

---

---

The Notebook  
of  
**CCNA**

---

HUY BUI

**PL**

THE PUBLISHER



# Contents

0.1 List of Codes . . . . .	6
<b>I NETWORK FUNDAMENTALS</b>	<b>7</b>
<b>1 Seven layers</b>	<b>9</b>
1.1 Introductions . . . . .	9
1.2 Physical layer . . . . .	10
1.3 Data Link layer . . . . .	12
1.4 Network layer . . . . .	16
1.5 Transport layer . . . . .	19
1.6 Application layer . . . . .	22
<b>2 LAN Design</b>	<b>27</b>
2.1 Hierarchical Design Model . . . . .	27
2.2 Expanding the network . . . . .	28
2.3 Selecting network devices . . . . .	29
<b>3 WAN concepts</b>	<b>31</b>
3.1 WAN Technologies overview . . . . .	31
3.2 WAN connection . . . . .	34
<b>4 Network evolution</b>	<b>37</b>
4.1 IoT . . . . .	37
4.2 Cloud computing . . . . .	37
4.3 Virtualization . . . . .	38
4.4 SDN . . . . .	39
<b>II SWITCHING TECHNOLOGIES</b>	<b>45</b>
<b>5 Switch</b>	<b>47</b>
5.1 Operation . . . . .	47
5.2 Recovering from system crash . . . . .	49
5.3 Configuration . . . . .	49
5.4 Troubleshooting . . . . .	51
<b>6 VLAN</b>	<b>55</b>
6.1 Overview . . . . .	55
6.2 VLAN tag . . . . .	56
6.3 Configuration . . . . .	57
6.4 Inter-VLAN Routing . . . . .	60
<b>7 VTP</b>	<b>63</b>
7.1 Overviews . . . . .	63
7.2 Operations . . . . .	64
7.3 VTP Caveats . . . . .	64

<b>8 STP</b>	<b>67</b>
8.1 Layer 2 loop . . . . .	67
8.2 STP overview . . . . .	67
8.3 Bridge ID . . . . .	68
8.4 BPDU frame . . . . .	68
8.5 Root bridge election . . . . .	68
8.6 Port Roles . . . . .	69
8.7 PVST+ . . . . .	70
8.8 RSTP and Rapid PVST+ . . . . .	73
8.9 Configuration . . . . .	73
<b>9 EtherChannel</b>	<b>75</b>
9.1 Introduction . . . . .	75
9.2 Configuration . . . . .	76
<b>III ROUTING TECHNOLOGIES</b>	<b>79</b>
<b>10 Router</b>	<b>81</b>
10.1 Introduction . . . . .	81
10.2 Path determination . . . . .	82
<b>11 IPv6</b>	<b>83</b>
11.1 Representation . . . . .	83
11.2 Types of IPv6 Addresses . . . . .	83
11.3 ICMPv6 . . . . .	85
11.4 EUI-64 Process . . . . .	86
11.5 IPv4 and IPv6 Coexistence . . . . .	86
<b>12 Static routing</b>	<b>87</b>
12.1 Overview . . . . .	87
12.2 IPv4 route configuration . . . . .	87
12.3 IPv6 route configuration . . . . .	88
12.4 Types . . . . .	88
<b>13 EIGRP</b>	<b>91</b>
13.1 Basic features . . . . .	91
13.2 Packet types . . . . .	91
13.3 Encapsulating EIGRP Messages . . . . .	92
13.4 Operation . . . . .	94
13.5 Configuration . . . . .	98
<b>14 OSPF</b>	<b>101</b>
14.1 Overview . . . . .	101
14.2 Protocol components . . . . .	103
14.3 DR election . . . . .	105
14.4 OSPF area . . . . .	106
14.5 Configuration . . . . .	107
<b>IV WAN TECHNOLOGIES</b>	<b>111</b>
<b>15 PPP</b>	<b>113</b>
15.1 Introduction . . . . .	113
15.2 Operation . . . . .	113
15.3 Configuration . . . . .	116

<b>CONTENTS</b>	<b>5</b>
<b>16 PPPoE, GRE, eBGP</b>	<b>119</b>
16.1 PPPoE . . . . .	119
16.2 GRE . . . . .	120
16.3 eBGP . . . . .	121
<b>17 Quality of Service</b>	<b>123</b>
17.1 Introduction . . . . .	123
17.2 Classification and marking . . . . .	123
17.3 Congestion Avoidance . . . . .	126
17.4 Congestion management . . . . .	126
17.5 QoS models . . . . .	127
<b>V INFRASTRUCTURE SERVICE</b>	<b>131</b>
<b>18 DHCPv4</b>	<b>133</b>
18.1 Operation . . . . .	133
18.2 Message . . . . .	134
18.3 Configuration . . . . .	134
<b>19 DHCPv6</b>	<b>137</b>
19.1 General Operation . . . . .	137
19.2 SLAAC . . . . .	137
19.3 SLAAC and Stateless DHCPv6 . . . . .	138
19.4 Stateful DHCPv6 . . . . .	139
19.5 Router as DHCPv6 client . . . . .	140
19.6 Relay agent . . . . .	140
19.7 Message . . . . .	140
<b>20 HSRP</b>	<b>141</b>
20.1 Operations . . . . .	141
20.2 Priority . . . . .	142
20.3 Preemption . . . . .	142
20.4 States and timers . . . . .	142
20.5 Configuration . . . . .	142
<b>VI INFRASTRUCTURE SECURITY</b>	<b>143</b>
<b>21 NAT</b>	<b>145</b>
21.1 What is NAT? . . . . .	145
21.2 Operation . . . . .	147
21.3 Static NAT . . . . .	147
21.4 Dynamic NAT . . . . .	147
21.5 PAT . . . . .	148
21.6 Port forwarding . . . . .	150
<b>22 ACL</b>	<b>151</b>
22.1 ACL Operation Overview . . . . .	151
22.2 Standard ACL . . . . .	152
22.3 Extended ACLs . . . . .	152
22.4 IPv6 ACLs . . . . .	153
22.5 Configurations . . . . .	154
22.6 Troubleshoot . . . . .	156

<b>VII INFRASTRUCTURE MANAGEMENT</b>	<b>159</b>
<b>23 Network Security and Monitoring</b>	<b>161</b>
23.1 Security attacks . . . . .	161
23.2 SNMP . . . . .	163
23.3 SPAN . . . . .	166
<b>24 Troubleshooting</b>	<b>169</b>
24.1 Documentation . . . . .	169
24.2 Troubleshooting process . . . . .	170
24.3 Using IP SLA . . . . .	172
24.4 Troubleshooting tools . . . . .	173
24.5 Symptoms and causes . . . . .	174
24.6 Troubleshooting end-to-end connectivity . . . . .	174
24.7 Scenarios . . . . .	177
<b>25 CDP, LLDP, NTP, and Syslog</b>	<b>179</b>
25.1 CDP . . . . .	179
25.2 LLDP . . . . .	180
25.3 NTP . . . . .	180
25.4 Syslog . . . . .	181
<b>26 Device maintenance</b>	<b>185</b>
26.1 File system . . . . .	185
26.2 Back up and Restore . . . . .	185
26.3 IOS image . . . . .	188
26.4 Software licensing . . . . .	189

## 0.1 List of Codes

1 Listing 1: Setting boot environment variable . . . . .	47
2 Listing 2: Basic switch management . . . . .	49
3 Listing 3: Duplex configuration . . . . .	50
4 Listing 4: Auto-MDIX . . . . .	50
5 Listing 5: SSH configuration . . . . .	50
6 Listing 6: Port security configuration . . . . .	51
7 Listing 7: Trunking . . . . .	58
8 Listing 8: Voice VLAN example . . . . .	59
9 Listing 9: VLAN verification . . . . .	59
10 Listing 10: Trunking verification . . . . .	59
11 Listing 11: Router-on-a-stick . . . . .	61
12 Listing 12: Multiplayer switch . . . . .	62
13 Listing 13: STP . . . . .	74
14 Listing 14: EtherChannel . . . . .	77
15 Listing 15: EIGRP for IPv4 . . . . .	98
16 Listing 16: EIGRP for IPv6 . . . . .	99
17 Listing 17: OSPF for IPv4 . . . . .	107
18 Listing 18: OSPF for IPv6 . . . . .	108
19 Listing 19: Basic PPP configuration . . . . .	116
20 Listing 20: Multilink PPP . . . . .	117
21 Listing 21: CHAP authentication . . . . .	117
22 Listing 22: PAP authentication . . . . .	118
23 Listing 23: PPPoE . . . . .	119
24 Listing 24: IP SLA example . . . . .	173
25 Listing 25: Assign a name to a router or switch . . . . .	177

# **Part I**

# **NETWORK FUNDAMENTALS**



# Chapter 1

## Seven layers

### 1.1 Introductions

A group of inter-related protocols necessary to perform a communication function is called a **protocol suite**. The **TCP/IP protocol suite** is an open standard, meaning these protocols are freely available to the public, and any vendor is able to implement these protocols on their hardware or in their software. The TCP/IP protocol suite includes many protocols, as shown in Figure 1.1.

Figure 1.1: TCP/IP protocol suite and Communication process

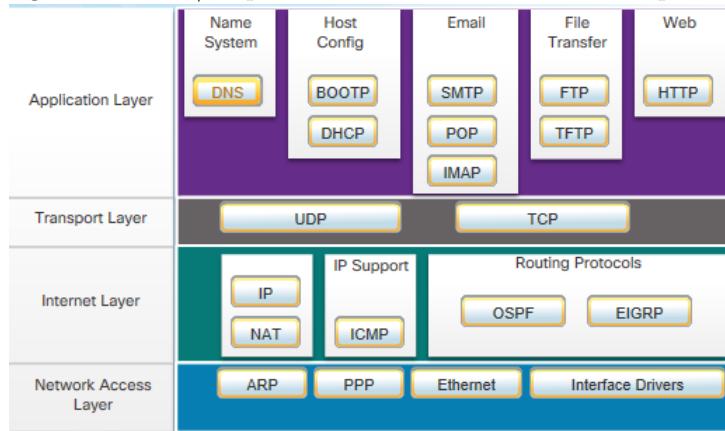
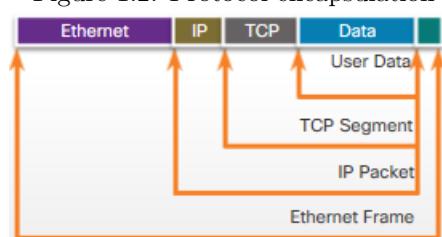


Figure 1.2 demonstrate the complete protocol encapsulation during the **TCP/IP communication process**. The process begins with the web server preparing the HTML page as *data* to be sent. The application protocol *HTTP* header is added to the front of the HTML data. The transport layer divides a data stream into segments and may add reliability and flow control information (TCP). Next, the Layer 3 (network layer) adds IP addresses and control information to each segment to create packets. Then, Layer 2 (Data Link layer) protocol (Ethernet, PPP, HDLC, etc.) adds information to both ends of the IP packet, known as a data link frame. This data is now ready to be transported through the internetwork via Layer 1 (Physical layer).

Figure 1.2: Protocol encapsulation



Whereas the TCP/IP model layers (table 1.2) are referred to only by name, the seven OSI model layers (table 1.1) are more often referred to by number rather than by name. For instance, the physical layer is referred to as Layer 1 of the OSI model.

Table 1.1: The OSI Reference Model

Number	Layer	Description
7	Application	contain protocols used for process-to-process communications
6	Presentation	provide for common representation of the data transferred between application layer services
5	Session	organize dialog and manage data exchange
4	Transport	segment, transfer, and reassemble
3	Network	exchange the individual pieces of data over the network
2	Data Link	exchange data frames between devices over a common media
1	Physical	describe the mechanical, electrical media to create physical connections for bit transmission

Table 1.2: The TCP/IP Reference Model

Layer	Description
Application	represent data to user plus encoding and dialog control
Transport	support communication between various devices across diverse networks
Internet	determine best path through the network
Network Access	control hardware devices and media

## 1.2 Physical layer

### 1.2.1 Introduction

The OSI physical layer provides the means to transport the bits that make up a data link layer frame across the network media. This layer accepts a complete frame from the data link layer and encodes it as a series of signals that are transmitted onto the local media.

The physical layer standards address three functional areas:

- **Physical Components:** electronic hardware devices, media, and other connectors that transmit and carry the signals to represent the bits.
- **Encoding:** a method of converting a stream of data bits into a predefined code. Codes are groupings of bits used to provide a predictable pattern that can be recognized by both the sender and the receiver.
- **Signaling:** The physical layer must generate the electrical, optical, or wireless signals that represent the 1 and 0 on the media. The method of representing the bits is called the *signaling method*. A common signaling method is using modulation techniques. *Modulation* is the process by which the characteristic of one wave (the signal) modifies another wave (the carrier).

**Bandwidth** is the capacity of a medium to carry data. Digital bandwidth measures the amount of data that can flow from one place to another in a given amount of time in kb/s, Mb/s, and Gb/s. A combination of factors determines the practical bandwidth of a network: The properties of the physical media, The technologies chosen

for signaling and detecting network signals.

**Throughput** is the measure of the transfer of bits across the media over a given period of time. Due to a number of factors, throughput usually does not match the specified bandwidth in physical layer implementations. Many factors influence throughput, including: The type and amount of traffic and Latency<sup>1</sup> between source and destination. *Throughput cannot be faster than the slowest link in the path from source to destination.*

There are three basic forms of network media: Copper cable (electrical pulses), fiber optic (light), and wireless (microwave transmissions).

### 1.2.2 Copper cable

Copper cable is inexpensive and easy to install but limited by distance and signal interference. There are two sources of signal interference:

- **EMI or RIF:** distort and corrupt the data signals; potential source: radio waves, electromagnetic devices, such as fluorescent lights or electric motors. To counter the negative effects of EMI and RFI, some types of copper cables are wrapped in metallic shielding and require proper grounding connections.
- **Crosstalk** is a disturbance caused by the electric or magnetic fields of a signal on one wire to the signal in an adjacent wire. To counter the negative effects of crosstalk, some types of copper cables have opposing circuit wire pairs twisted together.

There are three main types of copper media used in networking:

- **Unshielded Twisted-Pair (UTP):** These cables are used to interconnect nodes on a LAN and infrastructure devices such as PCs, switches, routers. UTP cabling, terminated with **RJ-45** connectors, consists of **four** pairs of color-coded wires that have been twisted together.
- **Shielded twisted-pair (STP)** provides better noise protection than UTP cabling.
- **Coaxial cable** contains two conductors that share the same axis.

Table 1.3: UTP Categories

Category	Speed
Cat3 Cable (UTP)	10Mb/s
Cat5	100 – 1000 Mb/s
Cat5e	1000 Mb/s
Cat6	1000 Mb/s – 10 Gb/s

Table 1.4: Copper cable types

UTP cable	Standard	Application
Straight-through	Both ends T568A or T568B	PC to switch, switch to router
Crossover	One end T568A, the other end T568B	switch to switch, router to router, PC to router
Rollover	Cisco proprietary	Connects a workstation serial port to a router console port

<sup>1</sup>Latency refers to the amount of time, to include delays, for data to travel from one given point to another.

### 1.2.3 Fiber optic

Optical fiber cable transmits data over longer distances and at higher bandwidths than any other networking media. Unlike copper wires, fiber-optic cable can transmit signals with less attenuation and is completely immune to EMI and RFI.

Table 1.5: Fiber Cable Components

Component	Description
Core	The core is actually the light transmission element at the center of the optical fiber. This core is typically silica or glass. Light pulses travel through the fiber core.
Cladding	It tends to act like a mirror by reflecting light back into the core of the fiber. This keeps light in the core as it travels down the fiber.
Buffer	Used to help shield the core and cladding from damage.
Strengthening material	Surrounds the buffer, prevents the fiber cable from being stretched when it is being pulled. The material used is often the same material used to produce bulletproof vests.
Jacket	a PVC jacket that protects the fiber against abrasion, moisture, and other contaminants.

Light pulses representing the transmitted data as bits on the media are generated by either Lasers or LEDs. Fiber-optic cables are broadly classified into two types:

- **Single-mode fiber (SMF)** consists of a very small core and uses *laser* to send a *single* ray of light. Popular in long-distance situations spanning hundreds of kilometers: long-haul telephony, cable TV, campus backbones .
- **Multimode fiber (MMF)** consists of a larger core and uses LED emitters to send light pulses. It provides bandwidth up to 10 Gb/s over link lengths of up to 550 meters.
- One of the highlighted differences between multimode and single-mode fiber is the amount of *dispersion*. Dispersion refers to the spreading out of a light pulse over time. The more dispersion there is, the greater the loss of signal strength.

### 1.2.4 Wireless

Wireless media provides the greatest mobility options of all media. Wireless does have some areas of concern, including Coverage area, Interference, Security, and Shared medium. There are many types of wireless media: WiFi (IEEE 802.11 standard), Bluetooth (IEEE 802.15 standard), and Wi Max (IEEE 802.16 Standard).

## 1.3 Data Link layer

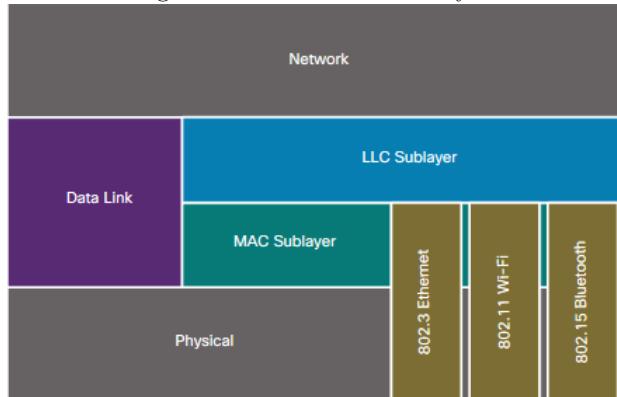
The PDU of Data Link layer is always **frame**. The data link layer is divided into two sublayers: Logical Link Control (LLC) and Media Access Control (MAC). See Figure 1.3.

### 1.3.1 Logical Link Control sublayer (LLC)

This upper sublayer communicates with the network layer. It places information in the frame that identifies which network layer protocol is being used for the frame. This information allows multiple Layer 3 protocols, such as IPv4 and IPv6, to utilize the same network interface and media.

LLC is implemented in software, and its implementation is independent of the hardware. In a computer, the LLC can be considered the driver software for the NIC. The NIC driver is a program that interacts directly with the hardware on the NIC to pass the data between the MAC sublayer and the physical media.

Figure 1.3: Data Link sublayers



### 1.3.2 Ethernet MAC sublayer

The Ethernet MAC sublayer has two primary responsibilities: Data encapsulation and Media Access Control.

The **data encapsulation** process includes frame assembly before transmission and frame disassembly upon reception of a frame. Data encapsulation provides three primary functions: Frame delimiting, Addressing, and Error detection.

**Media access control** is responsible for the placement of frames on the media and the removal of frames from the media. As its name implies, it controls access to the media. This sublayer communicates directly with the physical layer. The actual media access control method used depends on Topology and Media sharing.

**Multi-access networks** Ethernet LANs<sup>2</sup> and WLANs<sup>3</sup> are examples of a multiaccess network. At any one time, there may be a number of devices attempting to send and receive data using the same network media. Therefore, multi-access networks require rules to govern how devices share the physical media. Those rules, together, form a media access control method. There are two basic access control methods for shared media:

- **Contention-based access:** All nodes operating in half-duplex compete, only one device can send at a time. Ethernet LANs *using hubs* and WLANs are examples of this type of access control.
- **Controlled access:** Each node has its own time to use the medium, a device must wait its turn to access the medium. Legacy Token Ring LANs are an example of this type of access control.

**Note!** Ethernet LANs using *switches* do not use a contention-based system because the switch and the host NIC operate in full-duplex mode.

The **CSMA/CD** (Carrier Sense Multiple Access/Collision Detection) process is used in *half-duplex Ethernet LANs*. A PC's NIC needs to determine if anyone is transmitting on the medium. If it does not detect a carrier signal or receives transmissions from another device, it will assume the network is available to send. If another device wants to transmit, it must wait until the channel is clear. Additionally, CSMA/CD stops transmitting when congestion occurs. It also uses a random time to re-send a frame.

The **CSMA/CA** (Carrier Sense Multiple Access/Collision Avoidance) process is used in *WLAN*. CSMA/CA does not detect collisions but attempts to avoid them by waiting before transmitting. Each device that transmits includes the time duration that it needs for the transmission. All other wireless devices receive this information and know how long the medium will be unavailable.

<sup>2</sup>Various hosts connect to a single switch (or hub) using Ethernet cables.

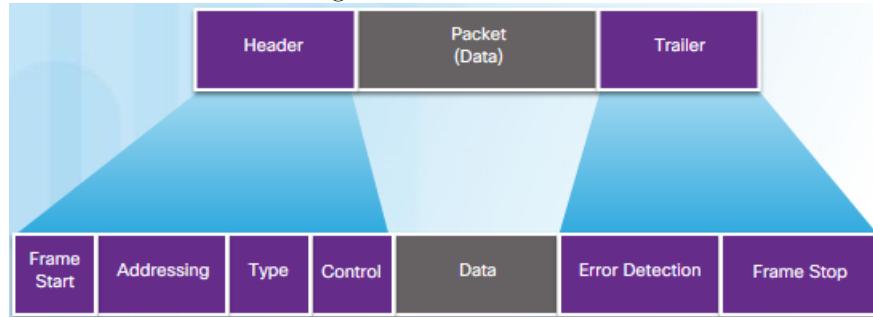
<sup>3</sup>Various hosts connect to an access point via radio transmissions

### 1.3.3 Frame

#### Generic Frame

The data link layer prepares a packet for transport across the local media by encapsulating it with a header and a trailer to create a frame. Each frame has three parts: Header, Data, and Trailer (Figure 1.4). The generic frame field types include:

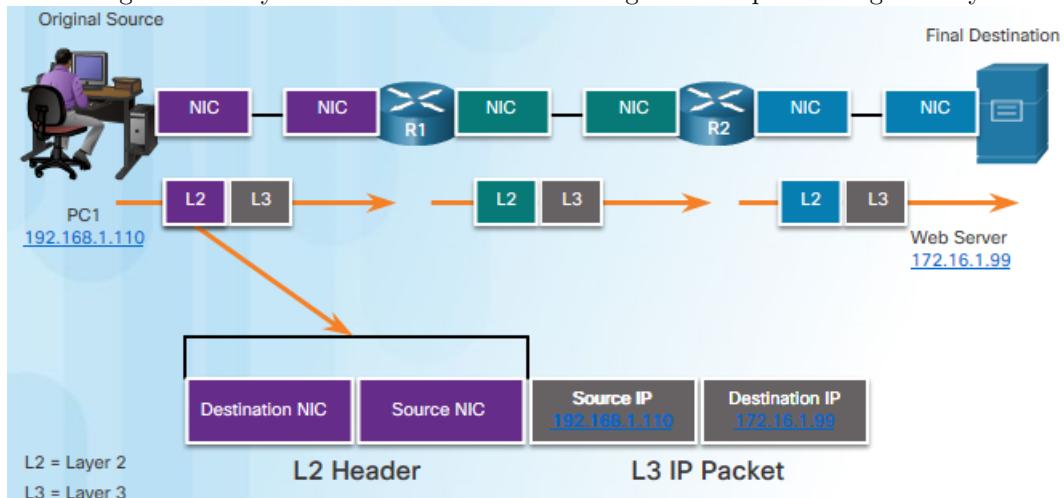
Figure 1.4: Frame fields



- **Frame start and stop indicator flags** identify the beginning and end limits of the frame.
- **Addressing** indicates the source and destination nodes on the media.
- **Type** identifies the Layer 3 protocol in the data field.
- **Control** identifies special flow control services such as quality of service (QoS).
- **Data** contains the frame payload
- **Error Detection**

The trailer is used to determine if the frame arrived without error. This process is called error detection. Cyclic Redundancy Check (CRC) value is placed in the Frame Check Sequence (FCS) field to represent the contents of the frame. In the Ethernet trailer, the FCS checks transmission errors.

Figure 1.5: Layer 2 Data Link addresses change at each point along the way



As the IP packet travels from host-to-router, router-to-router, and finally router-to-host, at each point along the way the IP packet is encapsulated in a new data link frame (Figure 1.5). Each data link frame contains the source data link address of the NIC card sending the frame and the destination data link address of the NIC card receiving the frame. Remember that the data link layer address is only used for local subnet delivery.

### Ethernet Frame

**Size** The minimum Ethernet frame size is 64 bytes and the maximum is 1518 bytes<sup>4</sup>. Any frame less than 64 bytes in length is called a *collision fragment* or *runt frame*. Frames with more than 1500 bytes of data are called *jumbo frames* or *baby giant frames*. If the size of a transmitted frame is less than the minimum or greater than the maximum, the receiving device drops the frame.

**Frame fields** The *Preamble* (7 bytes) and *Start Frame Delimiter* (1 byte) fields are used for synchronization between the sending and receiving devices. The *Type* field (2 bytes) identifies the upper layer protocol in hexadecimal: IPv4 = 0x800, IPv6 = 0x86DD, ARP = 0x806.

**Ethernet MAC addresses** (6 bytes) are made up of two parts: vendor code OUI (3 bytes) assigned by IEEE and device identifier (3 bytes). When the computer starts up, the first thing the NIC does is copy the MAC address from ROM into RAM. **Broadcast MAC** address is **FF-FF-FF-FF-FF-FF** (twelve F letters). The **multicast MAC** address associated with an IPv4 multicast address is a special value that begins with **01-00-5E** in hexadecimal. The remaining portion of the multicast MAC address is created by converting the lower 24 bits of the IP multicast group address into 6 hexadecimal characters. For an **IPv6** address, the multicast MAC address begins with **33-33**.

### 1.3.4 ARP

To determine the destination MAC address, the device uses ARP. ARP provides two basic functions: Resolving IPv4 addresses to MAC addresses, Maintaining a table of mappings.

When a packet is sent to the data link layer to be encapsulated into an Ethernet frame, the device refers to a table in its memory to find the MAC address that is mapped to the IPv4 address. This table is called the ARP table or the ARP cache. The ARP table is stored in the RAM of the device.

The sending device will search its ARP table for a destination IPv4 address and a corresponding MAC address. If the destination IPv4 address is on the *same* network as the source IPv4 address, the device will search the ARP table for the destination IPv4 address. Otherwise, the device will search the ARP table for the IPv4 address of the default gateway.

For each device, an ARP cache timer removes ARP entries that have not been used for a specified period of time. The times differ depending on the device's operating system. For example, Windows store ARP cache entries for 2 minutes.

**ARP request** If there is no entry is found in ARP table for a particular IPv4 address, then the device sends an ARP request. ARP messages are encapsulated directly within an Ethernet frame. There is no IPv4 header (Figure 1.6). Destination MAC address is a broadcast address. ARP messages have a type field of **0x806**.

Table 1.6: ARP request message

Destination MAC	Source MAC	Target IPv4	Target MAC
FF-FF	00-0A	192.68.1.5	—

**ARP reply** Only the device with an IPv4 address associated with the target IPv4 address in the ARP request will respond with an ARP reply. Only the device that originally sent the ARP request will receive the unicast ARP reply. The ARP reply is encapsulated in an Ethernet frame (Table 1.7).

**ARP broadcasts** If a large number of devices were to be powered up, and all start sending ARP requests, there could be some reduction in performance for a short period of time.

<sup>4</sup> The Preamble field is not included when describing the size of a frame.

Table 1.7: ARP request message

Destination MAC	Source MAC	Sender IPv4	Sender MAC
00-0A	00-0B	192.68.1.5	00-0B

**ARP spoofing** is a technique used by an attacker to reply to an ARP request for an IPv4 address belonging to another device, such as the default gateway. The attacker sends an ARP reply with its own MAC address. The receiver of the ARP reply will add the wrong MAC address to its ARP table and send these packets to the attacker.

## 1.4 Network layer

Network Layer PDU Is an **Packet**.

### 1.4.1 Introduction

The network layer has four basic processes: Addressing end devices, routing, encapsulation, de-encapsulation. There are several network layer protocols in existence, but there are only two network layer protocols that are commonly implemented: **IPv4** and **IPv6**.

IP encapsulates the transport layer segment or other data by adding an IP header. This header remains the same from the time the packet leaves the source host until it arrives at the destination host.

The protocols in network layer were not designed to track and manage the flow of packets. The basic characteristics of IP are

- **Connectionless:** no dedicated end-to-end connection is created before data is sent.
- **Best Effort:** The IP protocol does not guarantee that all packets that are received. Furthermore, IP is unreliable which means that IP does not have the capability to manage and recover from undelivered or corrupt packets
- **Media Independent:** Operation is independent of the medium (i.e., copper, fiber optic, or wireless) carrying the data.

There is, however, one major characteristic of the media that the network layer considers: **MTU** (maximum transmission unit). The data link layer passes the MTU value up to the network layer. The network layer then determines how large packets can be.

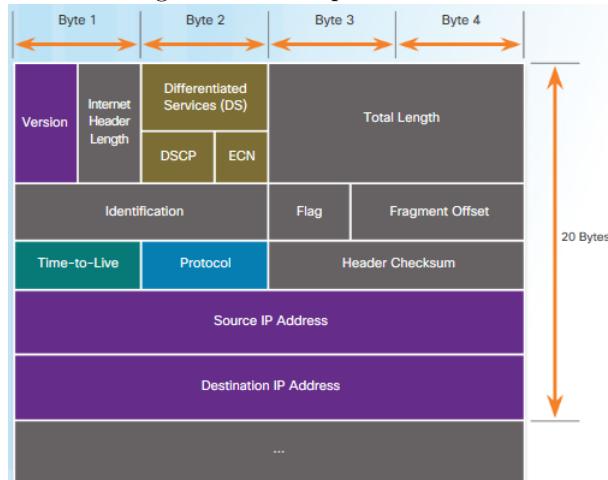
Sometimes, a IPv4 router must split up a packet when forwarding it from one medium to another medium with a smaller MTU. This process is called **fragmentation**. Unlike IPv4, IPv6-enabled routers do not fragment packets.

### 1.4.2 Packet header

Significant fields in the **IPv4** header include (Figure 1.6):

- **Version:** is set to **0100** that identifies this as an IP version 4 packet.
- **DiffServ:** used by QoS service to determine the priority.
- **TTL:** limit the lifetime of a packet. The value is decreased by one each time the packet is processed by a router. If the TTL field decrements to zero, the router discards the packet and sends an *ICMP Time Exceeded* message to the source IP address.
- **Protocol:** identify the next level protocol. Common values include **ICMP (1)**, **TCP (6)**, and **UDP (17)**.
- **Source IPv6 Address**
- **Destination IPv6 Address**

Figure 1.6: IPv4 packet header



Significant fields in the **IPv6** header include (Figure ??):

- **Version:** is set to **0110** that identifies this as an IP version 6 packet.
- **Traffic class:** equivalent to the IPv4 DiffServ field.
- **Payload Length:** indicates the length of the data portion of the packet.
- **Hop Limit:** equivalent to the IPv4 TTL field.
- **Next Header:** equivalent to the IPv4 Protocol field.

### 1.4.3 Routing table

When a router interface (g0/0) is configured with an IPv4 address (192.168.10.1), a subnet mask (255.255.255.0), and is activated, the following two routing table entries are automatically created:

```
C 192.168.10.0/24 is directly connected, GigabitEthernet0/0
L 192.168.10.1/32 is directly connected, GigabitEthernet0/0
```

The letter **C** identifies a directly connected *network*. The letter **L** shows that this is a local interface and its IP address is 192.168.10.1.

The routing table also stores information about remote network. For example, the entry for the remote network 10.1.1.0 is as follows:

```
D 10.1.1.0/24 [90/2170112] via 209.165.200.226, 00:00:09, Serial0/0/0
```

The details of the remote network routing table entry are explained as follow:

- **D** – Identifies how the network was learned by the router. Common route sources include **S** (static route), **D** (EIGRP), **O** (OSPF).
- **10.1.1.0/24** – Identifies the destination network.
- **90** – Identifies the AD of the route.
- **2170112** – Identifies the metric of the route.
- **209.165.200.226** – Identifies the IP address of the router to forward the packet.
- **00:00:09** – Router timestamp
- **Serial0/0/0** – Identifies the exit interface to use to forward a packet

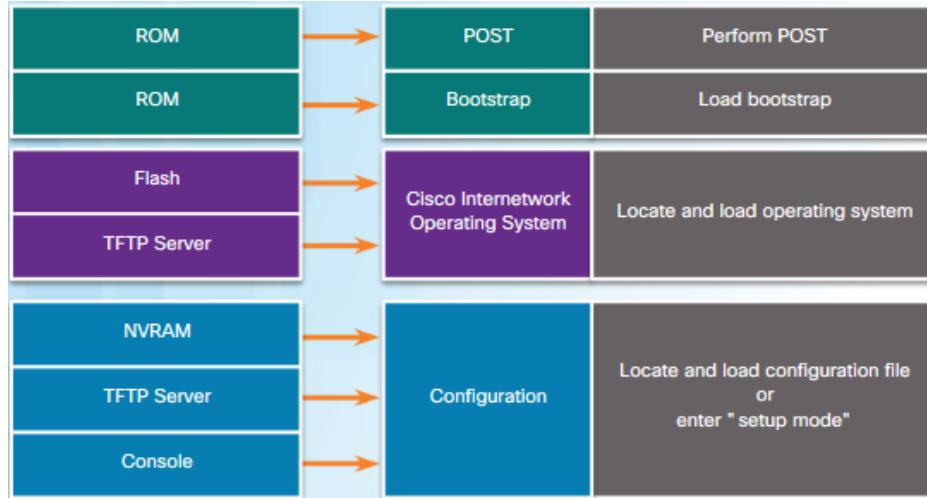
#### 1.4.4 Router Boot-up

This topic will introduce the structure of Cisco routers and how they boots up.

A Cisco router has four types of memory:

- **RAM:** This is volatile memory that stores: IOS image, Running configuration file, Routing table, ARP cache, and Packet buffer.
- **ROM:** This non-volatile memory is used to store Bootstrap program, Power-on self-test (POST), and Limited IOS (backup IOS).
- **NVRAM:** This non-volatile memory is used to store Startup configuration file.
- **Flash:** This non-volatile computer memory used as permanent storage for the IOS and other system-related files (log, HTML files, etc.) When a router is rebooted, the IOS is copied from flash into RAM.

Figure 1.7: Cisco router bootup process



There are three major phases to the bootup process, as shown in Figure 1.7.

1. **POST process and Bootstrap Program:** During the Power-On Self-Test (POST), the router executes diagnostics on hardware components (CPU, RAM, NVRAM, etc.). After the POST, the bootstrap program is copied from ROM into RAM.
2. **Loading Cisco IOS:** The IOS image is copied from Flash memory into RAM. If the image is not located in flash, then the router may look for it using a TFTP server. If a full IOS image cannot be located, a Limited IOS (in ROM) is copied into RAM, which can be used to diagnose problems and transfer a full IOS into Flash memory.
3. **Loading the Startup configuration File:** The bootstrap program then copies the startup configuration file from NVRAM into RAM. This becomes the running configuration. If the startup configuration file does not exist in NVRAM, the router may be configured to search for a TFTP server. If a TFTP server is not found, then the router displays the setup mode prompt.

**Note!** Setup mode is not used in this course to configure the router. When prompted to enter setup mode, always answer no. If you answer yes and enter setup mode, press Ctrl+C at any time to terminate the setup process.

Use `show version` to check the version of the Cisco IOS, the version of the bootstrap program, and information about the hardware configuration, including the amount of system memory.

Table 1.8: Initial router configuration

Task	Commands
Configure the device name	hostname BranchRouter
Secure user EXEC mode	line console 0 password cisco12345 login
Secure remote Telnet/SSH access	line vty 0 4 password cisco12345 login
Secure privilege EXEC mode	Router(config)# enable secret password
Secure all passwords	Router(config)# service password-encryption
Provide legal notification	Router(config)# banner motd "message"
Save the configuration	Router# copy run start
Configure the interface	interface s0/0/0 ip address 192.168.100.1 255.255.255.0 description Connect to R1 no shutdown
Verify interface configuration	show interface show ip interface show ip interface brief show ip route

#### 1.4.5 Router configuration

### 1.5 Transport layer

The PDU of transport layer is either **segment** or **datagram**. The transport layer is responsible for:

- Establishing a temporary communication session between two applications and delivering data between them.
- Tracking individual conversations
- Segmenting data and Reassembling segments
- To pass data streams to the proper applications, transport layer assigns each application an identifier called **a port number**.

#### 1.5.1 TCP

TCP is considered a reliable, full-featured transport layer protocol, which ensures that all of the data arrives at the destination. However, this requires additional fields in the TCP header which increases the size of the packet and also increases delay. With TCP, there are three basic operations of reliability:

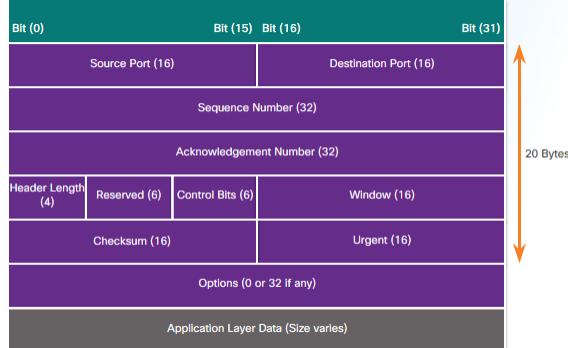
- Numbering and tracking data segments transmitted
- Acknowledging received data
- Retransmitting any unacknowledged data

TCP has the following features:

- **Connection-oriented:** TCP negotiates and establishes a virtual connection between source and destination devices prior to forwarding any traffic.
- **Reliable:** TCP ensures that each segment arrives at the destination.

- **Stateful protocol:** A stateful protocol is a protocol that keeps track of the state of the communication session.
- **Same-Order Delivery:** Because networks may provide multiple routes that can have different transmission rates, data can arrive in the wrong order. By numbering and sequencing the segments, TCP can ensure that these segments are reassembled into the proper order.
- **Flow control:** When TCP is aware that these resources are overtaxed, it can request that the sending application reduce the rate of data flow. To accomplish this, the TCP header includes a field called the *window size*.

Figure 1.8: TCP segment



Each TCP segment has 20 bytes of overhead (Figure 1.8) in the header encapsulating the application layer data:

- **Source Port and Destination Port**
- **Sequence number** — Used for data reassembly purposes. During session setup, an **initial sequence number (ISN)** is set. ISN is the starting value of Sequence number. As data is transmitted during the session, the sequence number is incremented.
- **Acknowledgement number** — Indicates the data that has been received. Acknowledgement number sent back to the source to indicate the next byte that the receiver expects to receive. This is called *expectational acknowledgement*. When TCP at the source host has not received an acknowledgement after a predetermined amount of time, it returns to the last ACK number received and retransmits the data from that point forward.
- **Header length** — Indicates the length of the TCP segment header.
- **Reserved** — This field is reserved for the future.
- **Control bits** — Known as flags, these identify bit codes that indicate the purpose and function of the TCP segment.
- **Window size** — Indicates the number of bytes that can be accepted at one time. The window size is included in every TCP segment, so the destination can modify the window size at any time depending on buffer availability. The initial window size is agreed upon TCP three-way handshake.
- **Checksum** — Used for error checking of the segment header and data.
- **Urgent** — Indicates if data is urgent.

A TCP connection is established in three-way handshake:

1. The client requests a client-to-server communication session with the server
2. The server acknowledges the client-to-server communication session and requests a server-to-client communication session
3. The client acknowledges the server-to-client communication session

The TCP connect is terminated in **two** two-way handshakes with four exchanges:

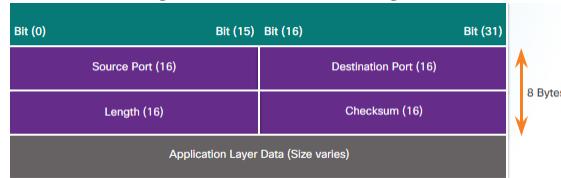
1. When the client has no more data to send in the stream, it sends a segment with the FIN flag set
2. The server sends an ACK to acknowledge the receipt of the FIN to terminate the session from client to server.  
So step 1 and 2 is the first two-way handshake
3. The server sends a FIN to the client to terminate the server-to-client session
4. The client responds with an ACK to acknowledge the FIN from the server

TCP is a **full-duplex** protocol, where each connection represents two one-way communication sessions. To establish the connection, the hosts perform a **three-way handshake**. Control bits in the TCP header indicate the progress and status of the connection.

### 1.5.2 UDP

UDP is a lightweight transport protocol that offers the same data segmentation and reassembly as TCP, but without TCP reliability and flow control. It is known as a **best-effort** delivery protocol. UDP is also a **stateless** protocol, meaning neither the client, nor the server, is obligated to keep track of the state of the communication session.

Figure 1.9: UDP datagram



UDP has no way to reorder the datagrams into their transmission order. Therefore, UDP simply reassembles the data in the order that it was received and forwards it to the application.

The pieces of communication in UDP are called datagrams. These datagrams are sent as best-effort by the transport layer protocol. UDP has a low overhead of 8 bytes (Figure 1.9).

There are three types of applications that are best suited for UDP:

- Live video and multimedia applications (e.g. VoIP, live streaming video)
- Simple request and reply applications (e.g. DNS<sup>5</sup>, DHCP)
- Applications that handle reliability themselves (SNMP<sup>6</sup>, TFTP)

### 1.5.3 Port number

The source port number is associated with the originating application on the local host. The destination port number is associated with the destination application on the remote host. Each application process running on the server is configured to use a port number. An individual server cannot have two services assigned to the same port number within the same transport layer services.

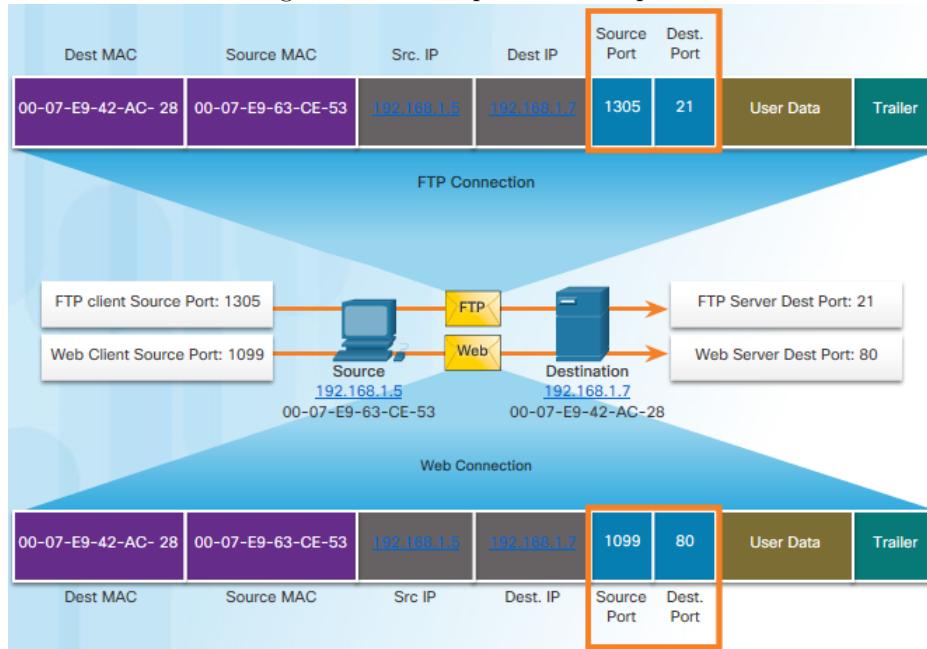
The combination of the source IP address and source port number, or the destination IP address and destination port number is known as a socket. The socket is used to identify the server and service being requested by the client. In figure 1.10, the client socket is 192.168.1.5:1099 with 1099 representing the source number, the socket on the web server is 192.168.1.7:80. Together, these two sockets combine to form a socket pair: 192.168.1.5:1099, 192.168.1.7:80

There are different types of port numbers:

<sup>5</sup>DNS can also use TCP if the DNS request or DNS response is more than 512 bytes

<sup>6</sup>SNMP uses UDP by default, but it can also use TCP

Figure 1.10: Examples of socket pairs



- **Well-known Ports (0 – 1023):** These numbers are used for popular applications such as FTP, HTTP, TFTP, etc.
- **Registered Ports (1024 – 49151):** These port numbers are assigned by IANA to a requesting entity to use with specific processes or applications.
- **Dynamic or Private Ports (49152 – 65535):** These are assigned dynamically by the client's OS to identify the application during communication.

## 1.6 Application layer

The PDU of application layer is **data**. Application layer protocols are used to exchange data between programs running on the source and destination hosts. The upper three layers of the OSI model (application, presentation, and session) define functions of the single TCP/IP application layer.

### 1.6.1 Presentation and Session Layer

The presentation layer has three primary functions: Formatting, Compressing, and Encrypting. Specifically, it formats data for the application layer, and it sets standards for file formats (QuickTime, MPEG, GIF, PNG, JPEG, etc.).

Functions at the session layer create and maintain dialogs between source and destination applications. The session layer handles the exchange of information to initiate dialogs, keep them active, and to restart sessions that are disrupted or idle for a long period of time.

### 1.6.2 Peer-to-peer network

In the peer-to-peer (P2P) networking model, the data is accessed from a peer device without the use of a dedicated server. The P2P network model involves two parts: P2P networks and P2P applications. Every connected end device (known as a peer) can function as both a server and a client.

Table 1.9: Protocol port number

Port number	Protocol	Application
20	TCP	FTP (data)
21	TCP	FTP (client)
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP, TCP	DNS
67, 68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP
443	TCP	HTTPS
110	TCP	POP3
143	TCP	IMAP
161	UDP	SNMP

With P2P applications, each computer in the network running the application can act as a client or a server for the other computers in the network running the application. Common P2P networks include: eDonkey, G2, BitTorrent, Bitcoin. Some P2P applications are based on the Gnutella protocol, where each user shares whole files with other users.

### 1.6.3 HTTP

When a web address or URL (uniform resource locator) is typed into a web browser, the web browser establishes a connection to the web service running on the server using the HTTP protocol. URLs and URIs (Uniform Resource Identifier) are the names most people associate with web addresses.

HTTP is a request/response protocol. When a client sends a request to a web server, HTTP specifies the message types used for that communication. The three common message types are GET (A client request for data), POST (Upload data files), and PUT (Upload resources or content).

**HTTPS** uses the same client request-server response process as HTTP, but the data stream is encrypted with Secure Socket Layer (SSL) before being transported across the network.

### 1.6.4 Email

Email is a *store-and-forward* method of sending, storing, and retrieving electronic messages across a network. Email messages are stored in databases on mail servers. An email client does not communicate directly with another email client when sending email. Instead, both clients rely on the mail server to transport messages.

Email supports three separate protocols for operation: SMTP, POP3, IMAP. The application layer process that sends mail uses SMTP. A client retrieves email using one of the two application layer protocols: POP or IMAP.

SMTP port is **25**.

When the server receives the message, it either places the message in a local account (if the recipient is local) or forwards the message to another mail server for delivery. The destination email server may not be online or may be busy when email messages are sent. Therefore, SMTP spools messages to be sent at a later time. Periodically, the

server checks the queue for messages and attempts to send them again. If the message is still not delivered after a predetermined expiration time, it is returned to the sender as undeliverable.

**POP3** is used to retrieve mail from a mail server. With POP, by default, mail is downloaded from the server to the client and then deleted on the server.

**IMAP** is another protocol that describes a method to retrieve email messages. Unlike POP, when the user connects to an IMAP-capable server, copies of the messages are downloaded to the client application. The original messages are kept on the server until manually deleted. With IMAP, users can create a file hierarchy on the server to organize and store mail. That file structure is duplicated on the email client as well. When a user decides to delete a message, the server synchronizes that action and deletes the message from the server.

### 1.6.5 DNS

In data networks, devices are labeled with numeric IP addresses to send and receive data over networks. Domain names were created to convert the numeric address into a simple, recognizable name. For example, a domain name, such as `http://www.cisco.com`, is much easier for people to remember than 198.133.219.25, which is the actual numeric address for this server.

The DNS (Domain Name Service) defines an automated service that matches resource names with the required numeric network address. The steps involved in DNS resolution are as follows:

1. User enters a fully qualified domain name (FQDN).
2. The client computer sends a DNS query to the DNS server requesting an IP address to match the FQDN
3. The DNS server resolve the FQDN to an IP address in the DNS server database
4. The DNS server sends back a response to the client with the IP address for the FQDN

The DNS server stores different types of resource records used to resolve names. These records contain the name, address, and type of record. Some of these record types are:

- A – An end device IPv4 address
- NS – An authoritative name server
- AAAA – An end device IPv6 address (pronounced quad-A)
- MX – A mail exchange record

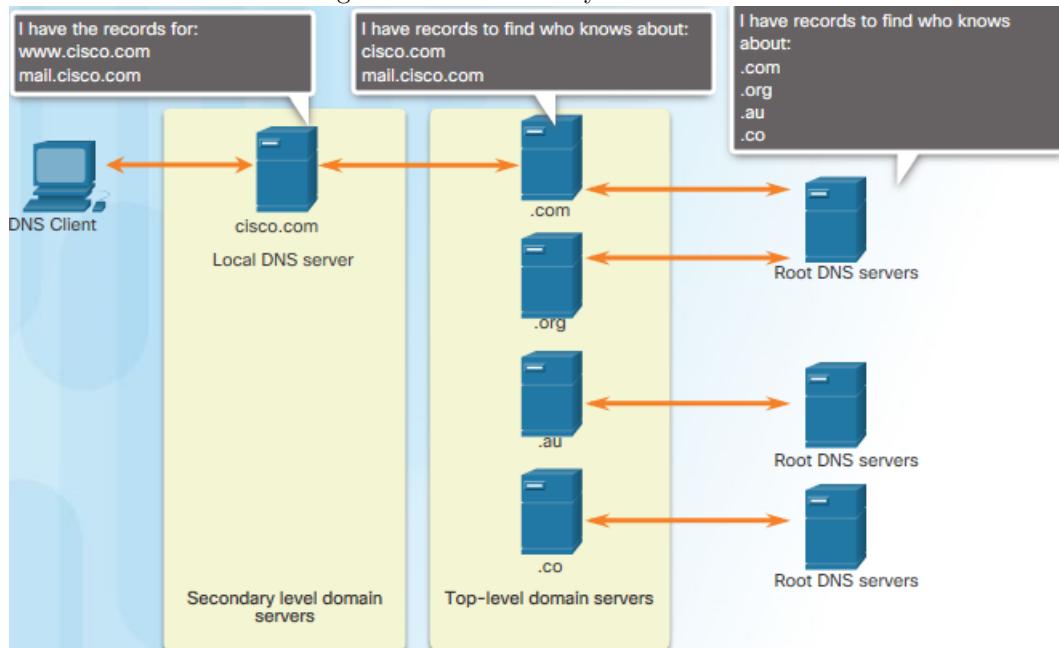
When a client makes a query, the server's DNS process first looks at its own records to resolve the name. If it is unable to resolve the name using its stored records, it contacts other servers to resolve the name. After a match is found and returned to the original requesting server, the server temporarily stores the numbered address in the event that the same name is requested again. The DNS Client service on Windows PCs also stores previously resolved names in memory. The `ipconfig /displaydns` command displays all of the cached DNS entries.

The DNS protocol uses a hierarchical system to create a database to provide name resolution (Figure 1.11). The naming structure is broken down into small, manageable zones. Each DNS server maintains a specific database file and is only responsible for managing name-to-IP mappings for that small portion of the entire DNS structure.

The different top-level domains represent either the type of organization or the country of origin. Examples of top-level domains are: .com (business or industry), .org (non-profit organization), .fi (Finland), etc.

Computer operating systems also have a utility called `nslookup` that allows the user to manually query the name servers to resolve a given host name. This utility can also be used to troubleshoot name resolution issues and to verify the current status of the name servers. When the `nslookup` command is issued, the default DNS server configured for your host is displayed.

Figure 1.11: A hierarchy of DNS server



### 1.6.6 FTP

FTP transfers data between a client and a server. To successfully transfer data, FTP requires two connections between the client and the server, one for commands and replies, the other for the actual file transfer:

- The **client** establishes the first connection to the server for control traffic using **TCP port 21**, consisting of client *commands* and server *replies*.
- The client establishes the second connection to the server for the *actual data transfer* using **TCP port 20**. This connection is created every time there is data to be transferred. The data transfer can happen in either direction. The client can download (pull) data from the server, or the client can upload (push) data to the server.

### 1.6.7 SMB

The Server Message Block (SMB) is a client/server file sharing protocol that describes the structure of shared network resources, such as directories, files, printers, and serial ports. It is a request-response protocol. SMB messages can

- Start, authenticate, and terminate sessions
- Control file and printer access
- Allow an application to send or receive messages to or from another device

Microsoft support SMB file-sharing. Unlike the file sharing supported by FTP, clients establish a long-term connection to servers. After the connection is established, the user of the client can access the resources on the server as if the resource is local to the client host. The LINUX and UNIX operating systems also provide a method of sharing resources with Microsoft networks using a version of SMB called *SAMBA*.



# Chapter 2

## LAN Design

### 2.1 Hierarchical Design Model

A hierarchical LAN design includes the following three layers, as shown in Figure 2.1:

- **Access layer** provides endpoints and users direct access to the network
- **Distribution layer** aggregates access layers and provides connectivity to services.
- **Core layer** provides connectivity between distribution layers for large LAN environments.

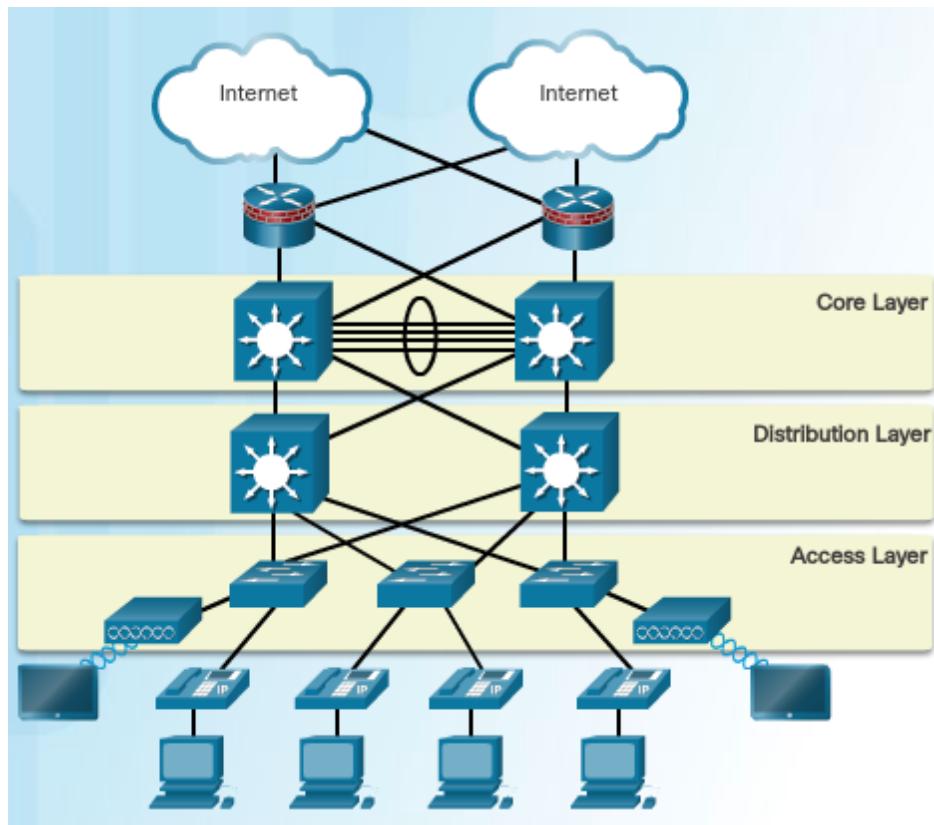


Figure 2.1: Three-layer hierarchical design model

Even though the hierarchical model has three layers, some smaller enterprise networks may implement a two-tier hierarchical design. In a two-tier hierarchical design, the core and distribution layers are collapsed into one layer, as shown in Figure 2.2.

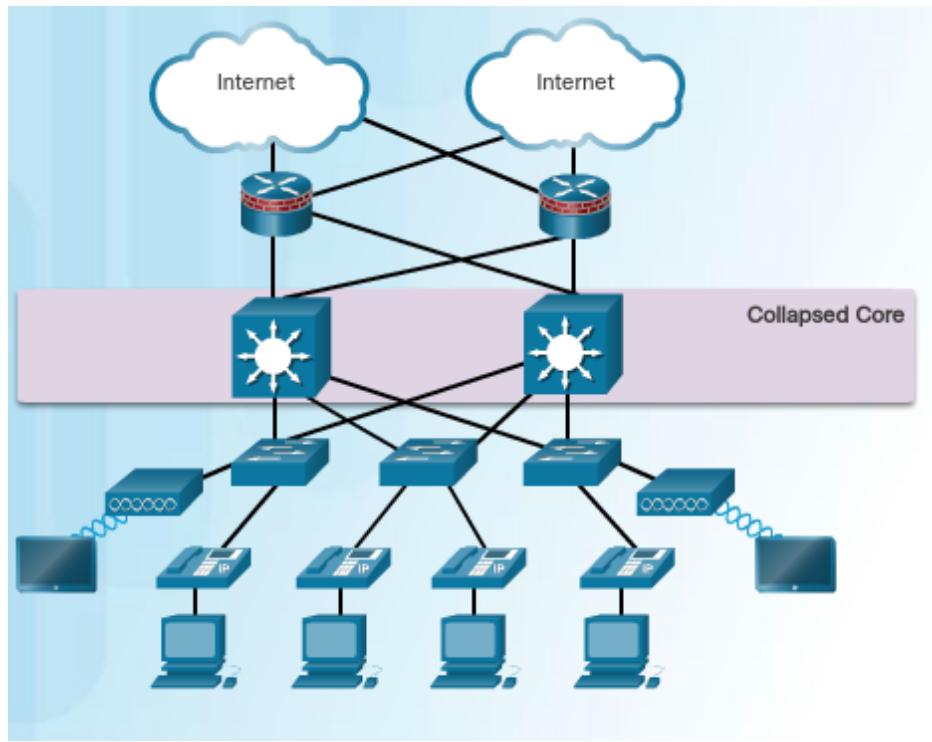


Figure 2.2: Collapsed Core

## 2.2 Expanding the network

One method of implementing **Redundancy** is by installing duplicate equipment and providing failover services for critical devices. Another method of implementing redundancy is redundant paths.

A **failure domain** is the area of a network that is impacted when a critical device or network service experiences problems. The use of redundant links and reliable enterprise-class equipment minimize the chance of disruption in a network. Smaller failure domains reduce the impact of a failure on company productivity.

To minimize failure domain, routers or multilayer switches, are usually deployed in pairs, with access layer switches evenly divided between them. This configuration is referred to as a **switch block**. Each switch block acts independently of the others.

**Increasing Bandwidth:** Link aggregation allows an administrator to increase the amount of bandwidth between devices by creating one logical link made up of several physical links. The EtherChannel is seen as one logical link using an EtherChannel interface. Most configuration tasks are done on the EtherChannel interface, instead of on each individual port, ensuring configuration consistency throughout the links.

**Wireless connection:** To communicate wirelessly, end devices require a wireless NIC that incorporates a radio transmitter/receiver and the required software driver to make it operational. Additionally, a wireless router or a wireless access point (AP) is required for users to connect.

**Fine-tuning Routing Protocols:** Advanced routing protocols, such as OSPF and EIGRP are used in large networks. Link-state routing protocols such as Open Shortest Path First (OSPF) works well for larger hierarchical networks where fast convergence is important. Another popular routing protocol for larger networks is Enhanced Interior Gateway Routing Protocol (EIGRP). Cisco developed EIGRP as a proprietary distance vector routing protocol with enhanced capabilities.

## 2.3 Selecting network devices

### 2.3.1 Switch hardwares

There are five categories of switches for enterprise networks:

- **Campus LAN Switches** – To scale network performance in an enterprise LAN, there are core, distribution, access, and compact switches.
- **Cloud-Managed Switches** – The Cisco **Meraki** cloud-managed access switches enable virtual stacking of switches. They monitor and configure thousands of switch ports over the web, without the intervention of onsite IT staff.
- **Data Center Switches** – A data center should be built based on switches that promote infrastructure scalability, operational continuity, and transport flexibility. The data center switch platforms include the Cisco Nexus Series switches and the **Cisco Catalyst 6500 Series** switches.
- **Service Provider Switches** – Service provider switches fall under two categories: aggregation switches and Ethernet access switches. **Aggregation switches** are carrier-grade Ethernet switches that aggregate traffic at the edge of a network. **Ethernet access switches** feature application intelligence, unified services, virtualization, integrated security, and simplified management.
- **Virtual Networking** – Cisco **Nexus** virtual networking switch platforms provide secure multi-tenant services by adding virtualization intelligence technology to the data center network.

There are some terminologies that an administrator to be able to choose the right switch platform:

- **Port density** is the number of ports available on a single switch.
- **Forwarding rates** define the processing capabilities of a switch by rating how much data the switch can process per second.
- **Wire speed** is the data rate that each Ethernet port on the switch is capable of attaining. Data rates can be 100 Mb/s, 1 Gb/s, 10 Gb/s, or 100 Gb/s.
- **PoE** (Power over Ethernet) allows the switch to deliver power to a device over the existing Ethernet cabling. This feature can be used by IP phones and some wireless access points.
- **Multilayer switches**, so called Layer-3 switches, are typically deployed in the core and distribution layers of an organization's switched network

### Router hardware

There are three categories of routers:

- **Branch Routers** – Branch routers optimize branch services on a single platform while delivering an optimal application experience across branch and WAN infrastructures.
- **Network Edge Routers** – Network edge routers enable the network edge to deliver high-performance, highly secure, and reliable services that unite campus, data center, and branch networks.
- **Service Provider Routers** – Service provider routers differentiate the service portfolio and increase revenues by delivering end-to-end scalable solutions and subscriber-aware services.



# Chapter 3

## WAN concepts

### 3.1 WAN Technologies overview

#### 3.1.1 Topology

A WAN operates beyond the geographic scope of a LAN. WANs are used to interconnect the enterprise LAN to remote LANs in branch sites and telecommuter sites. A WAN is owned by a service provider whereas a LAN is typically owned by an organization. An organization must pay a fee to use the WAN service provider's network services to connect remote sites.

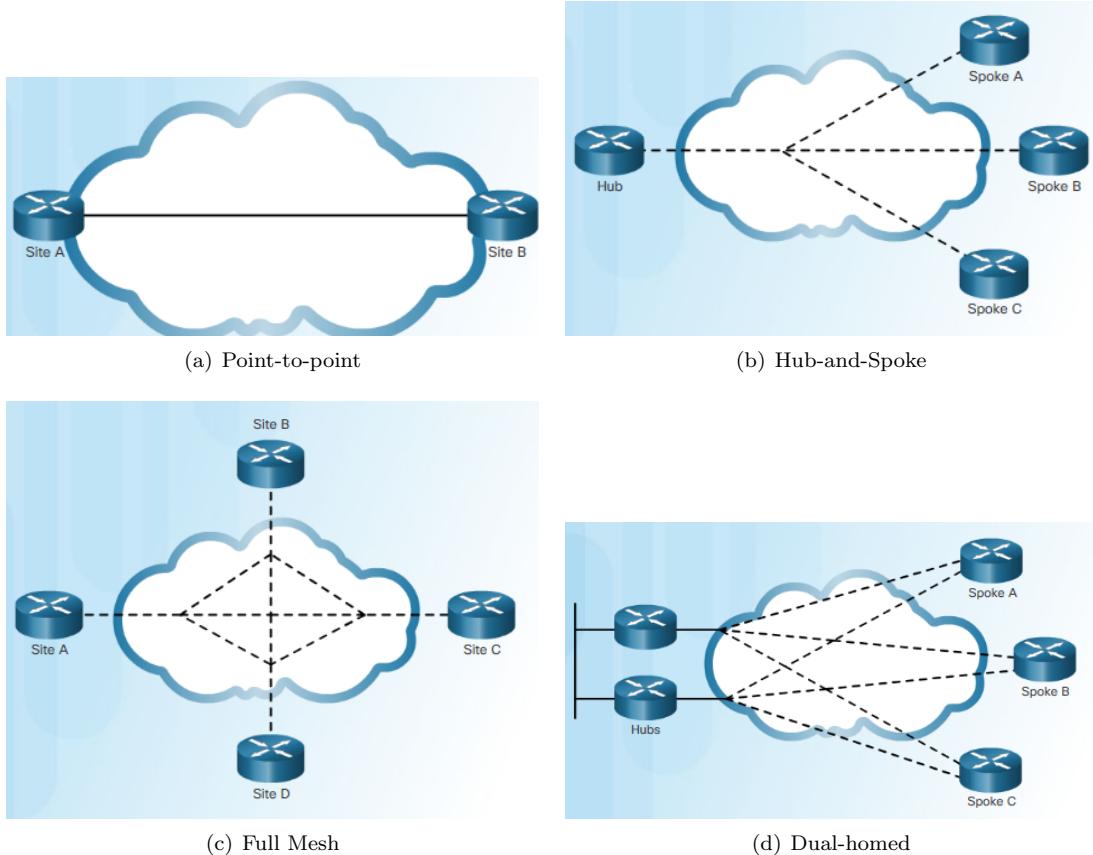


Figure 3.1: Four common WAN topologies

**Point-to-Point topology** employs a point-to-point circuit between two endpoints (Figure 3.1(a)). Typically involves a dedicated leased-line connection such as a T1/E1 line.

**Hub-and-Spoke** An example of a single-homed topology. Applicable when a private network connection between multiple sites is required. A single interface to the hub can be shared by all spoke circuits (Figure 3.1(b)).

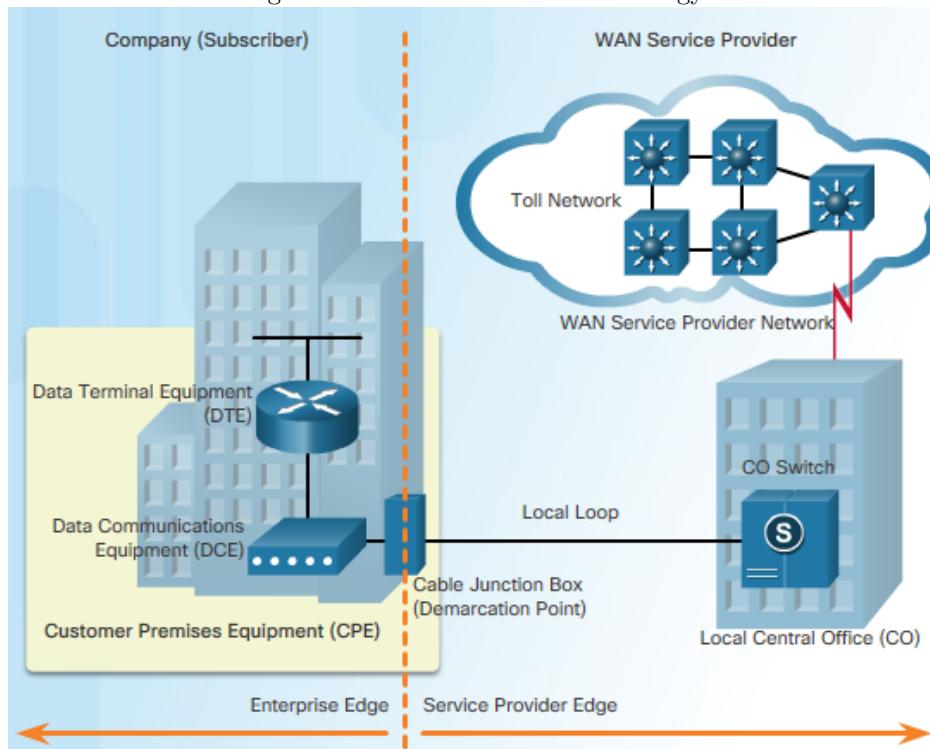
**Full Mesh** A disadvantage of the hub-and-spoke topology is that all communication has to go through the hub. With a full mesh topology using virtual circuits, any site can communicate directly with any other site (Figure 3.1(c)). A disadvantage is the large number of virtual circuits that need to be configured and maintained.

**Dual-homed Topology** Provides redundancy and load balancing, however more expensive to implement than single-homed topologies(Figure 3.1(d)). Requires additional networking hardware including routers and switches. More difficult to implement since they require complex configurations.

### 3.1.2 Terminology

WAN operations focus primarily on the Layer 1 and 2 of the OSI Model. One primary difference between a WAN and a LAN is that a company must subscribe to an outside WAN service provider to use WAN carrier network services.

Figure 3.2: Common WAN terminology

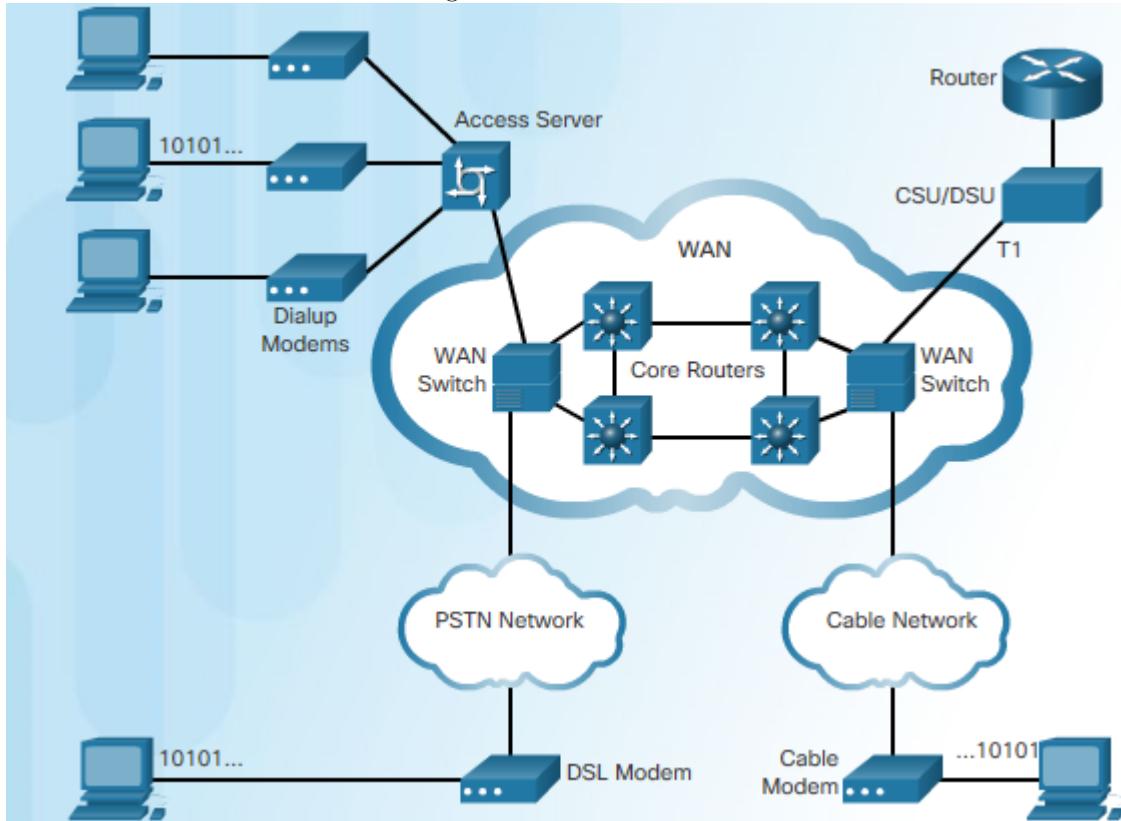


Terminology commonly used to describe WAN connections (Figure 3.2):

- **Customer Premises Equipment (CPE)** Consists of devices and inside wiring located on the enterprise edge connecting to a carrier.
- **Central Office (CO)** is the local service provider facility that connects the CPE to the provider network.
- **Local Loop (last mile)** is the actual copper or fiber cable that connects the CPE to the CO.
- **Data Terminal Equipment (DTE)** is usually a router that pass the data from a customer network to DCE.

- **Data Communications Equipment (DCE)** is usually a modem that puts data on the local loop by converting digital signals into analog signals. It connects subscribers to the ISP provider.
- **Demarcation Point** is a point established in a building to separate customer equipment from service provider equipment. It is the place where the responsibility for the connection changes from the user to the service provider.
- **Toll network** consists of the longhaul, all-digital, fiber-optic communications lines and other equipment inside the WAN provider network.

Figure 3.3: WAN devices



There are many types of devices that are specific to WAN environments (Figure 3.3):

- **Dialup modem** converts (modulates) the digital signals (produced by a computer) into analog signals (voice frequencies).
- **Broadband modem** converts the digital signals into analog signals transferred via high-speed DSL or cable Internet service.
- **Access server** controls and coordinates dialup modem, dial-in and dial-out user communications.
- **CSU/DSU** is only used for **Leased line**. The CSU provides termination for the digital signal and ensures connection integrity through error correction and line monitoring. The DSU converts line frames into frames that the LAN can interpret and vice versa.

WAN technologies are either circuit-switched or packet-switched:

**Circuit Switching** dynamically establishes a *dedicated virtual connection* for voice or data between a sender and a receiver. Communication can't start until the connection is established through the service provider network. The two most common types of circuit-switched WAN technologies are **PSTN** and **ISDN**.

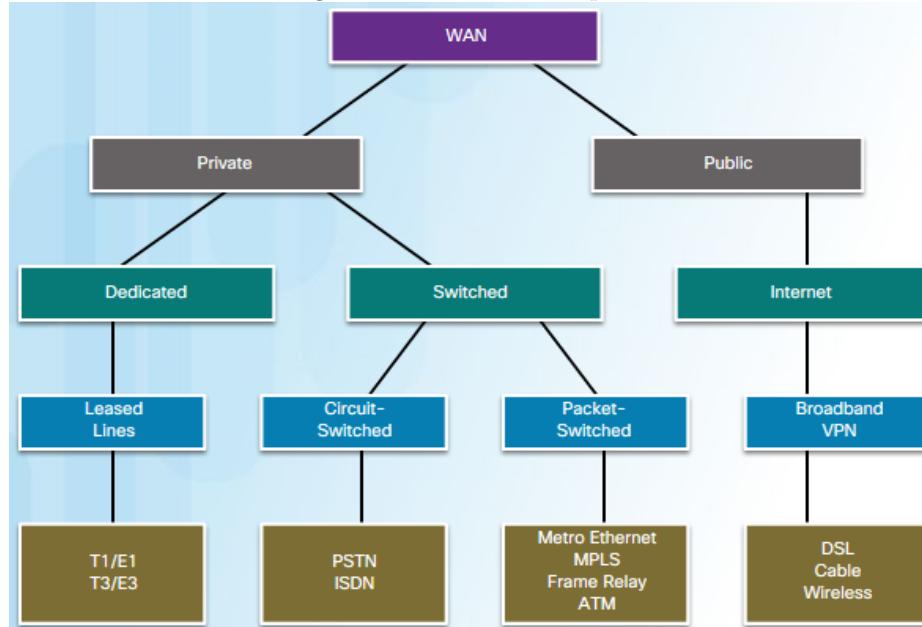
**Packet Switching** splits traffic data into packets that are routed over a shared network. A circuit does not need to be established and many pairs of nodes can communicate over the same channel. Packet switching costs less than circuit switching, however, latency and jitter are greater in packet-switching networks. There are two approaches to packet-switched network link determination:

- **Connectionless systems:** Full addressing information must be carried in each packet. The **Internet** is an example of a connectionless system.
- **Connection-oriented systems:** The network predetermines the route for a packet, and each packet only has to carry an identifier. An example of a connection-oriented system is **Frame Relay** (DLCIs are the identifiers).

## 3.2 WAN connection

There are several WAN access connection options (figure 3.4) that ISPs can use to connect the local loop to the enterprise edge.

Figure 3.4: WAN access options



Service provider networks are complex and consist mostly of high-bandwidth fiber-optic media, using SONET and SDH standard. A newer fiber-optic media development for long-range communications is called dense wavelength division multiplexing (DWDM).

### 3.2.1 Private WAN Infrastructures

**Leased lines** are *permanent dedicated point-to-point* connections from the customer premises to the provider network. The organization pays a monthly lease fee to a service provider to use the line. Leased lines require little installation and maintenance expertise, offer high quality and availability. However, they are expensive and has limited flexibility.

**Dialup** transports binary computer data through the voice telephone network using a modem. Dialup access is suitable when intermittent, low-volume data transfers are needed. The advantages of modem and analog lines are simplicity, availability, and low implementation cost. The disadvantages are the low data rates and a relatively long connection time.

**ISDN** is a *circuit-switching* technology that enables the local loop of a PSTN (Public switched telephone network) to carry digital signals. It can provide additional capacity as needed on a leased line connection or can also be used as a backup. ISDN has declined in popularity due to DSL and other broadband services. There are two types of ISDN Interfaces: BRI (2 B-channels, 1 D-channel), PRI (23 B-channel, 1 D-channel)

**Frame Relay** is a Layer-2 WAN technology used to interconnect enterprise LANs. Frame Relay creates PVCs (Private Virtual Circuits) to connect multiple sites, and carry voice and data traffic. PVCs are uniquely identified by a DLCI (Data-Link Connection Identifier). The PVCs and DLCIs ensure bidirectional communication between one DTE device to another.

**ATM** is built on a *cell-based* architecture rather than on a frame-based architecture. ATM cells are always a *fixed* length of **53 bytes**. ATM is well-suited for voice and video traffic because this traffic is intolerant of delay.s

**Ethernet WAN:** Originally Ethernet was not suitable as a WAN access technology because the maximum cable length was one kilometer. However, *fiber-optic* cables have made Ethernet a reasonable WAN access option. There are several benefits to an Ethernet WAN: Reduced expenses and administration, Easy integration with existing networks, Enhanced business productivity. Ethernet WANs have replaced Frame Relay and ATM.

**MPLS** is a *multiprotocol* high-performance WAN technology that directs data from one router to the next. MPLS is based on *short path labels* rather than IP network addresses. It uses labels which tell a router what to do with a packet. The labels identify paths between distant routers rather than endpoints, and while MPLS actually routes IPv4 and IPv6 packets, everything else is switched. Furthermore, MPLS can deliver any type of packet between sites and encapsulate them of various network protocols.

**VSAT** is a solution that creates a private WAN using *satellite* communications in remote locations where there are no service providers that offer WAN service.

### 3.2.2 Public WAN Infrastructures

**DSL** is an always-on connection technology that uses existing *twisted-pair telephone* lines to transport high-bandwidth data. A DSL modem converts an Ethernet signal from the user device to a DSL signal. Key components in the DSL connection: *DSL modem (subscriber end)* and *DSLAM (ISP end)*. The advantage that DSL has over cable technology is that DSL is not a shared medium – each user has a separate direct connection to the DSLAM.

**Cable** is widely used in urban areas to distribute television signals. Network access is available from television providers. This allows for greater bandwidth than the conventional telephone local loop. Two types of equipment are required: *Cable Modem (subscriber end)* and *CMTS (ISP end)*.

**WiMAX** is a new technology that operates in a similar way to Wi-Fi, but at higher speeds, over greater distances, and for a greater number of users. It uses a network of WiMAX towers that are similar to cell phone towers.

**Satellite Internet** Typically used by rural users where cable and DSL are not available. Cable and DSL have higher download speeds, but satellite systems are about 10 times faster than an analog modem.

**VPN** is an encrypted connection between private networks over Internet. VPN uses virtual connections called VPN tunnels, which are routed through the Internet from the private network of the company to the remote site or employee host. There are several benefits to using VPN: cost savings, security, scalability, compatibility with broadband technology. There are two types of VPN access:

- **Site-to-site VPN:** connect entire networks to each other, for example, connecting a branch office network to a company headquarters network.
- **Remote-access VPN:** enable individual hosts, such as extranet consumers, to access a company network securely over the Internet.

**Dynamic Multipoint VPN (DMVPN)** is a Cisco software solution for building multiple VPNs. DMVPN is built on three protocols: NHRP, IPsec, and mGRE. NHRP is the distributed address *mapping* protocol for VPN tunnels. IPsec *encrypts* communications on VPN tunnels. The mGRE protocol allows the dynamic creation of *multiple spoke tunnels* from one permanent VPN *hub*.

# Chapter 4

## Network evolution

### 4.1 IoT

The Cisco IoT System has six technology pillars: Network Connectivity, Fog computing, Security, Data analysis, Management and Automation, Application Enablement Platform.

**Fog computing** is an IoT network model. It enables end devices to run applications *locally* and make immediate decisions. This reduces the data burden on networks, as raw data does not need to be sent over network connections. It enhances resiliency by allowing IoT devices to operate when network connections are lost. It also enhances security by keeping sensitive data from being transported beyond the edge where it is needed.

The Cisco IoT security pillar offers scalable cybersecurity solutions:

- **Operational Technology (OT)** keeps power plants running and manages factory process lines
- **IoT Network security** includes network and perimeter security devices (e.g. switch, router)
- **IoT Physical security** enables surveillance in a wide variety of environments with Cisco Video Surveillance IP Cameras

### 4.2 Cloud computing

Cloud computing is a network model where servers and services are dispersed globally in distributed data centers. Cloud computing, with its “pay-as-you-go” model, allows organizations to treat computing and storage expenses more as a utility rather than investing in infrastructure. Capital expenditures are transformed into operating expenditures.

#### 4.2.1 Cloud service

- **SaaS** (Software as a Service): The cloud provider is responsible for access to services, such as email, communication, and Office 365 that are delivered over the Internet. The user is only needs to provide their data.
- **PaaS** (Platform as a Service): The cloud provider is responsible for access to the development tools and services used to deliver the applications.
- **IaaS** (Infrastructure as a Service): The cloud provider is responsible for access to the network equipment, virtualized network services, and supporting network infrastructure.
- **ITaaS** (IT support as a Service): cloud service providers provide IT support for each of the cloud computing services

### 4.2.2 Cloud model

- **Public clouds:** Cloud-based applications and services offered in a public cloud are made available to the general population. Services may be *free* or are offered on a *pay-per-use* model, such as paying for online storage. The public cloud uses the Internet to provide services.
- **Private clouds:** Cloud-based applications and services offered in a private cloud are intended for a specific organization or entity, such as the *government*. A private cloud can be set up using the organization's private network, though this can be expensive to build and maintain. A private cloud can also be managed by an outside organization with strict access security.
- **Hybrid clouds:** A hybrid cloud is made up of two or more clouds (example: part private, part public), where each part remains a distinctive object, but both are connected using a single architecture. Individuals on a hybrid cloud would be able to have degrees of access to various services based on user access rights.
- **Community clouds:** A community cloud is created for exclusive use by a specific community. The differences between public clouds and community clouds are the functional needs that have been customized for the community. For example, healthcare organizations must remain compliant with policies and laws (e.g., HIPAA) that require special authentication and confidentiality.

### 4.2.3 Cloud Computing, Data Center, Virtualization

The terms “data center” and “cloud computing” are often incorrectly used. These are the correct definitions of data center and cloud computing:

- **Data center:** Typically a data storage and processing facility run by an in-house IT department or leased offsite.
- **Cloud computing:** Typically an off-premise service that offers on-demand access to a shared pool of configurable computing resources. These resources can be rapidly provisioned and released with minimal management effort.

Cloud computing is possible because of data centers. A data center is a facility used to house computer systems and associated components. Cloud computing is a service provided by data centers. Cloud service providers use data centers to host their cloud services and cloud-based resources.

The terms “cloud computing” and “virtualization” are often used interchangeably; however, they mean different things. Virtualization is the foundation of cloud computing. Cloud computing separates the application from the hardware. Virtualization separates the OS from the hardware.

## 4.3 Virtualization

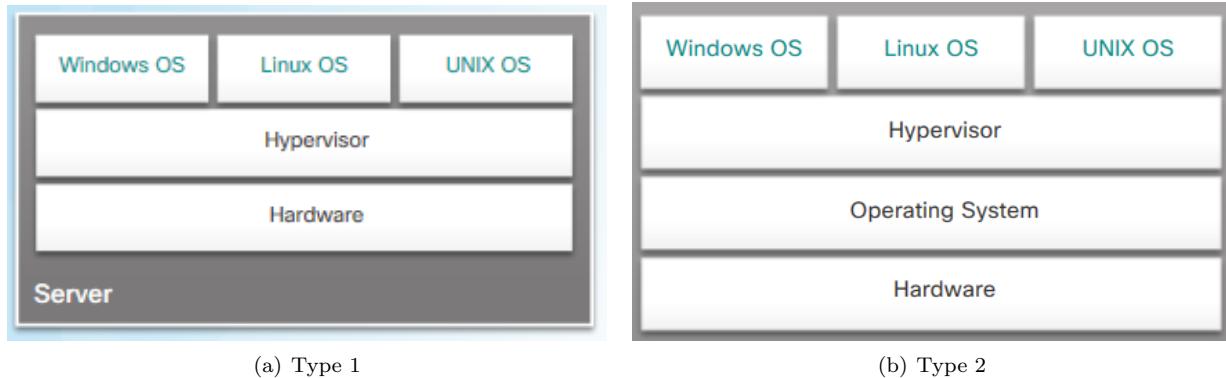
Server virtualization takes advantage of idle resources and consolidates the number of required servers. This also allows for multiple operating systems to exist on a single hardware platform. Virtualization offers a variety of benefits:

- Reduced cost: Virtualization enables server consolidation, which requires fewer physical servers and devices. It also lowers the monthly power, cooling costs and reduces the amount of required floor space.
- Easier prototyping
- Fault tolerance, improved disaster recovery

The **hypervisor** is a program that adds an abstraction layer on top of the real physical hardware. The abstraction layer is used to create virtual machines which have access to all the hardware of the physical machine such as CPUs, memory, disk controllers, and NICs. Each of these virtual machines runs a complete and separate operating system. There are two types of hypervisor:

- **Type 1:** the hypervisor is installed directly on the hardware. Then, instances of one or more OSes are installed on the hypervisor (Figure 4.2(a)). Type 1 hypervisors have direct access to the hardware resources; therefore, they are more efficient than Type 2. This type of hypervisor requires a *management console*.

Figure 4.1: Two types of hypervisor



- **Type 2:** the hypervisor is installed on top of the existing OS (Windows, Mac OS, Linux, etc.). Then, one or more additional OS instances are installed on top of the hypervisor (Figure 4.2(b)). A big advantage of Type 2 hypervisors is that management console software is not required. Type 2 hypervisors are very popular with consumers and for organizations experimenting with virtualization.

**Management console** is used to consolidate and turn on/off servers. It also provides recovery from hardware failure. If a server component fails, the management console automatically and seamlessly moves the VM to another server.

Some management consoles also allow **over allocation**. Over allocation is when multiple OS instances are installed, but their memory allocation exceeds the total amount of memory that a server has. For example, a server has 16 GB of RAM, but the administrator creates four OS instances with 10 GB of RAM allocated to each. This type of over allocation is a common practice because all four OS instances rarely require the full 10 GB of RAM at any one moment.

## 4.4 SDN

### 4.4.1 Introduction

A network device contains the following planes:

- **Control plane:** This is used to make forwarding decisions. The control plane contains Layer 2 and Layer 3 route forwarding mechanisms, such as routing protocol, neighbor tables and topology tables, routing tables, STP, and the ARP table.
- **Data plane:** Also called the forwarding plane, this plane use information from the control plane to forward traffic flows. Information in the data plane is processed by a special data plane processor, such as a digital signal processor (DSP), without the CPU getting involved.

In a traditional router or switch architecture, the control plane and data plane functions occur in the same device. **Software defined networking (SDN)** is a network architecture that virtualizes the control plane. It moves the control plane from each network device to a central network intelligence and policy-making entity called the *SDN controller* or *Network controller* (Figure 4.2). An example of Network controller is OpenDayLight platform.

### 4.4.2 SDN controller

The SDN controller enables network administrators to manage and dictate how the data plane of virtual switches and routers should handle network traffic. It orchestrates, mediates, and facilitates communication between applications and network elements.

Figure 4.2: Traditional and SDN architecture

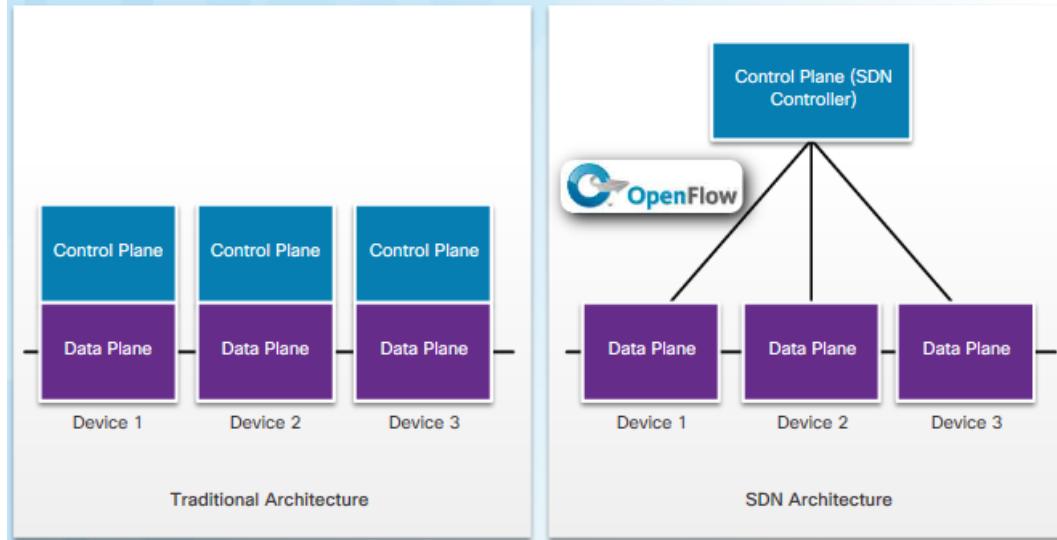
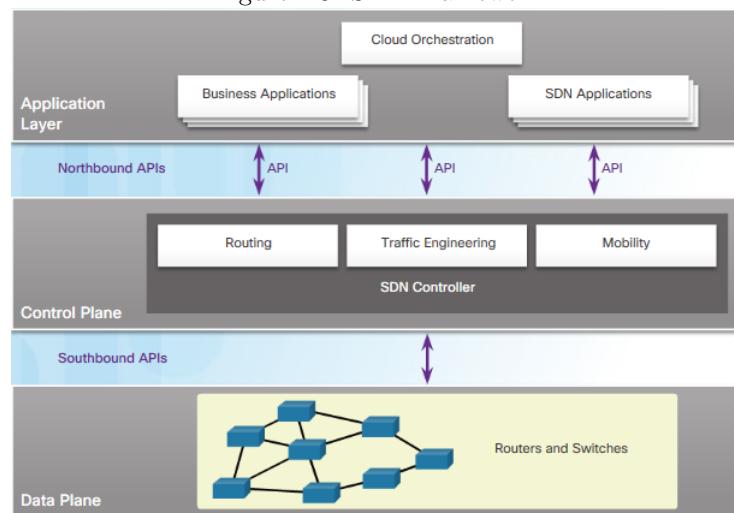


Figure 4.3: SDN Framework



Note the use of **API** (Application Programming Interfaces) within the SDN framework (Figure 4.3). An API is a set of standardized requests that define the proper way for an application to request services from another application. The SDN controller uses *northbound* APIs to communicate with the applications. These APIs help network administrators shape traffic and deploy services. The SDN controller also uses southbound APIs to define the behavior of switches and routers.

**Note!** Traffic in a modern data center is described as North-South (going between external data center users and the data center servers) and East-West (going between data center servers).

#### 4.4.3 Operation

A **flow** is a sequence of packets traversing a network that share a set of header field values. For example, a flow could consist of all packets with the same source and destination IP addresses, or all packets with the same VLAN identifier.

**OpenFlow** protocol is a basic element of SDN which manages traffic between routers, switches, access points, and a controller. OpenFlow is the original southbound API. The Open Networking Foundation is responsible for maintaining the OpenFlow standard.

Each flow traveling through the network must first get permission from the SDN controller. If the controller allows a flow, it computes a route for the flow to take. Then, it adds an entry for that flow in each of the switches along the path by populating *flow tables*. An SDN controller communicates with OpenFlow-compatible switches using the OpenFlow protocol. This protocol uses Transport Layer Security (TLS) to securely send control plane communications over the network.

#### 4.4.4 Types

There are three types of SDN:

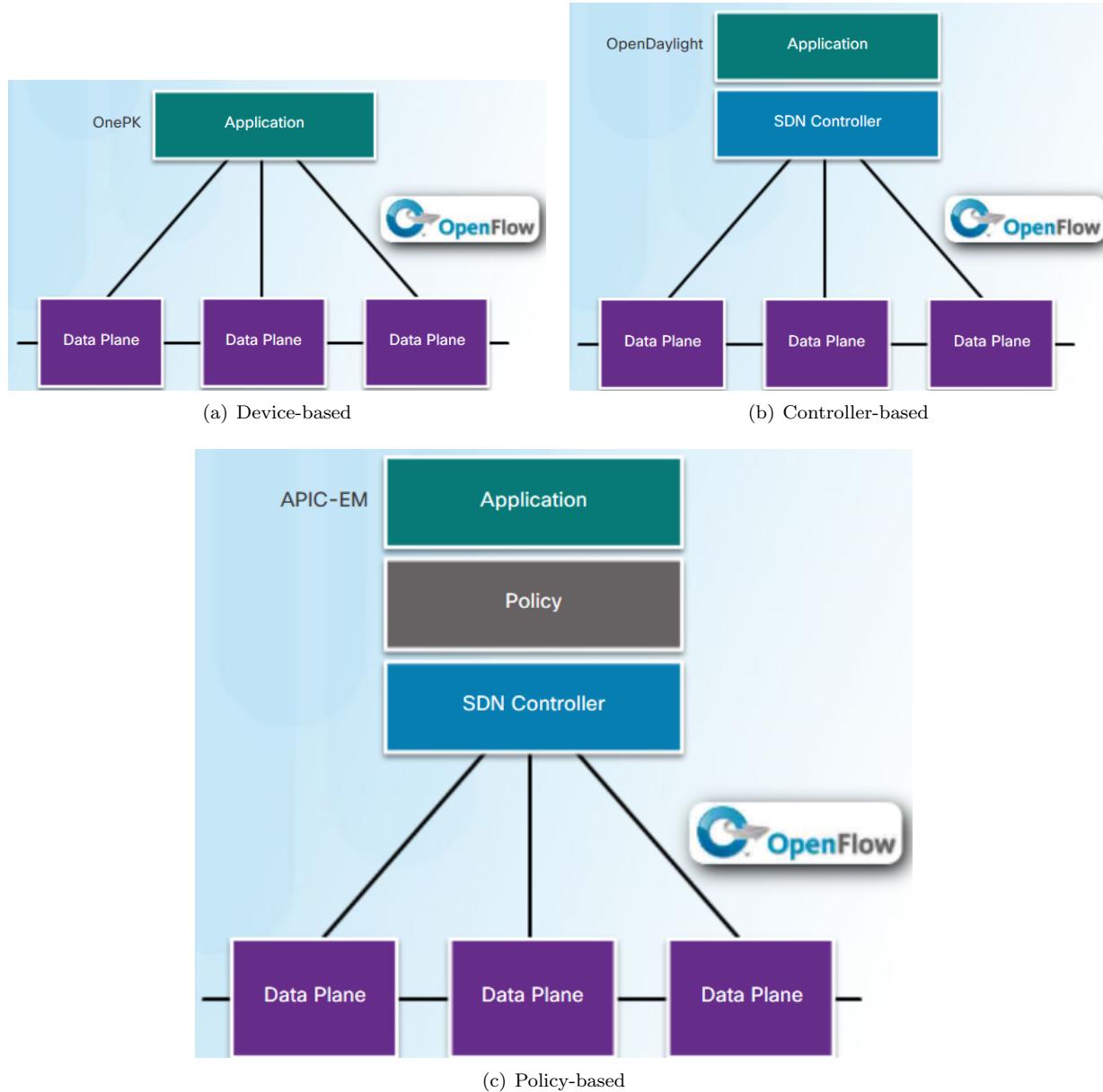
- **Device-based SDN:** the devices are programmable by applications running on the device itself or on a server in the network, as shown in Figure 4.5(a). **Cisco OnePK** is an example of a device-based SDN.
- **Controller-based SDN** uses a centralized controller that has knowledge of all devices in the network, and manipulates traffic flows throughout the network, as shown in Figure 4.5(b). The **OpenDaylight** and **Cisco Open SDN Controller** (commercial distribution of OpenDayLight) are examples of Controller-based SDN.
- **Policy-based SDN** is similar to controller-based SDN where a centralized controller has a view of all devices in the network, as shown in Figure 4.5(c). Policy-based SDN includes an additional Policy layer that operates at a higher level of abstraction. It uses built-in applications that automate advanced configuration tasks via a guided workflow and user-friendly GUI. No programming skills are required. **Cisco APIC-EM** is an example of Policy-based SDN.

#### 4.4.5 ACI

**ACI** (Cisco Application Centric Infrastructure) is the Cisco's approach to SDN. It involves three main things: ANP, APIC, and programmable switches (Figure 4.5(c)):

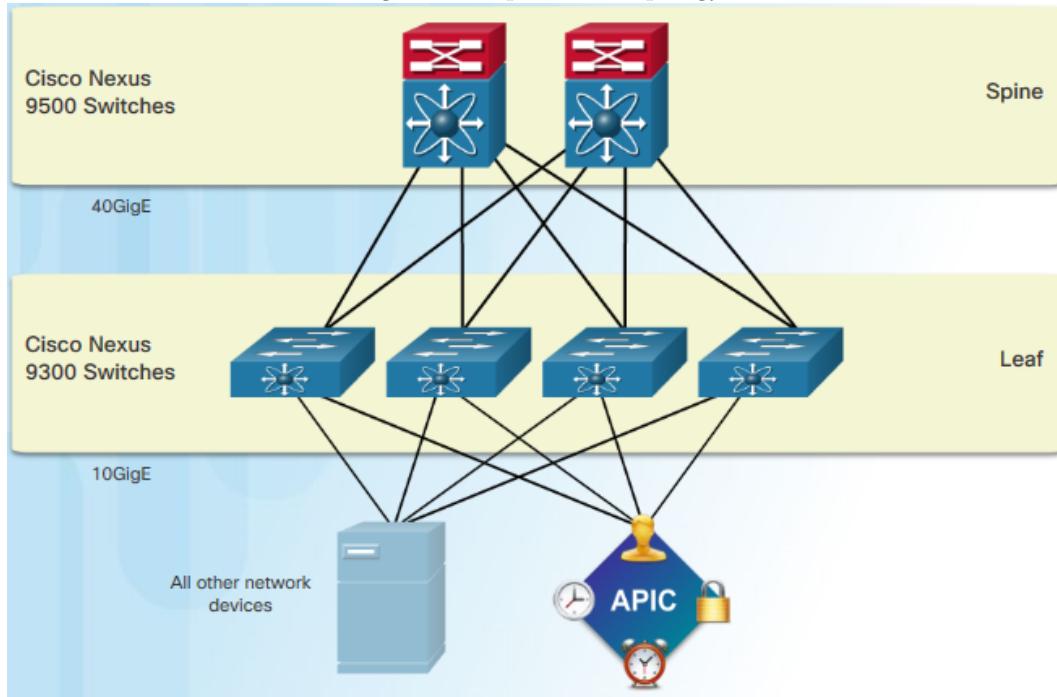
- **Application Network Profile (ANP)** (the gray box in figure 4.5(c)) is a collection of end-point groups, their connections, and the policies that define those connections, such as VLANs, Web services, and applications.
- **Application Policy Infrastructure Controller (APIC)** (blue box in figure 4.5(c)) is considered to be the brains of the ACI architecture. It manages and operates a scalable ACI clustered fabric. APIC is designed for programmability and centralized management. It translates application requirements into network programming. The APIC is positioned between the ANP and the ACI-enabled network infrastructure.
- **Cisco Nexus 9000 Series switches** (Figure 4.5) provide an application-aware switching fabric and work with an APIC to manage the virtual and physical network infrastructure.

Figure 4.4: Three types of SDN



**OpenStack** is an *orchestration* platform that builds scalable cloud environments and provide IaaS solution. OpenStack is often used with Cisco ACI. Orchestration in networking is the process of automating the provisioning of network components such as servers, storage, switches, routers, and applications.

Figure 4.5: Spine-Leaf topology



The Cisco ACI fabric is composed of the APIC and the Cisco Nexus 9000 series switches using two-tier spine-leaf topology (Figure 4.5). The spine switches only attach to the leaf and core switches (not shown). In this two-tier topology, everything is one hop from everything else.

When compared to SDN, the APIC controller does not manipulate the data path directly. Instead, the APIC centralizes the policy definition and programs the leaf switches to forward traffic based on the defined policies.

#### 4.4.6 APIC-EM

Cisco APIC-EM provides the following features:

- **Discovery:** populate the controller's device and host inventory database.
- **Device Inventory:** collects detailed information from network devices (device name, device status, MAC address, etc.)
- **Host Inventory:** - Collects detailed information from hosts (host name, user ID, MAC address, IPv4/IPv6 addresses, etc.)
- **Topology:** graphical view of the network, auto-visualization of Layer 2 and 3 topologies on top of the physical topology
- **Policy:** view and control policies across the entire network including QoS.
- **Policy Analysis:** quickly identify ACLs in use and problem areas, enable ACL change management with easy identification of redundancy, conflicts and incorrect ordering of access control entries.

One of the most important features of the APIC-EM controller is the ability to manage policies across the entire network. APIC-EM ACL Analysis and Path Trace provide tools to allow the administrator to analyze and understand ACL policies and configurations:

- **ACL Analysis** enables ACL *inspection* and *interrogation* across the entire network, exposing any problems and conflicts.
- **ACL Path Trace** *examines* specific ACLs on the path between two end nodes, displaying any potential issues.

## **Part II**

# **SWITCHING TECHNOLOGIES**



# Chapter 5

## Switch

### 5.1 Operation

#### 5.1.1 Switch boot sequence

After a Cisco switch is powered on, it goes through the following boot sequence:

1. The switch loads a power-on self-test (POST) program stored in ROM.
2. Boot loader software is loaded.
3. The boot loader initializes the CPU registers.
4. The boot loader initializes the flash file system.
5. The boot loader loads IOS image into RAM and gives control of the switch to the IOS.
6. The IOS initializes the startup-config file, which is stored in NVRAM.

The boot loader software finds the Cisco IOS image using *Boot environment variables*. If this variable is not set, the switch attempts to load and execute the first executable file it finds. Use the command `show boot` to see content of the the variable and what the current IOS boot file is set to.

**Listing 1: Setting boot environment variable**

```
boot system flash:/c2960-lanbasek9-mz.150-2.SE/c2960-lanbasek9-mz.150-2.SE.bin
```

#### 5.1.2 MAC address table

A Layer 2 Ethernet switch uses MAC addresses to make forwarding decisions. It consults a MAC address table to make a forwarding decision for each frame. By default, most Ethernet switches keep an entry in the MAC address table for 5 minutes.

**Learning MAC Address:** The switch dynamically builds the MAC address table by examining the source MAC address of the frames received on a port. Every frame that enters a switch is checked for new information to learn. If the source MAC address does not exist, it is added to the table along with the incoming port number. If the source MAC address does exist, the switch updates the refresh timer for that entry. If the source MAC address does exist in the table but on a different port, the switch treats this as a new entry.

**Forwarding MAC Address:** Next, if the destination MAC address is a unicast address, the switch will look for a match between the destination MAC address of the frame and an entry in its MAC address table. If the destination MAC address is in the table, it will forward the frame out the specified port. If the destination MAC address is not in the table, the switch will forward the frame out all ports except the incoming port. If the destination MAC address is a broadcast or a multicast, the frame is also flooded out all ports except the incoming port.

A switch can have multiple MAC addresses associated with a single port. This is common when the switch is connected to another switch.

### 5.1.3 Frame forwarding method

A switch makes a decision on frame forwarding based on two criteria: **Ingress port** and **Destination address**. Switches use one of the following forwarding methods: Store-and-forward switching and Cut-through switching.

**Store-and-forward switching:** has two primary characteristics that distinguish it from cut-through: **error checking** and **automatic buffering**. When the switch receives the frame, it stores the data in buffers until the complete frame has been received. In this process, the switch also performs an error checking using the CRC trailer portion of the Ethernet frame. When an error is detected in a frame, the switch discards the frame. Discarding frames with errors reduces the amount of bandwidth consumed by corrupt data. Store-and-forward switching is required for Quality of Service (QoS).

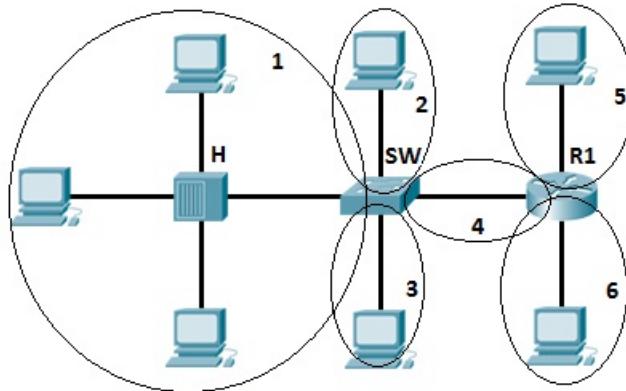
**Cut-through switching:** There are two primary characteristics of cut-through switching: **rapid frame forwarding** and **fragment free**. The switch buffers just enough of the frame to read the destination MAC address (first 6 bytes) so that it can determine to which port to forward the data. No error detection is performed. There are two variants of cut-through switching:

- **Fast-forward switching:** the switch immediately forwards a packet after reading the destination address. It offers the lowest level of latency.
- **Fragment-free switching** the switch waits for the collision window (64 bytes) to pass before forwarding the frame. The reason is that most network errors and collisions occur during the first 64 bytes.

### 5.1.4 Switching domain

**Collision domain:** In *hub-based* Ethernet segments, network devices must take turns when transmitting. The network segments that share the same bandwidth between devices are known as collision domains. Therefore, all ports of a hub are in the same collision domain. However, each port on a switch, or a router is a separate collision domain.

Figure 5.1: An example of collision domains



**Broadcast domain** is a domain in which a broadcast is forwarded. A broadcast domain contains all devices that can reach each other at the data link layer (OSI layer 2) by using broadcast. All ports on a hub or a switch are by default in the same broadcast domain. All hosts in the a VLAN are in the same broadcast domain regardless connecting to different switches. Every port on a router locates in a separate broadcast domain. Note that routers don't forward broadcasts from one broadcast domain to another.

### 5.1.5 Memory Buffering on Switches

There are two methods of memory buffering: Port-based Memory Buffering and Shared Memory Buffering.

**Port-based Memory Buffering:** Frames are stored in queues that are linked to specific incoming and outgoing ports. A frame is transmitted to the outgoing port only when all the frames ahead of it in the queue have been successfully transmitted.

**Shared Memory Buffering:** deposits all frames into a common memory buffer that all the ports on the switch share. The frames in the buffer are linked dynamically to the destination port. This allows the packet to be transmitted on a port without order and waiting. This method permits larger frames to be transmitted with fewer dropped frames.

## 5.2 Recovering from system crash

The boot loader command line supports commands to format the flash file system, reinstall the operating system software, recover from system crash, recover a forgotten password. The boot loader command line can be accessed through a console connection following these steps:

1. Connect a PC by console cable to the switch console port. Configure terminal emulation software to connect to the switch.
2. Power off the switch by unplugging the switch power cord.
3. Reconnect the power cord to the switch and, within 15 seconds, press and hold down the Mode button while the System LED until it turns briefly amber and then solid green.

## 5.3 Configuration

### 5.3.1 Basic switch management

To prepare a switch for remote management, it must be configured with an SVI, IP address, a subnet mask, and a default gateway. This is similar to configuring the IP address information on host devices. Please note that these IP settings are only for remote management access to the switch. The IP settings do not allow the switch to route Layer 3 packets.

**Listing 2: Basic switch management**

```
interface vlan 99
  ip address 172.17.99.11 255.255.255.0
  no shutdown
  exit
ip default-gateway 172.17.99.1
```

### 5.3.2 Duplex configuration

When a switch port is operating in full-duplex mode, there is no collision domain associated with the port. In contrast, half-duplex creates collision domain. The following command manually sets the F0/1 to full duplex and 100 Mb/s.

**Listing 3: Duplex configuration**

```
interface fa 0/1
  duplex full
  speed 100
end
show interface
```

**5.3.3 Auto-MDIX**

When auto-MDIX is enabled, the interface automatically detects the required cable connection type (straight-through or crossover) and configures the connection appropriately. In other words, with auto-MDIX enabled, either type of cable can be used to connect to other devices.

**Listing 4: Auto-MDIX**

```
interface fastethernet 0/2
  mdix auto
end
show controllers ethernet-controller fa 0/2 phy | include Auto-MDIX
```

**5.3.4 SSH**

Before configuring SSH, the switch must be minimally configured with a unique hostname and the correct network connectivity settings. To display the version and configuration data for SSH on the device that you configured as an SSH server, use the `show ip ssh` command. To check the SSH connections to the device, use the `show ssh` command.

**Listing 5: SSH configuration**

```
ip domain-name cisco.com
ip ssh version 2
crypto key generate rsa
username admin secret ccna

line vty 0 15
  transport input ssh
  login local
end

show ip ssh
show ssh
```

**5.3.5 Port security**

One way to secure ports is by implementing a feature called *port security*. Port-security limits the number of valid MAC addresses allowed on a port. The MAC addresses of legitimate devices are allowed access, whereas other MAC addresses are denied. If a port is configured as a secure port and the maximum number of MAC addresses is reached, any additional attempts to connect by unknown MAC addresses generate a security violation.

Enable port-security feature will not work until the `switchport port-security` interface configuration command is executed. The type of secure address is based on the configuration and includes the following:

- **Static secure MAC addresses** are manually configured on a port. They are stored in the address table and are added to the running configuration on the switch.
- **Dynamic secure MAC addresses** are dynamically learned and stored only in the address table. They are removed when the switch restarts.
- **Sticky secure MAC addresses** can be dynamically learned or manually configured, and then stored in the address table and added to the running configuration. When sticky learning is enabled, the switch converts all dynamically learned MAC addresses, including those that were dynamically learned before sticky learning was enabled, into sticky secure MAC addresses. If sticky learning is disabled, the sticky secure MAC addresses remain part of the address table but are removed from the running configuration.

An interface can be configured for one of three violation modes: Protected, Restricted, and Shutdown. In these modes, a switch drops all frames with unknown source addresses until a sufficient number of these MAC addresses are removed, or the number of maximum allowable addresses is increased. No error messages are displayed in these modes.

Table 5.1: Security violation mode

Violation mode	Syslog message	Increase violation counter	Shut down port
Protected			
Restricted	*	*	
Shutdown		*	*

Listing 6: Port security configuration

```
int f0/1
switchport port-security
switchport port-security mac-address sticky
switchport port-security violation restrict
switchport port-security maximum 2
end
show int
show port-security int
```

**Shutdown** is the default violation mode. A port-security violation in this mode causes the interface to become *error-disabled*. You can bring an error-disabled interface back to normal by entering `shutdown` followed by `no shutdown` command. When a port is error-disabled, the port LED will turn off, and the `show interface` command identifies the port status as `err-disabled`. The output of the `show port-security int` command shows the port status as `secure-shutdown`.

**Restrict** is the only port security mode that can assist with troubleshooting by sending syslog messages and keeping count of violations.

## 5.4 Troubleshooting

### 5.4.1 Gathering symptoms

The output from the `show int` command can be used to detect common media issues.

```
S1# show int f0/1
FastEthernet0/1 is up, line protocol is up (connected)
```

```
Hardware is Lance, address is 0022.91c4.0e01 (bia 0022.91c4.0e01)
MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
<output omitted>
```

- The first parameter (`FastEthernet0/1 is up`) refers to the *physical layer* and indicates whether the interface is receiving a carrier detect signal. The second parameter (`line protocol is up`) refers to the *data link layer* and indicates whether the data link layer protocol keepalives are being received.
- If the interface is up and the line protocol is down, there could be an *encapsulation* type mismatch, or the interface on the other end could be *error-disabled*.
- If the line protocol and the interface are both down, a cable is not attached, or the other end of the connection may be administratively down.
- If the interface is administratively down, it has been manually disabled (the `shutdown` command has been issued) in the active configuration.

```
S1# show interfaces FastEthernet 0/1
FastEthernet0/1 is up, line protocol is up (connected)
<output omitted>

3 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 120 multicast, 0 pause input
0 input packets with dribble condition detected
3594664 packets output, 436549843 bytes, 0 underruns
<output omitted>
```

**Input errors** is the sum of all errors in frames that were received on the interface being examined. The reported input errors from the above output include the following:

- **Runt frames:** Ethernet frames that are shorter than the 64-byte (minimum allowed length of a frame) are called runt frames. Runt frames are caused by malfunctioning NIC or collisions.
- **Giant:** Ethernet frames that are larger than 1518-byte (maximum allowed size of a frame) are called giants.
- **CRC errors:** Common causes come from the cable (electrical interference, loose or damaged connections, and incorrect cabling). If you see many CRC errors, there is too much noise on the cable and you should inspect the cable. You should also search for and eliminate noise sources.

```
S1# show interfaces FastEthernet 0/1
FastEthernet0/1 is up, line protocol is up (connected)
<output omitted>

8 output errors, 1790 collisions, 10 interface resets
0 unknown protocol drops
0 babbles, 235 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out
```

**Output errors** is the sum of all errors that prevented the final transmission of frames out the interface that is being examined. The reported output errors include the following:

- **Collisions:** Collisions in half-duplex operations are normal. However, you should never see collisions on an interface configured for full-duplex communication.
- **Late collision:** A late collision refers to a collision that occurs after **512 bits** of the frame have been transmitted. Common causes: Excessive cable lengths, Duplex mismatch.

### 5.4.2 Take action

If the interface is down, take these two actions:

1. Check the *cable*. Make sure the proper and non-damaged cables are being used.
2. If the interface is still down, the problem may be due to a *speed mismatch*. Manually set the same speed on both connection ends if this problem is suspected.

If the interface is up, but issues with connectivity are still present, do the following:

1. Using the `show interfaces` command, check for indications of excessive *noise*. Indications may include an increase in the counters for runts, giants, and CRC errors. If there is excessive noise, find and remove the source of the noise, verify that the cable does not exceed the maximum cable length and check the cable type.
2. If noise is not an issue, check for excessive *collisions*. If there are collisions or late collisions, the problem may be due to *duplex mismatch*. If this is true, manually set the duplex to full on both ends of the connection.



# Chapter 6

## VLAN

### 6.1 Overview

VLANs provide a way to group devices within a LAN. Devices within a VLAN act as if they are in their own independent network, even if they share a common infrastructure with other VLANs. The transmission of unicast, multicast, and broadcast traffic from a host in a particular VLAN are restricted to the devices that are in that VLAN.

The primary benefits of using VLANs are as follows:

- **Security:** Groups that have sensitive data are separated from the rest of the network, decreasing the chances of confidential information breaches.
- **Better performance:** Dividing flat Layer 2 networks into multiple logical workgroups (broadcast domains) reduces unnecessary traffic on the network and boosts performance.
- **Grouping:** VLANs allows logical grouping of users by function.

There are a number of distinct types of VLANs:

- **Default VLAN:** By default, all switch ports are assigned to default VLAN (VLAN 1). VLAN 1 has all the features of any VLAN, except it cannot be renamed or deleted.
- **Native VLAN** is assigned to trunk ports. It serves as a common identifier on opposite ends of a trunk link. It is a best practice to configure a native VLAN for all trunk ports in the switched domain.
- **Data VLAN** is configured to carry user-generated traffic. A VLAN carrying voice or management traffic would not be a data VLAN.
- **Management VLAN** is configured to access the management capabilities of a switch. To create the management VLAN, the SVI<sup>1</sup> of that VLAN is assigned an IP address and a subnet mask. VLAN 1 would be a bad choice for the management VLAN.
- **VoIP VLAN** is a separate VLAN needed to support Voice over IP (VoIP).

**Normal range:** VLANs are identified by a VLAN ID **between 1 and 1005**. Configurations are stored within a VLAN database file, called **vlan.dat**. This file is located in the flash memory of the switch. VTP can only learn and store normal range VLANs.

**Extended Range:** VLANs are identified by a VLAN ID **between 1006 and 4094**. Configurations are not written to the **vlan.dat** file, but instead, stored in **running configuration file**. VTP does *not* learn extended range VLANs.

---

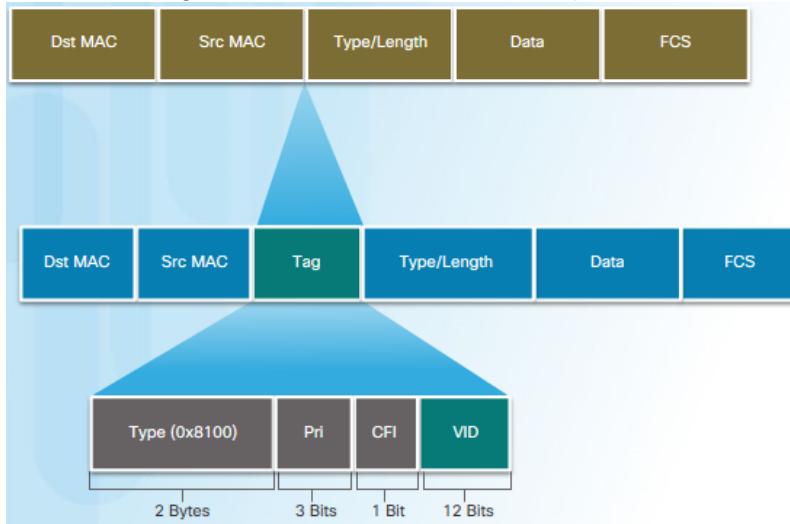
<sup>1</sup>Switch Virtual Interface

## 6.2 VLAN tag

### 6.2.1 What is tagging?

A **trunk** is a point-to-point link between two network devices that carries more than one VLAN. VLAN trunks allow all VLAN traffic to propagate between switches so that devices that are in the same VLAN, but connected to different switches. When Ethernet frames are placed on a trunk, information about the VLANs to which they belong must be added. This process, called **tagging**, is accomplished by using the IEEE **802.1Q** header.

Figure 6.1: Fields in Ethernet 801.Q frame



The 802.1Q header includes a 4-byte tag inserted within the original Ethernet frame header (Figure 6.1), specifying the VLAN to which the frame belongs. When the switch receives a frame on an *access* port assigned to a VLAN, the switch inserts a VLAN tag in the frame header, recalculates the FCS<sup>2</sup>, and sends the tagged frame out of a trunk port.

### 6.2.2 VLAN tag field

The VLAN tag field (Figure 6.1) consists of:

- **Type:** A 2-byte value called the tag protocol ID (TPID) value. For Ethernet, it is set to hexadecimal **0x8100**.
- **User priority:** A 3-bit value that supports level or service implementation
- **Canonical Format Identifier (CFI):** A 1-bit identifier that enables Token Ring frames to be carried across Ethernet links.
- **VLAN ID:** A 12-bit VLAN identification number that supports up to 4096 VLAN IDs.

After the switch inserts the Type and tag control information fields, it recalculates the FCS values and inserts the new FCS into the frame.

### 6.2.3 Native VLAN and tagging

A trunk port always forwards untagged frames to the native VLAN. If there are no devices associated with the native VLAN and there are no other trunk ports, then the frame is dropped. Additionally, if an 802.1Q trunk port receives a tagged frame with the VLAN ID that is the same as the native VLAN, it drops the frame.

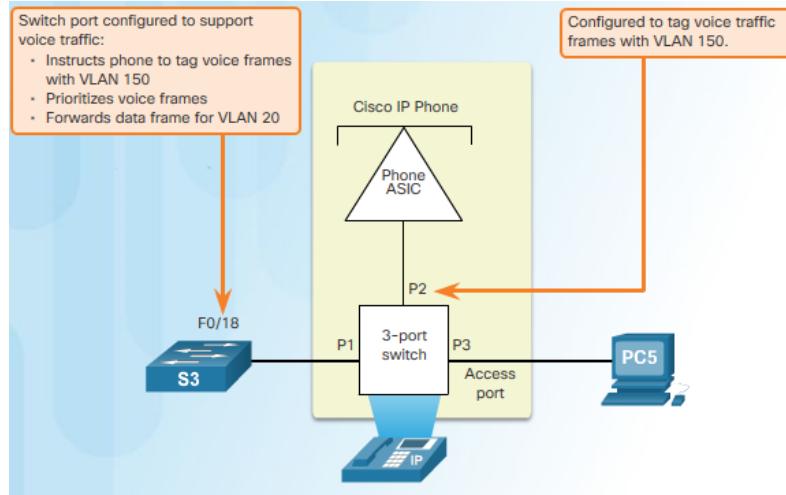
<sup>2</sup>Frame Check Sequence

### 6.2.4 Voice VLAN tagging

The link between the switch and the IP phone acts as a trunk to carry both voice VLAN traffic and data VLAN traffic. The Cisco IP Phone contains an integrated three-port 10/100 switch. The ports provide dedicated connections to these devices:

- Port 1 connects to the switch or other VoIP device.
- Port 2 is an internal 10/100 interface that carries the IP phone traffic.
- Port 3 (access port) connects to a PC or other device.

Figure 6.2: IP phone ports



On the switch, the access is configured to send CDP packets that instruct an attached IP phone to send voice traffic to the switch in one of three ways:

- In a voice VLAN tagged with a Layer 2 CoS<sup>3</sup> priority value
- In an access VLAN tagged with a Layer 2 CoS priority value
- In an access VLAN, untagged (no Layer 2 CoS priority value)

## 6.3 Configuration

### 6.3.1 Create VLAN

A VLAN is created using the `vlan <vlan-id>` global configuration command. This creates the VLAN and enters VLAN configuration mode. The VLAN can now be assigned a unique name using the `name <vlan-name>` subcommand.

```
S1(config)# vlan 20
S1(config)# name Student
```

Instead of creating one VLAN at a time, several VLANs can be created using one command. A series of VLAN IDs can be entered separated by commas, or a range of VLAN IDs can be entered separated by hyphens (-).

```
S1(config)# vlan 100,102,105-107
```

<sup>3</sup>Class of Service

### 6.3.2 Access port

After creating a VLAN, the next step is to assign access ports to the VLAN. First of all, the port must be assigned as an access port using the `sw mode access` interface configuration command. Next, assign the port to a VLAN using the `sw access vlan <vlan_id>` interface configuration command. Note that an access port can belong to only one VLAN at a time.

```
S1(config)# interface FastEthernet0/1
S1(config-if)# sw mode access
S1(config-if)# sw access vlan 20
S1(config-if)# end
```

### 6.3.3 Delete VLAN

We can delete a VLAN with `no vlan <vlan_id>` in global configuration mode. However, be careful when removing a VLAN. Before deleting a VLAN, reassign all member ports to a different VLAN. Any ports that are not moved to an active VLAN are unable to communicate with other hosts after the VLAN is deleted.

The entire VLAN configuration can be erased using `delete vlan.dat`. This command, along with `erase startup` effectively places the switch into its factory default condition.

### 6.3.4 Trunk port

To configure a switch port on one end of a trunk link, use the `sw mode trunk` interface configuration command. The native VLAN can also be changed (other than VLAN 1) using `sw trunk native vlan <vlan-id>`. Always configure both ends of a trunk link with the same native VLAN. Use `sw trunk allowed vlan <vlan-list>` command to specify the list of VLANs to be allowed on the trunk link.

**Listing 7: Trunking**

```
interface f0/1
  sw trunk encapsulation dot1q
  sw mode trunk
  sw trunk native vlan 99
  sw trunk allowed vlan 10,20,30,99
end
```

Please note that we have to configure 802.1Q encapsulation before trunking mode using `sw trunk encapsulation dot1q` command. Otherwise, the following message will appear:

**Command rejected: An interface whose trunk encapsulation is "Auto" can not be configured to "trunk" mode.**

To reset a trunk to allow all VLANs, use the `no sw trunk allowed vlan` interface configuration command. To reset the native VLAN to VLAN 1, use the `no sw trunk native vlan` interface configuration command. To set the port to a non-trunking port, use the `sw mode access` interface command.

### 6.3.5 Voice VLAN

Consider the topology in Figure 6.3. In this example, PC5 is connected to the Cisco IP phone, which in turn is connected to the F0/18 interface on S3. To implement this configuration, Data VLAN 20 (for PC) and Voice VLAN 150 (for IP phone) are created.

**Listing 8:** Voice VLAN example

```
vlan 20
  name Student
vlan 150
  name Voice
exit

interface f0/18
  sw mode access
  sw access vlan 20
  mls qos trust cos
  sw voice vlan 150
end
```

Figure 6.3: Sample topology with IP phone



The command `mls qos trust cos` enables QoS classification based on the class of service (CoS) assigned by the IP phone. The command `sw voice vlan 150` assigns voice VLAN 150 to port F0/18.

**Listing 9:** VLAN verification

```
show vlan name Student
show vlan summary
show vlan brief
show vlan id 20
show interfaces vlan 20
```

**Listing 10:** Trunking verification

```
show int switchport
show int f0/1 switchport
show int trunk
```

**6.3.6 Troubleshoot**

**IP address:** Each VLAN must correspond to a unique IP subnet. If two devices in the same VLAN have different subnet addresses, they cannot communicate. This is a common problem, and it is easy to solve by identifying the incorrect configuration and changing the subnet address to the correct one.

**Missing VLAN:** Verify whether the port is in the correct VLAN using `show vlan` and `sh mac address-table int <int>`. Verify if the VLAN is present in the VLAN database using `show vlan`. Use the `show interfaces switchport` command to verify if the inactive VLAN is assigned to the port.

**Native VLAN mismatches:** Trunk ports are configured with different native VLANs. This configuration error generates console notifications and can cause inter-VLAN routing issues. To solve the native VLAN mismatch, configure the native VLAN to be the same VLAN on both sides of the link.

**Trunk mode mismatches:** One trunk port is configured in a mode that is not compatible for trunking on the corresponding peer port. This configuration error causes the trunk link to stop working. Be sure both sides of the trunk are configured with the `sw mode trunk` command.

**Allowed VLANs** The list of allowed VLANs on a trunk has not been updated with the current VLAN trunking requirements.

## 6.4 Inter-VLAN Routing

Layer 2 switches have limited IPv4 and IPv6 functionality and cannot perform the routing between VLANs. Any device that supports Layer 3 routing, such as a router or a Layer 3 switch (multiplayer), can be used to perform VLAN routing. Regardless of the device used, the process of forwarding network traffic from one VLAN to another is known as inter-VLAN routing. There are three options for inter-VLAN routing: Legacy inter-VLAN routing, Router-on-a-stick, and Layer 3 switching using SVIs.

### 6.4.1 Legacy inter-VLAN routing

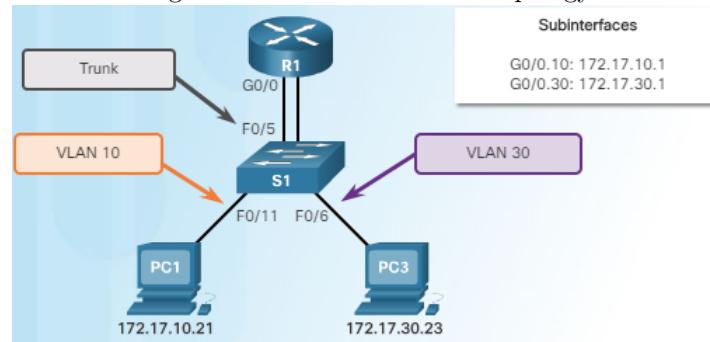
In this legacy approach, inter-VLAN routing is performed by connecting different physical router interfaces to different physical switch ports. The switch ports connected to the router are placed in access mode, and each physical interface is assigned to a different VLAN.

### 6.4.2 Router-on-a-stick

Router-on-a-stick uses a single physical interface to route traffic between multiple VLANs. The router interface is configured to operate as a trunk link and is connected to a switch port that is configured in trunk mode.

Router-on-a-stick performs inter-VLAN routing by accepting VLAN-tagged traffic coming from the switch, and then, *internally* routing between the VLANs using subinterfaces. Subinterfaces are virtual interfaces, associated with a single physical interface. Each subinterface is independently configured with an IP address and VLAN assignment (Figure 6.4). The IPv4 address of the subinterface acts as the default gateway for all hosts in the corresponding VLAN.

Figure 6.4: Router-on-a-stick topology



To enable Router-on-a stick, start by enabling trunking on the switch port that is connected to the router. As for the router, on the port connected to the switch, we create a subinterface for each VLAN. On each subinterface, configure 802.1Q trunk for a specific VLAN using `encap dot1q <vlan-id>`. Next, assign the IPv4 address corresponding to the VLAN subnet. Finally, the router interface connected to the switch must be enabled by `no shutdown` command. Remember to use the `native` keyword for the native VLAN, otherwise, the router would consider VLAN 1 as the native VLAN.

Listing 11: Router-on-a-stick

```

interface f0/5
  sw mode trunk
end

interface g0/0.10
  encapsulation dot1q 10
  ip address 172.17.10.1 255.255.255.0
interface g0/0.30
  encapsulation dot1q 30
  ip address 172.17.30.1 255.255.255.0
interface g0/0.99
  encapsulation dot1q 99 native
  ip address 192.168.100.1 255.255.255.0
exit

interface g0/0
  no shutdown
end

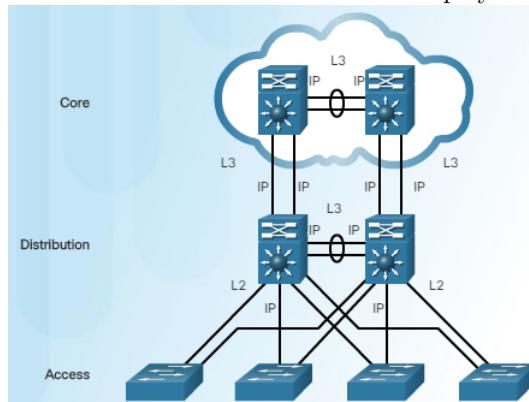
show vlan
show ip route

```

### 6.4.3 Layer 3 switching using SVIs

Most modern enterprise networks use multilayer switches to achieve much higher packet-switching throughputs (millions of packets per second). Layer 3 switching is much faster than router-on-a-stick, because everything is hardware switched and routed. Despite its price, multilayer switch provide lower latency, support EtherChannel to get more bandwidth. Furthermore, with the multiplayer switch, the network architecture (Figure 6.5) is not dependent on STP<sup>4</sup> anymore because layer 2 loops never occurs in the topology.

Figure 6.5: Network architecture with multiplayer switches

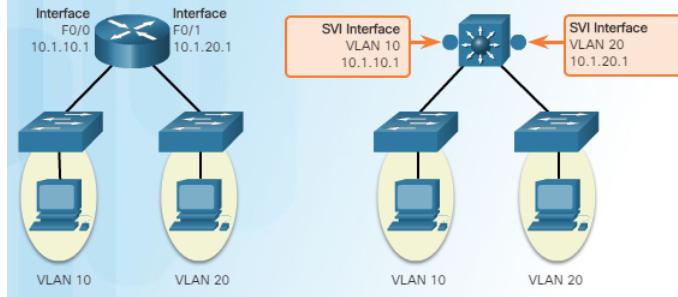


Inter-VLAN routing is performed using SVIs and routed ports.

- **Routed port:** A pure Layer 3 interface similar to a physical interface on a router. A routed port is not associated with any VLAN. Routed ports are normally implemented between the distribution and the core layer (usually between multiplayer switch and a router or a security device). To configure routed ports, use the `no switchport` interface configuration mode command.

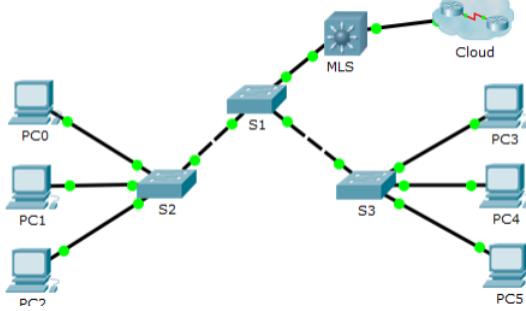
<sup>4</sup>Spanning Tree Protocol, see also section 8

Figure 6.6: Router-on-a-stick and Multiplayer switch



- **SVI:** An SVI is configured for each VLAN that exists on the switch. It can perform the same functions as subinterface in Router-on-a-stick method (Figure 6.6).

Figure 6.7: Sample topology



Take the multi-layer switch MLS in Figure 6.7 as an example. The first thing to do is enabling layer 3 routing. Because the interface g0/2 is connected to a router on the cloud, it should be configured as a routed port and assigned an IP address. Next, create VLANs and configure SVI for each of them.

Listing 12: Multiplayer switch

```

ip routing

interface g0/2
  no switchport
  ip address 209.165.200.225 255.255.255.252
exit

vlan 10
  name Staff
vlan 20
  name Student
exit

interface vlan 10
  ip address 192.168.10.254 255.255.255.0
interface vlan 20
  ip address 192.168.20.254 255.255.255.0
end

show ip route

```

# Chapter 7

## VTP

### 7.1 Overviews

VLAN trunking protocol (VTP) allows a network administrator to manage VLANs on a switch configured as a VTP server. The VTP server distributes and synchronizes VLAN information over trunk links to VTP-enabled switches throughout the switched network. The following provides a brief description of important components of VTP:

- **VTP domain** consists of all interconnected switches. All switches in a domain share VLAN configuration details. Switches resides in different domains do not exchange VTP messages. The boundary of a VTP domain is a router or a layer-3 switch.
- **Revision number** is a 32-bit number that indicates the level of revision for a VTP advertisements. Each VTP device tracks the VTP configuration revision number that is assigned to it. Each time that you make a VLAN change in a VTP device, the configuration revision is incremented by one. Therefore, this number is used to determine whether the received information is more recent than the current version.
- **Password:** Switches in the same domain are configured with the same password for security reason.
- **VTP modes:** A switch can be configured as a VTP server, client, or transparent.
- **VTP server** stores the VLAN information in NVRAM (`vlan.dat`), then advertises it to other switches. VLAN configuration is allowed, and affects the entire VTP domain.
- **VTP client** stores the VLAN information in RAM, therefore, a switch reset deletes all VLAN information. VLAN configuration is not allowed.
- **VTP transparent** does not allow switches to participate in VTP except to forward VTP advertisements to VTP clients and VTP server. VLANs that are created, renamed, or deleted on transparent switches are local to that switch only.

Each switch in a VTP domain sends periodic VTP advertisements so that its neighbors can update VLAN configuration. VTP includes three types of advertisements:

- **Summary advertisements** – These inform adjacent switches of VTP domain name and configuration revision number. By default, Cisco switches issue summary advertisements *every five minutes*.
- **Advertisement request** – These are in response to a summary advertisement message when the summary advertisement contains a higher configuration revision number than the current value.
- **Subset advertisements** – These contain VLAN information including any changes.

## 7.2 Operations

When the switch receives a summary advertisement packet, it compares the VTP domain name to its own VTP domain name. If the name is different, the switch simply ignores the packet. If the name is the same, the switch then compares the configuration revision number of the packet to its own. If the switch's revision number is lower, it means that its VLAN database is out-dated, and it sends an advertisement request to ask for updates (subset advertisement messages). If the switch's revision number is *not lower* to that of the packet, then the packet is ignored.

The configuration revision number reports any VLAN changes. When you add, delete, or change a VLAN on the VTP server, the VTP server increments the configuration revision. Then, the VTP server issues summary advertisements to announce that there have been some changes occur. Next, subset advertisements that contain information about these changes are sent to all VTP clients. This process is shown in the figure 7.1.

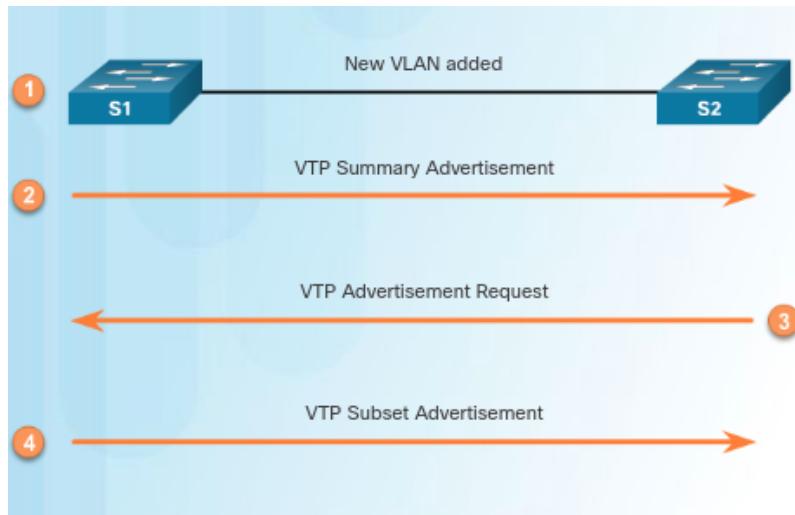


Figure 7.1: VTP operation

## 7.3 VTP Caveats

Suppose there is a new switch with a higher configuration revision number, and you want to install it into the existing switched network. Then, the existing VLAN configurations will be wiped out (see figure 7.2). Therefore, when a switch is added to a network, ensure that it has a default VTP configuration, or its revision number is reset to 0.

The VTP configuration revision number is stored in NVRAM and is not reset if you erase switch configuration and reload it. To reset VTP configuration revision number to zero you have two options:

- Change the switch's VTP domain to a nonexistent VTP domain and then change the domain back to the original name.
- Change the switch's VTP mode to transparent and then back to previous VTP mode (Recommended).

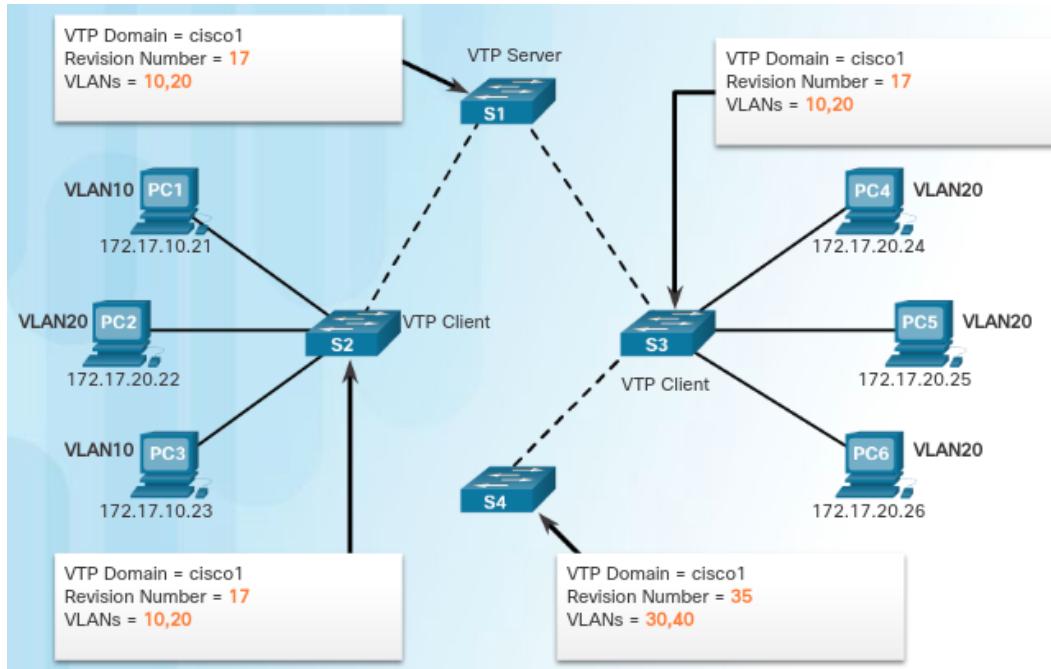


Figure 7.2: Incorrect VTP configuration revision number scenario



# Chapter 8

## STP

### 8.1 Layer 2 loop

Multiple cabled paths between switches provide physical redundancy. However, when redundancy is introduced into a design, loops and duplicate frames occur. Loops and duplicate frames have severe consequences for network:

- Duplicate unicast frames
- MAC database instability
- Broadcast storm

**MAC Database Instability:** Broadcast frames are forwarded out all switch ports, except the original ingress port. If there is more than one path to the destination, the frames may be forwarded back to the original switch, and create an endless loop. When a loop occurs, the MAC address table constantly changes with the updates from the broadcast frames, which results in MAC database instability. See this [link](#) for more explanation.

**Duplicate unicast frame:** MAC Database Instability makes the switch overwhelming and confused. It does not know destination MAC address and must forward the frame out all ports. Consequently, unicast frames arrive at the destination device more than once.

**Broadcast storm** occurs when there are so many broadcast frames caught in a Layer 2 loop. Consequently, all available bandwidth is consumed. No bandwidth is available for legitimate traffic and the network becomes unavailable for data communication. This is an effective denial of service (DoS). Broadcast storm also cause the end device to malfunction because of the processing requirements needed to sustain such a high traffic load on the NIC.

### 8.2 STP overview

#### 8.2.1 Operation

The Spanning Tree Protocol (STP) was developed to prevent loops and duplicate frames. STP ensures that there is only one logical path between all destinations on the network by blocking redundant paths that could cause a loop. If failure occurs, STP recalculates the paths and unblocks the necessary ports to allow the redundant path to become active. Below are some options for STP:

- **802.1D** – The first STP version, one spanning tree instance for the entire bridged network, regardless of the number of VLANs.
- **PVST+** – Cisco enhancement of STP, provide a separate spanning tree instance for each VLAN.
- **RSTP or 802.1w** – An standards-based evolution of STP.
- **Rapid PVST+** – Cisco enhancement of RSTP, provide a separate spanning tree instance for each VLAN.

- **MSTP** – Multiple Spanning Tree Protocol, an IEEE standard, map multiple VLANs into the same spanning tree instance. It can also ride on top of another spanning-tree protocol.
- **MST** – Cisco implementation of MSTP, provide up to 16 instances of RSTP.

### 8.3 Bridge ID

Bridge ID (BID) uniquely identifies a switch. The BID value is determined by the combination of three fields: *bridge priority*, *MAC address of the sending switch*, and *extended system ID*. The **extended system ID** reserves the rightmost 12 bits for the VLAN ID and the far left 4 bits for the bridge priority. Therefore, the **bridge priority** value can only be the multiples of 4096 ( $2^{12}$ ). If we consider BID as a hexadecimal number, the lowest left-most 4 bits (meaning lowest priority) always leads to lowest BID value.

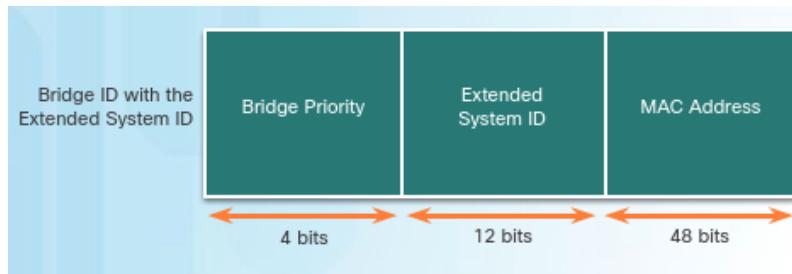


Figure 8.1: BID fields

### 8.4 BPDU frame

A Bridge Protocol Data Units (BPDU) is a frame exchanged by switches for STP. BPUDUs have a destination MAC address of **01:80:C2:00:00:00**, which is a multicast address for the spanning tree group. A BPDU frame contains 12 distinct fields:

- The first four fields identify the protocol, version, message type, and status flags.
- The next four fields are *root ID*, *bridge ID (BID)*, path cost.
- The last four fields are all timer fields. They determine how frequently BPDU messages are sent and how long the information received through the BPDU process is retained.

**Root ID** indicates the BID of the root bridge. When a switch first boots, the root ID is the same as the BID. However, as the election occurs, the lowest BID replaces the root ID to identify the root bridge.

**Path cost** is the cost of the path from the sending switch to the root bridge. When a switch receives the BPDU, it adds the ingress port cost of the segment to determine its internal root path cost. The path cost is determined by summing up the individual port costs along the path from the switch to the root bridge. The default port costs are defined by the speed at which the port operates.

### 8.5 Root bridge election

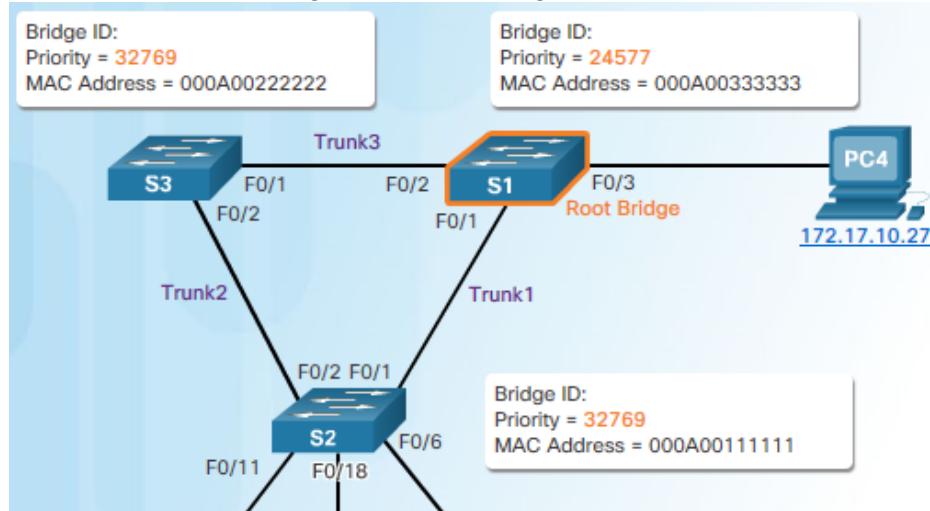
There is a root bridge elected for each spanning tree instance. The switch with the lowest priority is the root bridge. If the priorities are equal, then the switch with the lowest MAC address is elected the root bridge. All switches in the broadcast domain participate in the election process. After a switch boots, it begins to send out BPDU frames every two seconds. These BPUDUs contain the switch BID and the root ID.

Assuming that switch S1, S2, and S3 resides in the same broadcast domain (figure 8.2). As S1 forward a BPDU frame to S2, S2 reads the root ID from the BPDU frame, which is 24577-000A. Because the root ID from the BPDU

Table 8.1: Default path cost

Port and speed	Cost
10 Gb/s Ethernet	2
1 Gb/s Ethernet	4
100 Mb/s Ethernet	19
10 Mb/s Ethernet	100

Figure 8.2: Root bridge selection



frame is lower than the that of S2, which is 32769-000A.., then S2 updates its root ID to 24577-000A.. S2 then forwards BPDU frame with the newly updated root ID to S3. The same process repeats, S3 compares its current root ID with the root ID identified in the frames, and then updates its current root ID if needed. Eventually, the root ID of all switches equals 24577-000A.. (the lowest BID). Therefore, all switches know exactly which one is the root bridge, based on the root ID. At this point, the election is complete.

## 8.6 Port Roles

### 8.6.1 What is port role?

Spanning Tree Algorithm (STA) determines which switch ports on a network must be blocked to prevent loops. It designates a single switch as the root bridge and uses it as the reference point for all path calculations (see figure 8.3). After the root bridge has been determined, STA assigns port roles to the switch ports:

- **Root port** – Switch ports closest to the root bridge in terms of overall cost to the root bridge.
- **Designated port** – All non-root ports that are still permitted to forward traffic on the network.
- **Alternate and backup ports** – Alternate ports and backup ports are blocked to prevent loops.

STA determines port roles on the following rules:

- There can only be one root port per non-root switch
- If one end of a segment (the link between two switches) is a root port, then the other end is a designated port.
- All ports on the root bridge are designated ports.
- Alternate ports are elected only on segments where neither end is a root port.

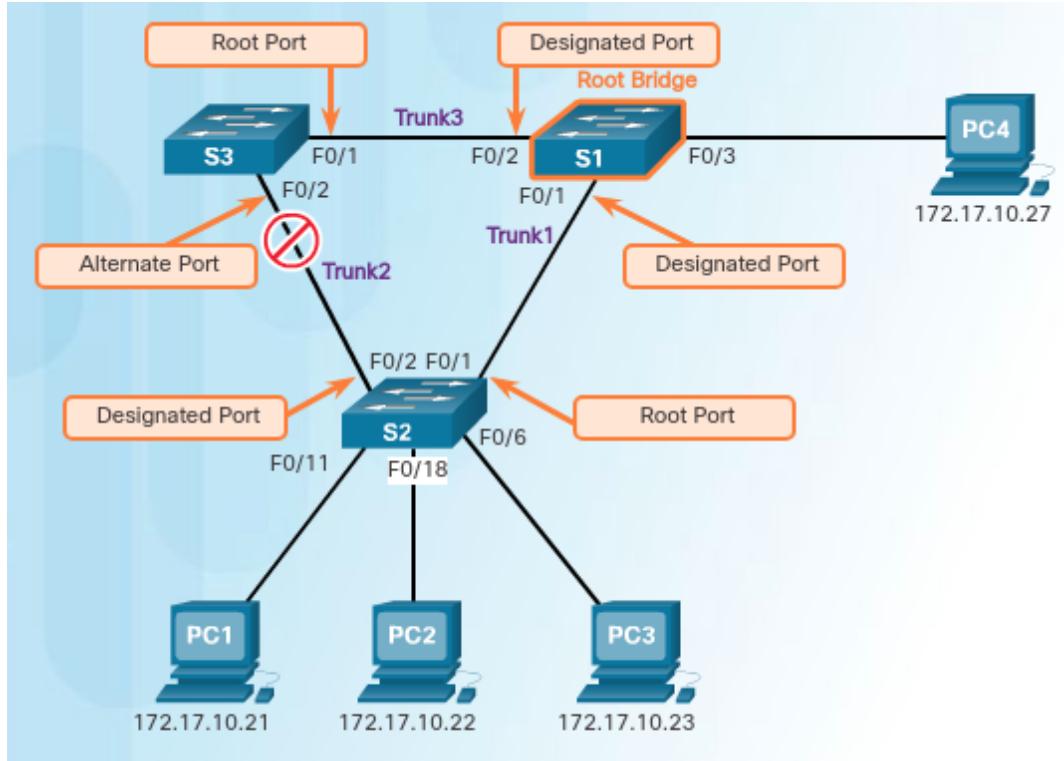


Figure 8.3: The STA designates a single switch as the root bridge

After the root bridge is elected, the STA determines port roles (figure 8.4) on the following steps in sequential order:

1. The root bridge automatically configures all of its switch ports in the designated role.
2. STP determines root port role. On each non-root switch, the port with lowest path cost to root bridge will become root port.
3. Non-root switches configure their non-root ports as designated or alternate ports.

In the last step, two ports on the same segment will negotiate with each other. If one end of a segment is a root port, then the other end is a designated port. In figure 8.4, the segment between S1(F0/1) and S2(F0/1) is an example.

However, when neither end of a segment cannot be a root port, the two switches on that segment exchange BPDU frames to decide which port to configure as a designated port. The switch with the lower path cost to the root bridge will have its port configured as a designated port. The other port will become an alternate port. If the path costs are equal, then the switch with the lower BID has its port configured as a designated port while the other has its port configured as an alternate port.

If there are more than two paths in a segment (figure 8.5), then the port with the lower number (e.g. port number of F0/3 is lower than F0/4) will become active, while the other will be blocked.

## 8.7 PVST+

### 8.7.1 Port state

To facilitate the learning of the logical spanning tree, each switch port transitions through five possible port states:

- **Blocking:** The port is an alternate port and does not participate in frame forwarding. The port receives BPDU frames to determine the location and root ID of the root bridge switch and which port roles each switch port should assume in the final active STP topology.

Figure 8.4: Port role assignment

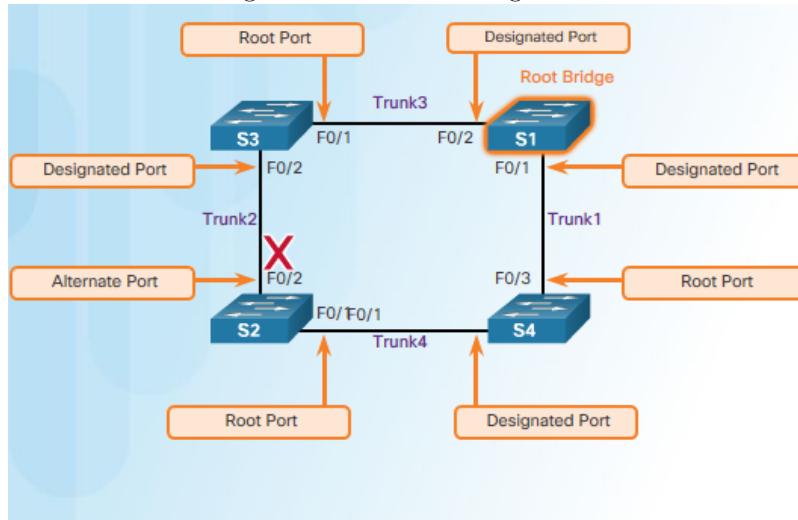
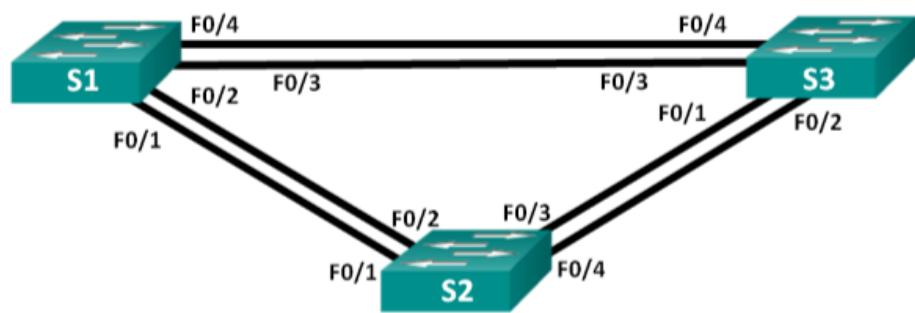


Figure 8.5: Redundant links



- **Listening:** The port is preparing to participate in frame forwarding. It listens for the path to the root. It also processes its own BPDU frames, and broadcast them to inform adjacent switches that the switch port is about to join the active topology.
- **Learning:** The port learns the MAC addresses and populates the MAC address table.
- **Forwarding:** The port is considered part of the active topology. It forwards data frames and sends and receives BPDU frames.
- **Disabled:** The Layer 2 port does not participate in spanning tree and does not forward frames. The disabled state is set when the switch port is administratively disabled.

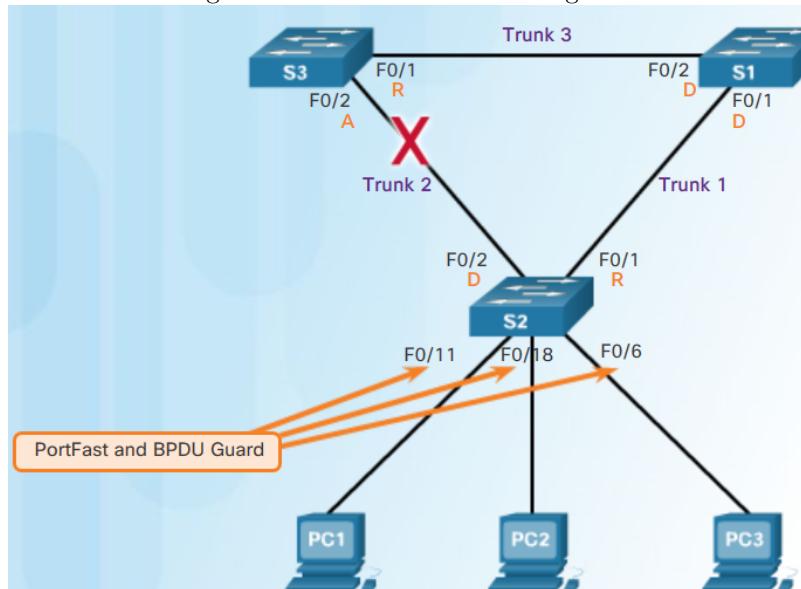
Table 8.2: Five port states in PVST+

	Port states				
Operation allowed	Blocking	Listening	Learning	Forwarding	Disabled
Receive BPDUs	*	*	*	*	
Process BPDUs		*	*	*	
Learn MAC addresses			*	*	
Forward data frames				*	

### 8.7.2 PortFast and BPDU guard

When a switch port is configured with PortFast that port transitions from blocking to forwarding state immediately, bypassing the usual transition states (the listening and learning states). You can use PortFast on *access ports*<sup>1</sup> to help devices connect to the network immediately, rather than waiting for STP to converge on each VLAN.

Figure 8.6: PortFast and BPDU guard



Because a PortFast-configured port is not connected to a switch, BPDUs should never be received. Otherwise, the broadcast domain will “think” that the port is connected to a switch. This consumption potentially causes layer 2 loops. To address this issue, Cisco switches support a feature called BPDU guard. This will effectively shut down

<sup>1</sup>Access ports are ports which are connected to an end device such as a PC or a server.

the port. The BPDU guard feature provides a secure response to invalid configurations because you must manually put the interface back into service.

Cisco PortFast technology is useful for DHCP. Because PortFast immediately changes the state to forwarding, the PC always gets a usable IP address from DHCP server.

## 8.8 RSTP and Rapid PVST+

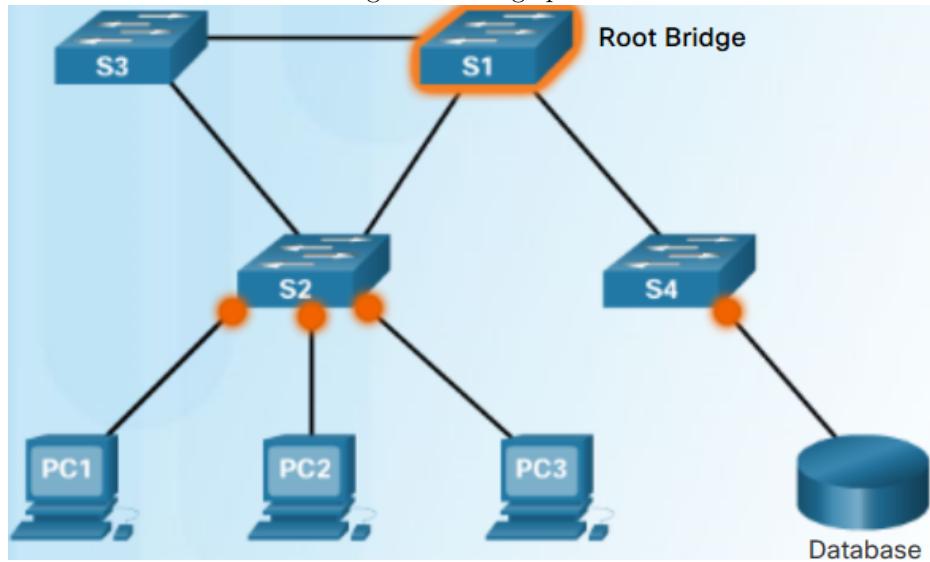
Rapid PVST+ is Cisco enhancement of RSTP. In this section, only the common features of these two protocols are discussed.

**BPDU:** RSTP uses BPDU version 2, while 802.1D uses version 0. Protocol information can be immediately aged on a port if Hello packets are not received for three consecutive Hello times (six seconds, by default) or if the max age timer expires. BPDUs are used as a keepalive mechanism. Therefore, three consecutively missed BPDUs indicate lost connectivity between a bridge and its neighboring.

RSTP expands the STP port roles by adding the alternate and backup roles. It also combine blocking and listening port state to allow faster convergence. Port states are defined as *discarding*, *learning*, or *forwarding*. By doing this, RSTP speeds the recalculation of the spanning tree when the Layer 2 network topology changes. If a port is configured to be an alternate port, it can immediately change to a forwarding state without waiting for the network to converge.

**Edge port:** RSTP introduces new types of port: edge port. An RSTP edge port is a switch port that is never intended to be connected to another switch. It immediately transitions to the forwarding state when enabled. An edge port is any port that does not have a switch connected to it.

Figure 8.7: Edge ports



## 8.9 Configuration

The default priority value of a switch is 32768. When an administrator wants a specific switch to become a root bridge, the bridge priority value must be adjusted to ensure it is lower than the bridge priority values of all the other switches on the network. By default, the priority for the root switch configured by `root primary` option is **24576**. The `root secondary` sets the priority of the switch to **28672**.

**Listing 13: STP**

```
spanning-tree mode pvst
spanning-tree vlan 10 root secondary
spanning-tree vlan 20 root primary

spanning-tree portfast default
spanning-tree portfast bpduguard default

interface F0/1
  spanning-tree portfast
  spanning-tree bpduguard enable
end

show spanning-tree
show spanning-tree vlan 10
show running-config interface f0/1
```

The command `spanning-tree vlan 10 priority 24576` can also be used to configure bridge priority. This command gives more granular control over the bridge priority value. Remember the priority value is configured in increments of 4,096 between 0 and 61,440. Cisco recommends caution when using this command.

The fourth command enables PortFast on all nontrunking interfaces, and the next command enables BPDU guard on all PortFast-enabled ports.

# Chapter 9

## EtherChannel

### 9.1 Introduction

EtherChannel creates an aggregation that is seen as one logical link, which provides redundancy and load balancing. Most configuration tasks can be done on the EtherChannel interface instead of each individual port, ensuring configuration consistency throughout the links. EtherChannel implementation is cost effective, because this protocol relies on existing switch ports, which means there is no need to upgrade the link to a faster and more expensive connection to have more bandwidth.

**PAgP** (pronounced “Pag – P”) is a **Cisco-proprietary** protocol that aids in the creation of EtherChannel links. PAgP helps create the EtherChannel link by detecting the configuration of each side and ensuring that links are compatible so that the EtherChannel link can be enabled when needed. PAgP can be configured in one of three models:

- **On** – This mode forces the interface to channel without PAgP. Interfaces configured in this mode do not exchange PAgP packets.
- **PAgP Desirable** – The interface in this mode *initiates negotiations* with other interfaces by sending PAgP packets.
- **PAgP Auto** – The interface in this mode only responds to the PAgP packets that it receives, but does not initiate PAgP negotiation.

The modes must be compatible on each side as shown in table 9.1. Otherwise, the status of port-channel will become **err-disabled**.

Table 9.1: PAgP Establishment

S1	S2	EtherChannel establishment
Desirable	Auto/Desirable	Yes
On	On	Yes
Auto	Auto/On	No
Not configured	Auto/Desirable/On	No
Desirable	on	No

**LACP** is a **standard-based** protocol and is a part of an *802.3ad* that aids in the creation of EtherChannel links. Because LACP is an IEEE standard, it can be used to facilitate EtherChannels in *multivendor* environments, including Cisco devices. LACP allows for eight active links, and also eight standby links. A standby link will become active should one of the current active links fail. PAgP can be configured in one of three models:

- **On** – This mode forces the interface to channel without LACP. Interfaces configured in this mode do not exchange LACP packets.

- **LACP Active** – The interface in this mode *initiates negotiations* with other interfaces by sending LACP packets.
- **LACP Passive** – The interface in this mode only responds to the LACP packets that it receives, but does not initiate LACP.

The modes must be compatible on each side as shown in table 9.2.

Table 9.2: LACP Establishment

S1	S2	EtherChannel establishment
Active	Passive/Active	Yes
On	On	Yes
Passive	Passive/On	No
Not configured	Passive/Active/On	No
Active	on	No

## 9.2 Configuration

### 9.2.1 Restrictions

The EtherChannel provides full-duplex bandwidth between one switch and another switch or host. Currently each EtherChannel can consist of up to eight compatibly-configured Ethernet ports. However, interface types cannot be mixed. For example, Fast Ethernet and Gigabit Ethernet cannot be mixed within a single EtherChannel.

EtherChannel creates a one-to-one relationship; that is, one EtherChannel link connects only two devices. The individual EtherChannel group member port configuration must be consistent on both devices:

- **EtherChannel support:** All Ethernet interfaces on all modules must support EtherChannel with no requirement that interfaces be physically contiguous, or on the same module.
- **Speed and duplex:** Configure all interfaces in an EtherChannel to operate at the same speed and in the same duplex mode.
- **VLAN match:** All interfaces in the EtherChannel bundle must be assigned to the same VLAN. If the physical ports of one side are configured as trunks, the physical ports of the other side must also be configured as trunks within the same native VLAN.
- **Range of VLANs:** An EtherChannel supports the same allowed range of VLANs on all the interfaces in a trunking EtherChannel. If the allowed range of VLANs is not the same, the interfaces do not form an EtherChannel, even when set to auto or desirable mode.

### 9.2.2 Configuration

1. Specify the interfaces that compose the EtherChannel group using the `interface range <interface>` global configuration mode command. A good practice is to start by shutting down those interfaces, so that any incomplete configuration does not create activity on the link. At the end of this step, make sure that none of restrictions above are broken.
2. Create the port channel interface with a channel group number. The mode active keywords identify this as an LACP EtherChannel configuration.
3. Change layer 2 settings on port channel interface, so that two sides of the EtherChannel link have the same configuration.

**Listing 14: EtherChannel**

```
interface range f0/1-2
    shutdown
    channel-group 1 mode active
    no shutdown
interface port-channel 1
    no shutdown
    switchport mode trunk
    switchport trunk allowed vlan 1,10,20,99
```



# **Part III**

# **ROUTING TECHNOLOGIES**



# Chapter 10

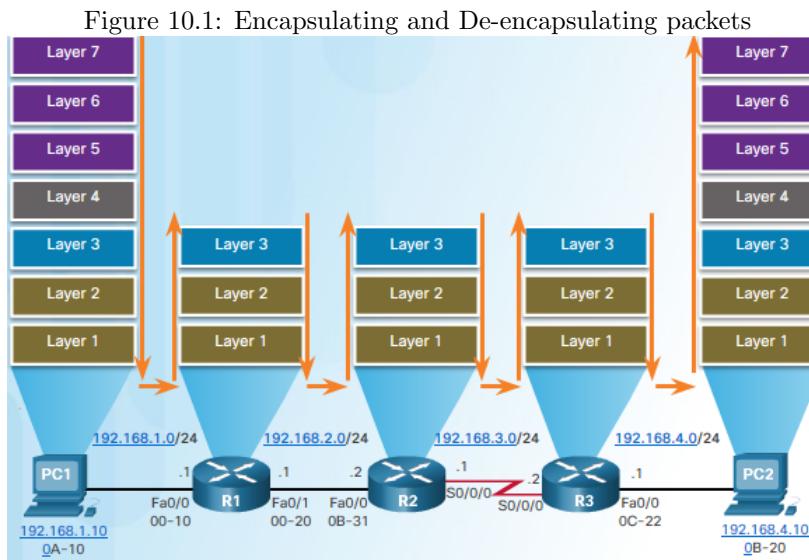
## Router

### 10.1 Introduction

The primary function of a router is: Determine the best path to send packets and Forward packets toward their destination. There are three Packet-Forwarding Mechanisms:

- **Process switching:** Process switching starts finding a path every time a packet arrives, even if the destination is the same for a stream of packets. This mechanism is slow and old-fashioned.
- **Fast switching:** The flow information for the packet is also stored in the *fast-switching cache*. If another packet going to the same destination arrives on an interface, the next-hop information in the cache is reused without CPU intervention.
- **Cisco Express Forwarding (CEF):** CEF builds a Forwarding Information Base (FIB), and an adjacency table, then stores them in the data plane. The FIB contains precomputed reverse lookups, next-hop information for routes including the interface, and Layer 2 information. Therefore, when a network has converged, the FIB and adjacency tables contain all the information a router would have to consider when forwarding a packet. This enables forwarding of packets to occur at the data plane without consulting the control plane.

The router performs the following three major steps (Figure 10.1):



1. De-encapsulates the Layer 2 frame header and trailer to expose the Layer 3 packet.
2. Examines the destination IP address of the IP packet to find the best path in the routing table.

3. If the router finds a path to the destination, it encapsulates the Layer 3 packet into a new Layer 2 frame and forwards the frame out the exit interface. MAC addresses are only required on Ethernet multiaccess networks. A serial link is a point-to-point connection and uses a different Layer 2 frame that does not require the use of a MAC address. Because there are no MAC addresses on serial interfaces, a router sets the data link destination address to an equivalent of a broadcast.

## 10.2 Path determination

A metric is the quantitative value used to measure the distance to a given network. The best path to a network is the path with the lowest metric. The following lists some dynamic protocols and the metrics they use:

- RIP – hop count
- OSPF – bandwidth
- EIGRP – Bandwidth, delay, load, reliability

When a router has two or more paths to a destination with equal cost metrics, then the router forwards the packets using both paths equally. This is called equal cost **load balancing**. The routing table contains the single destination network but has multiple exit interfaces, one for each equal cost path. Only *EIGRP* supports *unequal cost load balancing*.

A router uses what is known as the **administrative distance (AD)** to determine the route to install into the IP routing table. The AD represents the trustworthiness of the route; the lower the AD, the more trustworthy the route source. For example, when a router has the choice of a static route (AD = 1) and an EIGRP route (AD = 90), the static route takes precedence. The following list shows the AD value of common routes:

- Connected interface = 0
- Static route = 1
- BGP = 20
- Internal EIGRP = 90
- OSPF = 110
- RIP = 120
- External EIGRP = 170
- Internal BGP = 200

# Chapter 11

## IPv6

### 11.1 Representation

IPv6 addresses are 128 bits in length and written as a string of 32 hexadecimal values. Every 4 bits is represented by a single hexadecimal digit. The preferred format for writing an IPv6 address is  $x:x:x:x:x:x:x:x$ , where each “ $x$ ” is a single *hextet*<sup>1</sup>. For example,

2001:0DB8:0000:1111:0000:0000:0000:0200

There are two rules to help reduce the number of digits needed to represent an IPv6 address.

- **Omit leading 0s:** Omit any leading 0s (zeros) in any hextet. This rule only applies to leading 0s, NOT to trailing 0s; otherwise, the address would be ambiguous. For example, the hextet `0ABC` will become `ABC`.

2001:0DB8:0000:1111:0000:0000:0000:0200

2001: DB8: 0:1111: 0: 0: 0: 200

- **Omit 0 segments:** Double colon (:) can replace any contiguous string of one or more hextets consisting of all 0s. The double colon (:) can only be used once within an address; otherwise, there would be more than one possible resulting address.

2001:0DB8:0000:1111:0000:0000:0000:0200

2001: DB8: 0:1111: 0: 0: 0: 200

2001: DB8: 0:1111::200

### 11.2 Types of IPv6 Addresses

There are three types of IPv6 addresses: Unicast, Multicast, and Anycast. Unlike IPv4, IPv6 does not have a broadcast address. However, there is an IPv6 all-node multicast address that essentially gives the same result.

#### 11.2.1 Unicast IPv6

An IPv6 unicast address uniquely identifies an interface on an IPv6-enabled device. There are three types of IPv6 unicast address: Global unicast, Link-local and Unique local unicast.

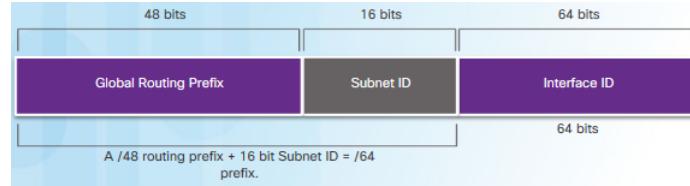
**Global unicast:** A global unicast address is a globally unique and Internet-routable address. The ICANN<sup>2</sup> assigns IPv6 Global Unicast address to organization. A global unicast address has three parts: Global routing prefix, Subnet ID, Interface ID (Figure 11.1). The **global routing prefix** (first three hextets) is the network portion of the IPv6 address that is assigned by the ISP. The **Subnet ID** (fourth hextet) is used by an organization to identify subnets within its site. The **Interface ID** (last four hextets) is the host portion of the address.

---

<sup>1</sup>four hexadecimal values

<sup>2</sup>Internet Committee for Assigned Names and Numbers

Figure 11.1: Global routing prefix



**Link-local:** Link-local addresses are used to communicate with other devices on the same local link<sup>3</sup>. Packets with a source or destination link-local address cannot be routed beyond the link from which the packet originated. Their uniqueness must only be confirmed on that link. If a link-local address is not configured manually on an interface, the device will automatically create its own. IPv6 link-local addresses are in the **FE80::/10** range<sup>4</sup>.

**Dynamic Link-Local Addresses** A link-local address can be established dynamically or configured manually as a static link-local address. Operating systems will typically use either EUI-64 process or a randomly generated 64-bit number to dynamically assign link-local address. By default, Cisco routers use EUI-64 to generate the Interface ID for all link-local address on IPv6 interfaces. For serial interfaces, the router will use the MAC address of an Ethernet interface.

**Static Link-Local Addresses** A drawback to using the dynamically assigned link-local address is its long interface ID, which makes it challenging to identify and remember assigned addresses. Configuring the link-local address manually provides the ability to create an address that is recognizable and easier to remember.

```
R1(config)# interface g0/0
R1(config-if)# ipv6 address fe80::1 link-local
```

The link-local address in the above example is used to make it easily recognizable as belonging to router R1. The same IPv6 link-local address is configured on all of R1's interfaces. FE80::1 can be configured on each link because it only has to be unique on that link. Similar to R1, router R2 would be configured with FE80::2 as the IPv6 link-local address on all of its interfaces

**Unique local unicast:** Unique local addresses are used for local addressing within a site or between a limited number of sites. These addresses should not be Internet-routable and should not be translated to a global IPv6 address. Unique local addresses can be used for devices that will never need or have access from another network. Unique local addresses are in the range of **FC00::/7 to FDFF::/7**.

### 11.2.2 Multicast IPv6

IPv6 multicast addresses have the prefix **FF00::/8**. Multicast addresses can only be destination addresses and not source addresses. There are two types of IPv6 multicast addresses: Assigned multicast address and Solicited node multicast address.

**Assigned mulicast addresses** are reserved multicast addresses for predefined groups of devices. For example, **FF02::1** is the all-nodes multicast address, which has the same effect as broadcast IPv4 address. The all-routers multicast address **FF02::2** identifies a group of all IPv6 routers<sup>5</sup> on the network.

**A solicited-node multicast address** is mapped to a special Ethernet multicast address. This allows the Ethernet NIC to filter the frame by examining the destination MAC address.

<sup>3</sup>With IPv6, the term link refers to a subnet.

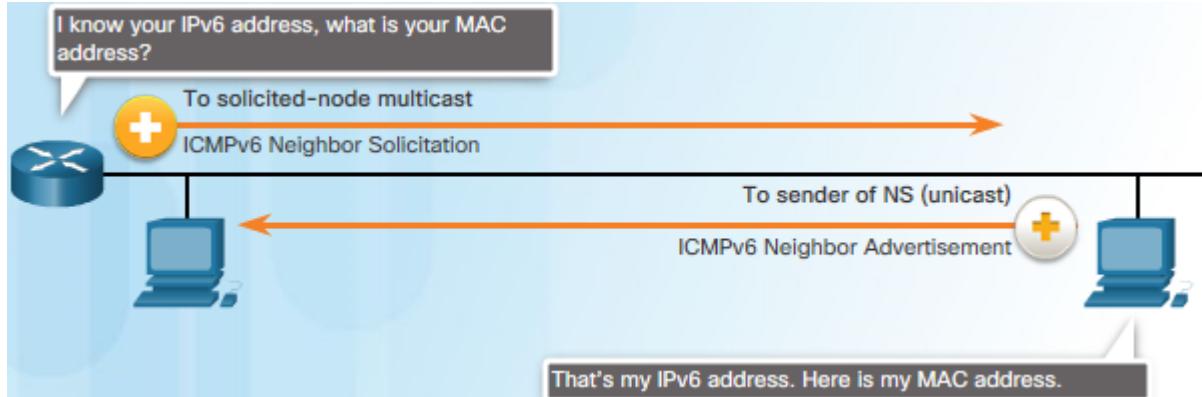
<sup>4</sup>The /10 indicates that the first 10 bits are 1111 1110 10xx xxxx. The first hextet has a range of 1111 1110 1000 0000 (FE80) to 1111 1110 1011 1111 (FEBF)

<sup>5</sup>A router with the ipv6 unicast-routing global configuration command executed

## 11.3 ICMPv6

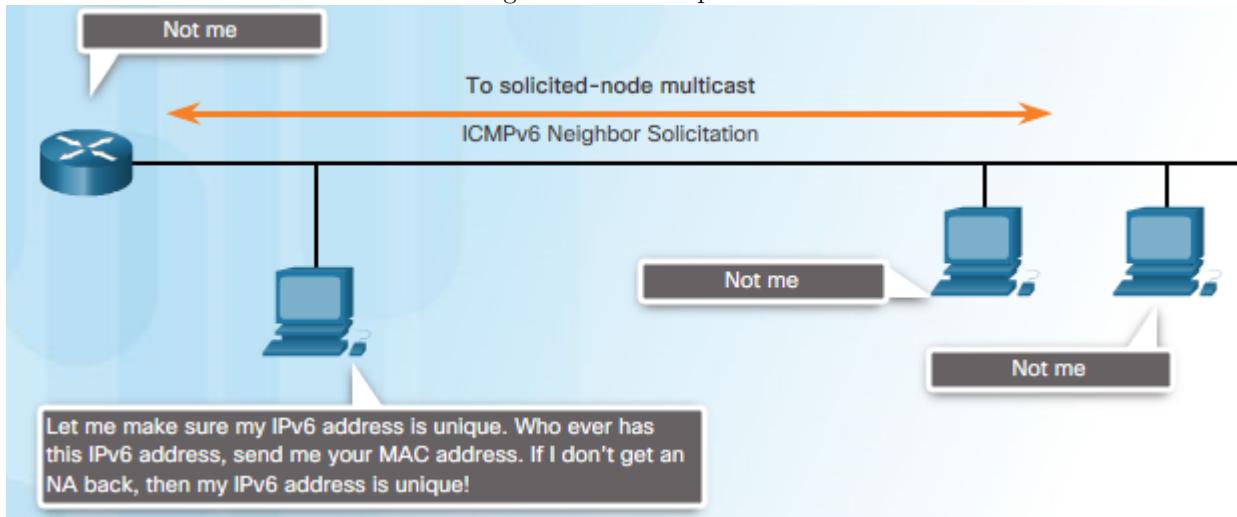
The informational and error messages found in ICMPv6 are very similar to the control and error messages implemented by ICMPv4. However, ICMPv6 has new features and improved functionality not found in ICMPv4. ICMPv6 includes four types of message: RS and RA message<sup>6</sup> (communication between a router and a device), NS and NA message<sup>7</sup> (communication between devices). Address resolution and DAD process use NA and NS message, while RS and RA message contribute to assigning IPv6 to devices.

Figure 11.2: Address Resolution



**Address resolution** acts like ARP of IPv4. It is used to determine the MAC address of a destination IPv6 address. A device will send an NS message to the solicited node address to ask for MAC address. The message will include the known (targeted) IPv6 address. The device that has the targeted IPv6 address will respond with an NA message containing its Ethernet MAC address. Figure 11.2 shows two PCs exchanging NS and NA messages.

Figure 11.3: DAD process



**DAD process** is a Duplicate Address Detection. To check the uniqueness of an address, the device will send an NS message with its own IPv6 address as the targeted IPv6 address, shown in Figure 11.3. If another device on the network has this address, it will respond with an NA message. This NA message will notify the sending device that the address is in use. If a corresponding NA message is not returned within a certain period of time, the unicast address is unique and acceptable for use.

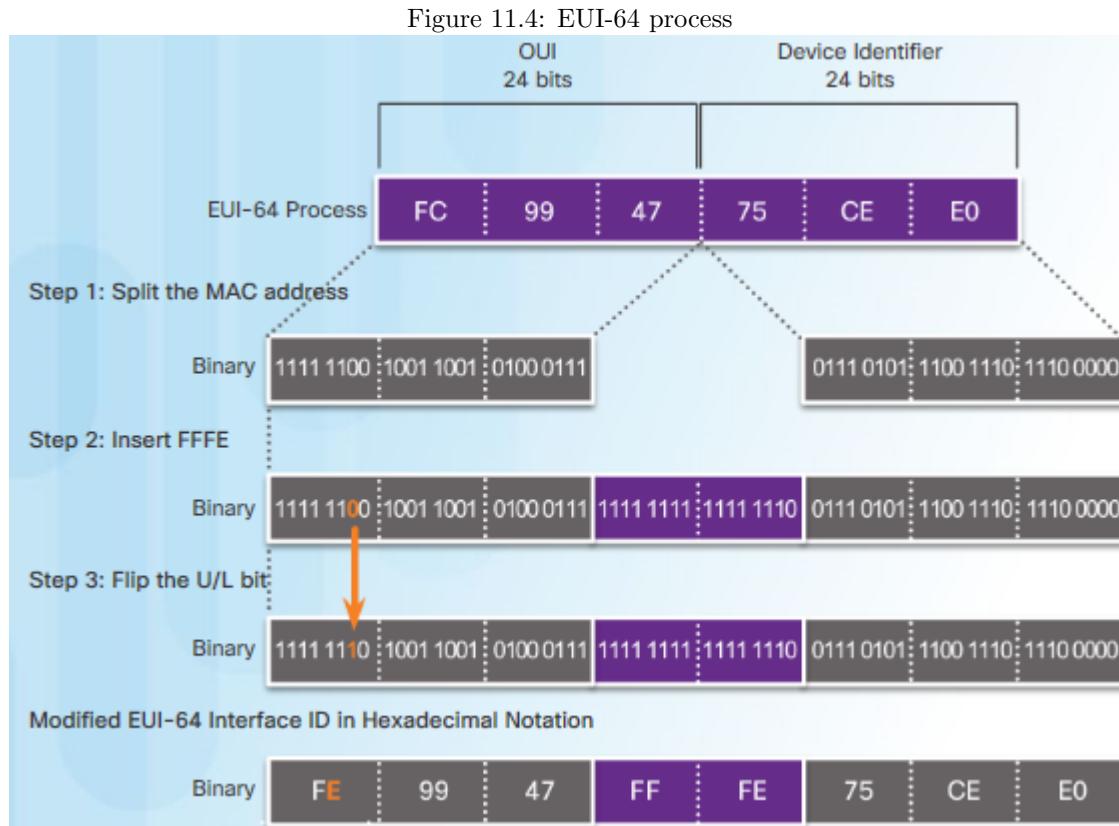
<sup>6</sup>Router Solicitation (RS) message, Router Advertisement (RA) message

<sup>7</sup>Neighbor Solicitation (NS) message, Neighbor Advertisement (NA) message

## 11.4 EUI-64 Process

When the RA message is either SLAAC or SLAAC with stateless DHCPv6, the client must generate its own Interface ID using EUI-64 Process or Randomly Generated. An EUI-64 Interface ID is represented in binary and is made up of three parts (Figure 11.4):

- **24-bit OUI** from the client MAC address<sup>8</sup>, but the 7th bit (the Universally/Locally (U/L) bit) is reversed. This means that if the 7th bit is a 0, it becomes a 1, and vice versa.
- The inserted 16-bit value **FFFE** (in hexadecimal).
- **24-bit Device Identifier** from the client MAC address.



The advantage of EUI-64 is the Ethernet MAC address can be used to determine the Interface ID. It also allows network administrators to easily track an IPv6 address to an end device using the unique MAC address. However, this has caused privacy concerns among many users. They are concerned that their packets can be traced to the actual physical computer. Due to these concerns, a randomly generated Interface ID may be used instead.

## 11.5 IPv4 and IPv6 Coexistence

Several techniques have been developed to accommodate a variety of transition IPv4-to-IPv6 scenarios:

- **Dual-stack:** A device interface is running both IPv4 and IPv6 protocols enabling it to communicate with either network.
- **Tunneling:** The process of encapsulating an IPv6 packet inside an IPv4 packet. This allows the IPv6 packet to be transmitted over an IPv4-only network.
- **Translation:** NAT64 allows IPv6-enabled devices to communicate with IPv4-enabled devices using a translation technique similar to NAT for IPv4. An IPv6 packet is translated to an IPv4 packet and vice versa.

<sup>8</sup>Ethernet MAC addresses are made up of two parts: vendor code OUI assigned by IEEE and device identifier

# Chapter 12

## Static routing

### 12.1 Overview

Table 12.1: Dynamic vs Static routing

Feature	Dynamic routing	Static routing
Configuration complexity	Independent of network size	Increase with network size
Topology changes	Automatically adapt to changes	Administrator intervention required
Usage	Complex and growing network, normal route	Small network, Stub network, default route, summary route, backup route
Resource	CPU, memory, bandwidth	No resources needed
Predictability	Depends on current topology	Always the same
Security	Advertise protocol message over the entire network, resulting security risk	Not advertised over the network, resulting in better security
Configuration	simple, nearly error-free	error-prone, require complete knowledge of the whole network for proper configuration

### 12.2 IPv4 route configuration

Static routes are configured using the `ip route` global configuration command. The basic syntax for the command is as follows:

```
Router(config)# ip route network-addr subnet-mask {ip-address | exit-intf} [distance]
```

The `network-addr` and `subnet-mask` are the IP address and subnet mask of the destination network. The `distance` parameter specifies AD. By default, static route has AD of 1.

The next hop can be identified by an IP address (`ip-address`), exit interface (`exit-intf`), or both. The way the destination is specified creates one of the three following route types:

- **Next-hop** static route: Only the next-hop IP address is specified.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.2
```

- **Directly connected** static route: Only the router exit interface is specified.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 s0/0/0
```

- **Fully specified** static route: The next-hop IP address and exit interface are specified. When the exit interface is a multiaccess network (e.g. Ethernet network), it is recommended that a fully specified static route be used.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 g0/0 172.16.2.2
```

A route that does not reference an exit interface must perform a *recursive lookup*. Recursive lookup means that the router searches through the routing table to find the interface that matches the next-hop IP address.

Along with `ping` and `traceroute`, useful commands to verify static routes include these:

```
R1# show ip route
R1# show ip route static
R1# show ip route 192.168.1.0
R1# show running-config | section ip route
```

## 12.3 IPv6 route configuration

The `ipv6 unicast-routing` global configuration command must be configured to enable the router to forward IPv6 packets. Static routes for IPv6 are configured using the `ipv6 route` global configuration command. The syntax is as follows:

```
Router(config)# ipv6 route ipv6-prefix/prefix-length {ipv6-address | exit-intf}
```

```
R1(config)# ipv6 route 2001:DB8:ACAD:2::/64 2001:DB8:ACAD:4::2
R1(config)# ipv6 route 2001:DB8:ACAD:2::/64 s0/0/0
R1(config)# ipv6 route 2001:db8:acad:2::/64 s0/0/0 fe80::2
```

**Note!** If the IPv6 static route uses an IPv6 link-local address as the next-hop address, a fully specified static route including the exit interface must be used.

Along with `ping` and `traceroute`, useful commands to verify static routes include these:

```
R1# show ipv6 route
R1# show ipv6 route static
R1# show ipv6 route 2001:db8:acad:3::
R1# show running-config | section ipv6 route
```

## 12.4 Types

There are five types static routes: Standard, Default, Summary, Static host and Floating routes.

**IPv4 Default static route** A default route is used when no other routes in the routing table match the destination IP address of the packet. It acts as the Gateway of Last Resort. The command syntax for a default static route is similar to any other static route, except that the network address is `0.0.0.0` and the subnet mask is `0.0.0.0`.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.2
```

**Note!** The asterisk (\*) next to the route with code S in the output of the `show` command. The asterisk indicates that this static route is a candidate default route, which is why it is selected as the Gateway of Last Resort.

```
R1# show ip route static
<output omitted>
Gateway of last resort is 172.16.2.2 to network 0.0.0.0
S* 0.0.0.0/0 [1/0] via 172.16.2.2
```

**IPv6 Default static route:** Unlike IPv4, IPv6 does not explicitly state that the default IPv6 is the Gateway of Last Resort. The command syntax for a default static route is similar to any other static route, except that the ipv6-prefix/prefix-length is `::/0`, which matches all routes.

```
Router(config)# ipv6 route ::/0 {ipv6-address | exit-intf}
R1(config)# ipv6 route ::/0 2001:DB8:ACAD:4::2
```

**Summary static route:** To reduce the number of routing table entries, multiple static routes can be summarized into a single static route in the following circumstances:

- The destination networks are contiguous and can be summarized into a single network address.
- All the multiple static routes use the same exit interface or next-hop IP address.

**Floating static route** Floating static routes are static routes that are used to provide a backup path, in the event of a link failure. This route is only used when the primary route is not available. To accomplish this, the floating static route is configured with a higher AD<sup>1</sup> than that of another static route or dynamic routes. The floating route is not shown in the routing table until the primary route is not available.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.2 5
R3(config)# ipv6 route ::/0 2001:db8:acad:6::2 5
```

### 12.4.1 Static host routes

A host route is an IPv4 address with a 32-bit mask or an IPv6 address with a 128-bit mask. There are three ways a host route can be added to the routing table:

- Automatically installed when an IP address is configured on the router
- Configured as a static host route
- Host route automatically obtained through other methods

Cisco IOS automatically installs a host route, also known as a *local host route*, when an interface address is configured on the router. A host route allows for a more efficient process for packets that are directed to the router itself, rather than for packet forwarding. This is in addition to the connected route, designated with a C in the routing table for the network address of the interface.

```
Branch# show ip route
Gateway of last resort is not set
    198.51.100.0/24 is variably subnetted, 2 subnets, 2 masks
C      198.51.100.0/30 is directly connected, Serial0/0/0
L      198.51.100.1/32 is directly connected, Serial0/0/0

Branch# show ipv6 route
C      2001:DB8:ACAD:1::/64 [0/0]
        via Serial0/0/0, directly connected
L      2001:DB8:ACAD:1::1/128 [0/0]
        via Serial0/0/0, receive
```

A host route can be a manually configured static route to direct traffic to a specific destination device, such as an authentication server.

```
Branch(config)# ip route 209.165.200.238 255.255.255.255 198.51.100.2
Branch(config)# ipv6 route 2001:db8:acad:2::99/128 2001:db8:acad:1::2
Branch(config)# ipv6 route 2001:db8:acad:2::99/128 s0/0/0 fe80::2
```

---

<sup>1</sup>The AD represents the trustworthiness of a route. If multiple paths to the destination exist, the router will choose the path with the lowest AD.



# Chapter 13

## EIGRP

### 13.1 Basic features

Enhanced Interior Gateway Routing Protocol (EIGRP) is a powerful *distance vector* routing protocol and is relatively easy to configure for basic networks. Features of EIGRP:

- Use DUAL algorithm
- Use RTP instead of TCP or UDP
- Partial and Bounded Updates – Sends updates only when there is a change and only to the routers that need the information
- Supports Equal and Unequal cost load balancing
- Does not encrypt the routing updates
- Authentication: Only accepts routing information from other routers with the same authentication information

**Protocol Dependent Modules (PDM)** sends and receives EIGRP packets that are encapsulated in IPv4 or IPv6. **Reliable Transport Protocol (RTP)** is responsible for both reliable and unreliable packet delivery, because EIGRP cannot use UDP or TCP. RTP can send EIGRP packets as unicast or multicast.

### 13.2 Packet types

#### 13.2.1 Hello packets

EIGRP uses Hello packets to discover other EIGRP-enabled routers on directly connected links and form EIGRP neighbor adjacencies. These packets are sent as multicast *every five seconds*. However, on slow network (e.g. multipoint, NBMA networks with access links of T1), they are sent as unicast packets every 60 seconds.

**Hold timer** determines the maximum time the router should wait to declare neighbor as unreachable. By default, the Hold timer is *three times the Hello interval*. If the hold time expires, EIGRP declares the route as down and DUAL searches for a new path by sending out queries.

Hello intervals and hold times are configurable on a per-interface basis and *do not have to match* with other EIGRP routers to establish or maintain adjacencies. If the Hello interval is changed, ensure that the Hold time value is not less than the Hello interval. Otherwise, neighbor adjacency goes down after the Hold timer expires and before the next Hello interval.

#### 13.2.2 Other packets

EIGRP sends **Update packets** to propagate routing information. These packets are sent only when necessary and to those routers that require it. This is known as *bounded update*, which minimizes the network bandwidth. Update

packets must also contain only the routing information needed, which is known as *partial update*.

EIGRP sends **Acknowledgment packets** when RTP reliable delivery is used. An EIGRP acknowledgment is an EIGRP Hello packet without any data. RTP uses reliable delivery for Update, Query, and Reply packets.

DUAL uses **Query and Reply packets** when searching for networks and other tasks. Queries can use multicast or unicast, whereas replies are always sent as unicast.

### 13.3 Encapsulating EIGRP Messages

IP packet header, EIGRP packet header, and TLV (Type, Length, Value) field are encapsulated in a Layer 2 frame (Figure 13.1). The destination MAC address of the frame is the multicast address **01-00-5E-00-00-0A**.

In the **IP packet header**, the protocol field is set to **88** to indicate EIGRP. The destination address is the multicast address **224.0.0.10** for IPv4, or **FF02::A** for IPv6.

Figure 13.1: Encapsulating EIGRP Messages

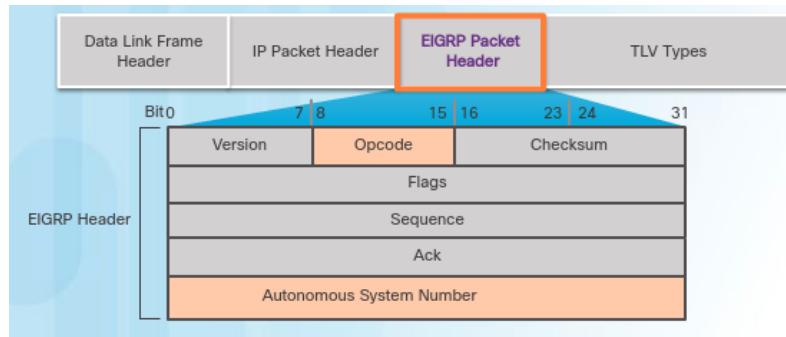


Figure 13.2: EIGRP packet header

The **EIGRP packet header** contains two important fields: Opcode and Autonomous system number (Figure 13.2). **Opcode field** specifies EIGRP packet type using number: Update(1), Query(3), Reply(4), and Hello(5). **Autonomous system (AS) number** specifies the EIGRP routing process. Unlike RIP, multiple instances of EIGRP can run on a network. The autonomous system number is used to track each running EIGRP instance.

There are three types of **TLV field**: EIGRP parameters (figure 13.3), IP internal routes (figure 13.4), and IP external routes (figure 13.5).

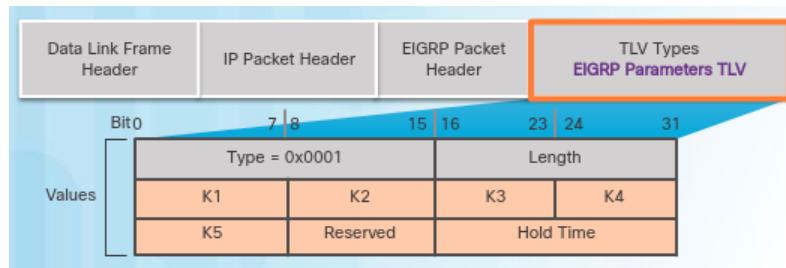


Figure 13.3: EIGRP TLV: EIGRP parameters

The *length* field identifies the size (in bytes) of the *value* field. The *value* field contains data for EIGRP message. The *type* field specifies the type of TLV: EIGRP parameters (1), IP internal routes (258), and IP external routes (259).

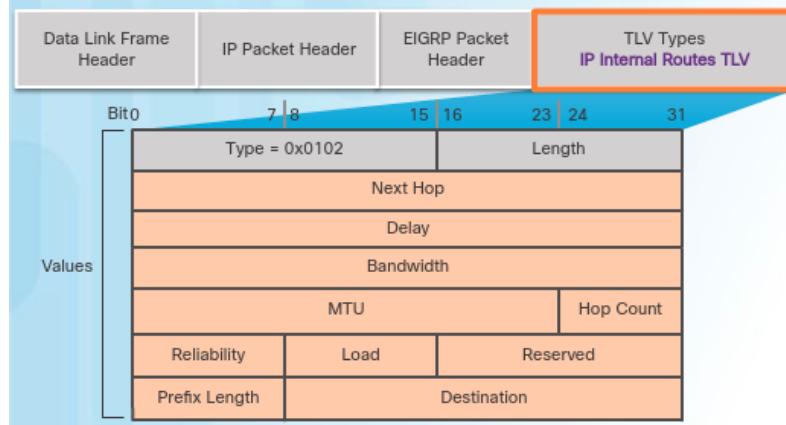


Figure 13.4: EIGRP internal routes TLV fields

The **EIGRP parameters** include the weights that EIGRP uses for its composite metric (K1 – K5). By default, only bandwidth and delay are weighted. Both are weighted equally; therefore, the K1 field for bandwidth and the K3 field for delay are both set to one (1). The other K values are set to zero (0).

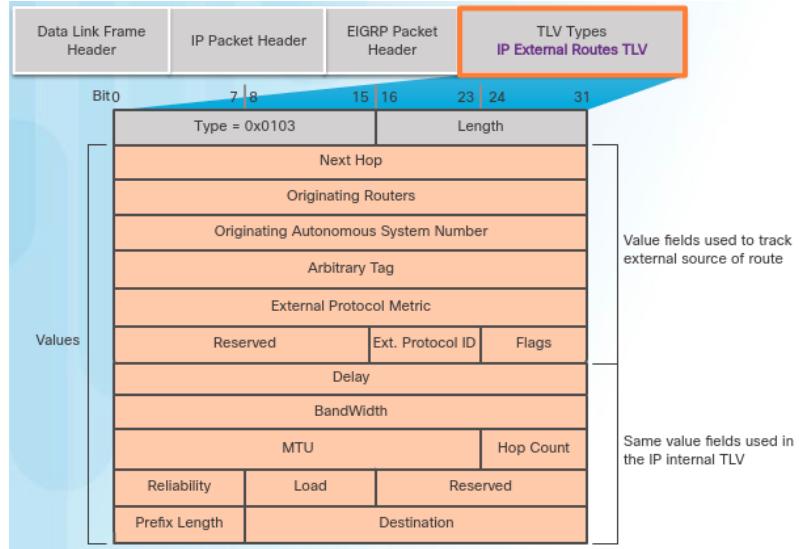


Figure 13.5: EIGRP external route TLV fields

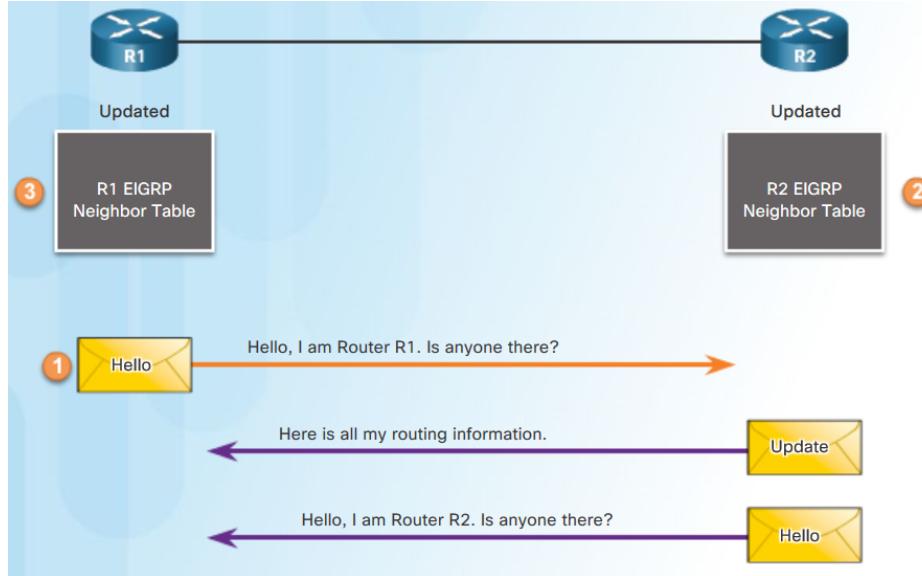
Each **IP internal route** or **IP external route** contains one route entry and the metric information for specific route. These types of TLV are included in EIGRP Update packets. The IP internal route message is used to advertise EIGRP routes within an autonomous system. The IP external message is used to import default static route, as well as routes outside the autonomous system, into the EIGRP routing process.

## 13.4 Operation

### 13.4.1 Neighbor adjacency

EIGRP uses Hello packets to establish and maintain neighbor adjacencies. To accomplish this, two EIGRP routers must use the same K values and autonomous system number.

Figure 13.6: Establish EIGRP adjacency



Each EIGRP router maintains a neighbor table, which contains a list of routers that have an EIGRP adjacency with this router. The neighbor table is used to track the status of these EIGRP neighbors.

Figure 13.6 shows two EIGRP routers exchanging initial EIGRP Hello packets. When an EIGRP enabled router receives a Hello packet on an interface, it adds that router to its neighbor table:

1. A new router (R1) comes up on the link and sends an EIGRP Hello packet through all of its EIGRP-configured interfaces.
2. Router R2 receives the Hello packet on an EIGRP-enabled interface. R2 replies with an EIGRP update packet that contains all the routes it has in its routing table, except those learned through that interface (split horizon). However, the neighbor adjacency is not established until R2 also sends an EIGRP Hello packet to R1.
3. After both routers have exchanged Hellos, the neighbor adjacency is established. R1 and R2 update their EIGRP neighbor tables adding the adjacent router as a neighbor.

### 13.4.2 Topology table

The topology table includes all destinations advertised by neighboring (adjacent) routers and the cost (metric) to reach each network. When a router receives the EIGRP update from a neighbor, it adds all update entries to its topology table. Because EIGRP update packets use RTP reliable delivery, the router replies with an EIGRP acknowledgment packet.

### 13.4.3 Metric

By default, EIGRP uses the following values in its composite metric to calculate the preferred path to a network:

- **Bandwidth** – The slowest bandwidth among all of the outgoing interfaces, along the path from source to destination.

- **Delay** – The cumulative (sum) of all interface delay along the path (in tens of microseconds).

Default composite formula:

$$\text{metric} = (\text{bandwidth} + \text{delay}) \times 256$$

Complete composite formula:

$$\text{metric} = \left( K1 \times \text{bandwidth} + \frac{K2 \times \text{bandwidth}}{256 - \text{load}} + K3 \times \text{delay} \right) \times \frac{K5}{K4 + \text{reliability}}$$

This is a conditional formula. If  $K5 = 0$ , the last term is replaced by 1. Default values for each parameter:

- $K1$  (bandwidth) = 1
- $K2$  (load) = 0
- $K3$  (delay) = 1
- $K4$  (reliability) = 0
- $K5$  (reliability) = 0

**Examining interface metric values:** The `show interfaces <interface>` command displays interface information (figure 13.7), including the parameters used to compute the EIGRP metric.

Figure 13.7: Examine interface metric values

```
R1# show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Internet address is 172.16.3.1/30
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
<output omitted>
R1#
```

- **BW** – Bandwidth of the interface (in kilobits per second).
- **DLY** – Delay of the interface (in microseconds).
- **Reliability** - Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes. By default, EIGRP does not include its value in computing its metric.
- **Txload, Rxload** – Transmit and receive load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes. By default, EIGRP does not include its value in computing its metric.

**Bandwidth:** EIGRP uses the slowest bandwidth along the path to the destination network. EIGRP divides a reference bandwidth value of  $10^7$  by the interface bandwidth value in kb/s. If the result is not a whole number, then the value is rounded down. For example,  $10^7$  divided by 1024 equals 9765.625. The .625 is dropped to yield 9765 for the bandwidth portion of the composite metric.

**Delay:** EIGRP uses the sum of all delays along the path to the destination. The sum of these delays is divided by 10. See also table 13.1 for default delay values. For example, along the path R1→R2→R3, the s0/0/1 interface on R2 has a delay of 20,000 microseconds, the g0/0 interface on R3 has a delay of 10 microseconds. The delay is  $(20000 + 10) \div 10 = 2001$ .

Table 13.1: Default delay values

Media	Delay
Ehternet	1000
Fast Ethernet	100
Gigabit Ethernet	10
Serial link	20000

### 13.4.4 DUAL algorithm

EIGRP uses the Diffusing Update Algorithm (DUAL) to provide the best loop-free path and loop-free backup paths. DUAL uses several terms, which are discussed in more detail throughout this section:

- **Successor** is a neighboring router that is used for packet forwarding and is the least-metric route to the destination network. The IP address of a successor is shown in a routing table entry right after the word via (see figure 13.8).
- **Feasible Distance (FD)** is the metric of the Successor to reach the destination network. FD is listed in the routing table entry as the second number inside the brackets (see figure 13.8).
- **Feasible Successor (FS)** is a neighbor that has a loop-free backup path to the same network as the successor, and it satisfies the Feasibility Condition (FC). FS is not represented in the routing table until the Successor is down. Instead, we can view FS in topology table (figure 13.9).
- **Reported Distance (RD)** is the FD (metric) of an FS to the same destination network.
- **Feasible Condition (FC)** is met when a neighbor's RD to a network is less than the local router's FD to the same one. If the RD is less, it represents a loop-free path.

```
R2# show ip route
<output omitted>
D 192.168.1.0/24 [90/3012096] via 192.168.10.10, 00:12:32,
Serial0/0/1
```

Figure 13.8: Successor and Feasible Distance

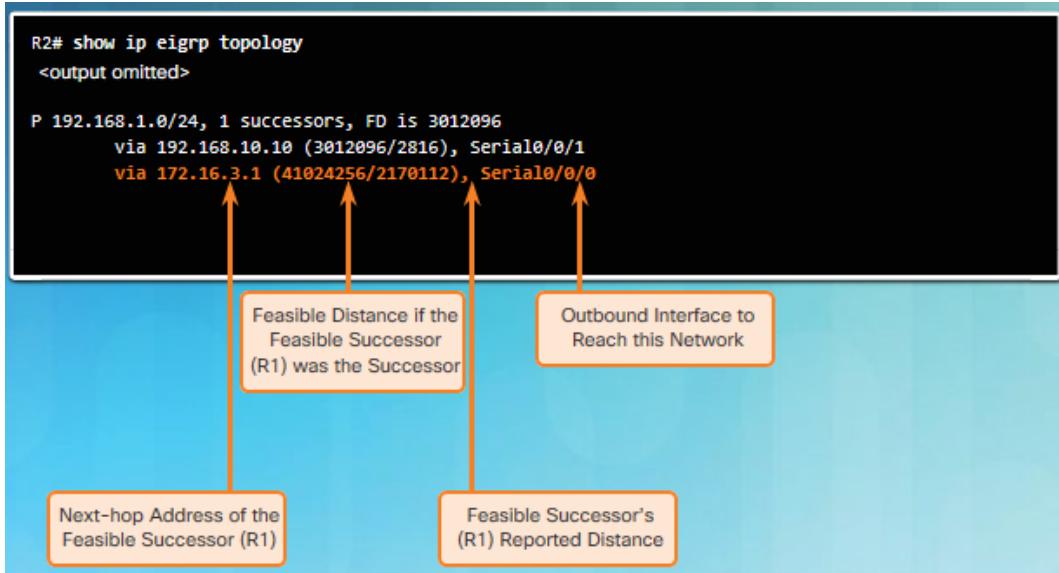
A route learned through EIGRP must meet two criteria to be installed in the local routing table:

- The route must be loop-free, being either an FS or having an RD that is less than the total distance.
- The metric of the route must be lower than the metric of the best route (the successor) multiplied by the *variance* configured on the router. For example, if the variance is set to 1, only routes with the same metric as the successor are installed in the local routing table. If the variance is set to 2, any EIGRP-learned route with a metric less than 2 times the successor metric will be installed in the local routing table.

The decision process for all route computations is done by the DUAL Finite State Machine (FSM). The DUAL FSM tracks all routes and uses EIGRP metrics to select efficient, loop-free paths, and to identify the routes with the least-cost path to be inserted into the routing table.

Recomputation of the DUAL algorithm can be processor-intensive. EIGRP avoids recomputation whenever possible by maintaining a list of backup routes that DUAL has already determined to be loop-free. If the primary route in

Figure 13.9: Feasible Successor in Topology table



the routing table fails, the best backup route is immediately added to the routing table.

When the Successor is no longer available and there is no FS, DUAL puts the route into an *active* state. DUAL sends EIGRP queries asking other routers for a path to the network. Other routers return EIGRP replies, letting the sender of the EIGRP query know whether or not they have a path to the requested network. If none of the EIGRP replies have a path to this network, the sender of the query does not have a route to this network. See also figure 13.10.

### 13.4.5 Automatic summarization

Route summarization allows a router to group networks together and advertises them as one large group using a single, summarized route. Summarization decreases the number of entries in routing updates and lowers the number of entries in local routing tables. It also reduces bandwidth utilization for routing updates and results in faster routing table lookups. However, in classful IP network, the only way that all routers can find the best routes for each individual subnet is for neighbors to send subnet information. In this situation, automatic summarization should be disabled.

A problem associated with automatic summarization is that a summary address also advertises networks that are not available on the advertising router. For example, R1 is advertising the summary address 172.16.0.0/16, but it is only connected to 172.16.1.0/24. Therefore, R1 may receive incoming packets to destinations that do not exist (for example, 172.16.2.0/24). It then forwards a request to a destination network that does not exist, creating a routing loop.

EIGRP uses the Null0 interface to avoid the above problem. The Null0 interface is a virtual IOS interface that is a route to nowhere. If R1 receives a packet destined for a network that is advertised by the classful mask but does not exist, it discards the packets by sending them to Null0.

**Note!** The Null0 summary route is removed when automatic summarization is disabled.

**Note!** EIGRP for IPv4 automatic summarization is disabled by default.

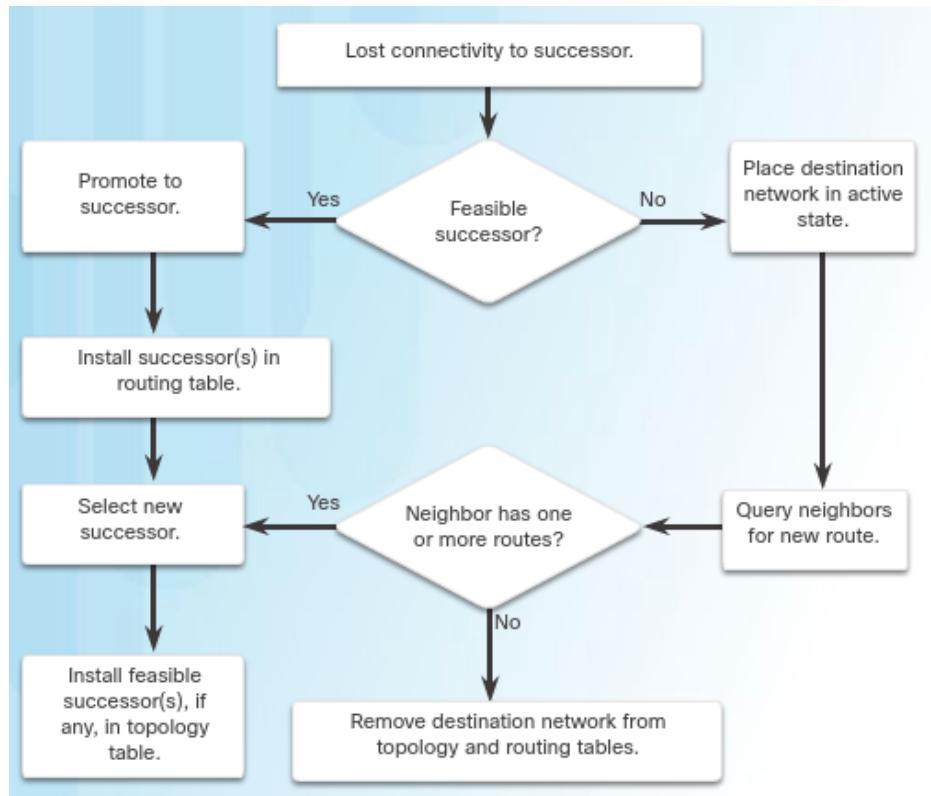


Figure 13.10: DUAL Finite State machine

## 13.5 Configuration

Listing 15: EIGRP for IPv4

```

router eigrp 10

no auto-summary

do show ip route | inc C
network 10.1.1.0 0.0.0.3
network 192.168.1.0 0.0.0.255

passive-interface default
no passive-interface s0/0/0

#redistribute static
end

show ip route eigrp
show ip protocol
show ip eigrp neighbors
show ip eigrp topology
  
```

**Note!** Only use `redistribute static` command on border router to propagate its default static route to other routers.

**Listing 16: EIGRP for IPv6**

```

ipv6 unicast-routing

ipv6 router eigrp 1
  no shutdown
  eigrp router-id 1.1.1.1
  passive-interface default
  no passive-interface s0/0/0
  #redistribute static

interface g0/0
  ipv6 eigrp 1
  no shutdown
end

show ipv6 route eigrp
show ipv6 protocols
show ipv6 eigrp neighbors
show ipv6 eigrp topology

```

**Bandwidth utilization:** By default, EIGRP uses only up to **50%** of an interface's bandwidth for EIGRP information. This prevents the EIGRP process from over-utilizing a link and not allowing enough bandwidth for the routing of normal traffic. Use the following command to configure the 80% of bandwidth that can be used by EIGRP (AS = 100) on an interface.

```

Router(config-if)# ip bandwidth-percent eigrp 100 80
Router(config-if)# ipv6 bandwidth-percent eigrp 100 80

```

**Hello and Hold timers:** Hello intervals and hold times are configurable on a per-interface basis and do not have to match with other EIGRP routers to establish or maintain adjacencies. If the Hello interval is changed, ensure that the hold time value is equal to, or greater than, the Hello interval. The following commands set Hello interval to 50 seconds and Hold timer to 150 seconds. The number **100** is the AS number of EIGRP.

**Note!** The *seconds* value for both Hello and Hold time intervals can range from 1 to 65,535.

```

Router(config-if)# ip hello-interval eigrp 100 50
Router(config-if)# ip hold-time eigrp 100 150

```

**Equal-cost load balancing:** Load balancing is the ability of a router to distribute outbound traffic using all interfaces that have the same metric from the destination address. For IP, Cisco IOS Software applies load balancing using up to *four* equal-cost paths by default. However, this can be modified using the below command. If the <value> is set to 1, load balancing is disabled. The `show ip protocols` command can be used to verify the number of equal-cost paths currently configured on the router.

```
Router(config-router)# maximum-paths <value>
```

**Unequal-cost load balancing:** EIGRP is the only routing protocol that supports unequal-cost load balancing. Setting a variance value using the following command enables EIGRP to install routes with unequal cost in a local routing table.

```
R1(config-router)# variance 2
```



# Chapter 14

# OSPF

## 14.1 Overview

### 14.1.1 Operation

1. **Establish neighbor adjacencies:** An OSPF-enabled router sends Hello packets out all OSPF-enabled interfaces to determine if neighbors are present on those links.
2. **Building the Link-State Packets:** Each router builds a link-state packet (LSP) containing the state of *each* directly connected link.
3. **Flooding the LSP:** Routers flood their LSPs to all neighbors. To do this, whenever a router receives an LSP from a neighboring router, it immediately sends that LSP out all other interfaces, except the interface that received the LSP. This process creates a flooding effect of LSPs from all routers throughout the routing area.
4. **Build the Topology Table:** Routers build the topology table (LSDB) based on the received LSPs.
5. **Execute the SPF Algorithm:** Routers use the SPF algorithm to create the SPF tree.
6. **Insert best path to routing table:** From the SPF tree, the best paths are inserted to the IP routing table. The route will remain the routing table unless there is another route with lower administrative distance (AD). Routing decisions are made based on the entries in the routing table, not LSDB.

### 14.1.2 OSPF network types

OSPF defines five network types:

- **Point-to-point** – Two routers interconnected over a common link. No other routers are on the link. (Figure 14.1(a))
- **Multiaccess** – Multiple routers interconnected over a switch. (Figure 14.1(b))
- **Nonbroadcast multiaccess (NBMA)** – Multiple routers interconnected in a network that does not allow broadcasts, such as Frame Relay. (Figure 14.1(c))
- **Point-to-multipoint** – Multiple routers interconnected in a hub-and-spoke topology over an NBMA network (Figure 14.1(d)).
- **Virtual links** – Special OSPF network used to interconnect distant OSPF areas to the backbone area (Figure 14.1(e)). In this scenario, area 51 cannot connect directly to area 0. A special OSPF area must be configured to connect area 51 to area 0. The R1 and R2 in area 1 must be configured as a virtual link.

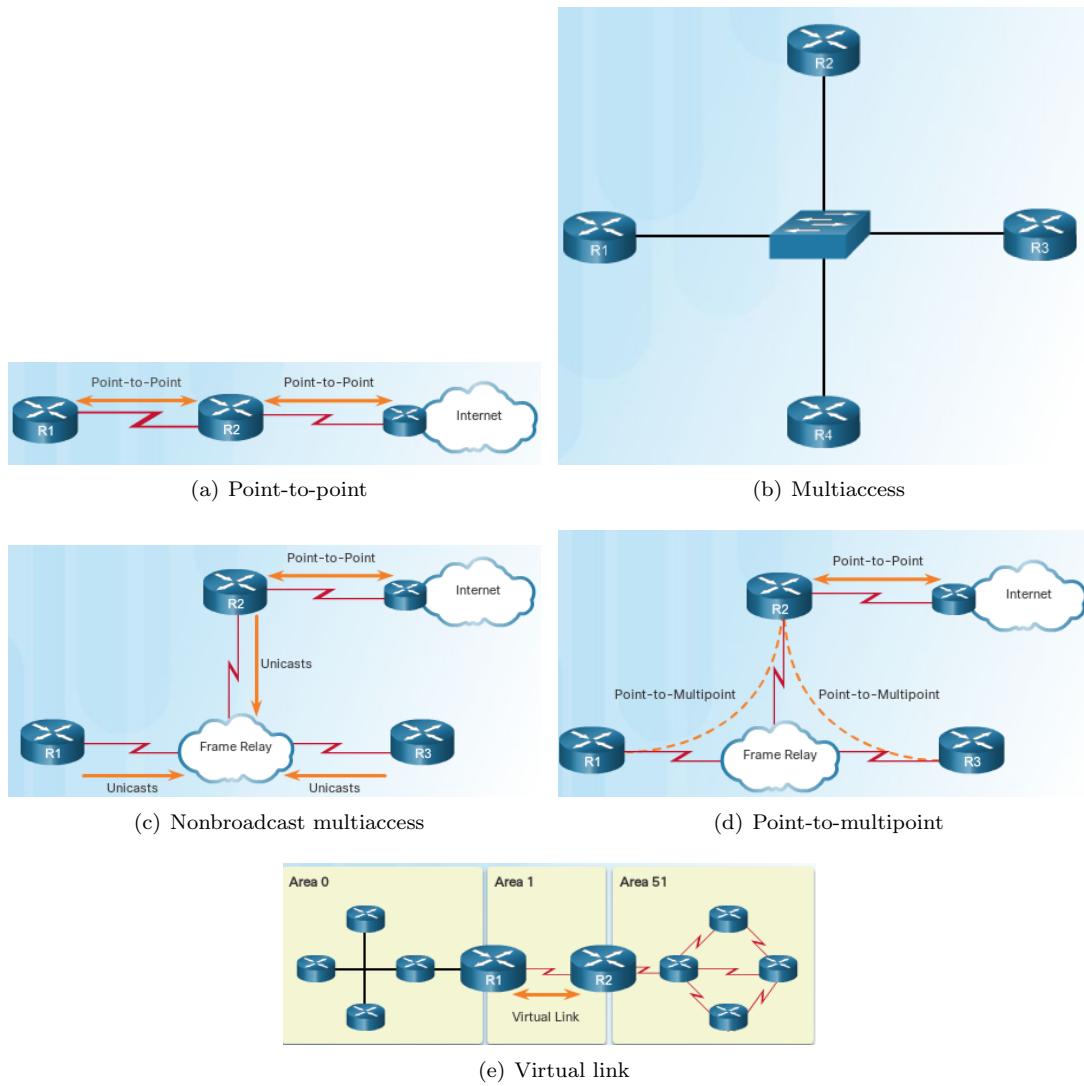


Figure 14.1: OSPF network types

### 14.1.3 OSPF cost

OSPF uses **cost** as a metric, where lower cost indicates a better path. The cost of an interface is inversely proportional to the bandwidth of the interface:

$$\text{cost} = \frac{\text{Reference bandwidth}}{\text{Interface bandwidth}}$$

The default reference bandwidth is 100 Mb/s, therefore the formula is:

$$\text{cost} = \frac{10^8}{\text{Interface bandwidth in bps}}$$

Notice that any interfaces faster than 100 Mb/s share the same cost 1, because the OSPF cost value must be an integer. To avoid this, changing reference bandwidth to a higher value than 100 Mb/s is required.

**Note!** Changing the reference bandwidth does not actually affect the physical bandwidth of the device; rather, it simply affects the calculation used to determine the metric.

## 14.2 Protocol components

The OSPF routing protocol has three main components:

- Data structures
- Routing protocol messages
- Algorithm

### 14.2.1 Data structure

Data structures are the tables or databases that OSPF builds in order to operate. OSPF creates and maintains three databases. These databases are kept and maintained in RAM.

- **Adjacency database** (neighbor table) is a list of all neighbor routers, and unique for each router. It can be viewed using `show ip ospf neighbor` command.
- **Link-state database or LSDB** (topology table) shows the network topology and is identical for all routers in one area. It can be viewed using `show ip ospf database` command.
- **Forwarding database** (Routing table) is a list of best routes to reach networks. In the routing table, OSPF intra-area routes start with O, inter-area routes start with O IA, external routes start with O E1 (or O E2).

### 14.2.2 Messages

OSPF messages are transmitted to multicast address 01-00-5E-00-00-05 and 01-00-5E-00-00-06 in MAC address, 224.0.0.5 and 224.0.0.4 in IPv4, or FF02::5 and FF02::6 in IPv6. The protocol field in IP packet header is set to **89** for OSPF protocol.

OSPF uses five types of packets to convey routing information:

- **Hello packets** establish neighbor adjacency, and facilitate the DR, BDR election in multiaccess network.
- **Database description (DBD) packets** contain an abbreviated LSDB.
- **Link-State Request (LSR) packets** request additional information about network.
- **Link-State Update (LSU) packets** are sent only to neighbors *every 30 minutes*, or as a response to LSRs, or when a change is perceived.
- **Link-State Acknowledgment (LSAck) packets** are used to confirm receipt of the LSU.

### Hello and dead intervals

The frequency at which the router sends hello packets is specified by hello interval in packet header. The default hello interval on point-to-point and multiaccess network is 10 seconds, on NBMA is 30 seconds.

Another important timer is dead interval, which is the period that the router waits to receive a hello packet before declaring the neighbor down. By default, dead interval is *four times hello interval*.

### Link-State Advertisement

The link-state advertisement (LSA) is a basic communication means of the OSPF routing protocol. It describes a building block of the LSDB. Individually, they act as database records and provide specific OSPF network details.

LSAs are not LSPs, they are actually packaged inside LSPs to convey different kinds of routing information. The use of terms LSU, LSP and LSA can sometimes confusing because these terms are often used interchangeably. However, they are different: *An LSU is a type of LSP, an LSP contains one or more LSAs*.

LSA has its own header, which includes link-state type, link's cost, sequence number, the address of advertising router, and link ID. The *link ID* field identifies the piece of the routing domain based on LSA type (see table 14.1).

Table 14.1: Link-State Advertisement types

LSA type	Sending router	Description	Flooding area	Link ID
1	all routers	Introduce directly connected networks to its neighbors	one area	router ID of the originating router
2	DR	Give other routers info about multiaccess network	one area	IP interface address of DR
3	ABR	Propagates info of each area to all other routers	between areas	network address
4	ABR	Identify ASBR and provide the route to it	entire routing domain	router ID of ASBR
5	ASBR, ABR	Advertise external network	entire routing domain	external network address

Receiving a type 3 LSA does not cause a router to run the SPF algorithm, but the routes being advertised in the type 3 LSAs are appropriately updated to the routing table.

#### 14.2.3 Algorithm

When an OSPF router is initially connected to a network, it goes through the following states in order:

1. **Down state** (send hello packets but cannot receive them)
2. **Init state** (send and receive hello packets)
3. **Two-way state** (elect a DR and a BDR)
4. **ExStart state** (decide which router will send the DBD packets first, the router with higher router ID will be the first one to send DBD packets)
5. **Exchange state** (exchange DBD packets)
6. **Loading state** (if the information in DBD packets is different from the LSDB, a router will transition to this state to gain additional route information, using LSRs)
7. **Full state** (reach convergence)

SPF algorithm calculates the best paths in order: calculate intra-area routes → calculate inter-area routes → calculate external routes.

## 14.3 DR election

### 14.3.1 Terminologies

**Multiaccess network** can create challenges for the flooding of LSPs. Ethernet network interconnects routers over a common link, therefore each router considers all counterparts in multiaccess network are its neighbors and establish adjacency to each of them (see figure 14.2). This could lead to extensive flooding of LSPs when OSPF is initialized or when topology changes.

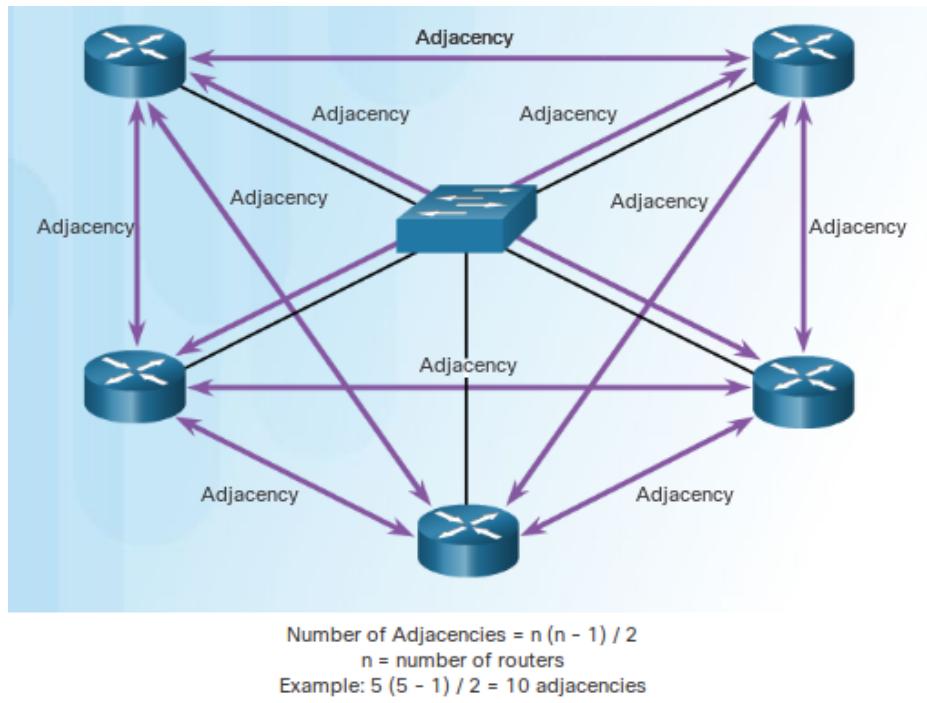


Figure 14.2: Creating adjacencies with every neighbors in multiaccess network

**Designated Router (DR)** is the solution to the problems on multiaccess network. On multiaccess networks and multiaccess networks only, OSPF elects a DR to be the collection and distribution point for LSPs sent and received. The router with the The router ID of DR can be viewed using `show ip ospf interface` command on any routers within the multiaccess network.

**Backup Designated Router (BDR)** is also elected. The BDR listens passively to this exchange and maintains a relationship with all the routers. If the DR stops producing Hello packets, the BDR promotes itself and assumes the role of DR.

**DROTHERs** are all other routers that are neither the DR nor the BDR. Each DROTHER forms full adjacencies (Full state) with the DR and BDR and form 2-way adjacencies (Two-way state) with any DROTHERs (see figure 14.3). DROTHERs only send their LSPs to the DR and BDR using the multicast address **224.0.0.6** (DR-routers address). All DROTHER routers still receive Hello packets from each other.

*Every router requires a router ID to participate in an OSPF domain.* The router ID is used by the OSPF-enabled router to Uniquely identify the router and Participate in DR election. Cisco routers derive the router ID in one of three ways and with the following precedence:

1. Router ID configured with the OSPF `router-id` command, if present
2. Highest IP address of any of the router's loopback addresses, if present
3. Highest active IP address on any of the router's physical interfaces

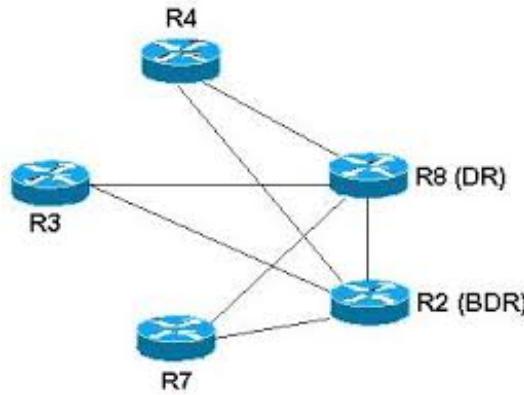


Figure 14.3: DROTHERs only form full adjacencies with the DR and BDR in the network

### 14.3.2 Election decision

The OSPF DR and BDR election decision is based on the following criteria, in sequential order:

1. The routers in the network elect the router with the highest interface priority as the DR. The router with the second highest interface priority is elected as the BDR.
2. If the interface priorities are equal, then the router with the highest router ID is elected the DR. The router with the second highest router ID is the BDR.

After the DR is elected, it remains the DR until one of the following events occurs:

- The DR fails
- The OSPF process on the DR fails or is stopped
- The multiaccess interface on the DR fails or is shutdown

The DR and BDR elections are not pre-emptive. If a new router with a higher priority or higher router ID is added to the network after the DR and BDR election, the newly added router does not take over the DR or the BDR role. This is because those roles have already been assigned. The addition of a new router does not initiate a new election process.

If the DR fails, the BDR is automatically promoted to DR. This is the case even if a new router with a higher priority or router ID is added to the network after the initial DR/BDR election. However, after a BDR is promoted to DR, a new BDR election occurs and the new router is elected as BDR.

## 14.4 OSPF area

An OSPF area is a group of routers that share the same link-state information in their LSDBs. An instance of OSPF can have several areas (called Multi-area OSPF). Implementing multiple OSPF areas has many advantages:

- Smaller routing table (There are fewer routing table entries as network addresses can be summarized between areas.)
- Reduced link-state update overhead (Fewer routers exchanging LSAs because LSA flooding stops at the area boundary.)
- Reduced frequency of SPF calculations (Routing still occurs between the areas, however, the CPU intensive routing operation of recalculating the SPF algorithm is done only for routes within an area.)

To make multi-area OSPF more efficient and scalable, a two-layer area hierarchy is implemented:

- **Backbone (Transit) area** – A backbone area directly connected with all other areas. All traffic moving from one area to another area must traverse the backbone area. Generally, end users are not found within a backbone area. The backbone area is always OSPF area 0.
- **Regular (Non-backbone) area** – Connects users and resources. By default, a regular area does not allow traffic from another area to use its links to reach other areas. All traffic from other areas must cross a transit area.

In multi-area OSPF, there are four different types of OSPF routers:

- Internal router (have all interfaces in the same area)
- Backbone router (reside in backbone area)
- Area border router or ABR (have interfaces attached to multiple areas)
- Autonomous System Boundary Router or ASBR (have at least one interface attached to an external network)

## 14.5 Configuration

### 14.5.1 Recommendations

The optimal number of routers per area varies based on factors such as network stability, but Cisco recommends the following guidelines:

- An area should have no more than 50 routers.
- A router should not be in more than three areas.
- Any single router should not have more than 60 neighbors.

Also keep the following notes in mind:

- Propagating type 3 and 5 LSAs can cause significant flooding problems. For this reason, it is strongly recommended that manual route summarization be configured on the ABRs and ASBR.
- For routers to become adjacent, their Hello interval, Dead interval, area ID (not process ID), subnet masks, and Stub area flag must match.
- The dead interval must be larger than Hello interval.

**Listing 17: OSPF for IPv4**

```
router ospf 1

network 192.168.1.0 0.0.0.255 area 0
network 192.168.12.0 0.0.0.3 area 55

passive-interface default
no passive-interface s0/0/0

#default-information originate
end

show ip ospf
show ip route ospf
show ip protocols
show ip ospf neighbor
show ip ospf interface brief
```

**Note!** Do not use `redistribute static` as this command will not recognize default static route (due to auto summarization, it only recognizes classful route).

#### Listing 18: OSPF for IPv6

```
ipv6 unicast-routing

ipv6 router ospf 1
  no shutdown
  router-id 1.1.1.1
  passive-interface default
  no passive-interface s0/0/0
exit

interface g0/0
  ipv6 ospf 1 area 0
  no shutdown
end

show ipv6 ospf
show ipv6 route ospf
show ipv6 protocols
show ipv6 ospf neighbor
show ipv6 ospf interface brief
```

### 14.5.2 Summarization

The following command enables Inter-area route summarization (for ABRs only). It summarizes all routes in an area to the specified summary address.

```
R1(config)# router ospf 1
R1(config-router)# area 1 range 192.168.64.0 255.255.224.0
```

The following command enables External route summarization (for ASBRs only). It advertises a single route for all redistributed routes that are covered by a specified network.

```
R1(config)# router ospf 1
R1(config-router)# summary-address 192.168.64.0 255.255.224.0
```

### 14.5.3 Priority

By default, all IOS OSPF routers are assigned a priority of 1. OSPF priority is configured in per-interface basis. The higher the priority value (from 0 to 255), the more likely the router becomes the DR or BDR on the interface. The priority of 0 means that the router will never become a DR or BDR. On the other hand, 255 means the router will always be DR or BDR. As for IPv6, replace `ip` by `ipv6`.

```
R1(config)# interface g0/0
R1(config)# ip ospf priority 150
```

### 14.5.4 Dead and Hello interval

Configure OSPF Dead interval, Hello interval (in seconds).

**Note!** The Dead interval must not be smaller than the Hello interval. By default, the Dead interval is four times the Hello interval. As for IPv6, replace ip by ipv6.

```
R1(config)# interface g0/0
R1(config-if)# ip ospf hello-interval 20
R1(config-if)# ip ospf dead-interval 80
```

#### 14.5.5 Reference bandwidth

Change reference bandwidth to 1000 Mb/s.

```
R1(config)# interface g0/0
R1(config-if)# auto-cost reference-bandwidth 1000
```

#### 14.5.6 Assign OSPF cost

Directly assign OSPF cost. The cost is a number between 1 and 65,535. As for IPv6, replace ip by ipv6.

```
R1(config)# interface g0/0
R1(config-if)# ip ospf cost 1564
```

#### 14.5.7 Verification

Examine the neighbor adjacencies using `show ip ospf neighbors`, routing table using `show ip route ospf`, topology using `show ip ospf interface`. Verify the parameters and current state of OSPF processes using `show ipv6 protocols`. As for IPv6, replace ip by ipv6.



# **Part IV**

# **WAN TECHNOLOGIES**



# Chapter 15

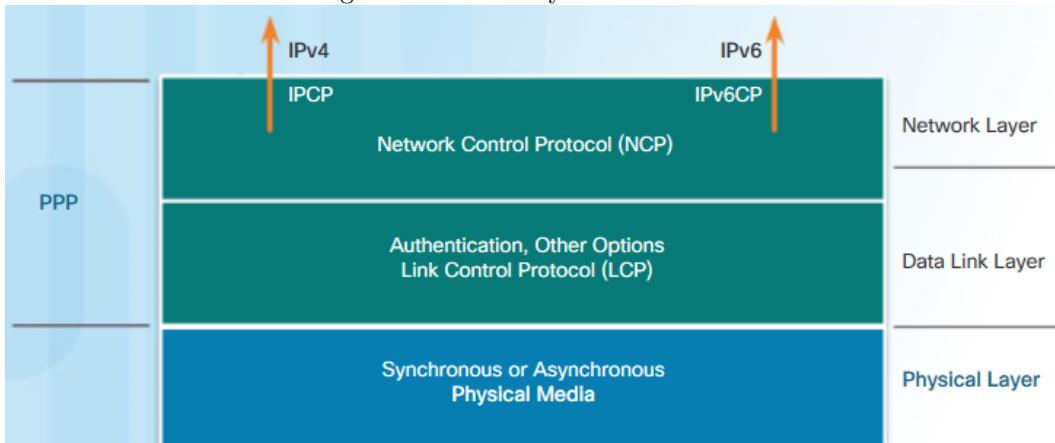
## PPP

### 15.1 Introduction

HDLC is the default serial encapsulation method when connecting two Cisco routers and it can only work with other Cisco devices. HDLC is now the basis for *synchronous* point-to-point used by many servers to connect to a WAN, most commonly the Internet. However, when there is a need to connect to a non-Cisco router, PPP encapsulation should be used.

There are many advantages to using PPP, including the fact that it is not proprietary. PPP provides router-to-router and host-to-network connections over *synchronous* and *asynchronous* circuits. PPP includes many features not available in HDLC: The link quality management feature (LQM) monitors the quality of the link, PAP and CHAP authentication.

Figure 15.1: PPP layered architecture



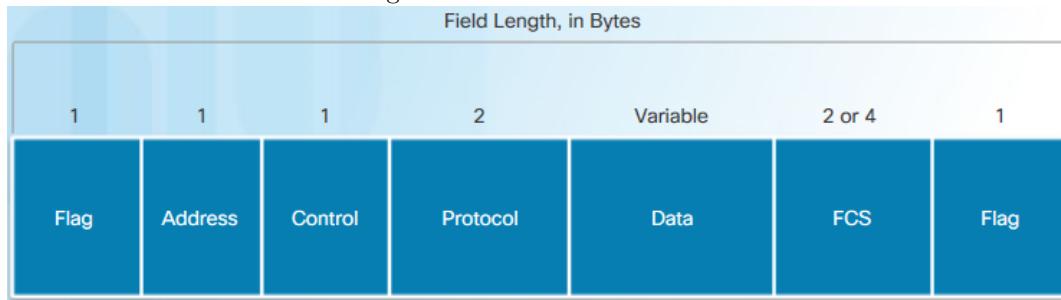
PPP contains three main components: HDLC-like framing, LCP, and NCPs. Figure 15.1 maps the layered architecture of PPP against the OSI model. PPP and OSI share the same physical layer, but PPP distributes the functions of LCP and NCP differently. Most of the work done by PPP happens at the data link and network layers, by LCP and NCPs.

### 15.2 Operation

#### 15.2.1 Frame Structure

A PPP frame consists of six fields. The following descriptions summarize the PPP frame fields illustrated in the figure 15.2:

Figure 15.2: PPP Frame fields



- **Flag:** A single byte that indicates the beginning or end of a frame. The Flag field consists of the binary sequence 01111110 (63 in decimal).
- **Address:** A single byte that contains the broadcast address because PPP does not assign individual station addresses.
- **Control:** A single byte that contains the binary sequence 00000011, which calls for transmission of user data in an unsequenced frame.
- **Protocol:** Two bytes that identify the protocol encapsulated in the information field of the frame. The 2-byte Protocol field identifies the protocol of the PPP payload.
- **Frame Check Sequence (FCS):** This is 2 bytes. If the receiver's calculation of the FCS does not match the FCS in the PPP frame, the PPP frame is silently discarded.

### 15.2.2 Establishing a PPP Session

There are three phases of establishing a PPP session:

- **Phase 1: Link establishment and configuration negotiation** – The LCP opens the connection and negotiates configuration options. This phase is complete when the receiving router sends a configuration-acknowledgment frame back to the router initiating the connection.
- **Phase 2: Link quality determination (optional)** – The LCP tests the link to determine whether the link quality is sufficient to bring up network layer protocols. The LCP can delay transmission of network layer protocol information until this phase is complete.
- **Phase 3: Network layer protocol configuration negotiation** – After the LCP has finished the link quality determination phase, the appropriate NCP can separately configure the network layer protocols. If the LCP closes the link, it informs NCPs so that they can take appropriate action.

### 15.2.3 LCP

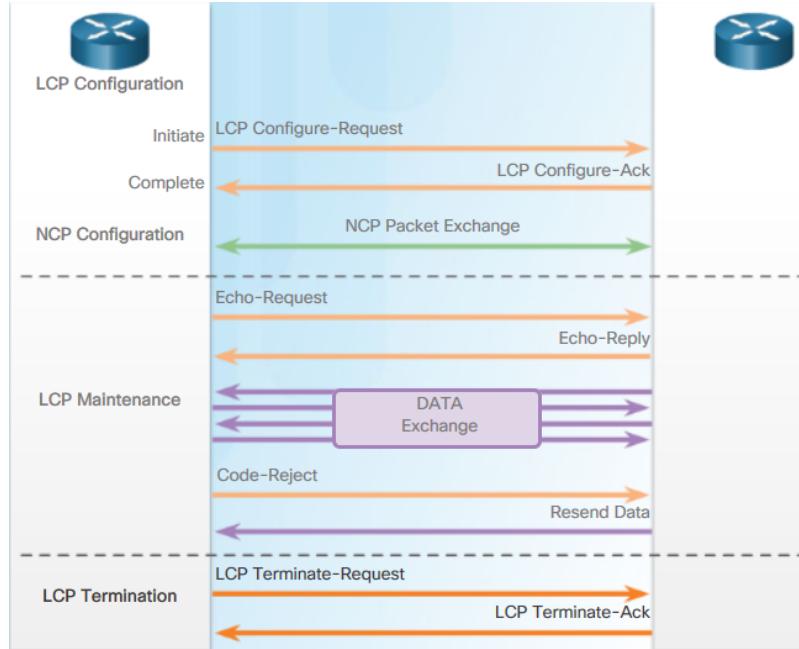
Link Control Protocol (LCP) operation uses three classes of LCP frames to accomplish the work of each of the LCP phases: Link-establishment → Link-maintenance → Link-termination (Figure 15.3).

#### Link Establishment

The link establishment process starts with the initiating device sending a Configure-Request frame to the responder. The initiator includes the options for how it wants the link created, including protocol or authentication parameters. The responder processes the request:

- If the options are not acceptable or not recognized, the responder sends a Configure-Nak or Configure-Reject message. If this occurs and the negotiation fails, the initiator must restart the process with new options.
- If the options are acceptable, the responder responds with a Configure-Ack message and the process moves on to the authentication stage. The operation of the link is handed over to the NCP.

Figure 15.3: Establish PPP session: Link-establishment, Link-maintenance, Link-termination



### Link Maintenance

When NCP has completed all necessary configurations, LCP transitions into link maintenance. During link maintenance, LCP can use messages to provide feedback and test the link using:

- Echo-Request, Echo-Reply, and Discard-Request: These frames can be used for testing the link.
- Code-Reject and Protocol-Reject: These frame types provide feedback when one device receives an invalid frame. The sending device will resend the packet.

### Link termination

After the transfer of data at the network layer completes, the LCP terminates the link, as shown in Figure 15.3. NCP only terminates the network layer and NCP link. The link remains open until the LCP terminates it. If the LCP terminates the link before NCP, the NCP session is also terminated.

The LCP closes the link by exchanging Terminate packets. The device initiating the shutdown sends a Terminate-Request message. The other device replies with a Terminate-Ack.

#### 15.2.4 NCP

After the LCP has configured and authenticated the basic link, the appropriate NCP is invoked to complete the specific configuration of the network layer protocol being used.

IPCP is an example of NCP. IPCP is responsible for configuring, enabling, and disabling the IPv4 modules on both ends of the link. IPCP negotiates two options:

- **Compression:** Allows devices to negotiate an algorithm to compress TCP and IP headers and save bandwidth.
- **IPv4-Address:** Allows the initiating device to specify an IPv4 address to use for routing IP over the PPP link, or to request an IPv4 address for the responder.

### 15.2.5 Authentication

**PAP** is a very basic **two-way handshake**. There is no encryption. The username and password are sent in plaintext. If it is accepted, the connection is allowed. CHAP is more secure than PAP. PAP may be used in the following environments: CHAP is not supported, or simulate a login at the remote host.

**PAP Process** After link establishment phase, the remote node sends a username-password pair in plain text across the link. At the receiving node, the username-password is checked. This device either allows or denies the connection. An accept or reject message is returned to the requester.

**CHAP** Unlike PAP, which only authenticates once, CHAP uses a **three-way handshake**, and conducts periodic challenges to make sure that the remote node still has a valid password value. The password value is variable and changes unpredictably while the link exists. Thus, CHAP provides protection against a playback attack.

**CHAP process** After link establishment phase, the local router sends a challenge message to the remote node. The remote node responds with a value that is calculated using a one-way hash function. The local router checks the response against its own calculation. If the values match, the initiating node acknowledges the authentication. If the values do not match, the initiating node immediately terminates the connection.

## 15.3 Configuration

To set PPP as the encapsulation method used by a serial interface, use the `(encapsulation ppp)` interface configuration command. Point-to-point software compression on serial interfaces can be configured after PPP encapsulation is enabled. If the traffic already consists of compressed files, such as .zip, .tar, or .mpeg, do not use this option. The `(ppp quality 80)` command ensures that the link meets the quality requirement set (80%); otherwise, the link closes down.

**Listing 19:** Basic PPP configuration

```
interface s0/0/0
  encapsulation ppp
  ppp quality 80
  compress [predictor | stac]
  no shutdown
```

**Multilink PPP:** provides a method for spreading traffic across multiple physical WAN links. allows packets to be fragmented and sends these fragments simultaneously over multiple point-to-point links to the same remote address.

**Listing 20:** Multilink PPP

```

interface s0/0/0
  no ip address
  encapsulation ppp
  ppp multilink
  ppp multilink group 1
  no shutdown

interface s0/0/1
  no ip address
  encapsulation ppp
  ppp multilink
  ppp multilink group 1
  no shutdown

interface Multilink 1
  ip address 10.0.1.1 255.255.255.252
  ppp multilink
  ppp multilink group 1

```

**CHAP Authentication:** The hostname (e.g. R3, R2, ISP) on one router must match the username the other router has configured in the command `username <name> password <password>`. The passwords must also match.

**Listing 21:** CHAP authentication

```

#ROUTER ISP
hostname ISP
username R3 secret cisco
interface s0/0/0
  encapsulation ppp
  ppp authentication chap
  no shutdown

#ROUTER R3
hostname R3
username ISP secret cisco
interface serial0/1/0
  encapsulation ppp
  ppp authentication chap
  no shutdown

```

**PAP Authentication:** The PAP username and password are configured in the command `ppp pap sent-username`. These username and password must match those specified with the `username <name> password <password>` command on the other router.

**Listing 22: PAP authentication**

```
#ROUTER R1
username R3 secret class
interface s0/0/0
  encapsulation ppp
  ppp authentication pap
  ppp pap sent-username R1 password cisco
  no shutdown

#ROUTER R3
username R1 secret cisco
interface s0/0/0
  encapsulation ppp
  ppp authentication pap
  ppp pap sent-username R3 password class
  no shutdown
```

# Chapter 16

## PPPoE, GRE, eBGP

### 16.1 PPPoE

Ethernet links do not natively support PPP. A solution to this problem is PPP over Ethernet (PPPoE). PPPoE creates a PPP tunnel over an Ethernet connection. This allows PPP frames to be sent across the Ethernet cable to the ISP from the customer's router.

To create a PPP tunnel, the configuration uses a *dialer interface*. A dialer interface is a virtual interface. The PPP configuration is placed on the dialer interface, not the physical interface. The PPP CHAP is then configured with hostname Cust1 and password cisco123. Then, dialer interface is linked to the Ethernet interface with the `dialer pool` command. Remember to set MTU to 1492 to accommodate PPPoE headers. The physical Ethernet interface g0/1 with PPPoE is enabled with the command `pppoe enable` interface configuration command. Then it is linked to the Dialer interface with the `pppoe-client dial-pool-number <number>` interface configuration command.

**Listing 23: PPPoE**

```
interface dialer 1
  encapsulation ppp
  ip address negotiated
interface dialer 1
  ppp authentication chap callin
  ppp chap host name Cust1
  ppp chap password cisco123
interface dialer 1
  dialer pool 1
  mtu 1492
  no shutdown
interface g0/1
  no ip address
  pppoe enable
  pppoe-client dial-pool-number 1
end

show ip int brief
show int dialer 1
show ip route
show pppoe session
debug ppp {negotiation | authentication | events}
```

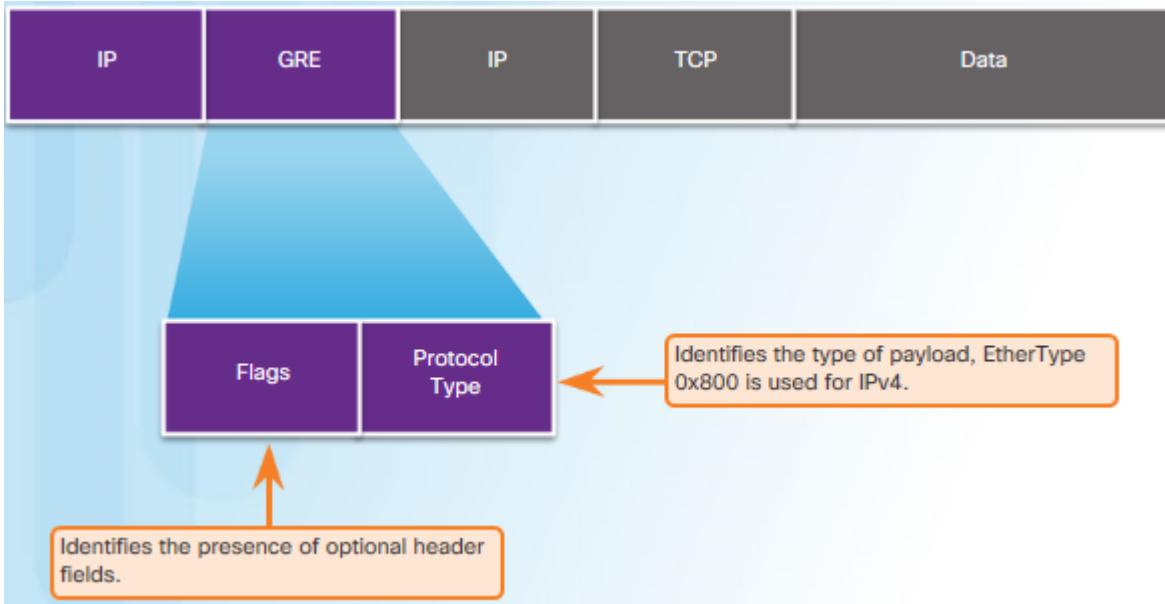
## 16.2 GRE

### 16.2.1 Introduction

GRE, IPsec, web-based SSL are the three methods of establishing a VPN connection offered by Cisco devices. GRE is a out-dated, non-secure, stateless, site-to-site VPN tunneling protocol. GRE supports the encapsulation of **any OSI Layer 3 protocol** and **47** is used in the protocol field in IP header (figure 16.1). GRE also supports multiprotocol and IP multicast tunneling.

GRE is the default tunnel interface mode for Cisco IOS software. GRE does not provide encryption or any other security mechanisms. Therefore, data that is sent across a GRE tunnel is not secure.

Figure 16.1: Header for GRE encapsulated packet header



Three circumstances can cause a GRE tunnel to be in an up/down state:

- The tunnel interface is down.
- A valid route to the destination address is missing from the routing table.
- The tunnel address is routed through the tunnel itself.

### 16.2.2 Configuration

Five steps to configuring a GRE tunnel (figure 16.2):

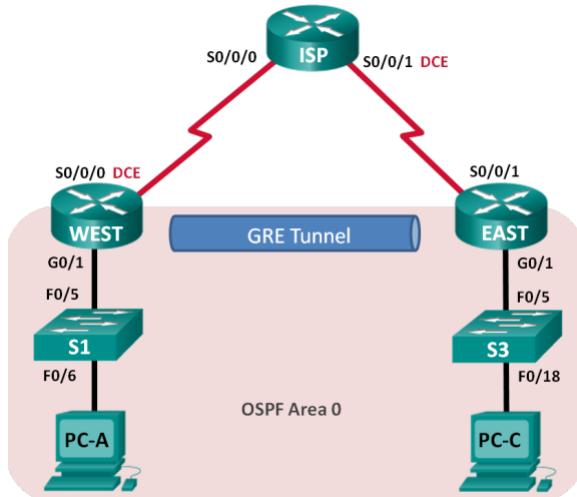
1. **Create a tunnel interface:**

Configure the tunnel interface on the WEST router. Use s0/0/0 as the tunnel source interface and 10.2.2.1 (IP address of EAST router s0/0/1) as the tunnel destination.

```
WEST(config)# interface tunnel 0
WEST(config-if)# ip address 172.16.12.1 255.255.255.252
WEST(config-if)# tunnel source s0/0/0
WEST(config-if)# tunnel destination 10.2.2.1
```

Configure the tunnel interface on the EAST router. Use s0/0/1 as the tunnel source interface and 10.1.1.1 (IP address of WEST router s0/0/0) as the tunnel destination.

Figure 16.2: Configure GRE VPN tunnel



```
EAST(config)# interface tunnel 0
EAST(config-if)# ip address 172.16.12.2 255.255.255.252
EAST(config-if)# tunnel source 10.2.2.1
EAST(config-if)# tunnel destination 10.1.1.1
```

2. **Verify that the GRE tunnel is functional:** Verify the status of the tunnel interface on the WEST and EAST routers using `show interface tunnel 0` and `show ip interface brief` commands.
3. **Enable routing over the GRE Tunnel:** After the GRE tunnel is set up, the routing protocol can be implemented. For GRE tunneling, a network statement will include the IP network of the tunnel, instead of the network associated with the serial interface. Remember that the ISP router is not participating in this routing process.

```
WEST(config)# router ospf 1
WEST(config-router)# network 172.16.12.0 0.0.0.3 area 0

EAST(config)# router ospf 1
EAST(config-router)# network 172.16.12.0 0.0.0.3 area 0
```

If BGP is used instead of OSPF or EIGRP, the neighbor statement will include the IP network of the tunnel, instead of the network associated with the serial interface.

```
WEST(config)# router bgp 65000
WEST(config-router)# neighbor 172.16.12.2 remote-as 65001

EAST(config)# router bgp 65001
EAST(config-router)# neighbor 172.16.12.1 remote-as 65000
```

## 16.3 eBGP

### 16.3.1 Introduction

Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP). BGP updates are encapsulated over TCP on port **179**. We use BGP when an autonomous system (AS) has connections to *multiple* ASs (known as multi-homed). BGP should not be used when there is a *single* connection to the Internet or another AS (known as single-homed).

There are three common ways an organization can choose to implement BGP in a multi-homed environment: Default Route Only, Default Route and ISP Routes, All Internet Routes (this would include routes to over 550,000 networks).

External BGP is the routing protocol used between routers in different autonomous systems. Internal BGP is the routing protocol used between routers in the same AS. Two routers exchanging BGP routing information are known as BGP peers.

Internal routing protocols (OSPF, EIGRP, RIP, etc.) use a specific metric (e.g. OSPF's cost) for determining the best paths to destination networks. BGP does *not* use a *single* metric like IGP's. Instead it uses several *attributes* including a list of AS numbers necessary to reach a destination network. Therefore BGP is known as a *path vector* routing protocol. Also, because of this, a misconfiguration of a BGP router could have negative effects throughout the entire Internet.

### 16.3.2 Configuration

To implement eBGP for this course, you will need to complete the following tasks:

1. Enable BGP routing and identify the AS number
2. Configure BGP neighbor(s) (peering).
3. Advertise network(s) originating from this AS.

```
R2(config)# router bgp 65000
R2(config-router)# neighbor 209.165.200.1 remote-as 65001
R2(config-router)# network 198.133.219.0 mask 255.255.255.248
R2(config-router)# end
R2# show ip route
R2# show ip bgp
R2# show ip bgp summary
```

# Chapter 17

## Quality of Service

### 17.1 Introduction

#### 17.1.1 Traffic characteristics

**Voice:** Voice traffic is predictable and smooth. However, voice is delay-sensitive and there is no reason to retransmit voice if packets are lost. Therefore, voice packets must receive a higher priority than other types of traffic. Latency should be no more than **150 ms**. Jitter should be no more than **30 ms**, and voice packet loss should be no more than **1%**. Voice traffic requires at least **30 Kb/s** of bandwidth.

**Video:** Video traffic tends to be unpredictable, inconsistent, and bursty compared to voice traffic. Compared to voice, video is less resilient to loss and has a higher volume of data per packet. Latency should be no more than **400 ms**. Jitter should be no more than **50 ms**, and video packet loss should be no more than **1%**. Video traffic requires at least **384 Kb/s** of bandwidth.

**Data:** Data traffic is relatively insensitive to drops and delays compared to voice and video. The two main factors a network administrator needs to ask about the flow of data traffic are the following: Does the data come from an interactive application? Is the data mission critical?

Network congestion causes **delay**. Two types of delays are fixed and variable. A *fixed delay* is a specific amount of time a specific process takes, such as how long it takes to place a bit on the transmission media. A *variable delay* take an unspecified amount of time and is affected by factors such as how much traffic is being processed. *Jitter* is the variation in the delay of received packets.

#### 17.1.2 QoS tools

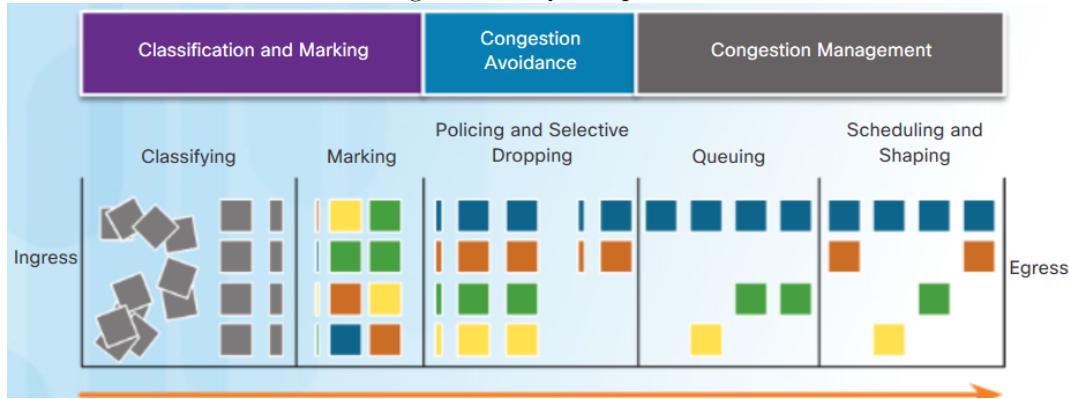
When the volume of traffic is greater than what can be transported across the network, network devices (router, switch, etc.) hold the packets in memory until resources become available to transmit them. If the number of packets continues to increase, the memory within the device fills up and packets are dropped. This problem can be solved by either increasing link capacity or implementing QoS.

A device implements QoS only when it is experiencing congestion. There are three categories of QoS tools: Classification and marking, Congestion avoidance, Congestion management. Refer to Figure 17.1 to help understand the sequence of how these tools are used when QoS is applied to packet flows.

### 17.2 Classification and marking

Before a packet can have a QoS policy applied to it, the packet has to be classified. Classification and marking identifies types of packets. Traffic should be classified and marked as close to its source as technically and administratively feasible. This defines the trust boundary.

Figure 17.1: QoS sequence



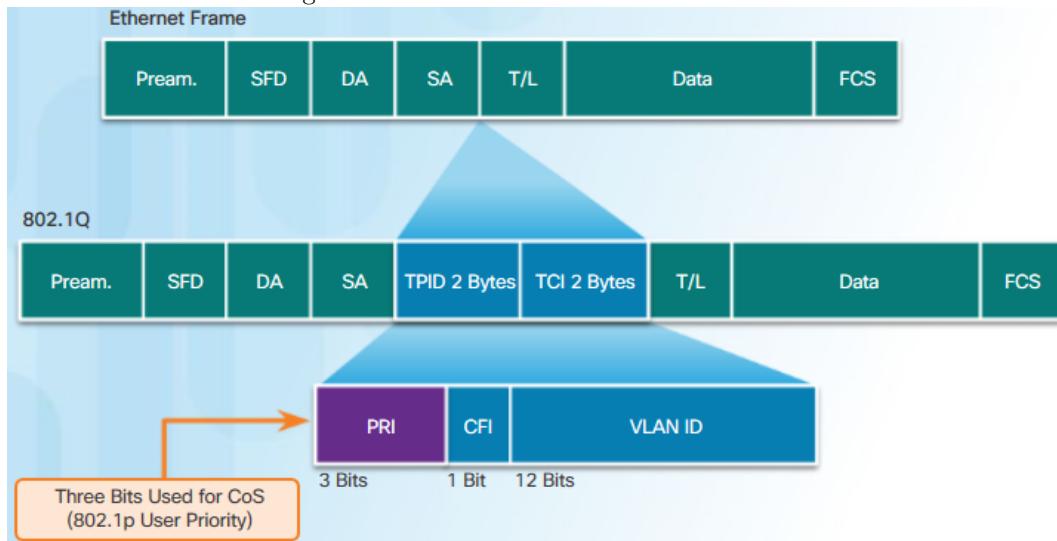
**Marking** means that we are adding a value to the packet header. Devices receiving the packet look at this field to see if it matches a defined policy. Trusted endpoints have the capabilities and intelligence to mark application traffic to the appropriate **Layer 2 CoS and/or Layer 3 DSCP** values. Examples of trusted endpoints include IP phones, wireless access points, videoconferencing gateways and systems, IP conferencing stations, and more.

Methods of **classifying** traffic flows at Layer 2 and 3 using interfaces, ACLs, and class maps.

### 17.2.1 Marking at Layer 2 (CoS field)

802.1Q is the IEEE standard that supports VLAN tagging at layer 2 on Ethernet networks. The 802.1Q standard also includes the QoS prioritization scheme known as **802.1p**. The 802.1p standard uses the first three bits in the TCI (Tag Control Information) field, to identify the CoS (Class of Service) markings (Figure 17.2).

Figure 17.2: Ethernet Class of Service values



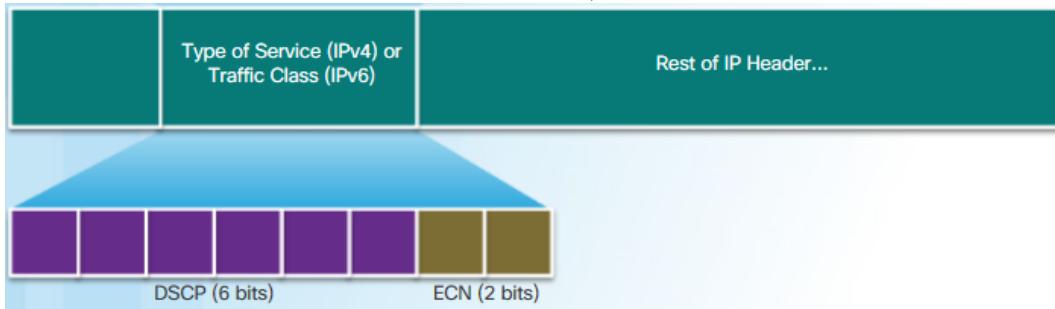
**Note!** As frame header is changed hop by hop, Layer 2 marking and the QoS information also change.

### 17.2.2 Marking at Layer 3 (DSCP field)

Unlike Layer 2 marking, Layer 3 marking does not change QoS information hop by hop, but instead, maintain the information from end to end.

Both IPv4 and IPv6 support Layer 3 marking: the **ToS** field of IPv4 packet and the **Traffic Class** field of IPv6 packet. The content of these two fields are identical, as shown in Figure 17.3. The most important portion in the field is the **DSCP** (DiffServ Code Point) field, which is designated for QoS. The DSCP values are organized into three categories:

Figure 17.3: Type of Service/Traffic Class Field



- **Best-Effort (BE):** When a router experiences congestion, these packets will be dropped. No QoS plan is implemented.
- **Expedited Forwarding (EF):** DSCP decimal value is 46 (binary 101110). At Layer 3, Cisco recommends that EF only be used to mark **voice** packets.
- **Assured Forwarding (AF):** Use the 5 bits to indicate queues and drop preference. As shown in Figure 17.4, the first 3 bits are used to designate the class. The remaining two bits are used to designate the drop preference. The 6th is set to zero. The AFxy formula shows how the AF values are calculated. For example, AF32 belongs to class 3 (binary 011) and has a medium drop preference (binary 10).

The ECN (Extended Congestion Notification) field can be used by routers to mark packets instead of dropping them. The ECN marking informs downstream routers that there is congestion in the packet flow.

Figure 17.4: Assured forwarding values

	Low Drop	Medium Drop	High Drop
<b>Class 4</b>	AF41 (34)	AF42 (36)	AF43 (38)
<b>Class 3</b>	AF31 (26)	AF32 (28)	AF33 (30)
<b>Class 2</b>	AF21 (18)	AF22 (20)	AF23 (22)
<b>Class 1</b>	AF11 (10)	AF12 (12)	AF13 (14)

AFxy      X    X    X    Y    Y    0    DSCP Field

Class              Drop Preference

Example - AF32    0    1    1    1    0    0    DSCP Value = 28

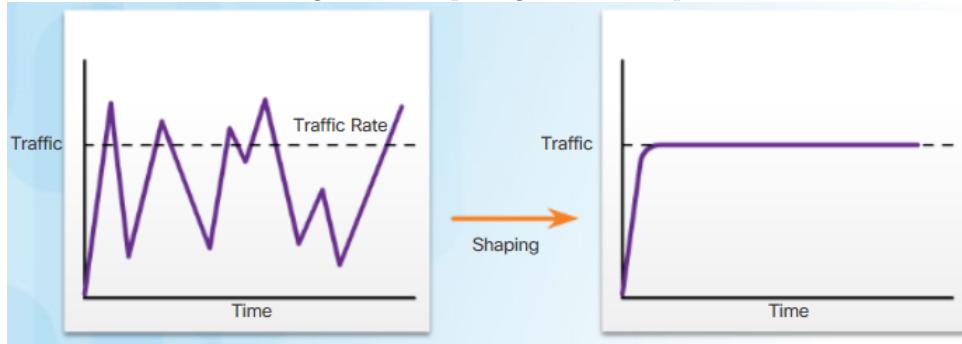
## 17.3 Congestion Avoidance

We avoid congestion by dropping lower-priority packets before congestion occurs. When the queue fills up to the maximum threshold, a small percentage of packets are dropped. When the maximum threshold is passed, all packets are dropped. WRED, traffic shaping, and traffic policing are three mechanisms provided by Cisco IOS QoS software to prevent congestion.

**WRED** is the primary congestion avoidance tool. It regulates TCP data traffic before tail drops (caused by queue overflows) occur.

**Traffic shaping** is applied to *outbound* traffic, meaning that excess packets going out an interface get queued and scheduled for later transmission. The result of traffic shaping is a *smoothed* packet output rate, as shown in Figure 17.5. Ensure that you have sufficient memory when enabling shaping.

Figure 17.5: Spacing traffic example



**Traffic policing** is applied to inbound traffic on an interface. When the traffic rate reaches the configured maximum rate, excess traffic is *dropped* (or *remarked*), as shown in figure 17.6.

Figure 17.6: Spacing traffic example

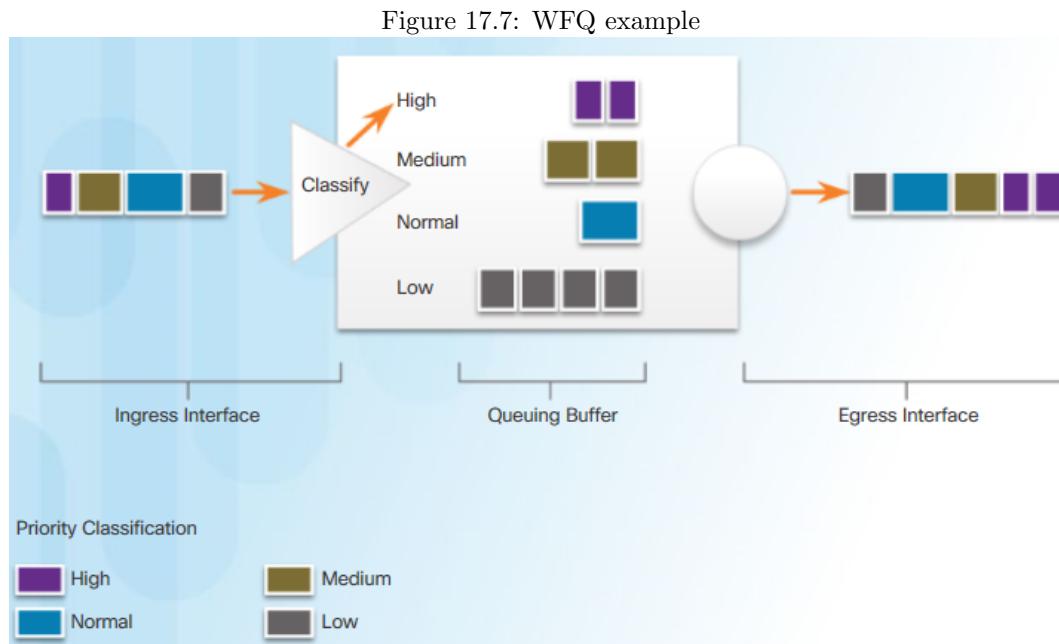


## 17.4 Congestion management

When traffic exceeds available network resources, Congestion management buffers and prioritizes packets before being transmitted to the destination. Common Cisco IOS-based congestion management tools include CBWFQ and LLQ algorithms.

### 17.4.1 WFQ

WFQ (Weighted Fair Queuing) is an automated scheduling method that provides fair bandwidth allocation to all network traffic. WFQ applies priority to identified traffic and classifies it into flows, as shown in the figure 17.7. WFQ then determines how much bandwidth each flow is allowed. WFQ classifies traffic into different flows based on packet header addressing.



WFQ is not supported with tunneling and encryption. It does not allow users to take control over bandwidth allocation.

### 17.4.2 CBWFQ

Class-Based Weighted Fair Queuing (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces.

To characterize a class, you assign it bandwidth, weight, and queue limit. After a queue has reached its configured queue limit, adding more packets to the class causes tail drop. Tail drop means a router simply discards any packet that arrives at the end of a queue.

### 17.4.3 LLQ

The Low Latency Queuing (LLQ) feature brings strict priority queuing to CBWFQ. *Strict PQ allows voice to be sent first.* Without LLQ, CBWFQ services fairly based on weight; no class of packets may be granted strict priority. This scheme poses problems for voice traffic that is largely intolerant of delay.

## 17.5 QoS models

The three models for implementing QoS are: Best-effort model, Integrated services (IntServ), Differentiated services (DiffServ). Best-effort model means *no QoS* is implemented. QoS is really implemented in a network using either IntServ or DiffServ.

### 17.5.1 Best effort

The best-effort model (meaning no QoS) treats all network packets in the same way. This model is used when QoS is not required. The table 17.1 lists the benefits and drawbacks of the best effort model.

Table 17.1: Pros and Cons of Best-effort

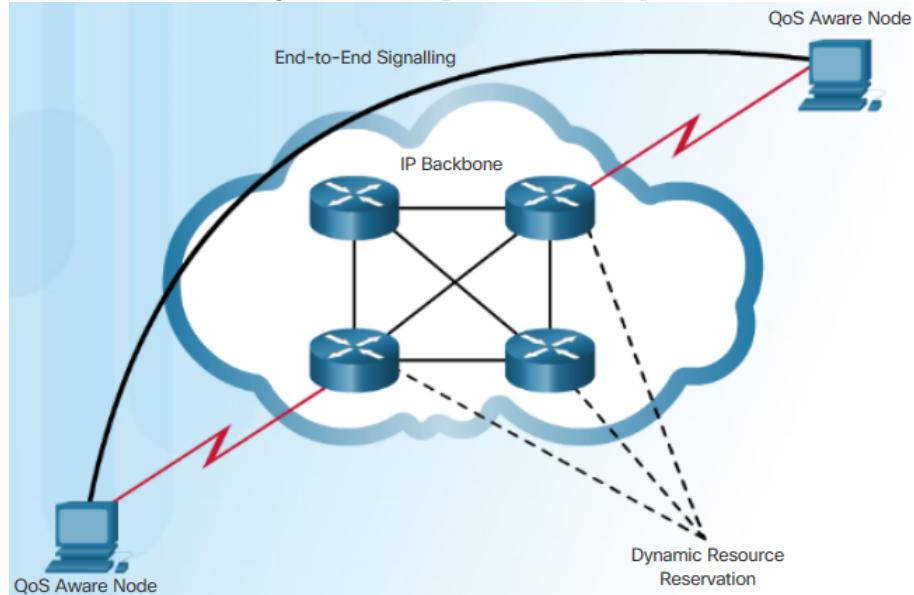
Benefits	Drawbacks
Most scalable	No guarantees of delivery
Scalability is limited by bandwidth	Packets can arrive in any order
No special QoS mechanism required	No packets have preferential treatment
Easy to deploy	Critical data is treated the same as casual one

### 17.5.2 Integrated services

Integrated Services (IntServ) is a multiple-service model that can accommodate multiple QoS requirements.

It uses *resource reservation* and *admission-control mechanisms* as building blocks to establish and maintain QoS. Each individual communication must explicitly specify its traffic descriptor and requested resources to the network (Figure 17.8). The edge router performs admission control to ensure that available resources are sufficient in the network.

Figure 17.8: Simple IntServ example



IntServ uses the **RSVP** (Resource Reservation Protocol) to reserve bandwidth for an application's traffic (e.g. VoIP) across the entire path. If this requested reservation fails along the path, the originating application does not send any data.

### 17.5.3 Differentiated services

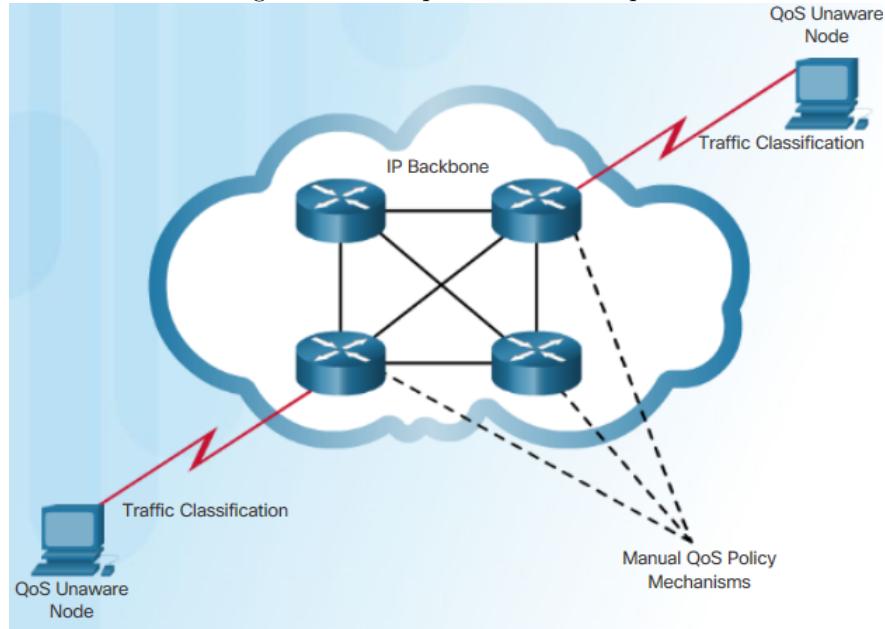
The DiffServ design overcomes the limitations of both the best-effort and IntServ models. Unlike IntServ, DiffServ is not an end-to-end QoS strategy and does not use signaling. Instead, DiffServ uses a “soft QoS” approach (Figure

Table 17.2: Pros and Cons of IntServ

Benefits	Drawbacks
Explicit end-to-end resource admission control	Resource intensive
Per-request policy admission control	Not scalable
Signaling of dynamic port numbers	

17.9). For example, DiffServ can provide low-latency guaranteed service to voice or video while providing best-effort traffic to web traffic or file transfers.

Figure 17.9: Simple DiffServ example



Specifically, DiffServ divides network traffic into classes based on business requirements. Each of the classes can then be assigned a different level of service. You pay for a level of service. Throughout the network, the level of service you paid for is recognized and your package is given either preferential or normal traffic, depending on what you requested.

Table 17.3: Pros and Cons of DiffServ

Benefits	Drawbacks
Highly scalable	No absolute guarantee of delivery
Many different levels of quality	Requires complex mechanisms



**Part V**

**INFRASTRUCTURE SERVICE**



# Chapter 18

## DHCPv4

### 18.1 Operation

DHCPv4 works in a client/server mode. When a client communicates with a DHCPv4 server, the server assigns or leases an IPv4 address to that client. The client connects to the network with that leased IP address until the lease expires.

The client must contact the DHCP server periodically to extend the lease. This lease mechanism ensures that clients that move or power off do not keep addresses that they no longer need. When a lease expires, the DHCP server returns the address to the pool where it can be reallocated as necessary.

#### 18.1.1 Lease origination

When the client wants to join a network, it begins DORA<sup>1</sup> process to obtain a lease.

1. A client starts the process with a broadcast **DHCP-DISCOVER** message to finds available DHCPv4 servers.
2. When the DHCPv4 server receives a DHCP-DISCOVER message, it reserves an available IPv4 address to lease to the client. It then sends a **DHCP-OFFER** message to the requesting client. At the same time, the server caches the MAC address and the leased IPv4 address of the requesting client to create an ARP entry.
3. When the client receives the DHCP-OFFER from the server, it broadcasts **DHCP-REQUEST** messages. This message serves as an acceptance notice to the selected server and an implicit decline to any other servers.
4. On receiving the DHCP-REQUEST message, the server verifies the lease information with an ICMP ping to that address to ensure it is not being used already. If there is *no* ICMP echo reply, then the address is not being used by any client. Otherwise, that address is being used and the server has to send DHCP-OFFER again.
5. DHCP server sends unicast **DHCP-ACK** message to the client. This message informs the client that the IP address is valid. Because The DHCP-ACK message is a duplicate of the DHCP-OFFER, it also provides IP information for the client.
6. When the client receives the DHCPACK message, it performs an ARP lookup for the assigned address. If there is *no* reply to the ARP request, the client knows that the IPv4 address is valid and starts using it as its own.

Sometimes, two clients use the same IPv4 address. This creates a **conflict**. In such case, the server uses the ping command to detect the clients that are experiencing address conflict. After that, the client uses ARP to detect conflicts. When an address conflict is detected, the address is *removed* from the pool and not assigned until an administrator resolves the conflict.

---

<sup>1</sup>DORA = Discovery, Offer, Request, Acknowledgement

### 18.1.2 Lease renewal

1. Before the lease expires, the client sends a **DHCP-REQUEST** message directly to DHCPv4 server.
2. On receiving the DHCP-REQUEST message, the server verifies the lease information by returning a **DHCP-ACK**.
3. If a DHCP-ACK is not received within a specified amount of time, the client broadcasts another DHCP-REQUEST message so that one of the other DHCPv4 servers can extend the lease.

### 18.1.3 Relay agent

Sometimes, network clients are not on the same subnet as DHCP servers. Because routers do not forward broadcasts, the DHCP-REQUEST from clients are not sent to DHCP server.

Cisco offers a solution called Cisco IOS helper address. This solution enables a router to forward DHCP-REQUEST broadcasts to the DHCPv4 server. When a router forwards address assignment/parameter requests, it is acting as a DHCPv4 *relay agent*.

## 18.2 Message

The DHCPv4 message format is used for all DHCPv4 transactions. DHCPv4 messages use **UDP port 67** for server, and **UDP port 68**. In other words, DHCPv4 messages sent from the client to server use UDP source port 68 and destination port 67, DHCPv4 messages sent from server to client use UDP source port 67 and destination port 68.

The figure shows the format of a DHCPv4 message. The fields are as follows:

- **OP code** specifies general type of message: request (1), reply (2).
- **Hardware type**: Ethernet (1), Frame Relay (15), Serial (20)
- **Hop** controls the forwarding of messages. Set to 0 by a client before transmitting a request.
- **Transaction identifier** used by client to match the request with replies from DHCPv4 server.
- **Seconds**: amount of time (in seconds) elapsed since a client attempted to acquire or renew a lease.
- **Flag** is used by a client that does not know its IPv4 address when it sends a request. Only one of the 16 bits—the broadcast flag—is used. A value of 1 in this field tells the DHCPv4 server or relay agent receiving the request that the reply should be sent as a broadcast.

**DHCP-DISCOVER:** When the client boots, it has no way of knowing the subnet to which it belongs. Therefore, destination IPv4 address of DCHP-DISCOVER is 255.255.255.255, the destination MAC address is FF:FF:FF:FF:FF:FF. The source MAC address is the MAC address of the client. The client does not have a configured IPv4 address yet, so the source IPv4 address is 0.0.0.0

**DHCP-OFFER** contains initial configuration information for the client: IPv4 address for client, subnet mask, lease duration, and IPv4 address of the DHCPv4 server. The DHCPOFFER message can be configured to include other information, such as the lease renewal time and DNS address.

## 18.3 Configuration

### 18.3.1 DHCPv4 server

**Step 1. Exclude addresses:** Some IPv4 addresses in a pool are assigned to network devices that require static address assignments. Therefore, these IPv4 addresses should not be assigned to other devices.

Figure 18.1: DHCPv4 message format

8	16	24	32		
OP Code (1)	Hardware Type (1)	Hardware Address Length (1)	Hops (1)		
Transaction Identifier					
Seconds - 2 bytes		Flags - 2 bytes			
Client IP Address (CIADDR) - 4 bytes					
Your IP Address (YIADDR) - 4 bytes					
Server IP Address (SIADDR) - 4 bytes					
Gateway IP Address (GIADDR) - 4 bytes					
Client Hardware Address (CHADDR) - 16 bytes					
Server Name (SNAME) - 64 bytes					
Boot Filename - 128 bytes					
DHCP Options - variable					

```
R1(config)# ip dhcp excluded-address <ip-address>
R1(config)# ip dhcp excluded-address 192.168.10.1
```

```
R1(config)# ip dhcp excluded-address <range-of-address>
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
```

**Step 2. Configure address pool:** Define a pool of addresses to assign to clients.

```
R1(config)# ip dhcp pool <pool-name>
R1(dhcp-config)# network <network-range>

R1(config)# ip dhcp pool LAN-POOL-1
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
```

**Step 3.** Define the default gateway router for clients.

```
R1(dhcp-config)# default-router 192.168.10.1
```

**Step 4. Relay agent:** If network clients are not on the same subnet as DHCP servers, configure the default gateway as a relay agent.

```
R1(config)# interface g0/0
R1(config-if)# ip helper-address 192.168.11.6
```

**Step 4 (Optional)** Other DHCP specifics

```
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
```

**Step 5.** Verification: The `show ip dhcp binding` command displays a list of all IPv4 address to MAC address bindings that have been provided by the DHCPv4 service. The `show ip dhcp server statistics` command verifies that messages are being received or sent by the router. The `show ip dhcp conflict` command displays all address conflicts recorded by the DHCPv4 server.

```
R1# show run | sec dhcp
R1# show ip dhcp binding
R1# show ip dhcp statistics
R1# show ip dhcp conflict
```

**Note!** If there is a switch between the client and the DHCPv4 server, and the client is unable to obtain the DHCP configuration, switch port configuration issues may be the cause. These causes may include issues from trunking and channeling, STP, and RSTP. *PortFast* and *Edge port* configurations resolve the most common DHCPv4 client issues that occur with an initial installation of a Cisco switch.

### 18.3.2 DHCPv4 client

Routers in small office/home office (SOHO) and branch sites have to be configured (by ISP) as DHCPv4 clients in a similar manner to client computers. In the simplest configuration, the Ethernet interface (not Serial interface) of the router is used to connect to a cable or DSL modem. To configure an Ethernet interface of a router as a DHCP client, use the following command:

```
SOHO(config)# interface g0/1
SOHO(config-if)# ip address dhcp
SOHO(config-if)# no shutdown
```

# Chapter 19

## DHCPv6

### 19.1 General Operation

When the client boots up, it sends DHCPv6 SOLICIT message to **FF02::1:2**, which is the multicast all-DHCPv6-server address. One or more servers respond with a DHCPv6 ADVERTISE unicast message to inform the client that the server is available for DHCPv6 service. The client responds with either a DHCPv6 REQUEST or a DHCPv6 INFORMATION-REQUEST unicast message:

- DHCPv6 INFORMATION-REQUEST unicast message is used for Stateless DHCPv6 (i.e. SLAAC, Stateless DHCPv6 and SLAAC) to request only additional information, such as DNS server addresses, domain name.
- DHCPv6 REQUEST unicast message is used for Stateful DHCPv6 to ask for all IP configuration parameters from the server.

The server sends a DHCPv6 REPLY unicast message to the client containing the information requested in the DHCPv6 REQUEST or DHCPv6 INFORMATION REQUEST message.

### 19.2 SLAAC

Stateless Address AutoConfiguration (SLAAC) is enabled by default. Both the M flag and the O flag are set to 0 in the RA. In SLAAC, a host automatically obtains its IP configuration from an IPv6-enabled router. The host generates its own global unicast IPv6 address. A DHCPv6 server is not required. SLAAC operation includes the following steps:

1. Client asks for IPv6 configuration by sending an RS to the router R1.
2. R1 receives the RS message and responds with an RA message.
3. PC1 receives the RA, and uses the information in the message to create its own IPv6 global unicast address.
4. Because SLAAC is a stateless process, PC1 must verify that this newly created IPv6 address is unique using DAD process.

Figure 19.1: Verify SLAAC method

```
R1# show ipv6 interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::D68C:B5FF:FECE:A0C1
  <output omitted>
    Hosts use stateless autoconfig for addresses.
R1#
```

To configure SLAAC, use the following commands:

```
R1(config)# ipv6 unicast-routing
R1(config)#
R1(config)# interface g0/1
R1(config-if)# ipv6 nd managed-config-flag
R1(config-if)# ipv6 nd other-config-flag
R1(config-if)# end
R1# show ipv6 dhcp pool
R1# show ipv6 interface
R1# debug ipv6 dhcp detail
```

### 19.3 SLAAC and Stateless DHCPv6

A host obtains IP configuration (prefix, prefix-length, default gateway) using SLAAC and additional information (DNS server, domain name, etc.) from a stateless DHCPv6 server. The host generates its own global unicast IPv6 address.

There are four steps to configure a router as a DHCPv6 server:

1. Enable IPv6 Routing using `ipv6 unicast-routing` command
2. Use `ipv6 dhcp pool <pool-name>` command to create a pool and enter the router in DHCPv6 configuration mode
3. (Optional) In DHCPv6 configuration mode, configure DNS server address (`dns-server <ip-address>`) and domain name (`domain-name <name>`).
4. In the interface configuration mode, we bind the DHCPv6 pool to the interface using `ipv6 dhcp server <pool-name>`
5. For stateless DHCPv6, the O flag is 1 and the M flag is 0, therefore, to indicate stateless DHCPv6, use the `ipv6 nd other-config-flag` in router interface configuration mode.

Figure 19.2: Verify Stateless DHCPv6 method

```
R1# show ipv6 interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::D68C:B5FF:FECE:A0C1
    <output omitted>
      Hosts use DHCP to obtain other configuration.
R1#
```

```
R1(config)# ipv6 unicast-routing
R1(config)#
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)#
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
R1(config-if)# end
R1# show ipv6 dhcp pool
R1# show ipv6 interface
R1# debug ipv6 dhcp detail
```

The `show ipv6 dhcp pool` command verifies the name of the DHCPv6 pool and its parameters. The `show ipv6 interface` command confirms that the interface has “Stateless address autoconfig enabled” and has an IPv6 global unicast address.

## 19.4 Stateful DHCPv6

In stateful DHCPv6, all IP information must be obtained from a stateful DHCPv6 server. However, the default router information is not from the DHCPv6 server, but it was determined by using the source IPv6 address from the RA message. This is known as *stateful* DHCPv6 because the DHCPv6 server maintains IPv6 state information.

DHCPv6 messages from the server to the client use **UDP** destination port **546**. The client sends DHCPv6 messages to the server using **UDP** destination port **547**.

There are four steps to configure a router as a DHCPv6 server:

1. Enable IPv6 Routing using `ipv6 unicast-routing` command
2. Use `ipv6 dhcp pool <pool-name>` command to create a pool and enter the router in DHCPv6 configuration mode
3. In DHCPv6 configuration mode, `address prefix <prefix/length> lifetime <time> <preferred-time>`. The `lifetime` option indicates the valid and preferred lease times in seconds.
4. (Optional) Configure DNS server address (`dns-server <ip-address>`) and domain name (`domain-name <name>`).
5. In the interface configuration mode, we bind the DHCPv6 pool to the interface using `ipv6 dhcp server <pool-name>`
6. Because the M flag indicates whether or not to use stateful DHCPv6 and the O flag is not involved, to signify stateful DHCPv6, we set the M flag as 1 using `ipv6 nd managed-config-flag` in router interface configuration mode.

Figure 19.3: Verify stateful DHCPv6

```
R1# show ipv6 interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::D68C:B5FF:FECE:A0C1
    <output omitted>
      Hosts use DHCP to obtain routable addresses.
R1#
```

```
R1(config)# ipv6 unicast-routing
R1(config)#
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)#
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
R1(config-if)# end
```

The `show ipv6 dhcp pool` command verifies the name of the DHCPv6 pool and its parameters. The `show ipv6 dhcp binding` command displays the automatic binding between the link-local address of the client and the address that the server assigns.

## 19.5 Router as DHCPv6 client

In the following commands, a Cisco router is used as the stateless DHCPv6 client. The `ipv6 enable` command is used because the router does not yet have a global unicast address. The `ipv6 address autoconfig` command enables automatic configuration of IPv6 addressing using SLAAC. By assumption, the server router is configured for stateless DHCPv6, so it sends an RA message to inform the client router to use stateless DHCPv6 to obtain DNS information.

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

## 19.6 Relay agent

If the DHCPv6 server is located on a different network than the client, the IPv6 router can be configured as a DHCPv6 relay agent using `ipv6 dhcp relay destination 2001:db8:cafe:1::6` command in interface configuration mode.

## 19.7 Message

**Router solicitation (RS)** The client sends an RS message to the router. The destination address of the message is **all-routers** multicast address **FF02::2**.

**Router advertisement (RA)** is sent by routers to provide IP configuration to clients. The RA message includes IPv6 configuration for clients (prefix, prefix-length, DNS server, MTU, and default gateway information). A router sends an RA message periodically (every 200s) or in response to an RS message. RA messages are always sent to the IPv6 **all-nodes** multicast address **FF02::1**.

The two flags are the Managed Address Configuration flag (M flag) and the Other Configuration flag (O flag). Using different combinations of the M and O flags, RA messages have one of three addressing options for the IPv6 device:

- SLAAC (M = 0, O = 0)
- SLAAC and Stateless DHCPv6 (M = 0, O = 1)
- Stateful DHCPv6 (M = 1)

# Chapter 20

## HSRP

### 20.1 Operations

One way to prevent a single point of failure at the default gateway is to implement a virtual router. Specifically, multiple routers are configured by First Hop Redundancy Protocol (FHRP) to work together as an illusion of a single router to the hosts on the LAN.

There are many options available for FHRP: HSRP (Cisco), GLBP (Cisco), VRRP (non-proprietary), IRDP (non-proprietary). This topic only introduces Hot Standby Router Protocol (HSRP), designed by Cisco. This protocol allows for gateway redundancy without any additional configuration on end devices.

HSRP selects exactly one active router and one standby router. The active router will act as the default gateway for end devices. The default gateway address is a virtual IPv4 address along with a virtual MAC address that is shared amongst both HSRP routers. End devices use this virtual IPv4 address as their default gateway address.(See figure 20.1). The virtual IPv4 address is configured by the network administrator. The virtual MAC address is created automatically.

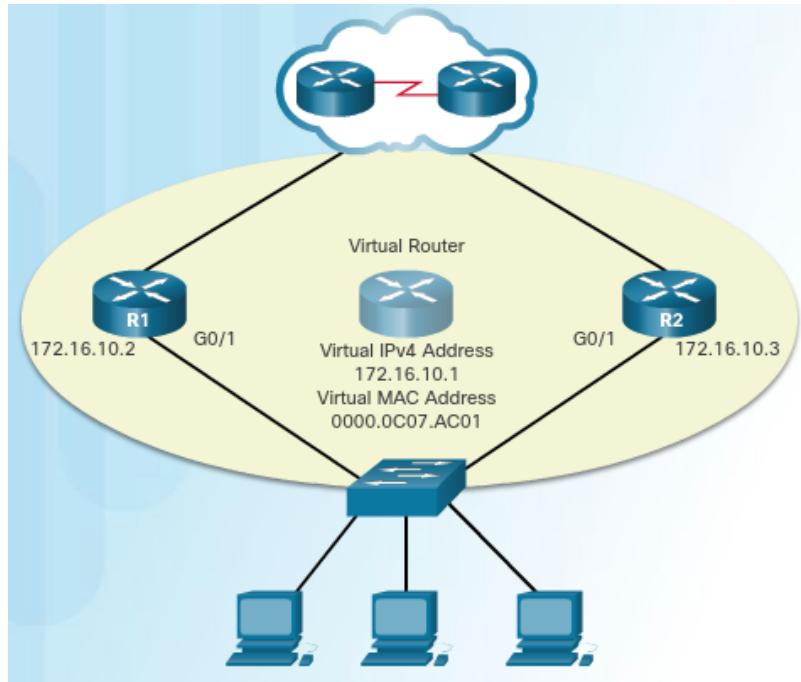


Figure 20.1: HSRP topology

## 20.2 Priority

HSRP priority can be used to determine the active router. The router with the highest HSRP priority will become the active router. By default, the HSRP priority is 100. If the priorities are equal, the router with the numerically highest IPv4 address is elected as the active router.

## 20.3 Preemption

By default, after a router becomes the active router, it will remain the active router even if another router comes online with a higher HSRP priority. This means that the router which boots up first will become the active router if there are no other routers online during the election process.

To force a new HSRP election process, preemption must be enabled. Preemption is the ability of an HSRP router to trigger the re-election process. With preemption enabled, a router that comes online with a higher HSRP priority will assume the role of the active router. Preemption only allows a router to become the active router if it has a higher priority. However, a router with equal priority will not preempt an active router, even if it has higher IPv4 address.

## 20.4 States and timers

When an interface is configured with HSRP, it exchanges HSRP hello packets to determine which router is active. Hello packets are sent to the HSRP group multicast address every 3 seconds (hello timer), by default. The standby router will become active if it does not receive a hello message from the active router after 10 seconds (hold timer). To avoid increased CPU usage and unnecessary standby state changes, do not set the hello timer below 1 second or the hold timer below 4 seconds.

## 20.5 Configuration

Complete the following steps to configure HSRP:

1. Configure HSRP version 2.
2. Configure the virtual IP address for the group.
3. Configure the priority for the desired active router to be greater than 100.
4. Configure the active router to preempt the standby router in cases where the active router comes online after the standby router.

```
R1(config)# interface g0/1
R1(config-if)# ip address 172.16.10.2 255.255.255.0
R1(config-if)# standby version 2
R1(config-if)# standby 1 ip 172.16.10.1
R1(config-if)# standby 1 priority 150
R1(config-if)# standby 1 preempt
R1(config-if)# no shutdown
```

# **Part VI**

# **INFRASTRUCTURE SECURITY**



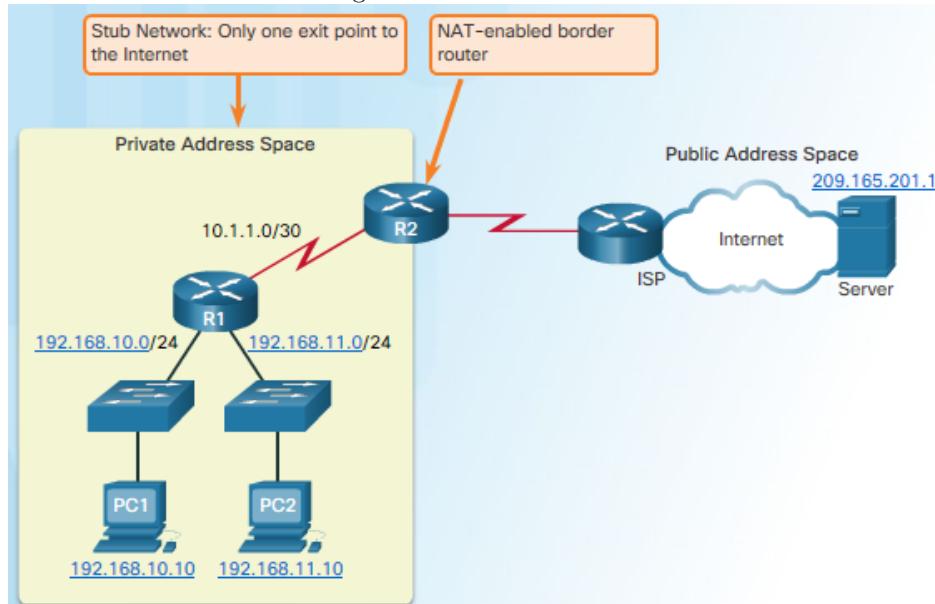
# Chapter 21

## NAT

### 21.1 What is NAT?

NAT provides the translation of private addresses to public addresses. Its primary use is to conserve public IPv4 addresses. A NAT router typically operates at the border of a stub network<sup>1</sup> (Figure 21.1). Table 21.1 shows the advantages and disadvantages of NAT.

Figure 21.1: NAT border



In NAT terminology, the inside network is the set of networks that is subject to translation. The outside network refers to all other networks. NAT includes four types of addresses:

- **Inside address:** The address of the device that NAT is translating.
- **Outside address:** The address of the destination device.
- **Local address:** A local address is any address that appears on the inside portion of the network.
- **Global address:** A global address is any address that appears on the outside portion of the network.

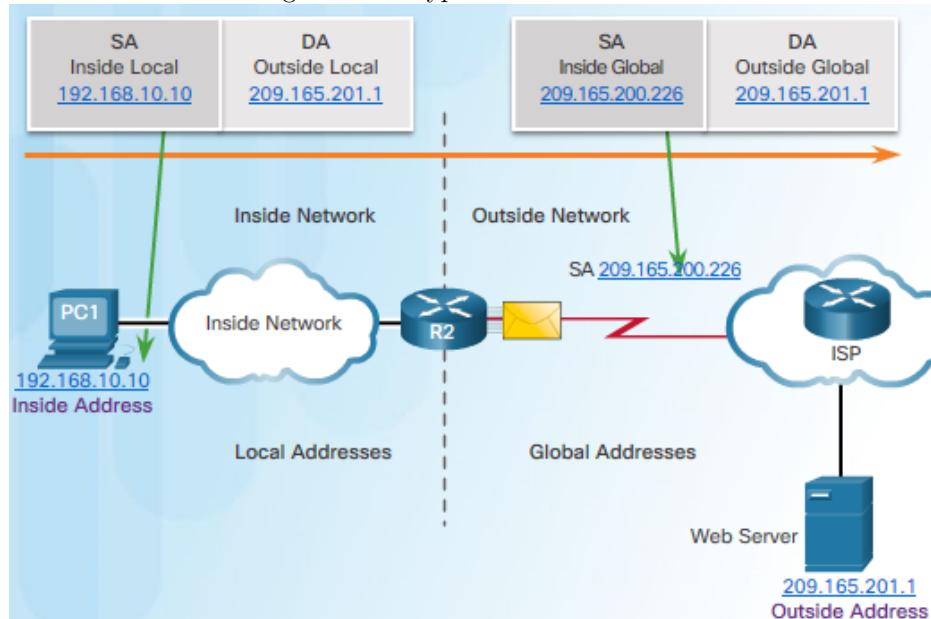
There are three types of NAT translation: static NAT, dynamic NAT, PAT (Port address translation).

<sup>1</sup>A stub network is a network that has a single connection to its neighboring network – one way in and one way out of the network.

Table 21.1: Pros and Cons of NAT

Advantages	Disadvantages
Conserves the legally registered addressing scheme by allowing the privatization of intranets	End-to-end functionality is degraded <sup>2</sup>
Increases the flexibility of connections to the public network	End-to-end IP traceability is lost
Provides consistency for internal network addressing schemes <sup>3</sup>	Tunneling becomes more complicated

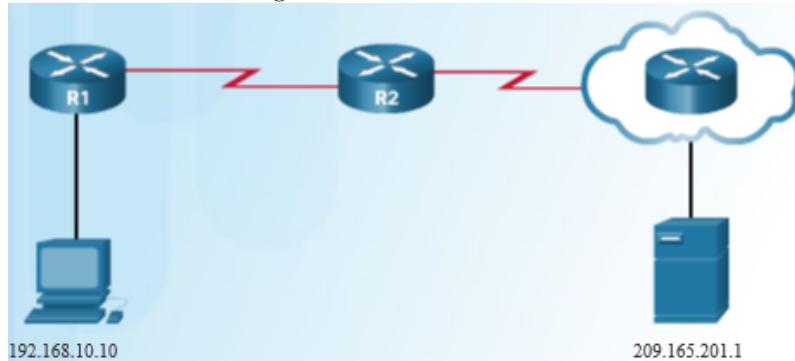
Figure 21.2: Types of NAT addresses



## 21.2 Operation

In Figure 21.3, PC1 with private address 192.168.10.10 wants to communicate with an outside web server with public address 209.165.201.1.

Figure 21.3: NAT in action



When the packet arrives at R2 (NAT-enabled router), R2 reads the source IPv4 address of the packet to determine if the packet matches the criteria specified for translation. In this case, the source IPv4 address does match the criteria and is translated from 192.168.10.10 (inside local address) to 209.165.200.226 (inside global address). R2 adds this mapping of the local to global address to the NAT table. R2 sends the packet with the translated source address toward the destination.

The web server responds with a packet addressed to the inside global address of PC1 (209.165.200.226). R2 receives the packet with destination address 209.165.200.226. R2 checks the NAT table and finds an entry for this mapping. R2 uses this information and translates the inside global address (209.165.200.226) to the inside local address (192.168.10.10), and the packet is forwarded toward PC1.

## 21.3 Static NAT

Static NAT is **one-to-one** address mapping between local and global addresses. Static NAT is particularly useful for servers or devices that must have a consistent address. There are three basic steps when configuring static NAT translations:

1. Create a mapping between the inside local address and the inside global addresses using the `ip nat inside source static <local> <global>` configuration command.
2. The interfaces participating in the translation are configured as inside or outside relative to NAT. Inside interfaces are configured with the `ip nat inside` interface configuration command, whereas the outside interface is configured with the `ip nat outside` interface configuration command.
3. Verify NAT configuration using `show ip nat translations` or `show ip nat statistics`.

```
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5
R2(config)#
R2(config)# interface Serial0/0/0
R2(config-if)# ip nat inside
R2(config-if)# interface Serial0/1/0
R2(config-if)# ip nat outside
R2(config-if)# end
```

## 21.4 Dynamic NAT

Dynamic NAT is **many-to-many** address mapping between local and global addresses. If there are 100 inside local addresses and 10 inside global addresses, at any given time only 10 of the 100 inside local addresses can be

translated. Dynamic NAT uses a pool of public addresses and assigns them on a first-come, first-served basis. There are six steps when configuring dynamic NAT translations:

1. Define the pool of addresses to be used for translation. The pool is assigned a name to identify it. Use `ip nat pool <pool-name> <start-ip> <end-ip> netmask <netmask>`.
2. Configure a standard ACL
3. Bind the ACL to the pool. Use the `ip nat inside source list <ACL-name> pool <pool-name>` global configuration.
4. The interfaces participating in the translation are configured as inside or outside relative to NAT. Inside interfaces are configured with the `ip nat inside` interface configuration command, whereas the outside interface is configured with the `ip nat outside` interface configuration command.
5. Verify NAT configuration using `show ip nat translations` or `show ip nat statistics`.

```
R2(config)# ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)#
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)#
R2(config)# ip nat inside source list 1 pool NAT-POOL1
R2(config)#
R2(config)# interface Serial0/0/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)#
R2(config)# interface Serial0/1/0
R2(config-if)# ip nat outside
R2(config-if)#+
```

## 21.5 PAT

PAT (also known as NAT overloading) is a **many-to-one** address mapping between local and global addresses. With PAT, multiple addresses can be mapped to one or to a few addresses because each private address is also tracked by a port number. For example, if there are 100 inside local addresses and 10 inside global addresses, PAT uses ports as an additional parameter to provide a multiplier effect, making it possible to reuse any one of the 10 inside global addresses.

When the NAT router receives a packet from the client, it uses its source port number to uniquely identify the specific NAT translation. PAT ensures that devices use a different TCP port number for each session with a server on the Internet. When a response comes back from the server, the source port number, which becomes the destination port number on the return trip, determines to which device the router forwards the packets.

Figure 21.4: PAT in action

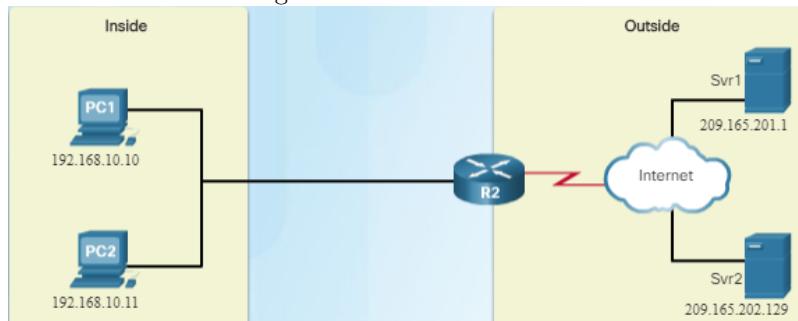


Table 21.2: PAT table of Figure 21.4

Inside local	Inside global	Outside global
192.168.10.10:1555	209.165.200.226:1555	209.165.201.1:80
192.168.10.11:1331	209.165.200.226:1331	209.165.202.129:80

In Figure 21.4, R2 uses port numbers (1331 and 1555) to identify the device from which the packet originated. In this example, the service port is 80, which is HTTP. For the source address, R2 translates the inside local address to an inside global address with the port number added. Note that, the client port numbers, 1331 and 1555, did not change at the NAT-enabled router. The destination address is the outside global IPv4 address of servers.

When there are no more ports available and there is more than one external address in the address pool, PAT moves to the next address to try to allocate the original source port. This process continues until there are no more available ports or external IPv4 addresses.

PAT also translates some protocols that do not use TCP or UDP. Each of these types of protocols is handled differently by PAT. For example, ICMPv4 query messages, echo requests, and echo replies include a Query ID. ICMPv4 uses the Query ID to identify an echo request with its corresponding echo reply.

There are six steps when configuring dynamic PAT translations. The six steps are identical to configuring dynamic NAT except for step 3.

1. Define the pool of addresses to be used for translation. The pool is assigned a name to identify it. Use `ip nat pool <pool-name> <start-ip> <end-ip> netmask <netmask>`.
2. Configure a standard ACL
3. Bind the ACL to the pool. Use the `ip nat inside source list <ACL-name> pool <pool-name> overload` in global configuration mode. The `overload` keyword is the primary difference between PAT and dynamic NAT.
4. The interfaces participating in the translation are configured as inside or outside relative to NAT. Inside interfaces are configured with the `ip nat inside` interface configuration command, whereas the outside interface is configured with the `ip nat outside` interface configuration command.
5. Verify NAT configuration using `show ip nat translations` or `show ip nat statistics`.

```
R2(config)# ip nat pool NAT-POOL2 209.165.200.226 209.165.200.240 netmask
255.255.255.224
R2(config)#
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)#
R2(config)# ip nat inside source list 1 pool NAT-POOL2 overload
R2(config)#
R2(config)# interface Serial0/0/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)#
R2(config)# interface Serial0/1/0
R2(config-if)# ip nat outside
R2(config-if)#[1]
```

There are two ways to configure PAT. In the example above, we allocate more than one public IPv4 address to the inside network, and in the following, it allocates a single public IPv4 address.

1. Configure a standard ACL

2. Bind the ACL to the pool. Use the `ip nat inside source list <ACL-name> interface <int-name> overload` in global configuration mode. The interface mentioned in this command is the one connected to the outside network.
3. The interfaces participating in the translation are configured as inside or outside relative to NAT. Inside interfaces are configured with the `ip nat inside` interface configuration command, whereas the outside interface is configured with the `ip nat outside` interface configuration command.
4. Verify NAT configuration using `show ip nat translations` or `show ip nat statistics`.

## 21.6 Port forwarding

Port forwarding forwards traffic addressed to a specific network *port* from one network node to another. In other words, in Port forwarding, NAT translates not only IP addresses but also ports that users want to access. This technique allows an external user to reach a port on a private IPv4 address (inside a LAN) from the outside, through a NAT-enabled router. A specific user usually access a single port on a server.

```
R2(config)# ip nat inside source static tcp 192.168.10.254 80 209.165.200.225 8080
R2(config)#
R2(config)# interface Serial0/0/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)#
R2(config)# interface Serial0/1/0
R2(config-if)# ip nat outside
R2(config-if)#+
```

In the example, 192.168.10.254 is the inside local IPv4 address of the web server listening on port 80. Users access this internal web server using the global IPv4 address 209.165.200.225, a globally unique public IPv4 address. The global port is configured as 8080. This will be the destination port used, along with the global IPv4 address of 209.165.200.225 to access the internal web server.

Notice within the NAT configuration the following command parameters:

- local-ip = 192.168.10.254
- local-port = 80
- global-ip = 209.165.200.225
- global-port = 8080

# Chapter 22

## ACL

### 22.1 ACL Operation Overview

An ACL contains a sequential list of permit or deny statements, known as access control entries (ACEs). ACEs are also commonly called ACL statements.

#### 22.1.1 ACEs Logic Operations

ACLs are processed in a top down manner. When an ACL is inspected, if the information in a packet header and an ACL statement match, the remaining statements are not examined, and the packet is either denied or permitted through as specified by the ACL.

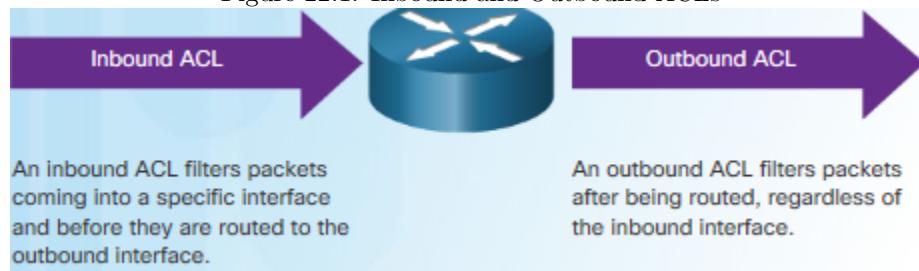
If a packet header does not match an ACL statement, the packet is tested against the next statement in the list. This matching process continues until the end of the list is reached. If no conditions match, the address is rejected. In a nut shell, ACL always stops testing conditions after the first match, therefore, the order of the ACEs is critical.

At the end of every ACL is a statement is an implicit deny any statement and because of this statement, an ACL should have at least one permit statement in it; otherwise, the ACL blocks all traffic.

#### 22.1.2 Inbound and Outbound ACL Logic

The Figure 22.1 shows the logic of routing and ACL processes. When a packet arrives at a router interface, the router checks for an ACL on the inbound interface. If an ACL exists, the packet is tested against the statements in the list.

Figure 22.1: Inbound and Outbound ACLs



If the packet matches a statement, the packet is either permitted or denied. If the packet is accepted, it is then checked against routing table entries to determine the destination interface. If a routing table entry exists for the destination, the packet is then switched to the outgoing interface, otherwise the packet is dropped.

Next, the router checks whether the outgoing interface has an ACL. If an ACL exists, the packet is tested against the statements in the list. If the packet matches a statement, it is either permitted or denied.

If there is no ACL or the packet is permitted, the packet is encapsulated in the new Layer 2 protocol and forwarded out the interface to the next device.

### 22.1.3 Numbered and Named ACLs

Standard and extended ACLs can be created using either a number or a name to identify the ACL and its list of statements.

**Numbered ACL** Assign a number based on the following rules:

- (1 to 99) and (1300 to 1999): Standard ACL
- (100 to 199) and (2000 to 2699): Extended ACL

**Named ACL** Assign a name based on the following rules:

- Cannot contain spaces or punctuation
- Names are case-sensitive
- Can contain alphanumeric characters
- It is suggested that the name be written in CAPITAL LETTER

## 22.2 Standard ACL

### 22.2.1 Overview

A standard IPv4 ACL can filter traffic based on source IP addresses only. Unlike an extended ACL, it cannot filter traffic based on Layer 4 ports.

Because standard ACLs do not specify destination addresses, place them as close to the destination as possible. If a standard ACL was placed at the source of the traffic, the *permit* or *deny* will occur based on the given source address no matter where the traffic is destined.

### 22.2.2 Standard ACL placement

In the figure 22.2, the administrator wants to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.

Following the basic placement guidelines of placing the standard ACL close to the destination, the figure shows two possible interfaces on R3 to apply the standard ACL:

- R3 S0/0/1 interface - Applying a standard ACL to prevent traffic from 192.168.10.0/24 from entering the S0/0/1 interface will prevent this traffic from reaching 192.168.30.0/24 and all other networks that are reachable by R3. This includes the 192.168.31.0/24 network. Because the intent of the ACL is to filter traffic destined only for 192.168.30.0/24, a standard ACL should not be applied to this interface.
- R3 G0/0 interface - Applying the standard ACL to traffic exiting the G0/0 interface will filter packets from 192.168.10.0/24 to 192.168.30.0/24. This will not affect other networks that are reachable by R3. Packets from 192.168.10.0/24 will still be able to reach 192.168.31.0/24.

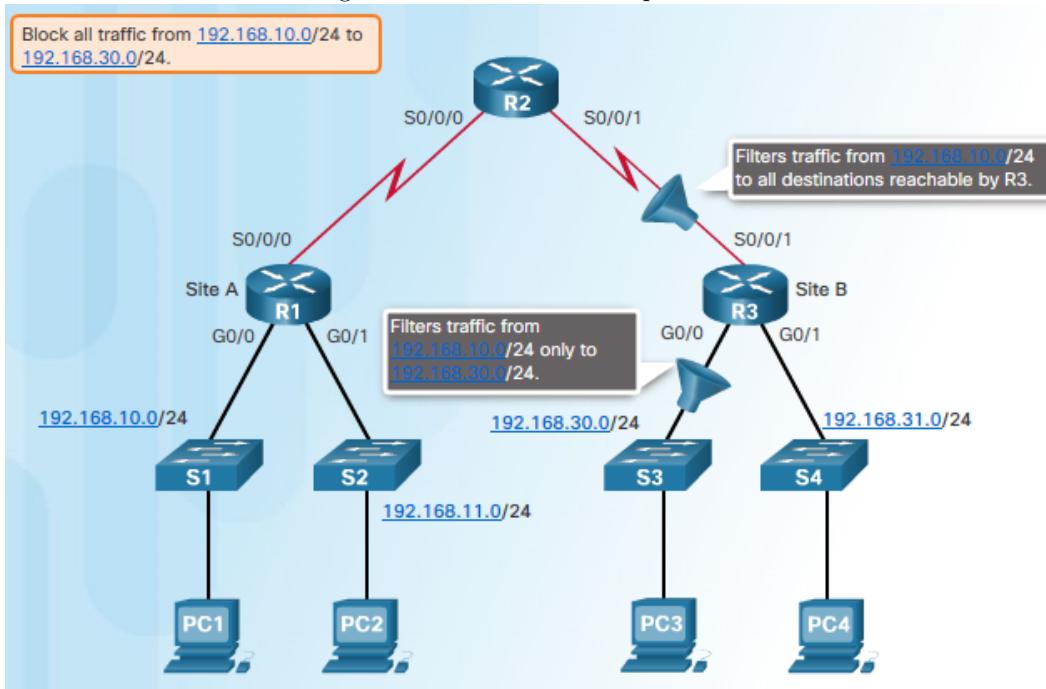
## 22.3 Extended ACLs

### 22.3.1 Overview

Extended ACLs filter packets based on:

- Protocol type (e.g. IP, ICMP, UDP, TCP)

Figure 22.2: Standard ACL placement



- Source and destination IP addresses
- Source and destination TCP and UDP ports (HTTP port 80, SSH port 22, etc.)

Extended ACLs are used more often than standard ACLs because they provide a greater degree of control. We usually locate extended ACLs as close as possible to the source of the traffic. This way, undesirable traffic is denied close to the source network without crossing the network infrastructure.

### 22.3.2 Extended ACL Placement

In figure 22.3, the administrator wants to deny Telnet and FTP traffic from the .11 network to Company B's 192.168.30.0/24 (.30, in this example) network. At the same time, all other traffic from the .11 network must be permitted to leave Company A without restriction.

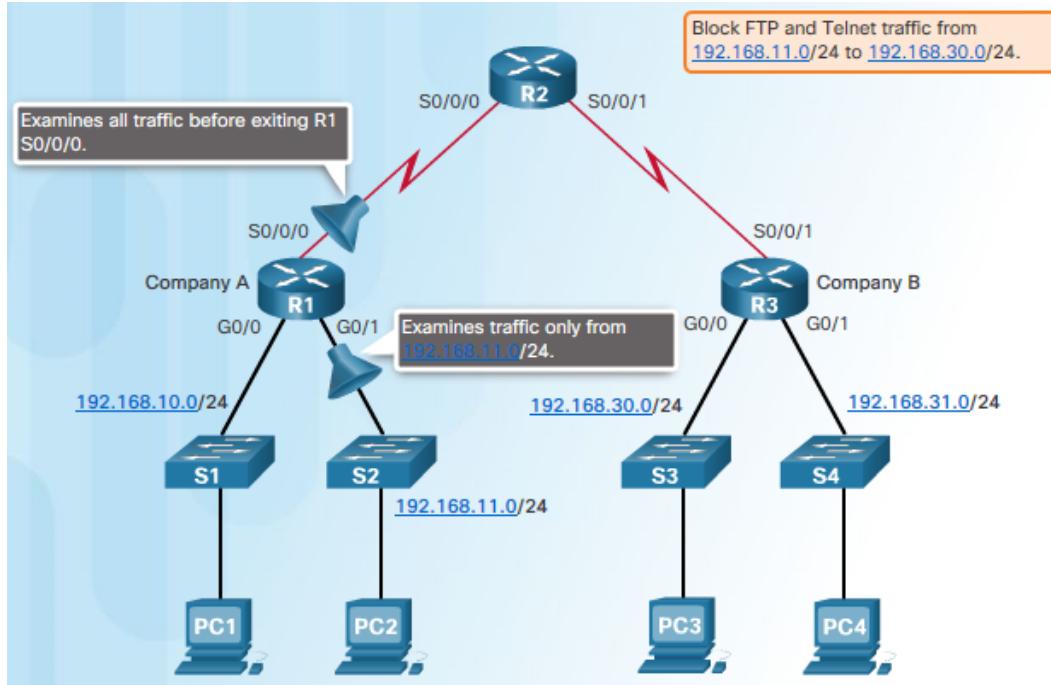
A better solution is to place an extended ACL on R1. There are two possible interfaces on R1 to apply the extended ACL:

- R1 S0/0/0 interface (outbound) - One possibility is to apply an extended ACL outbound on the S0/0/0 interface. Because the extended ACL can examine both source and destination addresses, only FTP and Telnet packets from 192.168.11.0/24 will be denied. Other traffic from 192.168.11.0/24 and other networks will be forwarded by R1. The disadvantage of placing the extended ACL on this interface is that all traffic exiting S0/0/0 must be processed by the ACL including packets from 192.168.10.0/24.
- R1 G0/1 interface (inbound) - Applying an extended ACL to traffic entering the G0/1 interface means that only packets from the 192.168.11.0/24 network are subject to ACL processing on R1. Because the filter is to be limited to only those packets leaving the 192.168.11.0/24 network, applying the extended ACL to G0/1 is the best solution.

## 22.4 IPv6 ACLs

In IPv4 there are two types of ACLs, standard and extended and both types of ACLs can be either numbered or named ACLs. With IPv6, there is only one type of ACL, which is equivalent to an IPv4 extended named ACL and

Figure 22.3: Extended ACL Placement



there are no numbered ACLs in IPv6. An IPv4 ACL and an IPv6 ACL cannot share the same name. There are three significant differences between IPv4 and IPv6 ACLs:

- The command used to apply an IPv6 ACL to an interface is `ipv6 traffic-filter` command.
- IPv6 ACLs do not use wildcard masks but instead specifies the prefix-length
- Besides `deny ipv6 any any`, An IPv6 ACL adds two implicit permit statements at the end of each IPv6 access list: `permit icmp any any nd-na` and `permit icmp any any nd-ns`

Because IPv6 ACLs must be configured with both a source and a destination, they should be applied closest to the source of the traffic.

## 22.5 Configurations

**Example 22.1.** The figure shows an example of an ACL designed to permit a single network. Only traffic from the 192.168.10.0/24 network will be permitted out the Serial 0/0/0 interface.

```
access-list 1 permit 192.168.10.0 0.0.0.255
interface s0/0/0
ip access-group 1 out
```

**Example 22.2.** Figure below shows the commands used to configure a standard named ACL on router R1, interface G0/0, which denies host 192.168.11.10 access to the 192.168.10.0 network.

```
ip access-list standard NO_ACCESS
  deny host 192.168.11.10
  permit any
  exit
interface g0/0
  ip access-group NO_ACCESS out
```

**Example 22.3.** Design an IPv4 named access list HQServer to prevent any computers attached to the g0/0 interface of the Branch router from accessing HQServer.pka (172.16.0.1). All other traffic is permitted. Configure the access list on the appropriate router, apply it to the appropriate interface and in the appropriate direction.

```
ip access-list extended HQServer
  deny ip any host 172.16.0.1
  permit ip any any
  exit
int g0/0
  ip access-group HQServer in
```

**Example 22.4.** Design an IPv4 named access list BranchServer to prevent any computers attached to the Gigabit Ethernet 0/0 interface of the HQ router from accessing the HTTP and HTTPS service of the Branch server (172.16.128.1/20). All other traffic is permitted. Configure the access list on the appropriate router, apply it to the appropriate interface and in the appropriate direction.

```
access-list extended BranchServer
  deny tcp any host 172.16.128.1 eq 80
  deny tcp any host 172.16.128.1 eq 443
  permit ip any any
  exit
int g0/0
  ip access-group BranchServer in
```

**Example 22.5.** Design an IPv6 access-list named NO-B1 to prevent any IPv6 traffic originating on B1 (2001:DB8:ACAD:B1::2/64) to reach the BranchServer.pka (2001:DB8:ACAD:B2::3/64). No traffic should be permitted from B1 to BranchServer.pka. Apply the IPv6 access to the most appropriated location (interface and direction).

```
ipv6 access-list NO-B1
  deny ipv6 host 2001:DB8:ACAD:B1::2 host 2001:DB8:ACAD:B2::3
  permit ipv6 any any
  exit
int g0/1
  ipv6 traffic-filter NO-B1 out
```

**Example 22.6.** The network administrator configured an ACL to allow users from the 192.168.10.0/24 network to browse both insecure and secure websites. In this topology (figure 22.4) the interface closest to the source of the target traffic is the G0/0 interface of R1. Web request traffic from users on the 192.168.10.0/24 LAN is inbound to the G0/0 interface. Return traffic from established connections to users on the LAN is outbound from the G0/0 interface. The example applies the ACL to the G0/0 interface in both directions. The inbound ACL, 103, checks for the type of traffic. The outbound ACL, 104, checks for return traffic from established connections. This will restrict 192.168.10.0 Internet access to allow only website browsing.

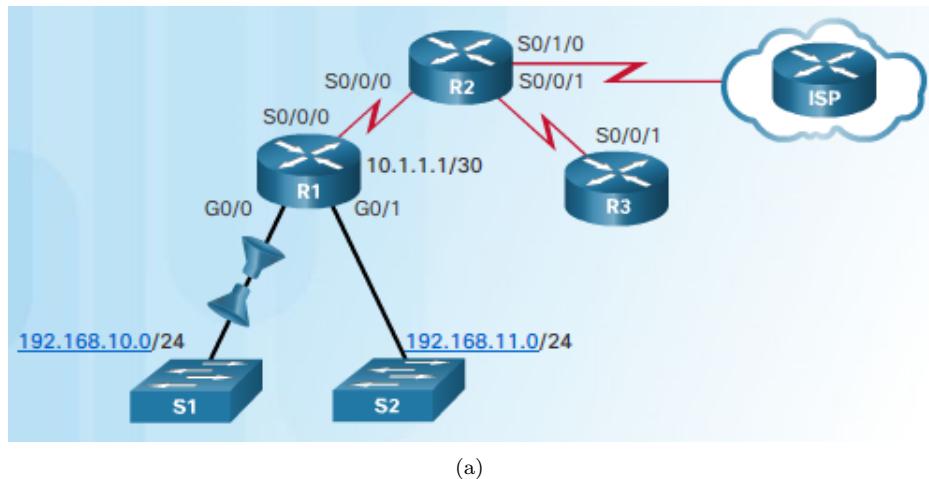
**Example 22.7.** Configure an extended IPv4 ACL named INTOHQ such that:

- Allow any hosts from the Internet to access the County DNS Svr. There should be two ACEs, one for TCP and the other UDP. Both use port 53.
- Allow any hosts from the Internet to access the County Web Svr. Only port 80 is needed.
- Allow return TCP traffic from the Internet that was initiated from the hosts in the Central networks to pass (with the established keyword).
- Apply the ACL to the Central S0/0/0 interface.

```
ip access-list extended INTOHQ
  permit tcp any host 172.16.10.5 eq 53
  permit udp any host 172.16.10.5 eq 53
  permit tcp any host 172.16.10.10 eq 80
  permit tcp any any established
  exit
int s0/0/0
  ip access-group INTOHQ in
```

**Example 22.8.** Configure an extended ACL named SNMPACCESS such that

Figure 22.4: Extended ACL example



(a)

```
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)# interface g0/0
R1(config-if)# ip access-group 103 in
R1(config-if)# ip access-group 104 out
```

(b)

- The SNMP operation runs UDP on port 161.
- Allow only the County-Admin-PC to access the Central router for the SNMP connection.
- SNMP connections from other hosts on the Central LAN should fail.
- Allow all other IP traffic.
- Apply this ACL on the Central router, G0/0 interface.

```
ip access-list extended SNMPACCESS
  permit udp host 192.168.10.5 host 192.168.10.1 eq 161
  deny udp any host 192.168.10.1 eq 161
  permit ip any any
  exit
interface g0/0
  ip access-group SNMPACCESS in
```

## 22.6 Troubleshoot

Using the `show access-lists` command to reveal most of the common ACL errors. The most common errors are entering ACEs in the wrong order and not applying adequate criteria to the ACL rules. Following these steps to troubleshoot ACL:

1. Check the criteria of ACL rules
2. Check the order of ACEs
3. Check the direction of ACL (inbound, outbound)
4. Check the location of ACL (which router, which interface). Remember that extended ACLs are placed as close as possible to the source and standard ACLs are placed as close as possible to the destination.

**Example 22.9.** The 192.168.10.0/24 network cannot use TFTP to connect to the 192.168.30.0/24 network.

```
R3# show access-lists 120
Extended IP access list 120
  10 deny tcp 192.168.10.0 0.0.0.255 any eq telnet
  20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.31.12 eq smtp
  30 permit tcp any any
```

The 192.168.10.0/24 network cannot use TFTP to connect to the 192.168.30.0/24 network because TFTP uses the transport protocol UDP. Statement 30 in access list 120 allows all other TCP traffic. However, because TFTP uses UDP instead of TCP, it is implicitly denied. Recall that the implied deny any statement does not appear in show access-lists output and therefore matches are not shown. Statement 30 should be `permit ip any any`.

**Example 22.10.** The 192.168.11.0/24 network can use Telnet to connect to 192.168.30.0/24, but according to company policy, this connection should not be allowed. The results of the show access-lists 130 command indicate that the permit statement has been matched.

```
R1# show access-lists 130
Extended IP access list 130
  10 deny tcp any eq telnet any
  20 deny tcp 192.168.11.0 0.0.0.255 host 192.168.31.12 eq smtp
  30 permit tcp any any (12 match(es))
```

The 192.168.11.0/24 network can use Telnet to connect to the 192.168.30.0/24 network because the Telnet port number in statement 10 of access list 130 is listed in the wrong position in the ACL statement. Statement 10 currently denies any source packet with a port number that is equal to Telnet. To deny Telnet traffic inbound on G0/1, deny the destination port number that is equal to Telnet, for example, `10 deny tcp 192.168.11.0 0.0.0.255 192.168.30.0 0.0.0.255 eq telnet`.



## **Part VII**

# **INFRASTRUCTURE MANAGEMENT**



# Chapter 23

# Network Security and Monitoring

## 23.1 Security attacks

### 23.1.1 CDP Reconnaissance Attack

The Cisco Discovery Protocol (CDP) is enabled on Cisco devices by default. CDP broadcasts are sent unencrypted and unauthenticated. Therefore, an attacker could interfere with the network infrastructure by sending crafted CDP frames containing bogus device information to directly-connected Cisco devices. To mitigate the exploitation of CDP, limit the use of CDP on devices or ports. For example, disable CDP on edge ports that connect to untrusted devices.

### 23.1.2 Telnet Attacks

There are two types of Telnet attacks:

- Brute Force Password Attack: The attacker tries to discover the administrative password.
- Telnet DoS Attack: The attacker continuously requests Telnet connections in an attempt to render the Telnet service unavailable and preventing an administrator from remotely accessing a device.

### 23.1.3 MAC Address Table Flooding Attack

MAC address tables are limited in size. MAC flooding attacks exploit this limitation with fake source MAC addresses until the switch MAC address table is full. When the MAC address table becomes full of fake MAC addresses, the switch enters into what is known as fail-open mode. In this mode, the switch broadcasts all frames to all machines on the network. As a result, the attacker can capture all of the frames, even frames that are not addressed to its MAC address table. Configure **port security** on the switch to mitigate MAC address table overflow attacks.

### 23.1.4 VLAN Attacks

The attacker attempts to gain VLAN access by configuring a host to trunk with the connecting switch. If successful, the switch establishes a trunk link with the host and the attacker can then access all the VLAN traffic on the switch. The best way to prevent basic VLAN attacks:

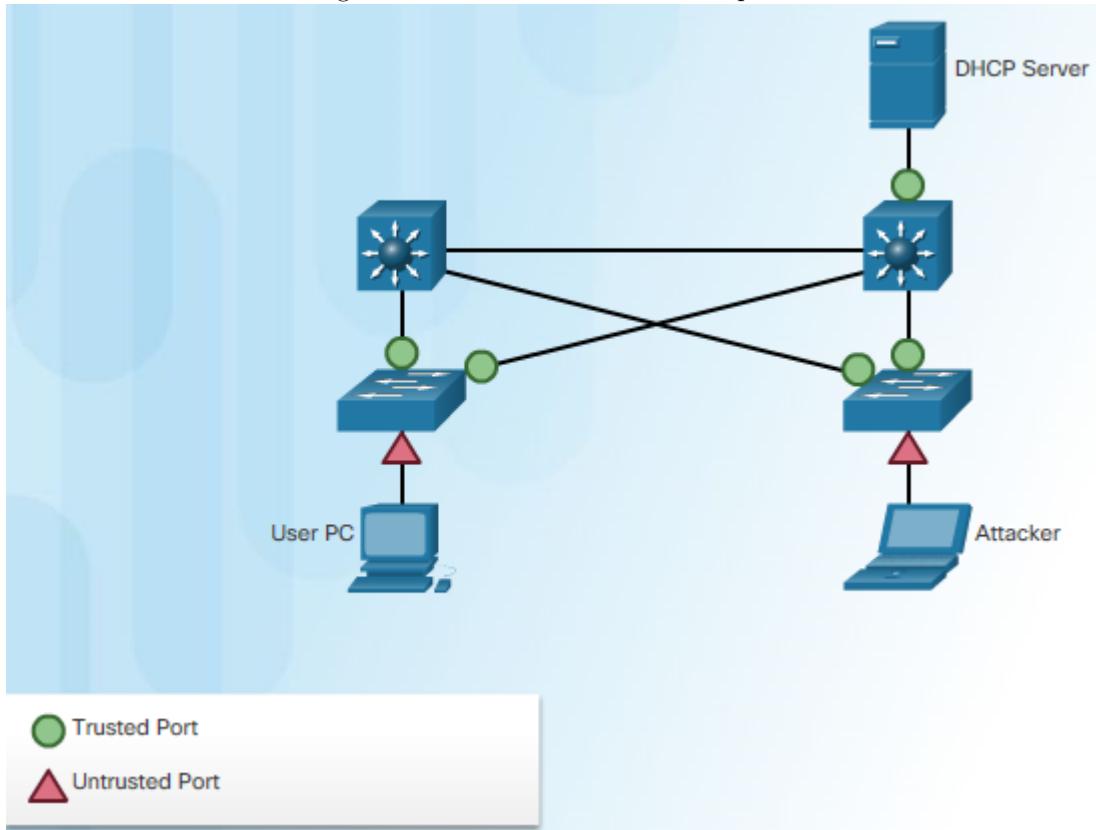
- Disable DTP negotiations on non-trunking ports using the `switchport nonegotiate` interface configuration command.
- Manually enable the trunk link using the `switchport mode trunk` interface configuration command.
- Manually enable access ports using the `switchport mode access` interface configuration command.
- Set the native VLAN to be something other than VLAN 1.
- Administratively shut down unused ports, and assign them to an unused VLAN.

### 23.1.5 DHCP Attacks

**DHCP spoofing attack:** A DHCP spoofing attack occurs when a rogue DHCP server is connected to the network and provides false IP configuration parameters to legitimate clients. Use *DHCP snooping* to mitigate DHCP spoofing attacks.

**DHCP starvation attack:** An attacker floods the DHCP server with bogus DHCP requests and eventually leases all of the available IP addresses in the DHCP server pool. After these IP addresses are issued, the server cannot issue any more addresses, and this situation produces a DoS attack<sup>1</sup> as new clients cannot obtain network access.

Figure 23.1: Trusted and Untrusted ports



DHCP snooping recognizes two types of ports (see figure 23.1):

- **Trusted DHCP ports:** Only ports connecting to *upstream* DHCP servers should be trusted. Trusted ports must be explicitly identified in the configuration.
- **Untrusted ports:** These ports connect to hosts that should not be providing DHCP server messages. By default, all switch ports are untrusted.

### 23.1.6 Cisco solution

There are four Cisco switch security solutions to help mitigate Layer 2 attacks:

- Port security prevents MAC address flooding, DHCP starvation
- DHCP snooping prevents DHCP spoofing and DHCP starvation
- DAI (Dynamic ARP inspection) prevents ARP spoofing and ARP poisoning
- IPSG (IP Source Guard) prevents MAC and IP address spoofing

<sup>1</sup>A DoS attack is any attack that is used to overload specific devices and network services with illegitimate traffic, thereby preventing legitimate traffic from reaching those resources.

### 23.1.7 The AAA framework

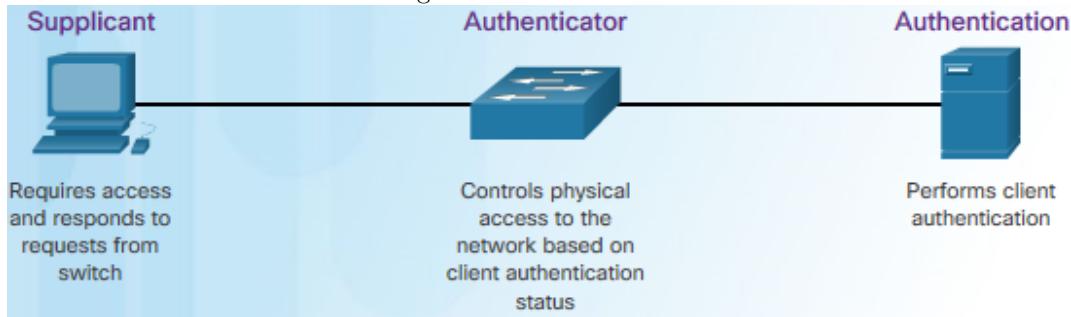
The Authentication, Authorization, and Accounting (AAA) framework is used to secure device access. An AAA-enabled router uses either TACACS+ or RADIUS protocol to communicate with the AAA server. TACACS+ is considered the more secure protocol, because all TACACS+ protocol exchanges are encrypted, while RADIUS only encrypts the user's password. RADIUS does not encrypt user names, accounting information, or any other information carried in the RADIUS message. Cisco provides two common methods of implementing AAA services:

- **Local AAA:** use a local database for authentication, store usernames and passwords locally in the Cisco router, and users authenticate against the local database. Local AAA is ideal for small networks.
- **Server-Based AAA Authentication:** The AAA server contains the usernames and password for all users and serves as a central authentication system for all infrastructure devices.

### 23.1.8 802.1X

The IEEE 802.1X standard defines a port-based access control and authentication protocol. IEEE 802.1X restricts unauthorized workstations from connecting to a LAN through publicly accessible switch ports.

Figure 23.2: 802.1X roles



With 802.1X port-based authentication, the devices in the network have specific roles, as shown in the figure 23.2:

- Client (Supplicant): The device is a PC running 802.1X-compliant client software.
- Switch (Authenticator): This controls physical access to the network based on the authentication status of the client. The switch requests identifying information from the client, verifies that information with the authentication server, and relays a response to the client.
- Authentication server: validates the identity of the client and notifies the switch or other authenticator such as a wireless access point whether the client is authorized to access the LAN and switch services.

## 23.2 SNMP

### 23.2.1 Introduction to SNMP

Simple Network Management Protocol (SNMP) was developed to allow administrators to manage nodes such as servers, workstations, routers, switches, and security appliances, on an IP network. The SNMP system consists of three elements:

- **SNMP manager:** a part of a network management system (NMS), run SNMP management software.
- **SNMP agents** (managed node): responsible for providing access to the MIB which resides on each SNMP client device.
- **MIB** (Management Information Base): store data about the device and operational statistics

### 23.2.2 SNMP requests

The SNMP manager uses the get and set actions to perform the operations, as described in the Figure 23.1.

Table 23.1: SNMP operations

Operation	Description
get-request	Retrieves a value from a specific variable
get-next-request	Retrieves a value from a variable within a table
get-bulk-request	Retrieve large block of data such as multiple rows in a table
get-response	Replies to a get-request, get-next-request, and set-request
set-request	Stores a value in a specific variable

### 23.2.3 SNMP Agent Traps

An NMS periodically polls the SNMP agents. Using this process, a network management application can collect information to monitor traffic loads and to verify device configurations. Periodic SNMP polling does have disadvantages. First, there is a delay between the time that an event occurs and the time that it is noticed (via polling) by the NMS. Second, there is a trade-off between polling frequency and bandwidth usage.

To mitigate these disadvantages, it is possible for SNMP agents to generate and send traps to inform the NMS immediately of certain events. Traps are unsolicited messages alerting the SNMP manager to a condition or event on the network.

### 23.2.4 Community string and Object ID

SNMPv1 and SNMPv2c use community strings as plaintext password to control access to the MIB. There are two types of community strings: Read-only (**ro**) and Read-write (**rw**).

MIB saves data in variables and organizes them hierarchically. Formally, the MIB defines each variable as an Object ID (OID). OIDs uniquely identify managed objects in the MIB hierarchy (figure 23.3). For example, OIDs belonging to Cisco, are numbered as follows: .iso (1).org (3).dod (6).internet (1).private (4).enterprises (1).cisco (9). Therefore the OID is 1.3.6.1.4.1.9.

### 23.2.5 Configuration

#### SNMPv2

1. (Required) Configure the community string and access level (read-only or read-write).

```
R1(config)# snmp-server community batonaug ro
```

2. (Optional) Restrict SNMP access to NMS hosts (SNMP managers) using ACL. Step 1 and 2 can be done in one command.

```
R1(config)# snmp-server community batonaug ro SNMP_ACL
```

3. (Optional) Enable traps on an SNMP agent. If no trap notification types are specified in this command, then all trap types are sent. Repeated use of this command is required if a particular subset of trap types is desired.

```
R1(config)# snmp-server enable traps
```

4. (Optional) Specify the recipient of the SNMP trap operations.

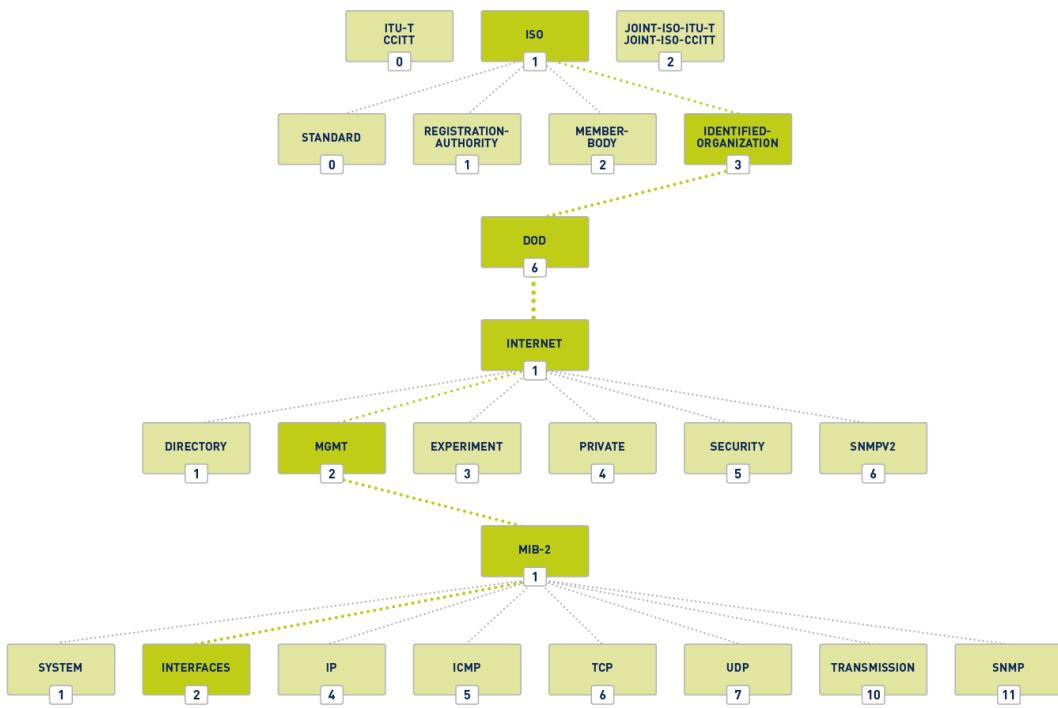
```
R1(config)# snmp-server host 192.168.1.3 version 2c batonaug
```

5. (Optional) Document the location of the SNMP server and the system contact.

```
R1(config)# snmp-server location NOC_SNMP_MANAGER
```

```
R1(config)# snmp-server contact Wayne World
```

Figure 23.3: OID tree



**Note!** To verify SNMP configuration, use any of the variations of the `show snmp` command.

**Note!** Only the first step is required, the rest are optional.

**Note!** By default, SNMP does not have any traps set. Without this command, SNMP managers must poll for all relevant information.

### SNMPv3

SNMPv3 provides three security features: Message integrity and authentication, Encryption, Access control. The following commands show an example of basic SNMPv3 configuration:

```

1 R1(config)# snmp-server view SNMP-RO iso included
2 R1(config)# snmp-server group ADMIN v3 priv read SNMP-RO access PERMIT-ADMIN
3 R1(config)# snmp-server user BOB ADMIN v3 auth sha cisco12345 priv aes 128 cisco54321
4 R1(config)# end
  
```

**Line 1.** Create an SNMP view `SNMP-RO` and include the entire ISO tree from the MIB.

**Line 2.** Create an SNMP group `ADMIN`; Set to version 3 (v3) with authentication and encryption required (`priv`). The group is allowed read-only access to the view `SNMP-RO` (`read SNMP-RO`). Access for the group is limited by the an ACL (`access PERMIT-ADMIN`).

**Line 3.** Create new SNMP user `BOB` as a member of the group `ADMIN`; Set to version 3 (v3) with authentication SHA (`auth sha`) and password `cisco12345`; Enable 128-bit AES encryption `priv aes 128` with password `cisco54321`.

## 23.3 SPAN

### 23.3.1 Introduction

A packet analyzer (such as Wireshark) is typically software that captures packets entering and exiting a network interface card (NIC). However, the basic operation of a modern switched network disables the packet analyzer ability to capture traffic from other sources. For instance, a user running Wireshark can only capture traffic going to their NIC.

The solution to this dilemma is to enable *port mirroring*. The port mirroring feature allows a switch to copy and send Ethernet frames from specific ports to the destination port connected to a packet analyzer. The Switched Port Analyzer (SPAN) feature on Cisco switches is a type of port mirroring. SPAN is commonly implemented to deliver traffic to specialized devices including: Packet analyzers and IPSS (Intrusion Prevention Systems).

There are three important things to consider when configuring SPAN:

- The destination port cannot be a source port, and the source port cannot be a destination port.
- The number of destination ports is platform-dependent. Some platforms allow for more than one destination port.
- The destination port is no longer a normal switch port. Only monitored traffic passes through that port.

**Local SPAN** is when traffic is mirrored to another port on the same switch. A SPAN session is the association between source ports (or VLANs) and a destination port. Traffic entering or leaving the source port (or VLAN) is replicated on the destination port.

**Remote SPAN (RSPAN)** allows source and destination ports to be in different switches. RSPAN uses two sessions. One session is used as the source and one session is used to copy or receive the traffic from a VLAN. The traffic for each RSPAN session is carried over trunk links in a user-specified RSPAN VLAN that is dedicated (for that RSPAN session) in all participating switches.

### 23.3.2 Configuration

```
S1(config)# monitor session 1 source interface f0/1
S1(config)# monitor session 1 destination interface f0/2
S1(config)# end
S1# show monitor
```

The above command is used to associate a source port and a destination port with a SPAN session. A separate `monitor session` command is used for each session. A VLAN can be specified instead of a physical port.

Figure 23.4: Verifying SPAN

```
S1# show monitor
Session 1
-----
Type          : Local Session
Source Ports  :
    Both      : Fa0/1
Destination Ports:
    : Fa0/2
Encapsulation  :
    : Native
Ingress       :
    : Disabled
```

```
S1# show monitor
Session 1
-----
```

```
Type          : Local session
Source port   :
  Both        : Fa0/1
Destination ports : Fa0/2
  Encapsulation : Native
  Ingress       : Disabled
```

The above output shows that the session 1 of a Local SPAN (Type: Local session), the source port for both traffic directions (receive and transmit) is F0/1 (Both: Fa0/1), and the destination port is F0/2. The ingress SPAN is disabled on the destination port, so only traffic that leaves the destination port is copied to that port.



# Chapter 24

## Troubleshooting

### 24.1 Documentation

For network administrators to be able to monitor and troubleshoot a network, they must have a complete set of accurate and current network documentation. This documentation includes:

- Configuration files, including network configuration files and end-system configuration files
- Physical and logical topology diagrams
- A network baseline

#### 24.1.1 Configuration files

Network configuration files contain accurate, up-to-date records of the hardware (routers, switches, cables etc.) and software (routing protocols, IOS, etc.) used in a network. End-system configuration files focus on the hardware and software used in end-system devices, such as servers, network management consoles, and user workstations. See also figures 24.1 and 24.2.

Figure 24.1: Network configuration file

Switch Information	Port	Speed	Duplex	STP	Port Fast	Trunk Status	Ether Channel L2 or L3	VLANs	Key
S1, Cisco WS-2960-24TT, <a href="http://192.168.10.2">192.168.10.2</a> /24, 2001:db6:acad:99::2, c2960-lanbasek9-mz.150-2.SE7.bin	G0/1	100 Gb/s	Auto	Fwd	No	On	None	1	Connects to R1
	F0/2	100 Mb/s	Auto	Fwd	Yes	No	None	1	Connects to PC1

Figure 24.2: End-system configuration file

Device Name, Purpose	Operating System	MAC Address	IP Address	Default Gateway
PC2	Windows 10	5475.D08E.9AD8	<a href="http://192.168.11.10">192.168.11.10</a> /24 2001:DB8:ACAD:11::10/64	<a href="http://192.168.11.1">192.168.11.1</a> /24 2001:DB8:ACAD:11::1
SRV1	Linux	000C.D991.A138	<a href="http://192.168.20.254">192.168.20.254</a> /24 2001:DB8:ACAD:4::100/64	<a href="http://192.168.20.1">192.168.20.1</a> /24 2001:DB8:ACAD:4::1

### 24.1.2 Topology diagrams

There are two types of network topology diagrams: the physical topology and the logical topology. A physical network topology shows the physical layout of the devices connected to the network. It is useful when troubleshooting physical layer problems. A logical network topology illustrates how devices are logically connected to the network, meaning how devices actually transfer data across the network when communicating with other devices. Symbols are used to represent network elements.

### 24.1.3 Network baseline

A baseline is used to establish normal network or system performance. Establishing a network performance baseline requires collecting performance data from the ports and devices that are essential to network operation. A network baseline helps:

- monitor network behavior
- keep track of the performance
- keep track of the traffic patterns
- check whether the current network design can meet business requirements
- measure the optimum nature of network traffic and congestion levels
- show the true nature of congestion or potential congestion in a network
- reveal areas in the network that are underutilized

To establish and capture an initial network baseline, perform the following steps:

1. Determine what types of data to collect
2. Identify devices and ports of interest
3. Determine the baseline duration (a baseline needs to last no more than six weeks, a two-to-four-week baseline is adequate.)

Baseline measurements should not be performed during times of unique traffic patterns, because the data would provide an inaccurate picture of normal network operations. Baseline analysis of the network should be conducted on a regular basis. Perform an annual analysis of the entire network or different sections of the network on a rotating basis. Analysis must be conducted regularly to understand how the network is affected by growth and other changes.

## 24.2 Troubleshooting process

### 24.2.1 General procedures

1. **Gather symptoms:** the network administrator determines which network components have been affected and how the functionality of the network has changed compared to the baseline.
2. **Isolate the problem:** Isolating is the process of eliminating variables until a single problem, or a set of related problems has been identified as the cause.
3. **Implement corrective action:** implementing, testing, and documenting possible solutions.

## Gathering symptoms

There are five information gathering steps:

1. **Gather information:** Get information from trouble ticket or questioning users. Table 24.2.1 provides some guidelines and sample end-user question.
2. **Determine ownership:** If the problem is outside the boundary of the organization's control, contact an administrator for the external system.
3. **Narrow the scope:** Determine in which layer the problem occurs (core, distribution, or access layer).
4. **Gather symptoms from suspect devices:** Use a layered troubleshooting approach and Gather hardware and software symptoms. To gather symptoms, use Cisco IOS commands (ping, traceroute, telnet, show, debug) or packet captures, device logs.
5. **Document symptoms**

Guidelines	Sample questions
Ask questions that are pertinent to the problem	What does not work?
Questions that help eliminate or discover the possible problems	Are things that do work and the things that do not work related?
Speak at technical level that the user can understand	Has the thing that does not work ever worked?
Ask the user when the problem was first noticed	When was the problem first noticed?
Determine whether anything unusual since the last time it worked	What has changed since the last time it did work?
Recreate the problem	Can you reproduce the problem?
What happened before the problem occurred	When exactly does the problem occur?

## Implement corrective action

The severity of the problem should be weighed against the impact of the solution. For example, if a critical server or router must be offline for a significant amount of time, it may be better to wait until the end of the workday to implement the fix. This is called **change-control procedures**.

If the corrective action creates another problem or does not solve the problem, the attempted solution is documented, the changes are removed, and the network administrator returns to gathering symptoms and isolating the issue.

### 24.2.2 Troubleshooting methods

**Bottom-up** you start with the physical components of the network and move up through the layers of the OSI model until the cause of the problem is identified. Bottom-up troubleshooting is a good approach to use when the problem is suspected to be a physical one.

**Top-down** starts with the end-user applications and moves down through the layers of the OSI model until the cause of the problem has been identified. Use this approach for simpler problems, or when you think the problem is with a piece of software. The disadvantage with the top-down approach is it requires checking every network application until the possible cause of the problem is found.

**Divide-conquer** The network administrator selects a layer and tests in both directions from that layer. You make an informed guess as to which OSI layer to start your investigation. When a layer is verified to be functioning properly, it can be assumed that the layers below it are functioning. The administrator can work up the OSI layers. If an OSI layer is not functioning properly, the administrator can work down the OSI layer model.

**Educated guess** The network administrator guesses the solution based on the symptoms of the problem. This method is more successfully implemented by seasoned network administrators, because seasoned network administrators rely on their extensive knowledge and experience to decisively isolate and solve network issues.

**Comparison** Comparing a working to a non-working situation involves comparing configurations, software versions, and hardware changes. The aim is to identify the changes that led to a non-working environment. Using this method may lead to a working solution, but without clearly revealing the cause of the problem. This method can be helpful when the network administrator is lacking an area of expertise, or when the problem needs to be resolved quickly. After the fix has been implemented, the network administrator can do further research on the actual cause of the problem.

**Substitution** involves swapping the problematic device with a known, working one. If the problem is fixed, that the network administrator knows the problem is with the removed device. If the problem remains, then the cause may be elsewhere. In specific situations, this can be an ideal method for quick problem resolution.

## 24.3 Using IP SLA

### 24.3.1 Introduction

Network engineers use IP SLAs to simulate network data and IP services to collect network performance information in real time. Multiple IP SLA operations may be configured on a device. There are additional benefits for using IP SLAs:

- Service-level agreement monitoring, measurement, and verification
- Network performance monitoring
- IP service network health assessment to verify that the existing QoS is sufficient for IP services
- Edge-to-edge network availability monitoring for proactive connectivity verification of network resources

Instead of using ping manually, a network engineer can use the IP SLA ICMP Echo operation to test the availability of network devices. The IP SLA ICMP Echo operation provides the following measurements:

- Availability monitoring (packet loss statistics)
- Performance monitoring (latency and response time)
- Network operation (end-to-end connectivity)

### 24.3.2 Configuration

To create an IP SLA operation and enter IP SLA configuration mode, use the `ip sla <operation-number>` global configuration command. From IP SLA configuration mode, you can configure the IP SLA operation as an ICMP Echo operation and enter ICMP echo configuration mode using the following command:

```
icmp-echo {dest-ip-address | dest-hostname} [source-ip {ip-address | hostname} | source-interface interface-id]
```

Next, set the rate at which a specified IP SLA operation repeats using the `frequency <seconds>` command. To schedule the IP SLA operation, use the following global configuration command:

```
ip sla schedule operation-number [life {forever | seconds}] [start-time {hh:mm[:ss] [month day | day month]
| pending | now] after hh:mm:ss} [ageout seconds]
[ recurring ]
```

### 24.3.3 Sample

The configuration below configures an IP SLA operation with an operation number of 1. Each operation can be referred to by its operation-number. The `icmp-echo` command identifies the destination address to be monitored. The frequency command is setting the IP SLA rate to 30 second intervals. The `ip sla schedule` command is scheduling the IP SLA operation number 1 to start immediately (now) and continue until manually cancelled (forever).

**Listing 24: IP SLA example**

```
ip sla 1
  icmp-echo 192.168.1.5
  frequency 30
  exit
ip sla schedule 1 start-time now life forever
```

## 24.4 Troubleshooting tools

### 24.4.1 Software

**Network management system (NMS)** includes device-level monitoring, configuration, and *fault-management* tools. These tools can be used to investigate and correct network problems.

**Knowledge base** is a *vendor-based webpage* that provides information on hardware and software. It contains troubleshooting procedures, implementation guides, and original white papers on most aspects of networking technology.

**Baselining tools** automate the network documentation and baselining process. For example, they can draw network diagrams, help keep network software and hardware documentation up-to-date, and help to cost-effectively measure baseline network bandwidth use.

**Protocol analyzer** useful to *investigate packet* content while flowing through the network. The information displayed by a protocol analyzer includes the physical, data link, protocol, and descriptions for each frame. Protocol analyzers such as Wireshark can help troubleshoot network performance problems, verify authentication, etc.

### 24.4.2 Hardware

**Digital multimeters (DMMs)** are test instruments that are used to directly measure electrical values of *voltage*, current, and resistance. We use them to check power supply voltage levels.

**Cable testers** are designed for testing *data communication cabling*. It can be used to detect broken wires, crossed-over wiring, shorted connections, and improperly paired connections. There is one type of cable testers called time-domain reflectometers (TDRs). These devices are used to pinpoint the *distance* to break in a cable.

**Cable analyzers** are used to test and *certify copper* and *fiber* cables. It can detect near-end *crosstalk* (NEXT), return loss (RL), etc.

**Portable network analyzers** are used for troubleshooting *VLANs* and *switched* networks. Using this device, the network engineer can view interface details, see which switch port is connected to which device, discover VLAN configuration, analyze network traffic, etc.

**Network analysis module (NAM)** provides embedded *browser-based* interface that generates report on the network *resources*.

## 24.5 Symptoms and causes

### 24.5.1 Physical layer

- **Performance lower than baseline:** overloaded or underpowered servers, unsuitable switch or router configurations, traffic congestion on a low-capacity link, chronic frame loss.
- **Loss of connectivity:** a cable or device fails, a loose or oxidized connection.
- **Network bottlenecks or congestion:** a router, interface, or cable fails.
- **High CPU utilization rates:** a device, such as a router, switch, or server, is operating at or exceeding its design limits

### 24.5.2 Data Link layer

- **Encapsulation errors:** the encapsulation at one end of a WAN link is configured differently from the encapsulation used at the other end.
- **Address mapping errors:** devices are configured not to respond to ARP requests, invalid ARP replies are received, a security attack, etc.
- **Framing errors:** A framing error occurs when a frame does not end on an 8-bit byte boundary. Causes: noisy serial line, an improperly designed cable (too long or not properly shielded), faulty NIC, *duplex mismatch*, or an incorrectly configured CSU line clock.
- **STP failures or loops:** STP topology changes, inconsistent configuration of STP timers, overloaded CPU.

### 24.5.3 Network layer

- **General network issues:** change in the topology (down link, install new routes, etc.), someone is currently working on the network infrastructure
- **Connectivity issues:** power problems, cabling problems, bad ports, and ISP problems.
- **Routing table:** missing or unexpected routes, improper routing protocol configuration
- **Neighbor issues:** routing protocol configuration, down link, routing protocol authentication

### 24.5.4 Transport layer

The most common issues with ACLs are caused by improper configuration. There are several areas where ACL misconfigurations commonly occur:

- Wrong interface or traffic direction (inbound, outbound)
- Order of access control entries
- Addresses and IPv4 wildcard masks
- Selection of transport layer protocol

## 24.6 Troubleshooting end-to-end connectivity

When there is no end-to-end connectivity, and the administrator chooses to troubleshoot with a bottom-up approach, these are common steps the administrator can take:

1. Check physical connectivity at the point where network communication stops. The problem might be with a faulty cable or interface, or involve misconfigured or faulty hardware.
2. Check for duplex mismatches.

3. Check data link and network layer. This includes IPv4 ARP tables, IPv6 neighbor tables, MAC address tables, and VLAN assignments.
4. Verify Default Gateway
5. Verify correct route from the source to the destination.
6. Verify the transport layer
7. Verify ACLs
8. Ensure that DNS settings are correct. There should be a DNS server that is accessible.

#### 24.6.1 Check physical connectivity

The most commonly used Cisco IOS commands for this purpose are `sh process cpu`, `sh memory`, and `sh int`. This topic discusses the `sh int` command. The output of the `sh int` command in the figure lists a number of important statistics that can be checked:

Figure 24.3: The output of show interface command

```
R1# show interfaces GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
Hardware is CN Gigabit Ethernet, address is
d48c.b5ce.a0c0(bia d48c.b5ce.a0c0)
Internet address is 10.1.10.1/24
<output omitted>
Input queue: 0/75/0/0 (size/max/drops/flushes); Total
output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
85 packets input, 7711 bytes, 0 no buffer
Received 25 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 5 multicast, 0 pause input
10112 packets output, 922864 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
11 unknown protocol drops
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out
R1#
```

**Input queue drops** signify that at some point, more traffic was delivered to the router than it could process. This could be normal during traffic peaks. However, it could be an indication that the CPU cannot process packets in time.

**Output queue drops** indicate that packets were dropped due to congestion on the interface. Seeing output drops is normal for any point where the aggregate input traffic is higher than the output traffic. However, consistently seeing output drops can be an indicator that you need to implement or modify QoS.

**Input errors** indicate errors that are experienced during the reception of the frame, such as CRC errors. High numbers of CRC errors could indicate cabling problems, interface hardware problems, or duplex mismatches.

**Output errors** indicate collisions during the transmission of a frame, and collisions, especially late collisions, often indicate duplex mismatches.

The network administrator corrects duplex mismatches using `duplex auto` interface configuration command.

### 24.6.2 Check data link and network layer

When troubleshooting end-to-end connectivity, it is useful to verify mappings between destination IP addresses and Layer 2 Ethernet addresses on individual segments. In IPv4, this functionality is provided by ARP. In IPv6, the ARP functionality is replaced by the neighbor discovery process and ICMPv6. The neighbor table caches IPv6 addresses and their resolved MAC addresses.

The `arp` Windows command displays and modifies entries in the ARP cache that are used to store IPv4 addresses and their resolved Ethernet physical (MAC) addresses. The cache can be cleared by using the `arp -d` Windows command if the network administrator wants to repopulate the cache with updated information.

The `netsh interface ipv6 show neighbor` Windows command lists all devices that are currently in the neighbor table. The neighbor table for Linux and MAC OS X can be displayed using `ip neigh show` command.

Use the `show mac address-table` command to display the MAC address table on the switch. Use `show vlan` command can be used to validate VLAN assignments on a switch.

### 24.6.3 Verify Default Gateway

Use `show ip route` Cisco IOS command and the `route print` Windows command to verify the presence of the IPv4 default gateway. If the IPv4 addressing information was obtained automatically from a DHCPv4 server, then the configuration on the DHCP server must be examined.

In IPv6, the default gateway can be configured manually, using stateless autoconfiguration (SLAAC), or by using DHCPv6. Use the `show ipv6 route` Cisco IOS command to check for the IPv6 default route and use the `ipconfig` Windows command to verify if a PC has an IPv6 default gateway.

### 24.6.4 Verify correct route

Use the `show ip route` and `show ipv6 route` command on a router to examine the routing table.

### 24.6.5 Verify transport layer

Two of the most common issues that affect transport layer connectivity include ACL configurations and NAT configurations. A common tool for testing transport layer functionality is the Telnet utility.

**Note!** While Telnet can be used to test the transport layer, for security reasons, SSH should be used to remotely manage and configure devices.

Although the Telnet server application runs on port number 23, a different port number can be specified on the client to connect to any TCP port that must be tested. This indicates whether the connection is accepted (as indicated by the word Open in the output), refused, or times out. Given an example, the administrator Telnets from PC1 to the server HQ using port 80 (Figure 24.4). The output verifies that the server is not accepting connections on port 80.

### 24.6.6 Verify DNS configuration

Use the running configuration file to view the DNS configuration information on the switch or router. When there is no DNS server installed, it is possible to enter names to IP mappings directly into the switch or router configuration. Use the `ip host` command to enter name to IPv4 mapping to the switch or router. The `ipv6 host` command is used for the same mappings using IPv6. To display the name-to-IP-address mapping information on the Windows-based PC, use the `nslookup` command.

Figure 24.4: Use Telnet to verify transport layer

```
C:\> telnet 2001:DB8:172:16::100 80
HTTP/1.1 400 Bad Request
Date: Wed, 26 Sep 2012 07:27:10 GMT
Server: cisco-IOS
Accept-Ranges: none
400 Bad Request
Connection to host lost.
```

**Listing 25:** Assign a name to a router or switch

```
ip host KEVIN-SERVER 172.16.1.100
ipv6 host HUY-SERVER 2001:db8:acad:4::100

ping KEVIN-SERVER
ping HUY-SERVER
```

Because domain names and DNS are a vital component of accessing servers on the network, many times the user thinks the “network is down” when the problem is actually with the DNS server.

## 24.7 Scenarios

**Example 24.1.** A network engineer is troubleshooting a network problem where users cannot access the FTP server at the same IP address where a website can be successfully accessed. Which troubleshooting method would be the best to apply in this case?

*Proof.* The fact that some application layer services provided by a network device are operating successfully but others are not means that the lower OSI or TCP/IP layers are functional with the problem likely to be in the application layer. Therefore, the troubleshooting method is *bottom-up*. □

**Example 24.2.** A network engineer is troubleshooting a network that has recently been updated with a new routing protocol, but the network is not working as expected. The engineer is comparing the running configuration from before and after the change was made. Which approach to troubleshooting the problem is the engineer using?

*Proof.* This is a variation on the divide-and-conquer method. Since a routing protocol change was recently made, the administrator can be fairly certain the issue resides with the network layer. □

**Example 24.3.** A company is setting up a web site with SSL technology to protect the authentication credentials required to access the web site. A network engineer needs to verify that the setup is correct and that the authentication is indeed encrypted. Which tool should be used?

*Proof.* To verify that the authentication is indeed encrypted, the authentication process needs to be captured and investigated, which can be accomplished through a protocol analyzer, such as Wireshark. □

**Example 24.4.** A user in a large office calls technical support to complain that a PC has suddenly lost connectivity to the network. The technician asks the caller to talk to nearby users to see if other machines are affected. The caller reports that several immediate neighbors in the same department have a similar problem and that they cannot ping each other. Those who are seated in other departments have connectivity. What should the technician check as the first step in troubleshooting the issue?

*Proof.* The status of the departmental workgroup switch in the wiring closet □

**Example 24.5.** A user reports that after an OS patch of the networking subsystem has been applied to a workstation, it performs very slowly when connecting to network resources. A network technician tests the link with a cable analyzer and notices that the workstation sends an excessive number of frames smaller than 64 bytes and also other meaningless frames. What is the possible cause of the problem?

*Proof.* The symptom of excessive runt packets and jabber is typically a Layer 1 issue, such as caused by a corrupted NIC driver, which could be the result of a software error during the NIC driver upgrade process. Note that A NIC driver is part of the operating system, it is not an application.  $\square$

**Example 24.6.** An internal corporate server can be accessed by internal PCs, but not by external Internet users that should have access. What could be the issue?

*Proof.* NAT/PAT allows a private IP address to be translated into a public address so that external users can access internal devices. Static NAT assigns one public address to a private address and is used with internal servers.  $\square$

# Chapter 25

## CDP, LLDP, NTP, and Syslog

### 25.1 CDP

Cisco Discovery Protocol (CDP) is a Cisco proprietary Layer 2 protocol that gathers information about Cisco devices sharing the same data link. CDP can also be used as a network discovery tool to determine the information about the neighboring devices.

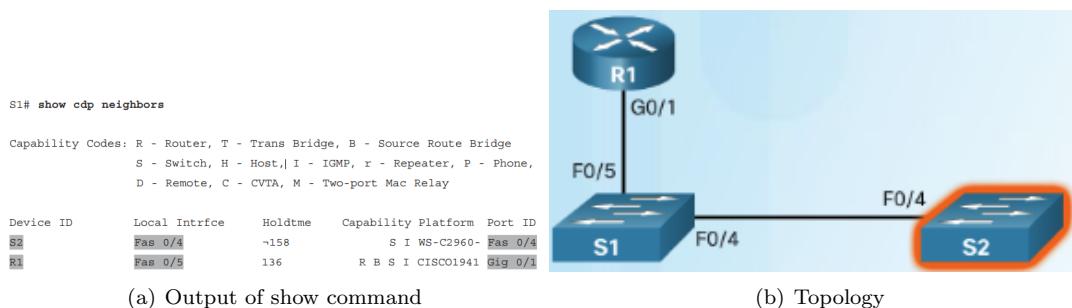
CDP is disabled globally for all interfaces using the `no cdp run` command. To enable CDP globally for all the supported interfaces on the device, enter `cdp run` in the global configuration mode. The `show cdp` command verifies the status and displays information about CDP.

To disable CDP on specific interfaces before enable CDP. For example, Branch-Edge has two active interface `s0/0/1` and `g0/0`. To turn on CDP, first disable CDP on the `s0/0/1` interface and then turn on the CDP protocol.

```
Branch-Edge# configure terminal
Branch-Edge(config)# interface s0/0/1
Branch-Edge(config-if)# no cdp enable
Branch-Edge(config-if)# exit
Branch-Edge(config)# cdp run
```

Use `show cdp` command to check if CDP is running. Use `show cdp interface` to display the interfaces that are CDP enabled on a device.

Figure 25.1: Discovering Device Connected to S1



With CDP enabled on the network, the `show cdp neighbors` and `show cdp neighbors detail` command can be used to determine the network layout (Figure 25.1).

## 25.2 LLDP

LLDP (Link Layer Discovery Protocol) is a vendor-neutral neighbor discovery protocol similar to CDP. To enable LLDP globally on a Cisco network device, enter the `lldp run` global configuration command. To disable LLDP, enter the `no lldp run` command in the global configuration mode. Similar to CDP, LLDP can be configured on specific interfaces. However, LLDP must be configured separately to transmit and receive LLDP packets:

```
S1(config)# lldp run
S1(config)#
S1(config)# interface gigabitethernet 0/1
S1(config-if)# lldp transmit
S1(config-if)# lldp receive
S1(config-if)# end
S1#
S1# show lldp
S1# show lldp neighbors
S1# show lldp neighbors detail
```

## 25.3 NTP

### 25.3.1 System clock

The software clock on a router or switch starts when the system boots and is the primary source of time for the system. It is important to synchronize the time across all devices on the network because all aspects of managing, securing, troubleshooting, and planning networks require accurate time-stamping. Typically, the date and time settings on a router or switch can be set using one of two methods:

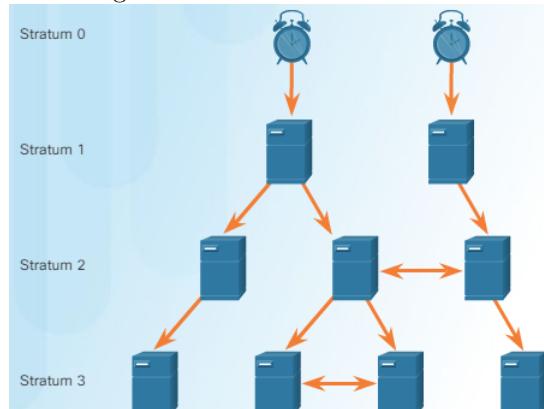
- Manually configure the date and time. For example, `clock set 20:36:00 aug 30 2016`
- Configure the NTP

### 25.3.2 NTP operation

NTP allows routers on the network to synchronize their time settings with an NTP server. It uses **UDP port 123**.

NTP networks use a hierarchical system of time sources. Each level in this hierarchical system is called a **stratum**. The stratum level is defined as the number of hop counts from the authoritative time source (Figure 25.2). Smaller stratum numbers indicate that the server is closer to the authorized time source than larger stratum numbers. The max hop count is 15. Stratum 16, the lowest stratum level, indicates that a device is unsynchronized.

Figure 25.2: NTP Stratum levels



**Stratum 0** An NTP network gets the time from authoritative time sources. Stratum 0 devices are represented by the clock in the figure 25.2.

**Stratum 1** are directly connected to the authoritative time sources. They act as the primary network time standard.

**Stratum 2 and Lower** The stratum 2 servers are connected to stratum 1 devices through network connections. Stratum 2 devices, such as NTP clients, synchronize their time using the NTP packets from stratum 1 servers. They can also act as servers for stratum 3 devices.

### 25.3.3 Configure and Verify NTP

To verify the system clock, use `show clock [detail]` command. To identify NTP server, use `ntp server <ip-addr>`. To see if the device is synchronized with the NTP server, use `show ip ntp associations` and `show ntp status`.

## 25.4 Syslog

### 25.4.1 Introduction

The syslog protocol allows networking devices to send their system messages across the network to syslog servers. The syslog server serves as an event message collector. Syslog messages are sent using **UDP port 514**. The syslog logging service provides three primary functions:

- Gather logging information
- Select the type of logging information
- Specify the destination of captured syslog messages

Syslog messages may be sent to an internal buffer. Messages sent to the internal buffer (RAM) are only viewable through the CLI of the device (console line, terminal line). Alternatively, syslog messages may be sent across the network to an external syslog server.

To view syslog messages, a syslog server must be installed on a workstation. One advantage of viewing syslog messages on a syslog server is the ability to perform granular searches through the data. Also, a network administrator can quickly delete unimportant syslog messages from the database.

### 25.4.2 Severity level and Facility

Cisco devices produce syslog messages as a result of network events. Every syslog message contains a **severity level** and a **facility**. The security level can be shown as a number. The smaller the number, the more critical syslog alarms (Table 25.1).

Level 0 – 4 are error messages. Level 5 notifies system messages such as interface up or down transitions and system restart messages. Level 6 generates messages , for example, when the device is booting. By default, Cisco routers and switches send log messages up to level 6 of severity (levels 0 through 6) to the console.

The facility represents the machine process that created the syslog event. For example, is the event created by the kernel, by the mail system, by security/authorization processes, etc.? The Facility value is a way of determining which process of the machine created the message.

### 25.4.3 Message format

By default, the format of syslog messages on the Cisco IOS Software is as follows:

```
seq no: timestamp: %facility-severity-MNEMONIC: description  
00:00:46: %LINK-3-UPDOWN: Interface Port-channel11, changed state to up
```

The fields contained in the syslog message above are explained in Table 25.2.

Table 25.1: Syslog Severity level

<b>Severity level</b>	<b>Name</b>	<b>Explanation</b>
0	Emergency	A "panic" condition, System unusable
1	Alert	Should be corrected immediately, e.g. loss of backup ISP connection
2	Critical	Critical condition
3	Error	Error condition, Non-urgent failures
4	Warning	NOT an error, but indication that an error will occur if action is not taken, e.g. file system 85% full
5	Notification	Normal but significant condition
6	Informational	Not affect functionality, harvested for reporting, measuring throughput,
7	Debugging	Debugging message

Table 25.2: Syslog message format

<b>Field</b>	<b>Example</b>	<b>Explanation</b>
seq no	—	Will be shown only if the service <code>sequence-numbers</code> is configured
timestamp	00:00:46	Date and time of the message, which appears only if the service <code>timestamps</code> is configured
facility	LINK	The facility to which the message refers
severity	3	A number from 0 to 7 that indicates the severity of the message
MNEMONIC	UPDOWN	Briefly and Uniquely describe the message
description	Interface ...	Report the event in detail

#### 25.4.4 Configuration

By default, log messages do not include a timestamp. To enable timestamp, use the following command

```
R1(config)# service timestamps log datetime
```

The `show logging` command displays the default logging service settings.

There are three steps to configuring the router to send system messages to a syslog server:

1. Use `logging <ip-addr>` to configure IP address of the syslog server
2. Use the `logging trap <level>` to configure severity level. For example, to limit the messages to levels 4 and lower (0 to 4), use the `logging trap 4` command.
3. (Optional) Configure the source interface with the `logging source <int>` command. This specifies that syslog packets contain the IP address of a specific interface, regardless of which interface the packet uses to exit the router.

```
R1(config)# logging 192.168.1.3
R1(config)# logging trap 4
R1(config)# logging source-interface GigabitEthernet 0/0
R1(config)# interface loopback 0
```

In this example, R1 is configured to send log messages of levels 4 and lower to the syslog server at 192.168.1.3. The source interface is set as the G0/0 interface. A loopback interface is created, shut down, and then brought back up. The console output reflects these actions.



# Chapter 26

## Device maintenance

### 26.1 File system

The `show file systems` command (Figure 26.1) lists all of the available file systems. This command provides useful information such as the amount of available and free memory, the type of file system, and its permissions. Permissions include read only (ro), write only (wo), and read and write (rw), shown in the Flags column of the command output.

Figure 26.1: The output of 'show filesystems' command

File Systems:				
Size(b)	Free(b)	Type	Flags	Prefixes
-	-	opaque	rw	archive:
-	-	opaque	rw	system:
-	-	opaque	rw	tmsys:
-	-	opaque	rw	null:
-	-	network	rw	tftp:
*	256487424	183234560	disk	rw flash0: flash:#
-	-	disk	rw	flash1:
262136	254779	nvram	rw	nvram:
-	-	opaque	wo	syslog:
-	-	opaque	rw	xmodem:
-	-	opaque	rw	ymodem:
-	-	network	rw	rcp:
-	-	network	rw	http:
-	-	network	rw	ftp:
-	-	network	rw	scp:
-	-	opaque	ro	tar:
-	-	network	rw	https:
-	-	opaque	ro	cns:

Notice that the flash file system also has an asterisk preceding it. This indicates that flash is the current default file system. The bootable IOS is located in flash; therefore, the pound symbol (#) is appended to the flash listing.

Use `dir <dir-name>` to view the content of the specified directory. You can change the current directory using `cd <dir>` command. The `pwd` (present working directory) command shows you the name of the current directory.

In the flash file system, the file with .bin extension (usually the last listing) is the IOS file image that is running in RAM.

### 26.2 Back up and Restore

#### 26.2.1 Using text file

The steps to save a configuration using Tera Term follow:

1. On the File menu, click Log.
2. Choose the location to save the file. Tera Term begins capturing text.
3. After capture has been started, execute the `show run` or `show startup` command. Text displayed in the terminal window is directed to the chosen file.
4. When the capture is complete, select Close in the Tera Term: Log window.
5. View the file to verify that it was not corrupted.

We can restore device configuration by copying the content of a text file to the terminal. Note that the text file requires editing to ensure that encrypted passwords are in plain text. Further, at the CLI, the device must be set at the global configuration mode to receive the commands from the text file being pasted into the terminal window.

When using Tera Term, follow these steps:

1. On the File menu, click Send file.
2. Locate the file to be copied into the device and click Open.
3. Tera Term pastes the file into the device.

### 26.2.2 Using TFTP

You can use a service like Trivial File Transfer Protocol (TFTP) to remotely back up and restore files. Follow these steps to back up the running configuration to a TFTP server:

1. Enter the `copy run tftp` command.
2. Enter the IP address of the host where the configuration file will be stored.
3. Enter the name to assign to the configuration file.
4. Press Enter to confirm each choice.

```
R1# copy running-config tftp
Address or name of remote host []? 192.168.10.254
Destination filename [r1-cfg]? R1-Jan-2016
Write file R1-Jan-2016 to 192.168.10.254? [confirm]
Writing R1-Jan-2016 !!!!! [OK]
```

Use these steps to restore the running configuration from a TFTP server:

1. Enter the `copy tftp run` command.
2. Enter the IP address of the TFTP server where the configuration file is stored.
3. Enter the name of the configuration file. For example, in the above example the file name was R1-Jan-2016.
4. Press Enter to confirm.

### 26.2.3 Using USB

Images, configurations, and other files can be copied to or from the Cisco USB flash memory with the same reliability as storing and retrieving files using the Compact Flash card. In addition, modular integrated services routers can boot any Cisco IOS Software image saved on USB flash memory.

```
R1# dir usbflash0:
Directory of usbflash0:/
1 -rw- 30125020 Dec 22 2032 05:31:32 +00:00 c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
```

When backing up to a USB port, it is a good idea to issue the `show file systems` command to verify that the USB drive is there and confirm the name (Figure 26.2). In this case, the USB has a name of `usbflash0`.

Use `copy run usbflash0:/` command to copy the configuration file to the USB. To restore configuration from a USB, it is necessary to edit the file in USB with a text editor. Assuming the file name is `R1-Config`, use the command `copy usbflash0:/R1-Config run` to restore a running configuration.

Figure 26.2: Verifying the USB drive is available

```
R1# show file systems
File Systems:

      Size(b)    Free(b)   Type   Flags  Prefixes
      -          -  opaque  rw    archive:
      -          -  opaque  rw    system:
      -          -  opaque  rw    tmpsys:
      -          -  opaque  rw    null:
      -          -  network rw    tftp:
*  256487424  184819712  disk   rw    flash0: flash:# 
      -          -  disk   rw    flash1:
262136     249270   nvram  rw    nvram:
      -          -  opaque  wo    syslog:
      -          -  opaque  rw    xmodem:
      -          -  opaque  rw    ymodem:
      -          -  network rw    rcp:
      -          -  network rw    http:
      -          -  network rw    ftp:
      -          -  network rw    scp:
      -          -  opaque  ro    tar:
      -          -  network rw    https:
      -          -  opaque  ro    cns:
4050042880  3774152704  usbflash rw    usbflash0:
R1#
```

#### 26.2.4 Password recovery

Console access to the device using terminal emulator software on a PC is required for password recovery. With console access, a user can access the ROMMON mode by using a *break sequence* during the bootup process or removing the external flash memory when the device is powered off.

Password recovery procedures on all devices follow the same principle:

1. Enter ROMMON mode
2. Change the configuration register to `0x2142` (use `confreg 0x2142` command) and reboot the device (use `reset` command)

```
 Readonly ROMMON initialized
monitor: command "boot" aborted due to user interrupt
```

```
rommon 1 > confreg 0x2142
rommon 2 > reset
```

```
System Bootstrap, Version 15.0(1r)M9, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2010 by cisco Systems, Inc.
<output omitted>
```

3. After the device has finished rebooting, copy the startup configuration file to the running configuration file, use `copy startup run` command. DO NOT enter `copy run startup` because this command erases your original startup configuration.
4. Because you are in privileged EXEC mode, you can now configure all the necessary passwords.
5. After the new passwords are configured, change the configuration register back to `0x2102` using the the following command:

- ```
Router(config)# config-register 0x2102
```
6. Save the new configuration using `copy run startup` command
  7. Reload the device

**ROMMON mode** enables an administrator to access flash memory, format the flash file system, reinstall the IOS, recover a lost password or change the configuration register. The configuration register instructs the router how to boot up. For example, when the configuration register is `0x2142`, the device ignores the startup config file during startup. The default configuration register is `0x2102`.

## 26.3 IOS image

### 26.3.1 Naming convention

Cisco Integrated Services Routers Generation Two (ISR G2) supports *Services on Demand* through the use of software licensing. Each router is shipped with the same *universal image*. The IOS features are enabled in the universal image via licensing keys.

There are two types of universal images supported in ISR G2:

- **With universalk9 in the name:** offer all Cisco IOS features
- **With universalk9\_npe in the name:** Not support strong cryptography such as payload cryptography.

The Cisco IOS image file has special naming convention. The `show flash` command displays all files stored in flash memory including IOS image(see table 26.1 for explanation):

`c1900-universalk9-mz.SPA.152-4.M3.bin`

Table 26.1: Cisco IOS image naming convention

| Part              | Example                  | Explanation                                                                                                                                              |
|-------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Image name        | <code>c1900</code>       | Identify the platform. In this example, the platform is Cisco 1900 router.                                                                               |
| Image designation | <code>universalk9</code> | The two designations for an ISR G2 are <code>universalk9</code> and <code>universalk9_npe</code> .                                                       |
| Status            | <code>mz</code>          | Indicates where the image runs and if it is compressed. In this example, the image is running in RAM and is compressed.                                  |
| Signature         | <code>SPA</code>         | Digitally signed by Cisco                                                                                                                                |
| Version           | <code>152-4.M3</code>    | Major release (15), minor release (2), maintenance release (4), maintenance rebuild number (3). The M indicates this is an extended maintenance release. |
| File extension    | <code>bin</code>         | This extension indicates that this file is a binary executable file.                                                                                     |

The most common designation for memory location and compression format is `mz`. The first letter indicates the location where the image is executed on the router. The locations can include: `f` (flash), `m` (RAM), `r` (ROM), and `l` (relocatable). The compression format can be either `z` for zip or `x` for mzip.

### 26.3.2 IOS image and TFTP server

To create a backup of the Cisco IOS image to a TFTP server, perform the following three steps:

1. Ping TFTP server to test connectivity
2. Use `show flash0:` command to determine the size of image file

3. Verify that TFTP server has sufficient disk space to accommodate the image file
4. Copy the image to TFTP server using `copy flash0: tftp:`
5. Answer the prompted questions

We can upgrade software in Cisco router by copying IOS image from TFTP server to the device:

1. Ping TFTP server to test connectivity
2. Use `show flash0:` command to determine the size of image file
3. Verify that TFTP server has sufficient disk space to accommodate the image file
4. Copy the image to TFTP server using `copy tftp: flash0:`
5. Answer the prompted questions

### 26.3.3 The boot system command

To upgrade to the copied IOS image after that image is saved on the router's flash memory, configure the router to load the new image during bootup using the `boot system` command

```
R1(config)# boot system flash0://c1900-universalk9-mz.SPA.152-4.M3.bin
```

After the router has booted, to verify the new image has loaded, use the `show version` command.

Several boot system commands can be entered in sequence to provide a fault-tolerant boot plan. If there are no boot system commands in the configuration, the router defaults to loading the first valid Cisco IOS image in flash memory and running it.

## 26.4 Software licensing

### 26.4.1 Licensing key

Recall that each device ships with the same universal image. Technology packages are enabled in the universal image via Cisco IOS Software Activation licensing keys.

Each licensing key is unique to a particular device and is obtained from product ID, serial number of the router and a *Product Activation Key (PAK)*. Cisco provides PAK at the time of purchase.

The IP Base is installed by default. Other technology Data, UC (Unified Communication), and SEC (Security) are enabled using licensing keys.

Use the `show license feature` command to view the technology package licenses and feature licenses supported on the router.

### 26.4.2 Licensing process

A permanent license is a license that never expires. Evaluation license, known as a temporary license, allows customers to try a new software package. If customers want to permanently activate a software package or feature on the router, they must get a new software license.

There are three steps to permanently activate a new feature on the router:

1. Purchase the feature to install. Software Claim Certificates are used for licenses that require software activation. The claim certificate provides the PAK and Cisco EULA. PAK , an 11-digit alphanumeric key, serves as a receipt to obtain a license. A PAK is not tied to a specific device until the license is created.

2. Obtain license file using either CLM<sup>1</sup> or Cisco License Registration Portal<sup>2</sup>. Then the license information is sent via email to install the license file. The license file is an XML text file with a .lic extension.
3. Use `license install <store-location>` to install license file

```
R1# license install flash0:seck9-C1900-SPE150_K9-FHH12250057.xml
```

4. Reboot the router
5. Verify license using `show license` and `show version` command.

Both of these processes require a PAK number and a unique device identifier (UDI). UDI is the combination of Product ID (PID), Serial number (SN) and hardware version. The SN is an 11-digit number that uniquely identifies a device. The PID identifies the type of device. The UDI can be displayed using `show license udi`.

#### 26.4.3 Activate an Evaluation license

Evaluation licenses are replaced with RTU<sup>3</sup> after **60 days** (i.e. expired). These licenses are available on the honor system and require customer's acceptance of EULA. The `license accept end user agreement` in global configuration mode is used to configure a one-time EULA acceptance for all IOS features.

Suppose you want to activate Data package on a Cisco 1900 router, the following command should be executed:

```
R1(config)# license boot module c1900 technology-package datak9
```

Technology package names are `ipbasek9`, `securityk9`, `datak9`, and `uck9`.

Remember that the device has to reboot to activate new software package.

#### 26.4.4 Back up the license

To copy all licenses in a device and store them in flash memory with the file name `all_licenses.lic`, execute the following command:

```
R1# license save flash0:all_licenses.lic
```

#### 26.4.5 Uninstall a permanent license

Perform the following steps to completely uninstall an active permanent license:

1. Disable technology package, then reboot

```
R1(config)# license boot module c1900 technology-package seck9 disable
R1(config)# exit
R1# reload
```

2. Clear technology package license

```
R1# license clear seck9
```

3. Remove the `license boot module` command in step 1, and then reload.

```
R1(config)# no license boot module c1900 technology-package seck9 disable
R1(config)# exit
R1# reload
```

---

<sup>1</sup>Cisco License Manager, a free software application deploying Cisco licenses across networks

<sup>2</sup>web-based for getting and registering individual software licenses

<sup>3</sup>Evaluation Right-to-Use licenses