Router Architecture And IOS Internals
Agenda

- Routing and Switching
- Cisco IOS Switching Paths
- Cisco Express Forwarding
- Router Architectures & Parallel Express Forwarding
Routing and Switching
Switching

- The destination in the layer two header remains the same when a packet passes through a switch.
### Routing

- Host A transmits the packet to the router.
- The router determines the correct outbound port, then rewrites the layer 2 header so the packet is now destined to B.

<table>
<thead>
<tr>
<th>Layer 2: to router</th>
<th>Layer 3: to B</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram of routing process](image-url)
Switching, in the context of routers, involves this process of looking up the next hop, finding the layer 2 rewrite “string,” rewriting the layer 2 header, and transmitting the packet.
Layer 3 Switching

• When the term “layer 3 switch” was first coined, it meant switching packets in hardware based on the layer 3 information

• However, the lines are rarely so neatly drawn in the real world
Routing Protocols and Other Sources Are Used to Build the Routing Table
Layer 3 Switching

ARP and Other Methods Are Used to Build the Layer 2 Mapping Tables
Layer 3 Switching

These Tables Are Used to Build a **Switching Table**

- **Information Sources**
- **Routing Table**
- **Layer 2 Mapping**

Control Plane vs. Data Plane
Layer 3 Switching

The Switching Table is then used to switch packets.
Layer 3 Switching

- Where is switching done?
  
  On the main processor, in a “normal” process
  On the main processor in a special mode (interrupt context)
  On a separate general purpose processor
  On an application specific chip (ASIC)
Cisco IOS Switching Paths
IOS Process Scheduling

- Each disk represents a *Process* in the *Process Ready Queue*.
- Each *Process* is assigned a *Priority* (Critical, High, Medium or Low)
Router Switching Operation
“Process Switching”

1

Interface Processor
DMA’s packet into RX
Ring Buffer

Software ‘Processes’….

Shared memory
Router Switching Operation
“Process Switching”

Interface Driver Code
Decodes packet header and builds Buffer Header with L3 Info
Router Switching Operation
“Process Switching”

Interface processor generates RX Interrupt to CPU.
Router Switching Operation
“Process Switching”

Software ‘Processes’ are resumed at the point they were suspended when the RX Interrupt arrived.

When Packet passed to Processor, Buffer ownership transferred to Processor.

As Ownership has passed Interrupt released.

Shared memory
Router Switching Operation
“Process Switching”

Processor returns to scheduled tasks. Packet is placed on Input Hold Q (protocol dependant). Packet is idle waiting for Input Process to deal with Packet.
Router Switching Operation
“Process Switching”

Input Process Looks up Destination in Forwarding Table. Determines O/P interface. Writes new MAC header. Places Packet in Output Q
Router Switching Operation
“Process Switching”

Software ‘Processes’....

Buffer | Header

System buffer

Shared memory

Output Process places packet on output interface TX Ring Buffer.

© 2001, Cisco Systems, Inc. All rights reserved.
Router Switching Operation
“Process Switching”

CPU

Buffer | Header

Shared memory

Tx Ring

Interface polls TX ring and DMA’s packets for transmission

Software ‘Processes’....
Router Switching Operation
"Process Switching"

Interface Instigates a TX interrupt. Increment counters, SNMP etc..

Software ‘Processes’….
Demand Generated Cache Based Switching (“Fast” Switching)

**Forwarding Table**
- 1.1.0.0/16 via 172.16.2.1
- 10.1.1.0/24 via 172.16.1.1

**ARP Table**
- 172.16.1.1: 0F000800
- 172.16.2.1: 10134567A...ECE030178654

**Fast Cache**
- Prefix/Length
  - 1.1.0.0/16
  - 10.1.1.0/24
- Age
  - 00:00:15
- Interface
  - Ethernet0
  - Serial1
- Next Hop
  - 172.16.2.1 14 00000C7EF7CF00E0B06423F60800
  - 172.16.1.1 4 0F000800
Router Switching Operation
"Fast" Switching

1
Interface Processor
DMA’s packet into RX
Ring Buffer
Router Switching Operation
"Fast" Switching

Interface Driver Code
Decodes packet header and builds Buffer Header with L3 Info

Software ‘Processes’…. X

Presentation_ID  © 2001, Cisco Systems, Inc. All rights reserved. 26
Router Switching Operation
”Fast” Switching

Simplified Optimum Cache

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Age</th>
<th>I/F</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.2.3/32</td>
<td>00:00:15</td>
<td>E0</td>
<td>10.1.2.1 14 aae0cd..</td>
</tr>
<tr>
<td>11.1.2.0/24</td>
<td>00:00:15</td>
<td>S1</td>
<td>10.2.3.1 4 0f000800</td>
</tr>
</tbody>
</table>

3

Interface processor
generates RX Interrupt to
CPU.

CPU Halts current process
and attempts to fast switch
packet
Router Switching Operation
"Fast" Switching

Simplified Optimum Cache

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Age</th>
<th>I/F</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.2.3/32</td>
<td>00:00:15</td>
<td>E0</td>
<td>10.1.2.1 14 aae0cd..</td>
</tr>
<tr>
<td>11.1.2.0/24</td>
<td>00:00:15</td>
<td>S1</td>
<td>10.2.3.1 4 0f000800</td>
</tr>
</tbody>
</table>

Optimum Cache entry used to Write MAC header
Router Switching Operation
"Fast" Switching

If Output Q is Empty packet is placed directly on the TX Ring.

A packet in the Output Hold Q, will force other packets destined for that interface to be placed in the Q.
Router Switching Operation
"Fast" Switching

Interface Instigates a TX interrupt. Increment counters, SNMP etc..
Demand Generated Cache Based Switching Issues

- First packet towards a given destination is always process switched
- Fast cache entries must be timed out periodically to prevent stale information from being used in switching
- When an arp entry or the routing table changes, we must clear some portion of the fast cache and wait for process switched traffic to rebuild it
- We store a prebuilt mac header for each possible destination. This waste space and causes duplicated effort
Show Processes

7206#show processes

CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%

<table>
<thead>
<tr>
<th>PID</th>
<th>QTy</th>
<th>PC</th>
<th>Runtime (ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>Stacks</th>
<th>TTY</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>M*</td>
<td>0</td>
<td>8</td>
<td>86</td>
<td>93</td>
<td>9888/12000</td>
<td>0</td>
<td>Exec</td>
</tr>
<tr>
<td>3</td>
<td>Lst</td>
<td>60655C58</td>
<td>345736</td>
<td>129733</td>
<td>2664</td>
<td>5740/6000</td>
<td>0</td>
<td>Check heaps</td>
</tr>
<tr>
<td>4</td>
<td>Cwe</td>
<td>6064C268</td>
<td>4</td>
<td>1</td>
<td>4000</td>
<td>5568/6000</td>
<td>0</td>
<td>Chunk Manager</td>
</tr>
<tr>
<td>5</td>
<td>Cwe</td>
<td>6065BC70</td>
<td>12</td>
<td>17</td>
<td>705</td>
<td>5596/6000</td>
<td>0</td>
<td>Pool Manager</td>
</tr>
<tr>
<td>14</td>
<td>Lwe</td>
<td>60719100</td>
<td>5604</td>
<td>103710</td>
<td>54</td>
<td>5236/6000</td>
<td>0</td>
<td>ARP Input</td>
</tr>
<tr>
<td>20</td>
<td>Cwe</td>
<td>60661090</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5608/6000</td>
<td>0</td>
<td>Critical Bkngd</td>
</tr>
<tr>
<td>21</td>
<td>Mwe</td>
<td>6061BC70</td>
<td>232</td>
<td>209650</td>
<td>110164/12000</td>
<td>0</td>
<td>Net Background</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Lwe</td>
<td>605ACD38</td>
<td>0</td>
<td>26</td>
<td>011504/12000</td>
<td>0</td>
<td>Logger</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Msp</td>
<td>6061B1C0</td>
<td>32336</td>
<td>1277140</td>
<td>25</td>
<td>6920/9000</td>
<td>0</td>
<td>Per-Second Jobs</td>
</tr>
<tr>
<td>35</td>
<td>Mwe</td>
<td>60747998</td>
<td>4276</td>
<td>64668</td>
<td>6610648/12000</td>
<td>0</td>
<td>IP Input</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>Msp</td>
<td>6061B200</td>
<td>85188</td>
<td>21328</td>
<td>3994</td>
<td>5660/6000</td>
<td>0</td>
<td>Per-minute Jobs</td>
</tr>
</tbody>
</table>

For the 5 Sec window we have both the total CPU time and the Interrupt time
**Show Processes CPU**

```
7206#show processes cpu

CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%

<table>
<thead>
<tr>
<th>PID</th>
<th>Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>TTY</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>68</td>
<td>227</td>
<td>299</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Exec</td>
</tr>
<tr>
<td>3</td>
<td>368920</td>
<td>138425</td>
<td>2665</td>
<td>0.08%</td>
<td>0.02%</td>
<td>0.00%</td>
<td>0</td>
<td>Check heaps</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4000</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Chunk Manager</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>21</td>
<td>952</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Pool Manager</td>
</tr>
<tr>
<td>14</td>
<td>6608</td>
<td>119562</td>
<td>55</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>ARP Input</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Critical Bkgnd</td>
</tr>
<tr>
<td>21</td>
<td>248</td>
<td>218242</td>
<td>1</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Net Background</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Logger</td>
</tr>
<tr>
<td>24</td>
<td>35704</td>
<td>1362619</td>
<td>26</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Per-Second Jobs</td>
</tr>
<tr>
<td>35</td>
<td>4520</td>
<td>68993</td>
<td>65</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>IP Input</td>
</tr>
<tr>
<td>82</td>
<td>90896</td>
<td>22759</td>
<td>3993</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0</td>
<td>Per-minute Jobs</td>
</tr>
</tbody>
</table>
```

More specific information on the CPU time occupied by the Processes
Cisco Express Forwarding
Cisco Express Forwarding

- Background
- CEF Theory
- The CEF Mtrie
- The Adjacency Table
- Adjacency Table Entries
- Load Sharing with CEF
- CEF Accounting
Background: Process Level Switching

- Process Level Switching has speed limitations on high speed networks
Background: Fast Switching

- Caching the results of the lookup routines was the first solution and is known as Fast Switching.

- This solution encounters scalability problems on Internet backbone routers where the routing table is changing rapidly and there are many different flows of traffic.

- CEF (Cisco Express Forwarding) was developed to address the scalability issues of Process and Fast Switching.

- CEF doesn’t cache switching information, it builds switching tables.
What Do We Need to Switch a Packet?

- Destination Address
- MAC Header Rewrite String
- Outbound Interface Information
CEF Theory

CEF Builds Two Tables to Contain this Information:

- The CEF Mtrie
- The Adjacency Table
CEF Packet Switching

- Read in packet from the interface and store packet into memory
- Raise an interrupt to the processor; the rest of the packet switching takes place within the interrupt
- Use CEF mtrie to lookup packet destination; determine correct next-hop info by following pointer in the last CEF mtrie node
- Use Adjacency table info to rewrite physical layer header
- Place packet on the outbound interface queue
CEF Theory

What’s the Difference between a Tree and a Trie?

The MAC Header Rewrite Information Is Stored in the Tree Itself
CEF Theory

What’s the Difference between a Tree and a Trie?

A Pointer to the MAC Header Information Is Stored in the Trie, and the MAC Header Information Itself Is Stored in a Separate Table
The CEF Mtrie

172.16.1.0

root

256 Children

256 Children

256 Children

256 Children
The CEF Mtrie

- Nodes point to other nodes or leaves
The CEF Mtrie

- Leaves point to the adjacency table
The CEF MTree

Router#sh ip cef summary

IP CEF with switching (Table Version 4)
  4 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 0
  **4 leaves, 8 nodes**, 8832 bytes, 4 inserts, 0 invalidations
  0 load sharing elements, 0 bytes, 0 references
  universal per-destination load sharing algorithm, id 20340B24
  1 CEF resets, 0 revisions of existing leaves
  0 in-place/0 aborted modifications
  Resolution Timer: Exponential (currently 1s, peak 1s)
  refcounts: 533 leaf, 536 node
The CEF Mtrie

The Pipe

Main Processor

RIB

RX

Input Queue

Output Queue

TX

The Pipe
The CEF Mtrie Notes

• Where in the switching path do we build the CEF table?

• Nowhere! The CEF table is built from the routing table before (and while) packets are being switched

• Because the CEF table is directly related to the routing table, we can build it for every destination in the routing table without waiting on any packets to be switched
Two Separate Tables

The Routing Table and the CEF Mtrie Are Directly Related

The CEF Table Contains Reachability and Next Hop Information
The CEF Mtrie

Empty Table
The CEF Mtrie

**Add 10.0.0.0/8**
The CEF Mtrie

Add 20.1.0.0/16
The CEF Mtrie

Add 20.1.1.0/24
The CEF Mtrie

Add 30.1.1.0/29
The CEF Mtrie

Add 30.1.1.0/30
The CEF Mtrie

Add 30.1.0.0/16
The CEF Mtrie

Add 0.0.0.0/0
The CEF Mtrie

```
<table>
<thead>
<tr>
<th></th>
<th>0-9</th>
<th>10</th>
<th>11-19</th>
<th>20</th>
<th>21-29</th>
<th>30</th>
<th>31-255</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0.0.0/0</td>
<td>10.0.0.0/8</td>
<td>0.0.0.0/0</td>
<td>20</td>
<td>0.0.0.0/0</td>
<td>30</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td></td>
<td>0.0.0.0/0</td>
<td>1</td>
<td>2-255</td>
<td>0.0.0.0/0</td>
<td>0.0.0.0/0</td>
<td>1</td>
<td>2-255</td>
</tr>
<tr>
<td></td>
<td>0.0.0.0/0</td>
<td>0</td>
<td>1</td>
<td>2-255</td>
<td>0.0.0.0/0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>20.1.0.0/16</td>
<td>20.1.1.0/24</td>
<td>20.1.0.0/16</td>
<td>30.1.0.0/16</td>
<td>0.0.0.0/0</td>
<td>1</td>
<td>2-255</td>
</tr>
<tr>
<td></td>
<td>0-3</td>
<td>4-7</td>
<td>8-255</td>
<td>30.1.1.0/30</td>
<td>30.1.1.0/29</td>
<td>30.1.0.0/16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30.1.0.0/16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

© 2001, Cisco Systems, Inc. All rights reserved.
The CEF Mtrie

- Normally there are 4 levels of nodes with each node having 255 children
- Prefix and traffic distribution sometimes makes the mtrie perform better if there are different numbers of children for nodes at each level
The CEF MTRie

16-8-8

10-9-5-8

11-8-5-8
Path Through the CEF Switch Code

Interface Processor
DMA's packet into RX Ring Buffer

Software ‘Processes’....
Path Through the CEF Switch Code

1. A packet arrives at an input interface, RX Interrupt generated
2. Read IP Destination Prefix
3. Search CEF’s FIB DB, using the Destination Prefix as Search Key

Diagram:
- Destination Prefix
- FIB DB
- Search for FIB Entry
Path Through the CEF Switch Code

1. A packet arrives at an input interface, RX Interrupt generated

2. Read IP Destination Prefix

3. Search CEF’s FIB DB, using the Destination Prefix as Search Key

4. A Successful MTRIE Lookup will result in a FIB Entry being Found

4a. If the MTRIE Lookup is unsuccessful, the packet will be dropped

4a  Not Switched – Packet DROPPED
Path Through the CEF Switch Code

1. A packet arrives at an input interface, RX Interrupt generated
2. Read IP Destination Prefix
3. Search CEF’s FIB DB, using the Destination Prefix as Search Key
4. A Successful MTRIE Lookup will result in a FIB Entry being Found
   4a. If the MTRIE Lookup is unsuccessful, the packet will be dropped
5. FIB Path is selected
Path Through the CEF Switch Code

1. A packet arrives at an input interface, RX Interrupt generated
2. Read IP Destination Prefix
3. Search CEF’s FIB DB, using the Destination Prefix as Search Key
4. A Successful MTRIE Lookup will result in a FIB Entry being Found
   4a. If the MTRIE Lookup is unsuccessful, the packet will be dropped
5. FIB Path is selected
6. Selected FIB Path will point to necessary entry in Adjacency Table
Switch During the Receive Interrupt

- Features are processed along each switching path.
- Each feature represents a function call which may fail, succeed, or just not exist.
Switch During the Receive Interrupt

- At any point while the packet is being processed, it can be punted to the next slower process by allowing the processor to jump to the next pointer in the chain.
Switch During the Receive Interrupt

• At any point in the chain, the packet may be also be enqueued for process switching.
The CEF Mtrie

Depending on the Type of Route, a CEF Table Entry Can Be Several Different Types

- Attached
- Connected
- Receive
- Recursive
The CEF Mtrie

- **Attached**—An “attached” mtrie entry means the destination is attached to the router
- **Connected**—A “connected” entry is the result of an ip address being configured on an interface
- An entry may be both Attached and Connected
The CEF Mtrie

- Receive—Indicates packets that are destined to the router and do not need to be switched to another interface
- Recursive—References another node to find the next-hop information
The Adjacency Table

- The Mtrie is used to look up the next-hop for a prefix
- The final node encountered in the Mtrie during a prefix lookup includes a pointer to the correct next-hop in the adjacency table
The Adjacency Table

• The ARP Cache and the Adjacency Table are directly related.

• The adjacency table doesn’t contain any information about networks; it only contains information about next hops.

```
router#show arp
Address   Hardware Addr   Interface
10.1.1.1    3C3C.3C3C.3C3C  POS 4/1
```
The Adjacency Table

• Allows next-hops to change without changing the mtrie

• A change in next-hop just requires the final mtrie node’s pointer to the adjacency table to be updated

• Routing table changes also don’t impact the adjacency table
The Adjacency Table

- Update the FIB when changes in the routing table occur
- Update the adjacency table when changes in connected adjacencies occur
Adjacency Table Entries

- Auto adjacencies
- Punt Adjacencies
- Glean Adjacency
- Drop Adjacencies
- Discard Adjacencies
- Null Adjacencies
- Cached Adjacencies
Adjacency Table Entries (Auto)

• Auto Adjacencies—The most common type of adjacency; include all the information needed to rewrite the packet header and place the packet in the proper interfaces output queue
Adjacency Table Entries (Auto)

Router(config)#ip route 70.0.0.0 255.0.0.0 10.1.1.2
Adjacency Table Entries

- Punt Adjacencies—A punt adjacency indicates that the packet should be switched by the next slower switching scheme
Adjacency Table Entries (Glean)

- Glean Adjacency—Only one per router; indicates that the destination is attached to the router but the layer two information has not been acquired; results in an ARP request when a packet is switched to this destination
### Adjacency Table Entries (Glean)

**Router#sh ip interface brief**

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK?</th>
<th>Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0/0</td>
<td>20.0.0.1</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

**Router#sh ip cef adjacency glean**

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0.0.0/8</td>
<td>attached</td>
<td>Ethernet0/0</td>
</tr>
</tbody>
</table>
Adjacency Table Entries (Glean)

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.1.0/24</td>
<td>Attached</td>
</tr>
<tr>
<td>10.1.1.0/32</td>
<td>Receive</td>
</tr>
<tr>
<td>10.1.1.255/32</td>
<td>Receive</td>
</tr>
<tr>
<td>10.1.1.1/32</td>
<td>Receive</td>
</tr>
<tr>
<td>10.1.1.0/24</td>
<td>Attached</td>
</tr>
</tbody>
</table>

Diagram:
- 10.1.1.1
- 10.1.1.2
- 10.1.1.0/24
- ADJ
- Glean
Adjacency Table Entries (Glean)

- 10.1.1.1
- 10.1.1.0/24

- 10.1.1.2
- 10.1.1.0/32 Receive
- 10.1.1.255/32 Receive
- 10.1.1.1/32 Receive
- 10.1.1.0/24 Attached
- 10.1.1.2/32 Attached

ADJ
Glean
MAC
Adjacency Table Entries

• Drop Adjacency—Indicates the packet should be dropped
### Adjacency Table Entries (Drop)

**Router**
```
Router#sh ip cef adjacency drop
```

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>224.0.0.0/4</td>
<td>drop</td>
<td></td>
</tr>
</tbody>
</table>
Adjacency Table Entries

- **Discard Adjacency**—Indicates destinations which are part of a loopback’s subnet, but are not the actual ip address configured on the interface
Adjacency Table Entries (Discard)

Router(config)#int loop0
Router(config-if)#ip addr 40.0.0.1 255.255.255.0

```
40.0.0.1/32
```

```
40.0.0.0/24
```

```
Router#sh ip cef 40.0.0.2
40.0.0.0/24, version 3, attached, connected
0 packets, 0 bytes
via Loopback0, 0 dependencies
 valid discard adjacency
```
Adjacency Table Entries

- Null Adjacency—Indicates the packet should be switched to a Null interface on the router
Adjacency Table Entries (Null)

Router(config)#ip route 60.0.0.0 255.0.0.0 null0
Router#sh ip cef adjacency null

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>60.0.0.0/8</td>
<td>attached</td>
<td>Null0</td>
</tr>
</tbody>
</table>
CEF Show Commands

```
router#show ip cef

Prefix              Next Hop             Interface
0.0.0.0/32          receive
10.97.1.0/24        attached             Serial4/3
10.97.1.0/32        receive
10.97.1.255/32      receive
42.40.183.0/24      12.51.142.5          POS1/0
```

Prefix: The Prefix of the Destination Network

Next Hop: The Type of Adjacency or the Next Hop Towards This Destination

Interface: The Interface Out Which to Send Traffic for This Destination
router#show ip cef 33.97.1.0 255.255.255.0 detail
33.97.1.0/24, version 13, attached, connected, cached adjacency to Serial4/3

0 packets, 0 bytes
via Serial4/3, 0 dependencies
valid cached adjacency

The Type of Adjacency This CEF Table Entry Points to

Number of Table Entries Which Point to (Depend On) This Entry

Number of Packets and Bytes Which Have Been Switched Through This Entry; Configure IP CEF Accounting Per-prefix for This to Work
router#show ip cef summary
IP CEF with switching (Table Version 46), flags=0x0
22 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 0
25 leaves, 19 nodes, 22960 bytes, 49 inserts, 24 invalidations
0 load sharing elements, 0 bytes, 0 references
universal per-destination load sharing algorithm, id F2F8D257

Total Number of Entries in the CEF Table
Number of Entries Which Need to Be Re-resolved
Number of Entries Which Do Not Have Resolved Recursions
### CEF Show Commands

```plaintext
router#show adjacency detail

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Interface</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Serial4/0</td>
<td>point2point(5)</td>
</tr>
</tbody>
</table>

| A          | 0 packets, 0 bytes                  |
| B          | 0F000800                                |
| C          | CEF expires: 00:02:32, refresh: 00:00:32 |

A: Packets and Bytes Switched Through This Adjacency
B: MAC Header Rewrite String
C: When This Entry Will Be Refreshed; In This Case, All Point2Points Are Refreshed Every Minute
**CEF Show Commands**

```
router#show int ethernet1/0 stat

Ethernet1/0

<table>
<thead>
<tr>
<th>Switching path</th>
<th>Pkts In</th>
<th>Chars In</th>
<th>Pkts Out</th>
<th>Chars Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>977121</td>
<td>70149655</td>
<td>578014</td>
<td>56457133</td>
</tr>
<tr>
<td>Route cache</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>977121</td>
<td>70149655</td>
<td>578014</td>
<td>56457133</td>
</tr>
</tbody>
</table>

Route cache Includes CEF Switched Packets
```
Router Architectures & Parallel Express Forwarding
Introduction

- Routers have to deal in three “Planes” of operation:-

  The “Control” Plane
  
  Building and maintaining data structures such as “forwarding tables”

  The “Management” Plane
  
  Dealing with configuration files, gathering and providing statistics, providing and responding to control protocol messages

  The “Data” Plane
  
  Switching of packets, manipulation of packet (header and content), packet delivery scheduling (queuing)
When any all or all of the resources are exhausted, inconsistent behavior will be observed.
Routers Operationally

- Maintain/manipulate routing information
  - Listen for updates/update neighbors
- Classify packets for manipulation/queuing/permit-deny, etc.
  - Compare packets to classification lists and perform control
- Perform Layer 3 switching
  - Create outbound Layer 2 encapsulation
  - Layer 3 checksum
  - TTL/hop count update
- Management/billing (statistics)
  - Interface statistics—NetFlow export
  - Telnet, SNMP, ping, trace route, HTTP
Routers Functionally

- **(Attempt to) switch packets**
  - Layer 3 switching based on routing information

- **(Attempt to) transmit packets**
  - Access outbound media

- **Manipulate packets**
  - Change contents of packet (CAR/NAT/compression/encryption)

- **Consume packets**
  - Routing protocol updates etc…/services advertisements(SAP)/ICMP/SNMP

- **Generate packets**
  - Routing protocol packets/SAPs/ICMP/SNMP
  - Tunnels—GRE, IPSec, DLSw etc…
Router Hardware

- Interface Processors
- The Central Processing Unit
- Memory
- The Backplane
The Central Processing Unit

- Provides horsepower for all control plane functions, such as system maintenance, building routing tables, etc.
- On some platforms, it also provides the horsepower for actually switching packets
Shared Memory Architecture

- **Applicable Platforms**
  
  - Cisco 1xxx
  - Cisco 2xxx
  - Cisco 3xxx
  - Cisco 4xxx
Shared Memory Architecture

- Memory
- CPU Allocated
- Memory
- Packet Switching
- CPU Allocated
- Memory
- Buffers
- Queues
- Pointers
- Headers
- Switching Tables
- S/W Image/Files
- CPU Buffers
- CPU Queues
- General Purpose CPU
- Data/Address/Control Bus’s
- Interface
- Interface
- Interface
- Interface
- Interface
- Interface
- Interface
- Interface
- Physical Media Interfaces (Fixed or Modular)
Shared Memory Architecture (Hardware “Assist”)

- Applicable Platforms
  - Cisco 7200
  - Cisco 7300
  - Cisco 7400
  - Cisco 10000
Shared Memory Architecture (Hardware “Assist”)

- General Purpose CPU
- Function Specific Hardware
- Memory
- Packet Switching
- Interface
- Physical Media Interfaces (Fixed or Modular)
- Buffers
- Queues
- Pointers
- Headers
- CPU Allocated Memory
- Switching Tables
- Image/Files
- CPU Buffers
- CPU Queues

Data/Address/Control Bus’s
Distributed Shared Memory

- **Applicable Platforms**

  - Cisco 7500
  - Catalyst 5xxx RSM
Some Line cards have Packet Memory, Forwarding Table Memory and a discrete Switching Hardware.
Distributed Cross Bar

- Applicable Platforms
  - Cisco 6500/7600 OSR
  - Cisco 12000 (GSR)
Cross Bar Data Path

Serial Input/Output Lines, “To” and “From” Fabric

ASIC X-Bar Fabric
Cross Bar Data Path (Multiple Fabrics)

Packet Slicing Allows Multiple Switching Fabrics

Packets split into cells

Serialization time = t/4

Packets reassembled

Serialization time = t

CPU

Interface Card

Packet Memory

(D) FT

64

256 bytes

XOR

64

64

64

64

256 bytes
Parallel Express Forwarding

- PXF is one kind of “Function Specific Hardware”
- PXF Architecture
- PXF packet switching
Cisco-Developed Unique Value-Add
PXF IP Services Processor

- Internally-Developed Cisco Processing Technology

US Patent 6,101,599
Benefit of PXF Acceleration

Number of Services Added to the Network

Performance

PXF Hardware Acceleration

Non-PXF Processing

Delta in CPU Usage
Parallel eXpress Forwarding (PXF) Engine

- New technology switching engine for high-touch L3 services with optimized throughput
- Programmable architecture to allow for future feature upgrades
- Based on custom pipelined array processor (ASIC)
Power of Cisco Parallel Processing

- Matrix of separate processors
- Implements “assembly line” for exceptional performance
- “Assembly line” enables consistent throughout
- Little division when services are enabled/disabled
PXF Processor Services

Example

Classification

NAT

CEF

Packet U → IM → Packet Q → IM → Packet M → IM → Packet I → IM → Packet E → IM → Packet A

CBWFQ, WRED, LLQ
Parallel eXpress Forwarding (PXF)

- Each PXF ASIC has 16 processors arranged in 4 rows x 4 columns
- Two PXF ASICs connected serially: 4x8, 32 CPUs total for an ESR
- Parallelism and pipelining => Improved feature throughput
PXF Packet Forwarding

Headers Pass through Toaster

SDRAM

IN

PXF

SDRAM

OUT

Modified Headers and Bodies Are Moved to Packet Buffer Memory

Within the SDRAM:

From-Toaster Complex

Input Packet Memory

Interface

From Line Cards

To-Toaster Complex

Output Queue Controller

Interface

To Line Cards

Complete Packets Are Moved from SDRAM to Output Line Cards
Summary
Summary

• 90 minutes is way too much to summarize in one slide, and not enough time to cover these topics!

• Routers scale based on CPU, processing hardware, memory and bandwidth

• Resource exhaustion results in dropped packets

• No one architecture has all the answers different platforms are appropriate for different roles in your network
Recommended Reading

Inside Cisco IOS
Software Architecture
(CCIE Professional Development)

IP Routing Fundamentals

Cisco Router Configuration,
Second Edition
ISBN: 1-57870-241-0

CEF Whitepaper:
Life Is Short, Eat Dessert First!
Cisco Systems

Empowering the Internet Generation