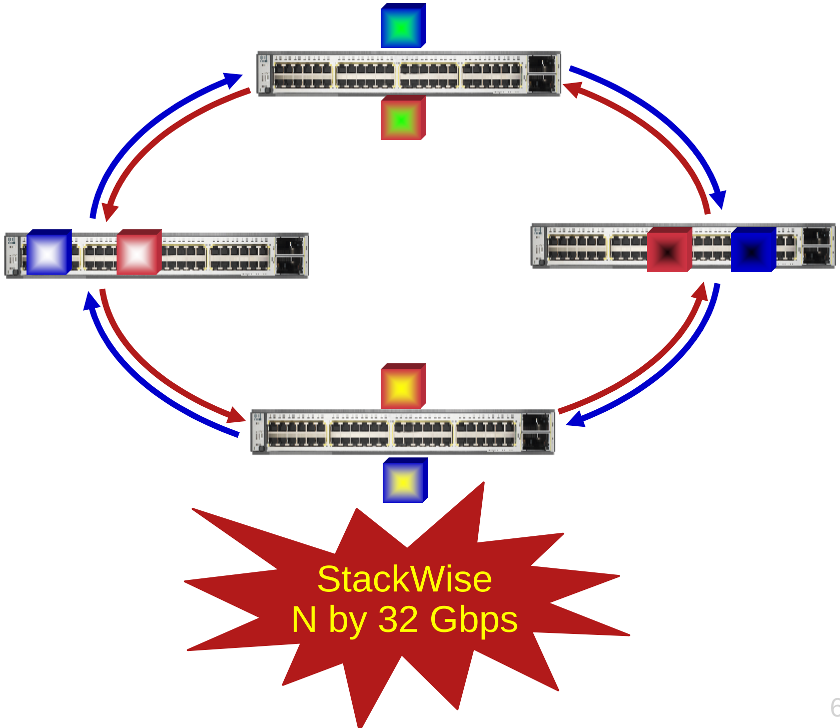
Spatial Reuse

- Since packets are destination stripped, and a true ring protocol is used, bandwidth is available on the links that the packets does not traverse

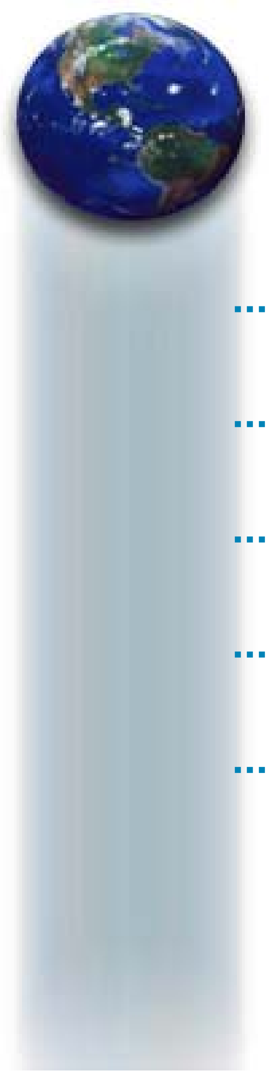
No Spatial Reuse
Only 2 Flows



Spatial Reuse
N by 2 Simultaneous Flows



Agenda



StackWise Operation

Mixing StackWise Plus and StackWise

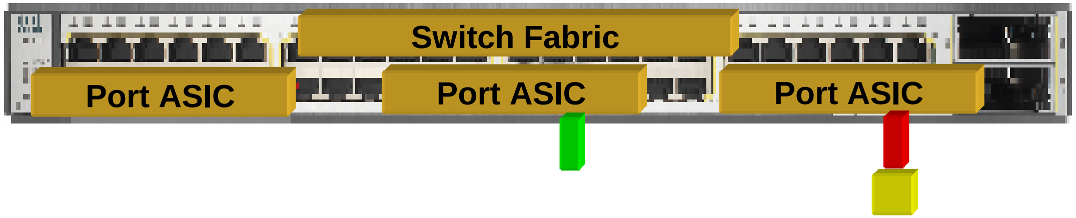
QoS

Hardware Detail

Packet Flow Detail

Port ASIC Detail

Unicast Packet Walk – Locally Switched



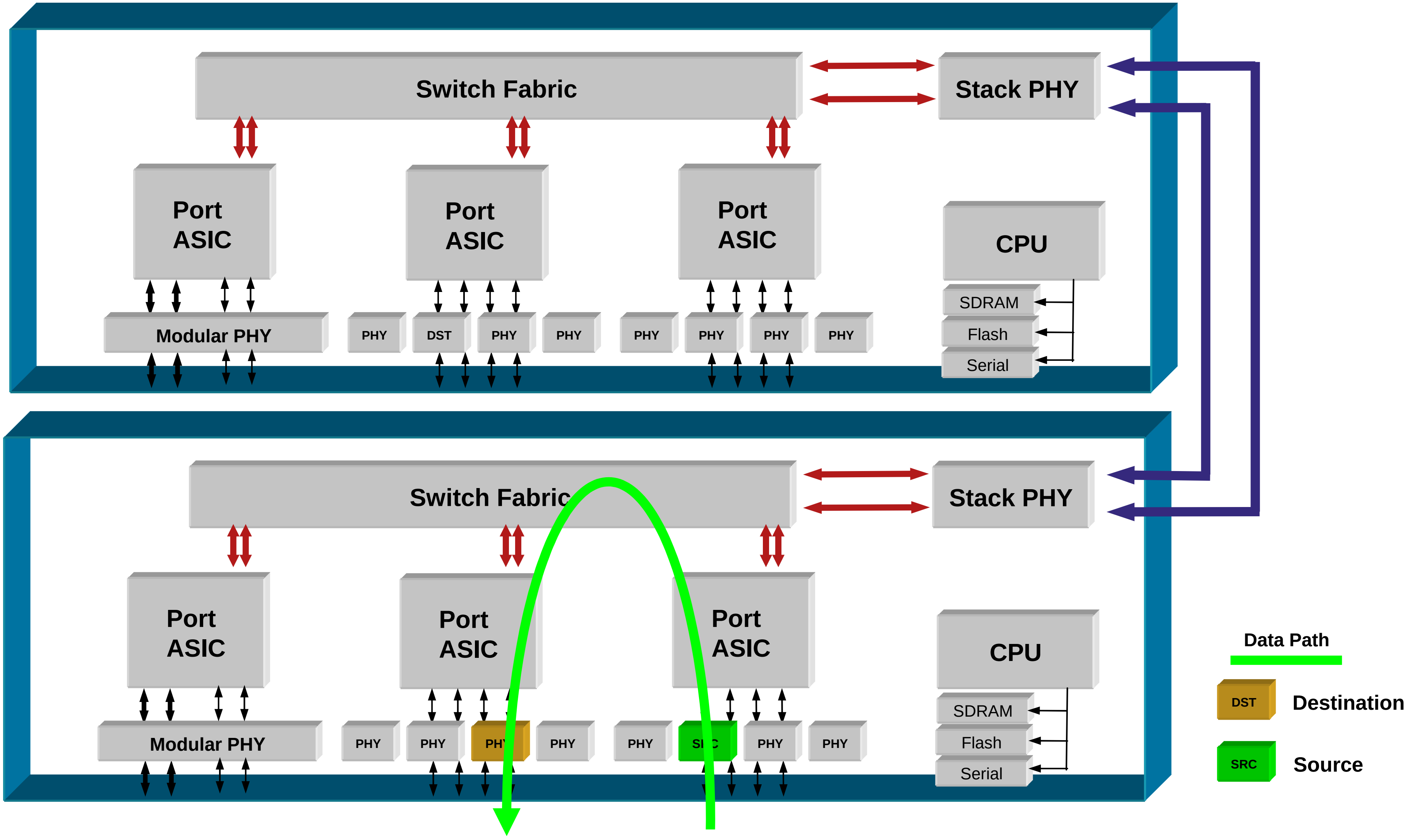
- The packet is sent to the switch Fabric and switched to the destination Port ASIC

🟩 Source

🟥 Destination

🟨 Data

Day In the Life Of A Unicast Packet-Locally Switched



Day in the Life of a Unicast Packet-Locally Switched

- The source port PHY
  forwards the packet to the Port ASIC

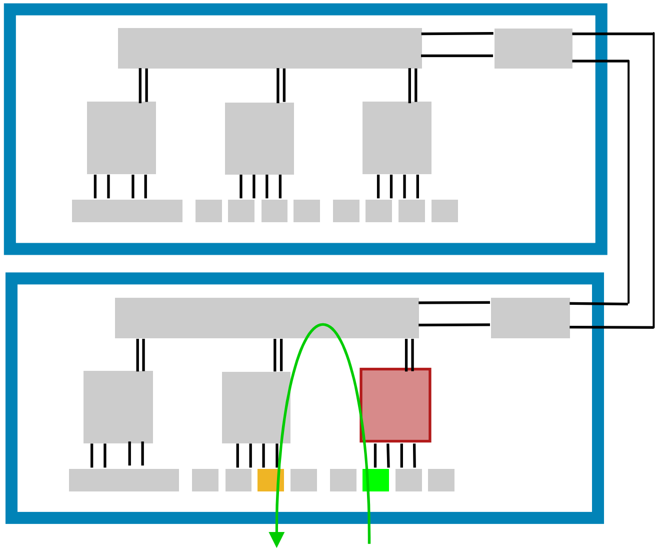
- The source Port ASIC

  Determines the packet’s destination

  Generates the 24B forwarding header
  (for internal use only)

  When necessary, it notifies the
  processor (ex. new MAC on this port).
  The processor then updates the
  TCAM/SRAM in the ring.

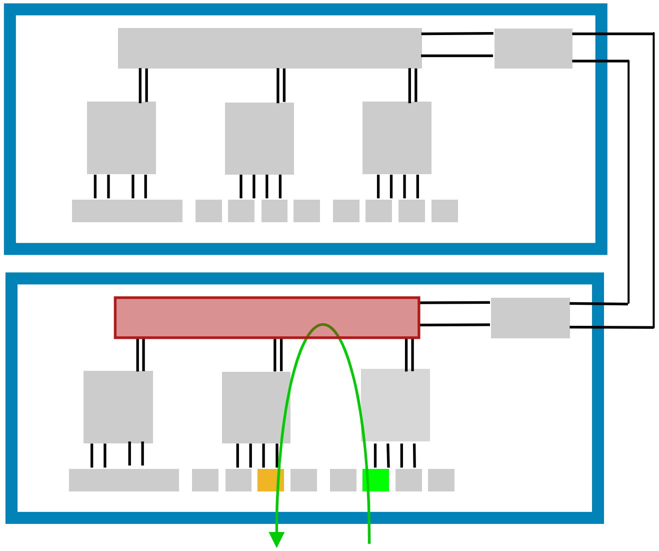
  The Port ASIC forwards the packet to the Switch Fabric



Day In the Life Of A Unicast Packet-Locally Switched

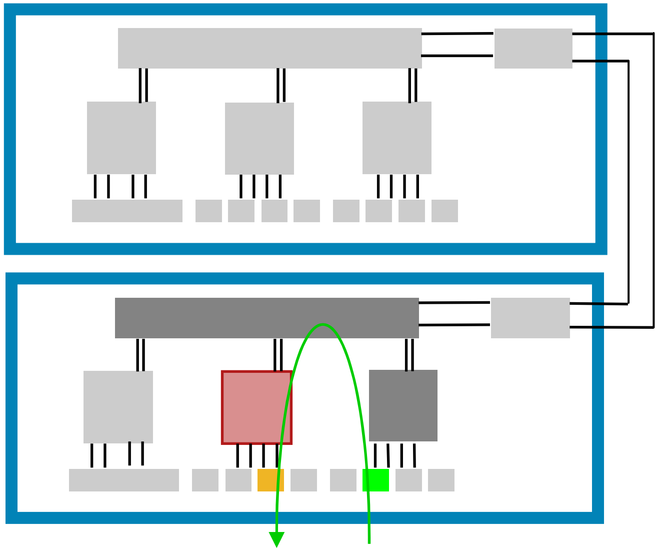
- The source Port ASIC forwards the packet to the Switch Fabric
- The Switch Fabric

  Forwards the packet to the destination port ASIC, which can be the same port ASIC that forwarded the packet to the Switch Fabric. The port ASIC can not switch packets by itself.

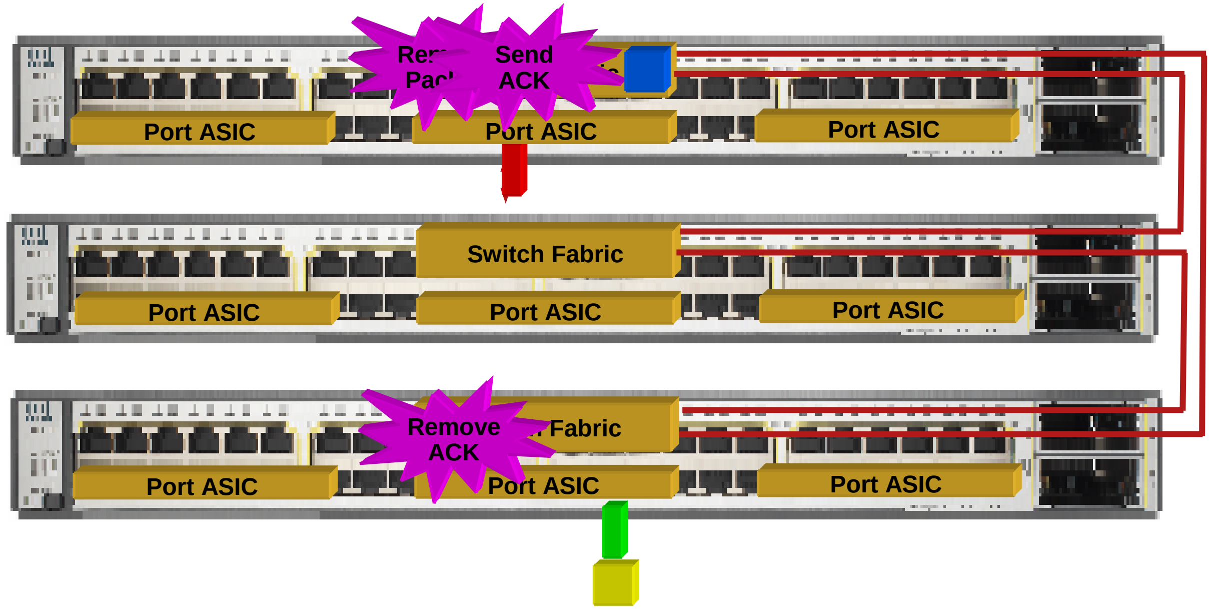


Day In the Life Of A Unicast Packet-Locally Switched

- The Destination Port ASIC forwards the packet to the destination PHY



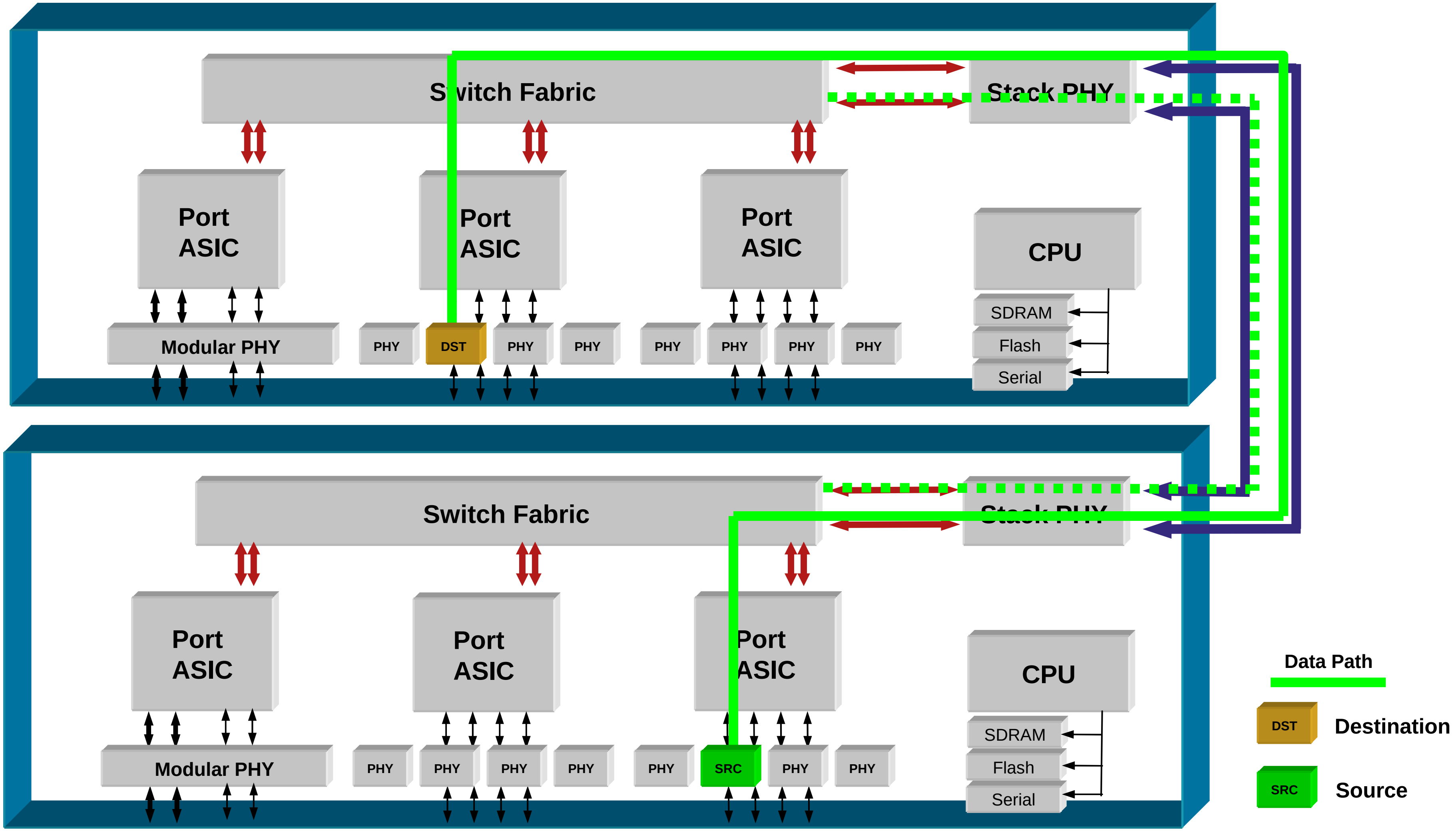
Unicast Packet Walk – Remote Destination



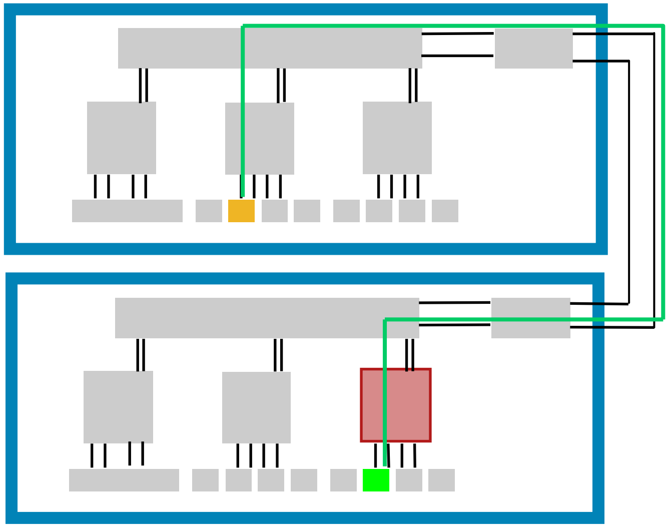
- The Source Port ASIC sends the packet to the Source Switch Fabric and it is switched to the Destination Switch Fabric
- The Destination Switch Fabric removes the packet and sends a 1 word ACK
- The Originating Switch Fabric receives and removes the ACK

🟩 Source
🟥 Destination
🟨 Data
🟦 ACK

Day In the Life Of A Unicast Packet-Remote Destination



Day in the Life of a Unicast Packet Day In the Life Of A Unicast Packet-Remote Destination



- The source port PHY
  forwards the packet to the Port ASIC
- The source Port ASIC

  Determines the packet’s destination

  Generates the 24B forwarding header
  (for internal use only)

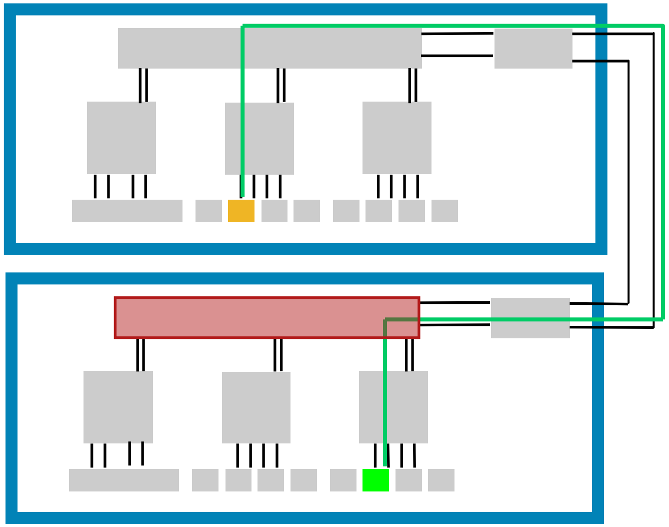
  When necessary, it notifies the
  processor (ex. new MAC on this port).
  The processor then updates the
  TCAM/SRAM in the ring.
- The Port ASIC forwards the packet to its Switch Fabric

Day In the Life Of A Unicast Packet-Remote Destination

- The source Switch Fabric forwards the packet to the Stack PHY

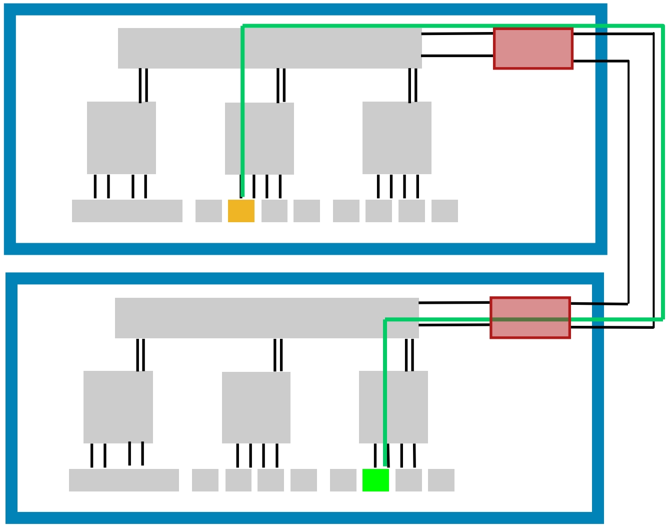
- The Switch Fabric

  For unicast traffic, determines the packet's if destination is on the same switch or on another switch in the stack



Day In the Life Of A Unicast Packet-Remote Destination

- The packet is copied on the stack PHY and passed to the next 3750-E in the stack

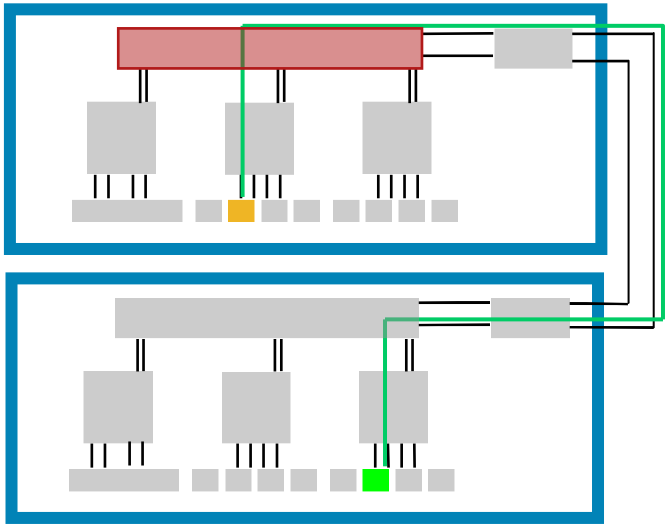


Day In the Life Of A Unicast Packet-Remote Destination

- The source Stack PHY
  forwards the packet to the Switch Fabric

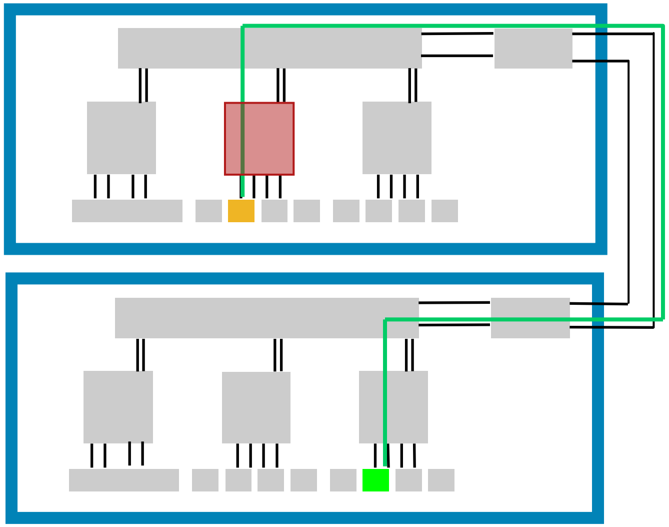
- The Switch Fabric

  Determines the packet’s if destination forwards it to the port ASIC



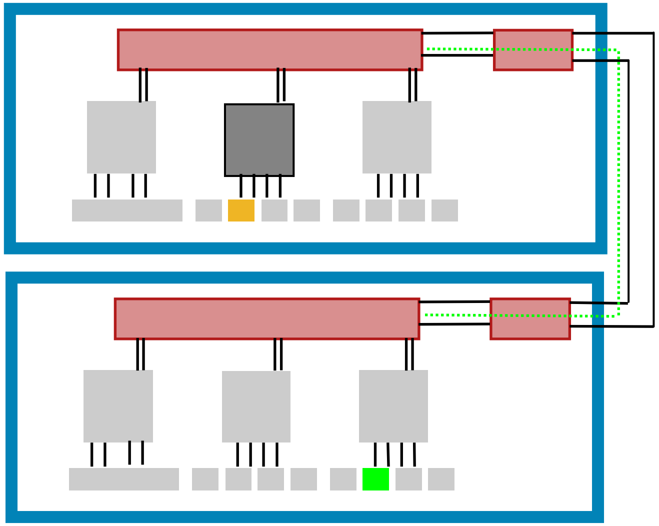
Day In the Life Of A Unicast Packet-Remote Destination

- The Switch Fabric sends the packet to the destination Port ASIC
- The destination Port ASIC
  - Determines it is the packet’s destination port
  - Removes the internal forwarding header
  - Forwards the packet to the PHY

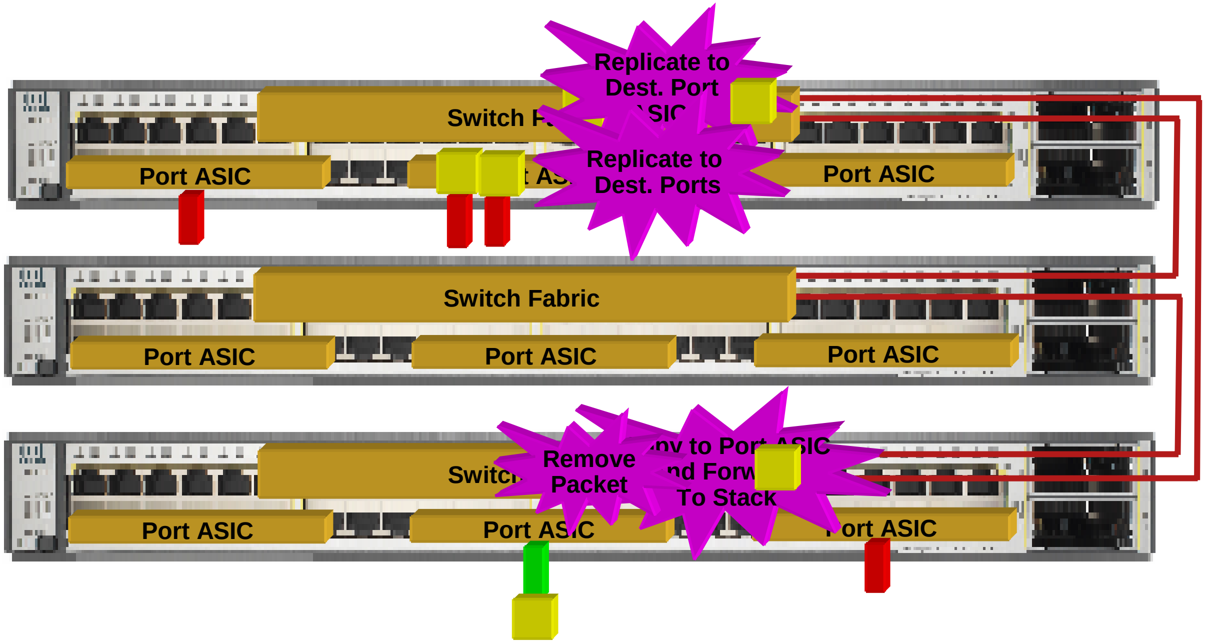


Day In the Life Of A Unicast Packet-Remote Destination

- The Destination Switch Fabric sends a 1 word ACK to the Source Switch Fabric
- The Source Switch Fabric
    - Strips the ACK



Multicast Packet Walk



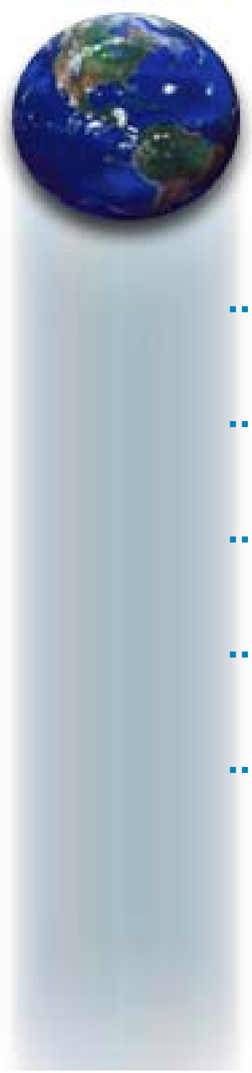
- The packet is passed all the way around the ring
- Port ASICs with Multicast ports in that group copy the packet
- The originating Port ASIC removes the packet from the ring
- Note: There is only one packet on the ring per multicast flow, replication only occurs at the local level

🟩 Source

🟥 Destination

🟨 Data

Agenda



StackWise Operation

Mixing StackWise Plus and StackWise

QoS

Hardware Detail

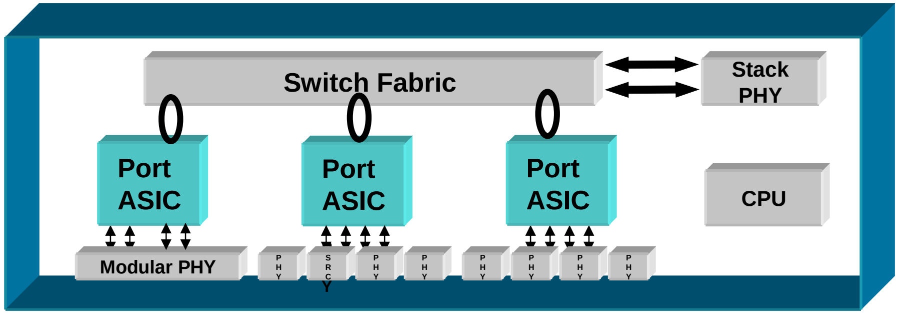
Packet Flow Detail

Port ASIC Detail

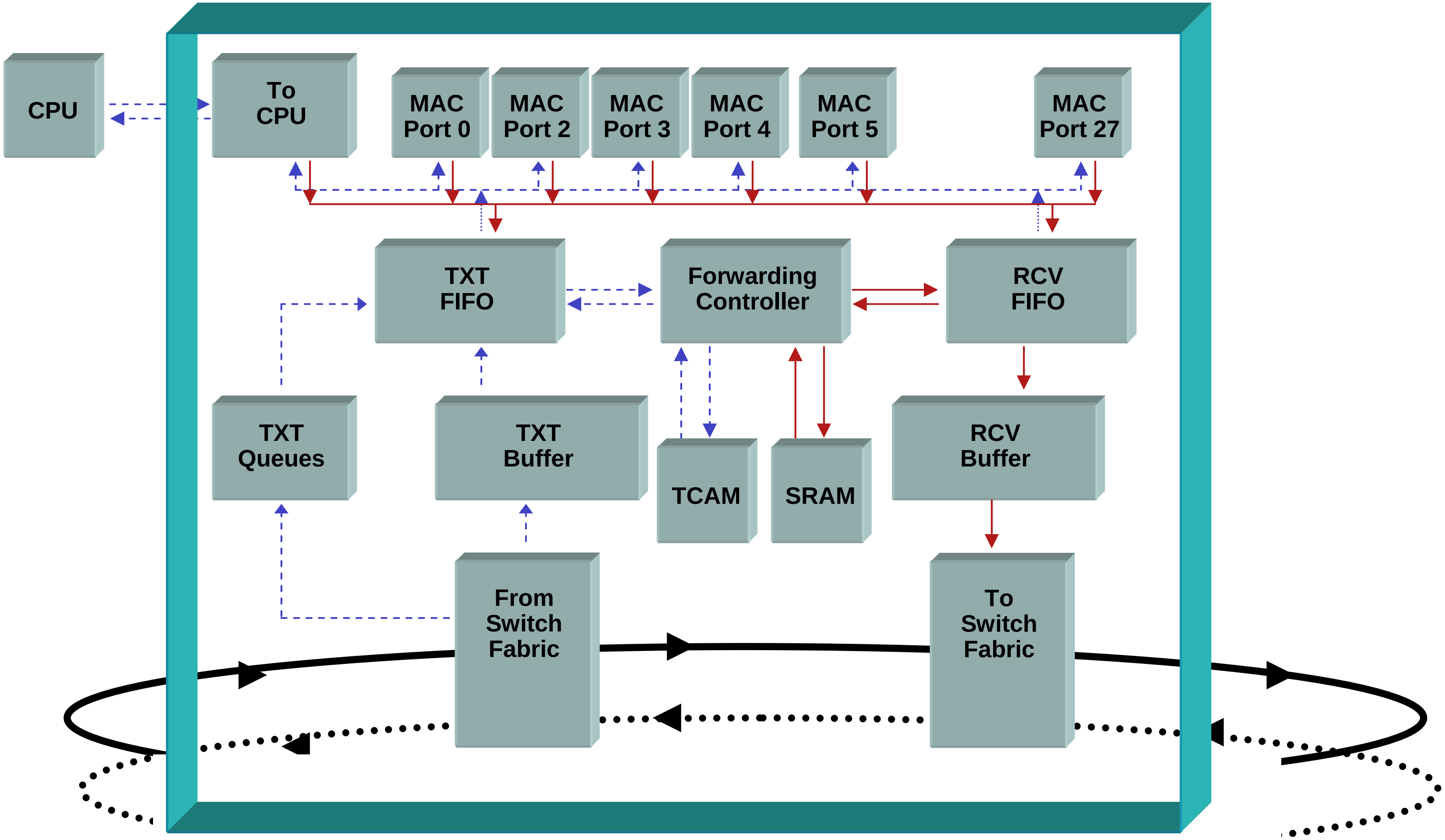
The Port ASIC

The Port ASIC Block:

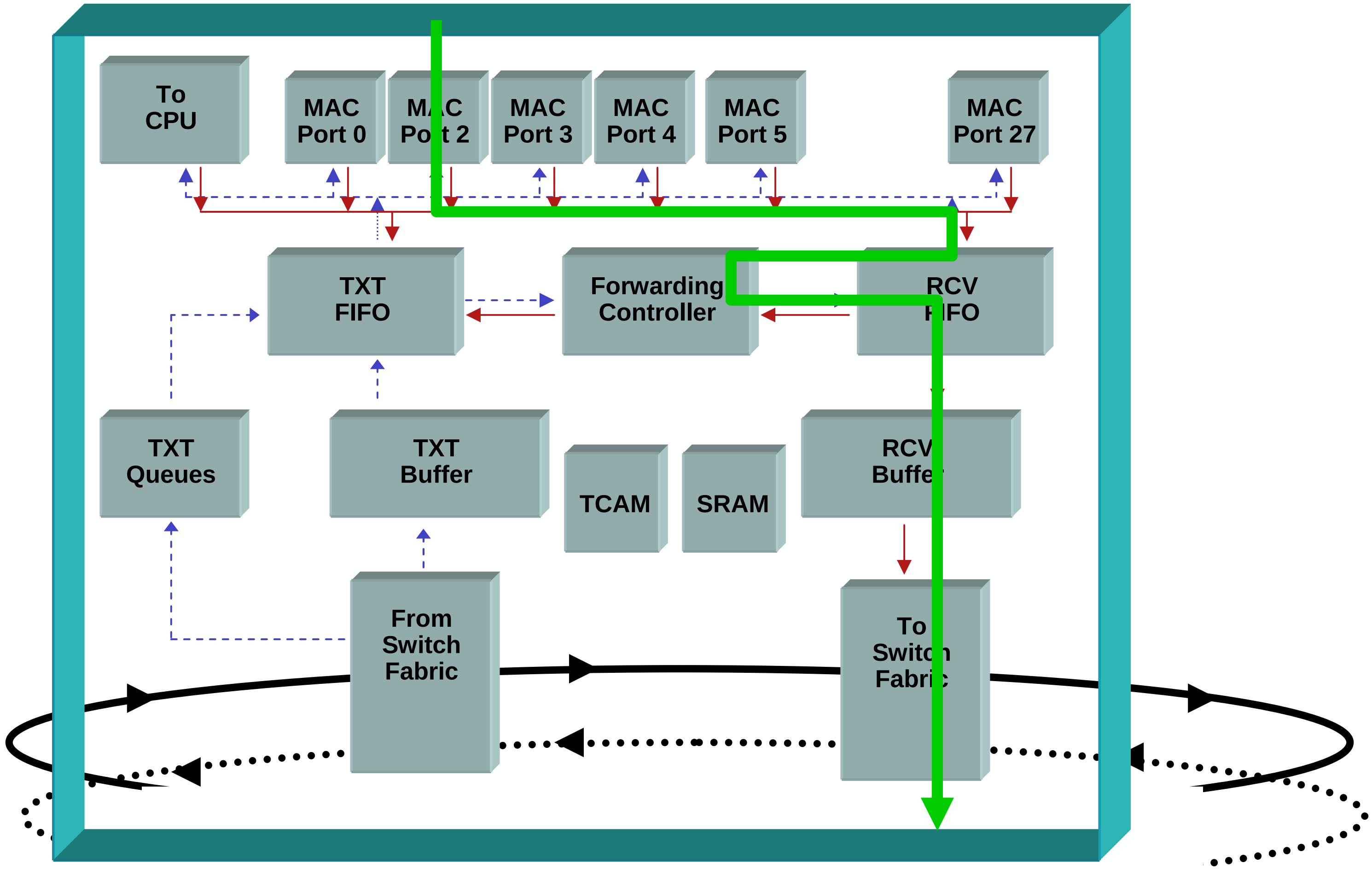
- Is a key value add of Cisco innovation
- Performs buffering
- Forwarding lookup
- Reads/Writes to Switch Fabric
- Quality of service
- ACL lookup



Port ASIC Architecture Exposed

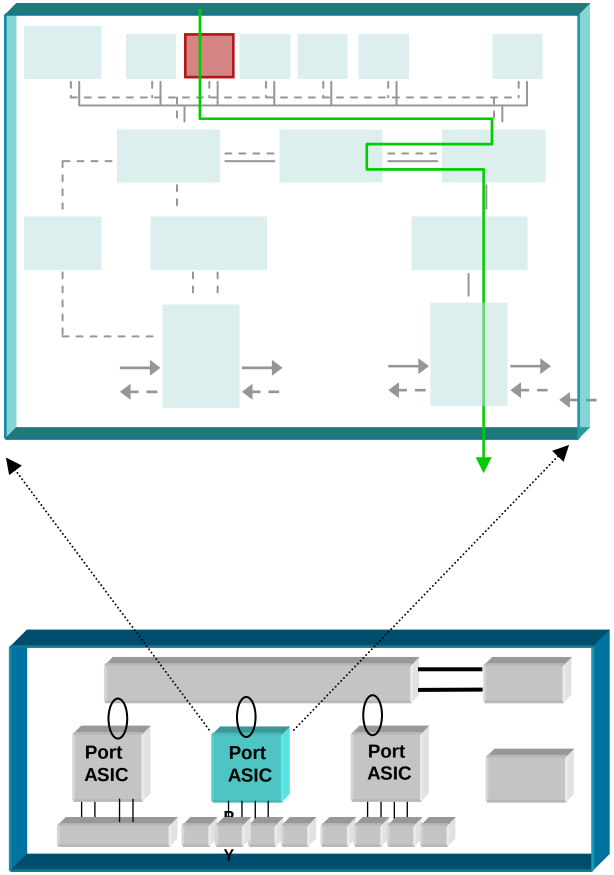


Ingress Flow



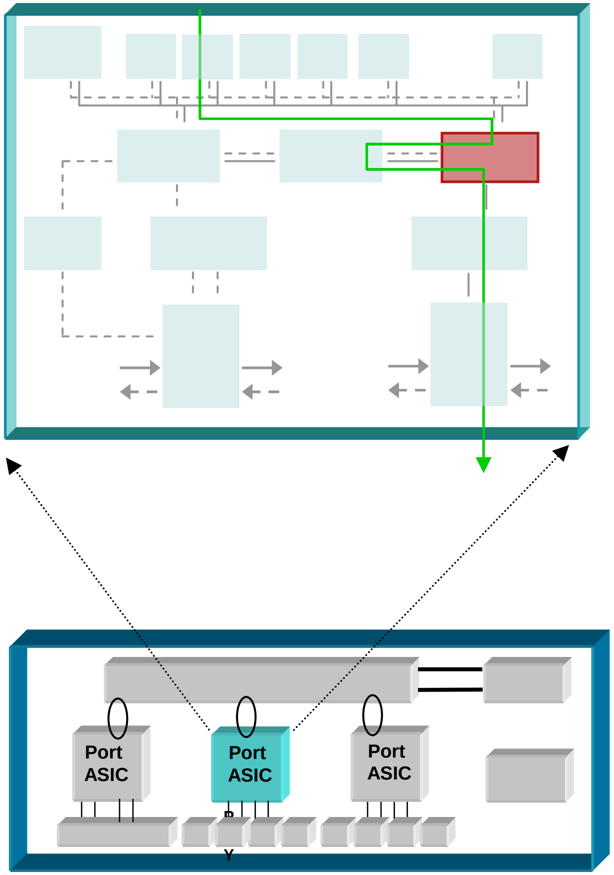
Ingress Flow: MAC Port

- All physical layer functionality is terminated prior to entering this port ASIC function
  - Encoding
  - Power over Ethernet
  - Etc.
- The MAC port's main function is to implement Ethernet Media Access Control
- The MAC port function also adds the 24B internal header, which may be modified later
- This header is used to guide the packet to its destination
- The packet is then passed to the RCV FIFO



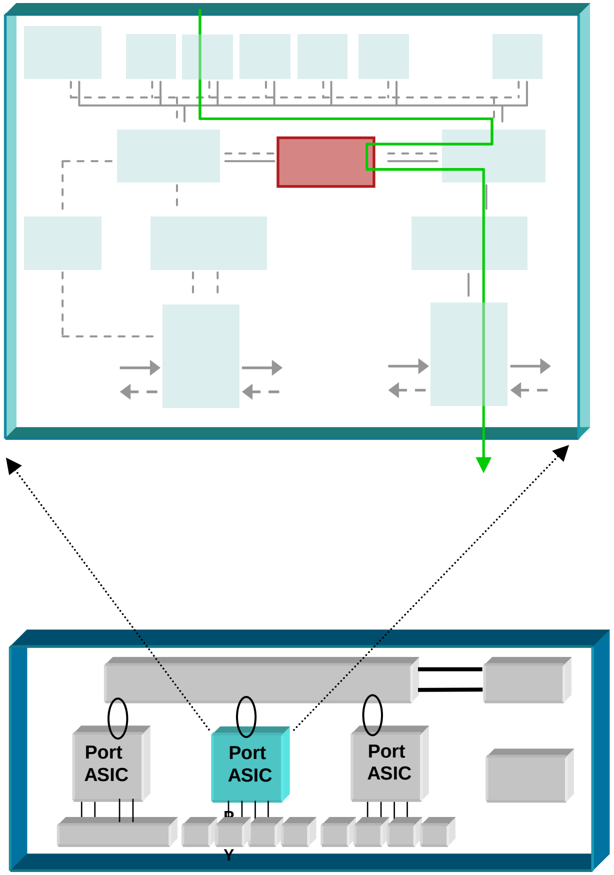
Ingress Flow: RCV FIFO

- The packet enters the RCV FIFO from the MAC port
- There is one physical memory divided into multiple logical RCV FIFOs to serve all of the MACS on the Port ASIC
- One FIFO per port
- The RVC FIFO absorbs time so the forwarding controller to do its job



Ingress Flow: Forwarding Controller

- The forwarding controller reads the 24 Byte header and up to 200 Bytes of the packet and performs
  - Forwarding lookups
  - QoS labeling
  - Marking (packet dropping is not performed at this point)
  - ACL lookup
- After the header is updated to the RCV FIFO, the packet is passed to the RCV buffer

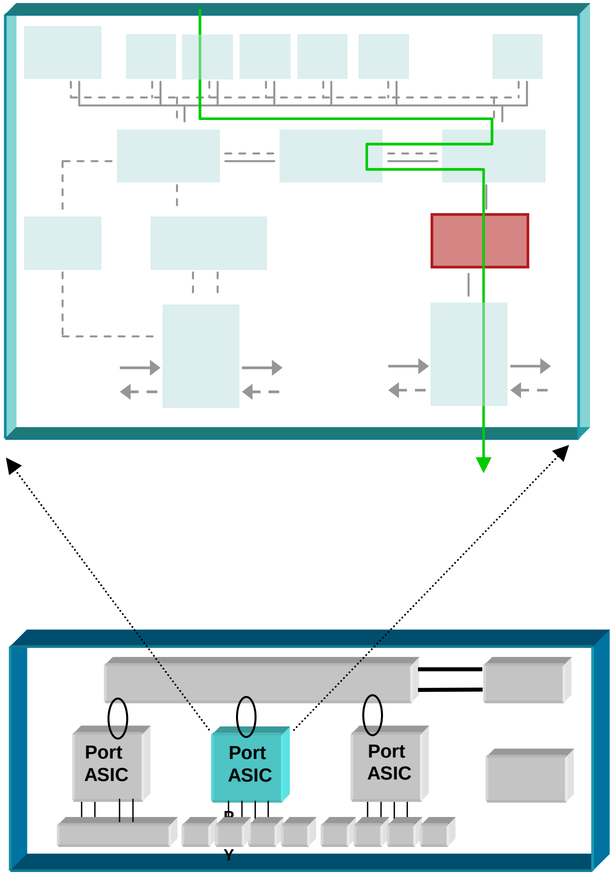


Ingress Flow: RCV Buffer

- The packet enters the RCV buffer while it waits for ring access
- This is where the two manageable egress queues can be configured and packets can be dropped
- SRR is performed on these queues
- WTD can be/is also performed here
- Each buffer:

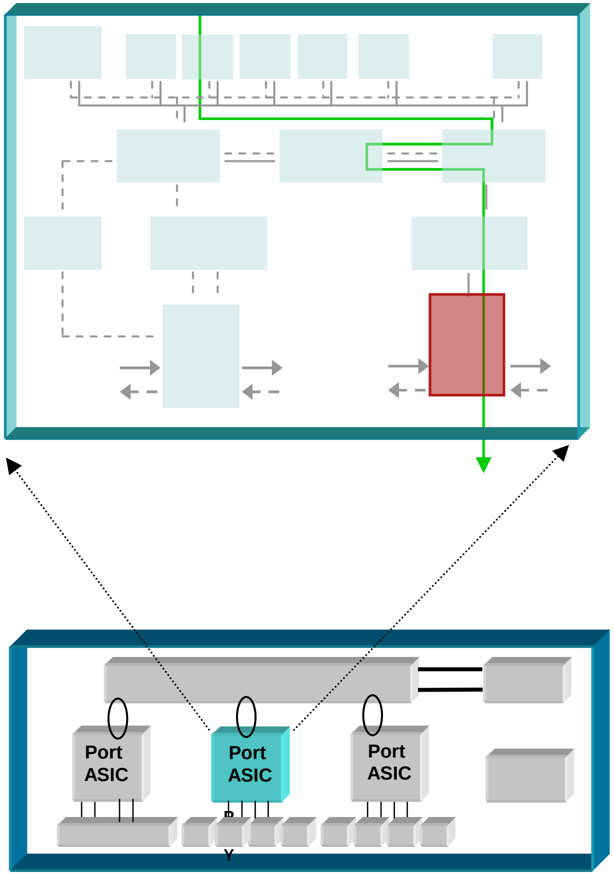
    Is shared (common) between all flows

    Minimum buffer space can be configured to makes sure ports are not buffer starved

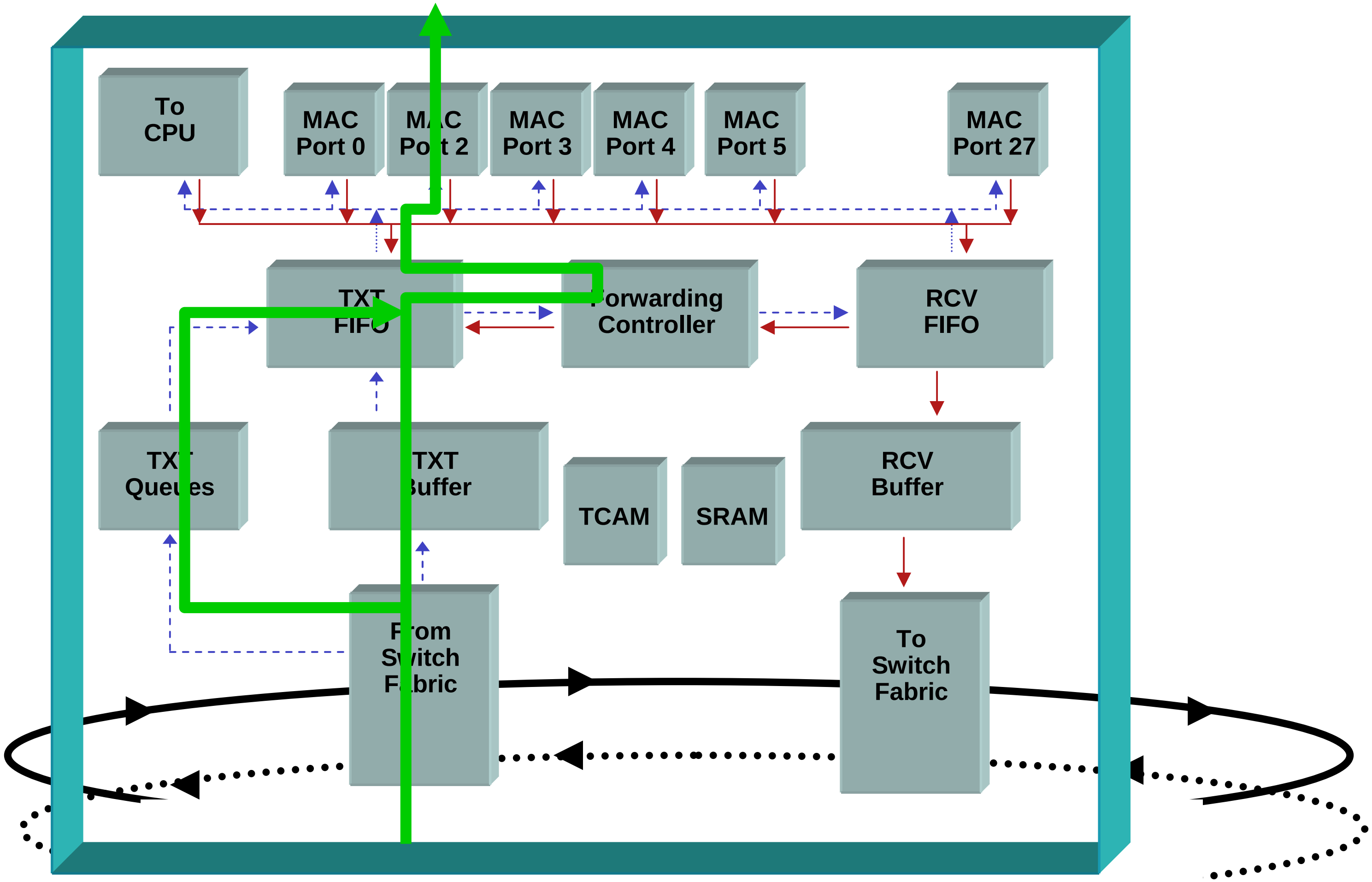


Ingress Flow: Ring Insert

- At this point the port ASIC sends the packet to the Switch Fabric via a point-to-point ring connection.
- This ring is a P2P and is not the same thing as the shared stack ring.

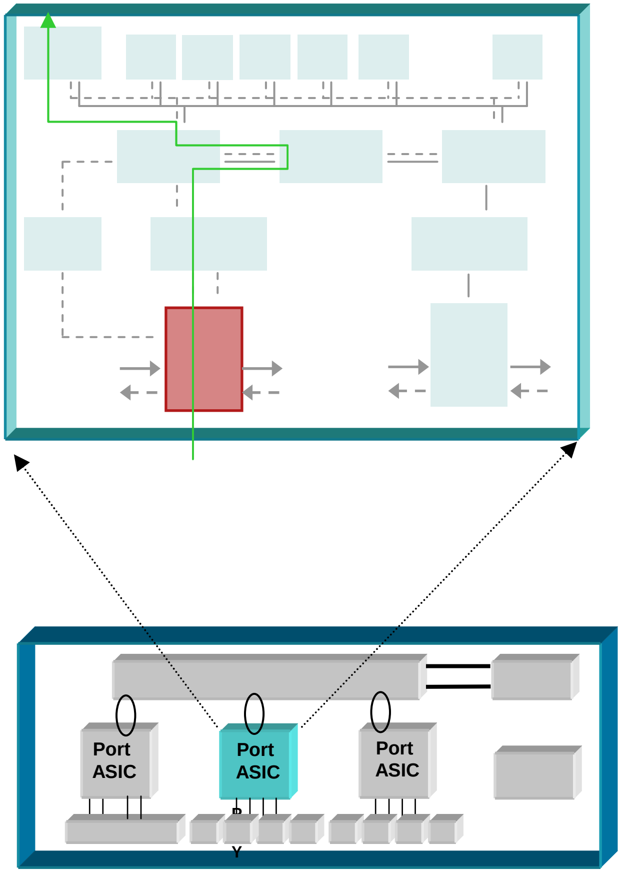


Egress Flow


To CPU
MAC Port 0
MAC Port 2
MAC Port 3
MAC Port 4
MAC Port 5
MAC Port 27
TXT FIFO
Forwarding Controller
RCV FIFO
TXT Queues
TXT Buffer
TCAM
SRAM
RCV Buffer
From Switch Fabric
To Switch Fabric

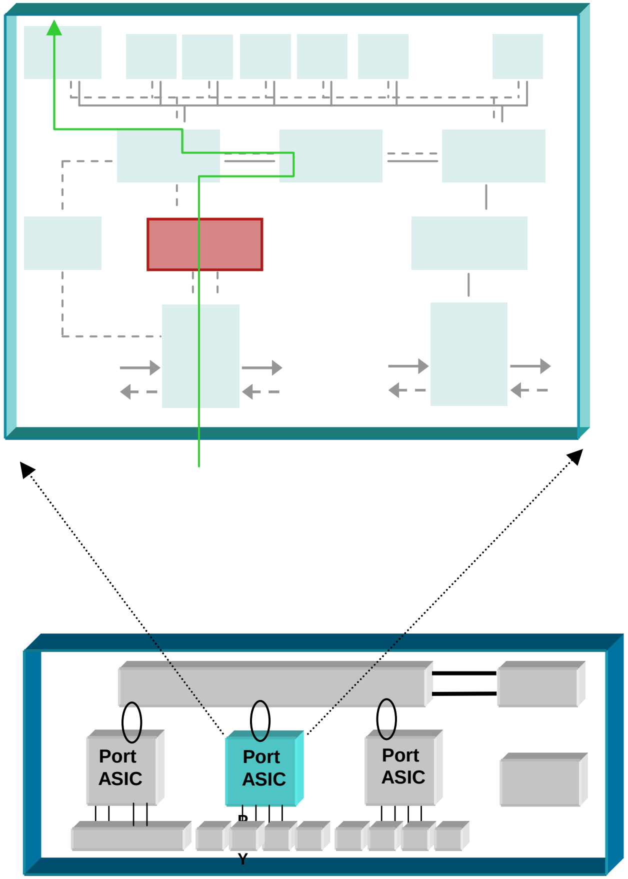
Egress Flow: Ring Copy

- At this point the packet enters the Port ASIC from the point-to-point ring that connects the port ASIC to the Switch Fabric



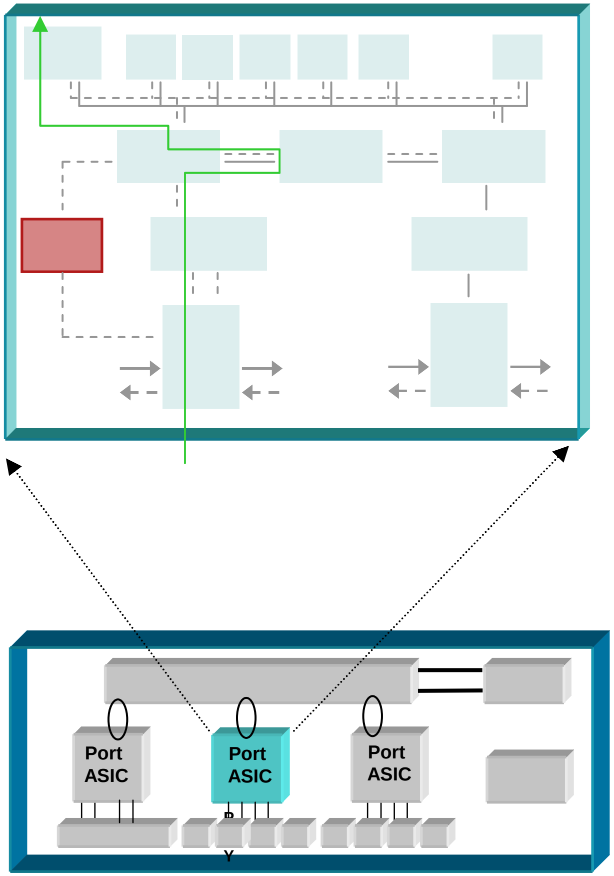
Egress Flow: TXT Buffer

- At this point the TXT queues control what happens to the packets in the TXT buffer
- The TXT buffer performs packet drops



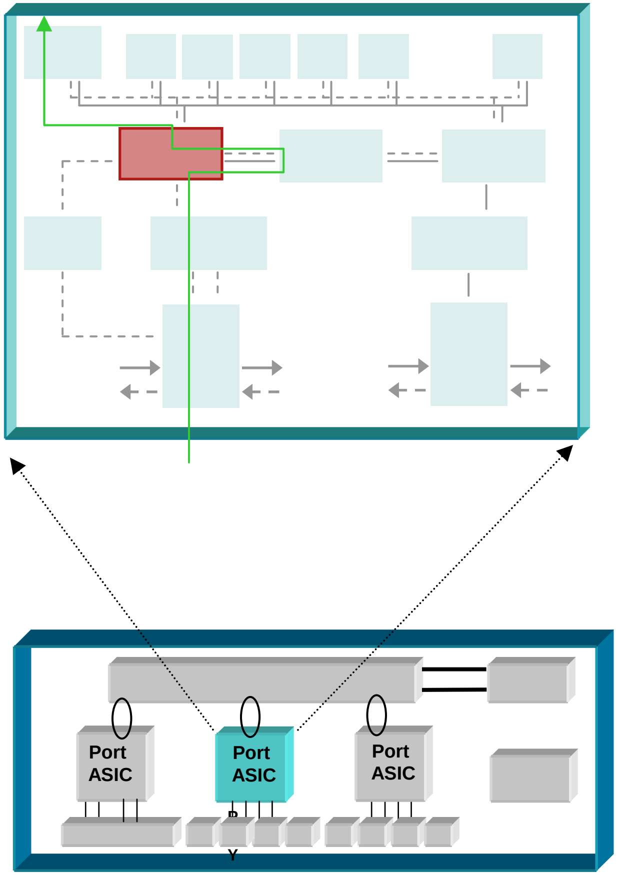
Egress Flow: TXT Queues

- There are four queues per MAC port
- Each queue is highly programmable
- The queues are scheduled with SRR and are susceptible to WTD
- Each buffer:
  - Is shared (common) between all flows
  - Minimum buffer space can be configured to makes sure ports are not buffer starved
- There also are 16 queues for the CPU. Each queue is statically allocated and dedicated to a different protocol



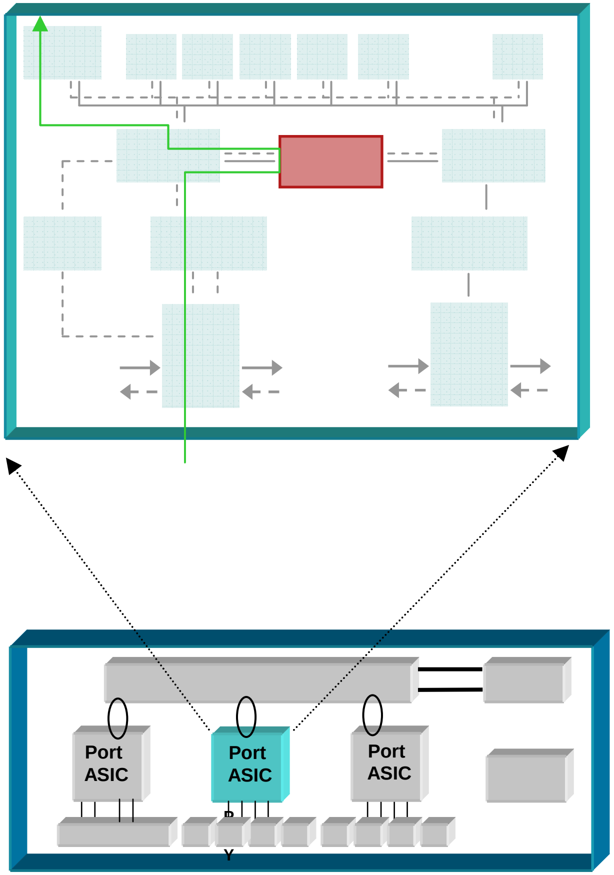
Egress Flow: TXT FIFO

- The packet enters the TXT FIFO from the TXT buffer
- There is one physical memory divided into multiple logical TXT FIFOs to serve all of the MACS on the Port ASIC
- One FIFO per port
- The TXT FIFO absorbs time so the forwarding controller to do its job



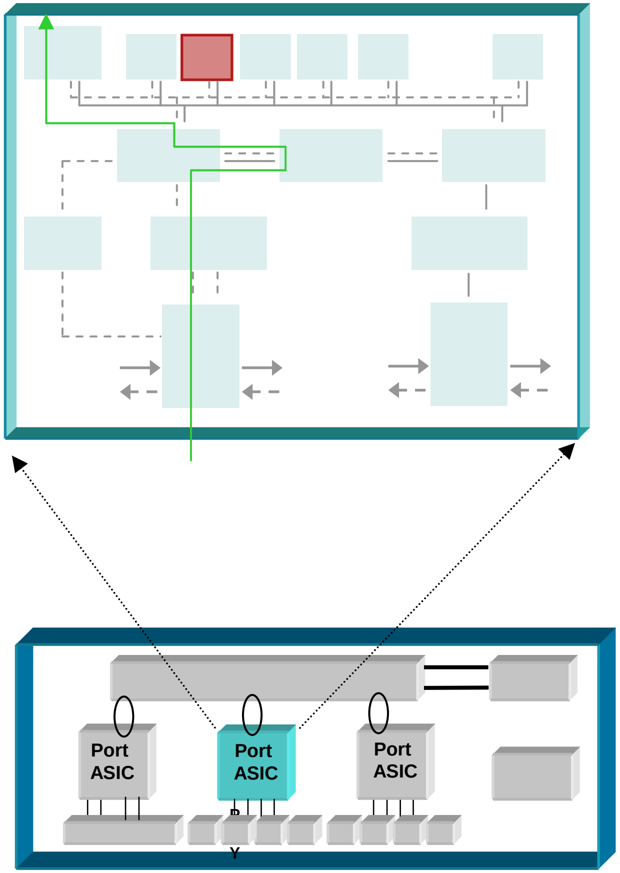
Egress Flow: Forwarding Controller

- The forwarding controller reads the 24B header + the first 200 B of the frame
- The controller performs:
  - Rewrites for the MAC header
  - Time To Live (TTL) decrements
  - Checksum calculation
  - SPAN coordination

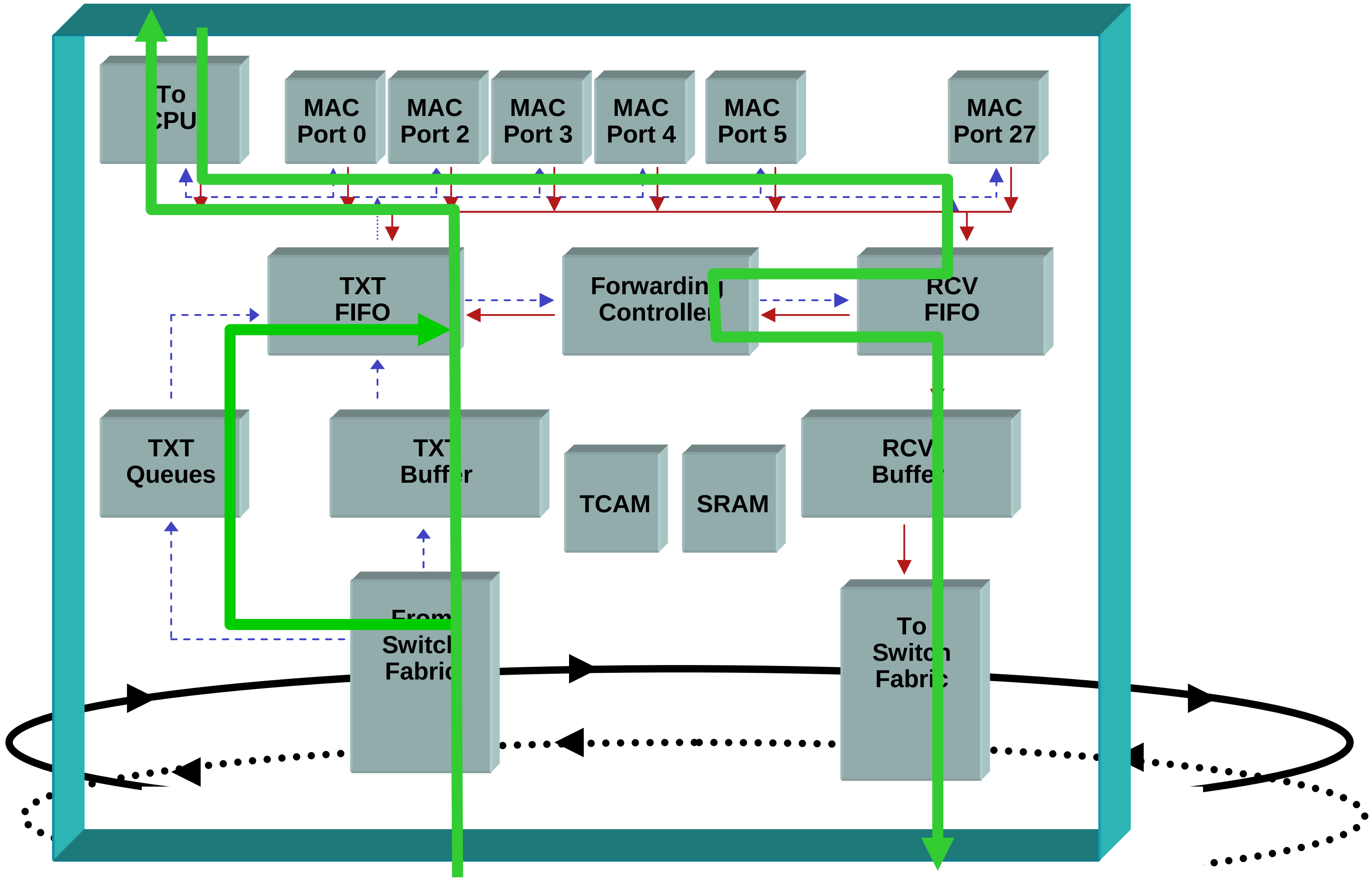


Egress Flow: MAC Port

- The packet is received from the TXT FIFO
- The MAC port function performs all Ethernet Media Access Control
- The MAC port function strips the 24B internal header
- All physical layer functionality is performed after leaving the port ASIC function
  - Encoding
  - Power over Ethernet
  - Etc.



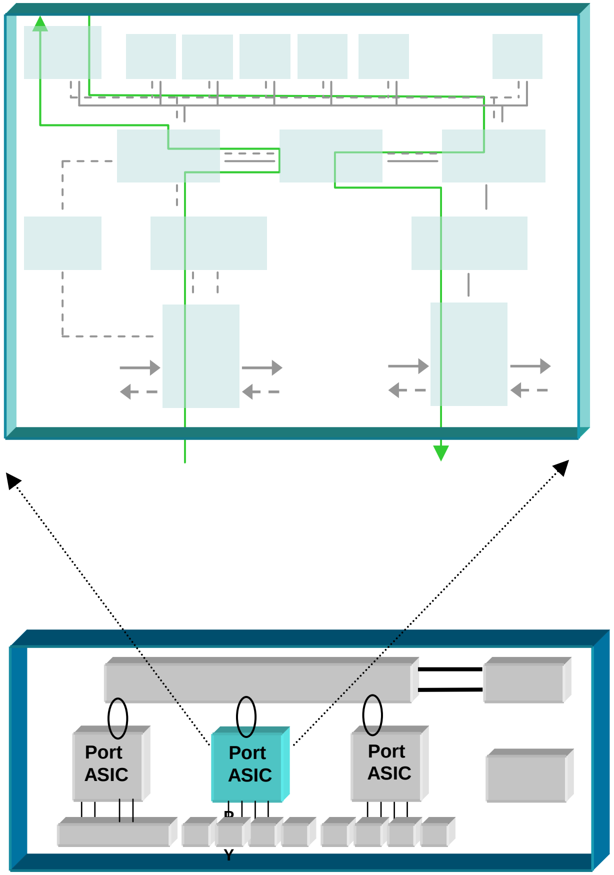
CPU Forwarded Flow



Reasons for CPU Flows

Flows Eligible for CP Forwarding Are:

- Control plane traffic
- Management traffic
- TCAM overflow traffic
  - ACL overflow
  - MAC entry overflow
  - Routing table overflow
- Special protocol flows, these are typically low volume and unofficially supported



CPU Flows: To the CPU

- To hit the CPU the packet must first enter the system
- The packet follows the typical egress path, because the CPU is treated like any other port

From Switch Fabric

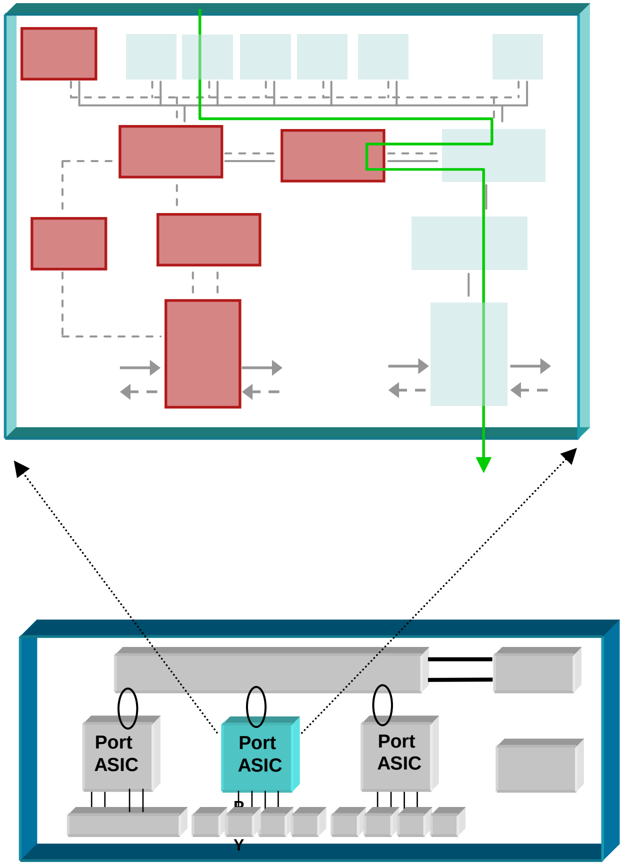
TXT buffer

TXT queues

TXT FIFO

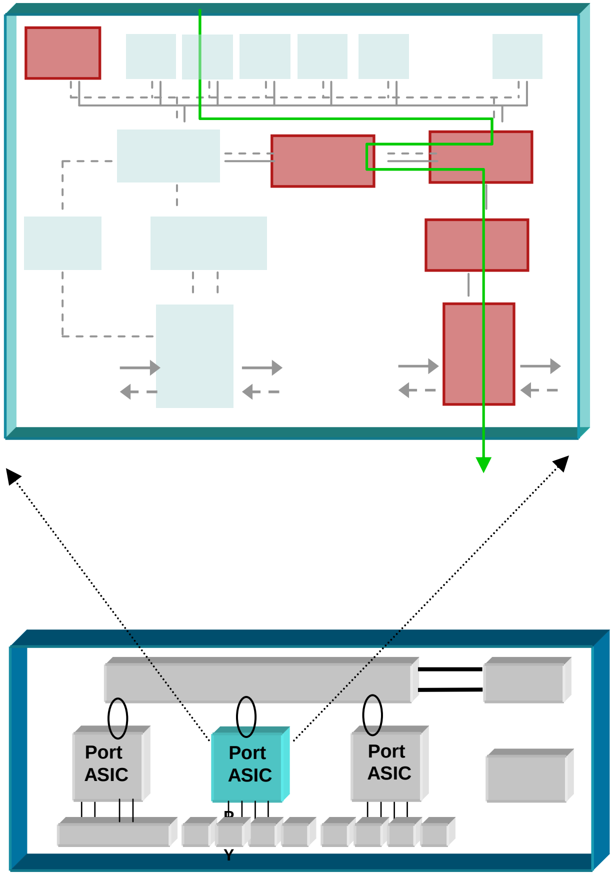
Forwarding controller

Off of the Port ASIC to the CPU



CPU Flows: Reentry

- The packet returns to the Port ASIC from the CPU and then follows the typical ingress path
  - RCV FIFO
  - Forwarding controller
  - TXT buffer
  - Switch Fabric
- After this it follows the transmit path to its destination port


Port ASIC
Port ASIC
Port ASIC