



# Inter VLAN Routing



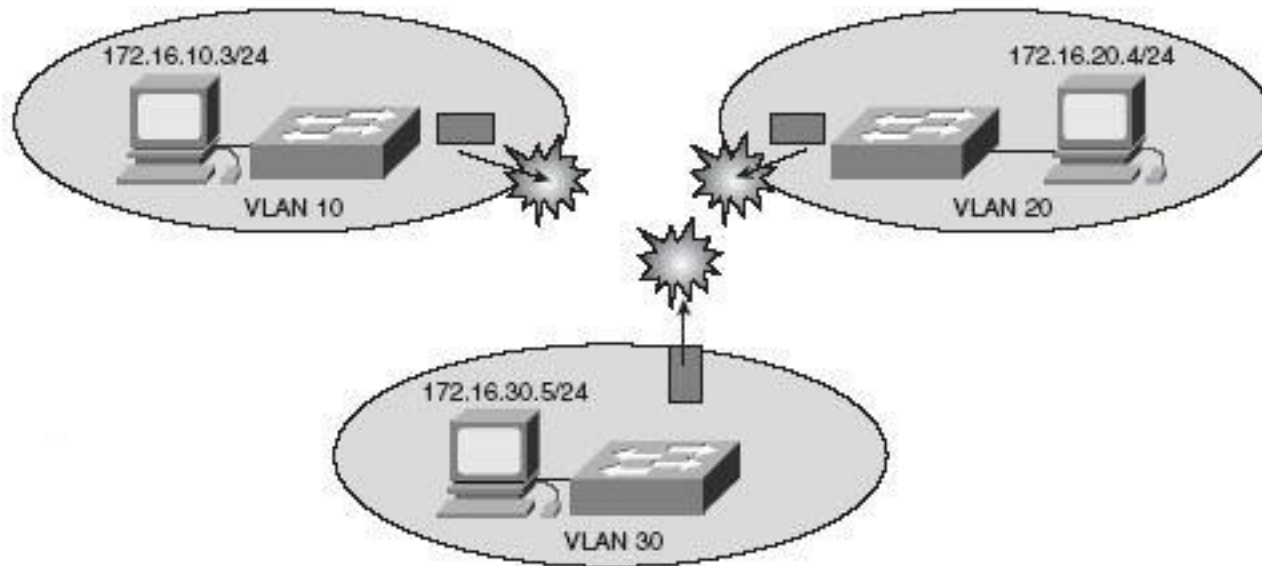
SWITCH Module 2 and 5

# Agenda

- **Intro**
- **Router-on-a-Stick**
- **MLS Routing**
  - Routed ports
  - SVIs
- **DHCP**
- **Multilayer Switch Processing**
  - Packet Switching
  - Fast Switching
  - Cisco Express Forwarding
- **Switch Database Management**
- **DHCPv6**

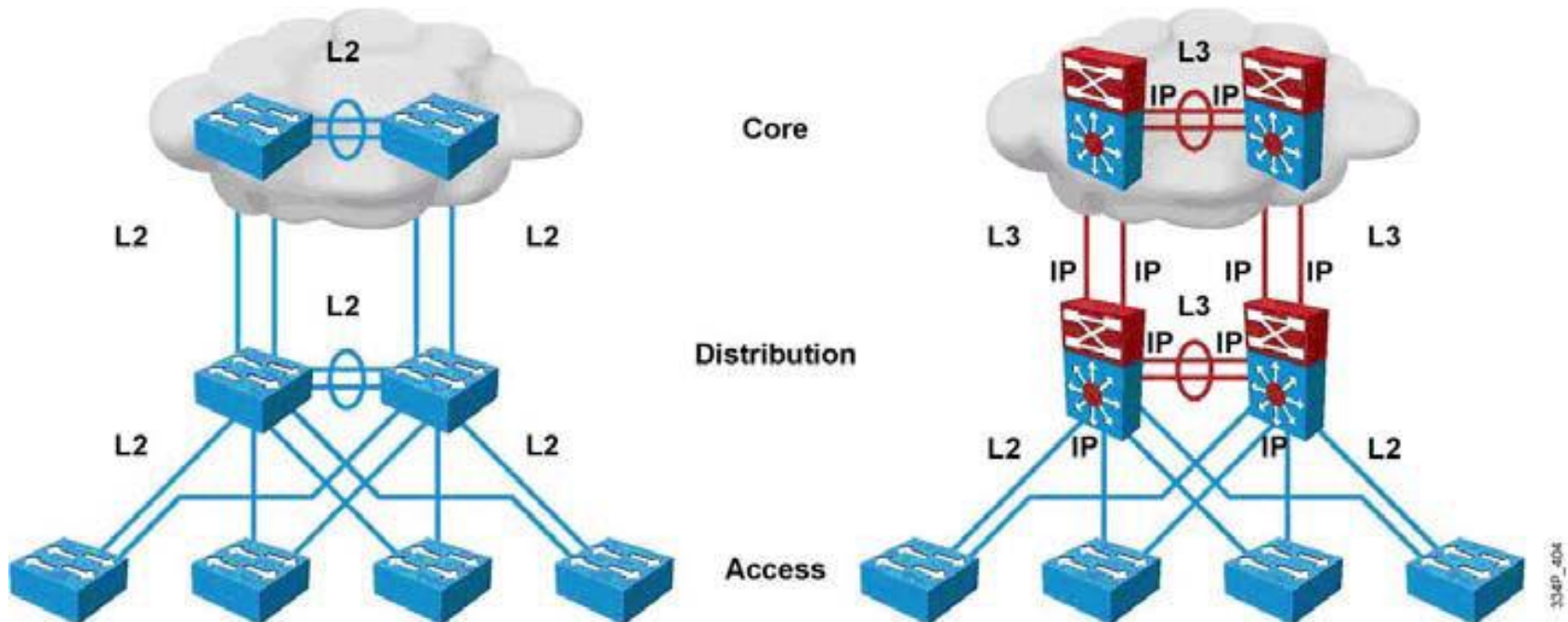
# Introduction to Inter-VLAN Routing

- VLANs isolate traffic by design
- Inter-VLAN router of some sort required
- Inter-VLAN routing should occur in the distribution layer
- Multilayer switch is recommended to terminate VLANs



# Routed vs. Switched Design

- Routing can now be performed at L2 switching speeds by switching frames/packets using specialized ASICs
- L3 switches serve as default gateways, terminating VLANs
- Routed ports or L2/L3 EC allow to isolate VLAN at the distribution layer
  - L3 connection between Distribution and Core layer allows to decrease L2 domains, use multipath and/or fast convergence

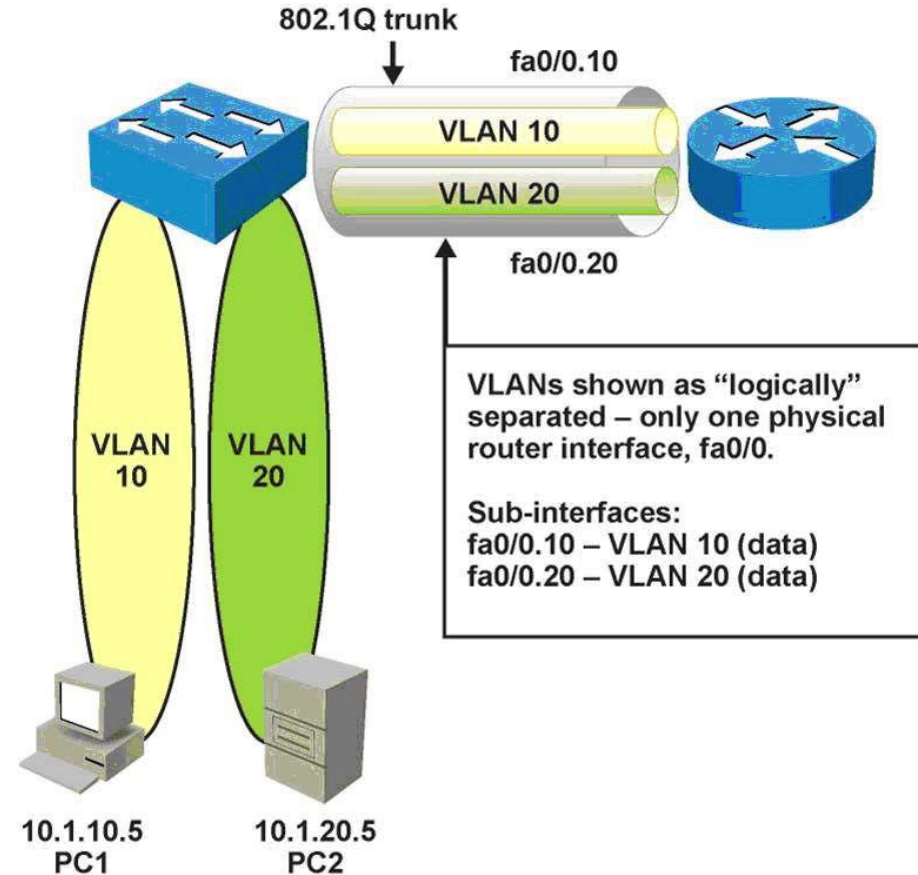


# Inter VLAN Routing Using External Router



# Router-on-a-Stick

- Router interface, typically Fast Ethernet, subdivided into logical subinterfaces, one per VLAN.
- L2 switch is connected to the router via trunk port
- [“Configuring InterVLAN Routing and ISL/802.1Q Trunking on a Catalyst 2900XL/3500XL/2950 Switch Using an External Router”](#)



# Configuring Router-on-a-Stick

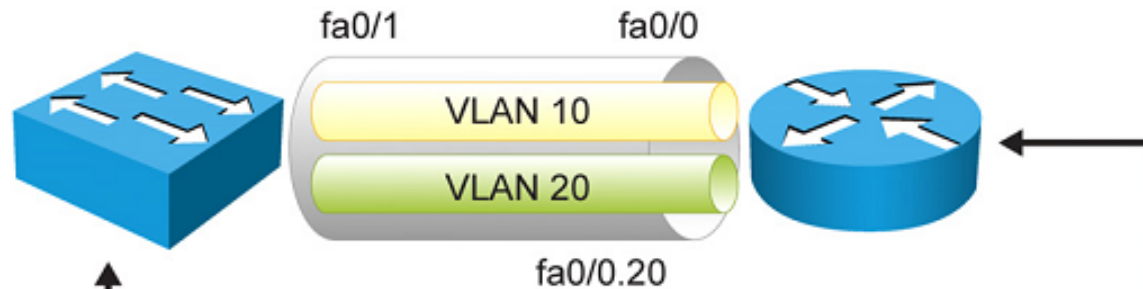
- 1) Enable trunking on a switch interface
- 2) Create sub-interfaces for each VLAN that requires inter-VLAN routing

```
Router(config)# interface type slot/port.subiface
```

- 3) Configure the trunking encapsulation and IP address on sub-interface; optionally mark native VLAN

```
Router(config-subif) #  
encapsulation [ dot1q | isl ] vlan-id {native}
```

# Example



```
switch(config)# vlan 10,20
switch(config)# interface fastethernet 0/1
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 99
switch(config-if)# switchport trunk allowed vlan 10,20
```

```
router(config)# interface fastethernet 0/0.10
router(config-if)# encapsulation dot1q 10
router(config-if)# ip address 10.1.10.1 255.255.255.0
router(config)# interface fastethernet 0/0.20
router(config-if)# encapsulation dot1q 20
router(config-if)# ip address 10.1.20.1 255.255.255.0
```



# External Router Scenario

## ■ *Advantages*

- Works with any switch, since Layer 3 services are not required on the switch
- Implementation is simple
- The router provides communication between VLANs

## ■ *Disadvantages*

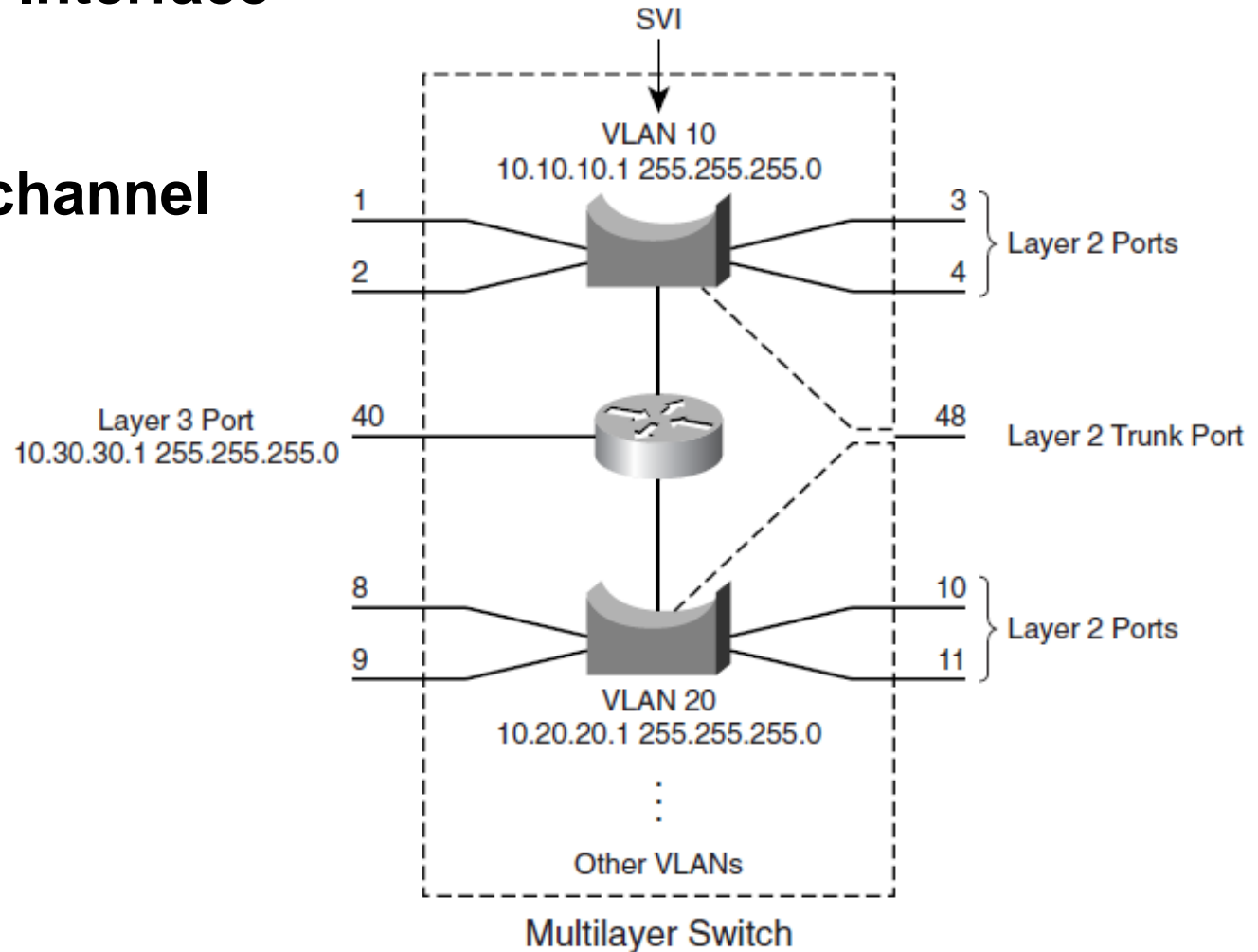
- The router is a single point of failure
- Single traffic path may become congested
- Latency may be introduced as frames leave and reenter the switch chassis multiple times, and the router makes software-based routing decisions

# MLS Routing



# MLS Routing Interfaces

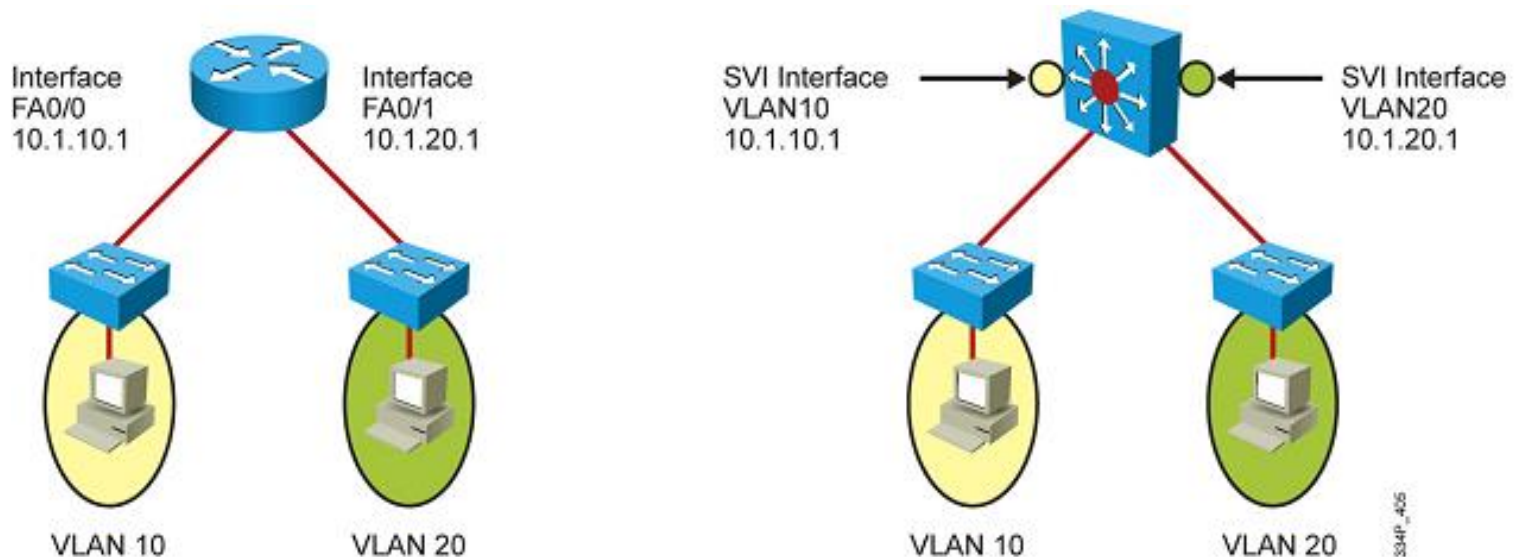
- Switch Virtual Interface
- Routed port
- Routed Etherchannel



# Switch Virtual Interface

## ■ Switch Virtual Interface (SVI)

- A virtual interface for internal connection of route processor and VLAN
- VLAN 1 SVI is created by default
- Every VLAN CAN have separate SVI
- MLS use SVI for routing between VLANs



# SVI Configuration Guide

## 1) Enable IP routing

```
Switch(config)# ip routing
```

## 2) Configure SVI for VLAN

```
Switch(config)# interface Vlan vlan-id
```

## 3) Assign IP address

```
Switch(config-if)# ip address address mask
```

## 4) Enable SVI

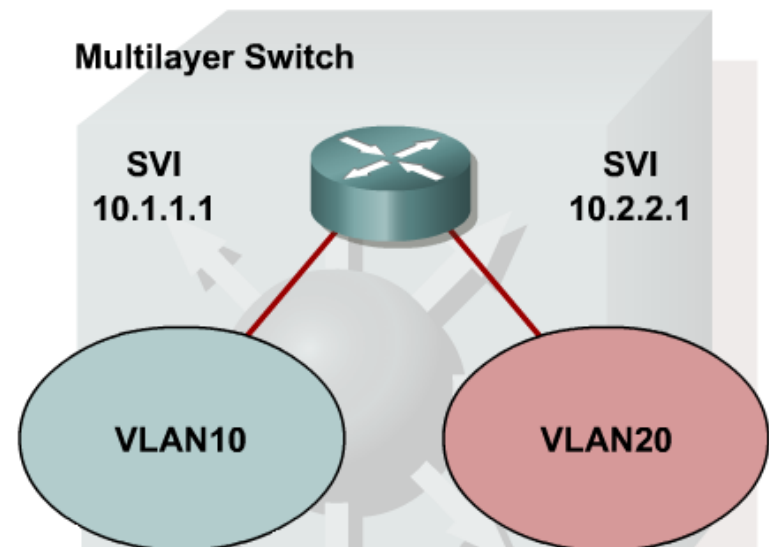
```
Switch(config-if)# no shutdown
```

## 5) Configure routing protocol

# SVI Example

```
Switch(config)# ip routing
Switch(config)# vlan 10,20
Switch(config-vlan)# exit
Switch(config)# int vlan10
Switch(config-if)# ip address 10.1.1.1 255.255.255.0
Switch(config-if)# int vlan20
Switch(config-if)# ip address 10.2.2.1 255.255.255.0
```

- **ip routing** enables L3 switching
- This command enables all L3 functions and commands
  - routing table
  - routing protocols



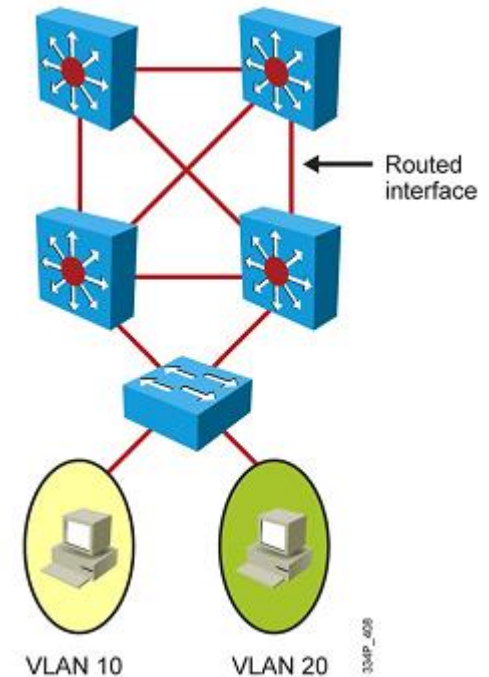
# SVI Interface

- SVI is not always active interface as loopback
- SVI is in “up/up” state `IF`
  1. VLAN associated with SVI is created and active
  2. The VLAN interface is not administratively down
  3. VLAN has at least one active port
    - a) An access port in VLAN
    - b) A trunk port where VLAN is enabled
  - Necessary for proper function of routing protocols/table
- `IF` it is desired to not include access/trunk VLAN presence in the SVI line-state calculation `THEN` configure:

```
Switch(config-if) # switchport autostate exclude
```

# Routed Port ①

- Physical port similar to Ethernet on a router
  - DOES NOT belong to any VLAN
  - DOES NOT support trunking
  - DOES NOT support subinterfaces
  - Requires removal of L2 functionality
- Routed port configuration:



```
Switch(config)# ip routing
Switch(config)# int fa0/5
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.0.3.1 255.255.255.0
```

*enable/disable L2 function*



## Routed Port ②

- A routed port is internally represented with an internal VLAN
  - Internal Usage VLAN

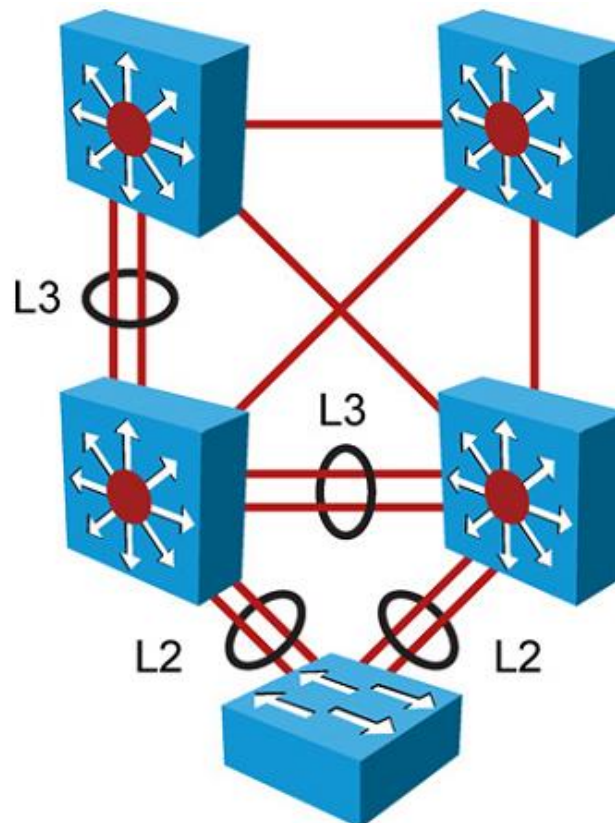
```
Switch# show vlan internal usage
```

- Internal VLANs are usually allocated starting from 1006
  - Some platforms can enable descending allocation from 4094
  - Recommendation:
    - Allocate extended VLAN from 4094 to avoid conflict with internal VLAN

```
Switch(conf-t) #  
vlan internal allocation policy ascending descending
```

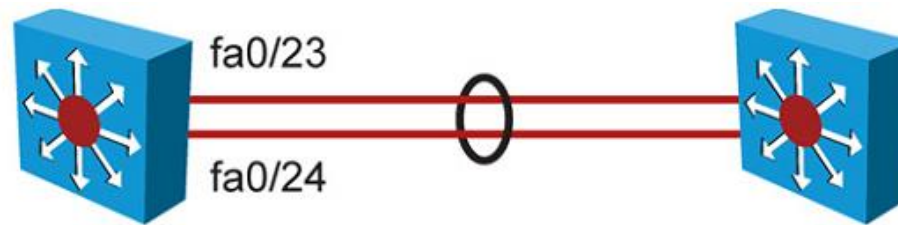
# Routed Etherchannel

- Layer 2 EtherChannel bundles access or trunk ports between switches or other devices (e.g., servers)
- Layer 3 EtherChannel bundles routed ports between switches



# Routed Etherchannel Example

- The **no switchport** command is applied both on the physical ports and on the EtherChannel interface



```
switch(config)# interface fastethernet 0/23
switch(config-if)# no switchport
switch(config-if)# channel-group 1 mode on
switch(config)# interface fastethernet 0/24
switch(config-if)# no switchport
switch(config-if)# channel-group 1 mode on
switch(config)# interface port-channel 1
switch(config-if)# no switchport
switch(config-if)# ip address 10.1.20.1 255.255.255.0
```

# Revert Back to L2 Switching

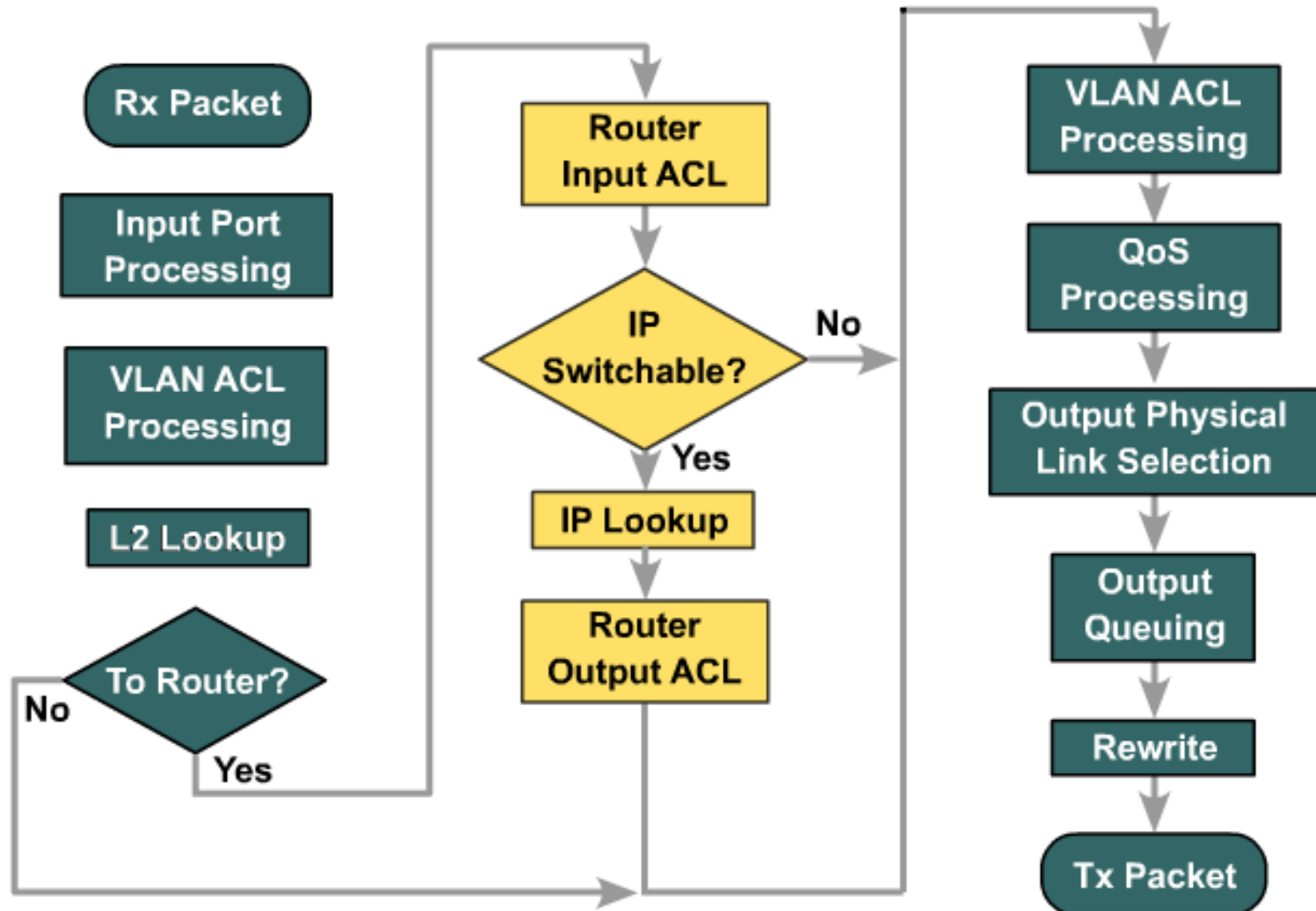
```
Switch(config)# int fa0/5  
Switch(config-if)# no ip address  
Switch(config-if)# switchport
```

# The `ip default-gateway` Command

- On L2 switches or L3 switches without **ip routing** enabled it is wise to configure default-gateway
- Without previous command, devices with IP addresses will not be able to cross borders of their subnet
- Hence, L2 switches are configured with following command:

```
Switch(config) # ip default-gateway ADDRESS
```

# Packet Processing



# Common Inter-VLAN Routing Problems

Problem	Possible Cause
<b>Missing VLAN</b>	VLAN might not be defined across all the switches. VLAN might not be enabled on the trunk ports. Ports might not be in the right VLANs.
<b>Layer 3 interface configuration</b>	Virtual interface might have the wrong IP address or subnet mask. Virtual interface might not be up. Virtual interface number might not be match with the VLAN number. Routing has to be enabled to route frames between VLAN. Routing might not be enabled.
<b>Routing protocol misconfig</b>	Every interface or network needs to be added in the routing protocol. The new interface might not be added to the routing protocol. Routing protocol configuration is needed only if VLAN subnets needs to communicate to the other routers, as previously mentioned in this chapter.
<b>Host misconfig</b>	Host might not have the right IP or subnetmask. Each host has to have the default gateway that is the SVI or Layer 3 interface to communicate the other networks and VLAN. Host might not be configured with the default gateway.

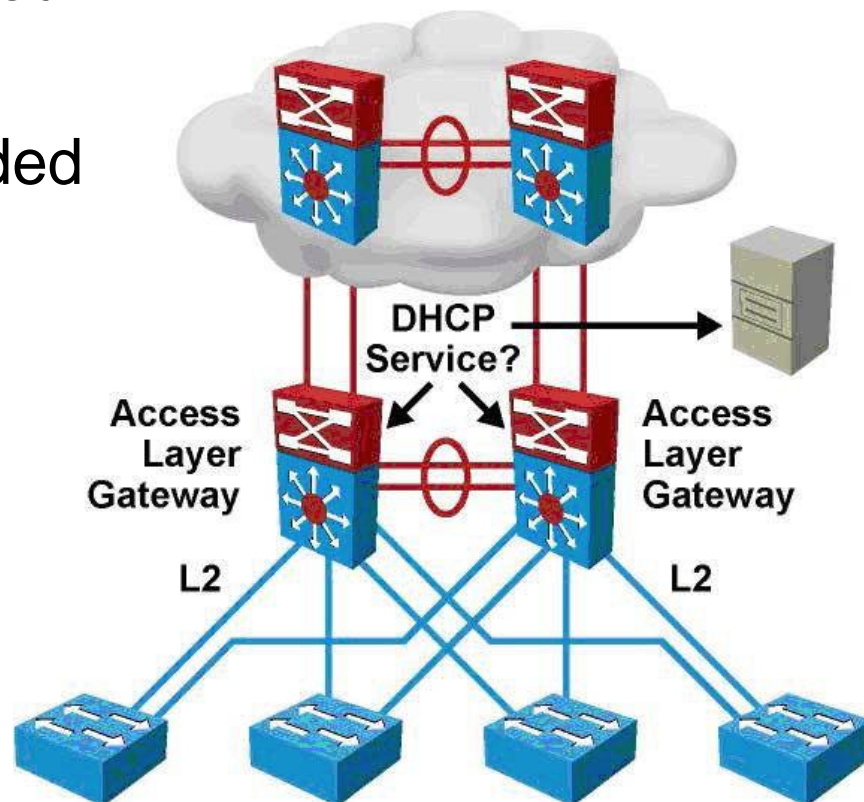
# Dynamic Host Control Protocol



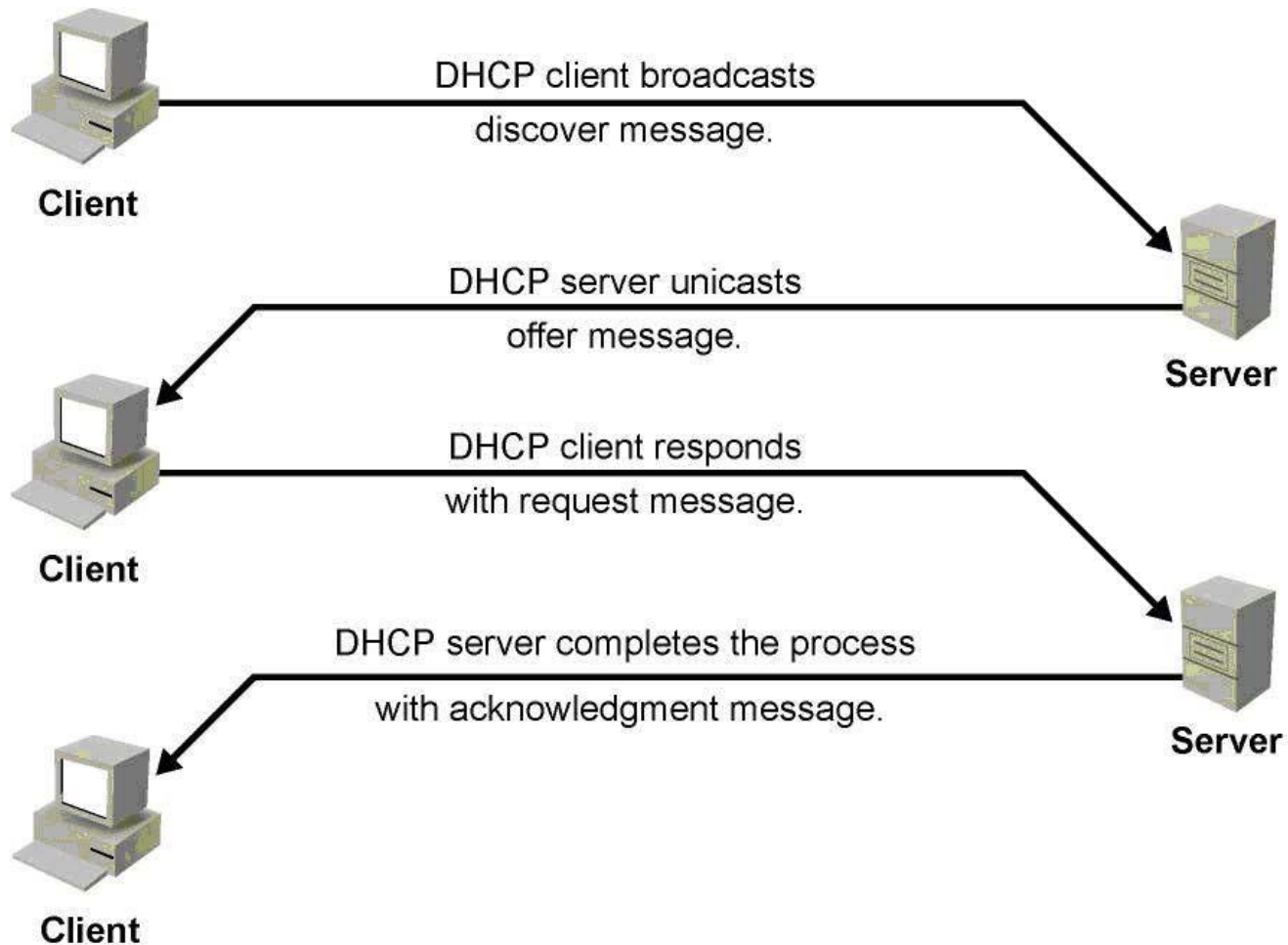


# Overview

- Clients in access VLANs need DHCP service
- DHCP service can be provided by the distribution switches, acting as gateways, or external DHCP server elsewhere in the network

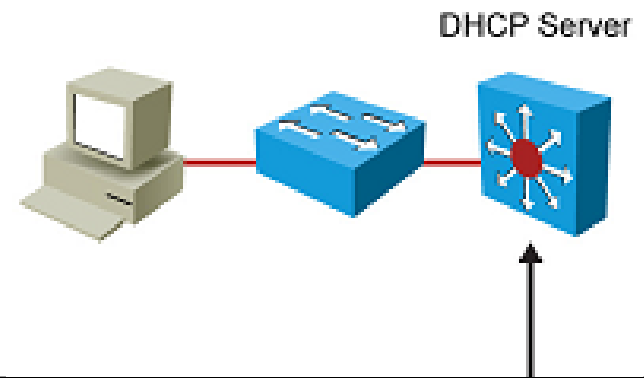


# DHCP Operation



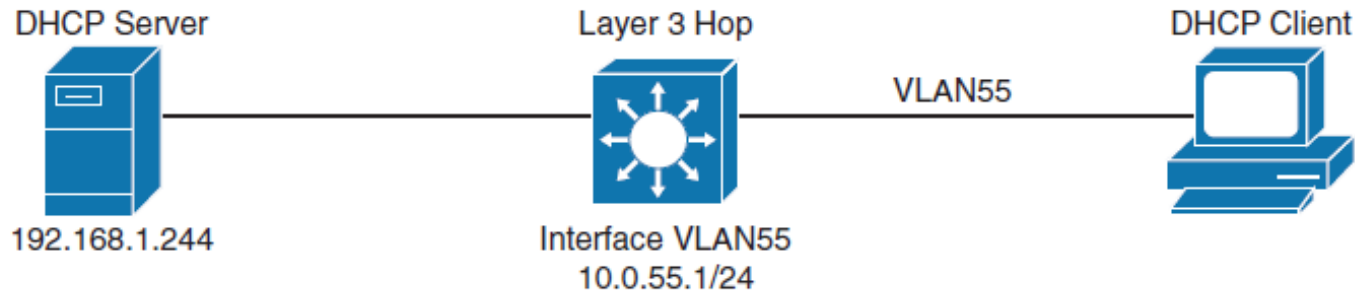
# Configuration and Example

- Configure DHCP pool with network, mask, and other parameters
- Configure excluded addresses
- Pool is selected when DHCP request is received from matching subnet



```
Switch(config)# ip dhcp excluded-address 10.1.10.1 10.1.10.20
Switch(config)# ip dhcp pool XYZ10
Switch(config-dhcp)# network 10.1.10.0 255.255.255.0
Switch(config-dhcp)# default-router 10.1.10.1
Switch(config-dhcp)# option 150 10.1.1.50
! Lease for 0 days 8 hours 0 minutes
Switch(config-dhcp)# lease 0 8 0
Switch(config)# interface vlan10
Switch(config-if)# ip address 10.1.10.1 255.255.255.0
```

# DHCP Relay Motivation

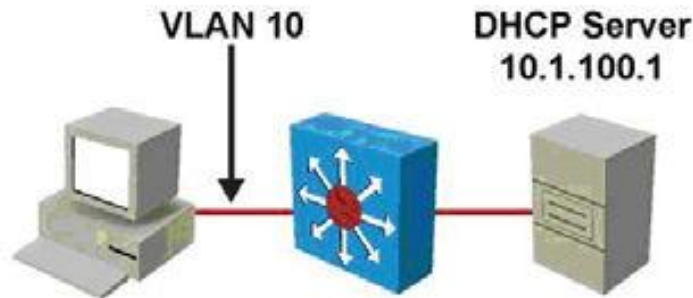


A client that resides in VLAN 55 needs to have the following two configurations to forward the DHCP broadcast to the centralized server 192.168.1.244:

- The multilayer switch must have a Layer 3 IP address that will receive the client DHCP request that is 10.0.55.1/24. This address may be a routed port or an SVI.
- The **ip helper-address** command must be configured on the multilayer switch Layer 3 interface. With the DHCP relay address, when the switch receives a DHCP request in the form of a broadcast message from a client, the switch forwards this request, as a unicast message, to the IP address that is specified in the **ip helper-address** command.
- The **ip helper-address** command not only forwards DHCP UDP packets but also forwards other UDP broadcasts such as TFTP, DNS, time, NetBIOS, name server, and BOOTP packets by default.

# DHCP Relay

- Use the **ip helper-address** command on the interface which connects to the subnet containing devices which request IP addresses from the DHCP server
- On a multilayer switch, the interface “connecting” to the relevant subnet is typically an SVI



```
Switch(config)# interface vlan10
Switch(config-if)# ip address 10.1.10.1 255.255.255.0
Switch(config-if)# ip helper-address 10.1.100.1
```

# Configuring DHCP Options

- Use DHCP options to “expand” the basic DHCP commands.
- The following are some of the commonly used options.
  - **Option 43:** Vendor-encapsulated option that enables vendors to have their own list of options on the server. For example, you can use it to tell a lightweight access point where the Wireless LAN Controller (WLC) is.
  - **Option 69:** SMTP server, if you want to specify available SMTP servers to the client.
  - **Option 70:** POP3 server, if you want to specify available POP3 servers to the client.
  - **Option 150:** TFTP server that enables your phones to access a list of TFTP servers.

```
Switch(config)# ip dhcp pool TELEPROFILES  
Switch(dhcp-config)# options 150.10.10.1
```

# Verifying and Troubleshooting DHCP

```
Switch# show ip dhcp binding
```

```
Bindings from all pools not associated with VRF:
```

IP address	Client-ID/ Hardware address/ User name	Lease expiration	Type
10.1.10.21	0100.1bd5.132a.d2	Jun 25 2009 06:09 AM	Automatic
10.1.10.22	0100.4096.a46a.90	Jun 25 2009 09:40 AM	Automatic
10.1.10.23	0100.4096.aa98.95	Jun 25 2009 11:28 AM	Automatic

```
Switch# debug ip dhcp server packet
```

```
DHCPD: DHCPDISCOVER received from client 0100.1bd5.132a.d2 on interface Vlan6.
```

```
DHCPD: Sending DHCP OFFER to client 0100.1bd5.132a.d2 (10.1.10.21).
```

```
DHCPD: broadcasting BOOTREPLY to client 001b.d513.2ad2.
```

```
DHCPD: DHCPREQUEST received from client 0100.1bd5.132a.d2.
```

```
DHCPD: Sending DHCPACK to client 0100.1bd5.132a.d2 (10.1.10.21).
```

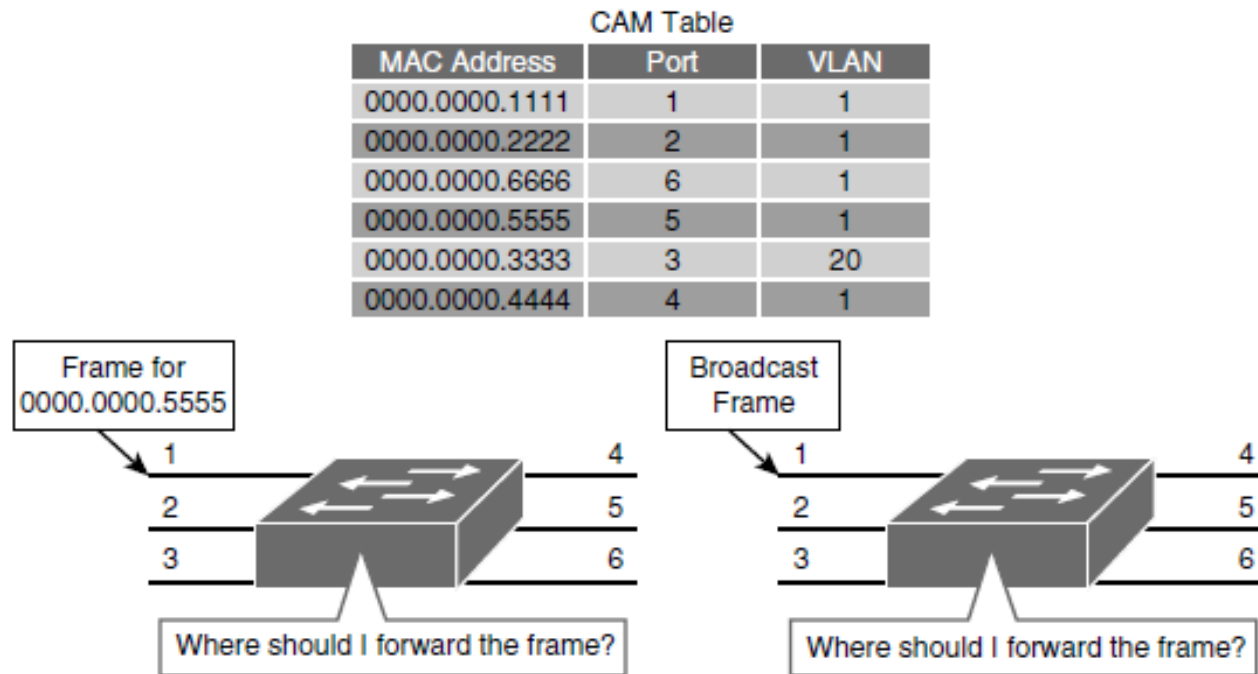
```
DHCPD: broadcasting BOOTREPLY to client 001b.d513.2ad2.
```

# MLS Processing





# MAC Address Forwarding



- *Where should the frame be forwarded?*
- *Are there restrictions preventing the forwarding of the frame?*
- *Is there any prioritization or marking that needs to be applied to the frame?*

# Layer 2 Switch Operation

- **Layer 2 forwarding table**

- The Layer 2 forwarding table, also called the *MAC table*, contains information about where to forward the frame. Specifically, it contains MAC addresses and destination ports. The switches reference the destination MAC address of the incoming frame in the MAC table and forward the frames to the destination ports specified in the table. If the MAC address is not found, the frame is flooded through all ports in the same VLAN.

- **ACLs**

- Access control lists (ACLs) do not only apply to routers. Switches can also apply ACLs based on MAC and IP addresses. Generally only higher-end switches support ACLs based on both MAC and IP addresses, whereas Layer 2 switches support ACLs only with MAC addresses.

- **QoS**

- Incoming frames can be classified according to QoS parameters. Traffic can then be marked, prioritized, or rate-limited.

# Frame Rewrite

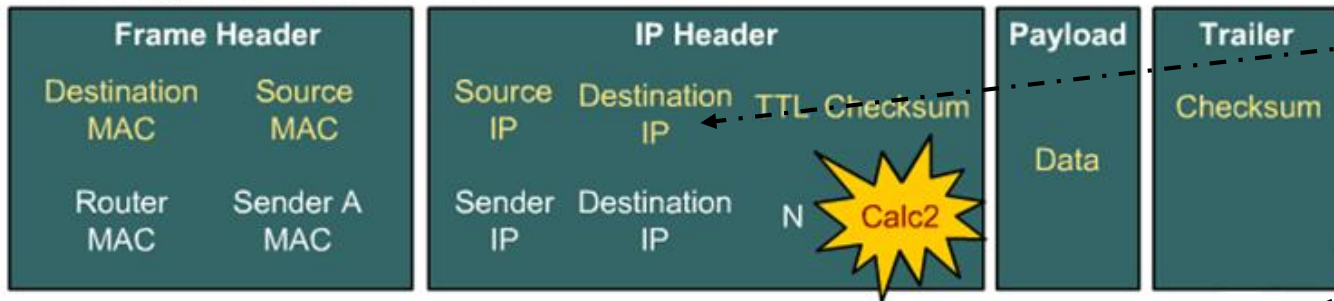
## Incoming frame



1. FCS check of incoming frame.

2. Payload extraction, IP checksum check

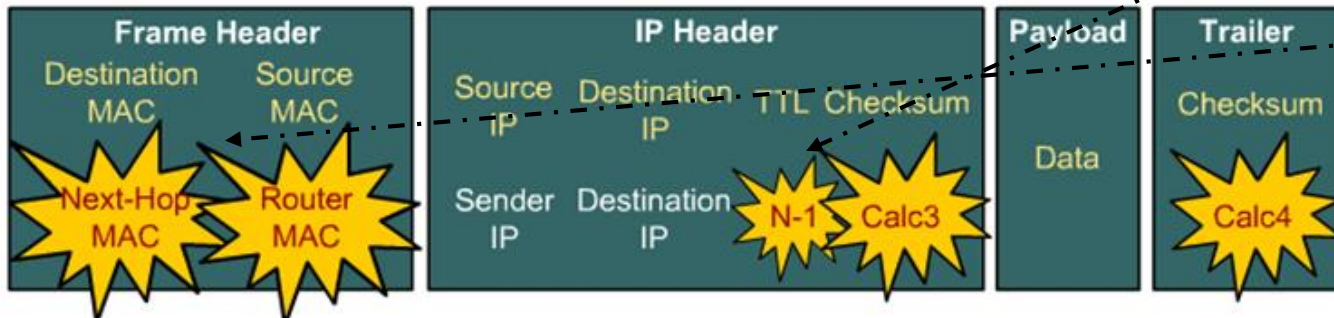
## Incoming IP Unicast Packet



3. Routing decision.

4. TTL decrementing, computing new checksum

## Rewritten IP Unicast Packet



5. Rewriting frame header

6. Computing FCS and sending the frame

# Unicast routing in IPv4 Networks ①

- 1) The frame has the right size and correct FCS
  - IF not THEN the frame is discarded
- 2) IP header checksum is correct
  - IF not THEN the packet is discarded
- 3) Is the packet designated to the local IP address?
  - IF yes THEN packet is not routed
- 4) TTL value is greater than 1
  - IF not THEN packet is discarded and ICMP message is sent

# Unicast routing in IPv4 Networks ②

- 5) The routing table is searched for an appropriate entry to the destination address
  - The routing table is **arranged in descending** order according to network masks
  - The routing table is searched for a row where hold:  
Destination IP address & network mask = network
  - IF there is no route to the destination THEN packet is discarded and ICMP message is sent as reply
- 6) The row in the routing table points to the outgoing interface
  - IF not THEN the row points to the address of next hop router, hence remember the address and go to the step ⑤
- 7) L2 information are searched for the last remembered IP address
  - ARP table, Frame Relay map ...

# Unicast routing in IPv4 Networks ③

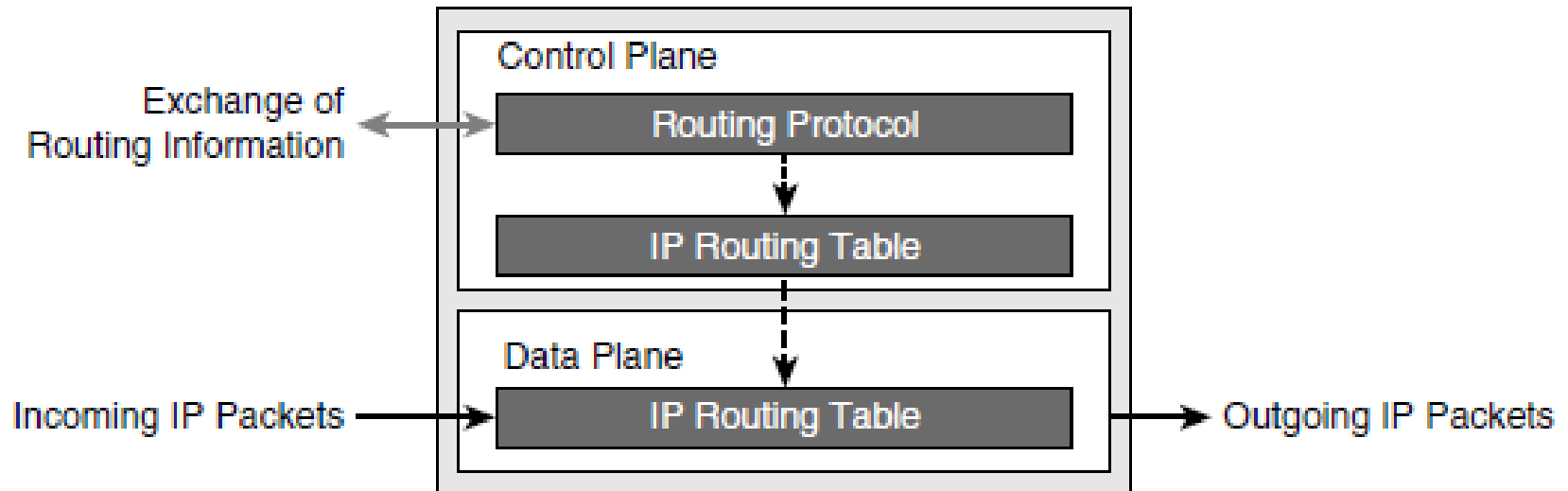
- 8) TTL field in the IP header is decremented and the checksum is recalculated
  - 9) IP packet is encapsulated to the frame (using information from the step ⑦) and sent to the interface (found in the step ⑥)
- Every packet is processed according to these steps, this is known as **Process switching** (slow way how to route)
  - The processing speed of packets depends on the router architecture and CPU load
  - Bigger traffic can cause huge processor load

# Efficiency of Routing

- Previously described algorithm DOES NOT take into account specific situations:
  - Fragmentation
  - ACL
  - Tunneling, ciphering/deciphering
  - NAT ...
- The most time consuming are steps ⑤, ⑥ and ⑦
  - Router performs lookups for every packet in several tables to make routing decisions
  - Checksum can be implemented using ASIC
- Research efforts focus on accelerating the routing lookup/decision
  - Process switching, Fast switching, Autonomous switching, Silicon switching engine (SSE) switching, Optimum switching, Distributed fast switching, Cisco Express Forwarding (CEF), Distributed Cisco Express Forwarding (dCEF)

# Distributed Hardware Forwarding

- Network devices contain at least three planes of operation:
  - **Management plane**
  - **Control plane**
  - **Forwarding plane**





# Distributed Hardware Forwarding

- The **management plane** is responsible for the network management, such as SSH access and SNMP, and may operate over an out-of-band (OOB) port.
- The **control plane** is responsible for protocols and routing decisions, and the **forwarding plane** is responsible for the actual routing (or switching) of most packets.
- Multilayer switches must achieve high performance at line rate across a large number of ports. To do so, multilayer switches deploy independent control and forwarding planes.
- The control plane will program the forwarding plane on how to route packets.
- Multilayer switches may also employ multiple forwarding planes. For example, a Catalyst 6800 uses forwarding planes on each line module, with a central control plane on the supervisor module.
- To continue the example of the Catalyst 6800, each line module includes a microcoded processor that handles all packet forwarding.
- For the control plane on the supervisor to communicate with the line module, a control layer communication protocol exists.

# Cisco Switching Methods

A Cisco IOS-based routers uses one of three methods to forward packets:

- **Process Switching**

- Process switching is the slowest form of routing because the processor must route and rewrite using software.

- **Fast Switching**

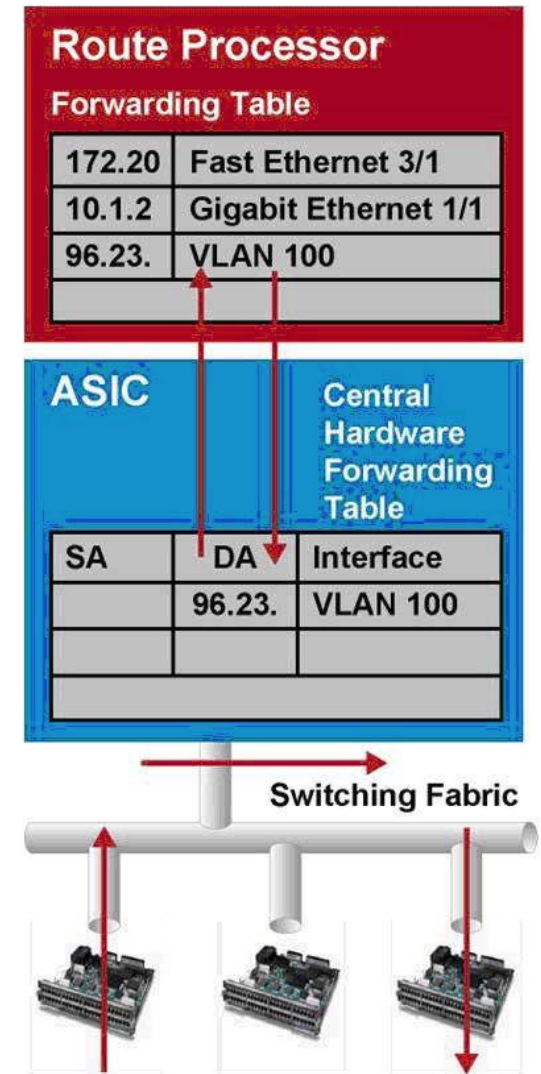
- Is a faster method by which the first packet in a flow is routed and rewritten by a route processor using software, and each subsequent packet is then handled by hardware.

- **Cisco Express Forwarding (CEF)**

- The CEF method uses hardware forwarding tables for most common traffic flows, with only a few exceptions. If you use CEF, the route processor spends its cycles mostly on other tasks.

# Fast Switching

- [Cisco Document ID: 13706, „How to Choose the Best Router Switching Path for Your Network“](#)
- **Fast Switching** is also known as **route cache**
  - Fast Switching = router, route cache = L3 switch
  - a.k.a. flow-based or demand-based switching
- *Main idea*
  - First packet in a stream is switched/routed in software
  - Information (outbound interface, MAC addresses needed for frame rewrite etc.) are then stored into the route cache (binary tree with 32 levels - fast access and search)
  - All subsequent packets in the flow are switched in the hardware
  - *Route once, forward many times*
- *Disadvantages*
  - Route cache is created only with incoming packets (first packet needs to be switched in software)
  - It is necessary to expire entries from route cache - synchronization
  - IF ARP table changes THEN some portion of the fast cache must be invalidated

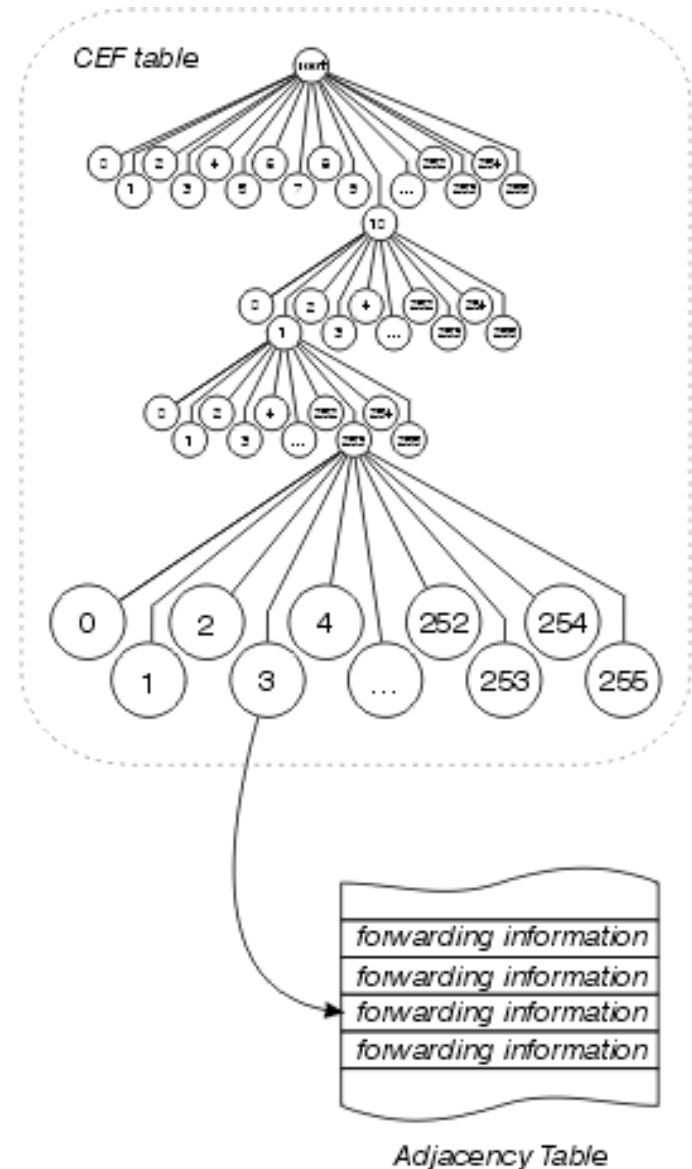


# Cisco Express Forwarding (CEF)

- a.k.a. **Topology-based Switching**
- *Main idea*
  - Routing information are cached in **Forwarding Information Base (FIB)** table
  - Frame header rewrite information for all FIB entries are cached in **adjacency table**
  - FIB entry points to appropriate entry in the adjacency table
- *Advantages*
  - Separation of control plane and data plane
  - FIB and adjacency table are created from existing entries (routing table, ARP table)
  - Change in adjacency table DOES NOT require change in FIB

# CEF

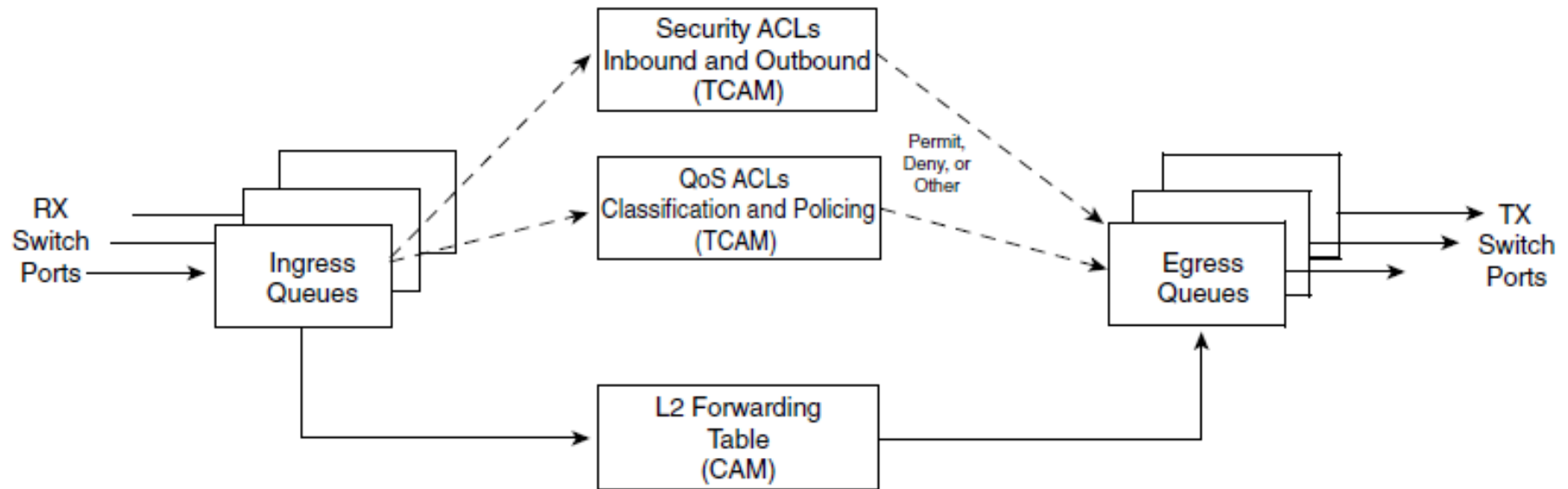
- Implementation either in software or hardware
- CEF in software
  - FIB is implemented using tree (**trie** or mtrie)
  - Adjacency table (association field)
  - Both tables are stored in RAM
  - FIB points to Adjacency table
  - Software implementation are used by smaller routers
- CEF in hardware
  - Hardware implementation uses specialized memory architecture for storing the FIB
  - **Ternary Content Addressable Memory (TCAM)**



# Caches

- CAM and TCAM are extremely fast access and allow for line-rate switching performance
- CAM and TCAM are two types of cache memory
  - CAM contains information needed for L2 switching
  - TCAM contains information needed for L3 switching, ACL, QoS
- CAM lookups for exactly the same information (exact match)
  - Matches based on two values: 0 (true) or 1 (false)
- TCAM lookups for the exact match, longest match or first match
  - Matches based on 0, 1, X (don't care)
  - Usually divided into regions with different match options
- TCAM is used for FIB typically on multilayer switch (MLS) or high-end routers
- [Cisco Support Community: “CAM vs. TCAM”](#)

# L2 Switches also Use CAM and TCAM



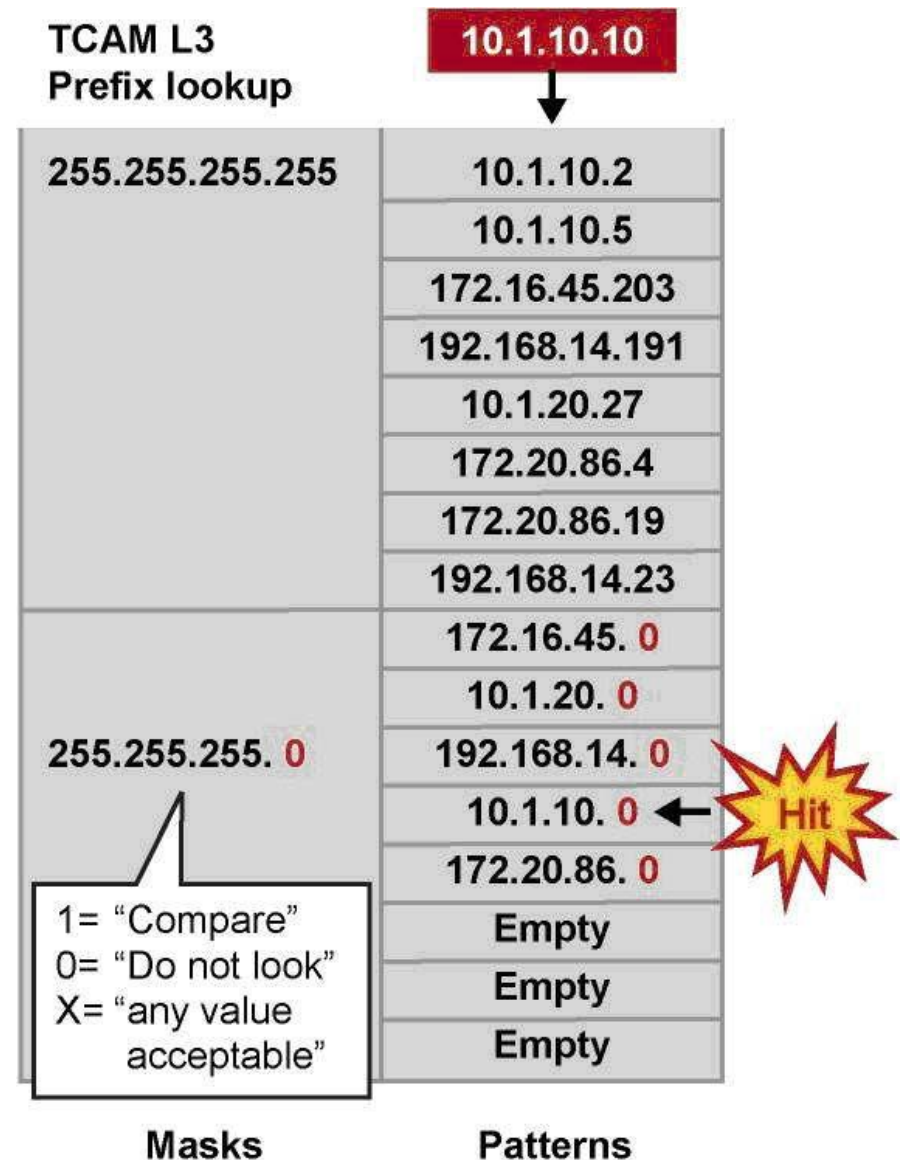
- Lookup key for CAM enters hash function
- Function returns appropriate row in table

CAM Table

MAC Address	Egress Port	VLAN

# Lookup in the TCAM

- TCAM is divided into several regions
  - Exact Match
  - Longest Match
  - First Match
- Some platforms allow to change the size of region

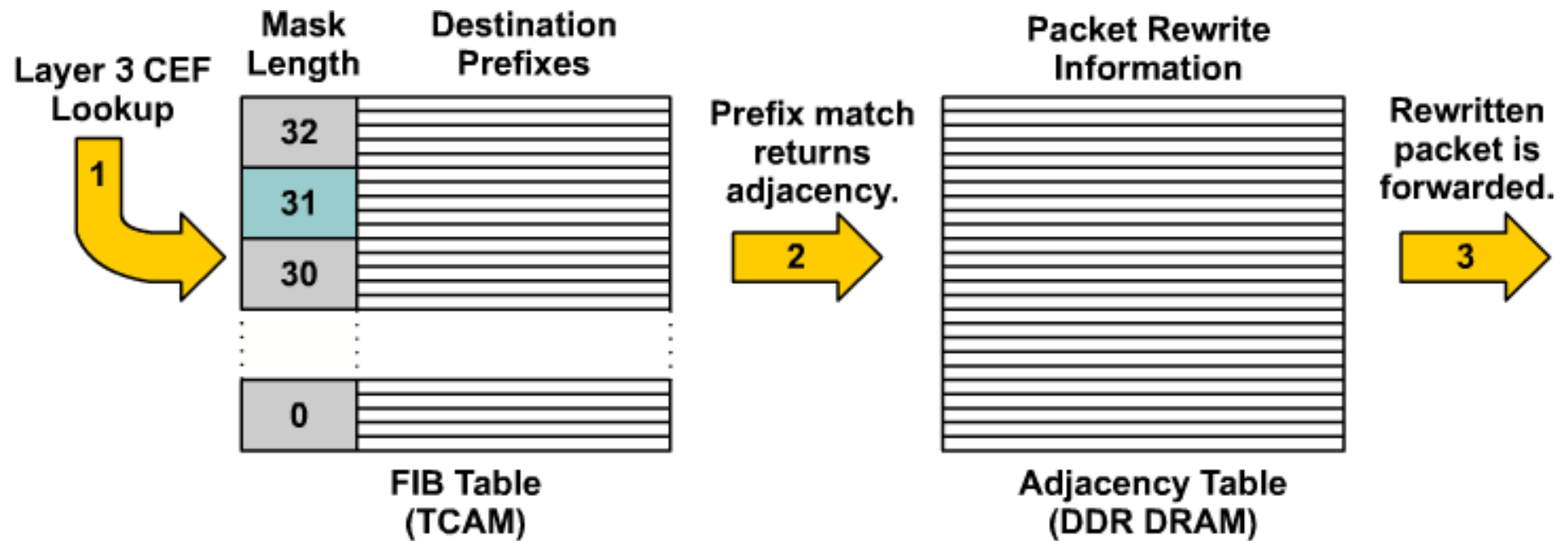




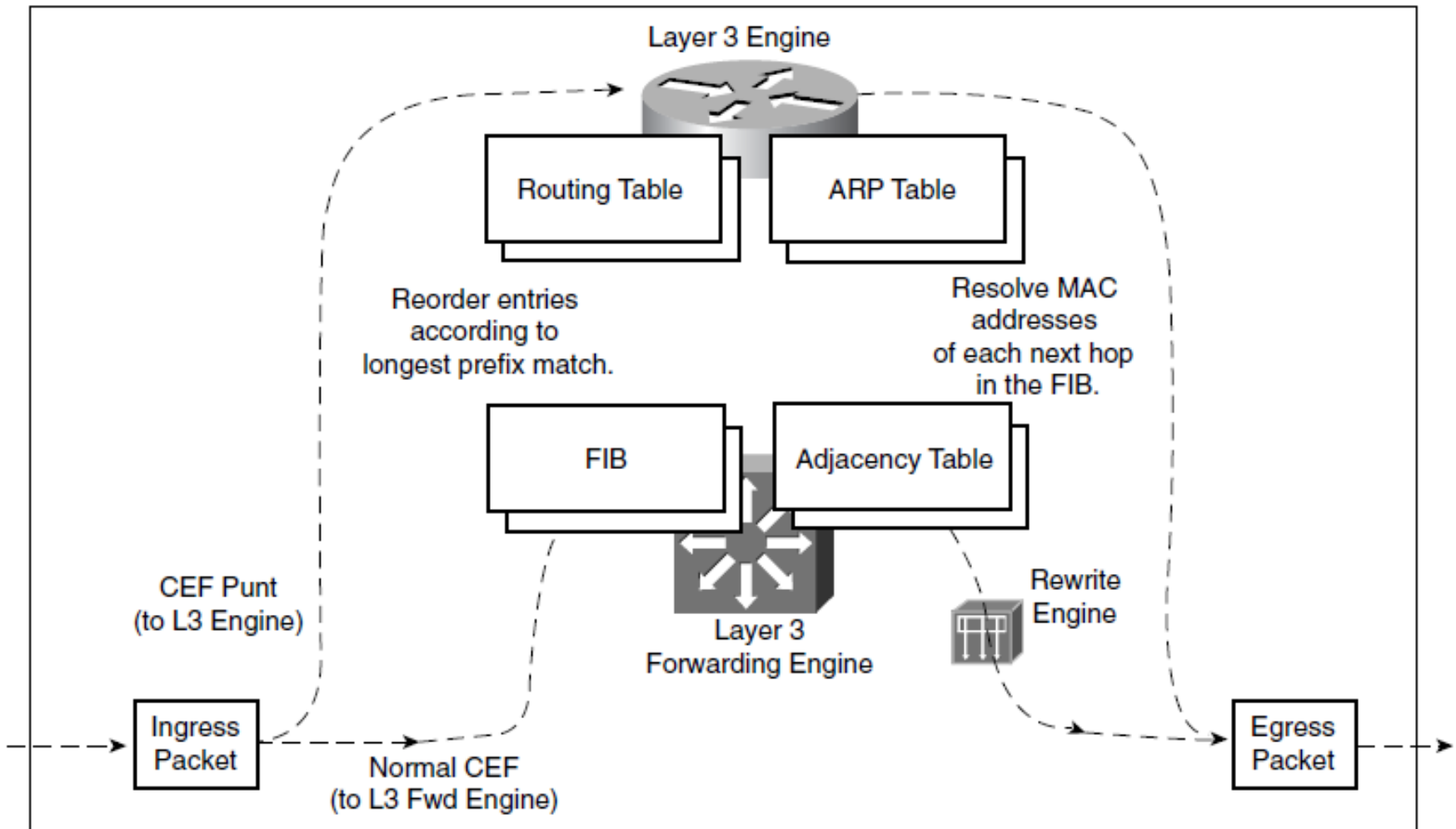
# TCAM Protocol Regions

Region Name	Cisco IOS Region Name	Lookup Type	Key Size	Sample Result
<b>IP adjacency</b>	ip-adjacency	Exact-match	32 bits	MAC address rewrite information
<b>IP prefix</b>	ip-prefix	Longest-match	32 bits	Next-hop routing information
<b>IP multicast</b>	ip-mcast	Longest-match	64 bits	Next-hop routing information
<b>Layer 2 switching</b>	l2-switching	Exact-match	64 bits	Destination interface and VLAN
<b>UDP flooding</b>	udp-flooding	Exact-match	64 bits	Next-hop routing or MAC address rewrite information
<b>Access Lists</b>	access-list	First-match	128 bits	Permit, deny, or wildcard

# Cisco Express Forwarding



# Packet Flow Using CEF MLS



# Configuring CEF

- Enabling CEF:

```
Switch(config)# ip cef
```

- CEF cannot be disabled on multilayer switch

- Enabling/disabling CEF on interface

```
Switch(config)# int fa0/1
```

```
Switch(config-if)# [no] ip route-cache cef
```

- CEF is enabled/disabled on ingress interface
- Route-cache is enabled/disabled on egress interface

- Show information from FIB and ADB

```
Switch# show ip cef [detail]
```

```
Switch# show adjacency [detail | internal | summary]
```

# Troubleshooting CEF

- 1) Verify that the IP routing information on the Layer 3 engine is correct

**show ip route**

**show ip route** *destination-network*

- 2) Verify that the next-hop address has a valid next-hop MAC address

**show ip arp** *ip-address*

- 3) Verify that the IP route entry in the FIB on the Layer 3 engine contains the same next-hop address as in ❶

**show ip cef** *destination-network*

- 4) Verify that the CEF adjacency table contains the same rewrite information as the ARP table from ❷

**show adjacency detail | begin** *next-hop-ip*

# Troubleshooting Example

!Something bad is happening with traffic to 194.160.136.5

```
sw-vd-FRI# show ip cef 194.160.136.5 detail
```

```
194.160.136.0/24, epoch 1
```

```
  nexthop 158.193.26.1 Vlan26
```

```
sw-vd-FRI# show ip arp 158.193.26.1
```

Protocol	Address	Age (min)	Hardware Addr	Type	Iface
Internet	158.193.26.1	0	00e0.4c38.c6d5	ARPA	Vlan26

```
sw-vd-FRI# show adjacency 158.193.26.1 detail
```

Protocol	Interface	Address
IP	Vlan26	158.193.26.1(11) 2 packets, 116 bytes epoch 0 sourced in sev-epoch 88 Encap length 14 00E04C38C6D5001B8F8FDE410800 ARP

# FIB Example

```
sw-vd-FRI# show ip cef
```

Prefix	Next Hop	Interface
0.0.0.0/0	158.193.7.158	Vlan709
0.0.0.0/32	receive	
158.193.7.84/30	158.193.7.158	Vlan709
158.193.7.92/30	158.193.7.158	Vlan709
158.193.7.152/29	attached	Vlan709
158.193.7.152/32	receive	Vlan709
158.193.7.157/32	receive	Vlan709
158.193.7.158/32	attached	Vlan709
158.193.7.159/32	receive	Vlan709
158.193.26.0/24	attached	Vlan26
158.193.26.0/32	receive	Vlan26
158.193.26.1/32	attached	Vlan26
158.193.26.6/32	receive	Vlan26
158.193.26.20/32	attached	Vlan26
158.193.26.21/32	attached	Vlan26

```
--More--
```

Default route  
FIB

# ADJ Example ①

```
! List of all next hop routers and connected end stations
sw-vd-FRI# show adjacency
Protocol Interface Address
...
...
IP      Vlan139      158.193.139.189 (8)
IP      Vlan139      158.193.139.190 (8)
IP      Vlan139      158.193.139.191 (8)
IP      Vlan139      158.193.139.192 (8)
IP      Vlan139      158.193.139.198 (8)
IP      Vlan139      158.193.139.199 (8)
IP      Vlan139      158.193.139.219 (8)
IP      Vlan139      158.193.139.247 (8)
IPV6    Vlan139      2001:4118:300:122:20A:5EFF:FE64:90CA (8)
IPV6    Vlan139      2001:4118:300:122:20C:29FF:FE59:1C48 (8)
IPV6    Vlan139      2001:4118:300:122:214:85FF:FEC8:ADE5 (8)
IPV6    Vlan139      2001:4118:300:122:250:4FF:FE2A:475F (8)
IPV6    Vlan139      FE80::20A:5EFF:FE64:90CA (3)
...
...
```



# ADJ Example ②

```
! Info about all next hops known in VLAN 26
sw-vd-FRI# show adjacency vlan 26 detail
Protocol Interface      Address
IP          Vlan26      158.193.26.1(13)
                2 packets, 116 bytes
                epoch 0
                sourced in sev-epoch 0
                Encap length 14
                0011D81BBE8F001B8F8FDE410800
                L2 destination address byte offset 0
                L2 destination address byte length 6
                Link-type after encap: ip
                ARP
IP          Vlan26      158.193.26.20(8)
                0 packets, 0 bytes
                epoch 0
                sourced in sev-epoch 0
                Encap length 14
                001B549640AE001B8F8FDE410800
                L2 destination address byte offset 0
                L2 destination address byte length 6
                Link-type after encap: ip
                ARP
```

# CEF Entries Flags

- **Attached**

- Entry for destination with /32 network mask. Destination is directly connected to MLS (usually the result after glean adjacency lookup)
- Next hop points to switch interface

- **Connected**

- Entry for a destination network from which the MLS has assigned IP address
- Entry is also attached

- **Receive**

- Entry for own address
- Packets whose final destinations include the router itself
  - packets destined to the router itself
  - broadcast and multicasts packets
- MLS MUST receive and process this packet

- **Recursive**

- Recursive lookup

- **Default Route Handler**

- Default routing entry

# Generic CEF Entries

- Always present in CEF
- 0.0.0.0/32 receive
  - Process packets sent to 0.0.0.0 broadcast address
- 224.0.0.0/4 drop
  - Discard multicast packets
- 224.0.0.0/24 receive
  - Process packets from link-local scope 224.0.0.0 – 224.0.0.255
  - Used by e.g. routing protocols
- 255.255.255.255/32 receive
  - Process packets sent to link-local broadcast address

# Types of CEF Adjacencies

- **Auto**

- Ordinary entries

- **Punt**

- Packets cannot be processed in CEF
- Packets are process by next best method (Fast / Process Switching)

- **Glean**

- Represents unknown end-stations on locally connected network (switch still does not know the MAC address)
- Prefix has Glean adjacency
- ARP throttling

- **Drop**

- Packets should be discarded, ICMP message can be sent

- **Discard**

- Packets should be discarded without sending the ICMP message
- Used for Loopback's IP addresses

- **Null**

- Packets should be forwarded to the Null0 interface, i.e. discard

- **No route**

- In the case of missing default route
- ICMP message is sent

- **Cached and uncached**

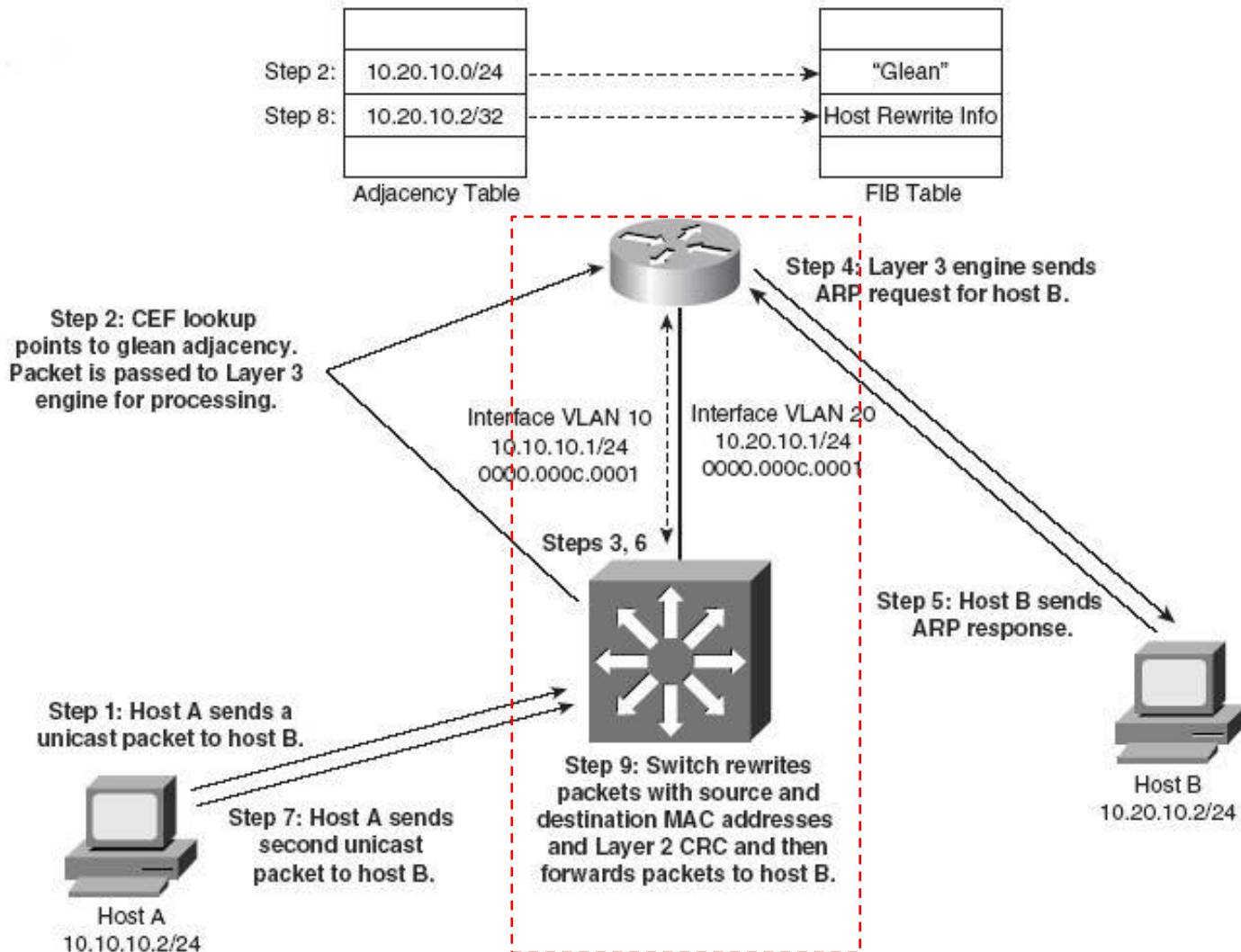
- **Unresolved**

- Adjacency for a next hop is missing

# Glean Adjacency

- Glean adjacency represents all end stations directly connected to the switch, with L2 rewriting information missing
  - Typically missing IP/MAC mapping in ARP table
- Incoming packet is processed as follows
  1. CEF lookup return Glean type
  2. MLS sends an ARP request and installs the drop adjacency (ARP throttling) for searched address
  3. MLS installs adjacency for the address and removes the drop adjacency with attached entry

# CEF Operation



# ADJ Example ③

```
sw-vd-FRI# show ip cef adjacency ?
```

adj-null	Null Adjacency
discard	Discard Adjacency
drop	Drop Adjacency
fcpa	Fiber Channel
glean	Glean Adjacency
punt	Punt Adjacency

```
sw-vd-FRI# show ip cef adjacency glean
```

Prefix	Next Hop	Interface
158.193.7.152/29	attached	Vlan709
158.193.26.0/24	attached	Vlan26
158.193.128.0/24	attached	Vlan128

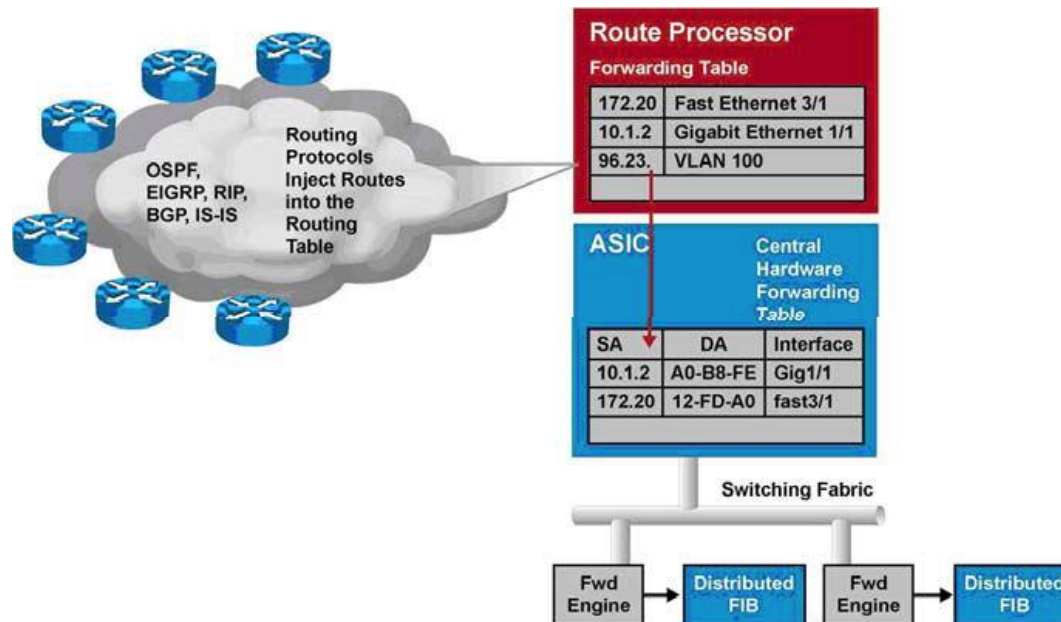
# Packet Types Forcing Software Processing

- CEF DOES NOT support following
  - Packets without valid CEF entry
  - Packets destined to the switch itself
  - Broadcast, multicast
  - Use of IP header options (packets that use TCP header options are switched in hardware because they do not affect the forwarding decision)
  - Fragmented IP packets (exceeded the outbound interface MTU)
  - Packets that have an expiring IP TTL counter
  - Network Address Translation
  - Encrypted packets
- IF the TCAM table is full THEN wildcard entry pointing to software process is inserted
  - May overload the CPU



# Distributed CEF (dCEF)

- Central mode: FIB and adjacency tables reside on the route processor
- **Distributed CEF** can be enabled on selected platforms
  - Line cards maintain identical copies of the FIB and adjacency tables
  - Line cards can perform the forwarding by themselves



# Switch Database Management



# SDM Templates ①

- TCAM is the essential component for L3 switch
  - Used by several applications
  - Expensive thus small
  - TCAM can be tuned according to MLS role
- **SDM Templates**
  - Predefined, depends on platform
- E.g. SDM templates for 3560:
  - **Access** – maximizes space for ACLs
  - **Default** – balanced distribution
  - **Routing** – maximizes space for routing entries
  - **VLANs** – maximizes space for L2 switching
  - **Dual-IPv4-and-IPv6 Default, Routing, VLAN**

# SDM Templates ②

- Some features are not supported when the TCAM is not properly initialized
  - E.g. IPv6 or Policy Based Routing

```
Switch# show sdm prefer
```

```
The current template is "desktop default" template.  
The selected template optimizes the resources in  
the switch to support this level of features for  
8 routed interfaces and 1024 VLANs.
```

number of unicast mac addresses:	6K
number of IPv4 IGMP groups + multicast routes:	1K
number of IPv4 unicast routes:	8K
number of directly-connected IPv4 hosts:	6K
number of indirect IPv4 routes:	2K
number of IPv4 policy based routing aces:	0
number of IPv4/MAC qos aces:	0.5K
number of IPv4/MAC security aces:	1K

# SDM Templates ③

```
Switch# show sdm prefer ?
```

```
access          Access bias
default         Default bias
dual-ipv4-and-ipv6  Support both IPv4 and IPv6
routing         Unicast bias
vlan            VLAN bias
<cr>
```

```
Switch# show sdm prefer routing
```

"desktop routing" template:

The selected template optimizes the resources in the switch to support this level of features for 8 routed interfaces and 1024 VLANs.

number of unicast mac addresses:	3K
number of IPv4 IGMP groups + multicast routes:	1K
number of IPv4 unicast routes:	11K
number of directly-connected IPv4 hosts:	3K
number of indirect IPv4 routes:	8K
number of IPv4 policy based routing aces:	0.5K
number of IPv4/MAC qos aces:	0.5K
number of IPv4/MAC security aces:	1K

# Change SDM Template

- Global config mode

```
Switch(config)# sdm prefer ?
```

access	Access bias
default	Default bias
dual-ipv4-and-ipv6	Support both IPv4 and IPv6
ipe	IPE bias
routing	Unicast bias
vlan	VLAN bias

```
Switch(config)# sdm prefer routing
```

Changes to the running SDM preferences have been stored, but cannot take effect until the next reload.

Use 'show sdm prefer' to see what SDM preference is currently active.

```
Switch(config)# exit
```

```
Switch# write
```

```
Switch# reload
```

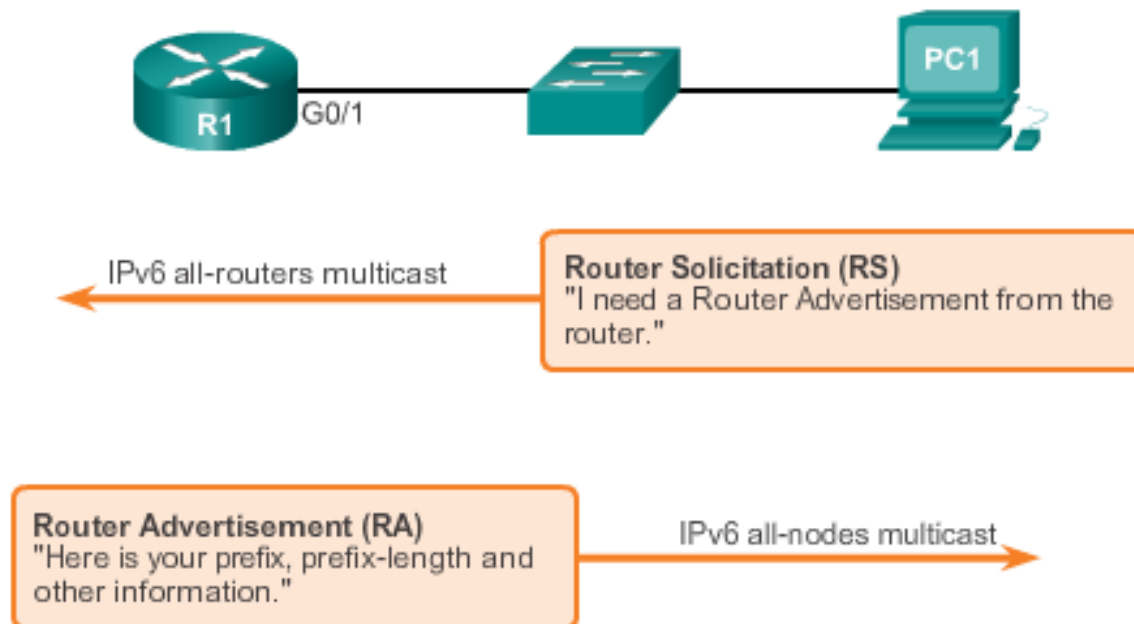
# DHCPv6



# Stateless Address Autoconfiguration

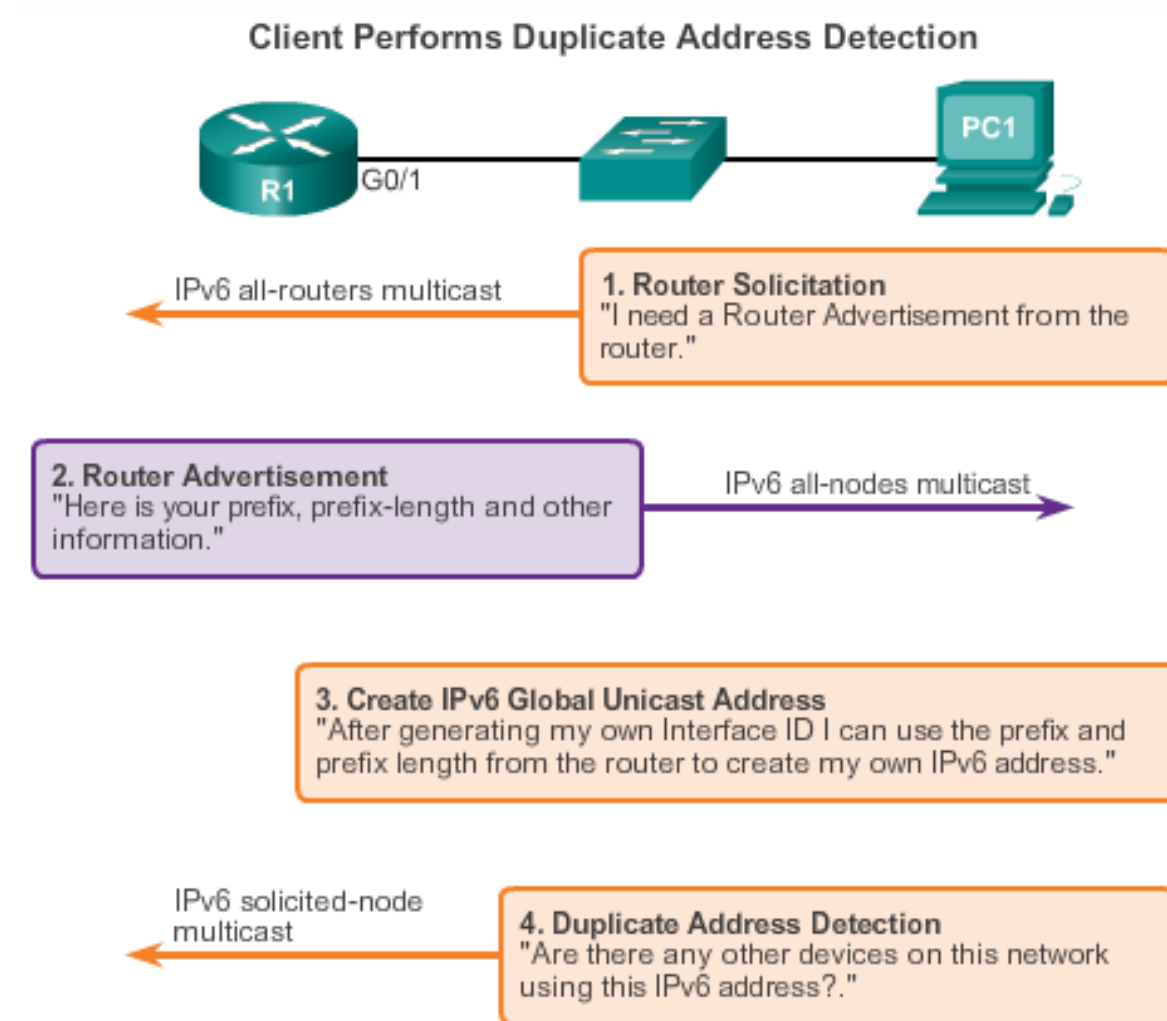
Stateless Address Autoconfiguration (SLAAC) is a method in which a device can obtain an IPv6 global unicast address without the services of a DHCPv6 server.

## ICMPv6 Stateless Address Autoconfiguration

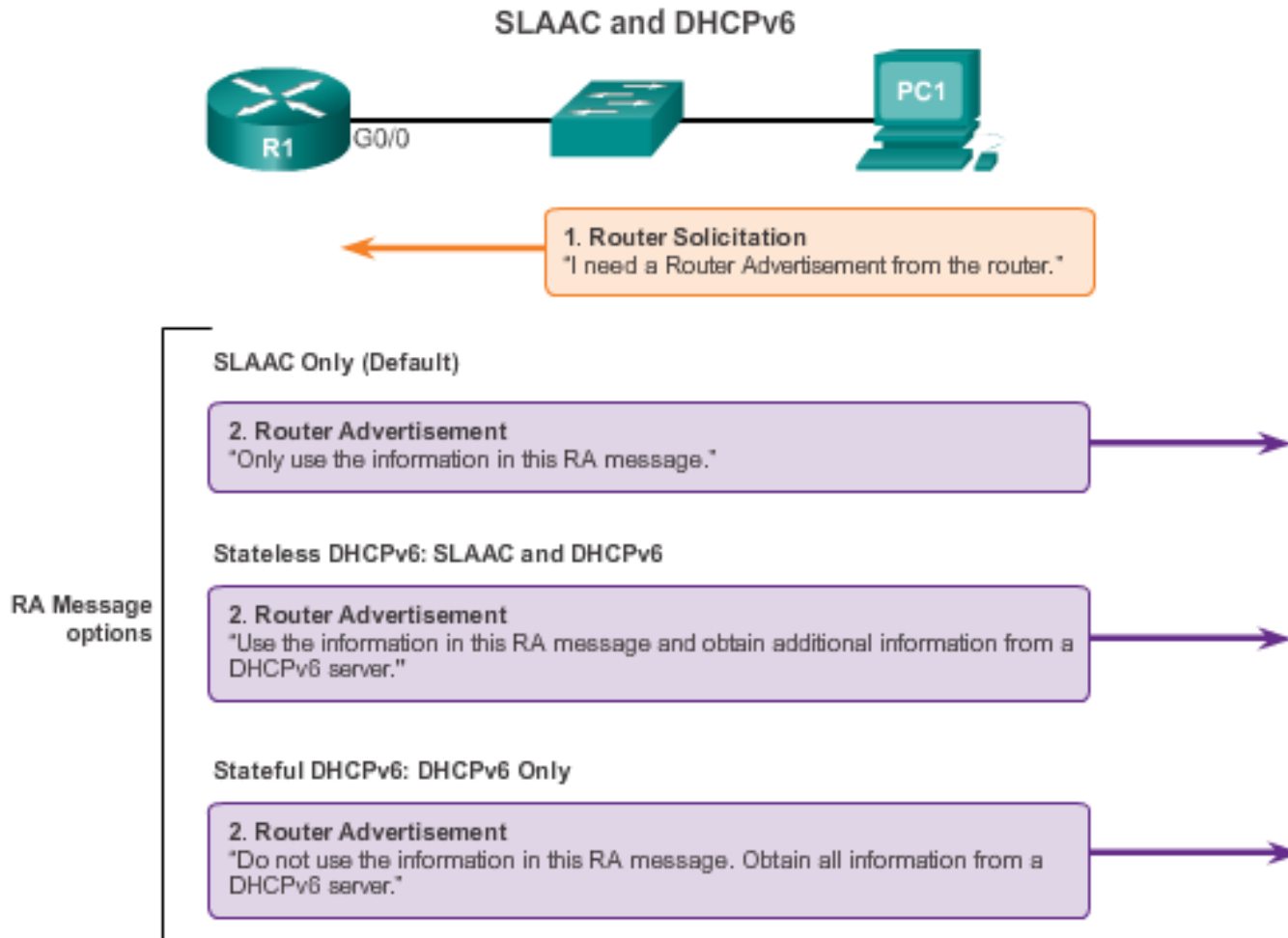




# SLAAC Operation

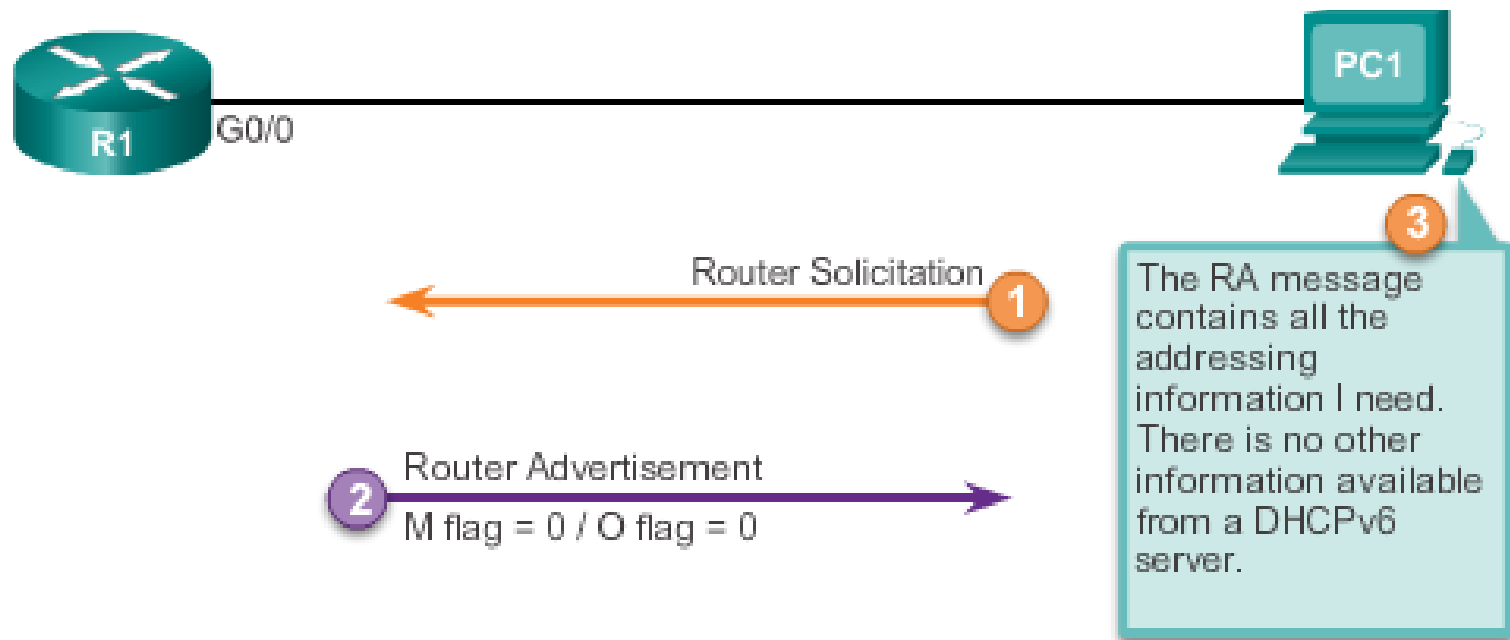


# SLAAC and DHCPv6



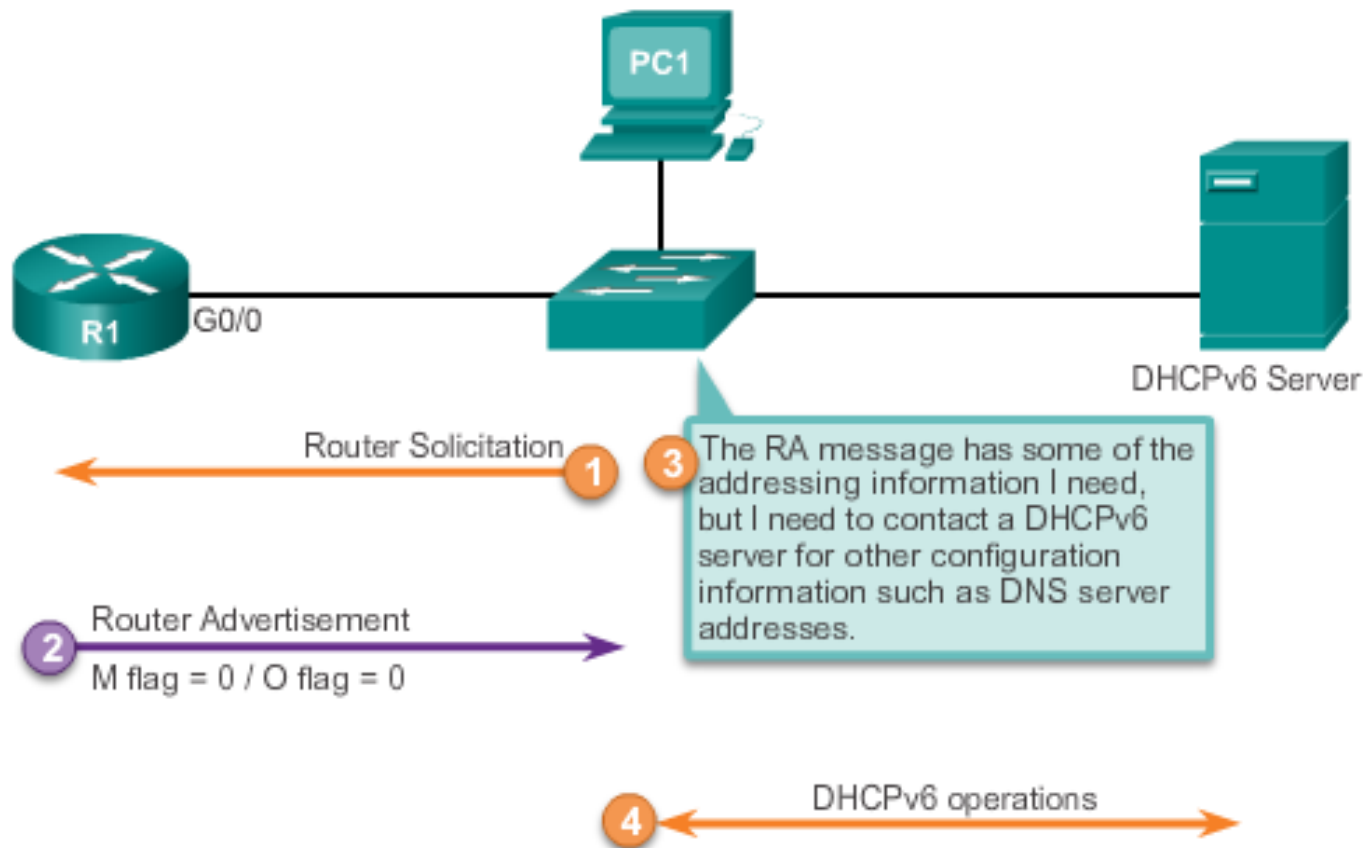
# SLAAC Option

## SLAAC Option



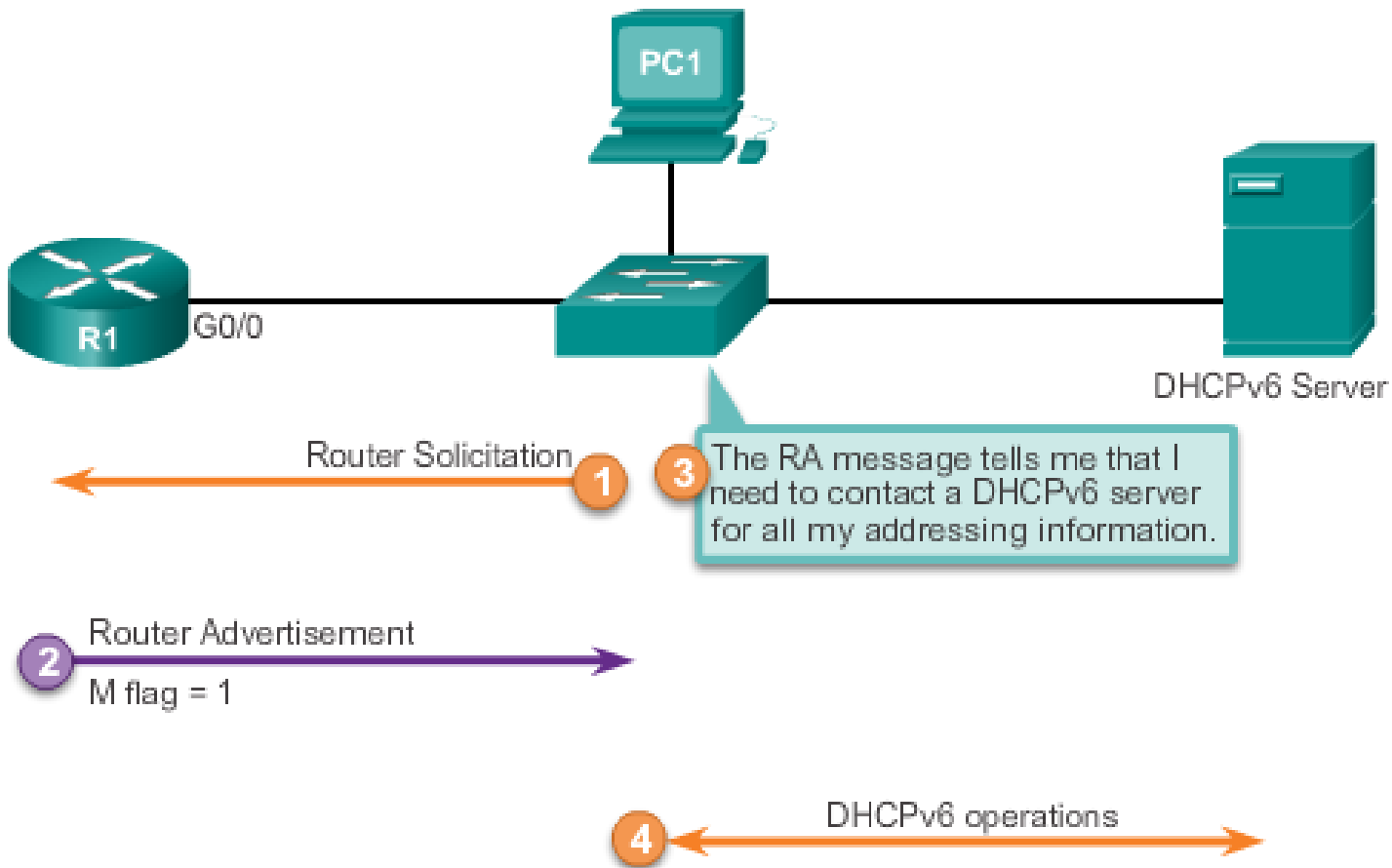
# Stateless DHCP Option

## Stateless DHCPv6 Option

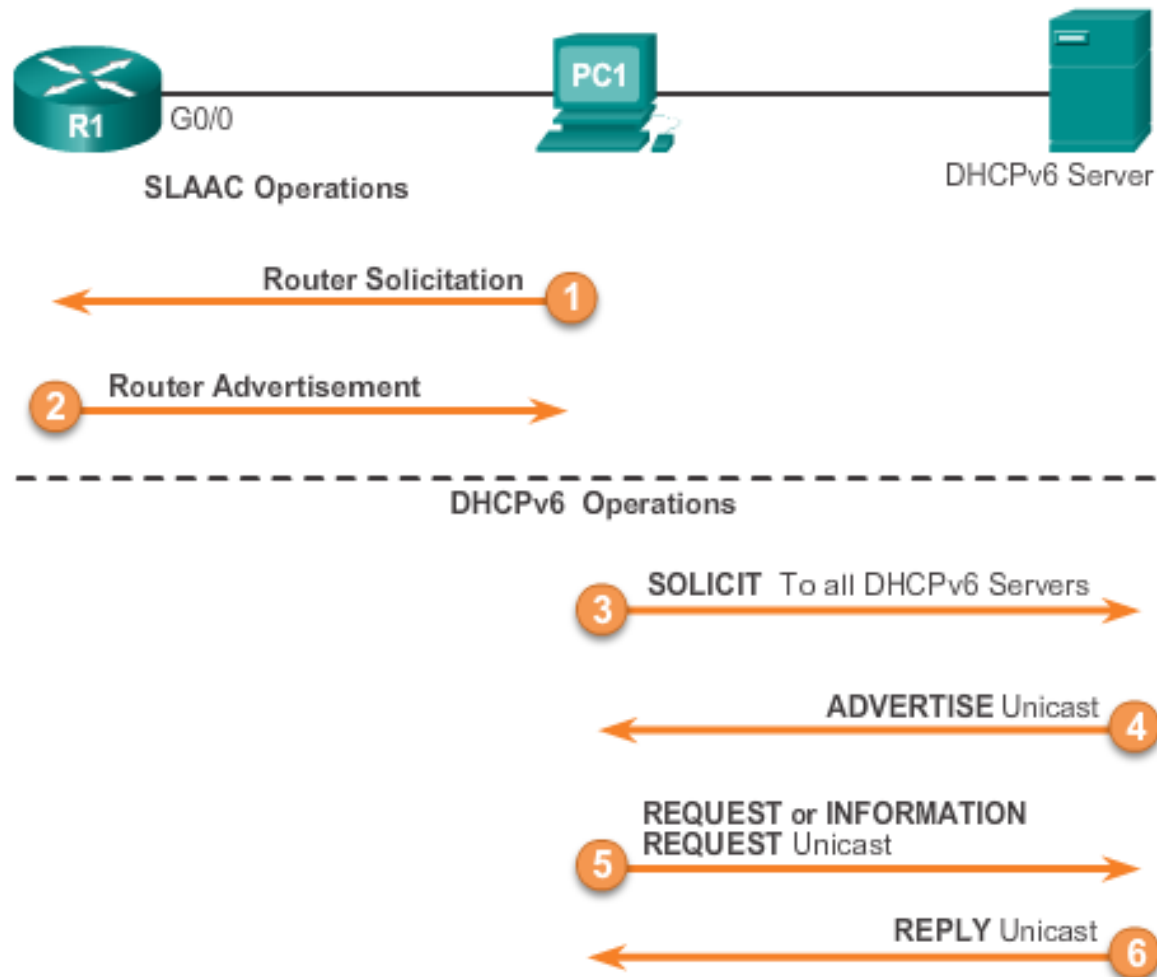


# Stateful DHCP Option

## Stateful DHCPv6 Option

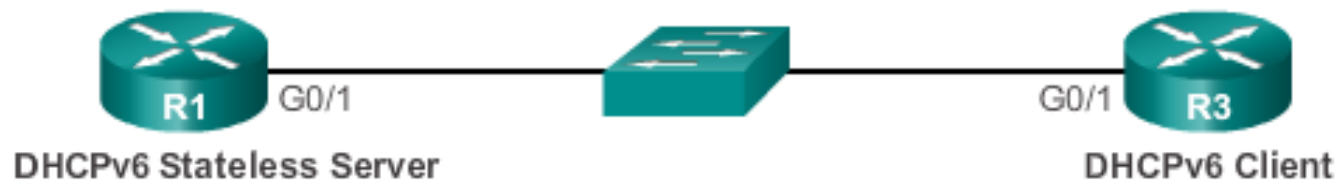


# DHCPv6 Operations



# Configuring a Stateless DHCPv6 Server

- a.k.a. **DHCPv6 Lite**



```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 dhcp pool IPV6-STATELESS
R1(config-dhcpv6)# dns-server 2001:db8:cafe:aaaa::5
R1(config-dhcpv6)# domain-name example.com
R1(config-dhcpv6)# exit
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:db8:cafe:1::1/64
R1(config-if)# ipv6 dhcp server IPV6-STATELESS
R1(config-if)# ipv6 nd other-config-flag
```

# Configuring a Router as a Stateless DHCPv6 Client



```
R3(config)# interface g0/1
R3(config-if)# ipv6 enable
R3(config-if)# ipv6 address autoconfig
R3(config-if)#
```



# Verifying Stateless DHCPv6

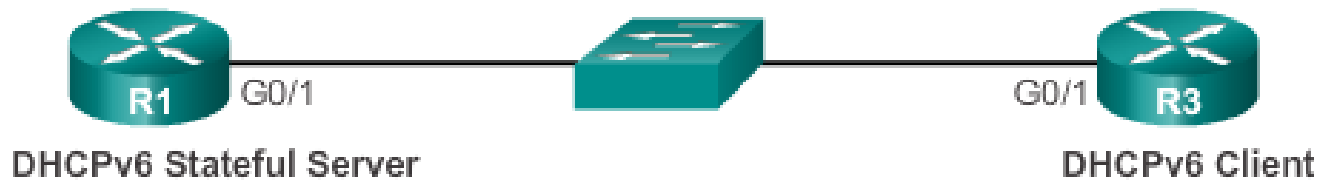
Verify the stateless DHCP client using the following commands:

- **show ipv6 interface**
- **debug ipv6 dhcp detail**

## Verifying the Stateless DHCPv6 Server

```
R1# show ipv6 dhcp pool
DHCPv6 pool: IPV6-STATELESS
  DNS server: 2001:DB8:CAFE:AAAA::5
  Domain name: example.com
  Active clients: 0
R1#
```

# Configuring a Router as a Stateful DHCPv6 Server



Configuring a Router as a Stateful DHCPv6 Server

```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 dhcp pool IPV6-STATEFUL
R1(config-dhcpv6)# address prefix 2001:DB8:CAFE:1::/64
                    lifetime infinite infinite
R1(config-dhcpv6)# dns-server 2001:db8:cafe:aaaa::5
R1(config-dhcpv6)# domain-name example.com
R1(config-dhcpv6)# exit
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:db8:cafe:1::1/64
R1(config-if)# ipv6 dhcp server IPV6-STATEFUL
R1(config-if)# ipv6 nd managed-config-flag
```

# Verifying Stateful DHCPv6

- Verify the stateful DHCPv6 server using the following commands:

```
show ipv6 dhcp pool
```

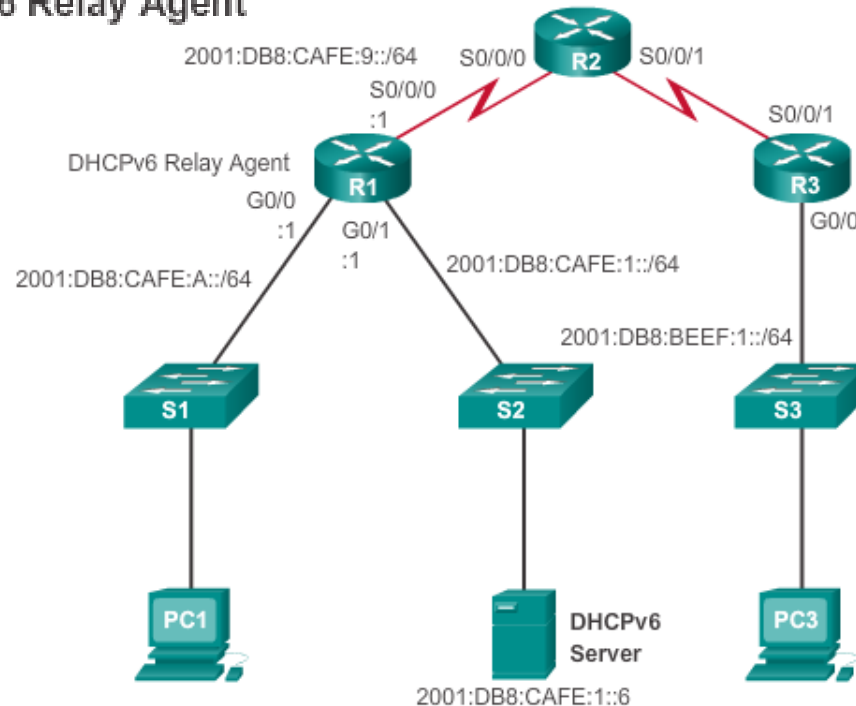
```
show ipv6 dhcp binding
```

- Verify the stateful DHCPv6 client using the **show ipv6 interface** command.

```
R3# show ipv6 interface g0/1
GigabitEthernet0/1 is up, line protocol is up
IPv6 is enabled, link-local address is
FE80::32F7:DFF:FE25:2DE1
No Virtual link-local address(es):
Global unicast address(es):
  2001:DB8:CAFE:1:5844:47B2:2603:C171, subnet is
2001:DB8:CAFE:1:5844:47B2:2603:C171/128
Joined group address(es):
  FF02::1
  FF02::1:FF03:C171
  FF02::1:FF25:2DE1
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 30000)
ND NS retransmit interval is 1000 milliseconds
Default router is FE80::D68C:B5FF:FECE:A0C1 on
```

# Configuring a Stateful DHCPv6 Relay Agent

## DHCPv6 Relay Agent



```
R1(config)# interface g0/0
R1(config-if)# ipv6 dhcp relay destination 2001:db8:cafe:1::6
R1(config-if)# end
R1# show ipv6 dhcp interface g0/0
GigabitEthernet0/0 is in relay mode
Relay destinations:
  2001:DB8:CAFE:1::6
R1#
```

# Verifying the Router DHCPv6 Configuration

```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 dhcp pool IPV6-STATEFUL
R1(config-dhcpv6)# address prefix 2001:DB8:CAFE:1::/64 lifetime
infinite infinite
R1(config-dhcpv6)# dns-server 2001:db8:cafe:aaaa::5
R1(config-dhcpv6)# domain-name example.com
R1(config-dhcpv6)# exit
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:db8:cafe:1::1/64
R1(config-if)# ipv6 dhcp server IPV6-STATEFUL
R1(config-if)# ipv6 nd managed-config-flag
```

## Stateless DHCPv6 Services

```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 dhcp pool IPV6-STATELESS
R1(config-dhcpv6)# dns-server 2001:db8:cafe:aaaa::5
R1(config-dhcpv6)# domain-name example.com
R1(config-dhcpv6)# exit
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:db8:cafe:1::1/64
R1(config-if)# ipv6 dhcp server IPV6-STATELESS
R1(config-if)# ipv6 nd other-config-flag
```

# Debugging DHCPv6

```
R1# debug ipv6 dhcp detail
    IPv6 DHCP debugging is on (detailed)
R1#
*Feb  3 21:27:41.123: IPv6 DHCP: Received SOLICIT from
FE80::32F7:DFF:FE25:2DE1 on GigabitEthernet0/1
*Feb  3 21:27:41.123: IPv6 DHCP: detailed packet contents
*Feb  3 21:27:41.123:   src FE80::32F7:DFF:FE25:2DE1
(GigabitEthernet0/1)
*Feb  3 21:27:41.127:   dst FF02::1:2
*Feb  3 21:27:41.127:   type SOLICIT(1), xid 13190645
*Feb  3 21:27:41.127:   option ELAPSED-TIME(8), len 2
*Feb  3 21:27:41.127:     elapsed-time 0
*Feb  3 21:27:41.127:   option CLIENTID(1), len 10
*Feb  3 21:27:41.127:     000
*Feb  3 21:27:41.127: IPv6 DHCP: Using interface pool IPV6-
STATEFUL
*Feb  3 21:27:41.127: IPv6 DHCP: Creating binding for
FE80::32F7:DFF:FE25:2DE1 in pool IPV6-STATEFUL
<Output omitted>
```

# Where to go next?

- [www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2\\_52\\_se/command/reference/3560cr.html](http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/command/reference/3560cr.html)

## Configuring IP Unicast Routing Configuration Guide:

- [www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2\\_52\\_se/configuration/guide/swi](http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/configuration/guide/swi)

## Configuring EtherChannels:

- [www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2\\_52\\_se/configuration/guide/swethchl.htmlprout.html](http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/configuration/guide/swethchl.htmlprout.html)

## Configuring DHCP:

- [www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2\\_52\\_se/configuration/guide/swdhcp82.html](http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/configuration/guide/swdhcp82.html)



Slides adapted by [Vladimír Veselý](#) partially from official course materials  
but most of credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

The last update: 2016-10-20