
TOMORROW
starts here.
CISCO
Cisco live!

# IOS Routing Internals

BRKARC-2350

Richard Farquhar – CCIE R&S #1636

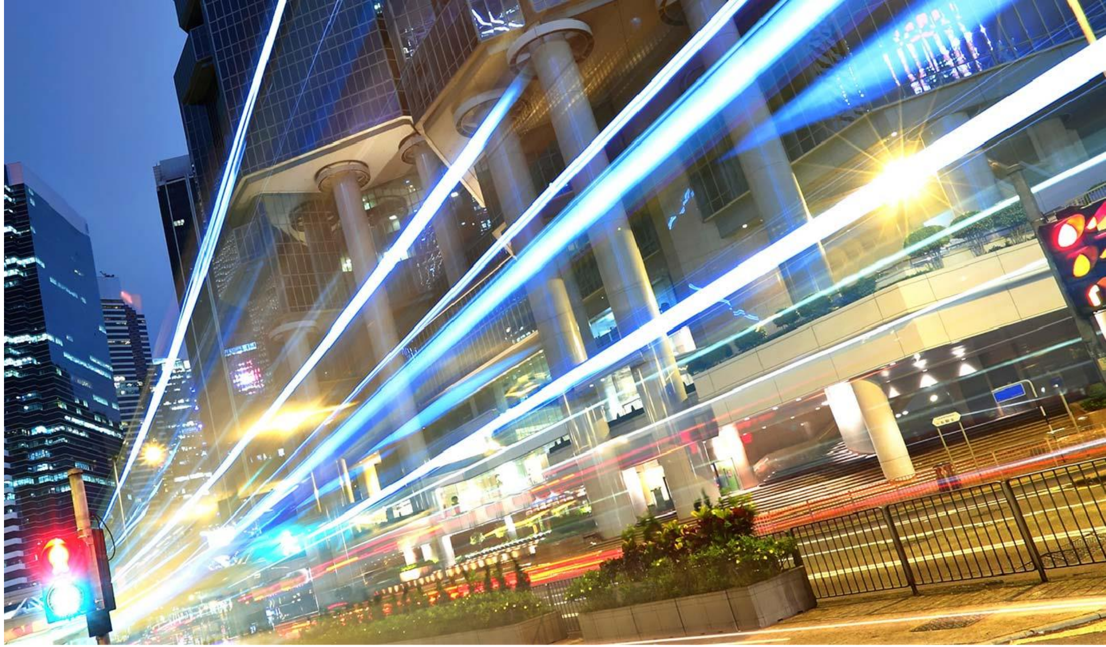
Network Consulting Engineer

#clmel

Cisco live!

# Agenda

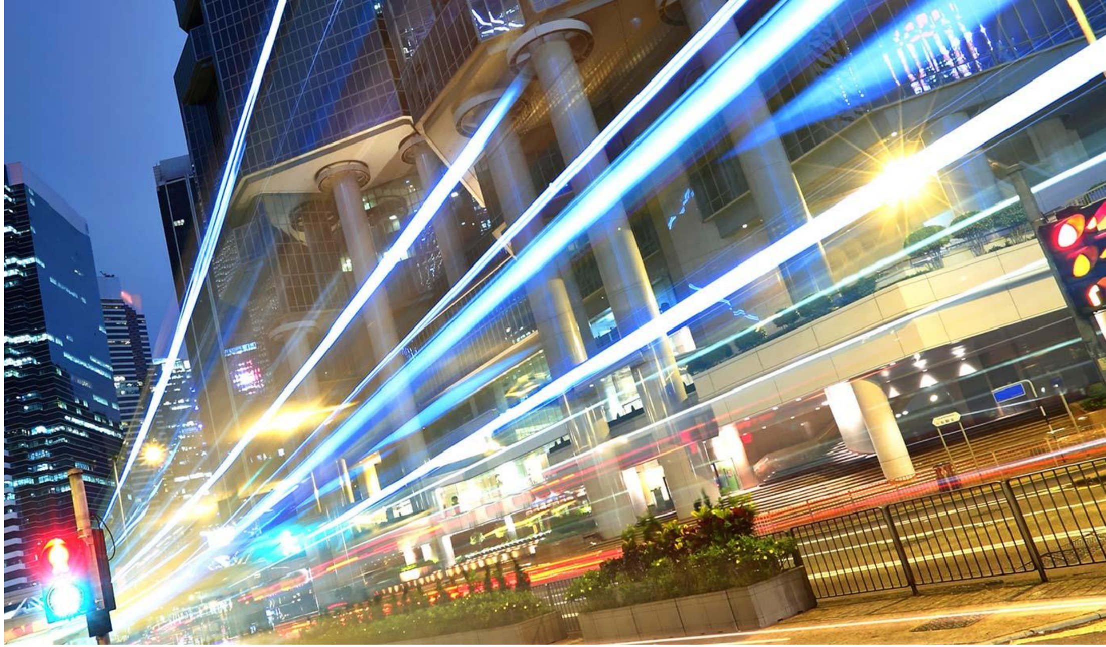
- Router Components
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing
- Routing Convergence Improvements



# Agenda

- ➢ **Router Components**
  - ➢ **Data and Control Planes**
  - – Software Based Routers
  - – Hardware Based Routers
  - – Hybrid Routers
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing
- Routing Convergence Improvements



# Router Components

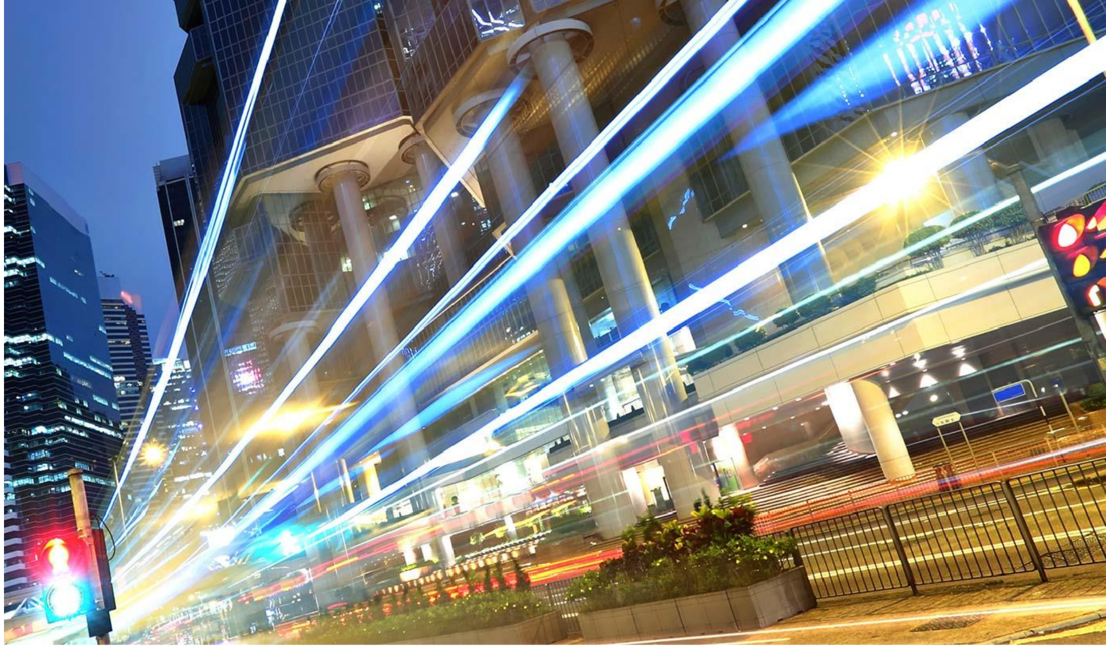
## Data and Control Planes

- Control Plane ← **Brains**
  - Control Traffic
    - Routing Updates (BGP, EIGRP, OSPF, etc.)
    - SSH
    - SNMP

- Data Plane ← **Brawn**
  - Through traffic

# Agenda

- ➢ Router Components
  - – Data and Control Planes
  - ➢ Software Based Routers
  - – Hardware Based Routers
  - – Hybrid Routers
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing
- Routing Convergence Improvements



# Router Components

## Software Based Routers

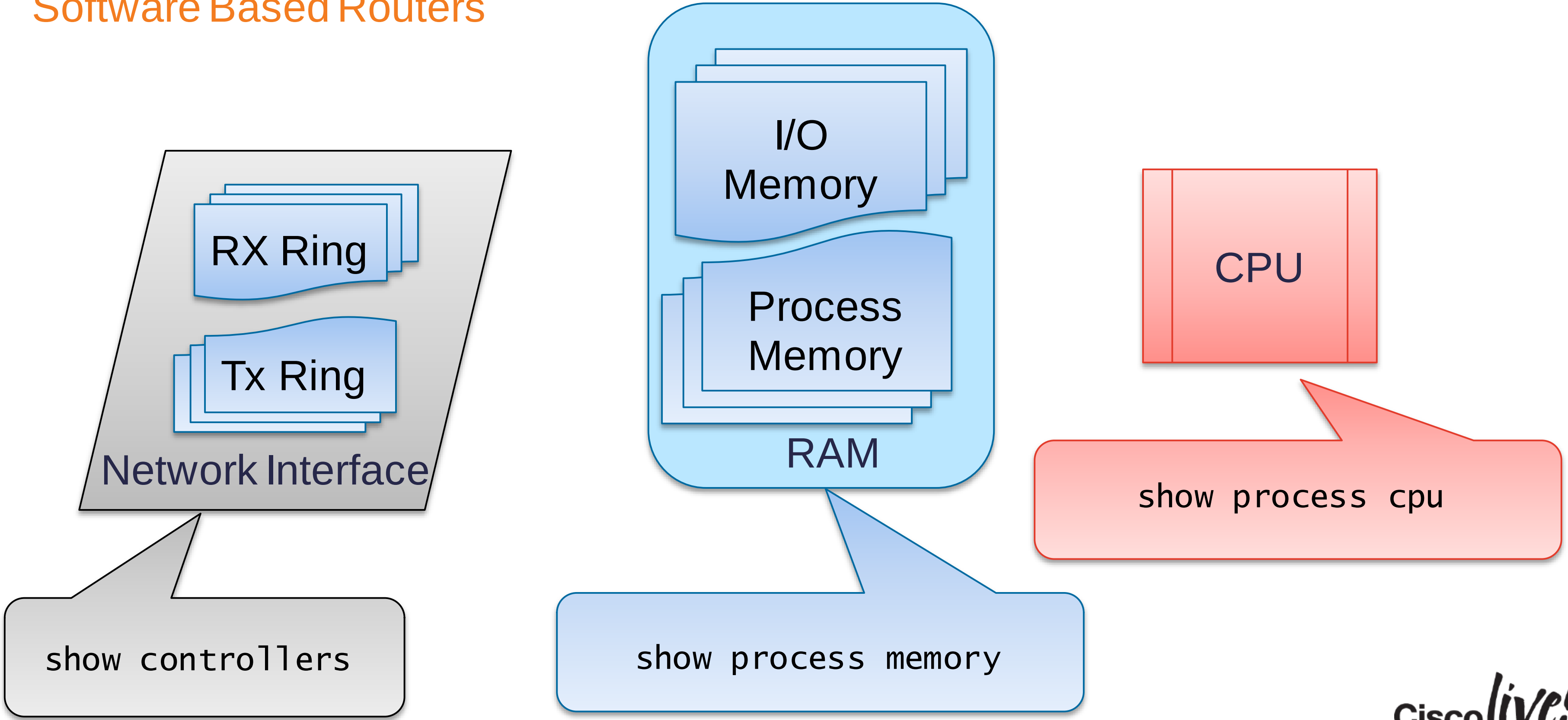
- Software Based
  - **Shared** control and data plane
  - General Purpose CPU (slow and smart)
    - Runs at CPU speed
    - Speed/flexibility tradeoff
  - CPU responsible for all operations

800/2800/2900/3900/7200 Series Routers are software based

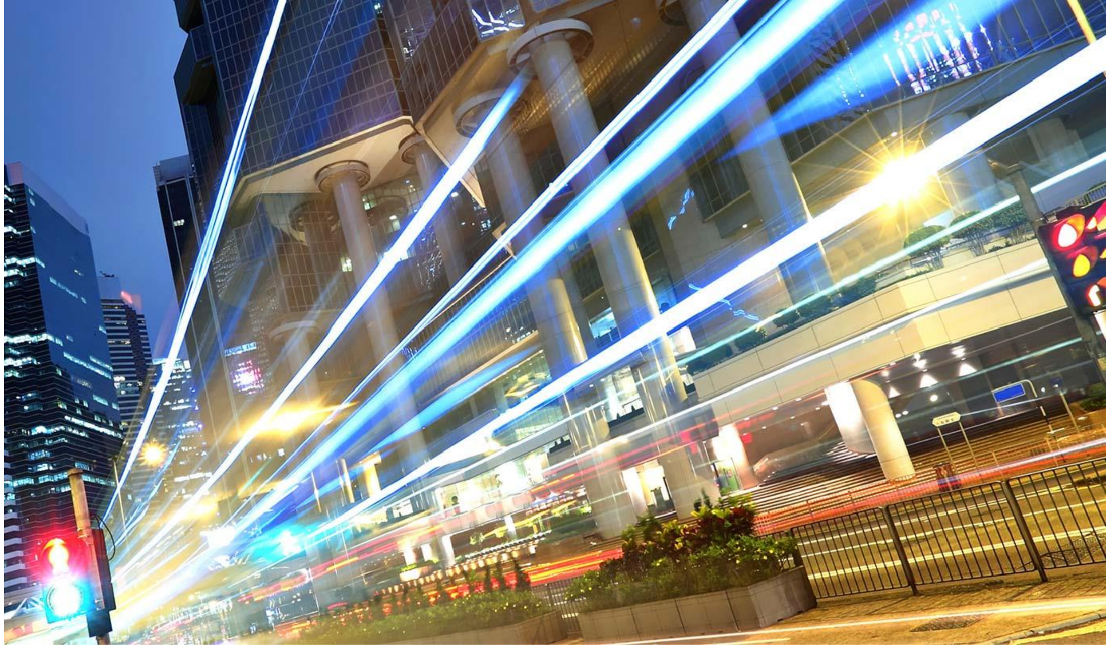
# Router Components

## Software Based Routers



# Agenda

- **Router Components**
  - Data and Control Planes
  - Software Based Routers
  - **Hardware Based Routers**
  - Hybrid Routers
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing
- Routing Convergence Improvements



# Router Components

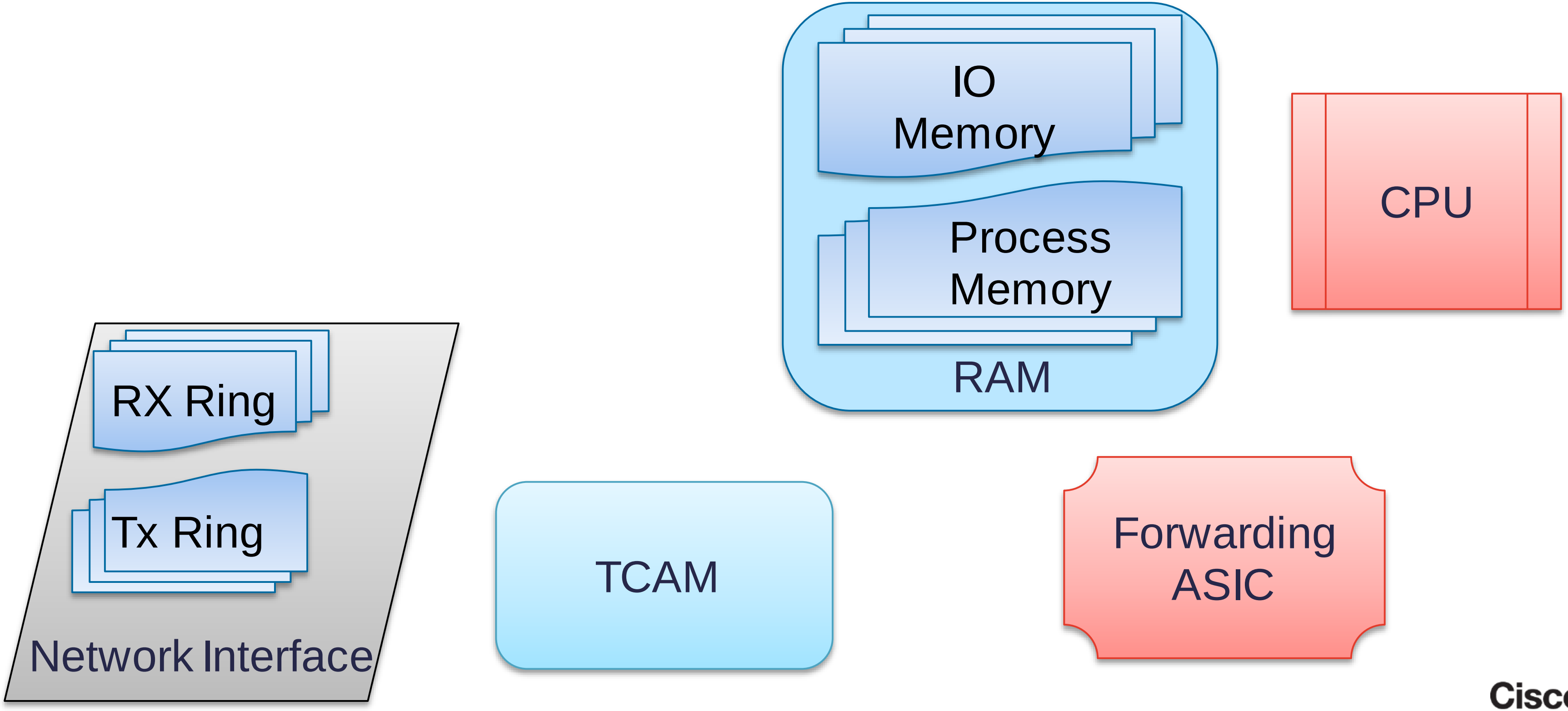
## Hardware Based Routers

- Hardware based
  - **Separated** control and data plane
  - CPU + ASIC (Application Specific Integrated Circuit)
  - ASIC designed specifically to move packets (fast and dumb)
  - CPU manages control plane
  - CPU only moves packets the ASIC can't
  - Data Plane packets sent to the CPU are "punted"

6500/7600, Nexus 7000 and ASR9000 are hardware based

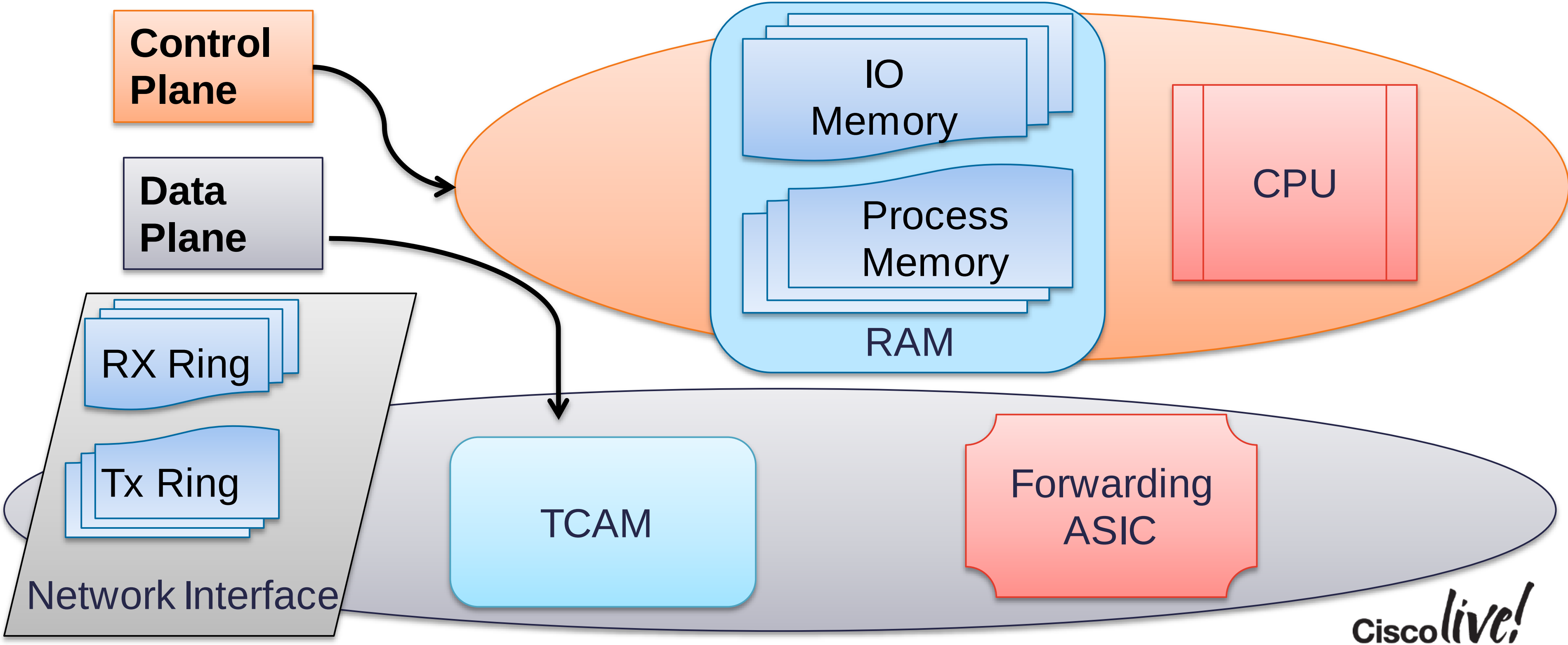
# Router Components

## Hardware Based Routers



# Router Components

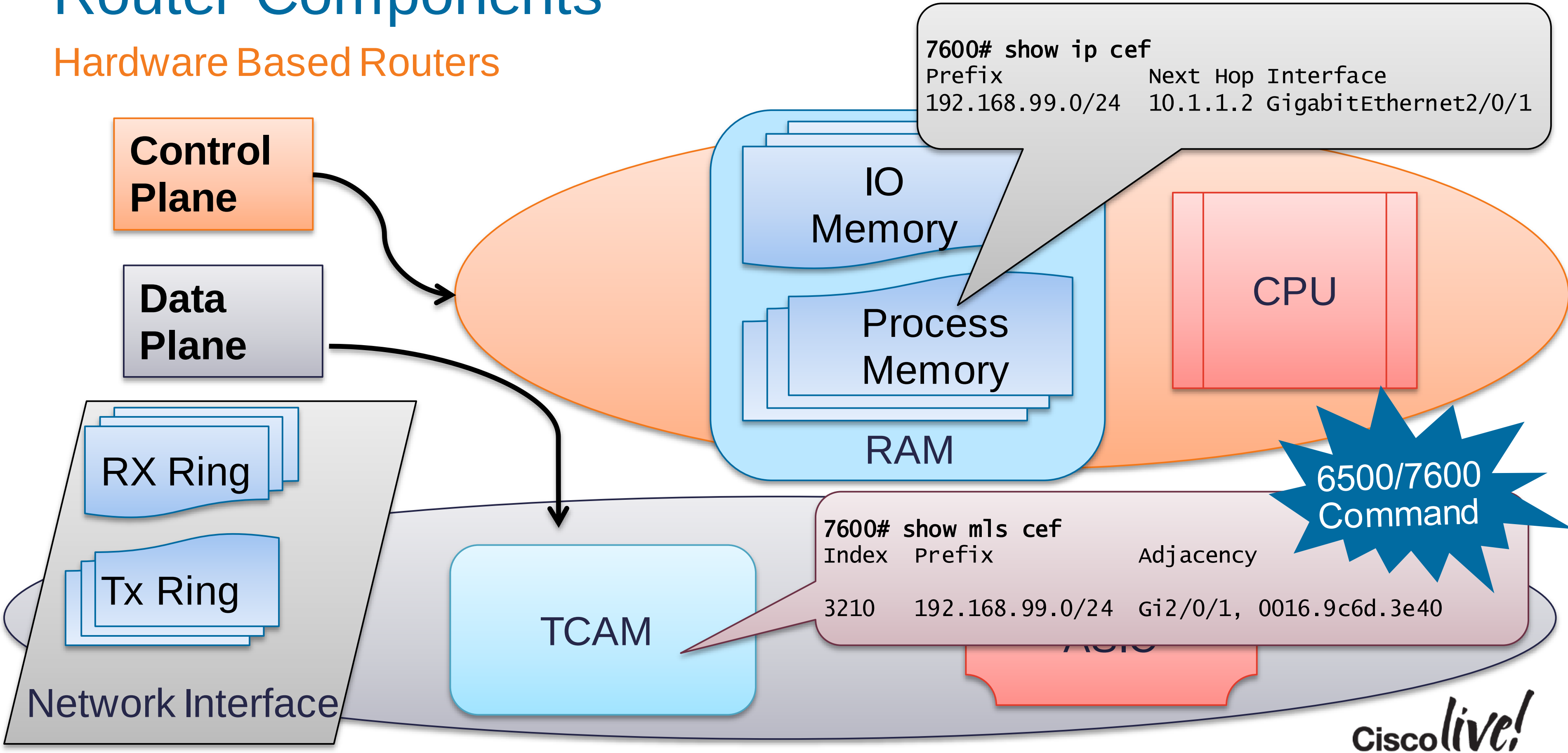
## Hardware Based Routers



# Router Components

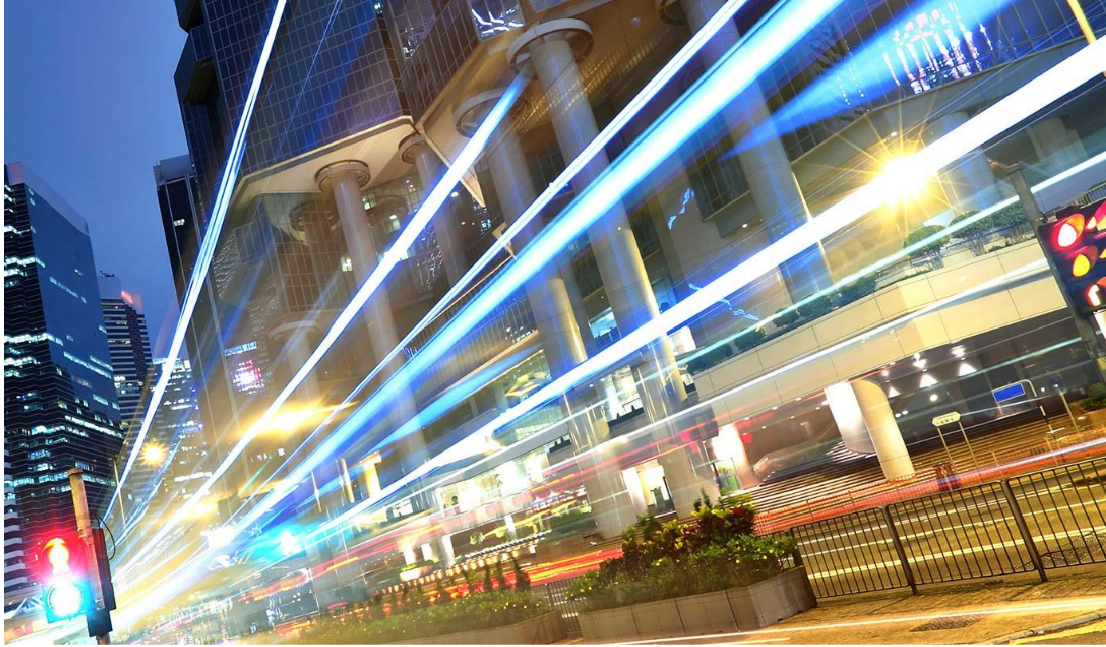
## Hardware Based Routers



# Agenda

- **Router Components**
  - Data and Control Planes
  - Software Based Routers
  - Hardware Based Routers
  - **Hybrid Routers**
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing
- Routing Convergence Improvements



# Router Components

## Hybrid Routers

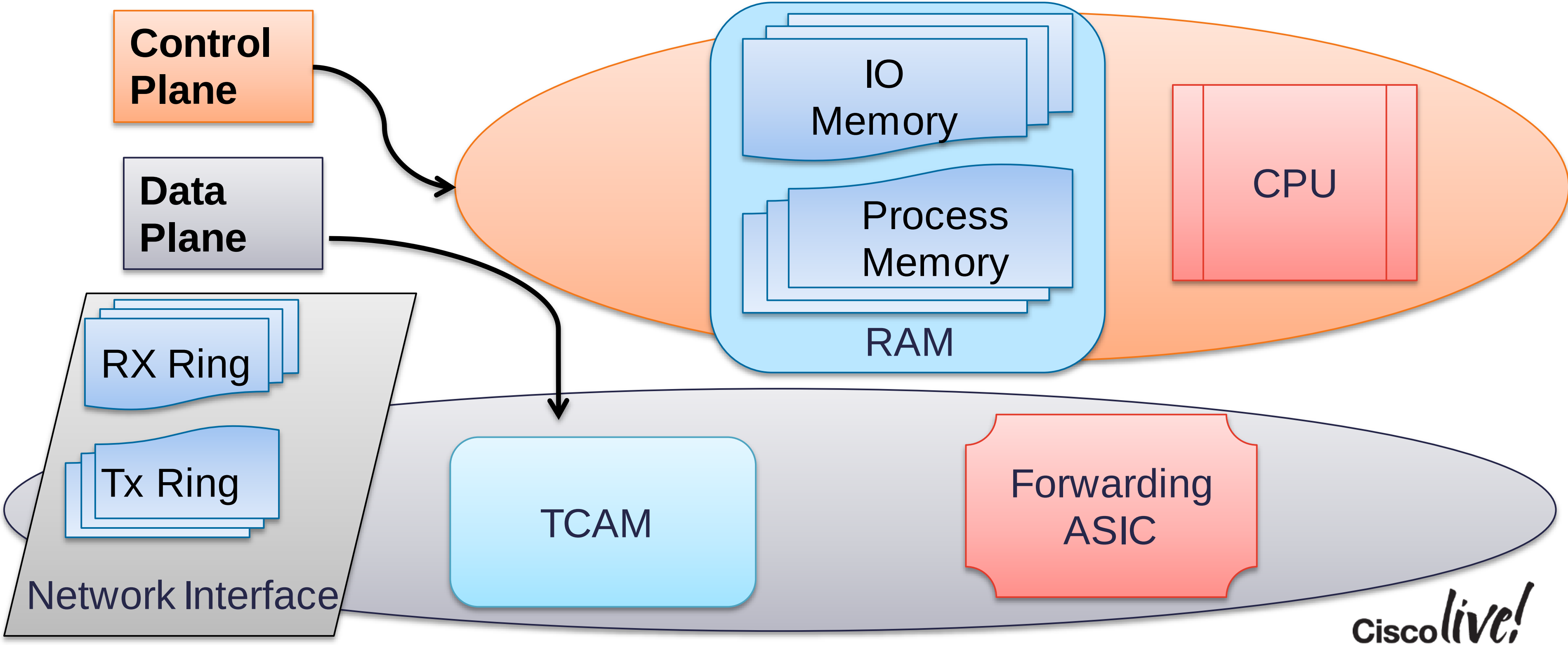
- Hardware assisted
  - **Separated** control and data plane
  - CPU + NP (Network Processor)
  - NP is multi-core specialised processor
  - NP is optimised to move packets
  - CPU manages control plane
  - CPU only moves packets the NP can't


ASR1000 and ISR4400 are Hardware Assisted Routers

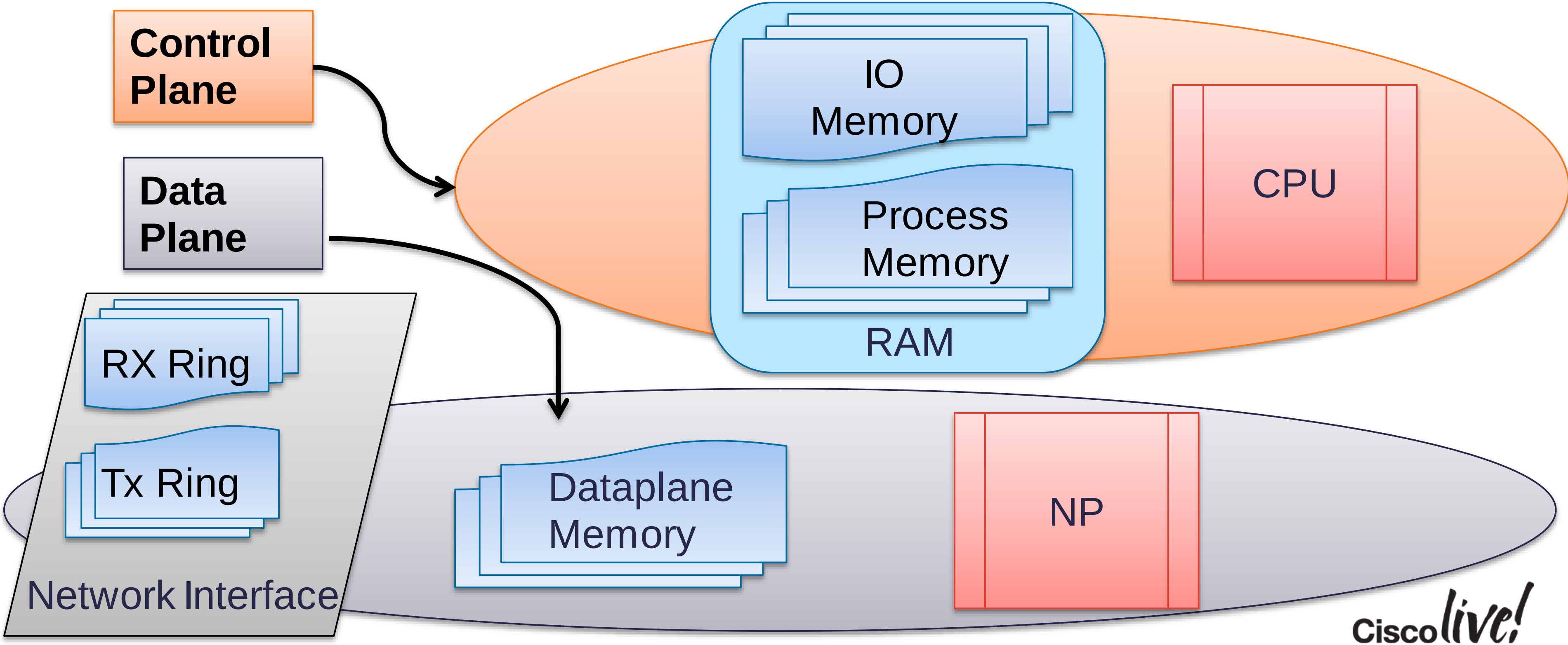
# Router Components

## Hybrid Routers



# Router Components

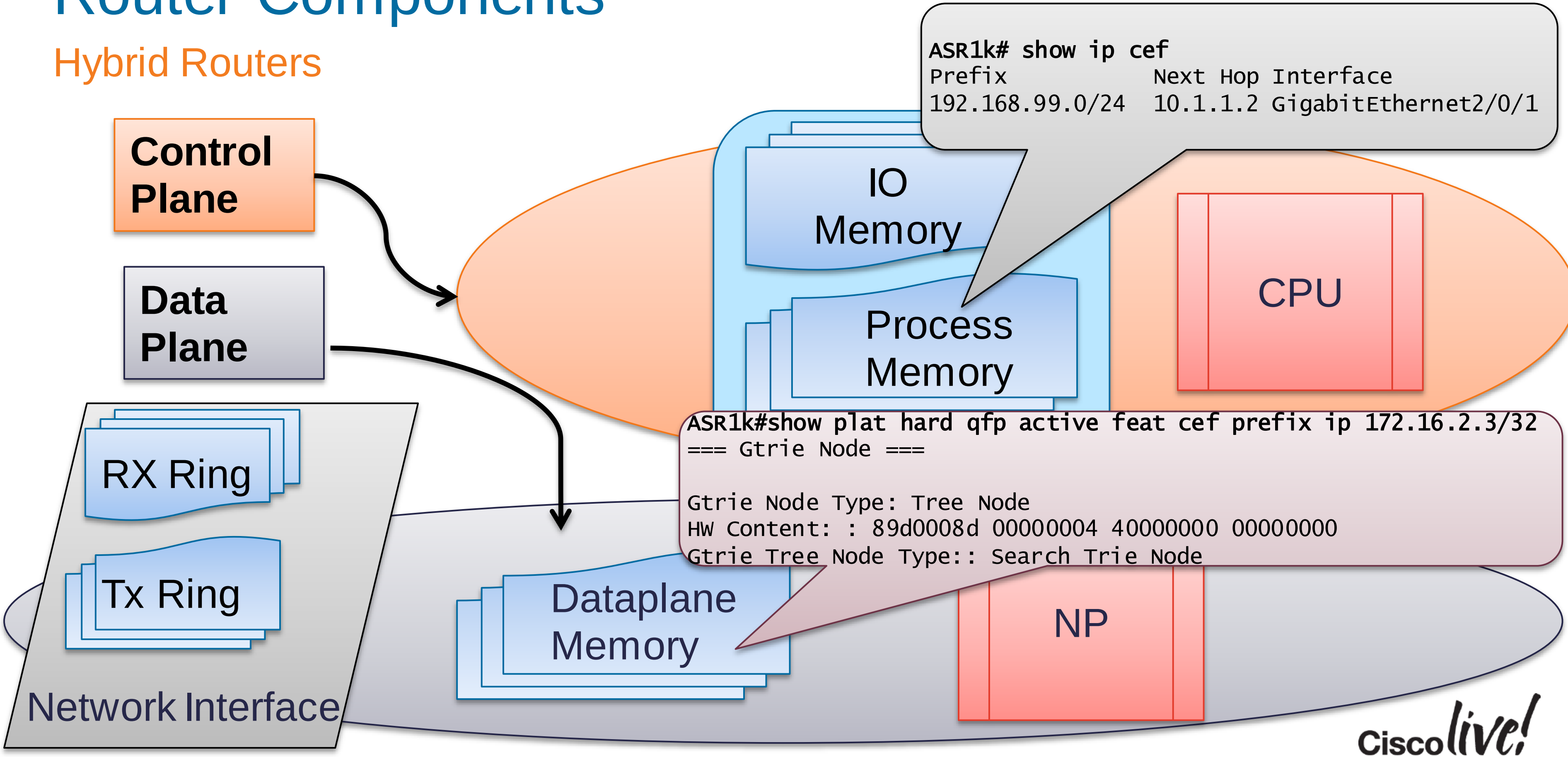
## Hybrid Routers



# Router Components

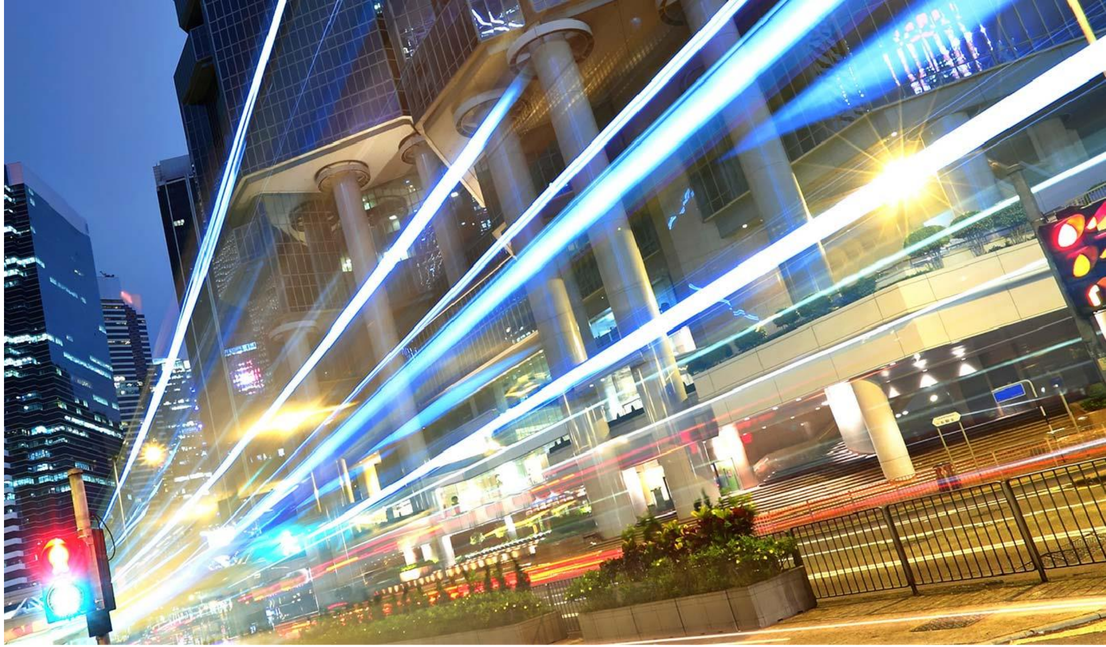
## Hybrid Routers



# Agenda

- Router Components
- ➢ **Moving Packets**
  - ➢ **Process Switching**
  - – CEF Switching
- CEF, CPU and Memory
- Outbound Load Sharing
- Routing Convergence Improvements



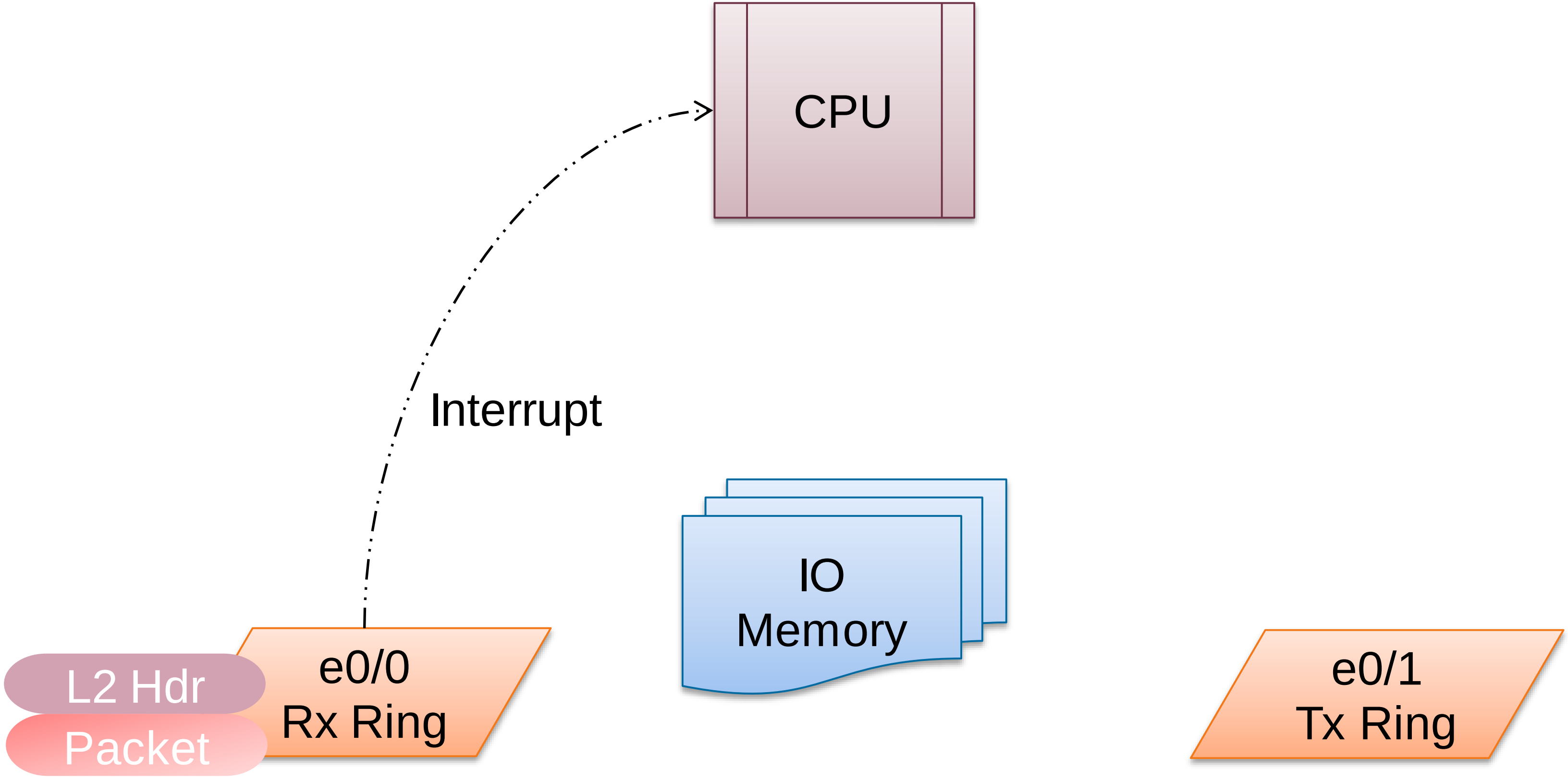
  

# Overview

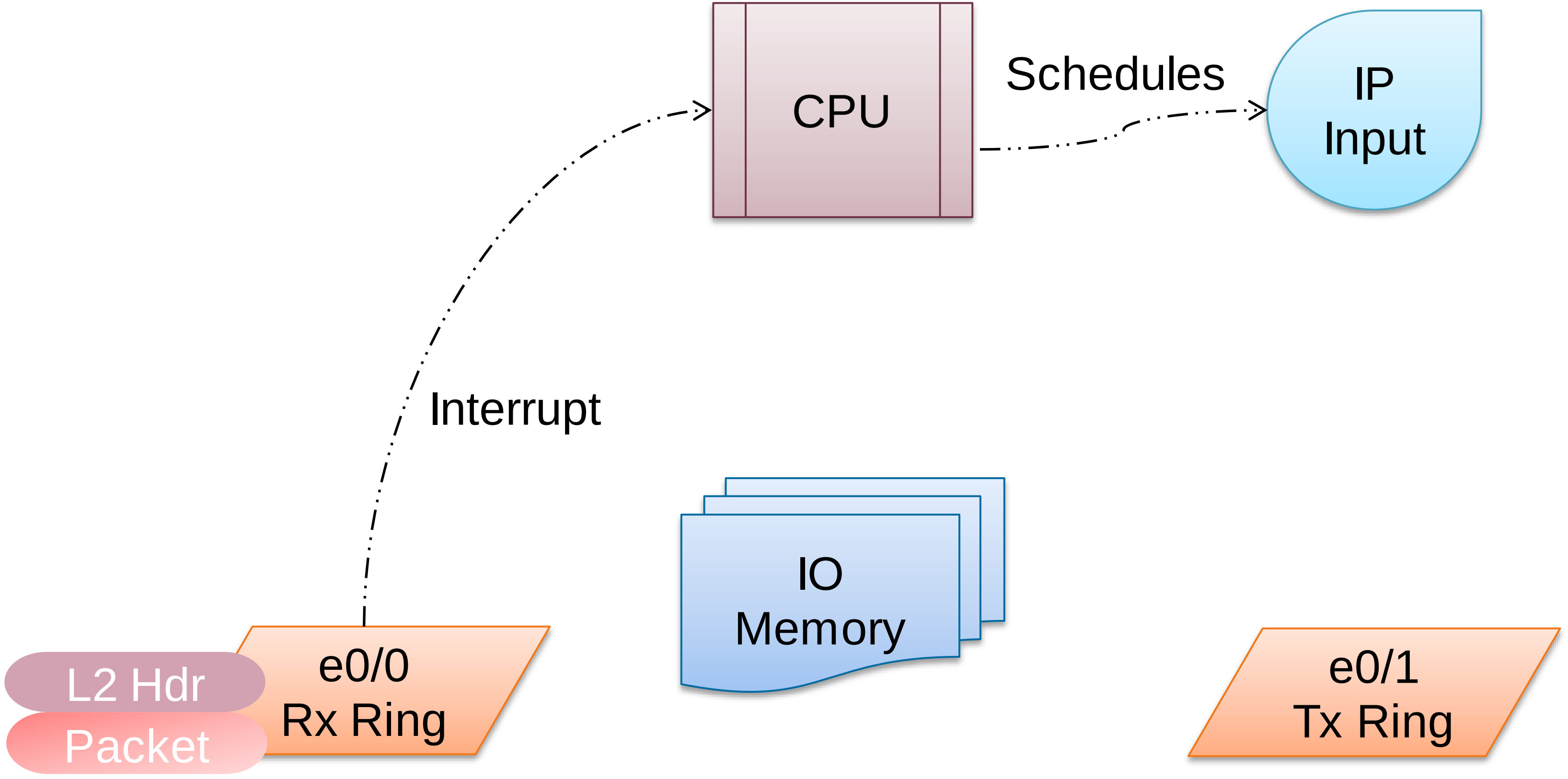
- CEF Switching and Process Switching
  - Fast Switching is deprecated as of 12.4(20)T
  - Not covered today
- CEF Switching is the default
- Process Switching is the fallback
  - Anything CEF can't handle

# Process Switching

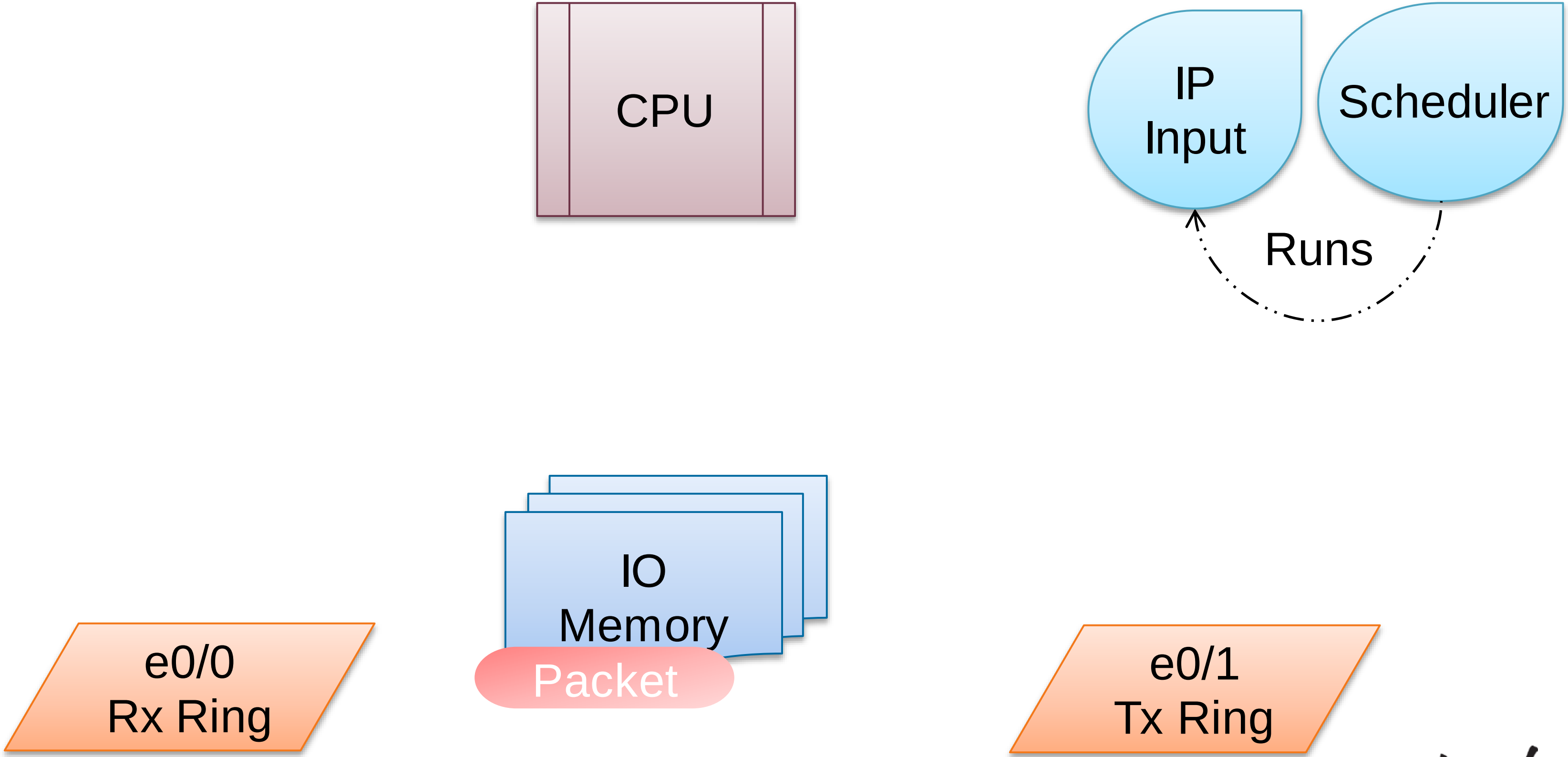


# Process Switching



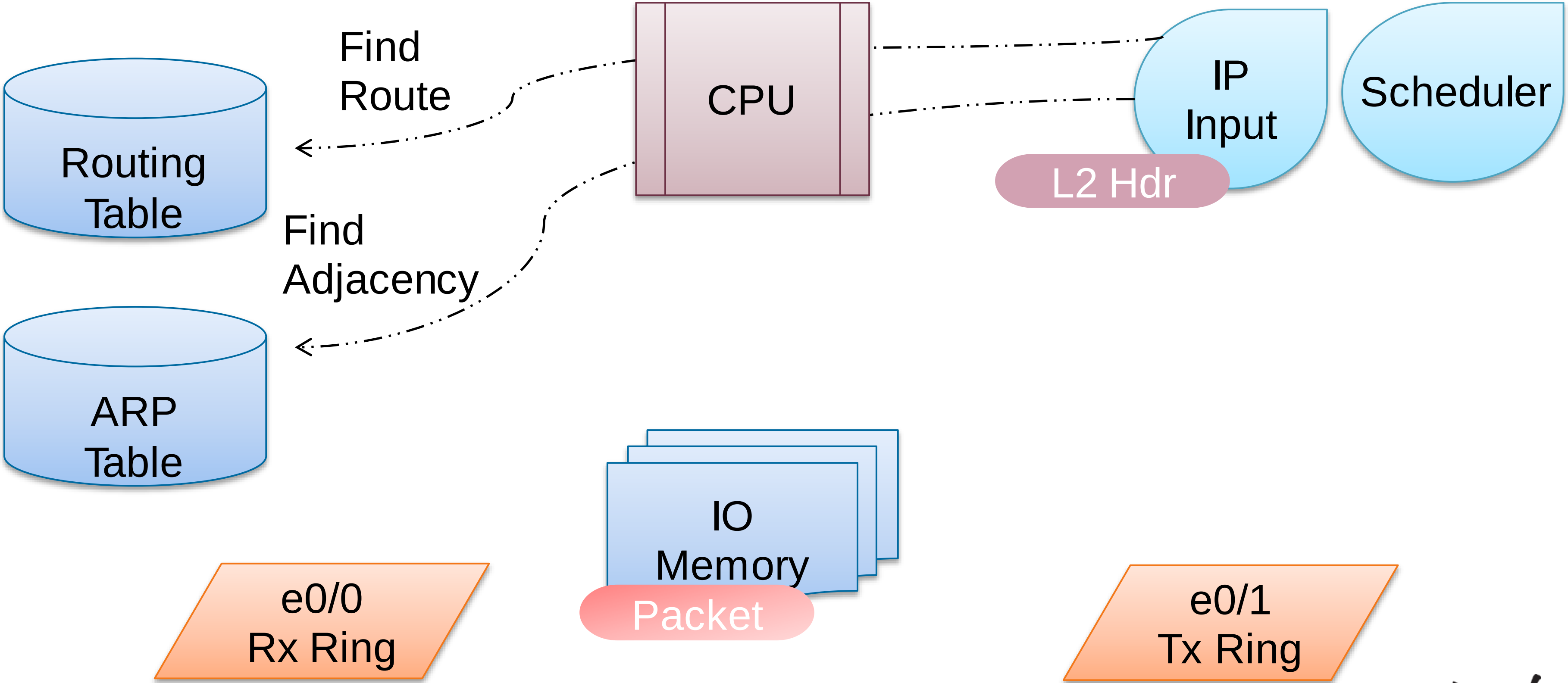

  

# Process Switching



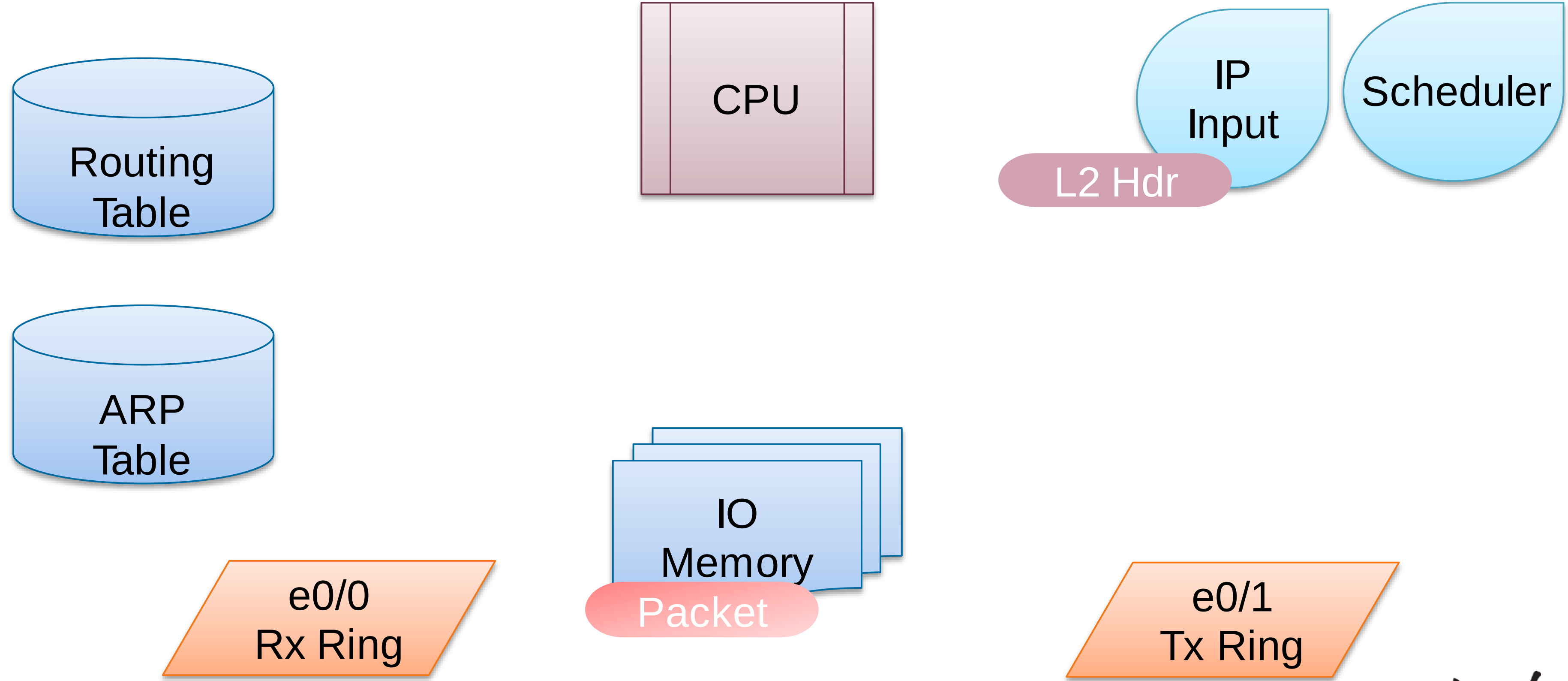

  

# Process Switching



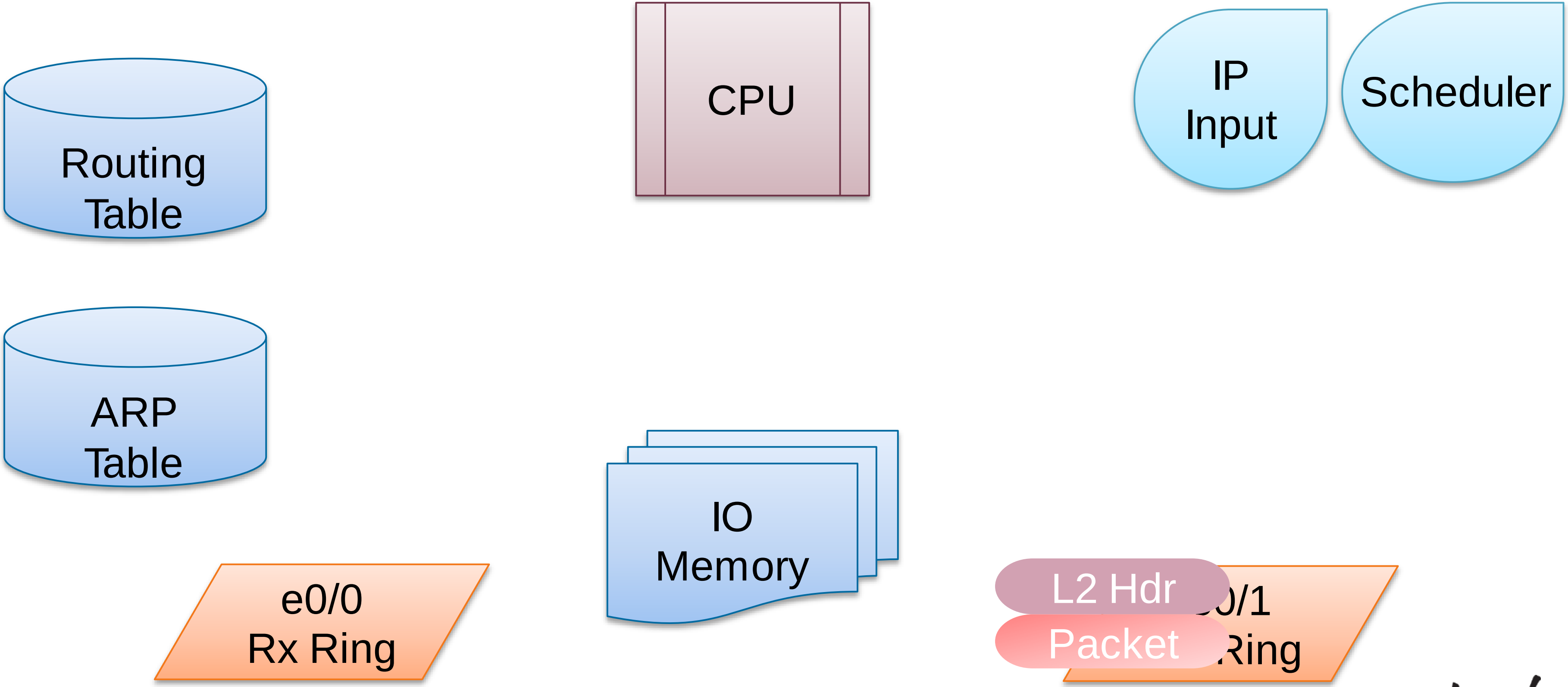

  

# Process Switching



# Process Switching



# Process Switching

- Process Switching is BAD
- Multiple lookups
- Inefficient data structures
- Process scheduling

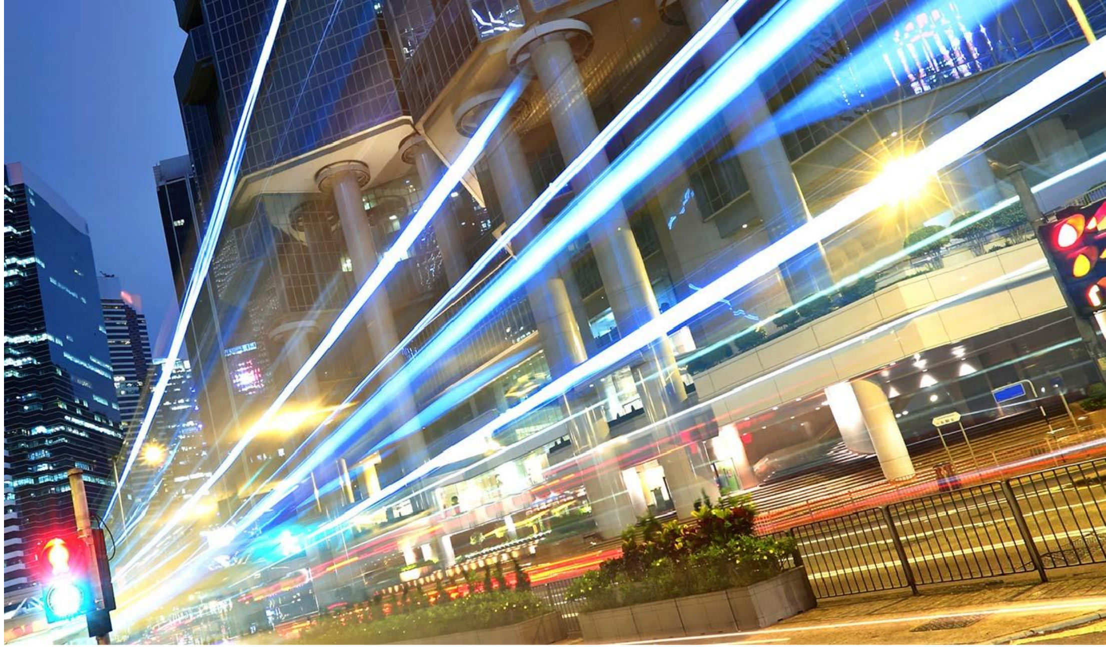
- What can we do to improve?
  - Better data structures
  - Pre-compile forwarding information

```
Router#show ip route 172.16.1.1
Routing entry for 172.16.1.1/32
  Known via "bgp 65530", distance 20, metric 0
  * 10.0.0.1, from 10.0.0.1, 00:00:07 ago
Router#show ip route 10.0.0.1
Routing entry for 10.0.0.1/32
  Known via "static", distance 1, metric 0
  * 192.168.1.1
Router#show ip route 192.168.1.1
Routing entry for 192.168.1.0/24
  Known via "connected", distance 0, metric 0
(connected, via interface)
  * directly connected, via Ethernet0/1
```

# Agenda

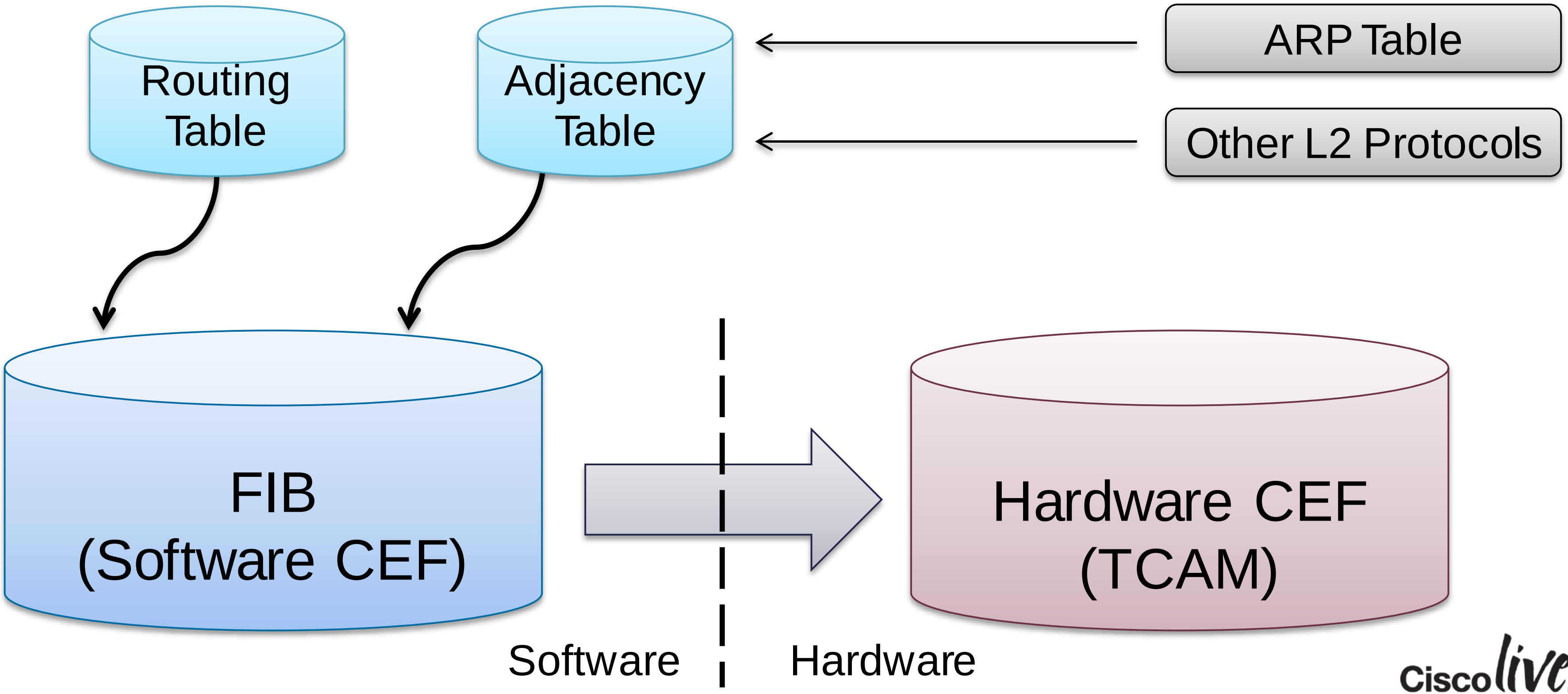
- Router Components
- ➢ **Moving Packets**
  - – Process Switching
  - ➢ **CEF Switching**
- CEF, CPU and Memory
- Outbound Load Sharing
- Routing Convergence Improvements



# The FIB (Forwarding Information Base)

## "Show IP CEF"



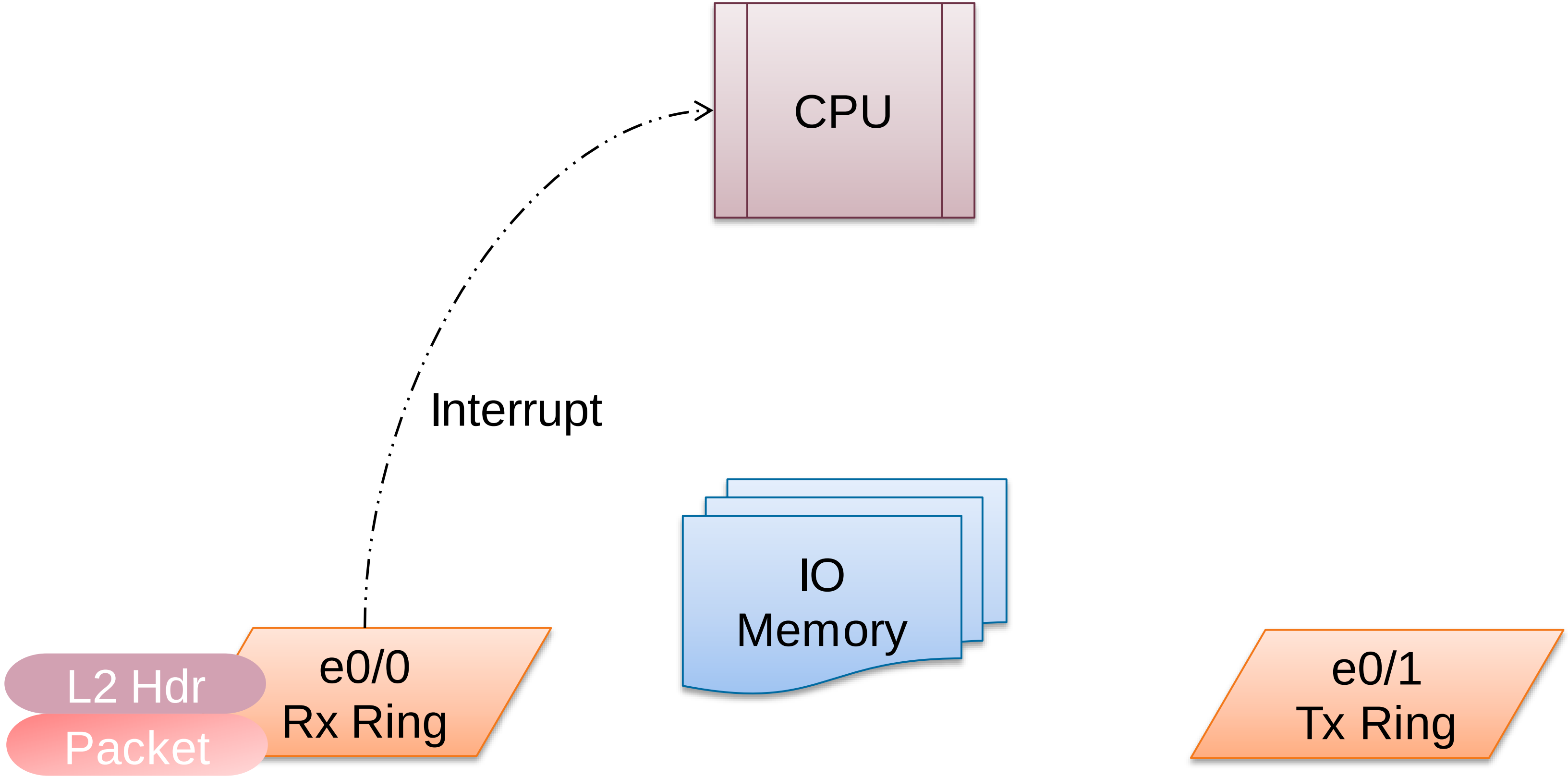
# CEF Overview

- CEF Table = Route + Egress Interface + L2 Destination
- Single lookup (and faster too!)
- No process scheduling

```
Router# show ip cef 172.16.1.1 det
172.16.1.1/32
  recursive via 10.0.0.1
    recursive via 192.168.1.1
      attached to Ethernet0/1
```

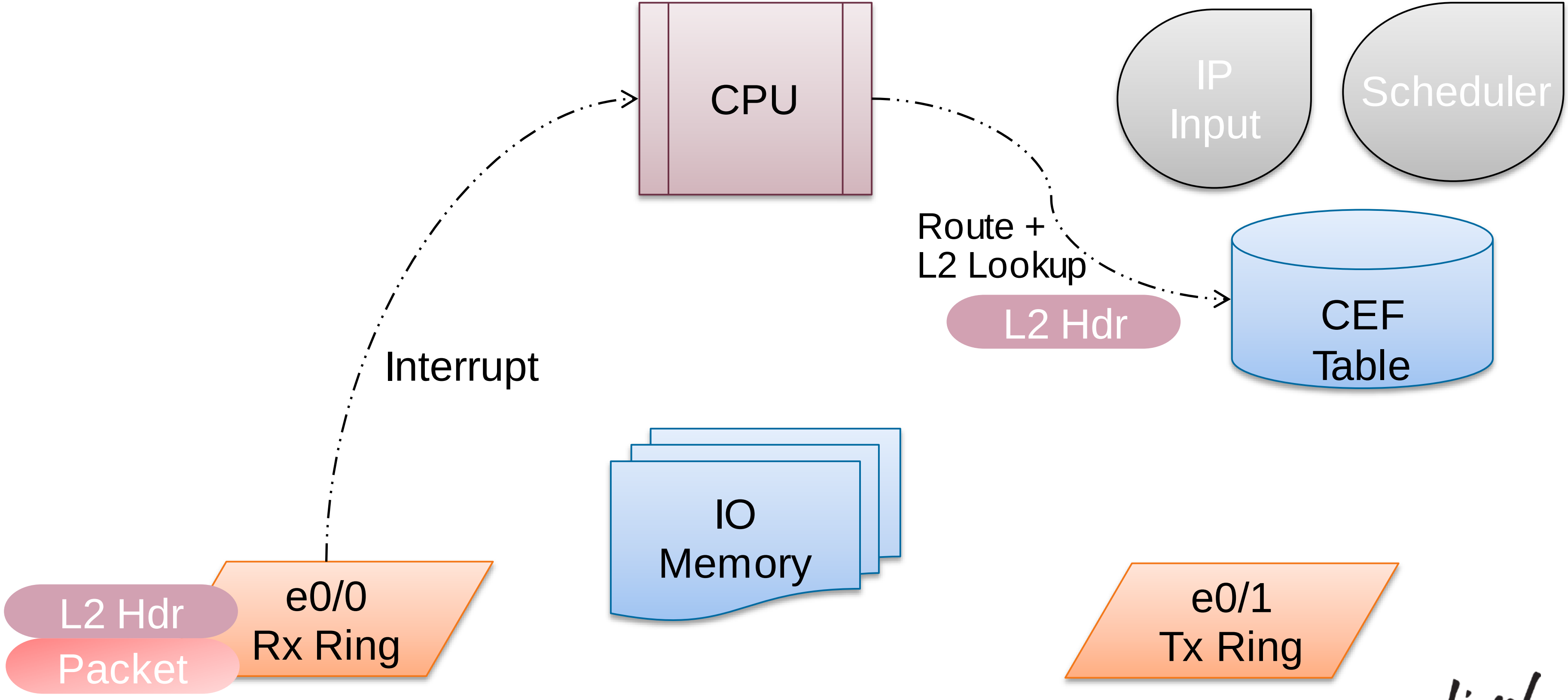
  

# CEF Switching



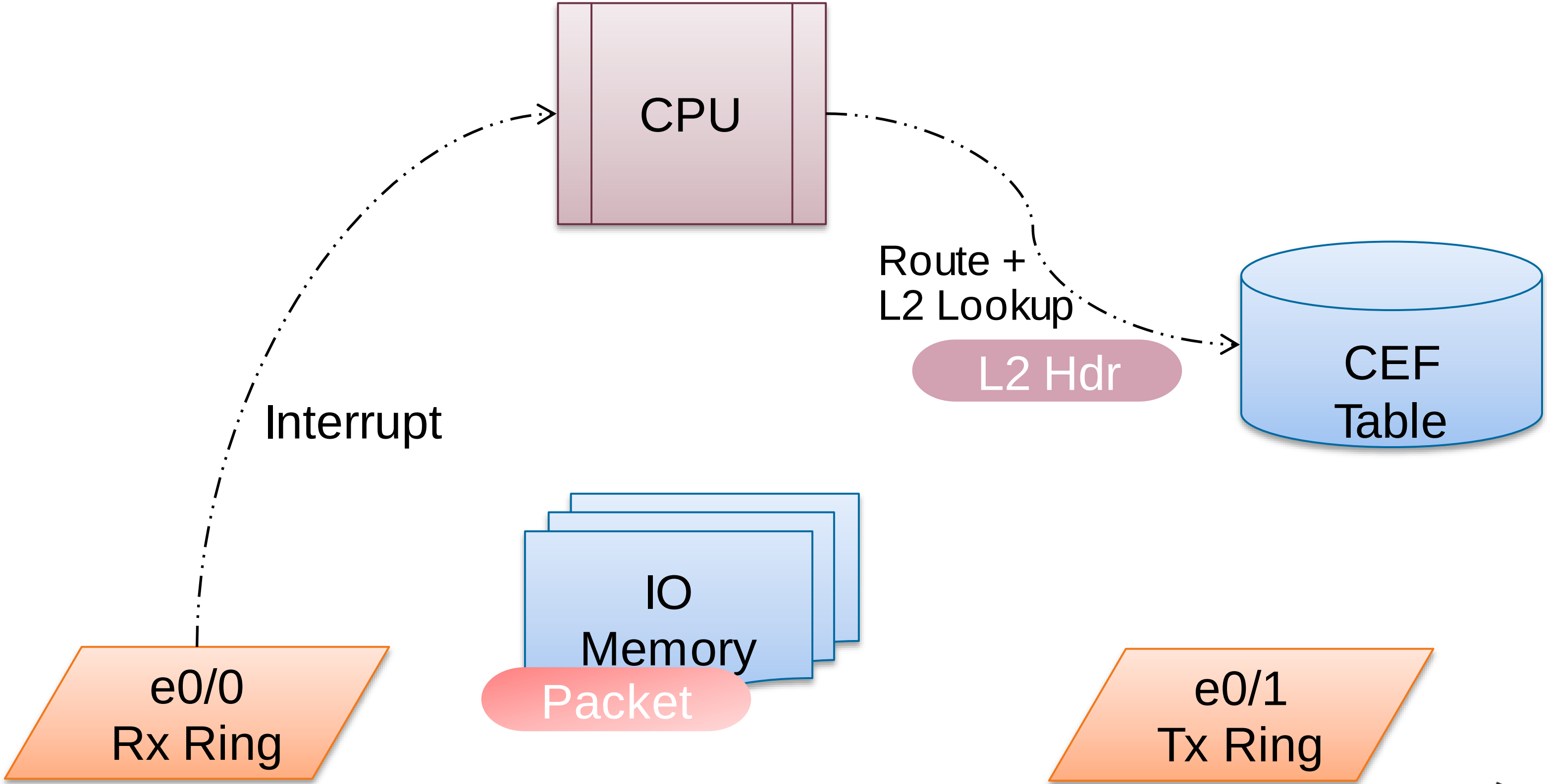

  

# CEF Switching

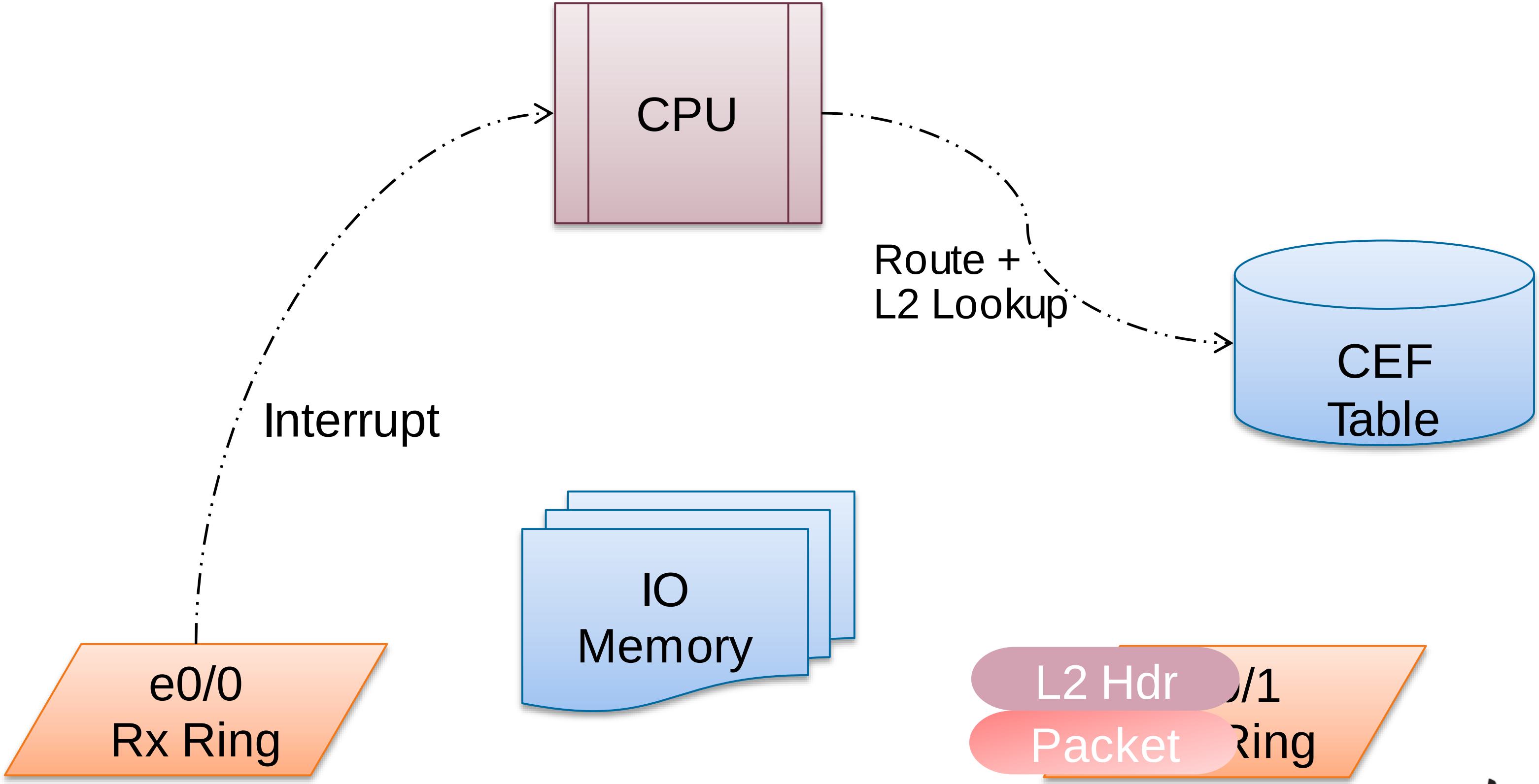


# CEF Switching



# CEF Switching



# CEF Switching - Summary

- Interrupt removes process scheduling
- Pre-compiled Interface + L2 information (cache)
- CEF table data structure improvement
  - RIB is a hash
  - CEF is a mtrie
- Single lookup for all necessary forwarding information

# CEF Switching - Features

- Supported in CEF
  - QoS
  - ACL
  - Zone Based Firewall
  - NAT
  - NetFlow
  - IPSec
  - GRE
  - PBR
  - Many more!
- Process Switching Only
  - ACL Logging
  - Packets destined to the router
  - No L2 Adjacency

# Agenda

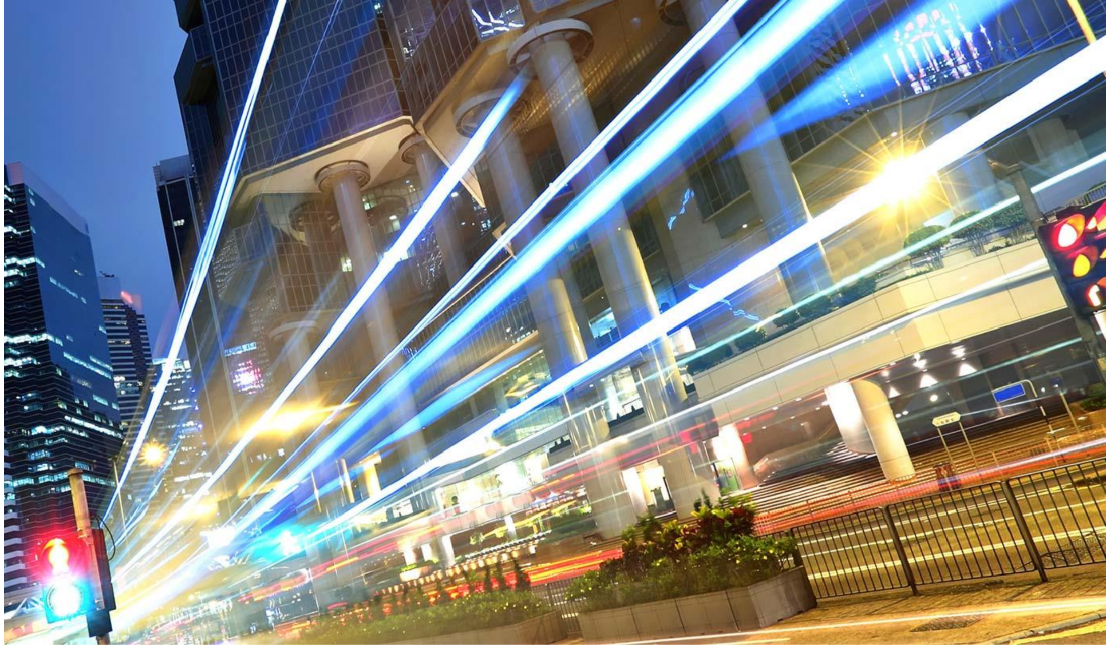
- Router Components
- Moving Packets

➢ **CEF, CPU and Memory**

 ➢ **Processes and Interrupts**

 – Routing Memory Utilisation

- Outbound Load Sharing
- Routing Convergence Improvements



# CEF and CPU Utilisation

- CPU does everything
- Total CPU vs. Interrupts
  - SPF, BGP
  - Routed Packets

Total CPU – Interrupts = Utilisation Due to Processes

```
CPU utilization for five seconds: 5%/2%; one minute: 3%; five minutes: 2%
  PID  Runtime (ms)     Invoked    uSecs       5Sec   1Min     5Min  TTY     Process
...
    2            68         585      116      1.00%  1.00%       0%    0     IP Input
   17            88        4232       20      0.20%  1.00%       0%    0     BGP Router
   18           152       14650       10         0%     0%       0%    0     BGP Scanner
...
```

# CPU Utilisation Examples

1. CPU Utilisation due to moderate traffic rates

```
CPU utilization for five seconds: 47%/46%; one minute: 40%; five minutes: 39%
```

# CPU Utilisation Examples

1. CPU Utilisation due to moderate traffic rates

```
CPU utilization for five seconds: 47%/46%; one minute: 40%; five minutes: 39%
```

2. High CPU due to OSPF Reconvergence

```
CPU utilization for five seconds: 99%/3%; one minute: 53%; five minutes: 49%
 PID  Runtime(ms)   Invoked    uSecs     5Sec     1Min    5Min   TTY Process
 357       319932    138750    21039   88.32%   41.18%  36.78%     0 OSPF-1 Router
```

# CPU Utilisation Examples

1. CPU Utilisation due to moderate traffic rates

```
CPU utilization for five seconds: 47%/46%; one minute: 40%; five minutes: 39%
```

2. High CPU due to OSPF Reconvergence

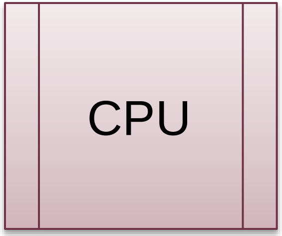
```
CPU utilization for five seconds: 99%/3%; one minute: 53%; five minutes: 49%
 PID  Runtime(ms)   Invoked    uSecs      5Sec      1Min    5Min    TTY Process
 357        319932   138750    21039    88.32%    41.18%   36.78%     0 OSPF-1 Router
```

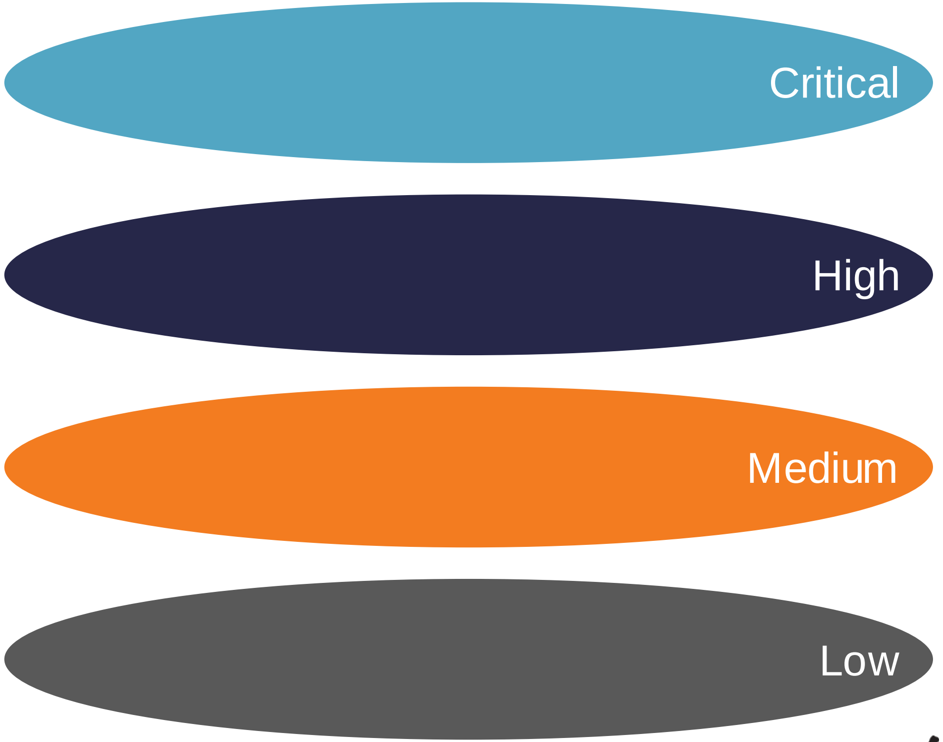
3. High CPU due to multiple Virtual Exec Processes

```
CPU utilization for five seconds: 99%/3%; one minute: 99%; five minutes: 99%
 PID Runtime(ms)    Invoked        uSecs    5Sec    1Min    5Min TTY Process
  3      24871276   47622133         522 30.62% 31.62% 31.57%   2 Virtual Exec
122      24812452   47528825         522 30.53% 31.62% 31.60%   3 Virtual Exec
131      24790280   47490842         522 32.84% 31.88% 31.31%   4 Virtual Exec
```

# Process Priority

- Processes assigned priority
  - Critical/High/Medium/Low

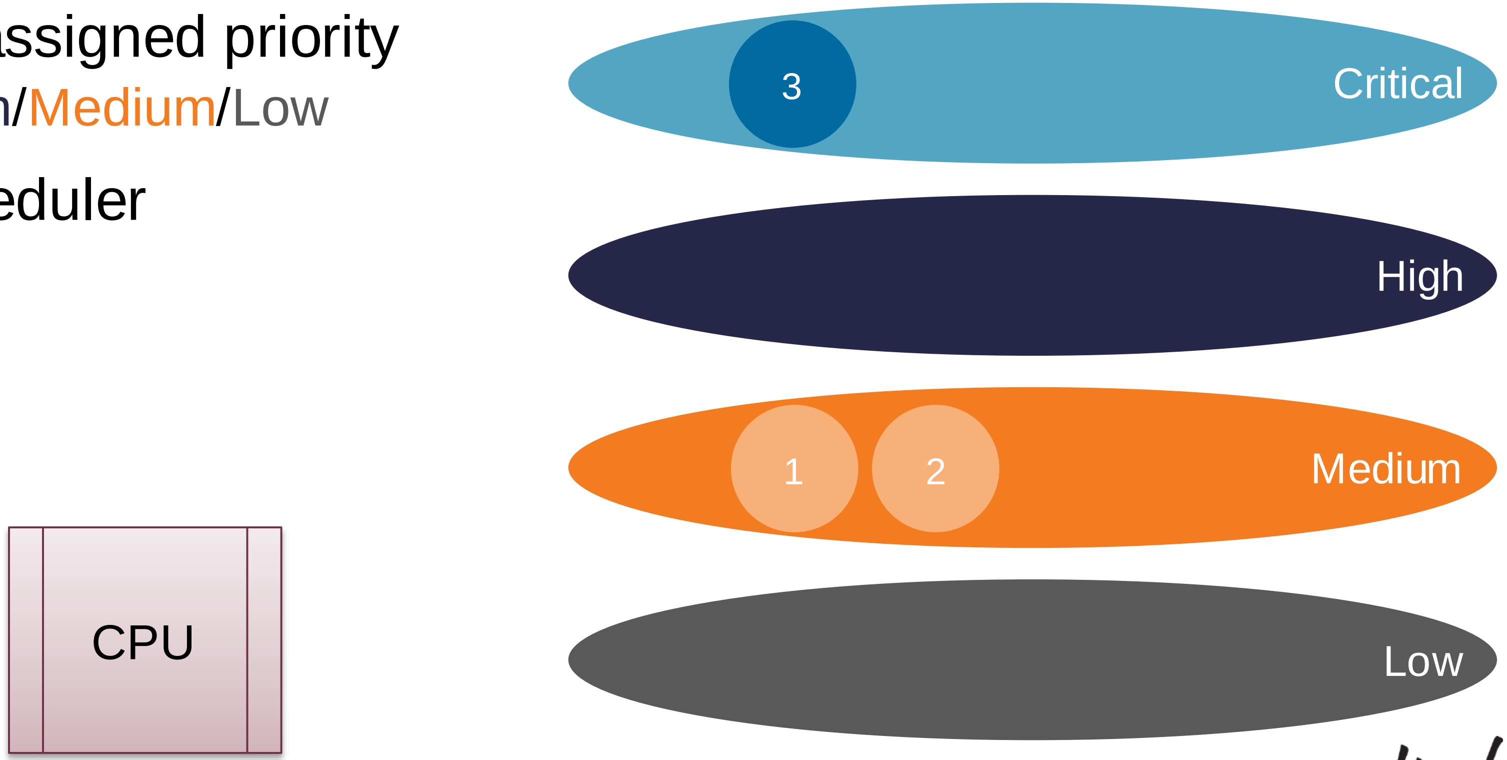


# Process Priority

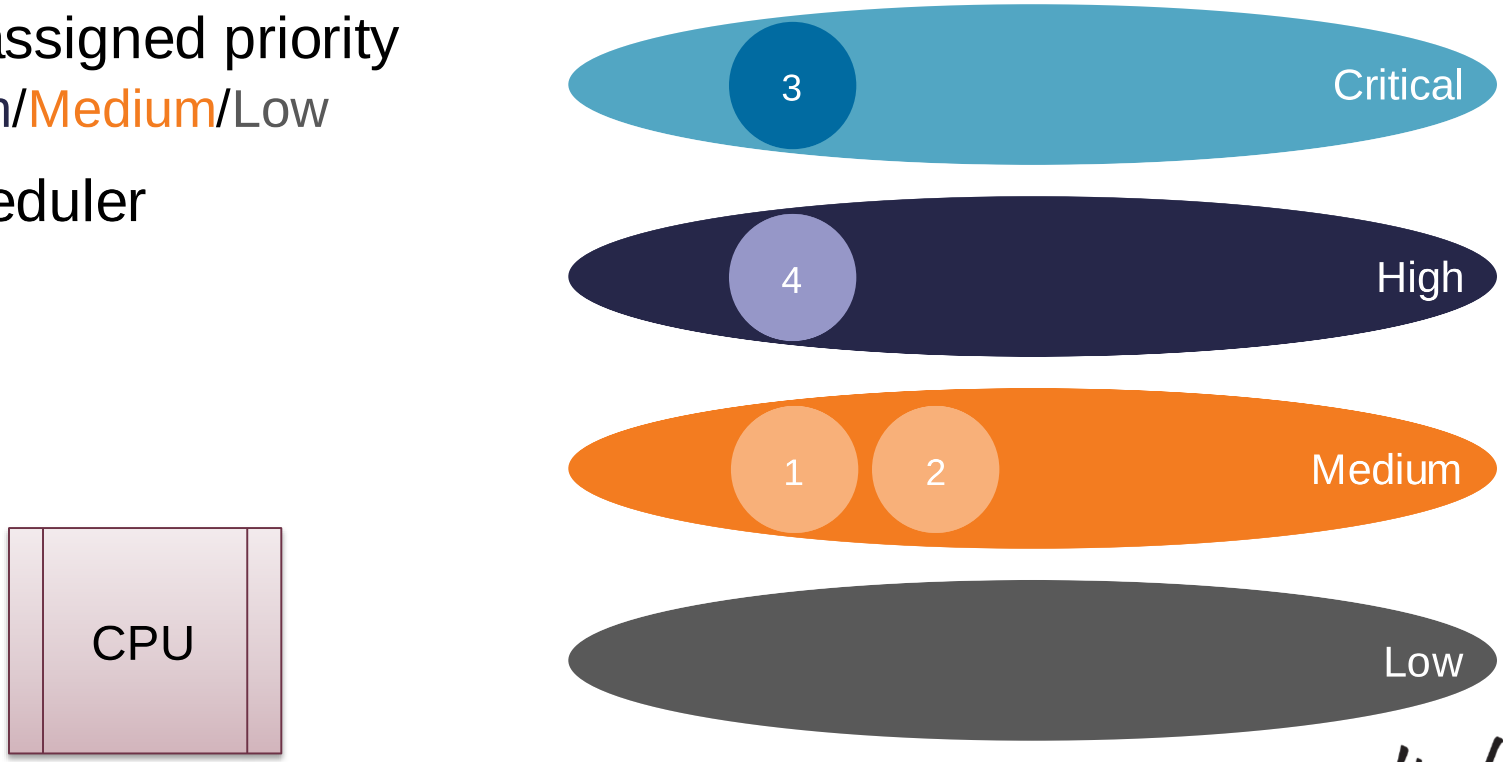
- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler



# Process Priority

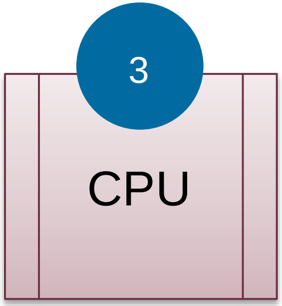
- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler

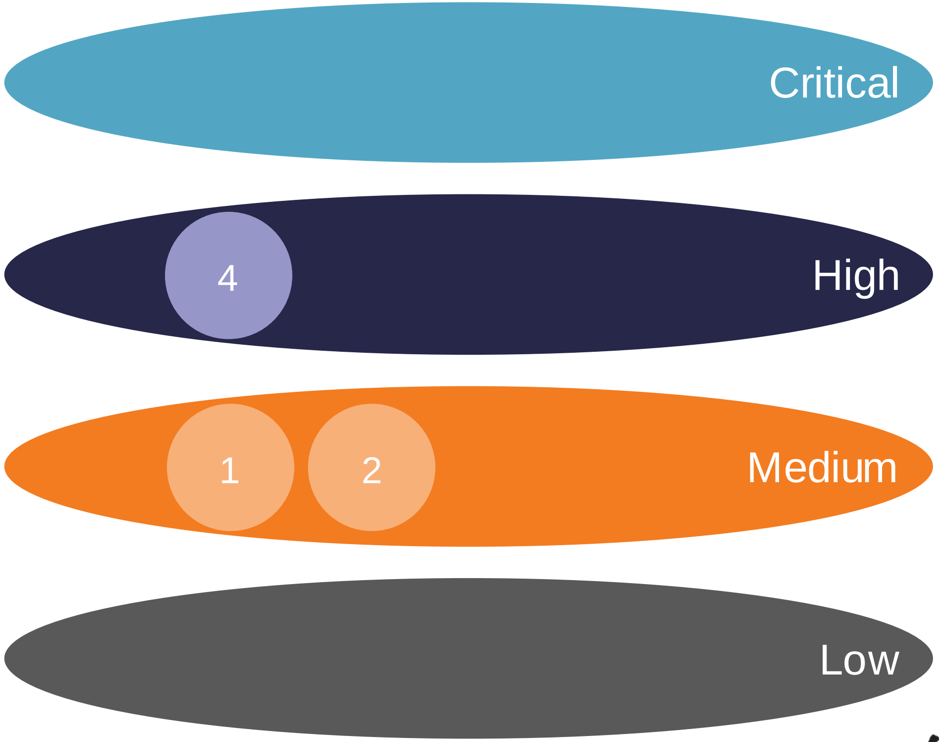


# Process Priority

- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler

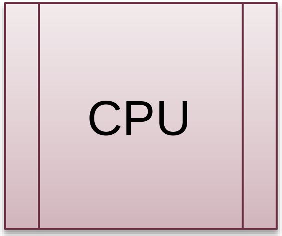


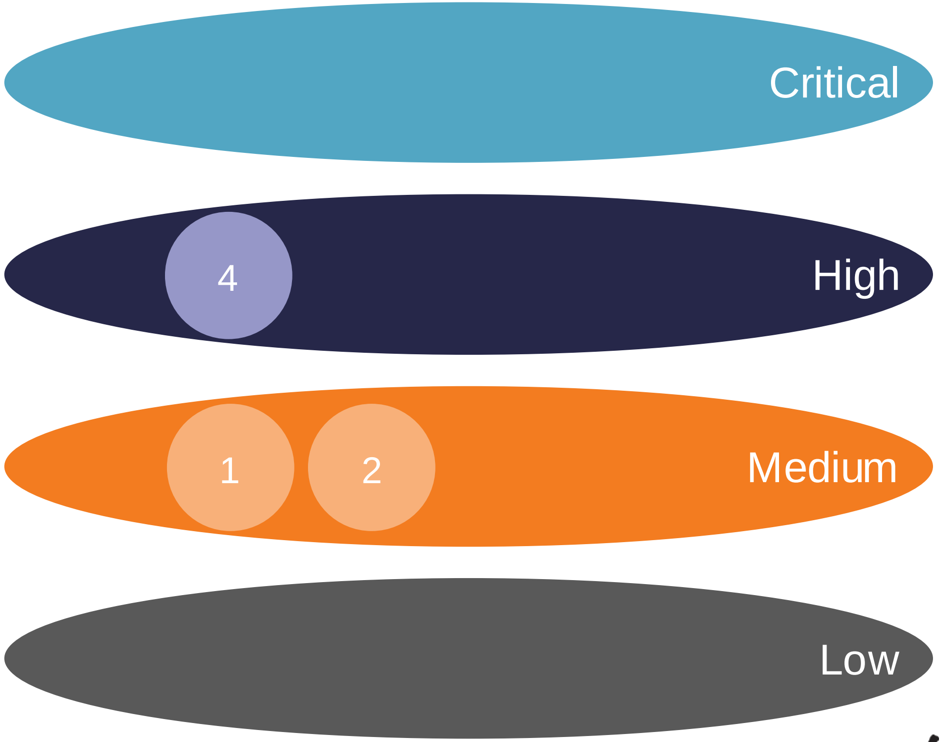



# Process Priority

- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler

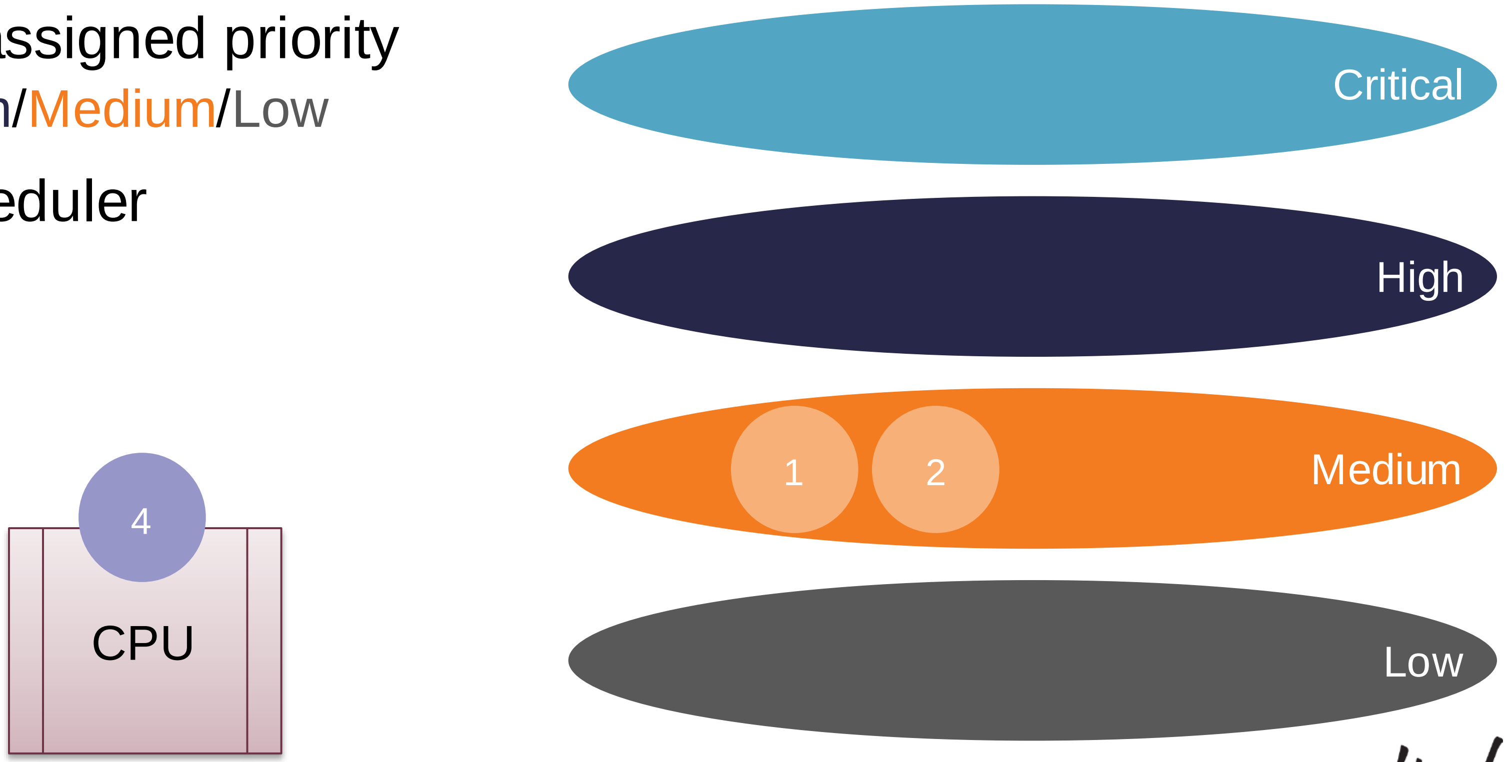


# Process Priority

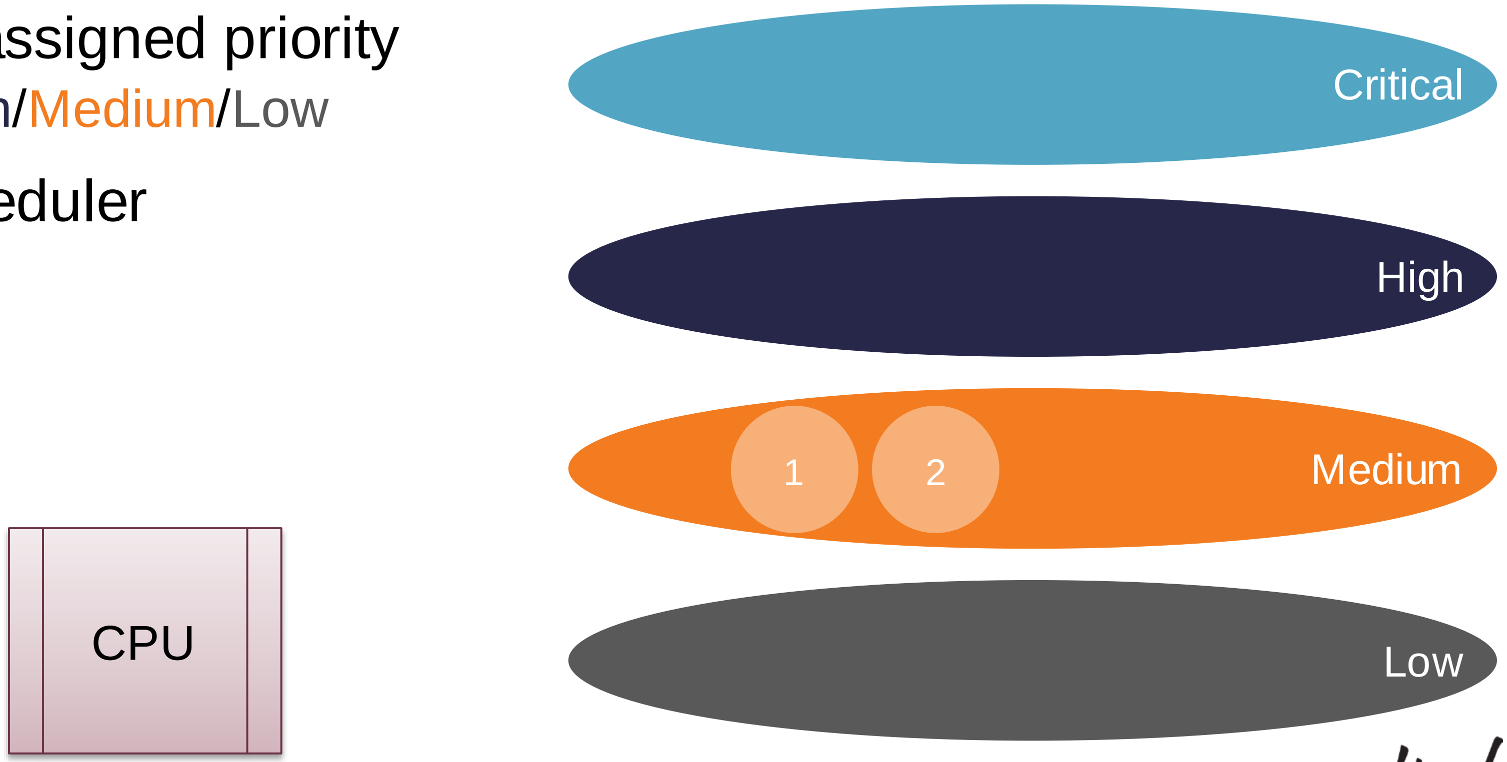
- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler



# Process Priority

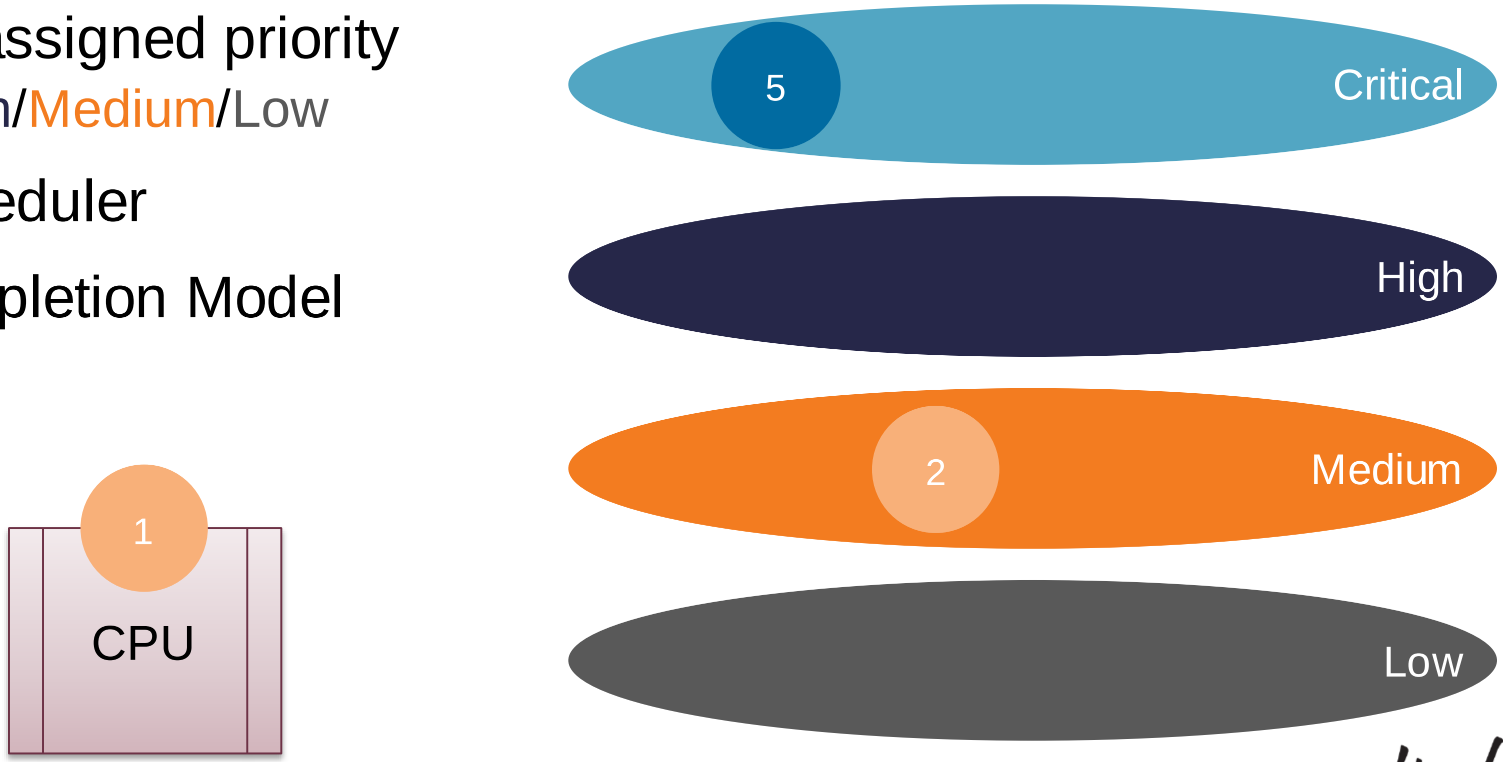
- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler



# Process Priority

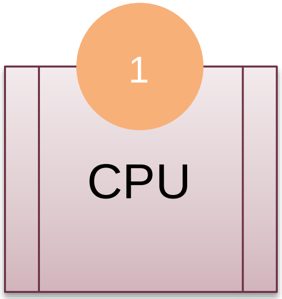
- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler
- Run to Completion Model

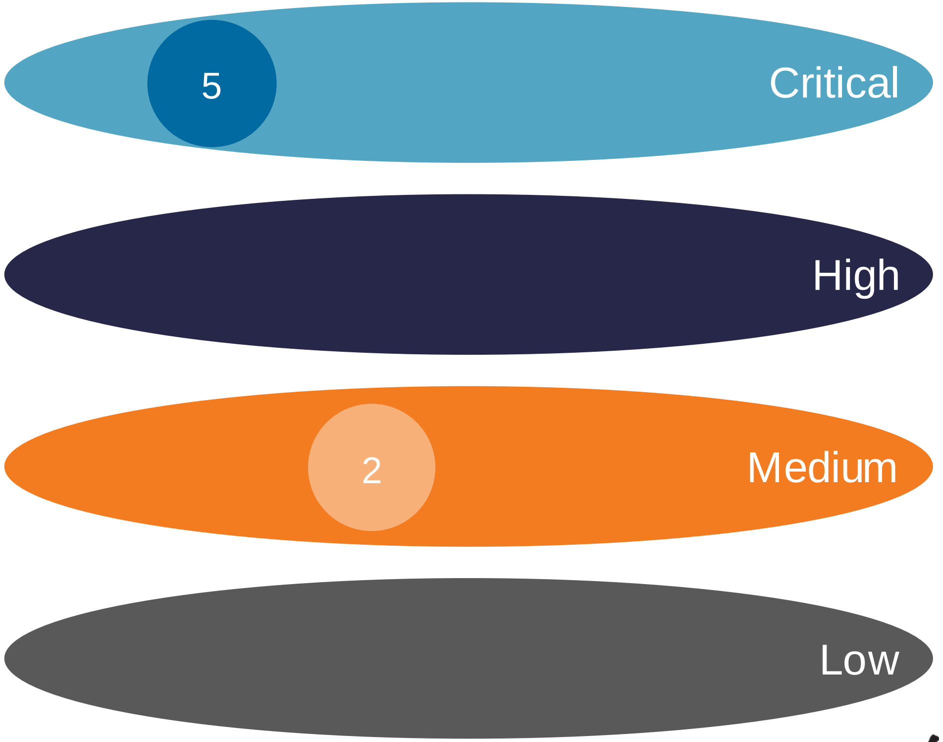


# Process Priority

- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler
- Run to Completion Model
  - Processes choose to suspend

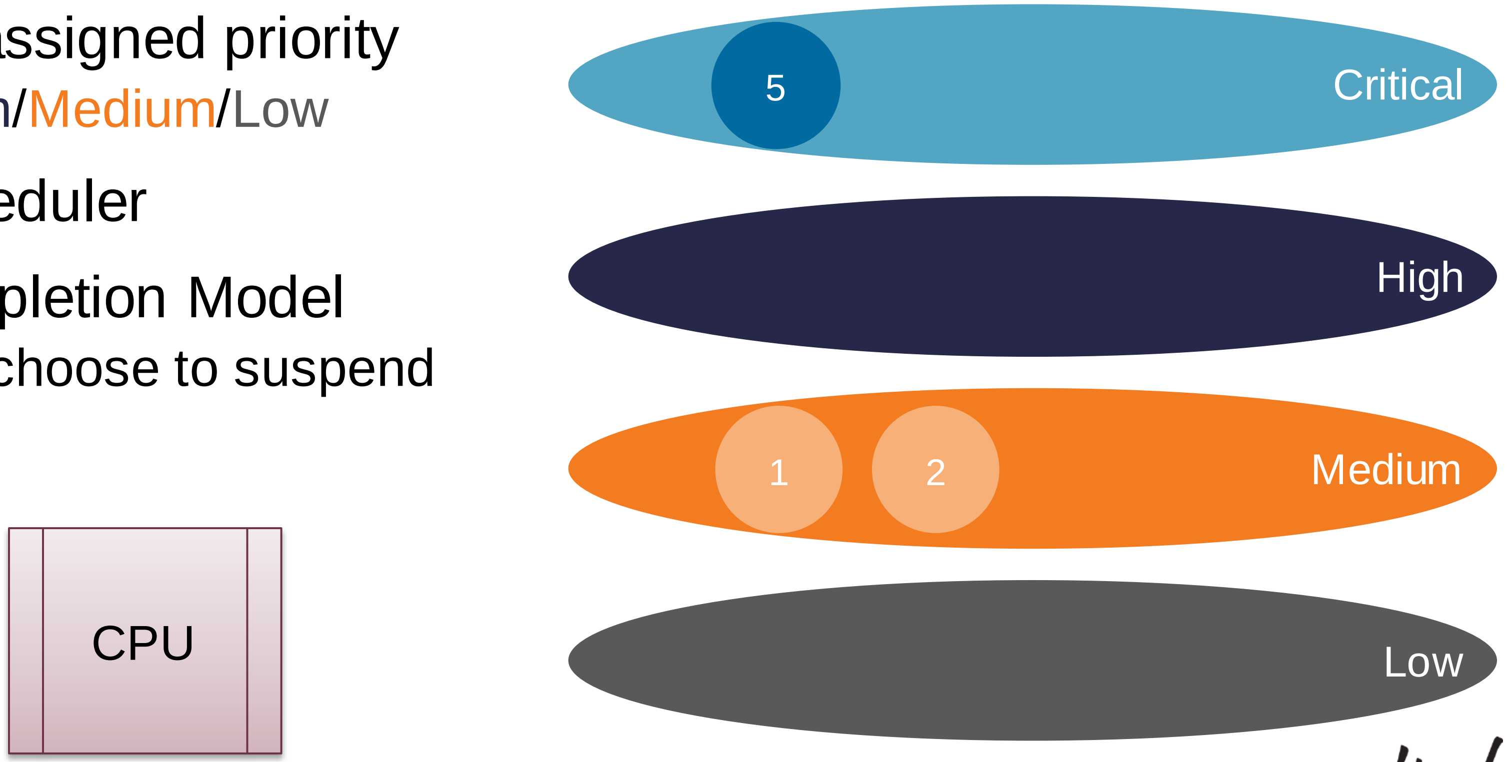


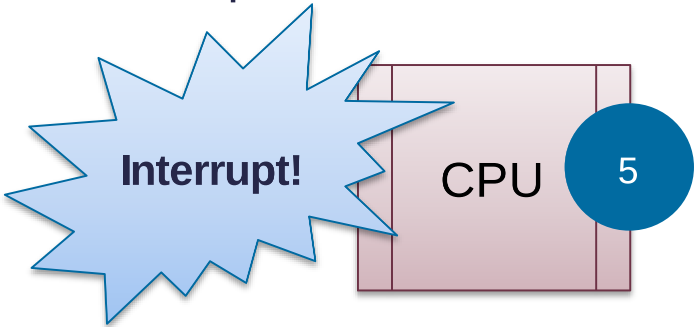
# Process Priority

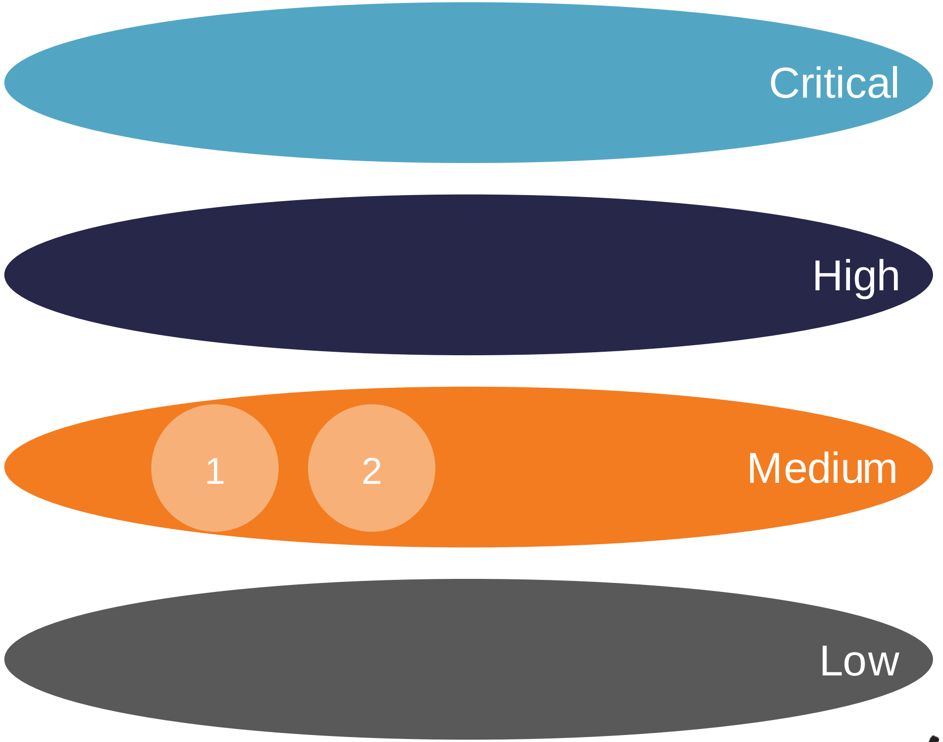
- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler
- Run to Completion Model
  - Processes choose to suspend



# Process Priority

- Processes assigned priority
  - Critical/High/Medium/Low
- Priority Scheduler
- Run to Completion Model
  - Processes choose to suspend
  - Interrupts break the rules



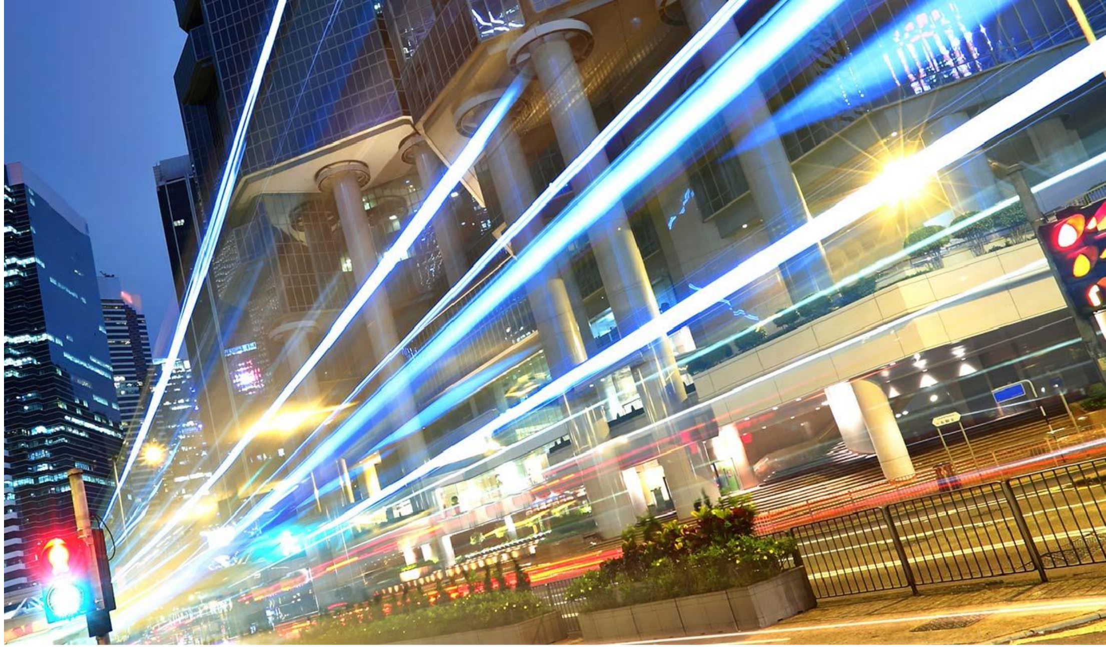

  

# Agenda

- Router Components
- Moving Packets

➢ **CEF, CPU and Memory**
  - Processes and Interrupts
  - ➢ **Routing Memory Utilisation**

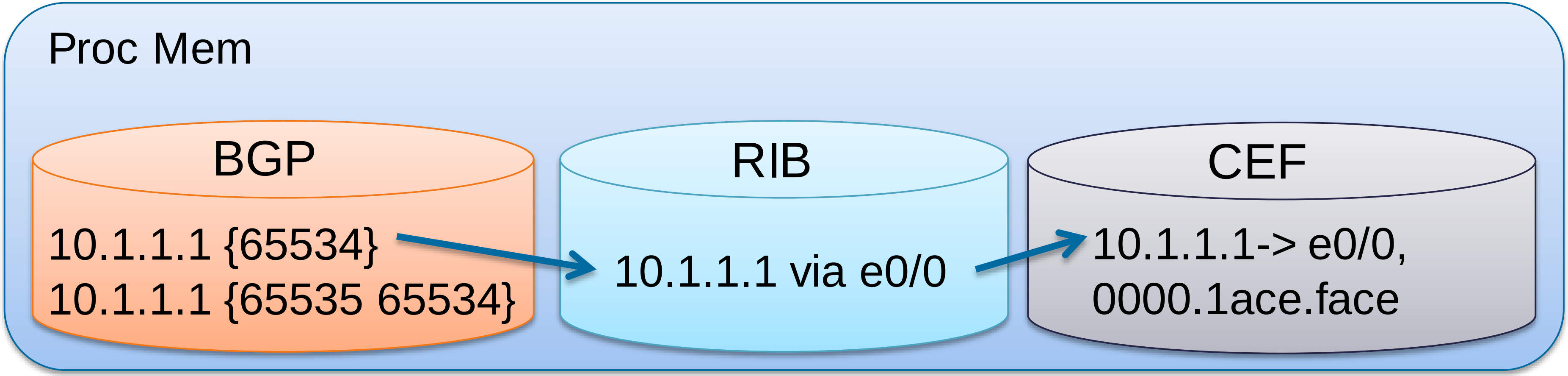
- Outbound Load Sharing
- Routing Convergence Improvements



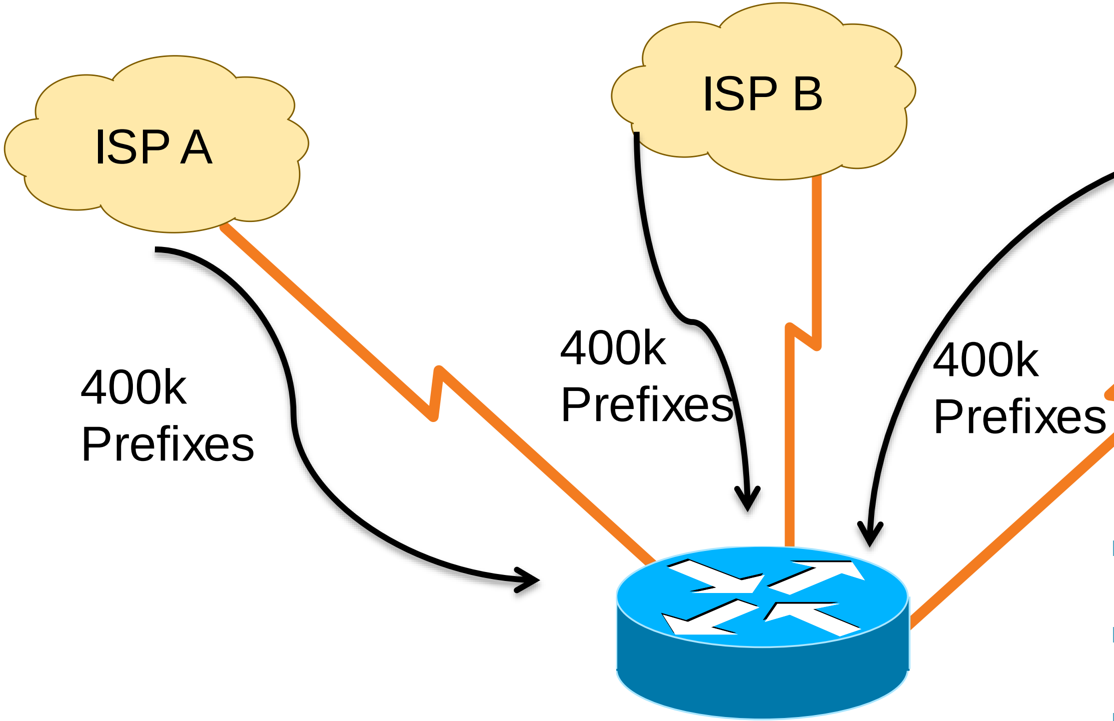
# Routing Process Memory

- Routing Protocol, RIB, and CEF each take their own memory
- RIB built from Routing Protocols
- CEF built from RIB



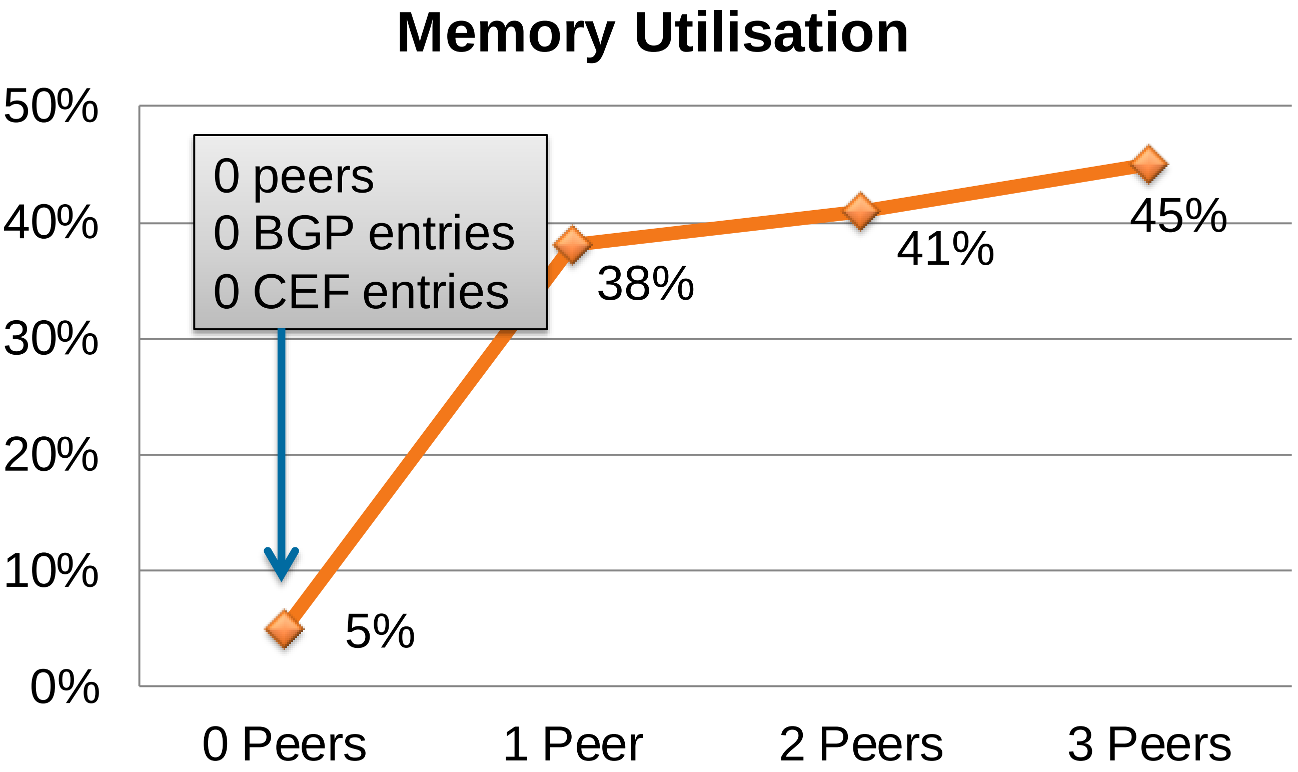

  

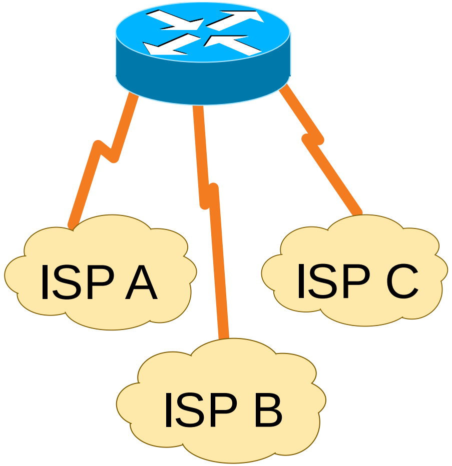
# Memory Impact of Multiple Prefixes




- 3 BGP Peers
- 400k **Identical** Routes
- 15.2(2)T

# Memory Impact of Multiple Prefixes

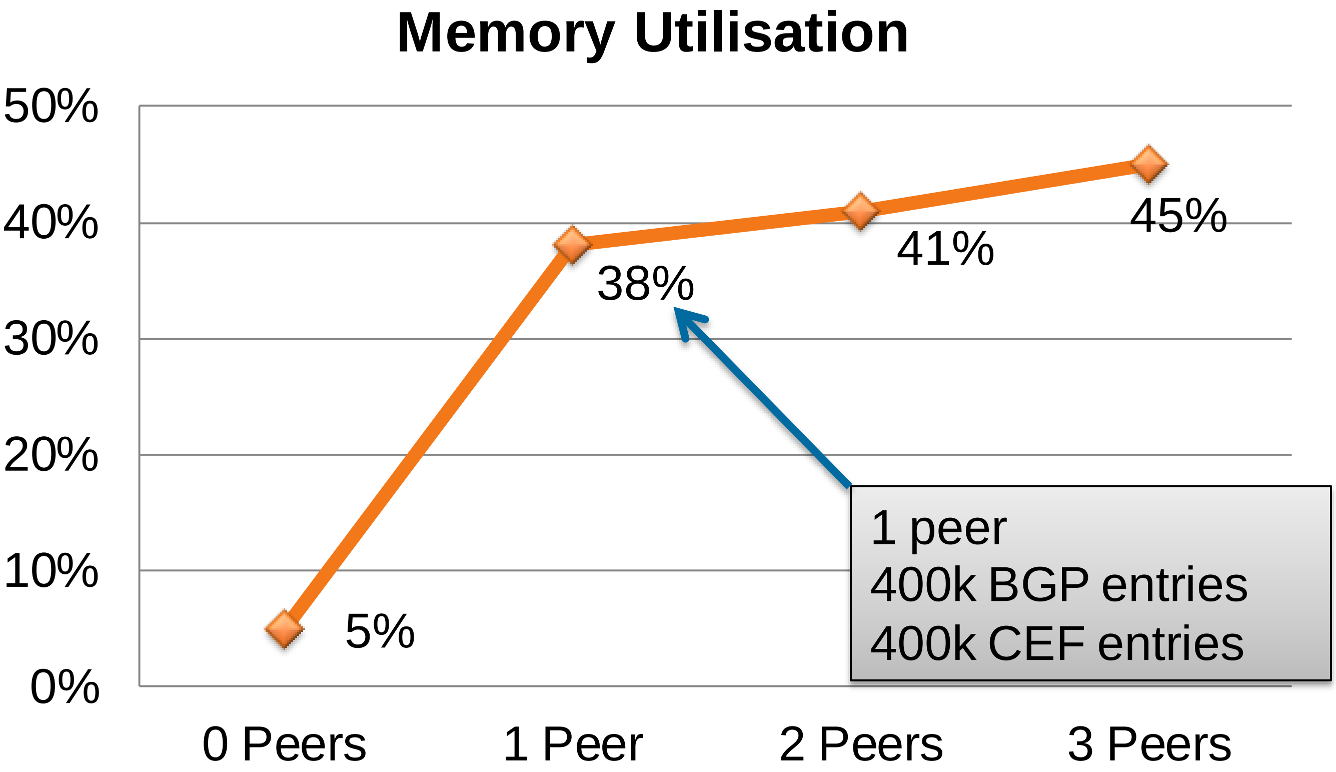


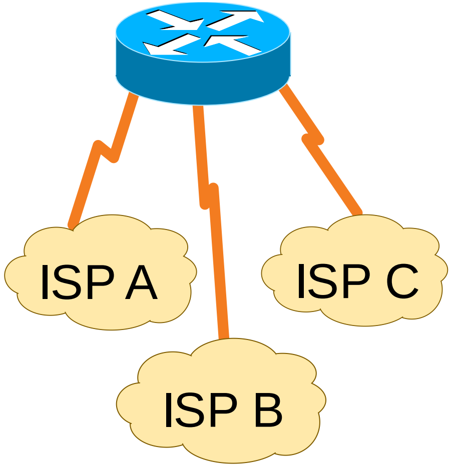




- 3 BGP Peers
- 400k Identical Routes
- 15.2(2)T

# Memory Impact of Multiple Prefixes

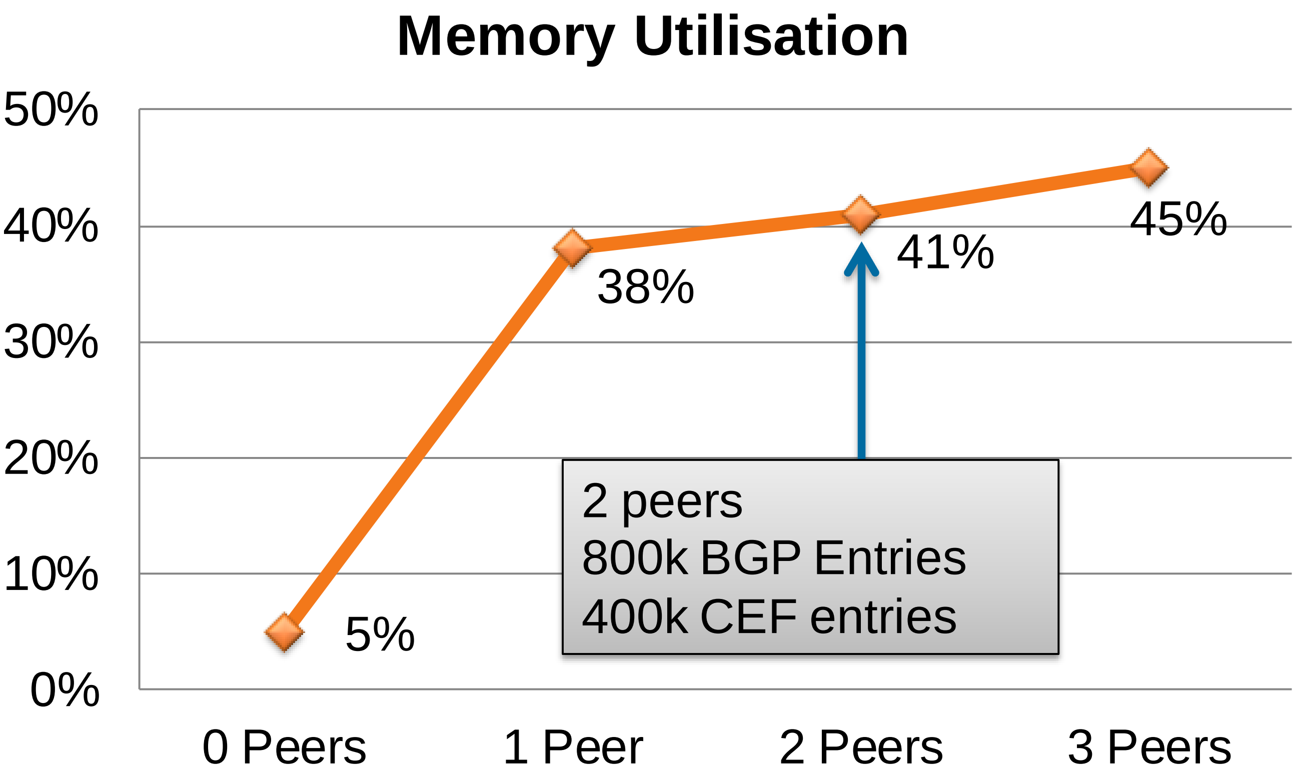


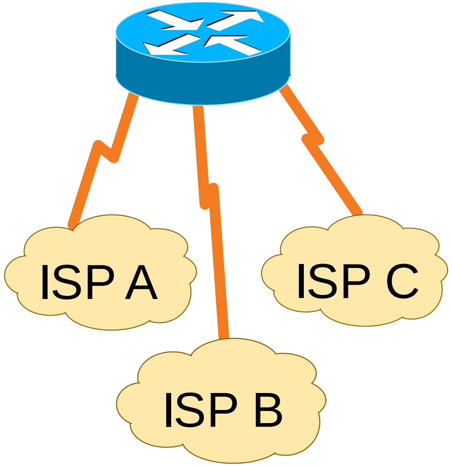




- 3 BGP Peers
- 400k Identical Routes
- 15.2(2)T

# Memory Impact of Multiple Prefixes

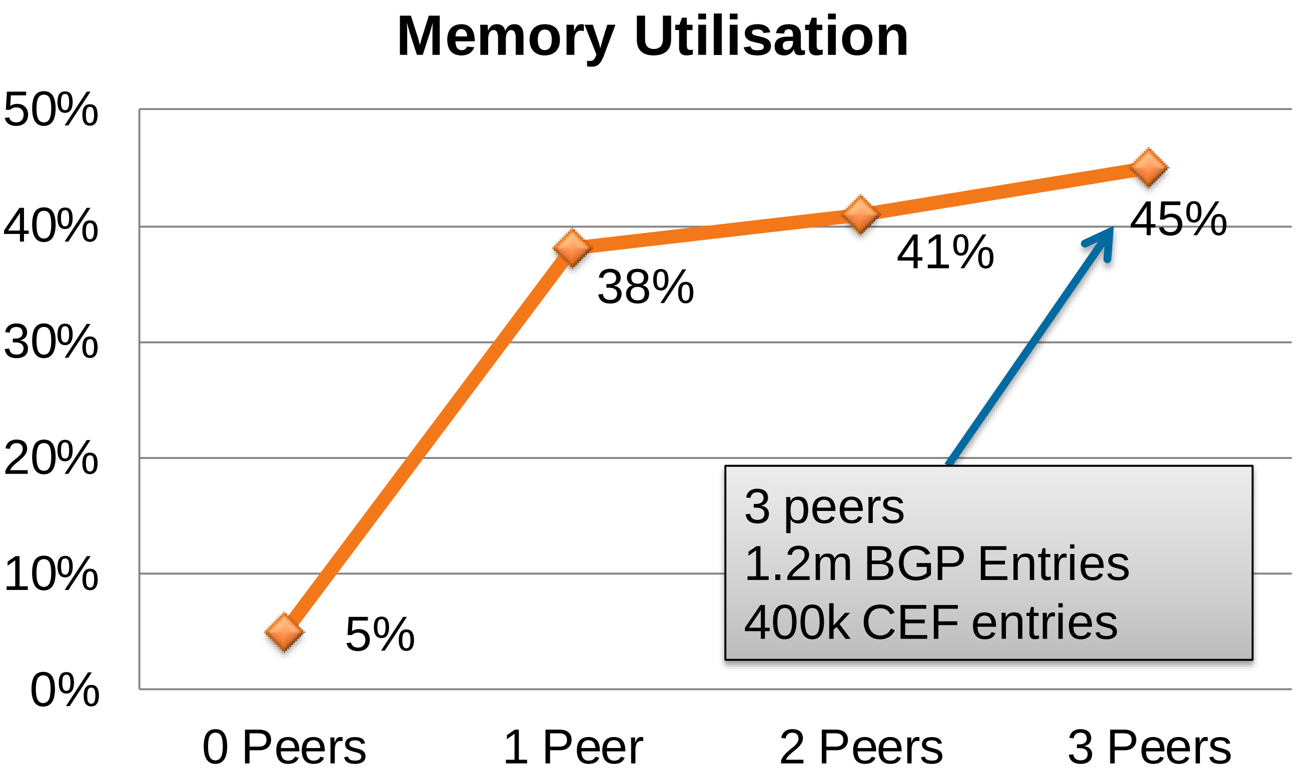


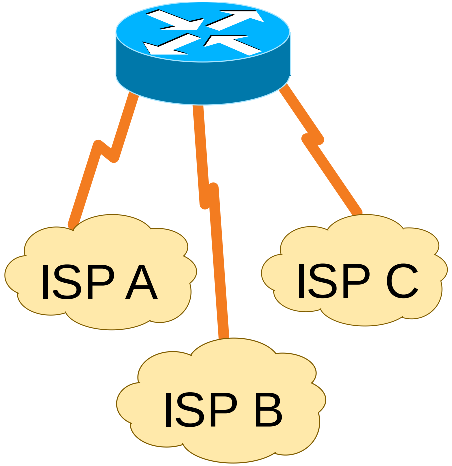




- 3 BGP Peers
- 400k Identical Routes
- 15.2(2)T

# Memory Impact of Multiple Prefixes



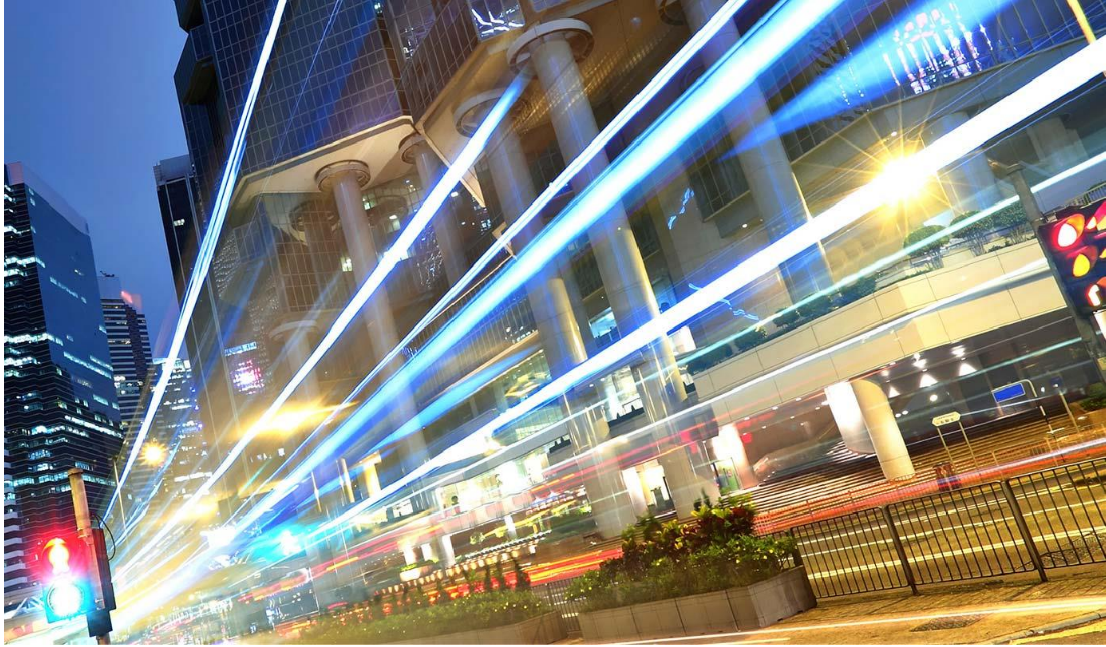

- 3 BGP Peers
- 400k Identical Routes
- 15.2(2)T

# Agenda

- Router Components
- Moving Packets
- CEF, CPU and Memory

➢ **Outbound Load Sharing**
  - ➢ **CEF Equal Cost Multipath (ECMP)**
  - – Load Sharing with Performance Routing (PfR)
- Routing Convergence Improvements



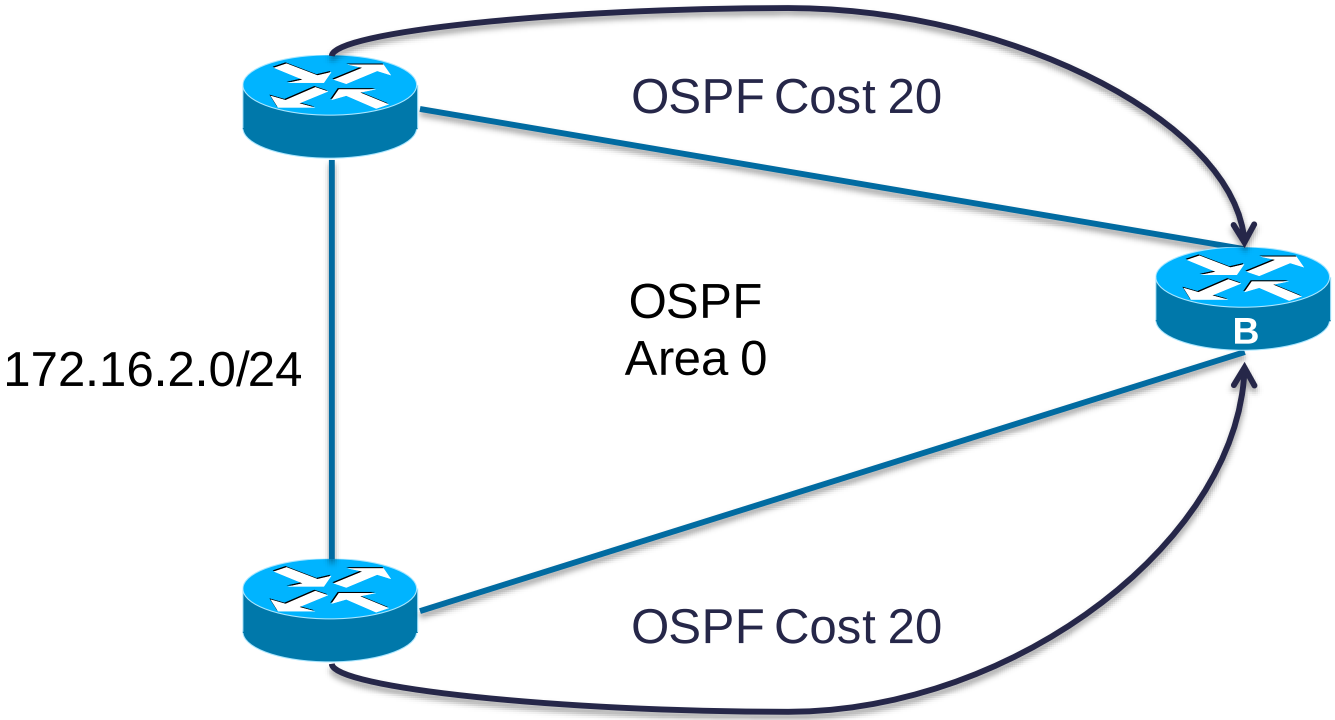
# Load Sharing vs. Load Balancing

- Load balancing implies intelligence
- Load sharing is simple

- Load balancing has fairness
- Load sharing has no measurements

# Equal Cost Loadsharing



# Routing Table – Equal Cost Routes

```
RouterB#show ip route 172.16.2.0

Routing entry for 172.16.2.0/24
  Known via "ospf 1", distance 110, metric 20, type intra area
  Last update from 172.16.1.1 on Ethernet0/0, 1d02h ago
  Routing Descriptor Blocks:

  * 192.168.100.1, from 192.168.200.1, 1d02h ago, via Ethernet0/1
      Route metric is 20, traffic share count is 1

    172.16.1.1, from 192.168.200.1, 1d02h ago, via Ethernet0/0
      Route metric is 20, traffic share count is 1
```

# Routing Table – Equal Cost Routes

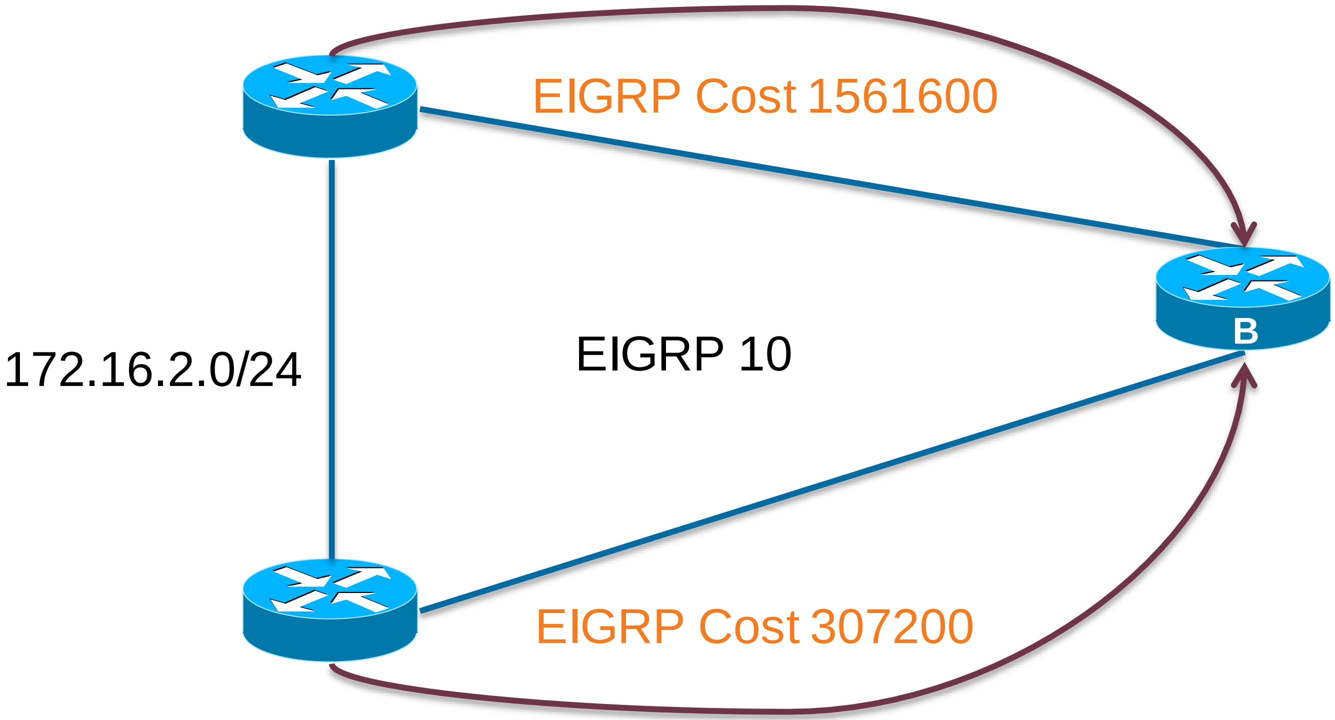
```
RouterB#show ip route 172.16.2.0

Routing entry for 172.16.2.0/24
  Known via "ospf 1", distance 110, metric 20, type intra area
  Last update from 172.16.1.1 on Ethernet0/0, 1d02h ago
  Routing Descriptor Blocks:

  * 192.168.100.1, from 192.168.200.1, 1d02h ago, via Ethernet0/1
      Route metric is 20, traffic share count is 1

    172.16.1.1, from 192.168.200.1, 1d02h ago, via Ethernet0/0
      Route metric is 20, traffic share count is 1
```

# Routing Table – Equal Cost Routes

```
RouterB#show ip route 172.16.2.0

Routing entry for 172.16.2.0/24
  Known via "ospf 1", distance 110, metric 20, type intra area
  Last update from 172.16.1.1 on Ethernet0/0, 1d02h ago
  Routing Descriptor Blocks:

  * 192.168.100.1, from 192.168.200.1, 1d02h ago, via Ethernet0/1
      Route metric is 20, traffic share count is 1

    172.16.1.1, from 192.168.200.1, 1d02h ago, via Ethernet0/0
      Route metric is 20, traffic share count is 1
```

# Unequal Cost Load Sharing



# Routing Table – Unequal Cost Routes

```
RouterB#show ip route 172.16.2.0

Routing entry for 172.16.2.0/24
  Known via "eigrp 10", distance 90, metric 307200, type internal
  Last update from 172.16.1.1 on Ethernet0/0, 1d02h ago
  Routing Descriptor Blocks:

    192.168.100.1, from 192.168.200.1, 1d02h ago, via Ethernet0/1
      Route metric is 1561600, traffic share count is 47

      ...

  * 172.16.1.1, from 172.16.1.1, 00:00:16 ago, via Ethernet0/0
      Route metric is 307200, traffic share count is 240
```

Unequal Metrics

# Routing Table – Unequal Cost Routes

```
RouterB#show ip route 172.16.2.0

Routing entry for 172.16.2.0/24
  Known via "eigrp 10", distance 90, metric 307200, type internal
  Last update from 172.16.1.1 on Ethernet0/0, 1d02h ago
  Routing Descriptor Blocks:

    192.168.100.1, from 192.168.200.1, 1d02h ago, via Ethernet0/1
      Route metric is 1561600, traffic share count is 47

      ...

  * 172.16.1.1, from 172.16.1.1, 00:00:16 ago, via Ethernet0/0
      Route metric is 307200, traffic share count is 240
```

Unequal Traffic Share Count

# Routing Table – Unequal Cost Routes

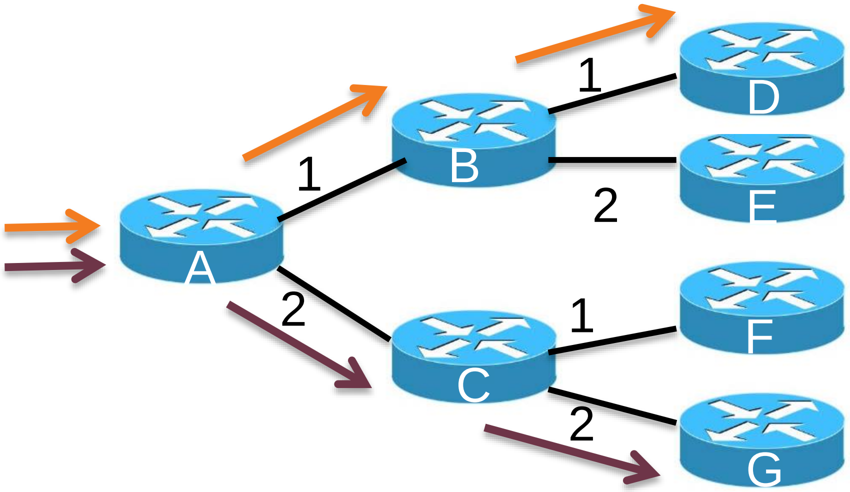
```
RouterB#show ip route 172.16.2.0

Routing entry for 172.16.2.0/24
  Known via "eigrp 10", distance 90, metric 307200, type internal
  Last update from 172.16.1.1 on Ethernet0/0, 1d02h ago
  Routing Descriptor Blocks:

    192.168.100.1, from 192.168.200.1, 1d02h ago, via Ethernet0/1
      Route metric is 1561600, traffic share count is 47

      ...

  * 172.16.1.1, from 172.16.1.1, 00:00:16 ago, via Ethernet0/0
      Route metric is 307200, traffic share count is 240
```

Only Accomplished with EIGRP **variance** command

# CEF Hashing

- CEF hash is deterministic
  - Same input always provides the same output

Packet 1 = src 10.1.1.1 dst 10.2.2.2
Packet 2 = src 10.1.1.1 dst 10.3.3.3




- Without randomisation every router makes the same decision
- Downstream routers never loadshare

# CEF Hashing Algorithm

- Default hash is “Universal”
- Source IP + Destination IP + Universal Identifier
- Universal ID prevents polarisation
- Other hashes can be used for fixing unequal load sharing

```
RouterB#show cef state
CEF Status:
…
 universal per-destination load sharing algorithm, id 0F33353C
```

# CEF Loadsharing Options

- Per-Packet
  - More even load sharing
  - Jitter
  - Out of Order packets (bad for lots of applications)

- Per-Destination (default)
  - Can be less even load sharing
  - Ordered delivery
  - Hashing challenges

# CEF Hashing

```
RouterB#show ip CEF 172.16.2.1 internal

172.16.2.0/24, epoch 0, RIB[I], refcount 5, per-destination sharing
…
  ifnums:
   Ethernet0/0(3): 172.16.1.1
   Ethernet0/1(4): 192.168.200.1

  path 08172748, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 172.16.1.1 Eth0/0, adj IP adj out Eth0/0, addr 172.16.1.1 081E35A0

  path 08172898, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 192.168.200.1 Eth0/1, adj IP adj out Eth0/1, addr 192.168.200.1 0F75D9F8

  flags: Per-session, for-rx-IPv4, 2buckets
    2 hash buckets

      < 0 > IP adj out of Ethernet0/0, addr 172.16.1.1 081E35A0
      < 1 > IP adj out of Ethernet0/1, addr 192.168.200.1 0F75D9F8
```

# CEF Hashing

```
RouterB#show ip CEF 172.16.2.1 internal

172.16.2.0/24, epoch 0, RIB[I], refcount 5, per-destination sharing
…
  ifnums:
   Ethernet0/0(3): 172.16.1.1
   Ethernet0/1(4): 192.168.200.1

  path 08172748, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 172.16.1.1 Eth0/0, adj IP adj out Eth0/0, addr 172.16.1.1 081E35A0

  path 08172898, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 192.168.200.1 Eth0/1, adj IP adj out Eth0/1, addr 192.168.200.1 0F75D9F8

  flags: Per-session, for-rx-IPv4, 2buckets
    2 hash buckets

      < 0 > IP adj out of Ethernet0/0, addr 172.16.1.1 081E35A0
      < 1 > IP adj out of Ethernet0/1, addr 192.168.200.1 0F75D9F8
```

# CEF Hashing

```
RouterB#show ip CEF 172.16.2.1 internal

172.16.2.0/24, epoch 0, RIB[I], refcount 5, per-destination sharing
…
  ifnums:
   Ethernet0/0(3): 172.16.1.1
   Ethernet0/1(4): 192.168.200.1

  path 08172748, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 172.16.1.1 Eth0/0, adj IP adj out Eth0/0, addr 172.16.1.1 081E35A0

  path 08172898, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 192.168.200.1 Eth0/1, adj IP adj out Eth0/1, addr 192.168.200.1 0F75D9F8

  flags: Per-session, for-rx-IPv4, 2buckets
    2 hash buckets

      < 0 > IP adj out of Ethernet0/0, addr 172.16.1.1 081E35A0
      < 1 > IP adj out of Ethernet0/1, addr 192.168.200.1 0F75D9F8
```

# CEF Hashing

```
RouterB#show ip CEF 172.16.2.1 internal

172.16.2.0/24, epoch 0, RIB[I], refcount 5, per-destination sharing
…
  ifnums:
   Ethernet0/0(3): 172.16.1.1
   Ethernet0/1(4): 192.168.200.1

  path 08172748, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 172.16.1.1 Eth0/0, adj IP adj out Eth0/0, addr 172.16.1.1 081E35A0

  path 08172898, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 192.168.200.1 Eth0/1, adj IP adj out Eth0/1, addr 192.168.200.1 0F75D9F8

  flags: Per-session, for-rx-IPv4, 2buckets
    2 hash buckets

      < 0 > IP adj out of Ethernet0/0, addr 172.16.1.1 081E35A0
      < 1 > IP adj out of Ethernet0/1, addr 192.168.200.1 0F75D9F8
```

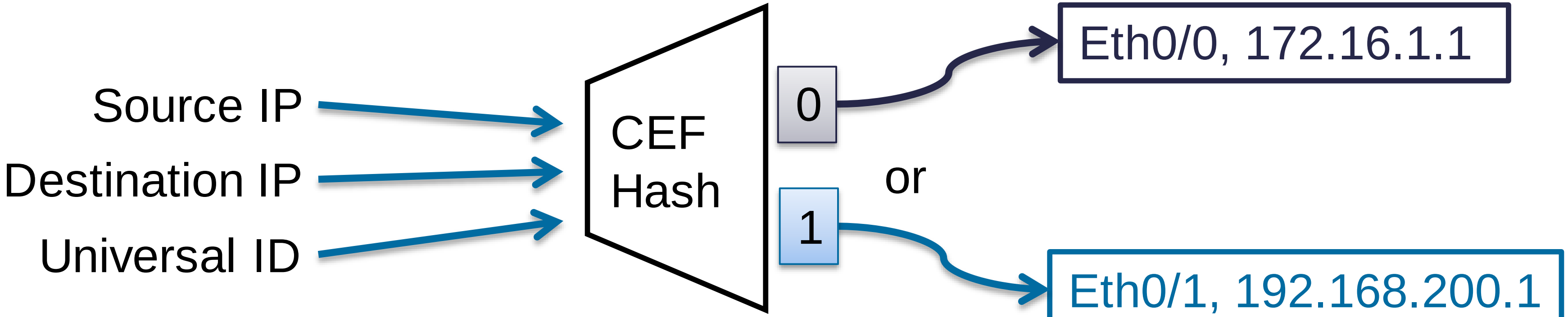
# CEF Hashing

```
RouterB#show ip CEF 172.16.2.1 internal

172.16.2.0/24, epoch 0, RIB[I], refcount 5, per-destination sharing
…
  ifnums:
   Ethernet0/0(3): 172.16.1.1
   Ethernet0/1(4): 192.168.200.1

  path 08172748, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 172.16.1.1 Eth0/0, adj IP adj out Eth0/0, addr 172.16.1.1 081E35A0

  path 08172898, path list 100071A8, share 1/1, type attached nexthop, for IPv4
  nexthop 192.168.200.1 Eth0/1, adj IP adj out Eth0/1, addr 192.168.200.1 0F75D9F8

  flags: Per-session, for-rx-IPv4, 2buckets
    2 hash buckets

      < 0 > IP adj out of Ethernet0/0, addr 172.16.1.1 081E35A0
      < 1 > IP adj out of Ethernet0/1, addr 192.168.200.1 0F75D9F8
```

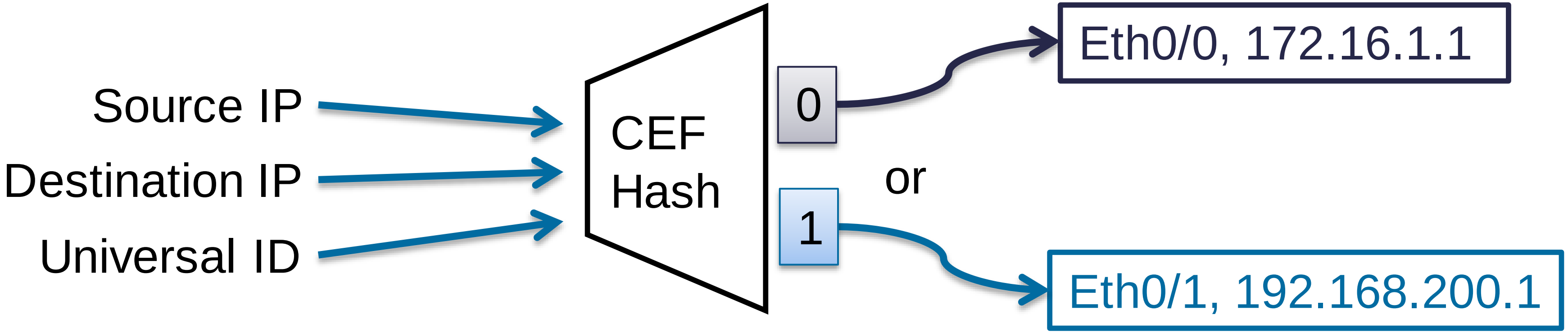
# CEF Hashing

```
2 hash buckets
      < 0 > IP adj out Ethernet0/0, addr 172.16.1.1 081E35A0
      < 1 > IP adj out Ethernet0/1, addr 192.168.200.1 0F75D9F8
```



# CEF Hashing

```
RouterB#show ip CEF exact-route 192.168.2.38 172.16.2.24
192.168.2.38 -> 172.16.2.24 => IP adj out Ethernet0/1, addr 192.168.200.1

RouterB#show ip CEF exact-route 192.168.2.40 172.16.2.24
192.168.2.40 -> 172.16.2.24 => IP adj out Ethernet0/0, addr 172.16.1.1
```



# Equal Cost Multipath - Summary

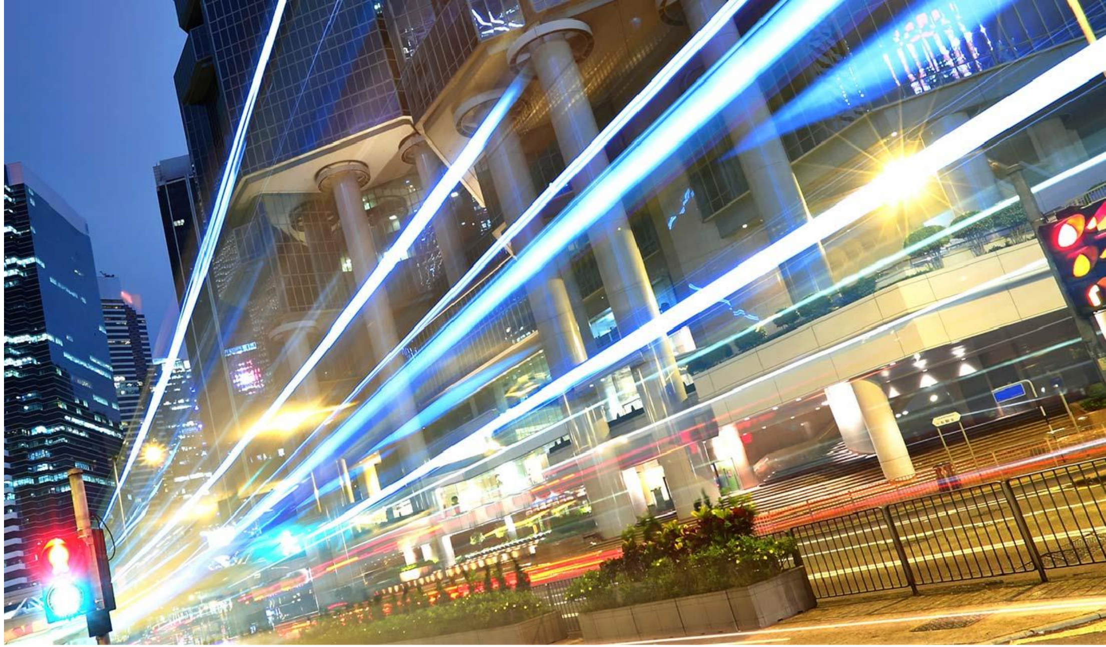
- CEF is built from the routing table
- Load sharing is part of routing decision
- Not 100% equal
- Based on Source IP + Destination IP + Universal ID
- Only one router

**How do I load share on more than one router?**

# Agenda

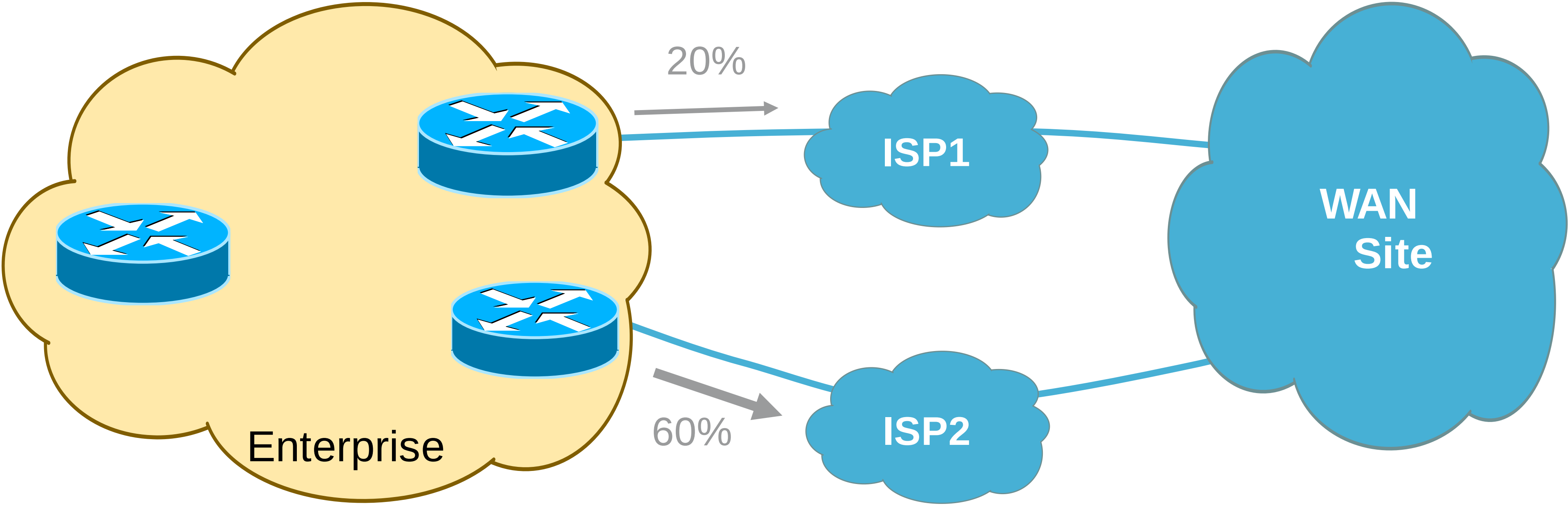
- Router Components
- Moving Packets
- CEF, CPU and Memory
- ➢ **Outbound Load Sharing**
  - CEF Equal Cost Multipath (ECMP)
  - ➢ **Load Sharing with Performance Routing (PfR)**
- Routing Convergence Improvements



# Loadsharing Across Routers

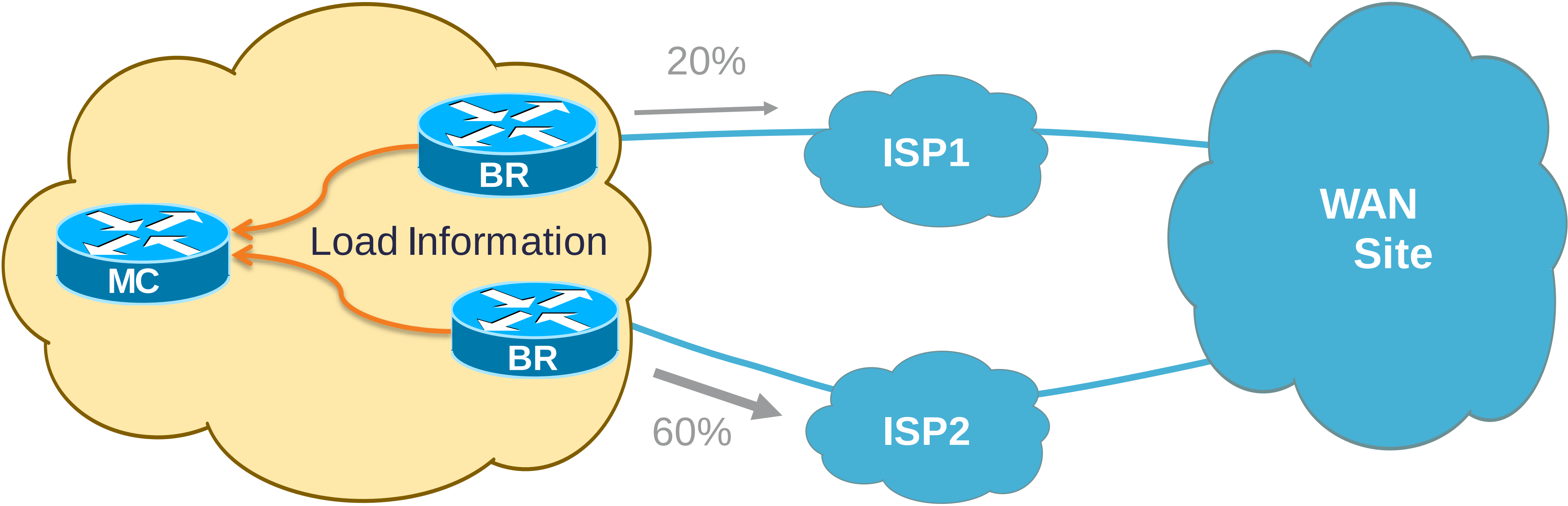
- CEF ECMP works per-router
- No dynamic way to get even sharing across routers



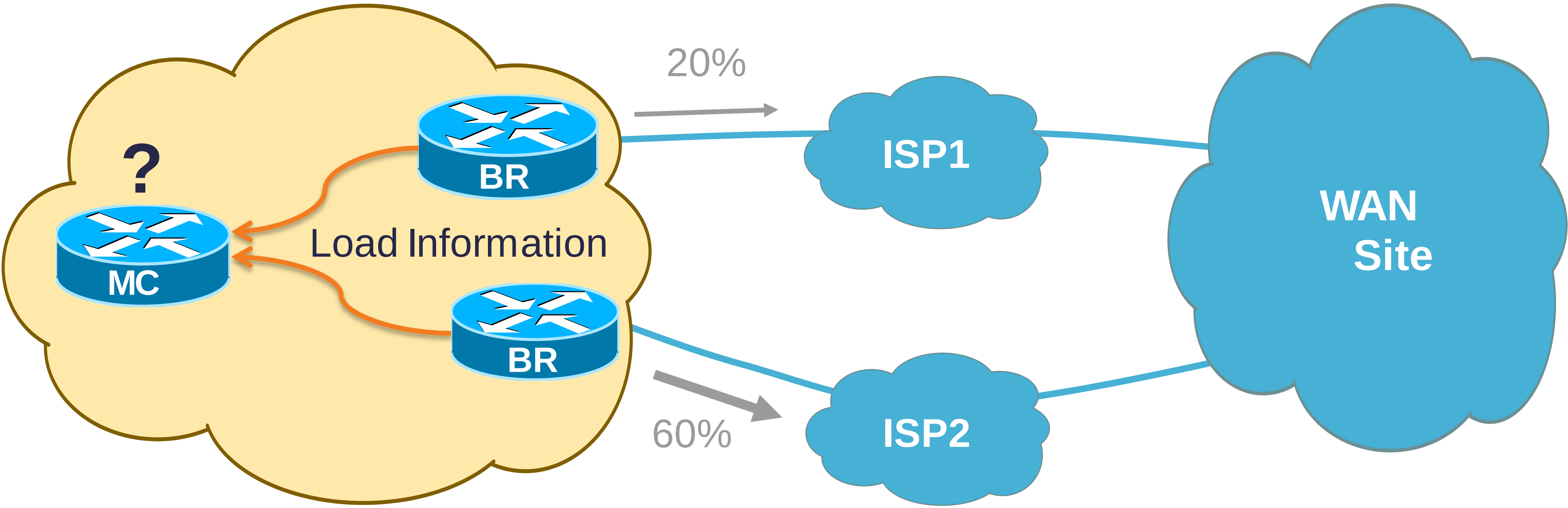
# PfR Operations

- Command and Control Infrastructure
- Border Routers (BRs) communicate load to Master Controller (MC)



# PfR Operations

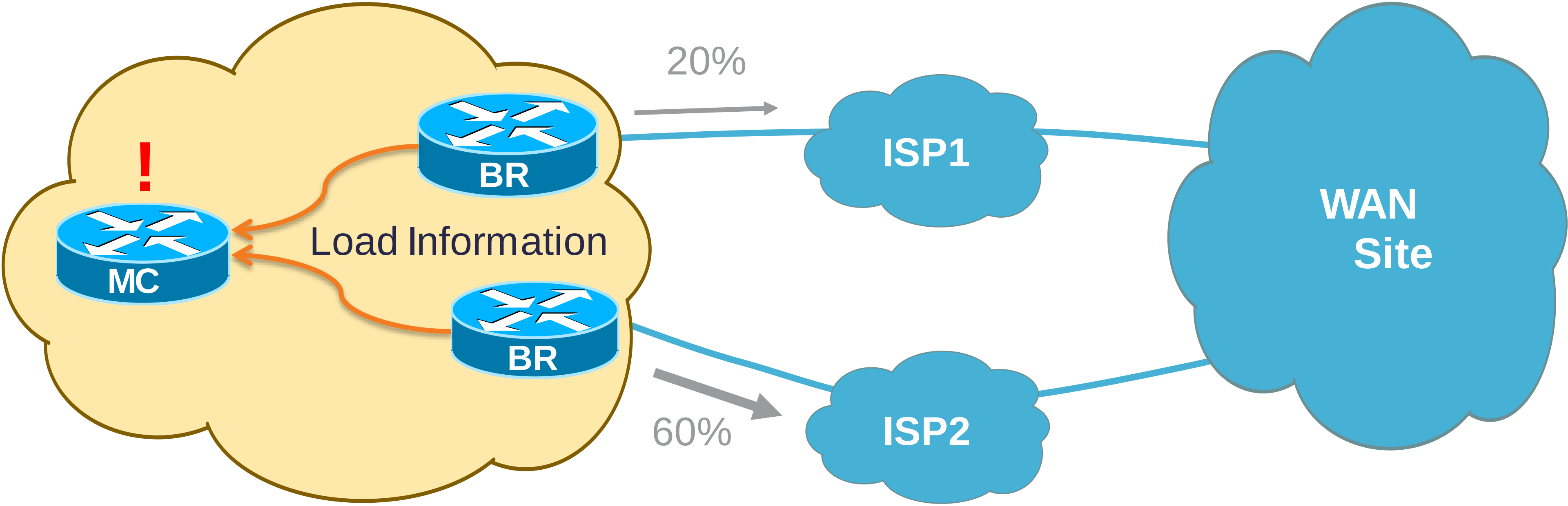
- Master Controller analyses reports from Border Routers



# PfR Operations

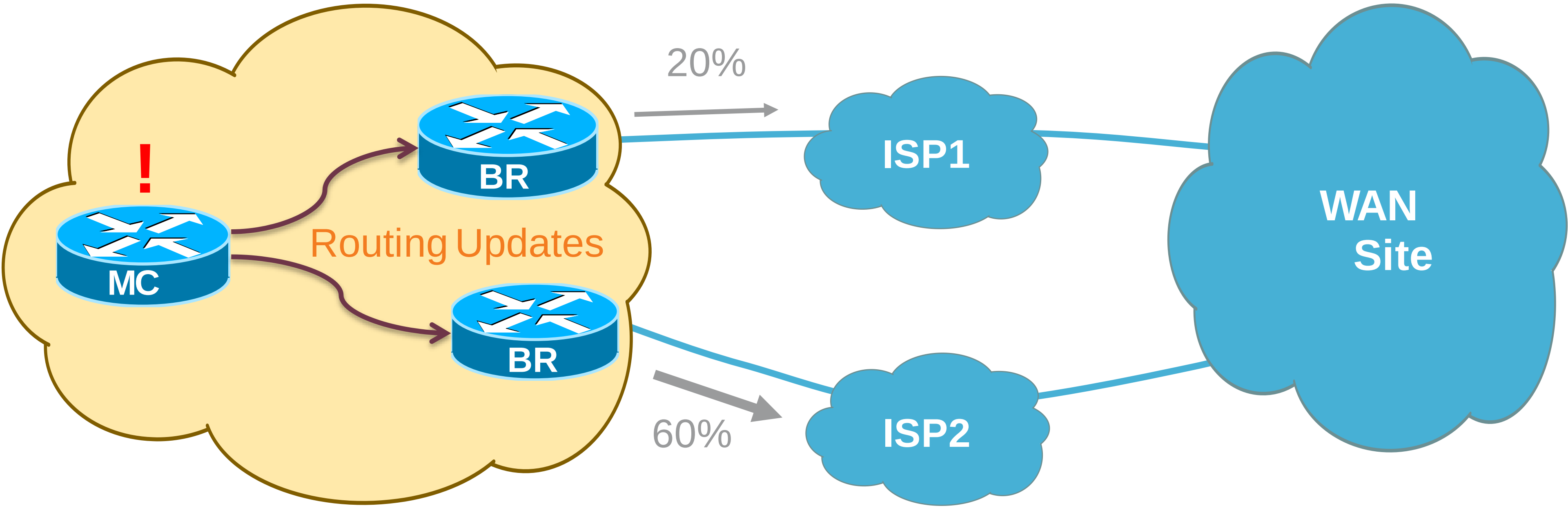
- Master Controller analyses reports from Border Routers
- MC detects policy violation



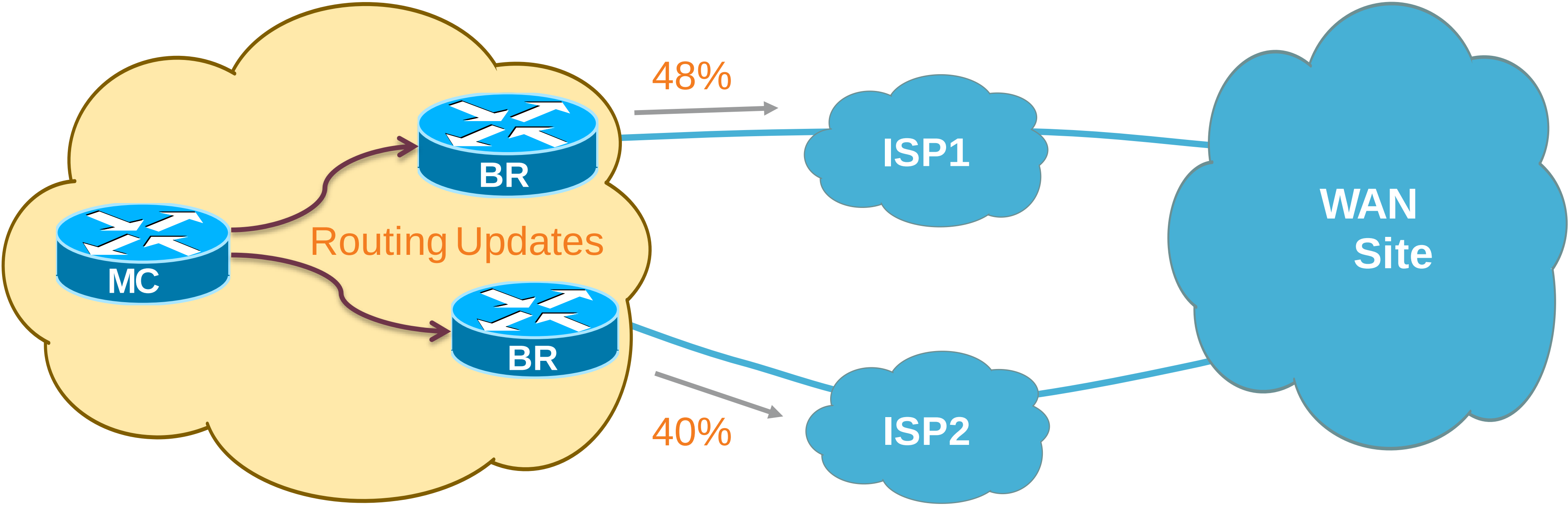
# PfR Operations

- Master Controller pushes routing updates



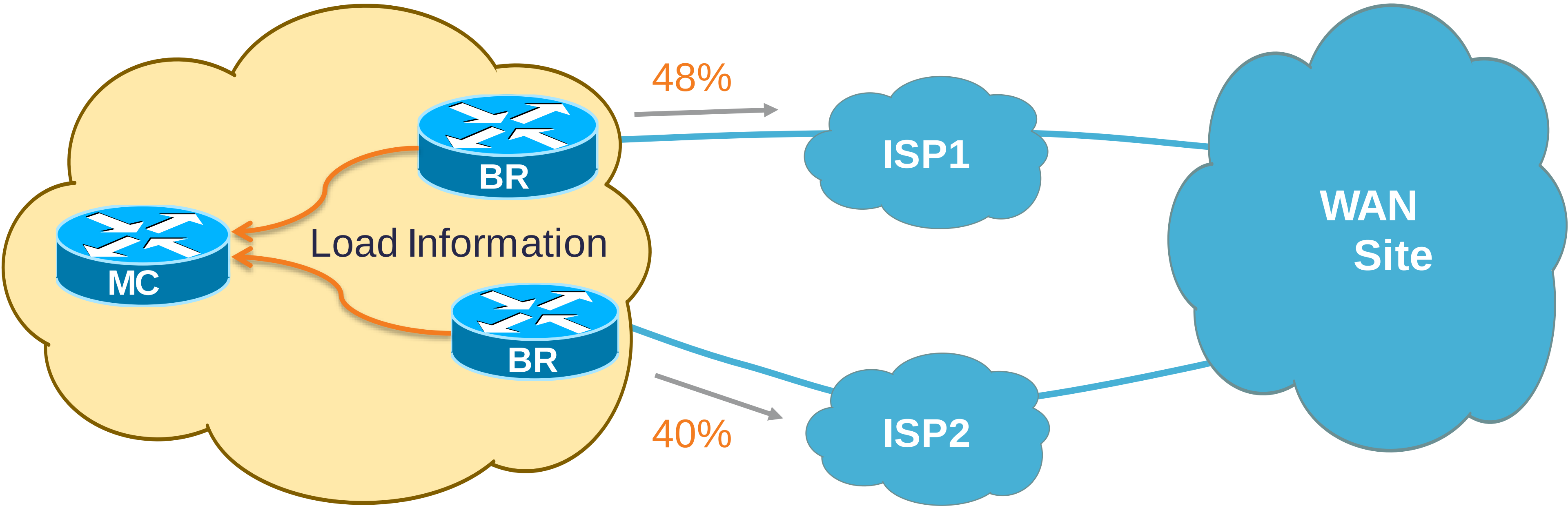
# PfR Operations

- Master Controller pushes routing updates
- Border Routers adjust routing impacting load



# PfR Operations

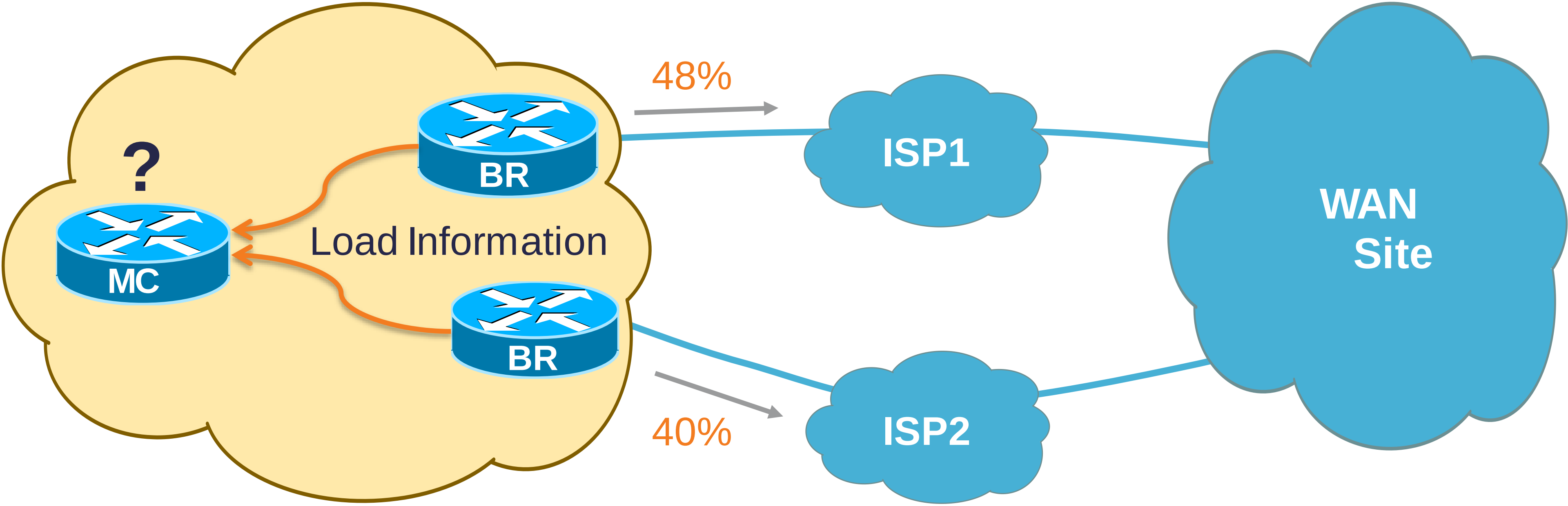
- Border Routers continue reporting



# PfR Operations

- Border Routers continue reporting
- Master Control continues analysing



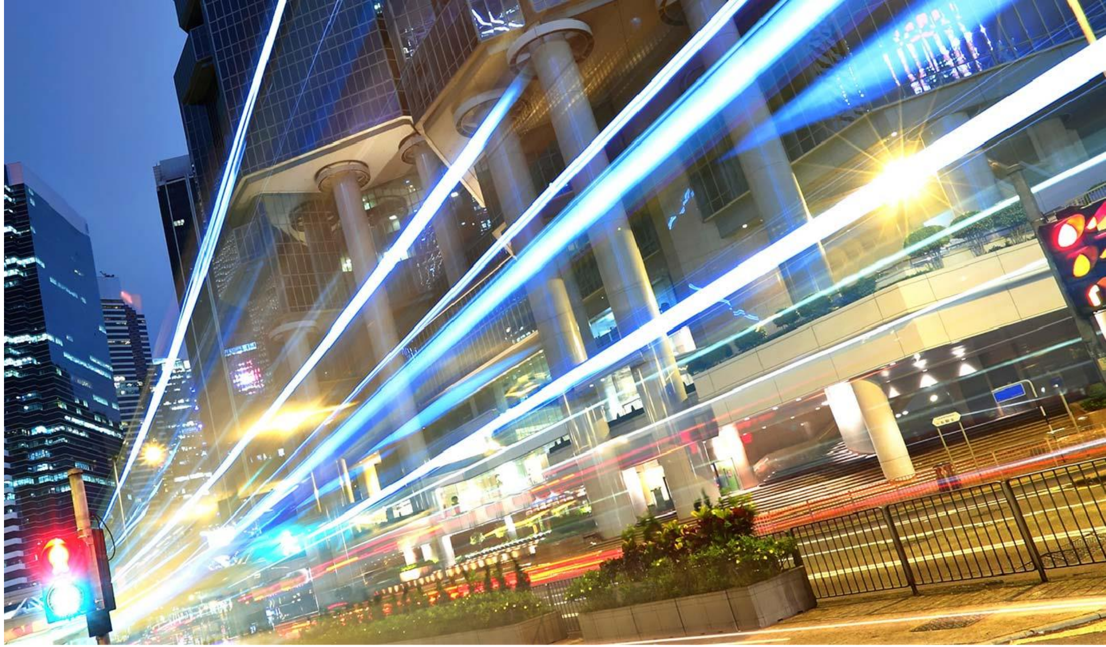
# PfR Summary

- PfR “lifecycle”
- Policy Enforcement
  - BGP Local Preference
  - Static Routes
  - PBR
- PfR provides routing intelligence
- CEF and RIB are the same

# Agenda

- Router Components
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing

➢ **Routing Convergence Improvements**
  - ➢ **Fast Convergence Overview**
  - – OSPF LFA
  - – EIGRP Feasible Successor
  - – BGP PIC-Edge



# Routing Convergence – What's to improve?

- Routing changes are bad
- Small changes can require (potentially) large recalculation
- Routing Protocols are slow
  - Failure detection is fast
  - Event propagation + calculation is the bottleneck
- Chain Reaction
  - Protocol Change -> RIB Change -> CEF Change
- Protocol can already know what to do before failure

# Failure Detection with BFD

- Bidirectional Forwarding Detection
- VERY fast (50ms hello/150ms dead)
- Lightweight
  – 24 bytes BFD Hello vs. 56 byte OSPF Hello
- Handled in Interrupt
- Protocols are BFD clients
- Offloaded to hardware*

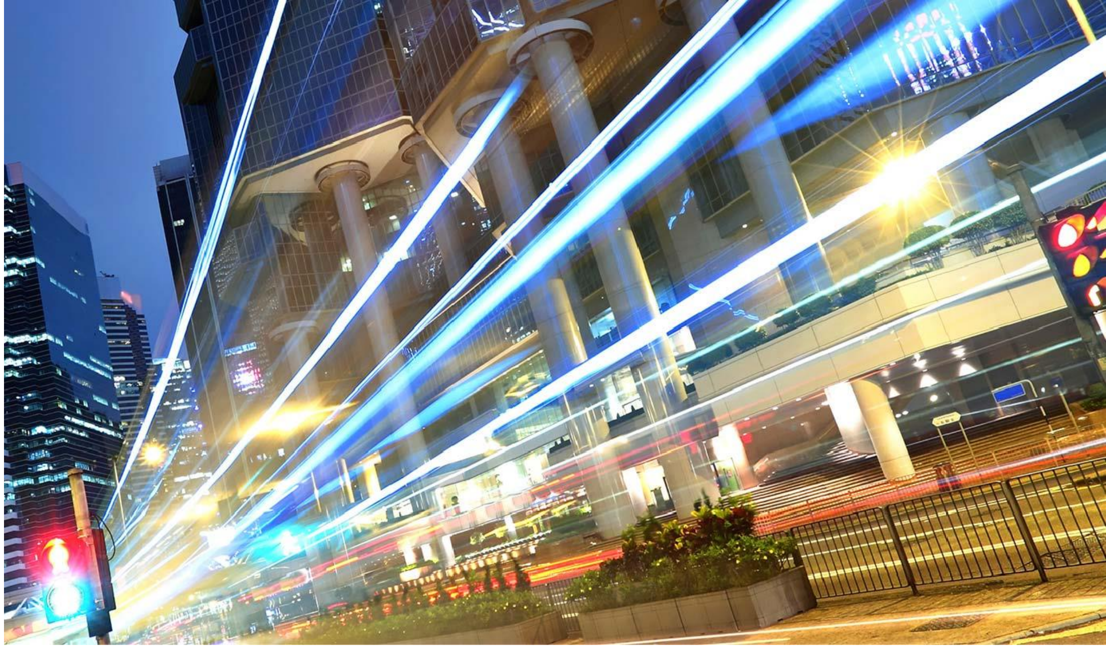
- *12k, 7600 with ES+, Nexus 7000, ASR1000

# Agenda

- Router Components
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing

➢ **Routing Convergence Improvements**
  - Fast Convergence Overview
  - ➢ **OSPF LFA**
  - EIGRP Feasible Successor
  - BGP PIC-Edge



# OSPF Overview

- Link State Algorithm
  - LSDB provides a view of the entire network
- Network changes exchanged via LSA (Link State Advertisement)
  - Multiple events cause throttling (5000ms default)
- SPF algorithm determines best path
  - Runs on receipt of LSA, delayed 5000ms (default)

# OSPF Convergence

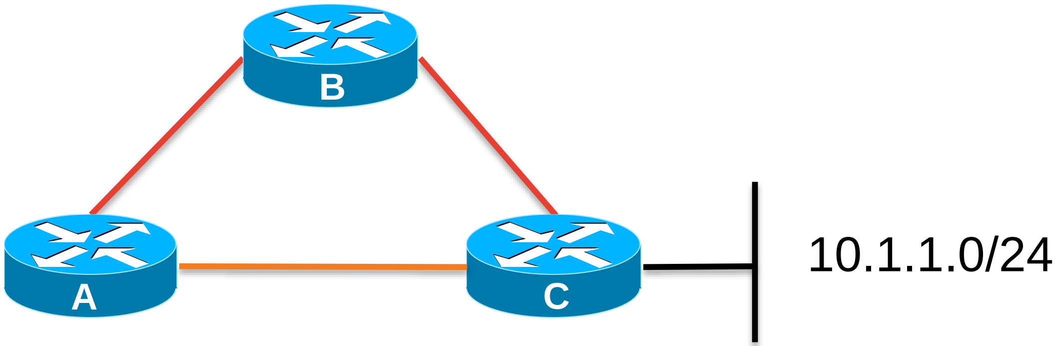
- Convergence =

Failure Detection + Event Propagation + SPF + FIB Update

Neighbour Down    LSA generation    RIB + CEF + Hardware

- Best case: ~160ms (SPF Tuning + BFD)
- Worst case: ~50 seconds (Dead Time + LSA throttle + SPF defaults)
- Failure Detection is easy (hardware)
- Control plane is difficult (software)

# OSPF Loop Free Alternate




- A has a primary (A-C) and secondary (A-B-C) path to 10.1.1.0/24
- Link State allows A to know entire topology
- A should know that B is an alternative path
- Loop Free Alternate (LFA)

# OSPF Loop Free Alternate

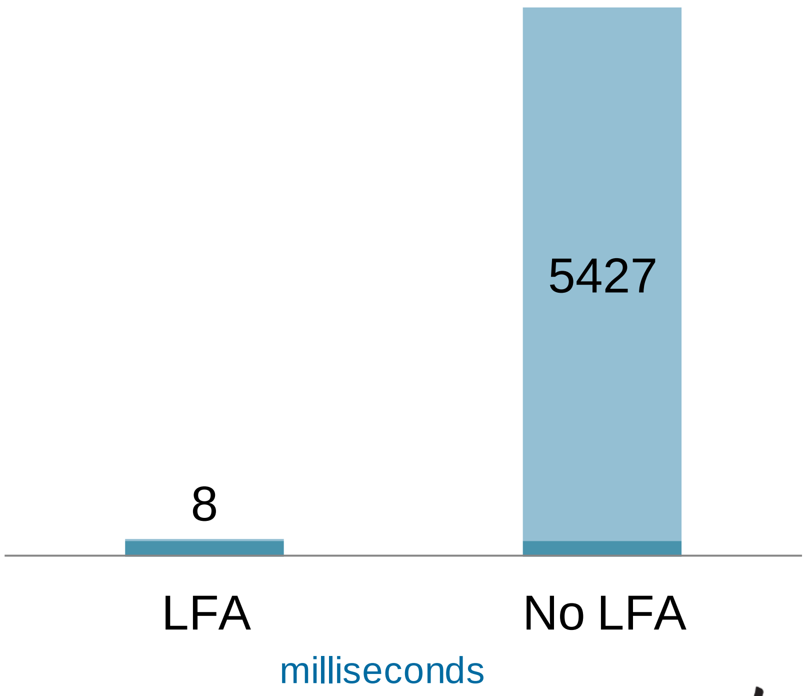
- OSPF presents a primary and backup to CEF
  - Backup calculated from secondary SPF run

```
RouterA# show ip route 10.1.1.0
Routing Descriptor Blocks:
 * 172.16.0.1, from 192.168.255.1, 00:01:57 ago, via Ethernet4/1/0
      Route metric is 2, traffic share count is 1
      Repair Path: 192.168.0.2, via Ethernet4/2/0

RouterA#show ip CEF 10.1.1.0
10.1.1.0/24
  nexthop 172.16.0.1 Ethernet4/1/0
    repair: attached-nexthop 192.168.0.2 Ethernet4/2/0
```

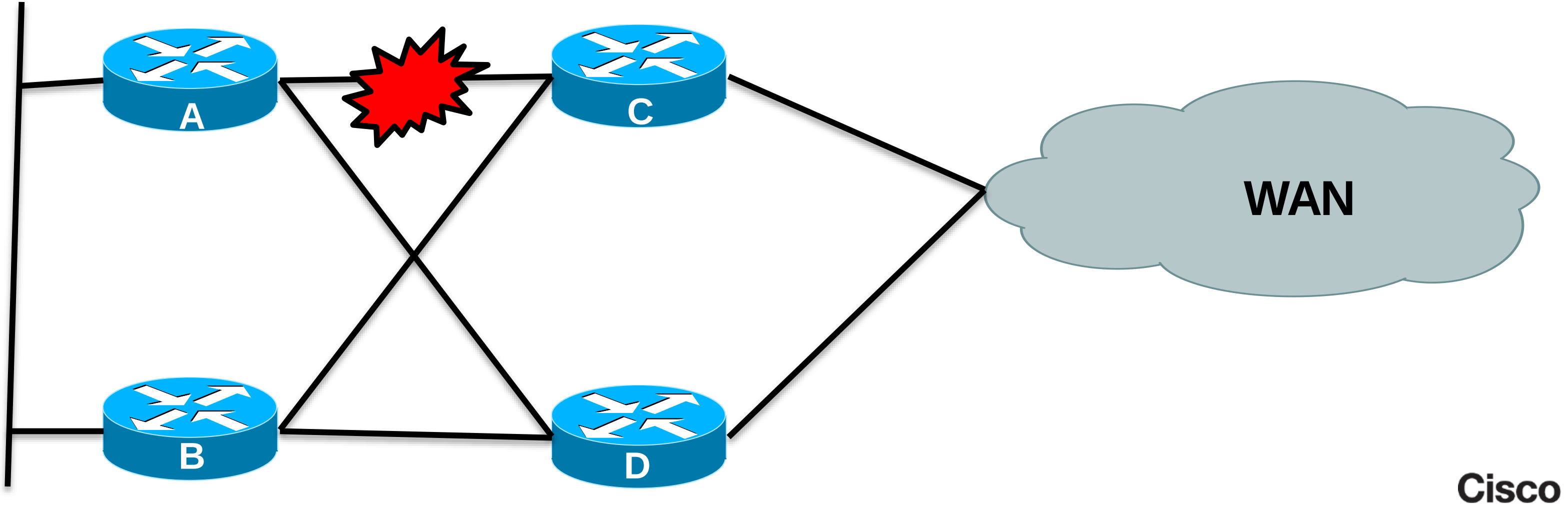
# OSPF Loop Free Alternate

- Aims for <50ms reconvergence
- Triggers as soon as the failure is detected
  - NO fast hellos
  - Use BFD!
- Not enabled by default
  - Added to 7600/ASR1000 in 15.1(3)S
  - Added to NX-OS in 5.0(2)



# OSPF Loop Free Alternate

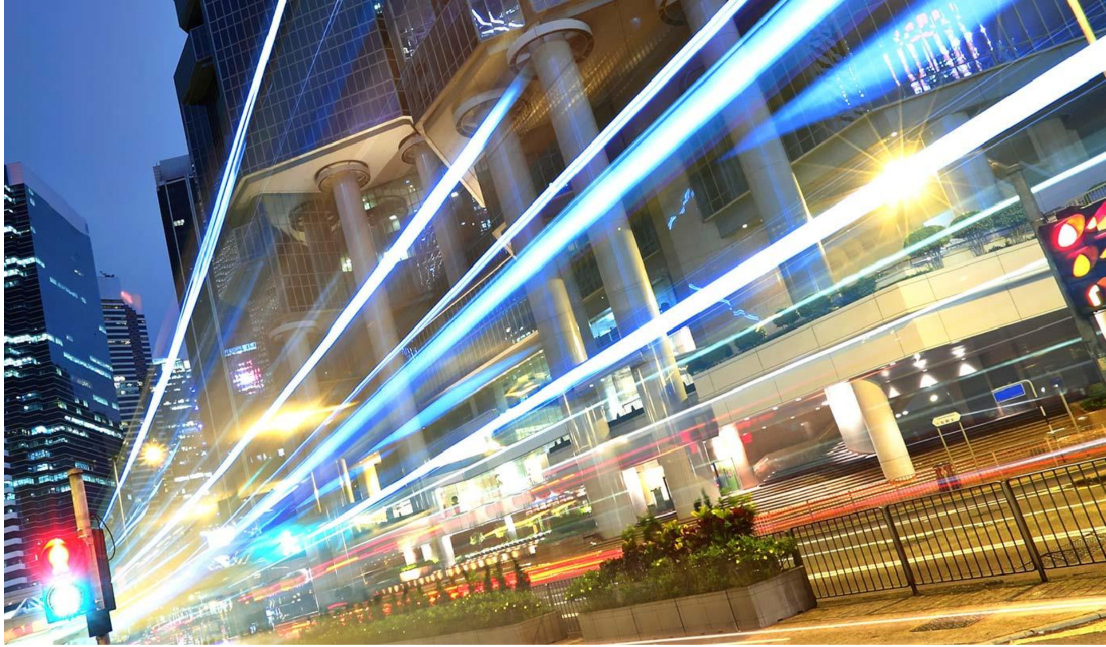
- Fast failure detection is key!
- Single Box
- Not a replacement for SPF Tuning



# Agenda

- Router Components
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing
- **Routing Convergence Improvements**
  - Fast Convergence Overview
  - OSPF LFA
  - **EIGRP Feasible Successor**
  - BGP PIC-Edge



# EIGRP Overview

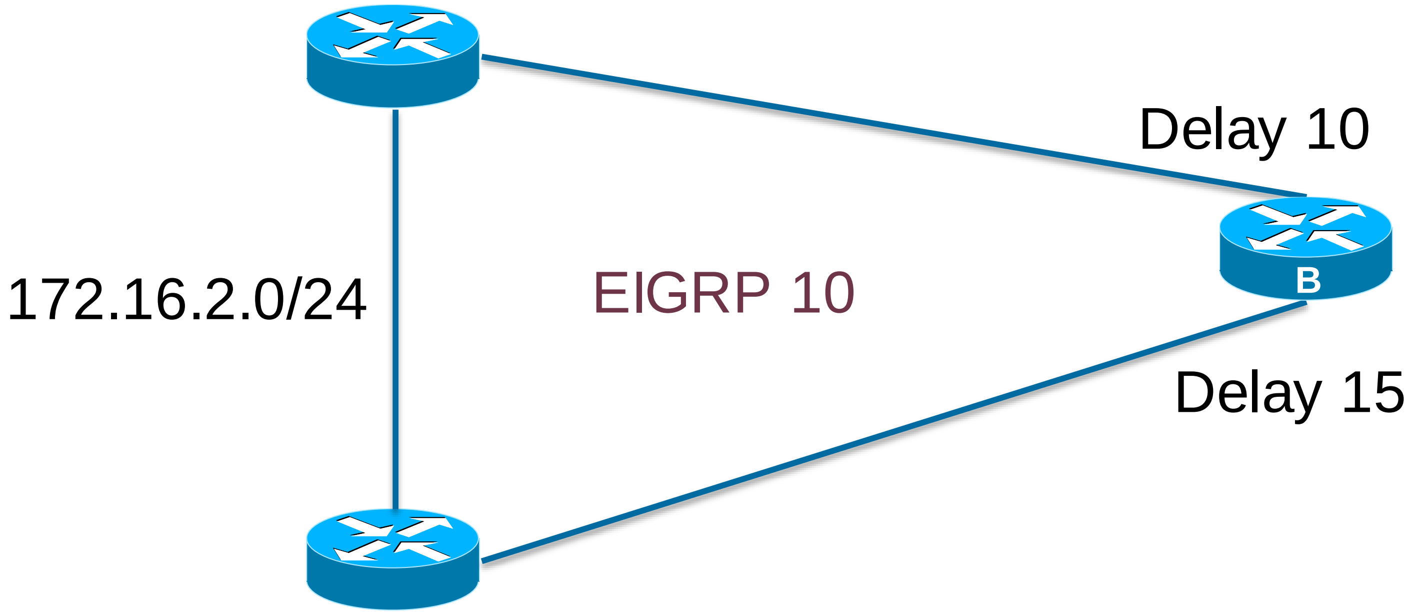
- Distance Vector Protocol
  - Doesn't see the entire network like OSPF
- Based on QUERY and ACK messages for convergence
  - QUERY sent to determine best path for failed route
  - ACK sent when alternative path found or no other paths
- DUAL algorithm determines best path
  - Runs as soon as all outstanding QUERIES are received
- Query domain size can effect convergence time

# EIGRP Feasible Successors

- EIGRP selects Successor and Feasible Successor
- Successor is the best route
- Feasible Successor is 2nd best route
- Must be mathematically loop-free (meets feasibility condition)
- Feasible Successor acts as a “backup route”
- Kept in topology table (not routing table)
- Up to 6 Feasible Successors
- Built into the protocol, nothing to enable

# EIGRP Feasible Successors

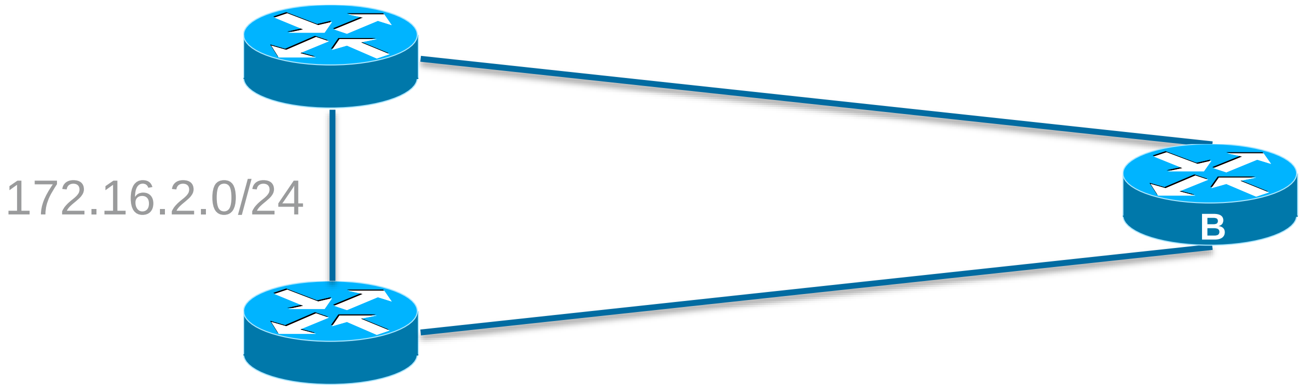



Metric based on bandwidth and delay

# EIGRP Feasible Successors

```
RouterB# show ip route 172.16.2.0
Routing entry for 172.16.2.0/24
  Known via "eigrp 10", distance 90, metric 285440, type internal
  Routing Descriptor Blocks:
  * 192.168.200.1, from 192.168.200.1, 00:34:19 ago, via Eth0/1
      Route metric is 285440, traffic share count is 1
```



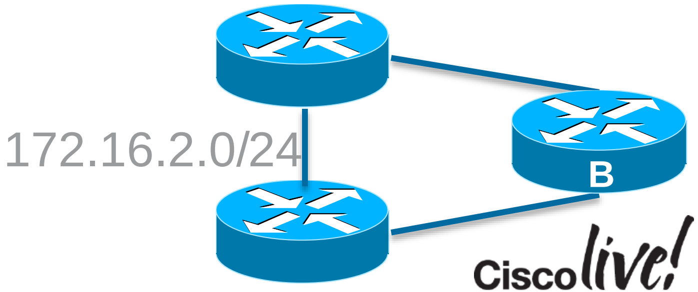

  

# EIGRP Feasible Successors

```
RouterB#show ip eigrp topology
P 172.16.2.0/24, 1 successors, FD is 285440
        via 192.168.200.1 (285440/281600), Ethernet0/1
        via 172.16.1.1 (307200/281600), Ethernet0/0
```

Feasible Successor reported distance (281600) is less than Successor feasible distance (285440)

- Feasibility Condition met
- Instant convergence after Successor loss



# EIGRP LFA

- Just like OSPF LFA
- Feasible Successors acts as Loop Free Alternate
- Installs Feasible Successors in hardware for instant failover
- EIGRP Fast Reroute available in 15.2.4S
- Not enabled by default

# EIGRP LFA

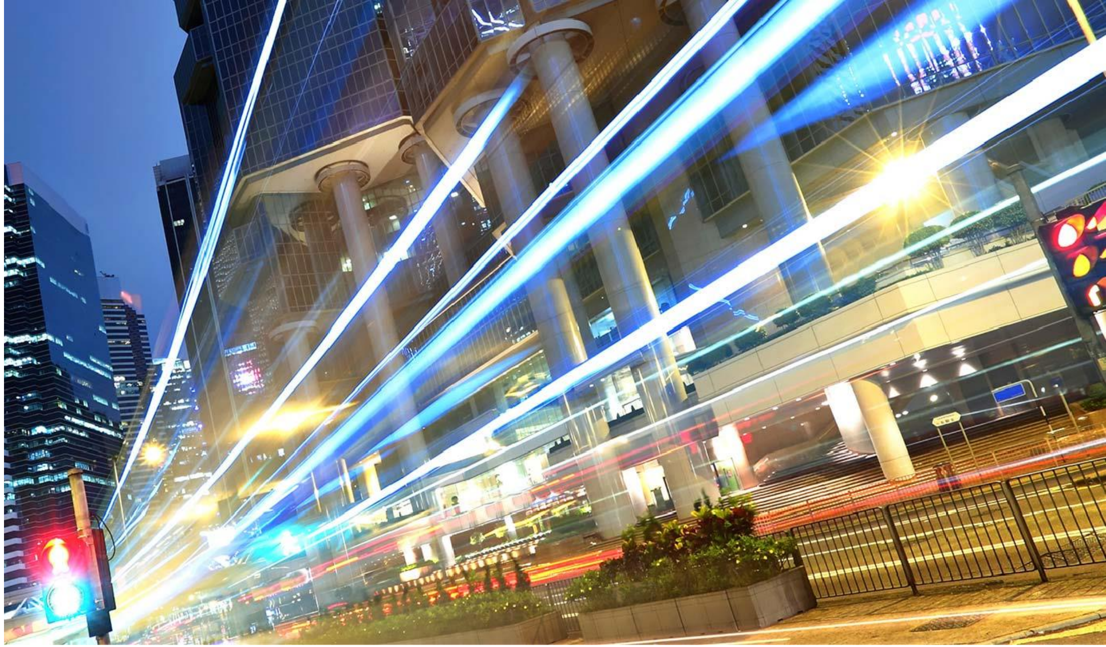
```
RouterB#show ip route 172.16.2.0
 Known via "eigrp 10", distance 90, metric 1100800, type
internal
   * 172.16.1.2, from 172.16.1.2, 00:00:17 ago, via Ethernet0/1
      Route metric is 281600, traffic share count is 1
      Repair Path: 192.168.1.1, via Ethernet0/0


RouterB#show ip cef 172.16.2.0
172.16.2.0/24
  nexthop 172.16.1.2 Ethernet0/1
    repair: attached-nexthop 192.168.1.1 Ethernet0/0
```

# Agenda

- Router Components
- Moving Packets
- CEF, CPU and Memory
- Outbound Load Sharing

➢ **Routing Convergence Improvements**
  – Fast Convergence Overview
  – OSPF LFA
  – EIGRP Feasible Successor
  ➢ **BGP PIC-Edge**



# BGP Prefix Independent Convergence

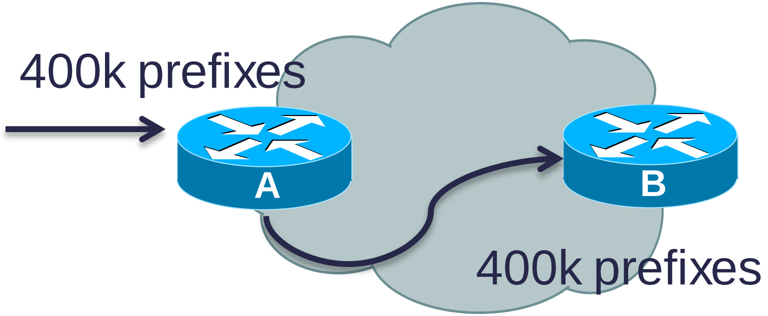
- Today's RIB is flat

10.1.1.0/24 → 192.168.1.1

10.1.2.0/24 → 192.168.1.1

10.1.3.0/24 → 192.168.1.1

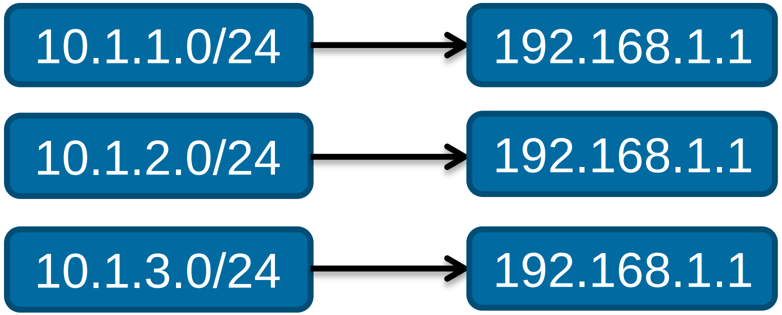
- 400k routes -> 400k updates
- BGP often has same next hop
- We can do better!



# BGP Prefix Independent Convergence

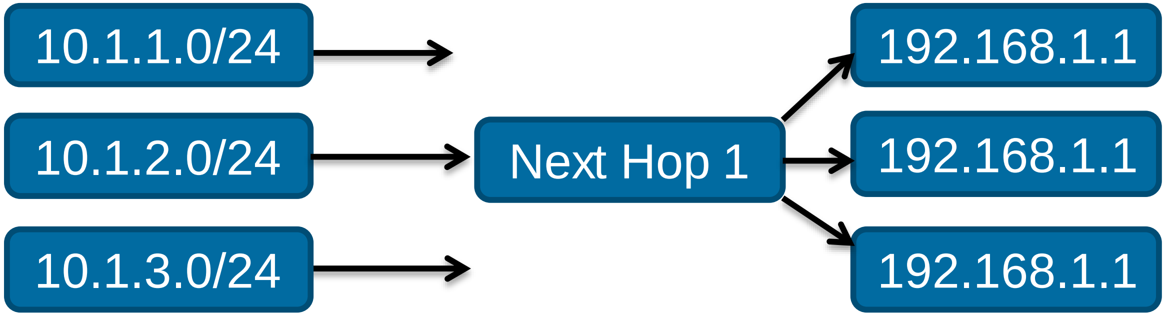
- Instead of flat FIB, Hierarchical



# BGP Prefix Independent Convergence

- Instead of flat FIB, Hierarchical

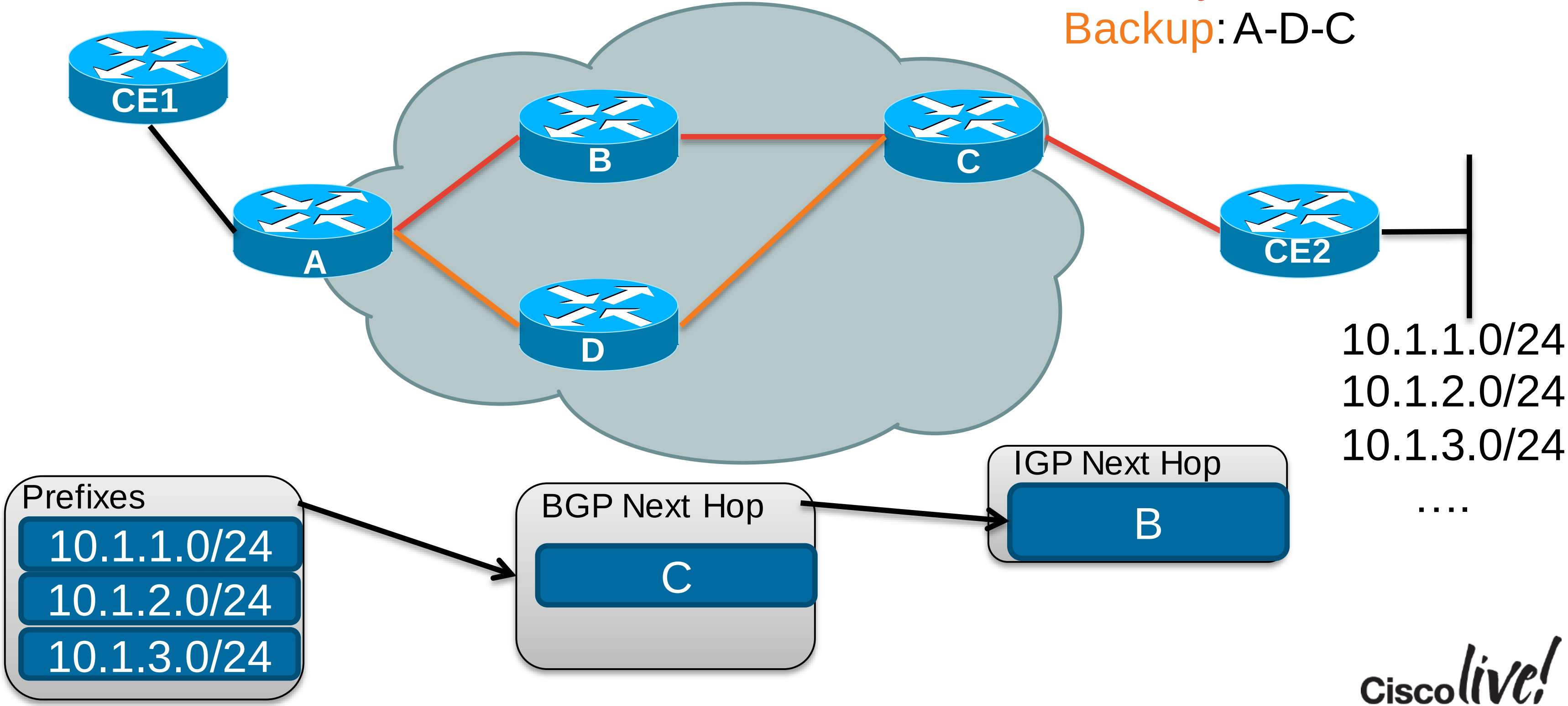



- Single change updates multiple entries
- Convergence time independent from prefix count

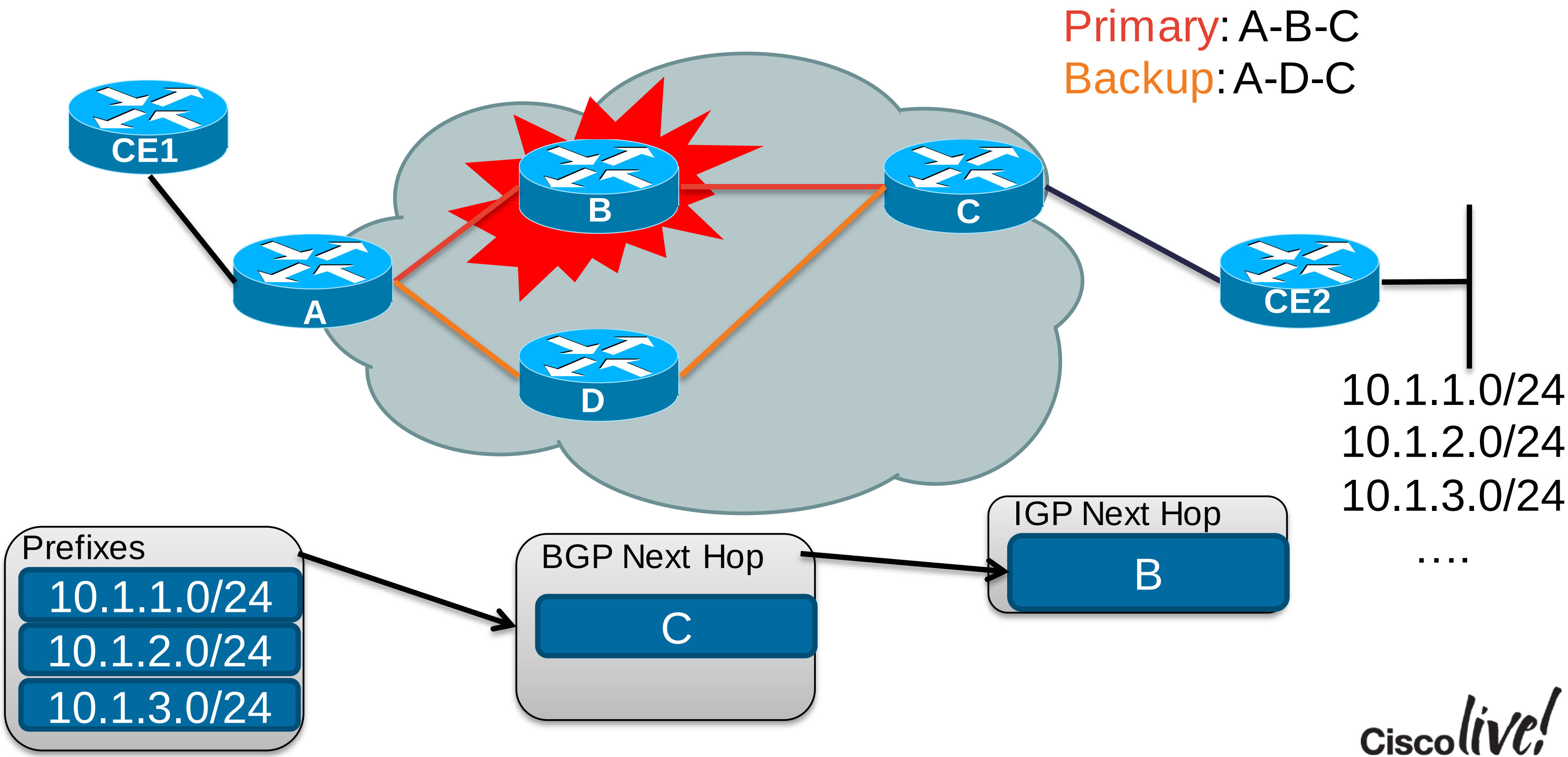
# BGP PIC Core

Primary: A-B-C
Backup: A-D-C



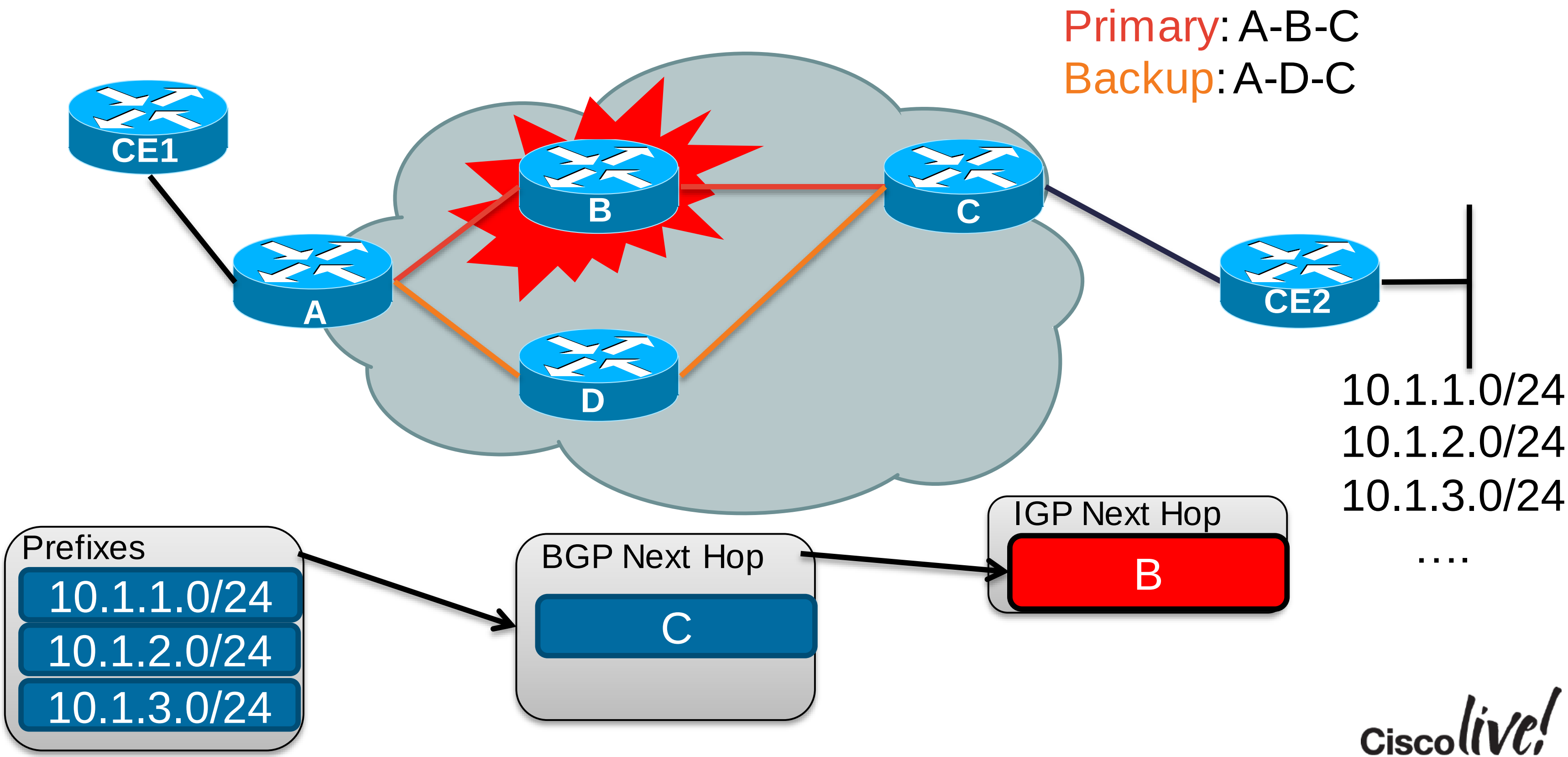
# BGP PIC Core



Primary: A-B-C
Backup: A-D-C
CE1
A
B
C
D
CE2
10.1.1.0/24
10.1.2.0/24
10.1.3.0/24
....
Prefixes
10.1.1.0/24
10.1.2.0/24
10.1.3.0/24
BGP Next Hop
C
IGP Next Hop
B

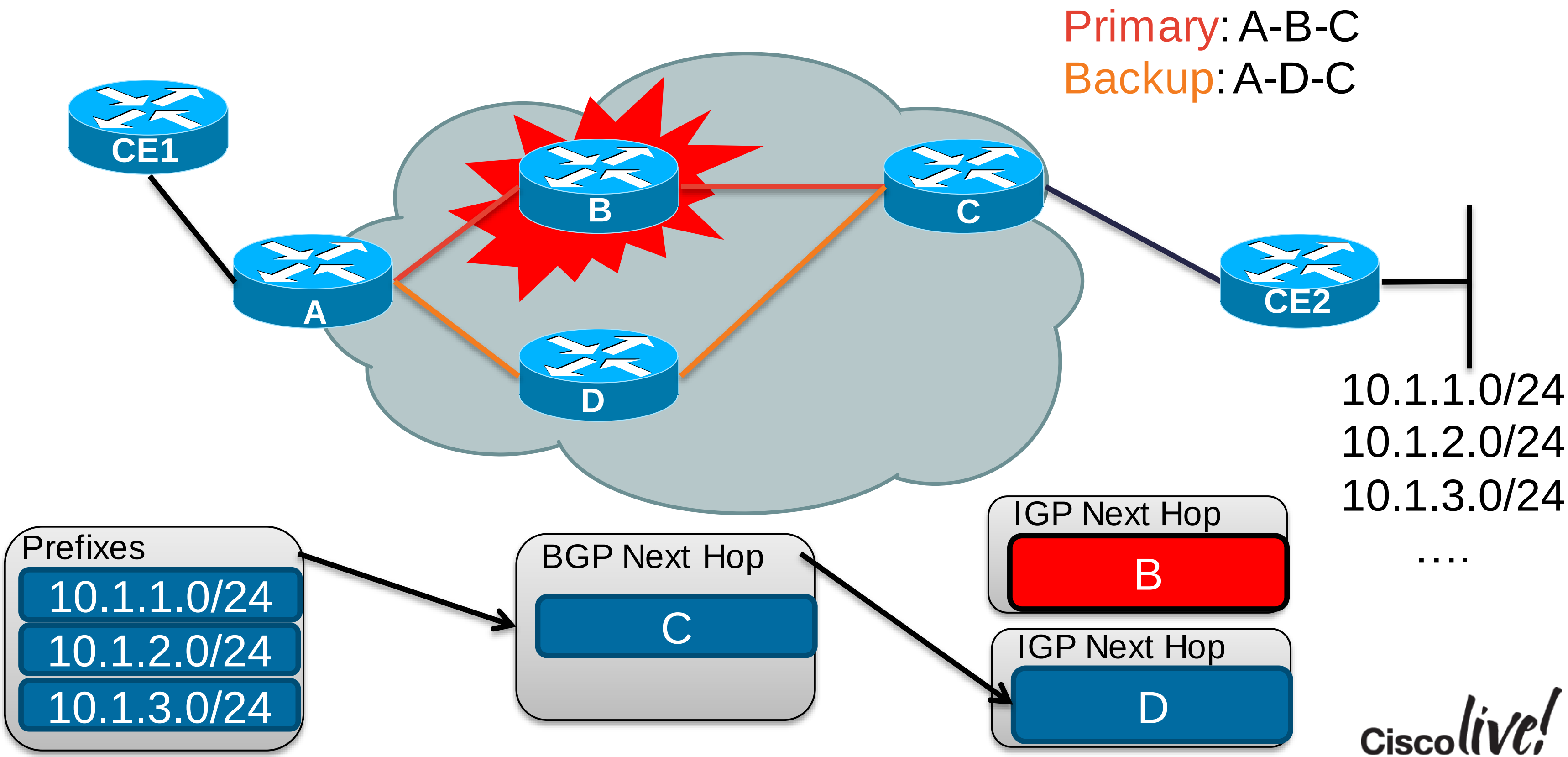
# BGP PIC Core



Primary: A-B-C
Backup: A-D-C
CE1
A
B
C
D
CE2
10.1.1.0/24
10.1.2.0/24
10.1.3.0/24
....
Prefixes
10.1.1.0/24
10.1.2.0/24
10.1.3.0/24
BGP Next Hop
C
IGP Next Hop
B

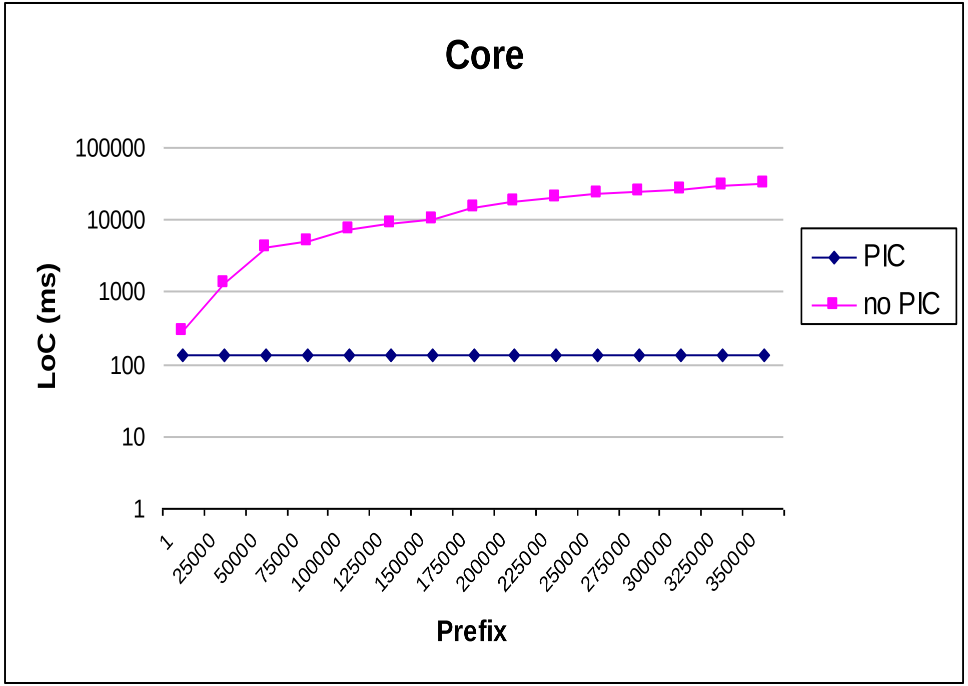
# BGP PIC Core



Primary: A-B-C
Backup: A-D-C
CE1
A
B
C
D
CE2
10.1.1.0/24
10.1.2.0/24
10.1.3.0/24
....
Prefixes
10.1.1.0/24
10.1.2.0/24
10.1.3.0/24
BGP Next Hop
C
IGP Next Hop
B
IGP Next Hop
D

# BGP PIC Core




- BGP convergences starts after IGP convergence

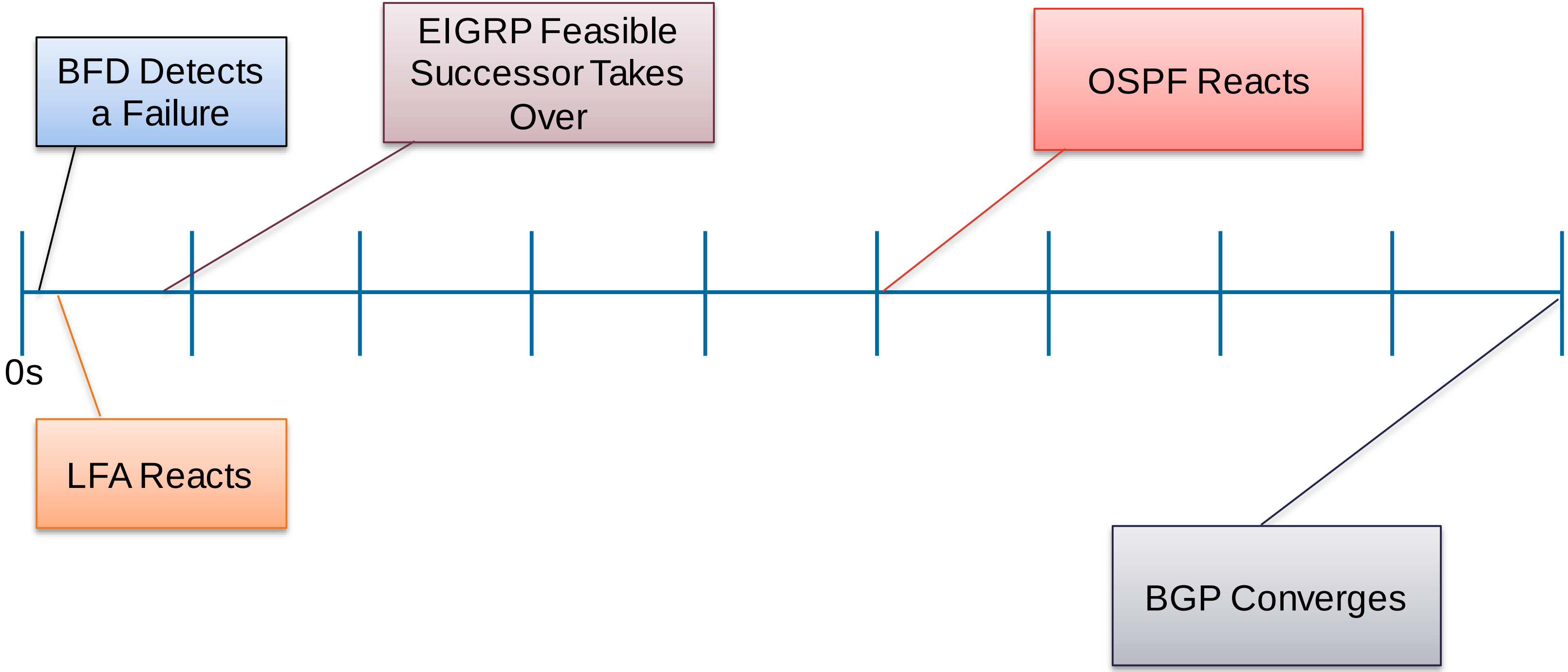
# BGP PIC Core

- PIC Core part of migration to hierarchical FIB
- Still requires IGP convergence
  - OSPF LFA
  - EIGRP FS and LFA
- PIC Edge
  - Mainly for MPLS/VPN environments
  - Fast convergence for edge node failures
  - Beyond the scope of today's talk

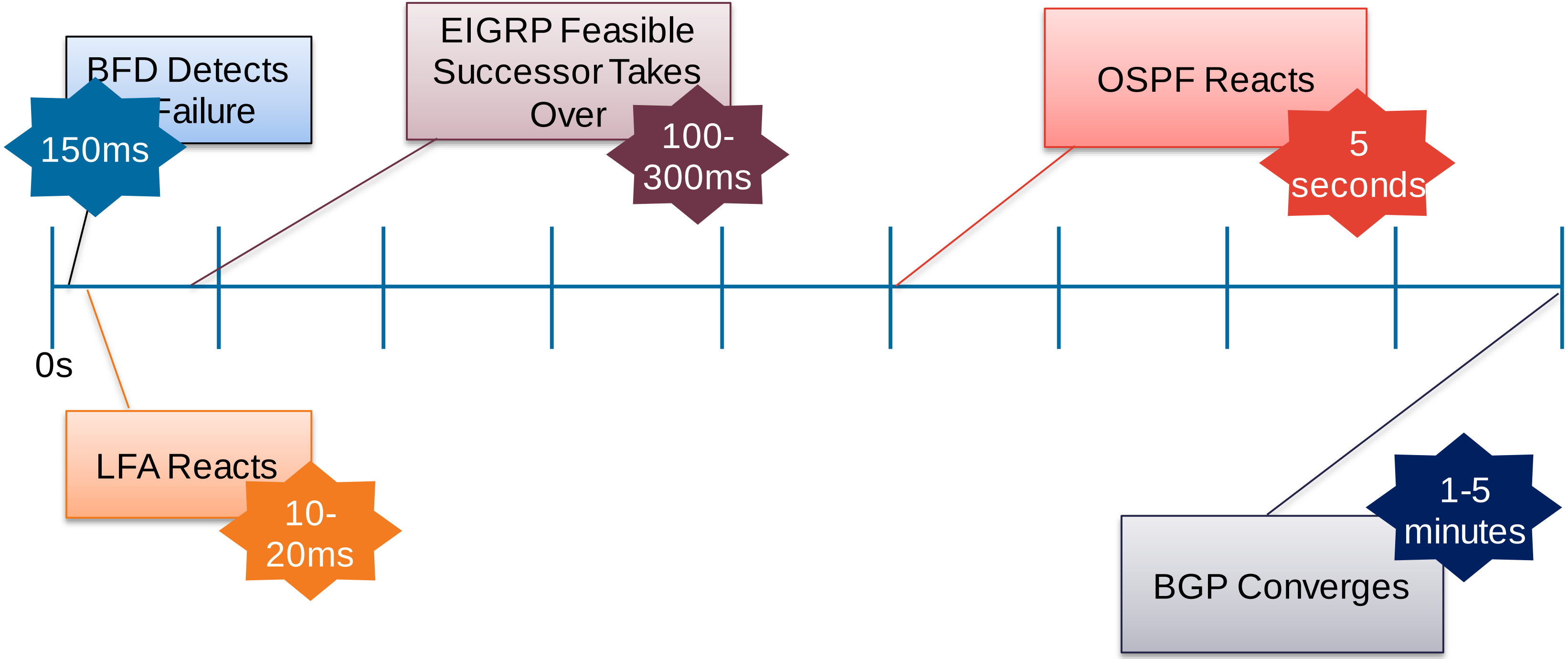
```
7600(config)# cef table output-chain build favor convergence-speed
```

# Fast Convergence Timeline



# Fast Convergence Timeline

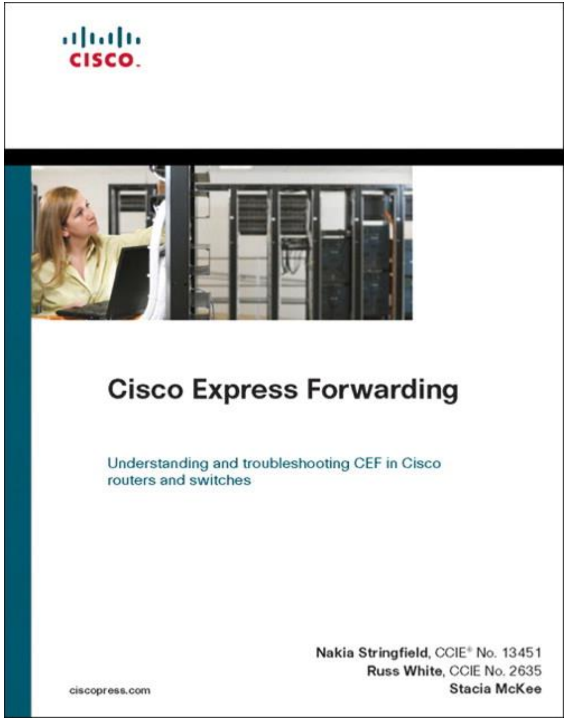


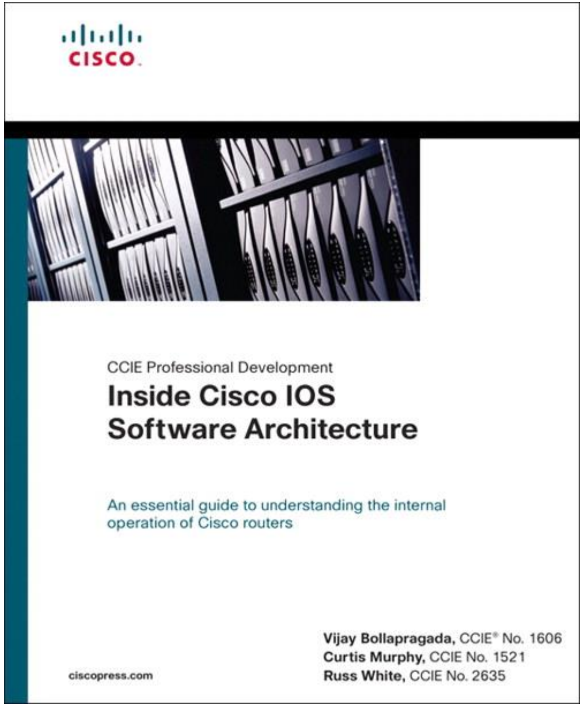

  

# Review

- Router Components
  - Control vs. Data plane
  - Software vs. Hardware vs. Hybrid based routers
- CPU and Memory
  - Interrupt (CEF) vs. Process (Routing Protocol)
  - Memory concerns for multiple routes
- Load Sharing
  - CEF and PfR
- Routing Enhancements
  - OSPF LFA/EIGRP Feasible Successors/BGP PIC

# Further Reading



# Suggested Sessions

- BRKRST-3363 - Routed Fast Convergence
- BRKRST-2042 - Highly Available Wide Area Network Design
- BRKRST-2337 - OSPF Deployment in Modern Networks
- BRKRST-2336 - EIGRP Deployment in Modern Networks
- BRKARC-2019 - Operating an ASR1000
- BRKSPG-2000 - Getting the most out of your IOS-XE router before and after deployment
- BRKSPG-2904 - ASR-9000/IOS-XR Understanding forwarding, troubleshooting the system and XR operations
- BRKARC-3465 - Cisco Catalyst 6800 Switch Architectures
- BRKARC-3470 - Advanced - Cisco Nexus 7000/7700 Switch Architecture

# Continue Your Education

- Demos in the Cisco Campus
- Walk-in Self-Paced Labs
- Table Topics
- Meet the Engineer 1:1 meetings

# Q & A

# Complete Your Online Session Evaluation

**Give us your feedback and receive a Cisco Live 2015 T-Shirt!**

Complete your Overall Event Survey and 5 Session Evaluations.

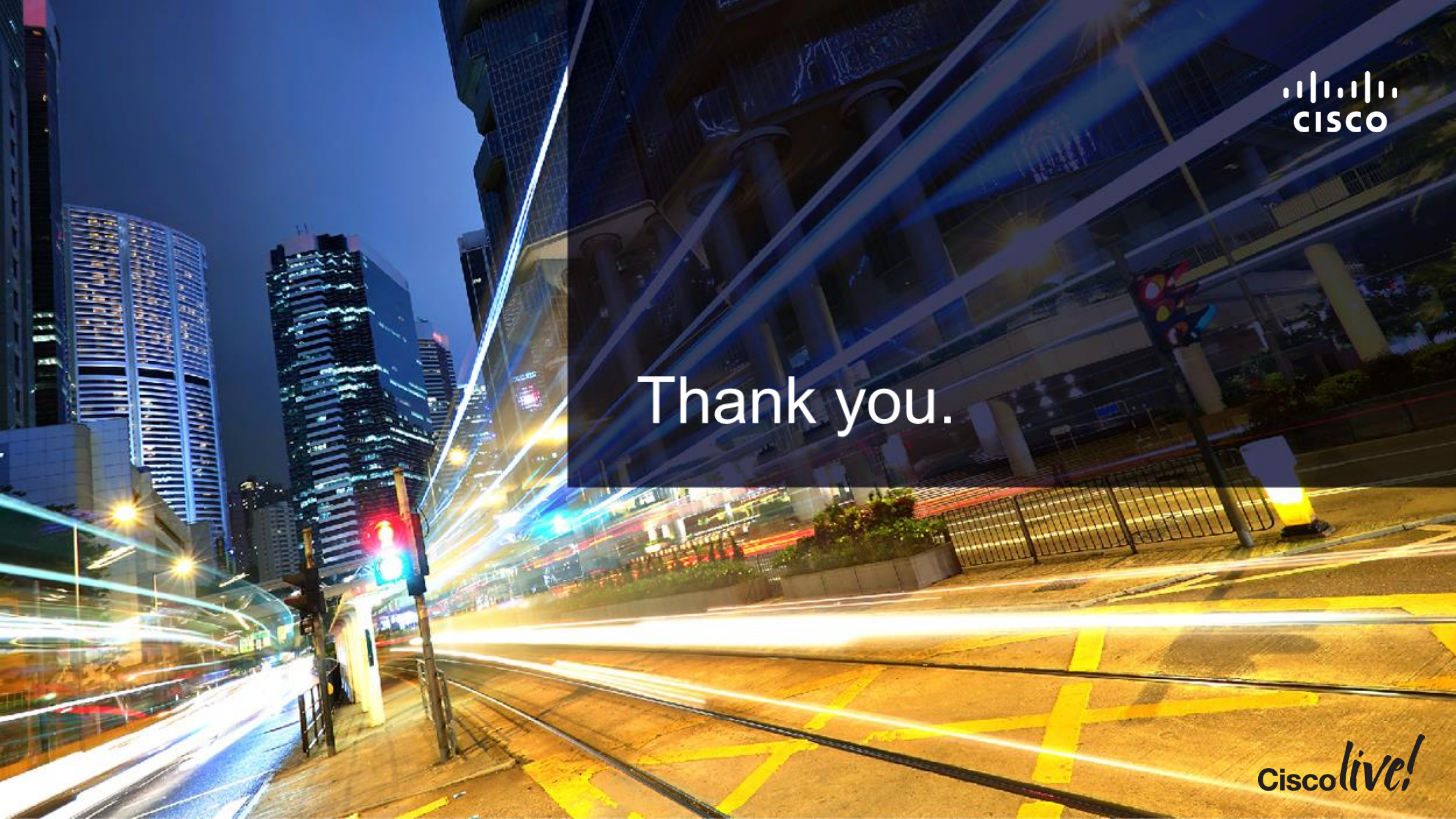
- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site http://showcase.genie-connect.com/clmelbourne2015
- Visit any Cisco Live Internet Station located throughout the venue

T-Shirts can be collected in the World of Solutions on Friday 20 March 12:00pm - 2:00pm

**Learn online with Cisco Live!**
Visit us online after the conference for full access to session videos and presentations. www.CiscoLiveAPAC.com

Cisco live!


Thank you.

cisco