



| YOUR DEFENCE PARTNER OF CHOICE

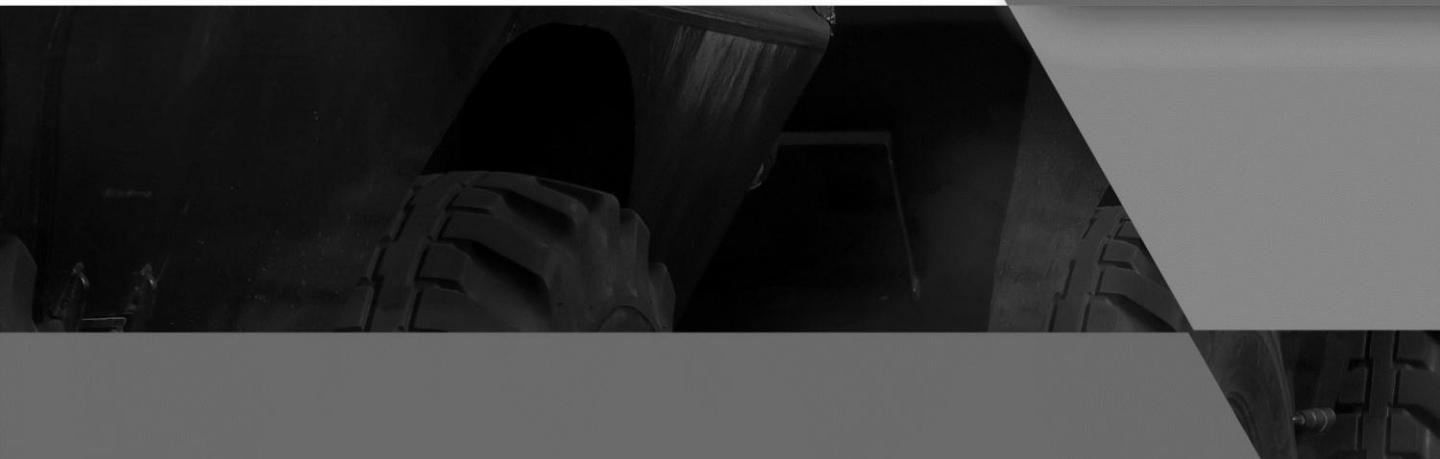
Cyber Shield Academy



CYBER SHIELD
ACADEMY

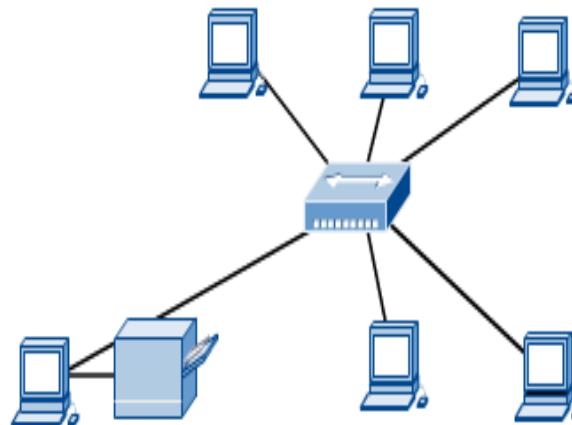
Computer Networking

version one



Computer Networks

- Computer network connects two or more autonomous computers.
- The computers can be geographically located anywhere.



LAN, MAN & WAN



- Network in small geographical Area (Room, Building or a Campus) is called LAN (Local Area Network)
- Network in a City is call MAN (Metropolitan Area Network)
- Network spread geographically (Country or across Globe) is called WAN (Wide Area Network)

Applications of Networks



■ Resource Sharing

- Hardware (computing resources, disks, printers)
- Software (application software)

■ Information Sharing

- Easy accessibility from anywhere (files, databases)
- Search Capability (WWW)

■ Communication

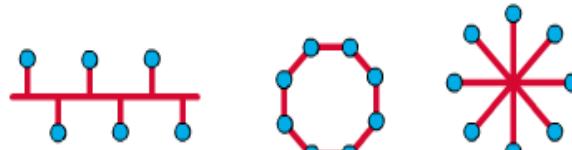
- Email
- Message broadcast

■ Remote computing

■ Distributed processing

Network Topology

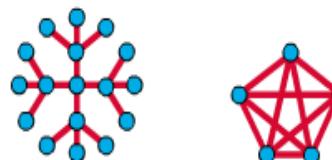
The network topology defines the way in which computers, printers, and other devices are connected. A network topology describes the layout of the wire and devices as well as the paths used by data transmissions.



Bus Topology

Ring Topology

Star Topology

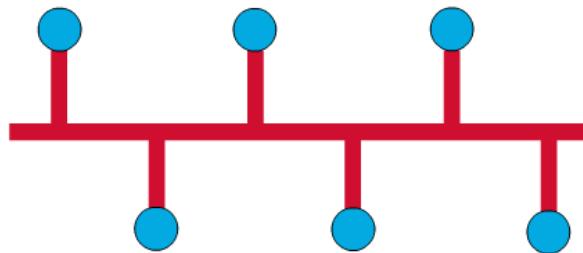


Extended Star Topology

Mesh Topology

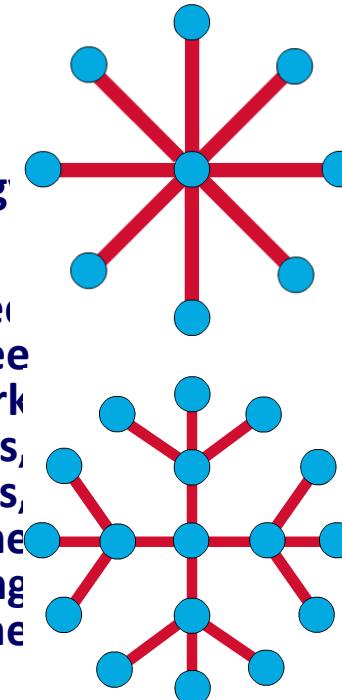
Bus Topology

- Commonly referred to as a linear bus, all the devices on a bus topology are connected by one single cable.



Star & Tree Topology

- The star topology is the most commonly used architecture in Ethernet LANs.
- When installed, the star topology resembles spokes in a bicycle wheel.
- Larger networks use the extended star topology also called tree topology. When used with network devices that filter frames or packets, like bridges, switches, and routers, this topology significantly reduces the traffic on the wires by sending packets only to the wires of the destination host.



Protocols and Models



CYBER SHIELD
ACADEMY



100% zoomed correctly to display

The Benefits of Using a Layered Model



These are the benefits of using a layered model:

- Assist in protocol design because protocols that operate at a specific layer have defined information that they act upon and a defined interface to the layers above and below
- Foster competition because products from different vendors can work together
- Prevent technology or capability changes in one layer from affecting other layers above and below
- Provide a common language to describe networking functions and capabilities

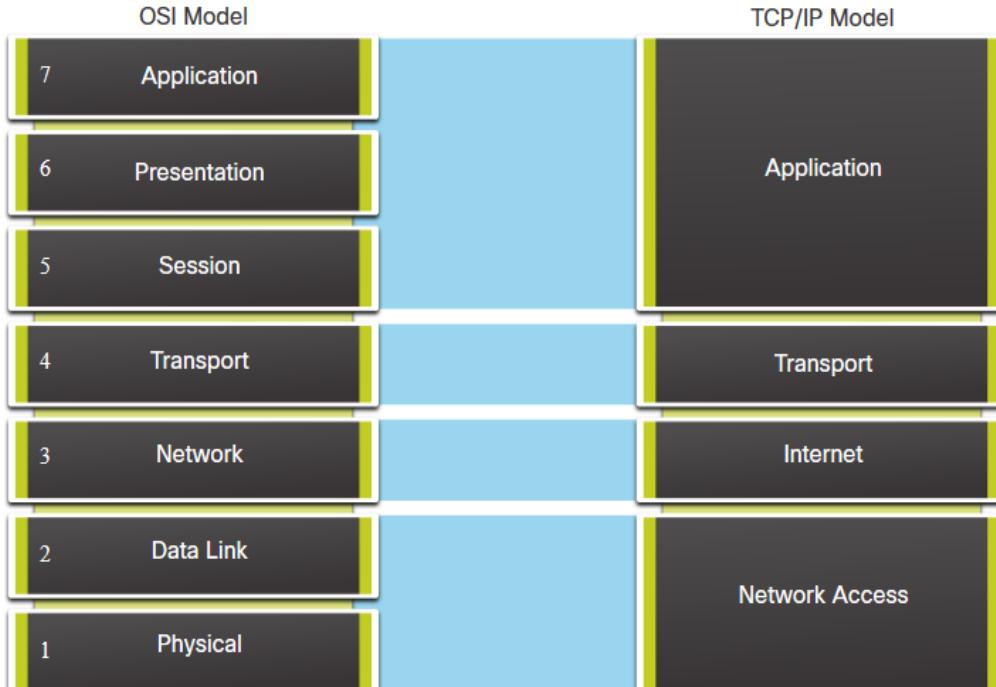
The OSI Reference Model

OSI Model Layer	Description
7 - Application	Contains protocols used for process-to-process communications.
6 - Presentation	Provides for common representation of the data transferred between application layer services.
5 - Session	Provides services to the presentation layer and to manage data exchange.
4 - Transport	Defines services to segment, transfer, and reassemble the data for individual communications.
3 - Network	Provides services to exchange the individual pieces of data over the network.
2 - Data Link	Describes methods for exchanging data frames over a common media.
	Describes the means to activate, maintain, and de-activate

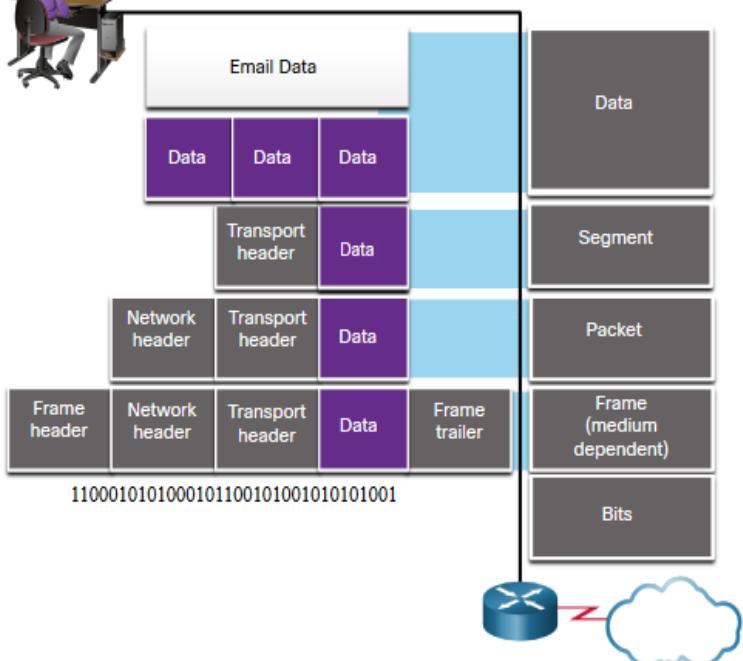
The TCP/IP Reference Model

TCP/IP Model Layer	Description
Application	Represents data to the user, plus encoding and dialog control.
Transport	Supports communication between various devices across diverse networks.
Internet	Determines the best path through the network.
Network Access	Controls the hardware devices and media that make up the network.

OSI and TCP/IP Model Comparison



- The OSI model divides the network access layer and the application layer of the TCP/IP model into multiple layers.
- The TCP/IP protocol suite does not specify which protocols to use when transmitting over a physical medium.
- OSI Layers 1 and 2 discuss the necessary procedures to access the media and the physical means to send data over a network.



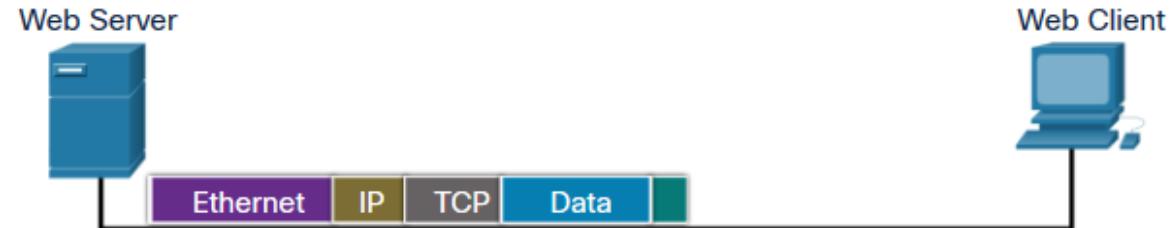
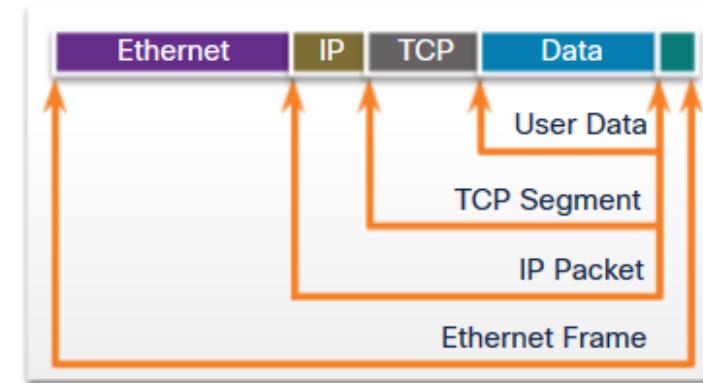
Encapsulation is the process where protocols add their information to the data.

- At each stage of the process, a PDU has a different name to reflect its new functions.
- There is no universal naming convention for PDUs, in this course, the PDUs are named according to the protocols of the TCP/IP suite.
- PDUs passing down the stack are as follows:
 1. Data (Data Stream)
 2. Segment
 3. Packet
 4. Frame
 5. Bits (Bit Stream)

Encapsulation Example



- Encapsulation is a top down process.
- The level above does its process and then passes it down to the next level of the model. This process is repeated by each layer until it is sent out as a bit stream.



De-encapsulation Exam

- Data is de-encapsulated as it moves up the stack.
- When a layer completes its process, that layer strips off its header and passes it up to the next level to be processed. This is repeated at each layer until it is a data stream that the application can process.

1. Received as Bits (Bit Stream)
2. Frame
3. Packet
4. Segment
5. Data (Data Stream)



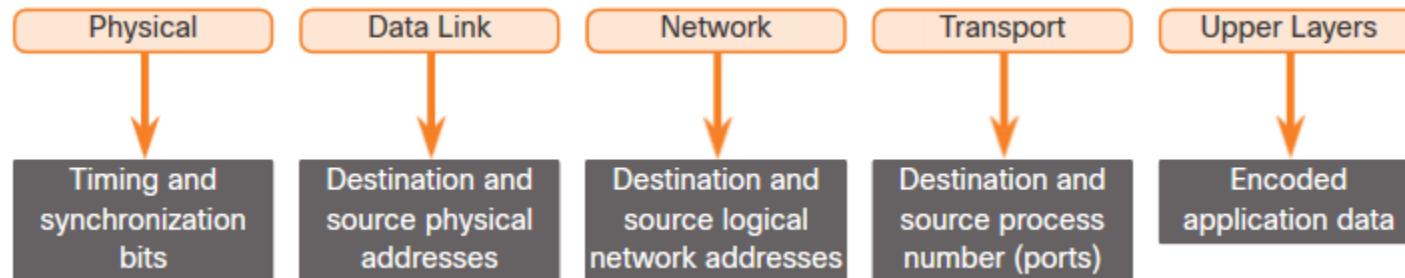
Addresses



Both the data link and network layers use addressing to deliver data from source to destination.

Network layer source and destination addresses - Responsible for delivering the IP packet from original source to the final destination.

Data link layer source and destination addresses – Responsible for delivering the data link frame from one network interface card (NIC) to another NIC on the same network.



Layer 3 Logical Address

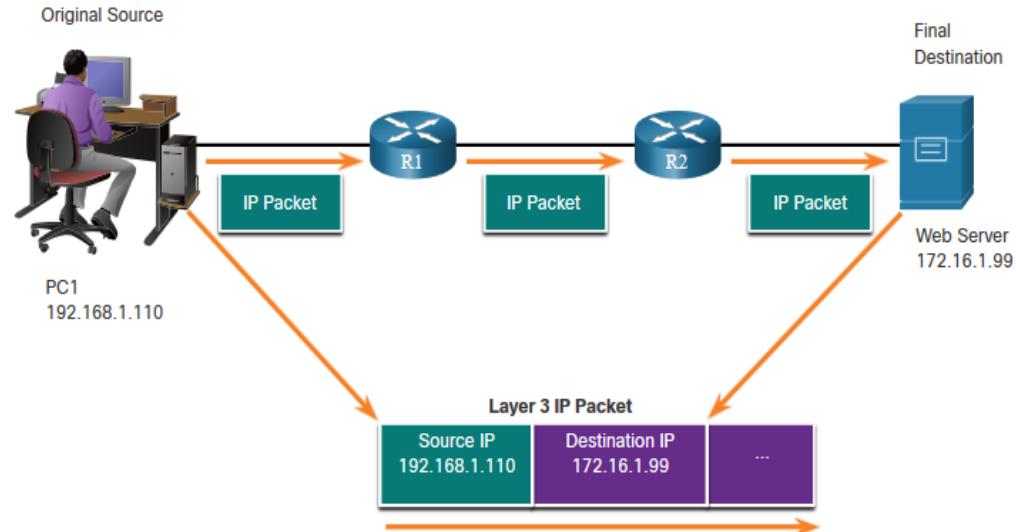


The IP packet contains two IP addresses:

Source IP address - The IP address of the sending device, original source of the packet.

Destination IP address - The IP address of the receiving device, final destination of the packet.

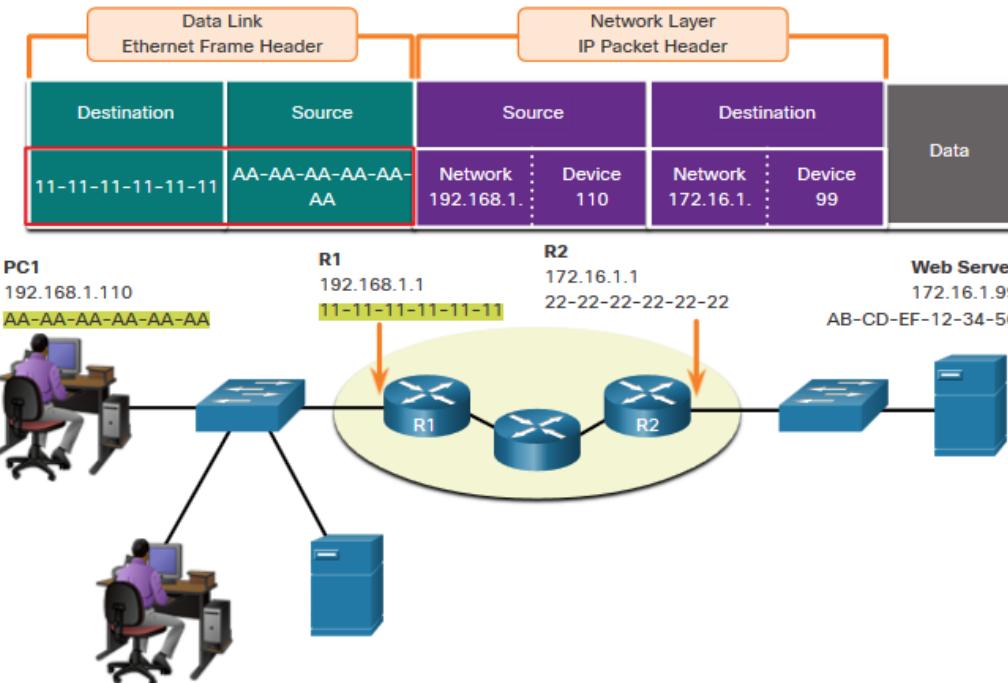
These addresses may be on the same link or remote.



Role of the Data Link Layer Addresses: Different IP Networks

When the final destination is remote, Layer 3 will provide Layer 2 with the local default gateway IP address, also known as the router address.

- The default gateway (DGW) is the router interface IP address that is part of this LAN and will be the “door” or “gateway” to all other remote locations.
- All devices on the LAN must be told about this address or their traffic will be confined to the LAN only.
- Once Layer 2 on PC1 forwards to the default gateway (Router), the router then can start the routing process of getting the information to actual destination.



Data Access

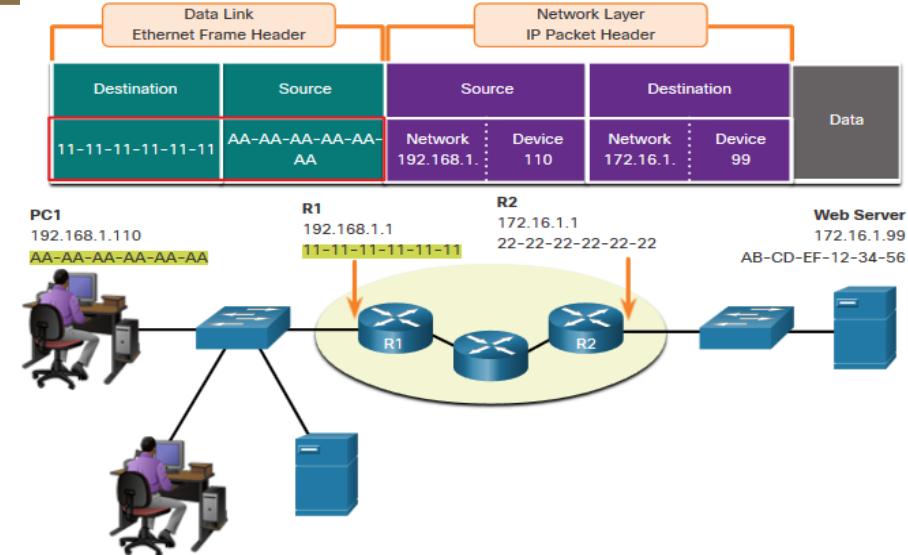
Role of the Data Link Layer Addresses: Different IP Networks



- The data link addressing is local addressing so it will have a source and destination for each link.

- The MAC addressing for the first segment is :
Source – AA-AA-AA-AA-AA-AA (PC1) Sends the frame.
Destination – 11-11-11-11-11-11 (R1- Default Gateway MAC) Receives the frame.

Note: While the L2 local addressing will change from link to link or hop to hop, the L3 addressing remains the same.

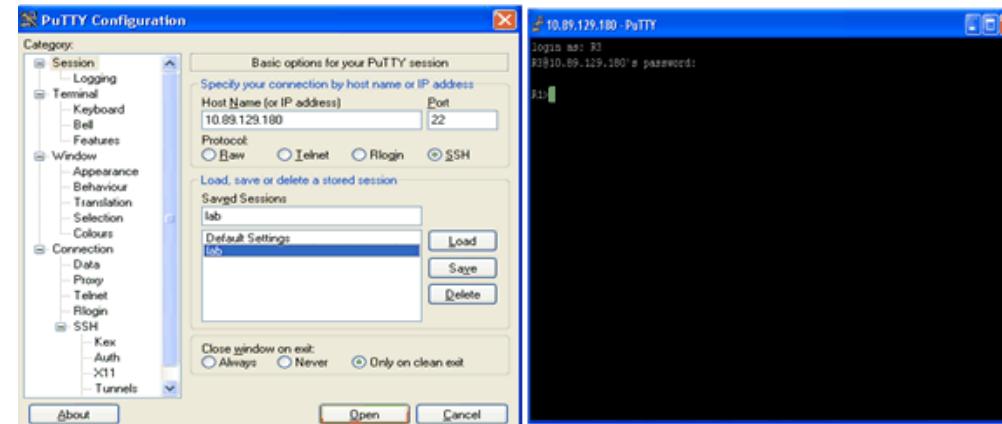


Basic Switch and End Device Configuration

Access Methods



- **Console** – A physical management port used to access a device in order to provide maintenance, such as performing the initial configurations.
- **Secure Shell (SSH)** – Establishes a secure remote CLI connection to a device, through a virtual interface, over a network. (Note: This is the recommended method for remotely connecting to a device.)
- **Telnet** – Establishes an insecure remote CLI connection to a device over the network. (Note: User authentication, passwords and commands are sent over the network in plaintext.)



Primary Command Mode



User EXEC Mode:

Allows access to only a limited number of basic monitoring commands

Identified by the CLI prompt that ends with the > symbol

```
Router>
```

```
Switch>
```

Privileged EXEC Mode:

- Allows access to all commands and features
- Identified by the CLI prompt that ends with the # symbol

```
Router#
```

```
Switch#
```

Configuration Mode and Subconfiguration Modes



Global Configuration Mode:

Used to access configuration options on the device

```
Switch(config)#
```

Line Configuration Mode:

Used to configure console, SSH, Telnet or AUX access

```
Switch(config-line)#
```

Interface Configuration Mode:

Used to configure a switch port or router interface

```
Switch(config-if)#
```

IOS Navigation

Navigation Between IOS Modes

Privileged EXEC Mode:

To move from user EXEC mode to privilege EXEC mode, use the **enable** command.

Global Configuration Mode:

To move in and out of global configuration mode, use the **configure terminal** command.

To return to privilege EXEC mode, use the **exit** command.

Line Configuration Mode:

To move in and out of line configuration mode, use the **line** command followed by the management line type. To return to global configuration mode, use the **exit** command.



```
Switch> enable  
Switch#
```

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

```
Switch(config)#line console 0  
Switch(config-line)#exit  
Switch(config)#
```

Navigation Between IOS Modes (Cont.)



Subconfiguration Modes:

To move out of any subconfiguration mode to get back to global configuration mode, use the **exit** command. To return to privilege EXEC mode, use the **end** command or key combination **Ctrl +Z**.

To move directly from one subconfiguration mode to another, type in the desired subconfiguration mode command. In the example, the command prompt changes from **(config-line)#** to **(config-if)#**.

```
Switch(config)#line console 0  
Switch(config-line)#end  
Switch#
```

```
Switch(config-line)#interface FastEthernet 0/1  
Switch(config-if)#
```

IOS Command Syntax Check (Cont.)



- The command is **ping** and the user-defined argument is the *ip-address* of the destination device. For example, **ping 10.10.10.5**.
- The command is **traceroute** and the user-defined argument is the *ip-address* of the destination device. For example, **traceroute 192.168.254.254**.

```
ping ip-address
```

```
traceroute ip-address
```

Device Names



- The first configuration command on any device should be to give it a unique hostname.
- By default, all devices are assigned a factory default name. For example, a Cisco IOS switch is "Switch."

```
Switch# configure terminal  
Switch(config)# hostname Sw-Floor-1  
Sw-Floor-1(config)#
```

Note: To return the switch to the default prompt, use the **no hostname** global config command.

Configure Passwords

Securing user EXEC mode access:

First enter line console configuration mode using the **line console 0** command in global configuration mode.

Next, specify the user EXEC mode password using the **password password** command.

Finally, enable user EXEC access using the **login** command.

Securing privileged EXEC mode access:

- First enter global configuration mode.
- Next, use the **enable secret password** command.

```
Sw-Floor-1# configure terminal  
Sw-Floor-1(config)# line console 0  
Sw-Floor-1(config-line)# password cisco  
Sw-Floor-1(config-line)# login  
Sw-Floor-1(config-line)# end  
Sw-Floor-1#
```

```
Sw-Floor-1# configure terminal  
Sw-Floor-1(config)# enable secret class  
Sw-Floor-1(config)# exit  
Sw-Floor-1#
```



Configure Passwords (Cont.)



Securing VTY line access:

First enter line VTY configuration mode using the **line vty 0 15** command in global configuration mode.

Next, specify the VTY password using the **password password** command.

Finally, enable VTY access using the **login** command.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# line vty 0 15
Sw-Floor-1(config-line)# password cisco
Sw-Floor-1(config-line)# login
Sw-Floor-1(config-line)# end
Sw-Floor-1#
```

- Note: VTY lines enable remote access using Telnet or SSH to the device. Many Cisco switches support up to 16 VTY lines that are numbered 0 to 15.

Encrypt Passwords

The startup-config and running-config files display most passwords in plaintext.

To encrypt all plaintext passwords, use the **service password-encryption** global config command.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# service password-encryption
Sw-Floor-1(config)# exit
Sw-Floor-1#
```

- Use the **show running-config** command to verify that the passwords on the device are now encrypted.

```
Sw-Floor-1# show running-config
!
!
line con 0
password 7 094F471A1A0A
login
!
Line vty 0 4
Password 7 03095A0F034F38435B49150A1819
Login
!
!
end
```



Banner Messages

A banner message is important to warn unauthorized personnel from attempting to access the device.

To create a banner message of the day on a network device, use the **banner motd #** *the message of the day #* global config command.

Note: The “#” in the command syntax is called the delimiting character. It is entered before and after the message.

```
Sw-Floor-1# configure terminal  
Sw-Floor-1(config)# banner motd #Authorized Access Only!#
```

The banner will be displayed on attempts to access the device.



```
Press RETURN to get started.
```

```
Authorized Access Only!
```

```
User Access Verification
```

```
Password:
```

Configuration Files



There are two system files that store the device configuration:

startup-config - This is the saved configuration file that is stored in NVRAM. It contains all the commands that will be used by the device upon startup or reboot. Flash does not lose its contents when the device is powered off.

running-config - This is stored in Random Access Memory (RAM). It reflects the current configuration. Modifying a running configuration affects the operation of a Cisco device immediately. RAM is volatile memory. It loses all of its content when the device is powered off or restarted.

To save changes made to the running configuration to the startup configuration file, use the **copy running-config startup-config** privileged EXEC mode command.

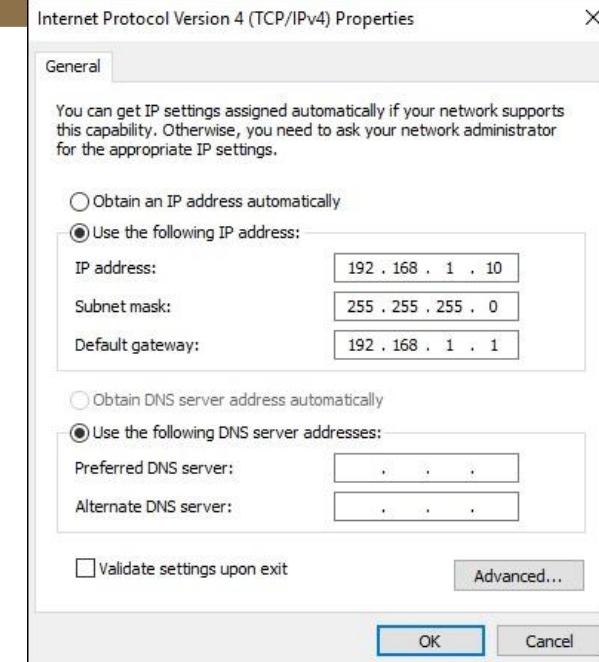
```
Router#show startup-config
Using 624 bytes
!
version 15.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
```

```
Router#show running-config
Building configuration...

Current configuration : 624 bytes
!
version 15.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
```

IP Addresses

- The use of IP addresses is the primary means of enabling devices to locate one another and establish end-to-end communication on the internet.
- The structure of an IPv4 address is called dotted decimal notation and is represented by four decimal numbers between 0 and 255.
- An IPv4 subnet mask is a 32-bit value that differentiates the network portion of the address from the host portion. Coupled with the IPv4 address, the subnet mask determines to which subnet the device is a member.
- The default gateway address is the IP address of the router that the host will use to access remote networks, including the internet.



Configure IP Addressing

Switch Virtual Interface Configuration



To access the switch remotely, an IP address and a subnet mask must be configured on the SVI.

To configure an SVI on a switch:

Enter the **interface vlan 1** command in global configuration mode.

Next assign an IPv4 address using the **ip address ip-address subnet-mask command**.

Finally, enable the virtual interface using the **no shutdown** command.

```
Switch# configure terminal
Switch(config)# interface vlan 1
Switch(config-if)# ip address 192.168.1.20 255.255.255.0
Switch(config-if)# no shutdown
```

Basic Device Configuration

Switch SVI Configuration Example (Cont.)

Task	IOS Commands
Enter global configuration mode.	S1# configure terminal
Enter interface configuration mode for the SVI.	S1(config)# interface vlan 99
Configure the management interface IPv4 address.	S1(config-if)# ip address 172.17.99.11 255.255.255.0
Configure the management interface IPv6 address	S1(config-if)# ipv6 address 2001:db8:acad:99::1/64
Enable the management interface.	S1(config-if)# no shutdown
Return to the privileged EXEC mode.	S1(config-if)# end
Save the running config to the startup config.	S1# copy running-config startup-config

Switch SVI Configuration Example (Cont.)

Step 2: Configure the Default Gateway

- The switch should be configured with a default gateway if it will be managed remotely from networks that are not directly connected.
- Note:** Because, it will receive its default gateway information from a router advertisement (RA) message, the switch does not require an IPv6 default gateway.

Task	IOS Commands
Enter global configuration mode.	S1# configure terminal
Configure the default gateway for the switch.	S1(config)# ip default-gateway 172.17.99.1
Return to the privileged EXEC mode.	S1(config-if)# end
Save the running config to the startup config.	S1# copy running-config startup-config

Configure a Switch with Initial Settings

Switch SVI Configuration Example (Cont.)

Step 3: Verify Configuration

- The **show ip interface brief** and **show ipv6 interface brief** commands are useful for determining the status of both physical and virtual interfaces. The output shown confirms that interface VLAN 99 has been configured with an IPv4 and IPv6 address.

Note: An IP address applied to the SVI is only for remote management access to the switch; this does not allow the switch to route Layer 3 packets.

```
S1# show ip interface brief
Interface      IP-Address      OK? Method     Status      Protocol
Vlan99          172.17.99.11    YES manual    down       down
(output omitted)
S1# show ipv6 interface brief
Vlan99                  [down/down]
FE80::C27B:BCFF:FEC4:A9C1
2001:DB8:ACAD:99::1
(output omitted)
```

Configure Switch Ports

Switch Verification Commands

Task	IOS Commands
Display interface status and configuration.	S1# show interfaces [<i>interface-id</i>]
Display current startup configuration.	S1# show startup-config
Display current running configuration.	S1# show running-config
Display information about flash file system.	S1# show flash
Display system hardware and software status.	S1# show version
Display history of command entered.	S1# show history
Display IP information about an interface.	S1# show ip interface [<i>interface-id</i>] OR S1# show ipv6 interface [<i>interface-id</i>]
Display the MAC address table.	S1# show mac-address-table OR S1# show mac address-table

Configure Switch Ports

Verify Switch Port Configuration (Cont.)

The **show interfaces** command is another commonly used command, which displays status and statistics information on the network interfaces of the switch. The **show interfaces** command is frequently used when configuring and monitoring network devices.

The first line of the output for the **show interfaces fastEthernet 0/18** command indicates that the FastEthernet 0/18 interface is up/up, meaning that it is operational. Further down, the output shows that the duplex is full and the speed is 100 Mbps.

```
S1# show interfaces fastEthernet 0/18
FastEthernet0/18 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 0025.83e6.9092 (bia 0025.83e6.9092)
    MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
      reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA, loopback not set
    Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is 10/100BaseTX
```

Configure Switch Ports

Interface Input and Output Errors

“Input errors” is the sum of all errors in datagrams that were received on the interface being examined. This includes runts, giants, CRC, no buffer, frame, overrun, and ignored counts. The reported input errors from the **show interfaces** command include the following:

- **Runt Frames** - Ethernet frames that are shorter than the 64-byte minimum allowed length are called runts. Malfunctioning NICs are the usual cause of excessive runt frames, but they can also be caused by collisions.
- **Giants** - Ethernet frames that are larger than the maximum allowed size are called giants.
- **CRC errors** - On Ethernet and serial interfaces, CRC errors usually indicate a media or cable error. Common causes include electrical interference, loose or damaged connections, or incorrect cabling. If you see many CRC errors, there is too much noise on the link and you should inspect the cable. You should also search for and eliminate noise sources.

Telnet Operation

Telnet uses TCP port 23. It is an older protocol that uses unsecure plaintext transmission of both the login authentication (username and password) and the data transmitted between the communicating devices.

A threat actor can monitor packets using Wireshark. For example, in the figure the threat actor captured the username **admin** and password **ccna** from a Telnet session.

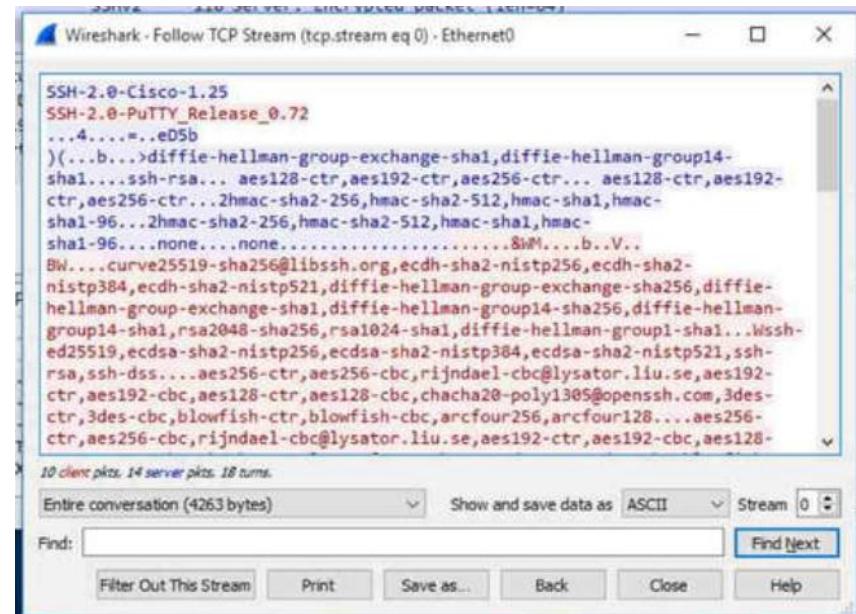


Secure Remote Access

SSH Operation

Secure Shell (SSH) is a secure protocol that uses TCP port 22. It provides a secure (encrypted) management connection to a remote device. SSH should replace Telnet for management connections. SSH provides security for remote connections by providing strong encryption when a device is authenticated (username and password) and also for the transmitted data between the communicating devices.

The figure shows a Wireshark capture of an SSH session. The threat actor can track the session using the IP address of the administrator device. However, unlike Telnet, with SSH the username and password are encrypted.



Verify the Switch Supports SSH

To enable SSH on a Catalyst 2960 switch, the switch must be using a version of the IOS software including cryptographic (encrypted) features and capabilities. Use the **show version** command on the switch to see which IOS the switch is currently running. An IOS filename that includes the combination “k9” supports cryptographic (encrypted) features and capabilities.

The example shows the output of the **show version** command.

```
S1# show version
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(2)SE7, RELEASE SOFTWARE
(fcl)
```

Configure Basic Router Settings

Cisco routers and Cisco switches have many similarities. They support a similar modal operating system, similar command structures, and many of the same commands. In addition, both devices have similar initial configuration steps. For example, the following configuration tasks should always be performed. Name the device to distinguish it from other routers and configure passwords, as shown in the example.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# hostname R1
R1(config)# enable secret class
R1(config)# line console 0
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# service password-encryption
R1(config)#
```

Basic Router Configuration

Configure Basic Router Settings (Cont.)

Configure a banner to provide legal notification of unauthorized access, as shown in the example.

```
R1(config)# banner motd $ Authorized Access Only! $  
R1(config)#
```

Save the changes on a router, as shown in the example.

```
R1# copy running-config startup-config  
Destination filename [startup-config]?  
Building configuration...  
[OK]
```

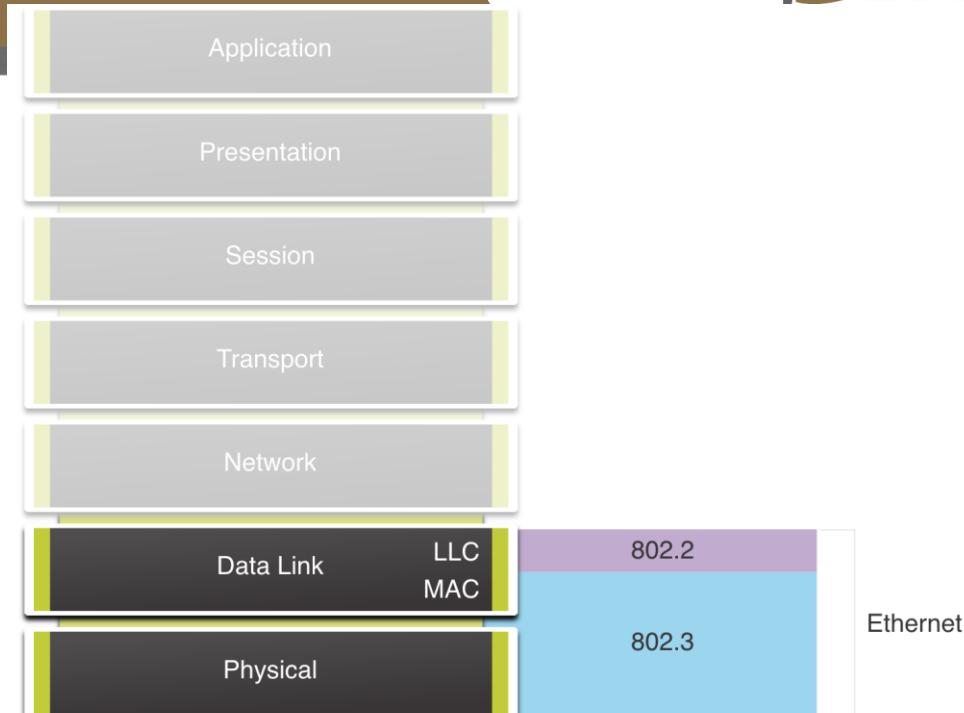
Ethernet Switching

Ethernet Frames

Ethernet Encapsulation



- Ethernet operates in the data link layer and the physical layer.
- It is a family of networking technologies defined in the IEEE 802.2 and 802.3 standards.



Ethernet Frames

MAC Sublayer

The MAC sublayer is responsible for data encapsulation and accessing the media.

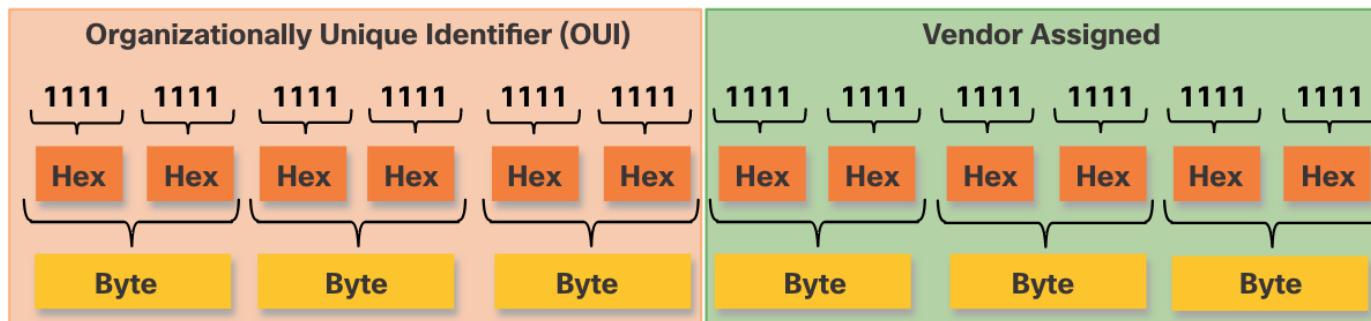
Data Encapsulation

IEEE 802.3 data encapsulation includes the following:

1. **Ethernet frame** - This is the internal structure of the Ethernet frame.
2. **Ethernet Addressing** - The Ethernet frame includes both a source and destination MAC address to deliver the Ethernet frame from Ethernet NIC to Ethernet NIC on the same LAN.
3. **Ethernet Error detection** - The Ethernet frame includes a frame check sequence (FCS) trailer used for error detection.

Ethernet MAC Address

- In an Ethernet LAN, every network device is connected to the same, shared media. MAC addressing provides a method for device identification at the data link layer of the OSI model.
- An Ethernet MAC address is a 48-bit address expressed using 12 hexadecimal digits. Because a byte equals 8 bits, we can also say that a MAC address is 6 bytes in length.
- All MAC addresses must be unique to the Ethernet device or Ethernet interface. To ensure this, all vendors that sell Ethernet devices must register with the IEEE to obtain a unique 6 hexadecimal (i.e., 24-bit or 3-byte) code called the organizationally unique identifier (OUI).
- An Ethernet MAC address consists of a 6 hexadecimal vendor OUI code followed by a 6 hexadecimal vendor-assigned value.



Ethernet MAC Addresses

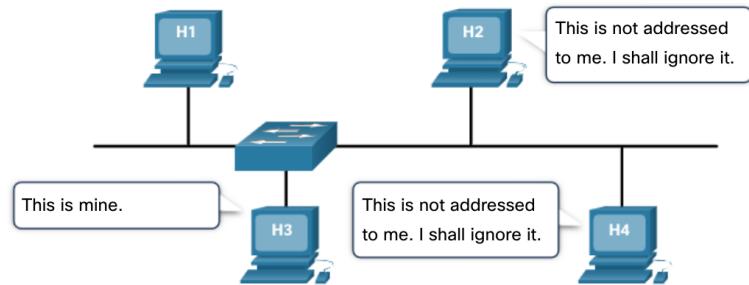
Frame Processing

- When a device is forwarding a message to an Ethernet network, the Ethernet header include a Source MAC address and a Destination MAC address.
- When a NIC receives an Ethernet frame, it examines the destination MAC address to see if it matches the physical MAC address that is stored in RAM. If there is no match, the device discards the frame. If there is a match, it passes the frame up the OSI layers, where the de-encapsulation process takes place.

Note: Ethernet NICs will also accept frames if the destination MAC address is a broadcast or a multicast group of which the host is a member.

- Any device that is the source or destination of an Ethernet frame, will have an Ethernet NIC and therefore, a MAC address. This includes workstations, servers, printers, mobile devices, and routers.

Destination Address	Source Address	Data
CC:CC:CC:CC:CC:CC	AA:AA:AA:AA:AA:AA	Encapsulated data
Frame Addressing		

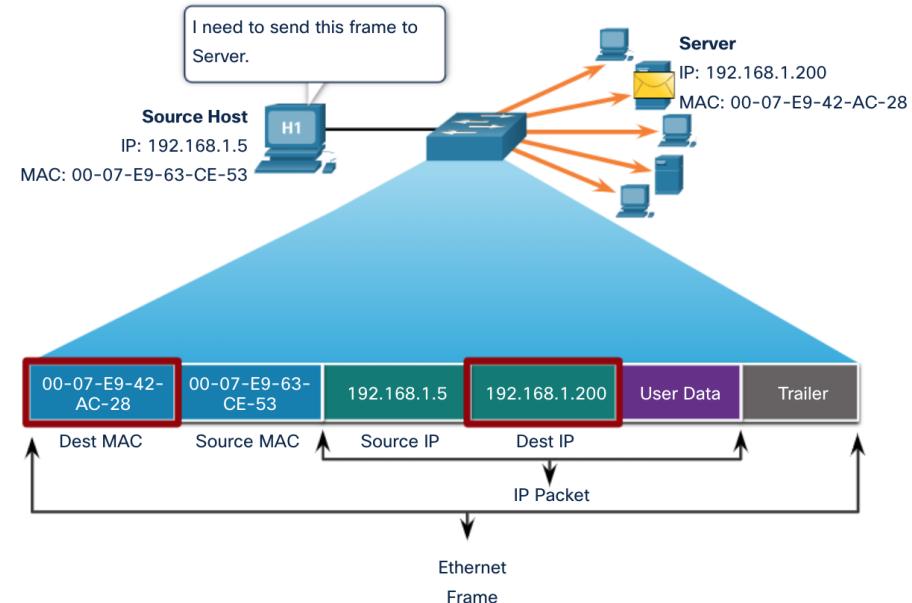


Unicast MAC Address

In Ethernet, different MAC addresses are used for Layer 2 unicast, broadcast, and multicast communications.

- A unicast MAC address is the unique address that is used when a frame is sent from a single transmitting device to a single destination device.
- The process that a source host uses to determine the destination MAC address associated with an IPv4 address is known as Address Resolution Protocol (ARP). The process that a source host uses to determine the destination MAC address associated with an IPv6 address is known as Neighbor Discovery (ND).

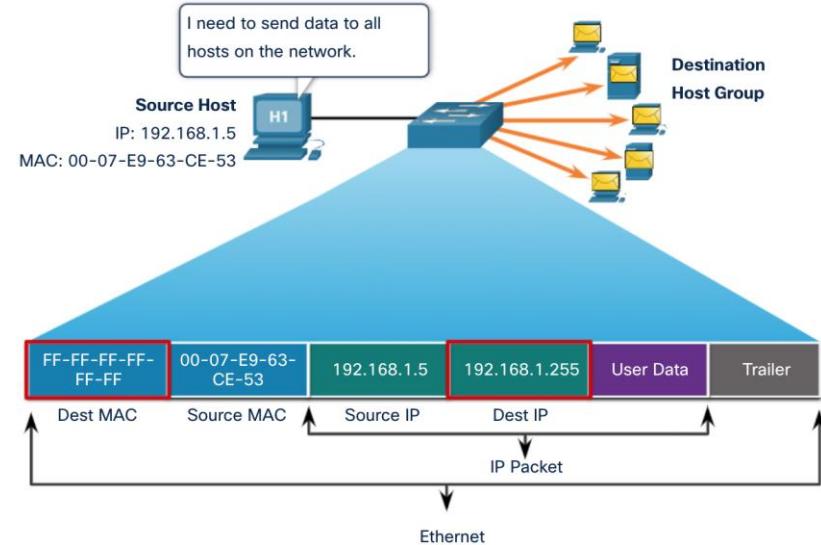
Note: The source MAC address must always be a unicast.



Broadcast MAC Address

An Ethernet broadcast frame is received and processed by every device on the Ethernet LAN. The features of an Ethernet broadcast are as follows:

- It has a destination MAC address of FF-FF-FF-FF-FF-FF in hexadecimal (48 ones in binary).
- It is flooded out all Ethernet switch ports except the incoming port. It is not forwarded by a router.
- If the encapsulated data is an IPv4 broadcast packet, this means the packet contains a destination IPv4 address that has all ones (1s) in the host portion. This numbering in the address means that all hosts on that local network (broadcast domain) will receive and process the packet.

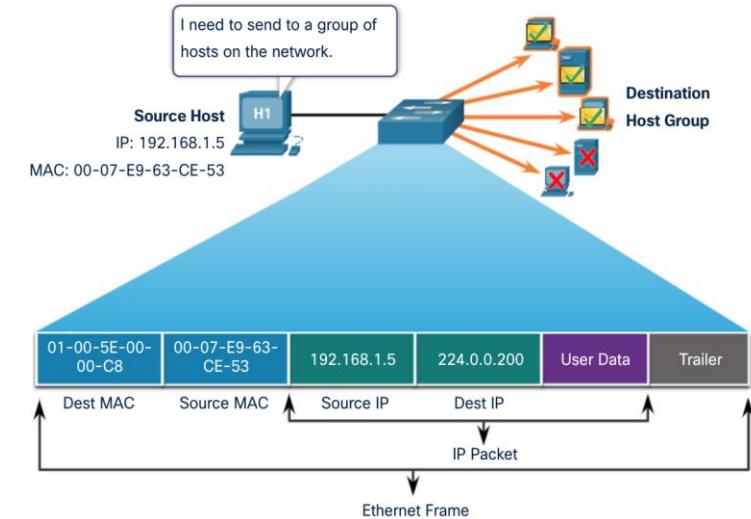


Ethernet MAC Addresses

Multicast MAC Address

An Ethernet multicast frame is received and processed by a group of devices that belong to the same multicast group.

- There is a destination MAC address of 01-00-5E when the encapsulated data is an IPv4 multicast packet and a destination MAC address of 33-33 when the encapsulated data is an IPv6 multicast packet.
- There are other reserved multicast destination MAC addresses for when the encapsulated data is not IP, such as Spanning Tree Protocol (STP).
- It is flooded out all Ethernet switch ports except the incoming port, unless the switch is configured for multicast snooping. It is not forwarded by a router, unless the router is configured to route multicast packets.
- Because multicast addresses represent a group of addresses (sometimes called a host group), they can only be used as the destination of a packet. The source will always be a unicast address.
- As with the unicast and broadcast addresses, the multicast IP address requires a corresponding multicast MAC address.

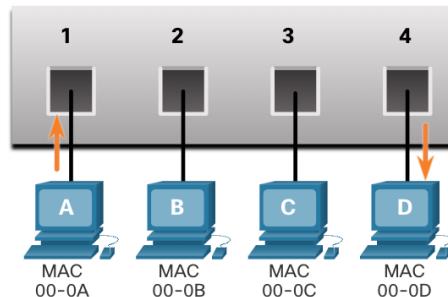


The MAC Address Table

Filtering Frames

As a switch receives frames from different devices, it is able to populate its MAC address table by examining the source MAC address of every frame. When the MAC address table of the switch contains the destination MAC address, it is able to filter the frame and forward out a single port.

MAC Address Table	
Port	MAC Address
1	00-0A
4	00-0D



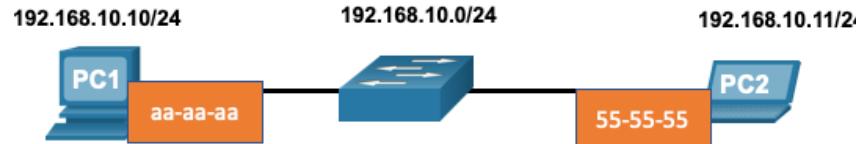
Address Resolution

Destination on Same Network

There are two primary addresses assigned to a device on an Ethernet LAN:

- **Layer 2 physical address (the MAC address)** – Used for NIC to NIC communications on the same Ethernet network.
- **Layer 3 logical address (the IP address)** – Used to send the packet from the source device to the destination device.

Layer 2 addresses are used to deliver frames from one NIC to another NIC on the same network. If a destination IP address is on the same network, the destination MAC address will be that of the destination device.

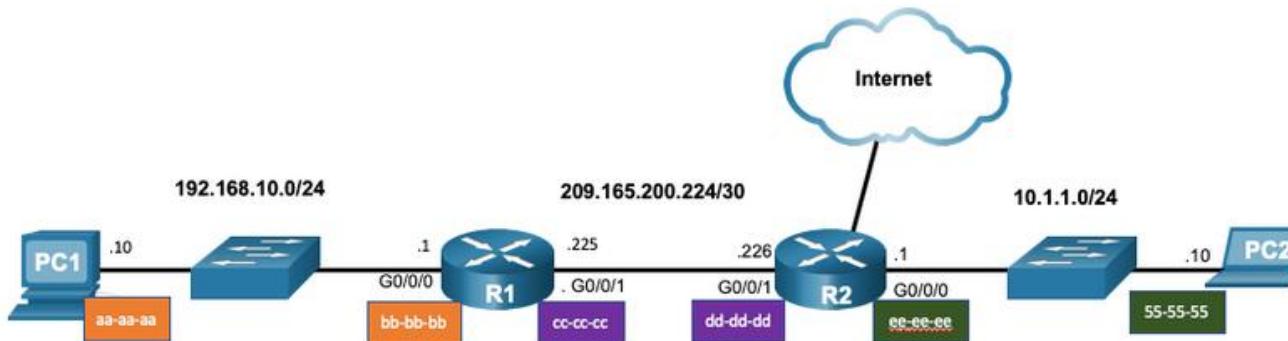


Destination MAC	Source MAC	Source IPv4	Destination IPv4
55-55-55	aa-aa-aa	192.168.10.10	192.168.10.11

Destination on Remote Network

When the destination IP address is on a remote network, the destination MAC address is that of the default gateway.

- ARP is used by IPv4 to associate the IPv4 address of a device with the MAC address of the device NIC.
- ICMPv6 is used by IPv6 to associate the IPv6 address of a device with the MAC address of the device NIC.



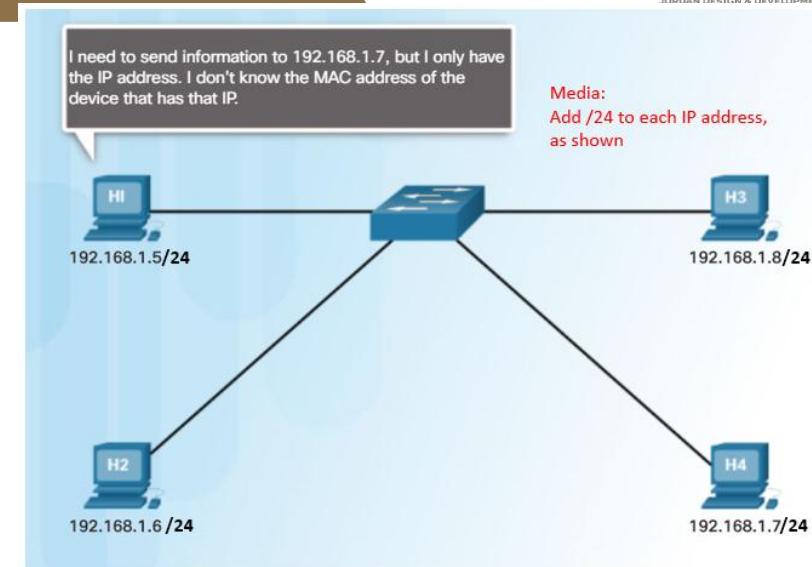
Destination MAC	Source MAC	Source IPv4	Destination IPv4
bb-bb-bb	aa-aa-aa	192.168.10.10	10.1.1.10

ARP Overview

A device uses ARP to determine the destination MAC address of a local device when it knows its IPv4 address.

ARP provides two basic functions:

- Resolving IPv4 addresses to MAC addresses
- Maintaining an ARP table of IPv4 to MAC address mappings



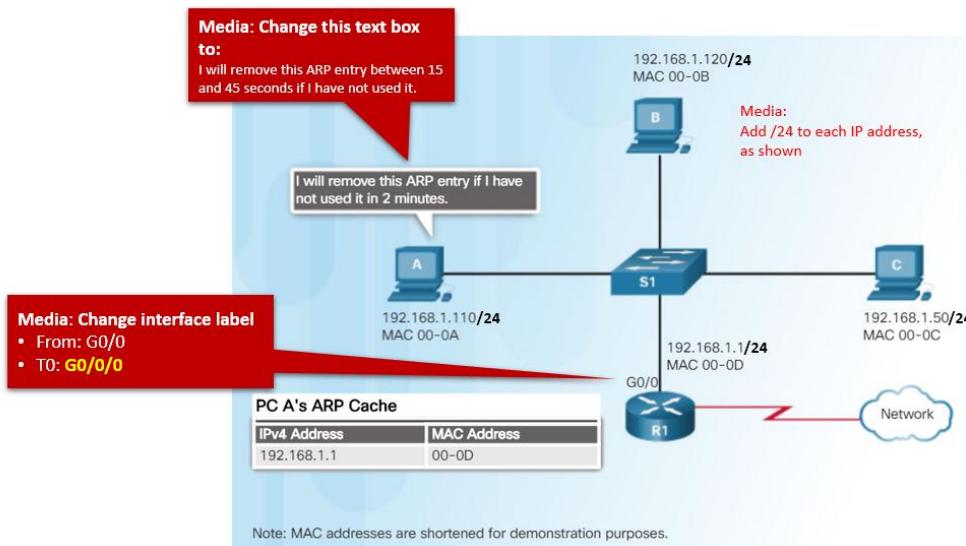
ARP Functions

To send a frame, a device will search its ARP table for a destination IPv4 address and a corresponding MAC address.

- If the packet's destination IPv4 address is on the same network, the device will search the ARP table for the destination IPv4 address.
- If the destination IPv4 address is on a different network, the device will search the ARP table for the IPv4 address of the default gateway.
- If the device locates the IPv4 address, its corresponding MAC address is used as the destination MAC address in the frame.
- If there is no ARP table entry is found, then the device sends an ARP request.

Removing Entries from an ARP Table

- Entries in the ARP table are not permanent and are removed when an ARP cache timer expires after a specified period of time.
- The duration of the ARP cache timer differs depending on the operating system.
- ARP table entries can also be removed manually by the administrator.



ARP Tables on Networking Devices

- The **show ip arp** command displays the ARP table on a Cisco router.
- The **arp -a** command displays the ARP table on a Windows 10 PC.

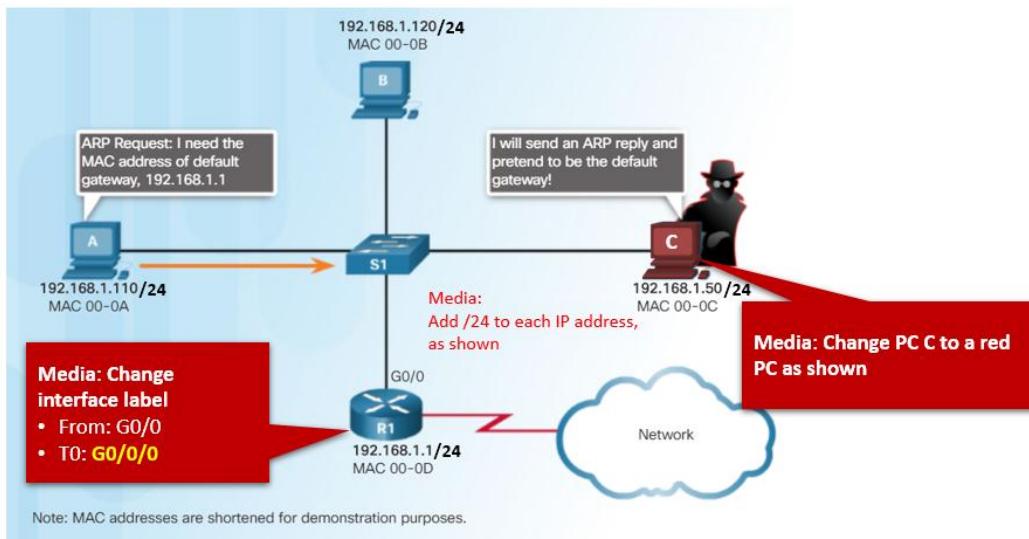
```
R1# show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type    Interface
Internet  192.168.10.1      -          a0e0.af0d.e140  ARPA   GigabitEthernet0/0/0
```

```
C:\Users\PC> arp -a

Interface: 192.168.1.124 --- 0x10
 Internet Address      Physical Address      Type
 192.168.1.1           c8-d7-19-cc-a0-86    dynamic
 192.168.1.101         08-3e-0c-f5-f7-77    dynamic
```

ARP Issues – ARP Broadcasting and ARP Spoofing

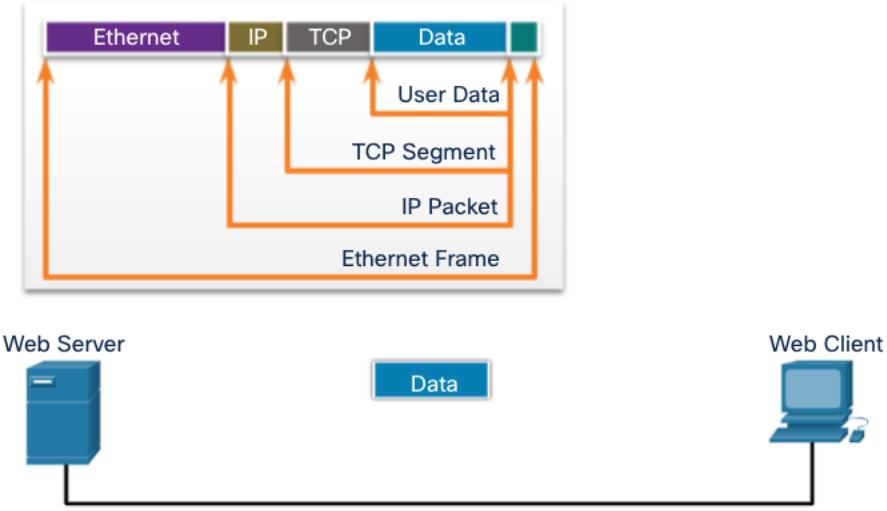
- ARP requests are received and processed by every device on the local network.
- Excessive ARP broadcasts can cause some reduction in performance.
- ARP replies can be spoofed by a threat actor to perform an ARP poisoning attack.
- Enterprise level switches include mitigation techniques to protect against ARP attacks.



Physical Layer

The Physical Layer

- Transports bits across the network media
- Accepts a complete frame from the Data Link Layer and encodes it as a series of signals that are transmitted to the local media
- This is the last step in the encapsulation process.
- The next device in the path to the destination receives the bits and re-encapsulates the frame, then decides what to do with it.



Physical Components

Physical Layer Standards address three functional areas:

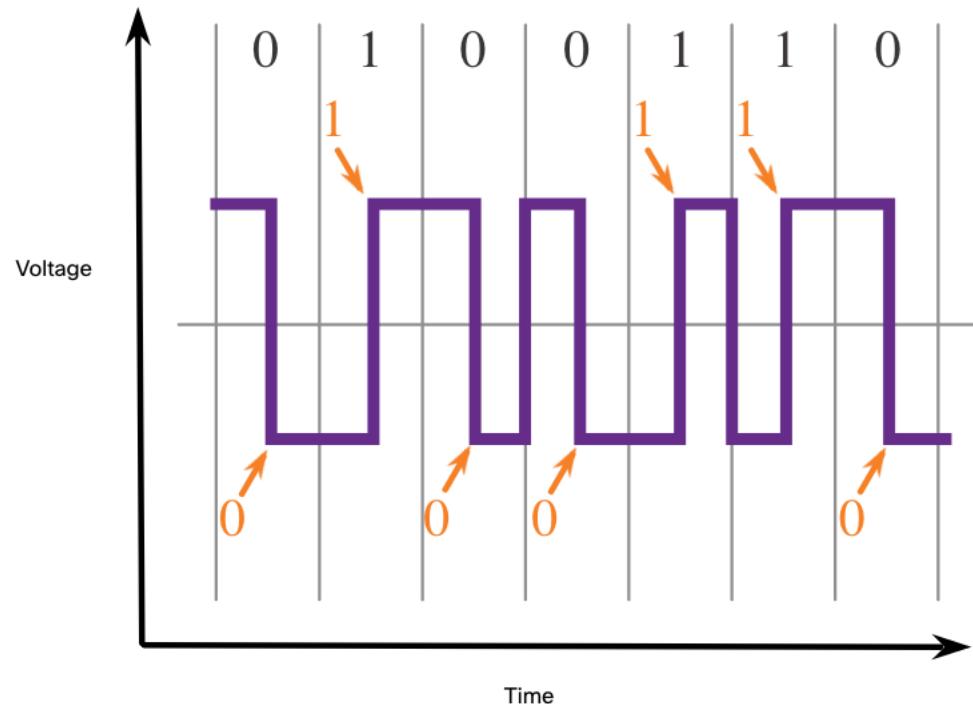
- Physical Components
- Encoding
- Signaling

The Physical Components are the hardware devices, media, and other connectors that transmit the signals that represent the bits.

- Hardware components like NICs, interfaces and connectors, cable materials, and cable designs are all specified in standards associated with the physical layer.

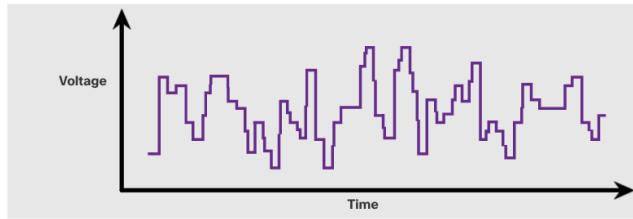
Encoding

- Encoding converts the stream of bits into a format recognizable by the next device in the network path.
- This ‘coding’ provides predictable patterns that can be recognized by the next device.
- Examples of encoding methods include Manchester (shown in the figure), 4B/5B, and 8B/10B.

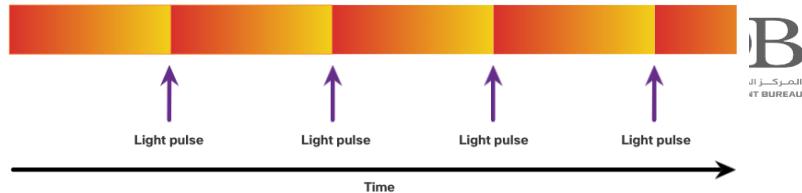


Signaling

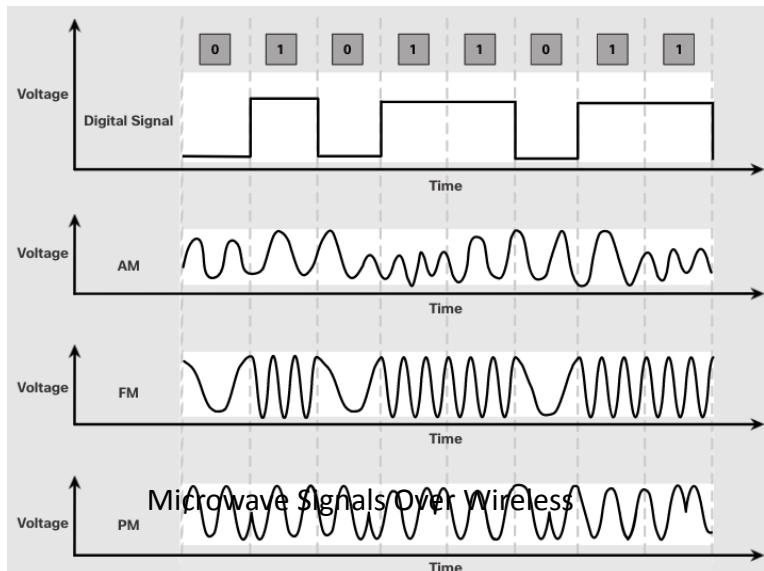
- The signaling method is how the bit values, “1” and “0” are represented on the physical medium.
- The method of signaling will vary based on the type of medium being used.



Electrical Signals Over Copper Cable



Light Pulses Over Fiber-Optic Cable



Characteristics of Copper Cabling

Copper cabling is the most common type of cabling used in networks today. It is inexpensive, easy to install, and has low resistance to electrical current flow.

Limitations:

- Attenuation – the longer the electrical signals have to travel, the weaker they get.
- The electrical signal is susceptible to interference from two sources, which can distort and corrupt the data signals (Electromagnetic Interference (EMI) and Radio Frequency Interference (RFI) and Crosstalk).

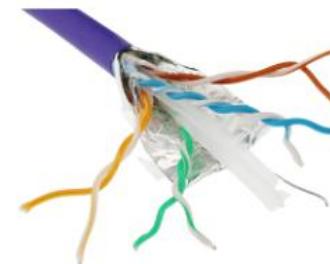
Mitigation:

- Strict adherence to cable length limits will mitigate attenuation.
- Some kinds of copper cable mitigate EMI and RFI by using metallic shielding and grounding.
- Some kinds of copper cable mitigate crosstalk by twisting opposing circuit pair wires together.

Types of Copper Cabling



Unshielded Twisted-Pair (UTP) Cable

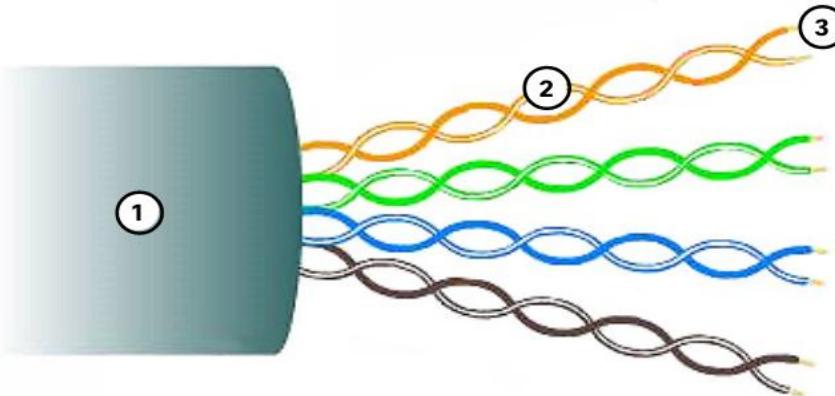


Shielded Twisted-Pair (STP) Cable



Coaxial Cable

Unshielded Twisted Pair (UTP)

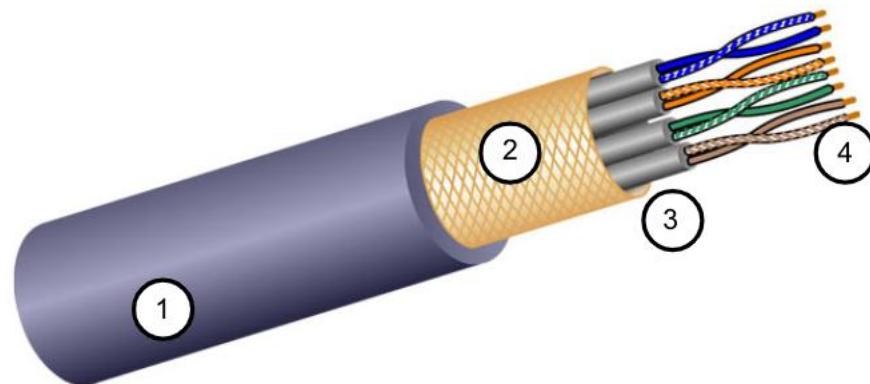


- UTP is the most common networking media.
- Terminated with RJ-45 connectors
- Interconnects hosts with intermediary network devices.

Key Characteristics of UTP

1. The outer jacket protects the copper wires from physical damage.
2. Twisted pairs protect the signal from interference.
3. Color-coded plastic insulation electrically isolates the wires from each other and identifies each pair.

Shielded Twisted Pair (STP)



- Better noise protection than UTP
- More expensive than UTP
- Harder to install than UTP
- Terminated with RJ-45 connectors
- Interconnects hosts with intermediary network devices

Key Characteristics of STP

1. The outer jacket protects the copper wires from physical damage
2. Braided or foil shield provides EMI/RFI protection
3. Foil shield for each pair of wires provides EMI/RFI protection
4. Color-coded plastic insulation electrically isolates the wires from each other and identifies each pair

Coaxial Cable

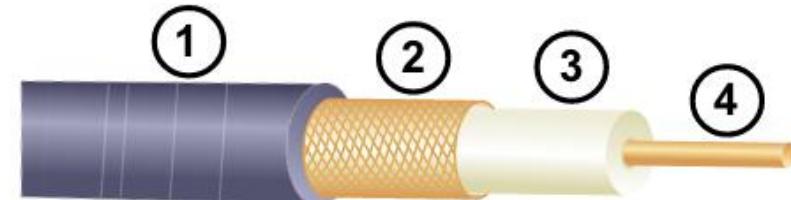
Consists of the following:

1. Outer cable jacket to prevent minor physical damage
2. A woven copper braid, or metallic foil, acts as the second wire in the circuit and as a shield for the inner conductor.
3. A layer of flexible plastic insulation
4. A copper conductor is used to transmit the electronic signals.

There are different types of connectors used with coax cable.

Commonly used in the following situations:

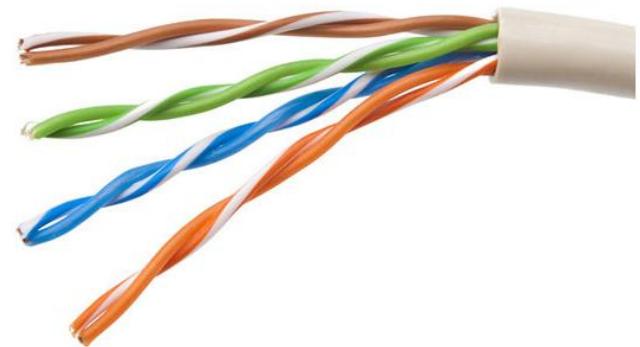
- Wireless installations - attach antennas to wireless devices
- Cable internet installations - customer premises wiring



Properties of UTP Cabling

UTP has four pairs of color-coded copper wires twisted together and encased in a flexible plastic sheath. No shielding is used. UTP relies on the following properties to limit crosstalk:

- Cancellation - Each wire in a pair of wires uses opposite polarity. One wire is negative, the other wire is positive. They are twisted together and the magnetic fields effectively cancel each other and outside EMI/RFI.
- Variation in twists per foot in each wire - Each wire is twisted a different amount, which helps prevent crosstalk amongst the wires in the cable.



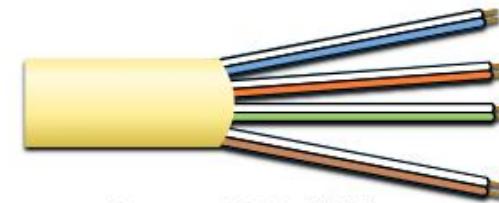
UTP Cabling Standards and Connectors

Standards for UTP are established by the TIA/EIA. TIA/EIA-568 standardizes elements like:

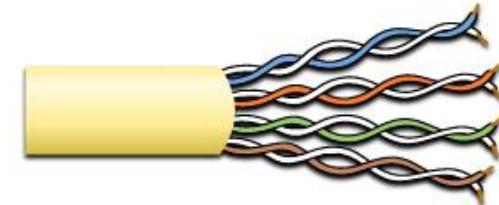
- Cable Types
- Cable Lengths
- Connectors
- Cable Termination
- Testing Methods

Electrical standards for copper cabling are established by the IEEE, which rates cable according to its performance. Examples include:

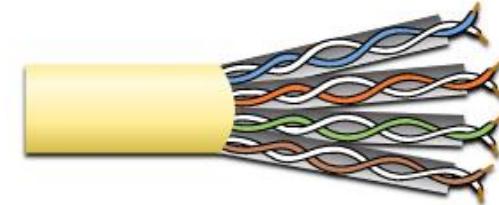
- Category 3
- Category 5 and 5e
- Category 6



Category 3 Cable (UTP)



Category 5 and 5e Cable (UTP)

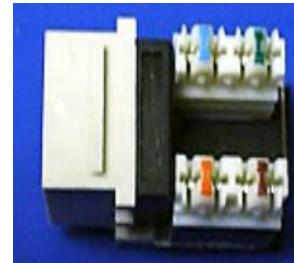


Category 6 Cable (UTP)

UTP Cabling Standards and Connectors



RJ-45 Connector



RJ-45 Socket

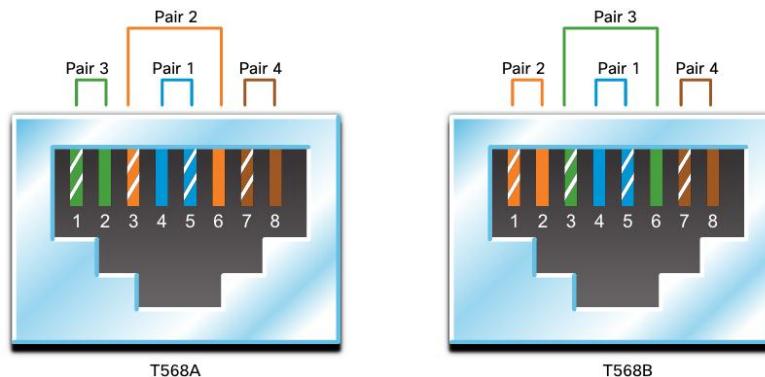


Poorly terminated UTP cable



Properly terminated UTP cable

Straight-through and Crossover UTP Cables



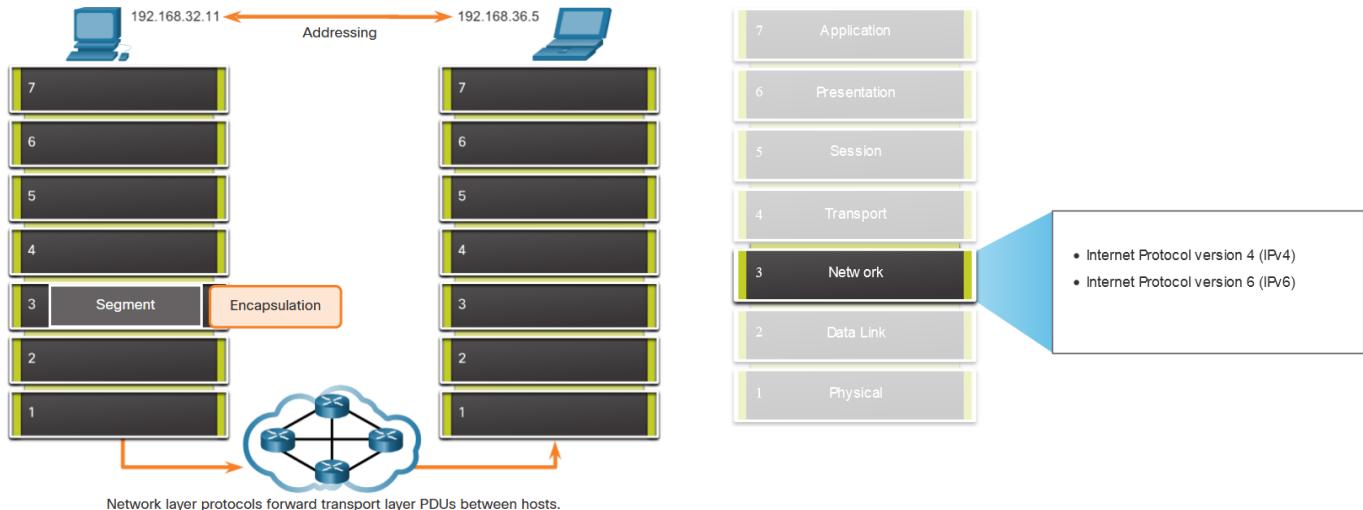
Cable Type	Standard	Application
Ethernet Straight-through	Both ends T568A or T568B	Host to Network Device
Ethernet Crossover *	One end T568A, other end T568B	Host-to-Host, Switch-to-Switch, Router-to-Router
* Considered Legacy due to most NICs using Auto-MDIX to sense cable type and complete connection		
Rollover	Cisco Proprietary	Host serial port to Router or Switch Console Port, using an adapter

Network Layer

The Network Layer



- Provides services to allow end devices to exchange data
- IP version 4 (IPv4) and IP version 6 (IPv6) are the principle network layer communication protocols.
- The network layer performs four basic operations:
 - Addressing end devices
 - Encapsulation
 - Routing
 - De-encapsulation



Characteristics of IP

IP is meant to have low overhead and may be described as:

Connectionless

Best Effort

Media Independent



Connectionless

IP is Connectionless

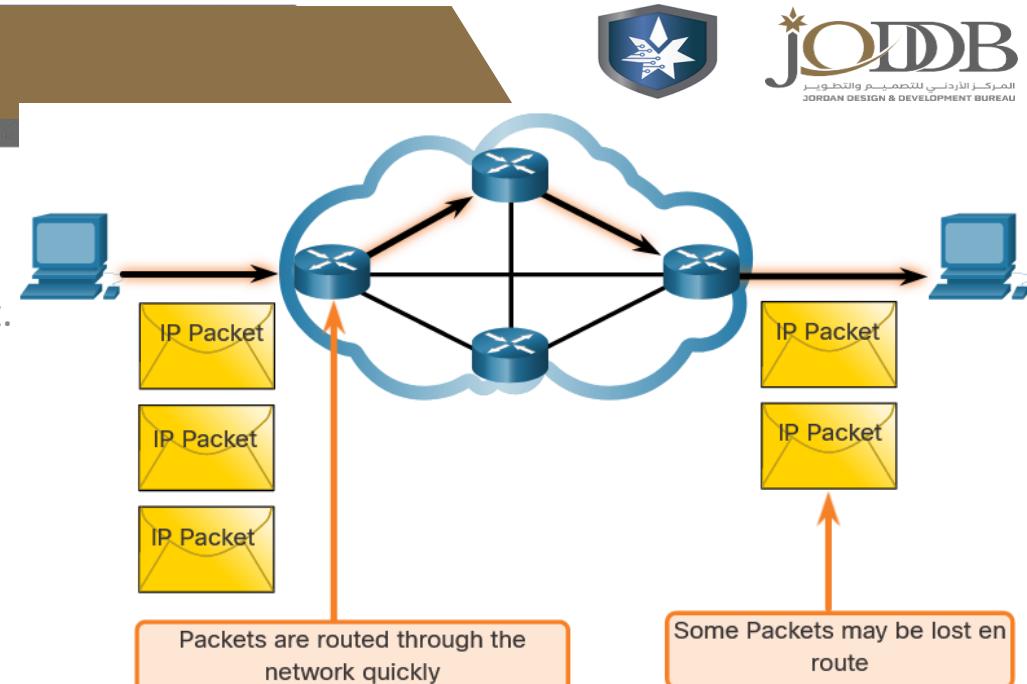
- IP does not establish a connection with the destination before sending the packet.
- There is no control information needed (synchronizations, acknowledgments, etc.).
- The destination will receive the packet when it arrives, but no pre-notifications are sent by IP.
- If there is a need for connection-oriented traffic, then another protocol will handle this (typically TCP at the transport layer).



Best Effort

IP is Best Effort

- IP will not guarantee delivery of the packet.
- IP has reduced overhead since there is no mechanism to resend data that is not received.
- IP does not expect acknowledgments.
- IP does not know if the other device is operational or if it received the packet.



Media Independent

IP is unreliable:

- It cannot manage or fix undelivered or corrupted packets.

- IP cannot retransmit after an error.

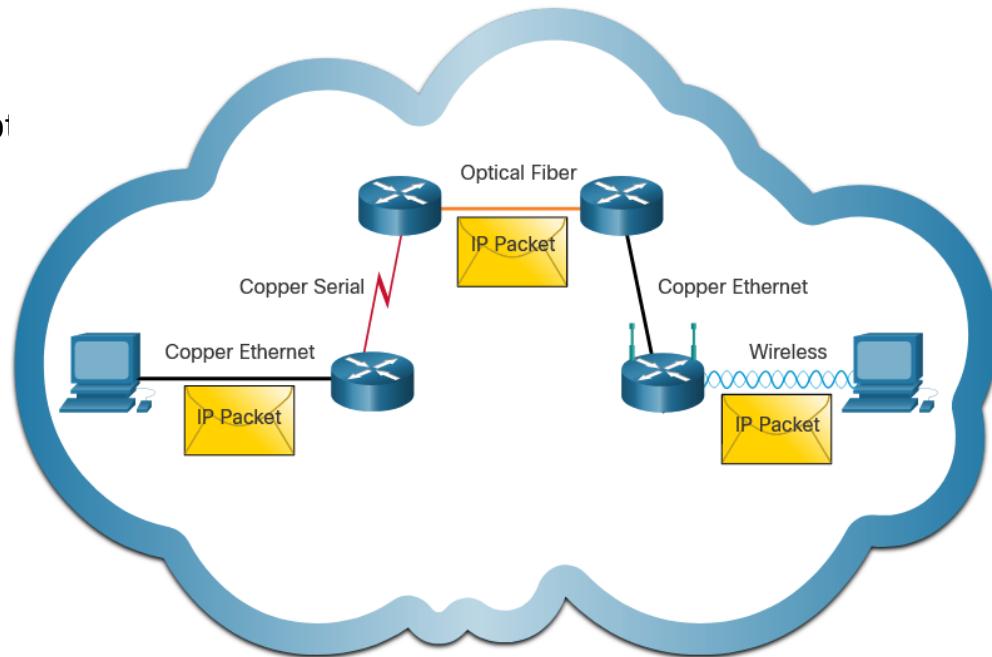
- IP cannot realign out of sequence packets.

- IP must rely on other protocols for these functions.

IP is media Independent:

- IP does not concern itself with the type of frame required at the data link layer or the media type at the physical layer.

- IP can be sent over any media type: copper, fiber, or wireless.



Media Independent (Co.)

The network layer will establish the Maximum Transmission Unit (MTU).

Network layer receives this from control information sent by the data link layer.

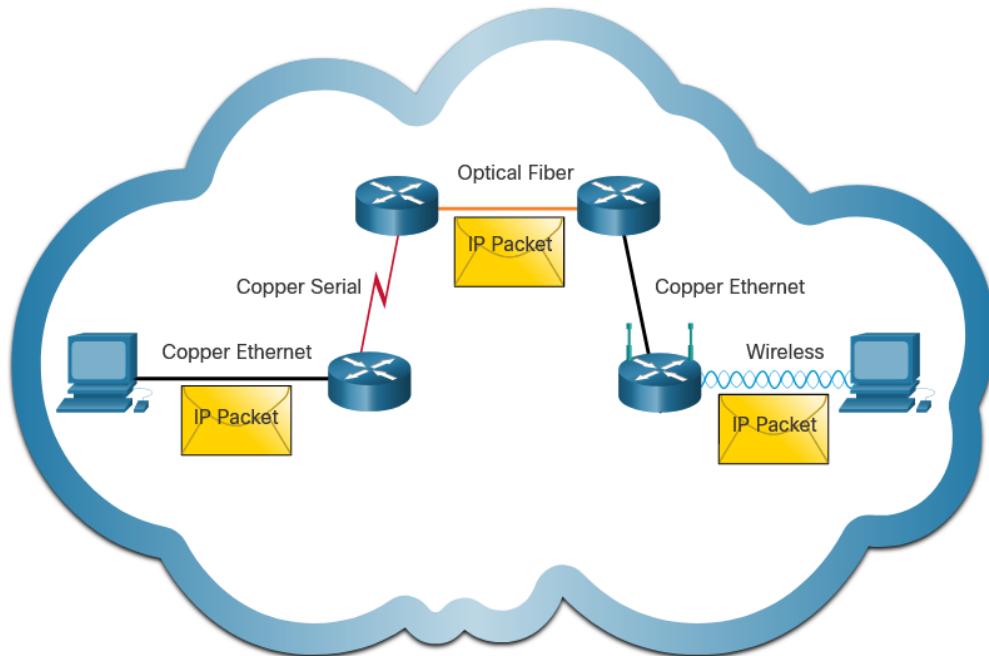
The network then establishes the MTU size.

Fragmentation is when Layer 3 splits the IPv4 packet into smaller units.

Fragmenting causes latency.

IPv6 does not fragment packets.

Example: Router goes from Ethernet to a slow WAN with a smaller MTU



IPv4 Packet Header Fields



The IPv4 network header characteristics:

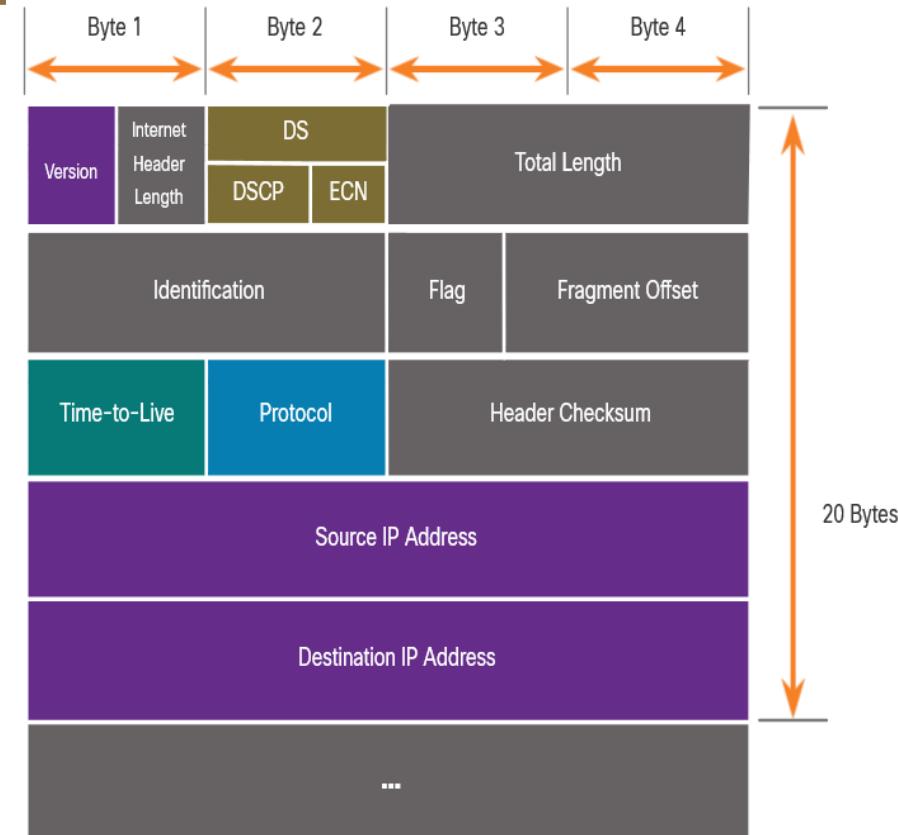
It is in binary.

Contains several fields of information

Diagram is read from left to right, 4 bytes per line

The two most important fields are the source and destination.

Protocols may have one or more functions.



IPv4 Packet Header Fields



Function	Description
Version	This will be for v4, as opposed to v6, a 4 bit field= 0100
Differentiated Services	Used for QoS: DiffServ – DS field or the older IntServ – ToS or Type of Service
Header Checksum	Detect corruption in the IPv4 header
Time to Live (TTL)	Layer 3 hop count. When it becomes zero the router will discard the packet.
Protocol	I.D.s next level protocol: ICMP, TCP, UDP, etc.
Source IPv4 Address	32 bit source address
Destination IPV4 Address	32 bit destination address

Default Gateway



A router or layer 3 switch can be a default-gateway.

Features of a default gateway (DGW):

- It must have an IP address in the same range as the rest of the LAN.
- It can accept data from the LAN and is capable of forwarding traffic off of the LAN.
- It can route to other networks.

If a device has no default gateway or a bad default gateway, its traffic will not be able to leave the LAN.

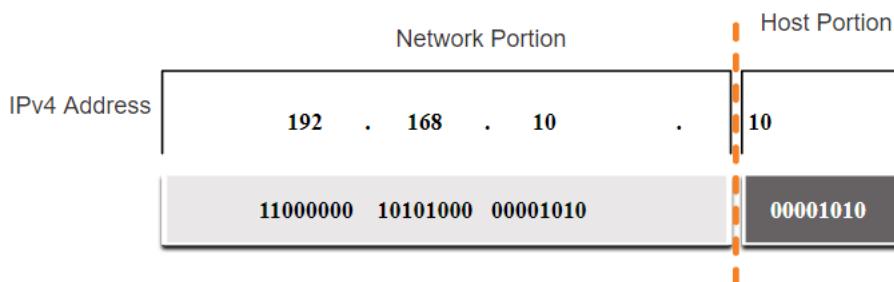
Trace a Route

- Test network connectivity using **ping**.
- Trace a route to a remote server using Windows **tracert**.

IPv4 Addressing

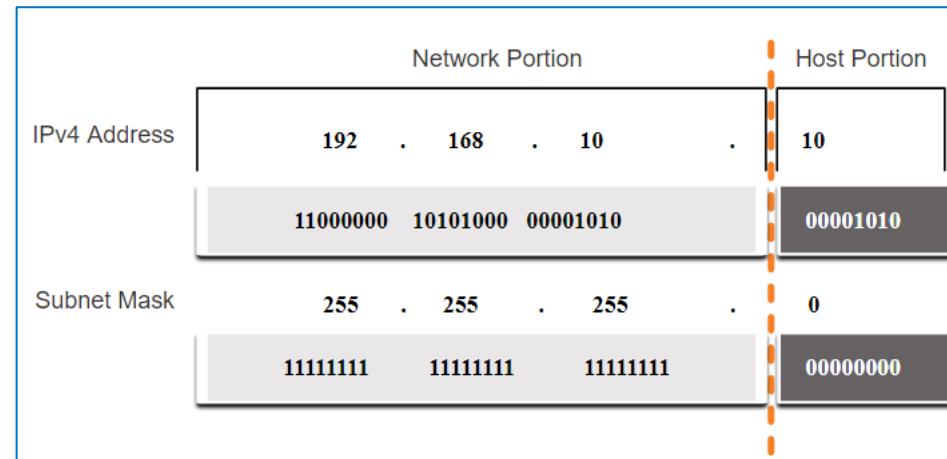
Network and Host Portions

- An IPv4 address is a 32-bit hierarchical address that is made up of a network portion and a host portion.
- When determining the network portion versus the host portion, you must look at the 32-bit stream.
- A subnet mask is used to determine the network and host portions.



The Subnet Mask

- To identify the network and host portions of an IPv4 address, the subnet mask is compared to the IPv4 address bit for bit, from left to right.
- The actual process used to identify the network and host portions is called ANDing.



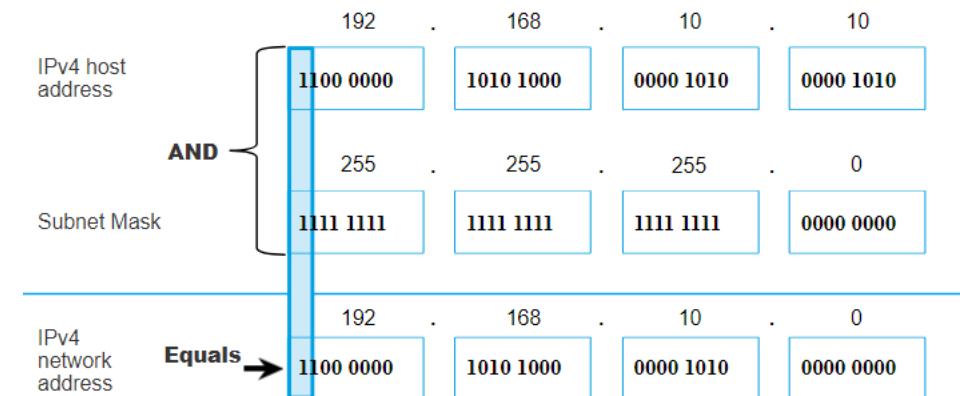
The Prefix Length

- A prefix length is a less cumbersome method used to identify a subnet mask address.
- The prefix length is the number of bits set to 1 in the subnet mask.
- It is written in “slash notation” therefore, count the number of bits in the subnet mask and prepend it with a slash.

Subnet Mask	32-bit Address	Prefix Length
255.0.0.0	11111111.00000000.00000000.00000000	/8
255.255.0.0	11111111.11111111.00000000.00000000	/16
255.255.255.0	11111111.11111111.11111111.00000000	/24
255.255.255.128	11111111.11111111.11111111.10000000	/25
255.255.255.192	11111111.11111111.11111111.11000000	/26
255.255.255.224	11111111.11111111.11111111.11100000	/27
255.255.255.240	11111111.11111111.11111111.11110000	/28
255.255.255.248	11111111.11111111.11111111.11111000	/29
255.255.255.252	11111111.11111111.11111111.11111100	/30

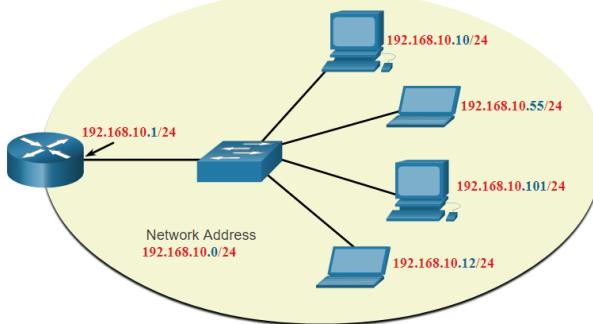
Determining the Network: Logical AND

- A logical AND Boolean operation is used in determining the network address.
- Logical AND is the comparison of two bits where only a 1 AND 1 produces a 1 and any other combination results in a 0.
- $1 \text{ AND } 1 = 1$, $0 \text{ AND } 1 = 0$, $1 \text{ AND } 0 = 0$, $0 \text{ AND } 0 = 0$
- 1 = True and 0 = False
- To identify the network address, the host IPv4 address is logically ANDed, bit by bit, with the subnet mask to identify the network address.



Network, Host, and Broadcast Address

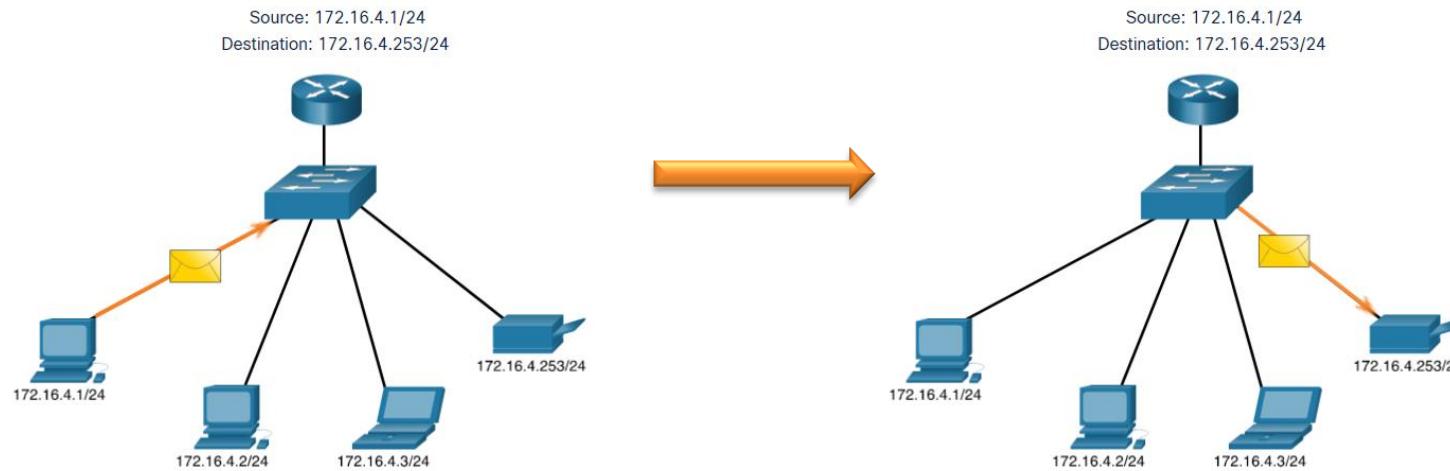
- Within each network are three types of IP addresses:
 - Network address
 - Host addresses
 - Broadcast address



	Network Portion			Host Portion	Host Bits
Subnet mask 255.255.255.0 or /24	255 11111111	255 11111111	255 11111111	0 00000000	
Network address 192.168.10.0 or /24	192 11000000	168 10100000	10 00001010	0 00000000	All Os
First address 192.168.10.1 or /24	192 11000000	168 10100000	10 00001010	1 00000001	All Os and a 1
Last address 192.168.10.254 or /24	192 11000000	168 10100000	10 00001010	254 11111110	All 1s and a 0
Broadcast address 192.168.10.255 or /24	192 11000000	168 10100000	10 00001010	255 11111111	All 1s

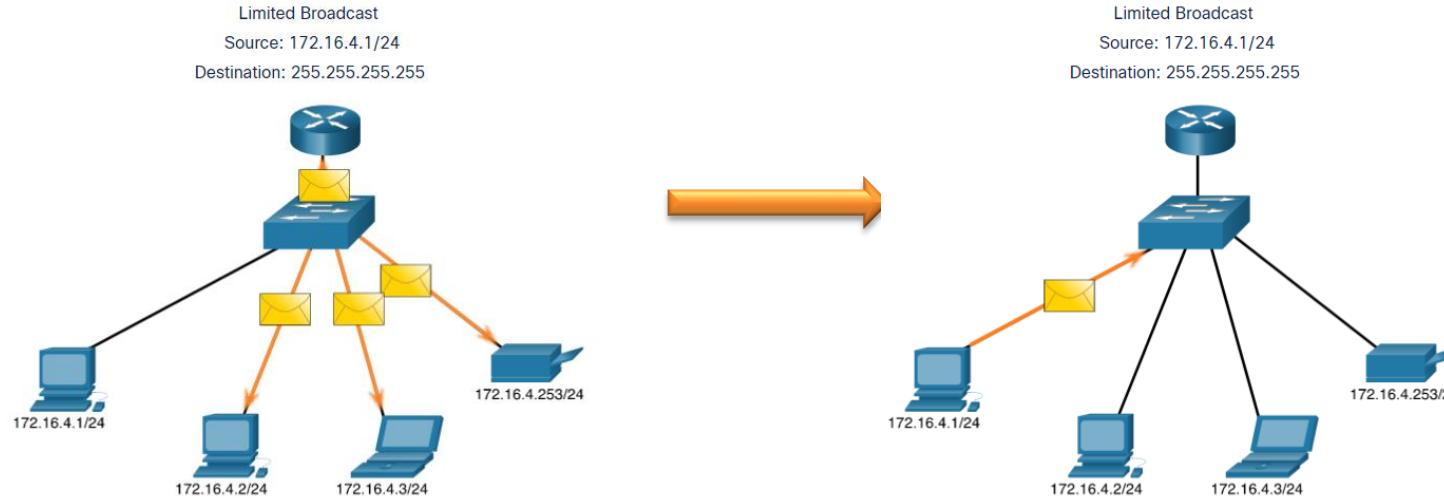
Unicast

- Unicast transmission is sending a packet to one destination IP address.
- For example, the PC at 172.16.4.1 sends a unicast packet to the printer at 172.16.4.253.



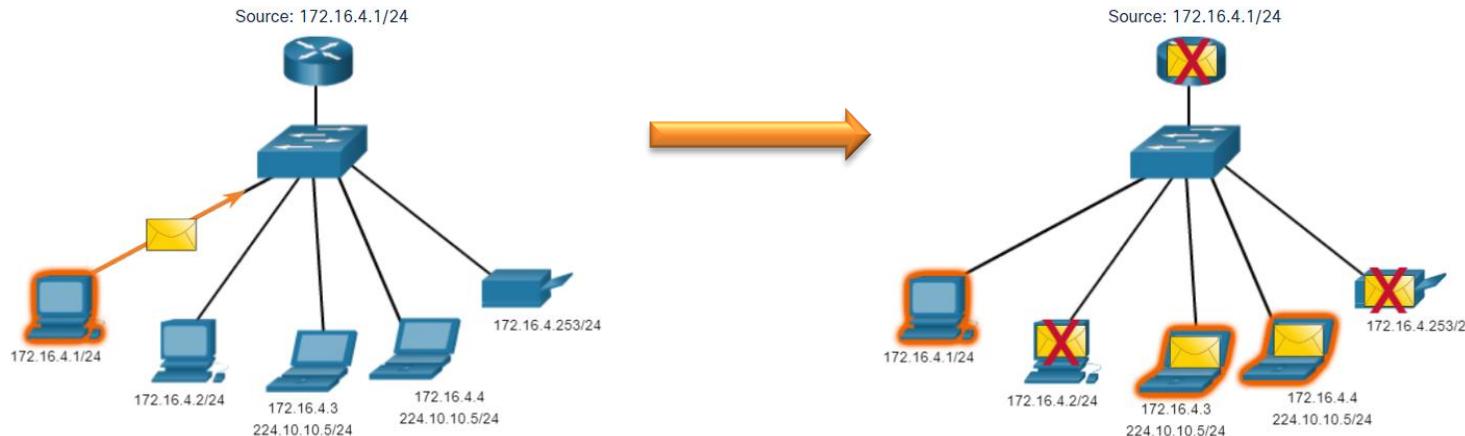
Broadcast

- Broadcast transmission is sending a packet to all other destination IP addresses.
- For example, the PC at 172.16.4.1 sends a broadcast packet to all IPv4 hosts.



Multicast

- Multicast transmission is sending a packet to a multicast address group.
- For example, the PC at 172.16.4.1 sends a multicast packet to the multicast group address 224.10.10.5.



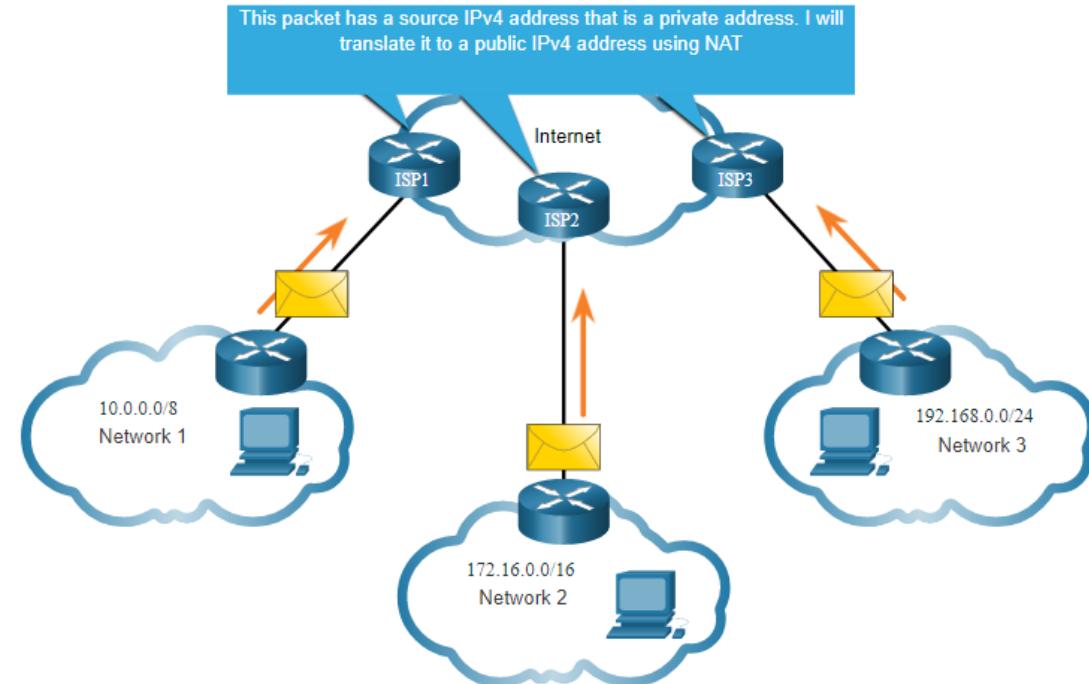
Public and Private IPv4 Addresses

- As defined in RFC 1918, public IPv4 addresses are globally routable between internet service provider (ISP) routers.
- Private addresses are common blocks of addresses used by most organizations to assign IPv4 addresses to internal hosts.
- Private IPv4 addresses are not unique and can be used internally within any network.
- However, private addresses are not globally routable.

Network Address and Prefix	RFC 1918 Private Address Range
10.0.0.0/8	10.0.0.0 - 10.255.255.255
172.16.0.0/12	172.16.0.0 - 172.31.255.255
192.168.0.0/16	192.168.0.0 - 192.168.255.255

Routing to the Internet

- Network Address Translation (NAT) translates private IPv4 addresses to public IPv4 addresses.
- NAT is typically enabled on the edge router connecting to the internet.
- It translates the internal private address to a public global IP address.



Special Use IPv4 Addresses

Loopback addresses

- 127.0.0.0 /8 (127.0.0.1 to 127.255.255.254)
- Commonly identified as only 127.0.0.1
- Used on a host to test if TCP/IP is operational.

```
C:\Users\NetAcad> ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

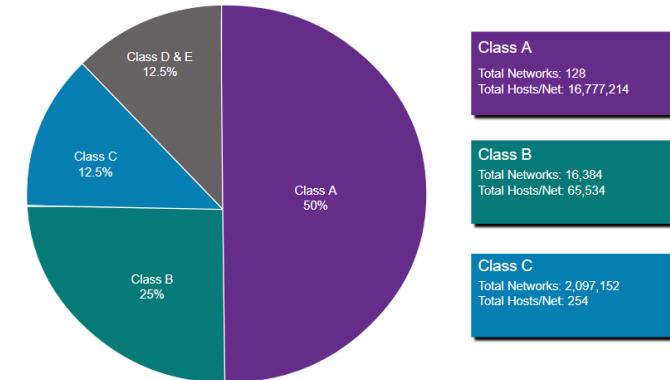
Link-Local addresses

- 169.254.0.0 /16 (169.254.0.1 to 169.254.255.254)
- Commonly known as the Automatic Private IP Addressing (APIPA) addresses or self-assigned addresses.
- Used by Windows DHCP clients to self-configure when no DHCP servers are available.

Legacy Classful Addressing

RFC 790 (1981) allocated IPv4 addresses in classes

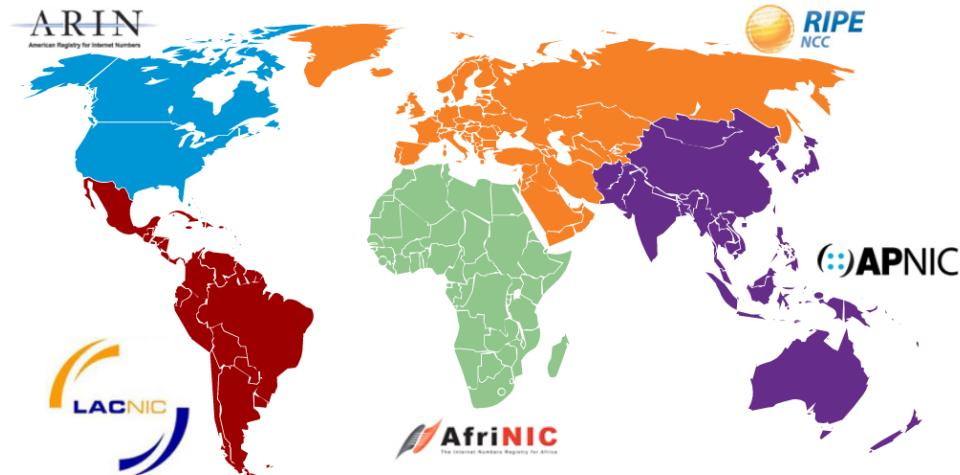
- Class A (0.0.0.0/8 to 127.0.0.0/8)
- Class B (128.0.0.0 /16 – 191.255.0.0 /16)
- Class C (192.0.0.0 /24 – 223.255.255.0 /24)
- Class D (224.0.0.0 to 239.0.0.0)
- Class E (240.0.0.0 – 255.0.0.0)
- Classful addressing wasted many IPv4 addresses.



Classful address allocation was replaced with classless addressing which ignores the rules of classes (A, B, C).

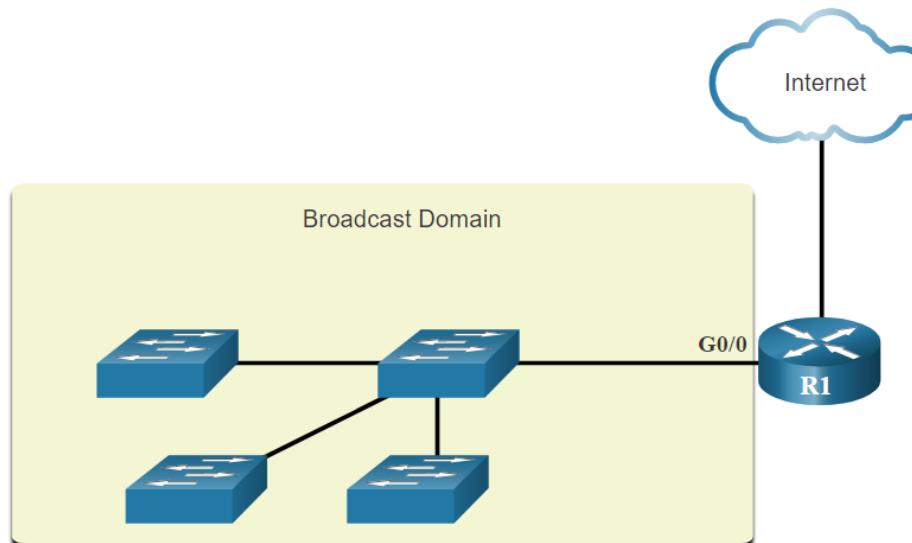
Assignment of IP Addresses

- The Internet Assigned Numbers Authority (IANA) manages and allocates blocks of IPv4 and IPv6 addresses to five Regional Internet Registries (RIRs).
- RIRs are responsible for allocating IP addresses to ISPs who provide IPv4 address blocks to smaller ISPs and organizations.



Broadcast Domains and Segmentation

- Many protocols use broadcasts or multicasts (e.g., ARP use broadcasts to locate other devices, hosts send DHCP discover broadcasts to locate a DHCP server.)
- Switches propagate broadcasts out all interfaces except the interface on which it was received.

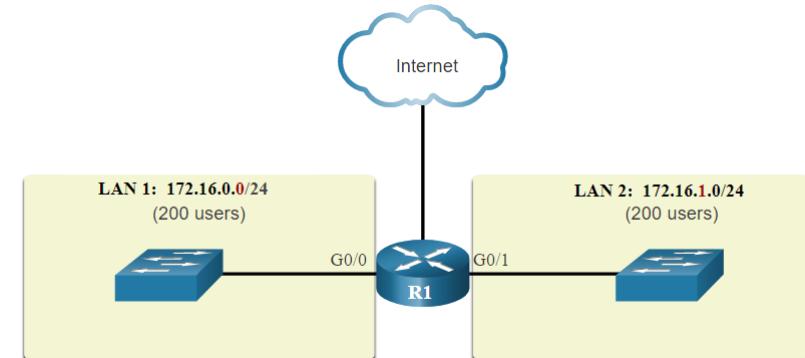
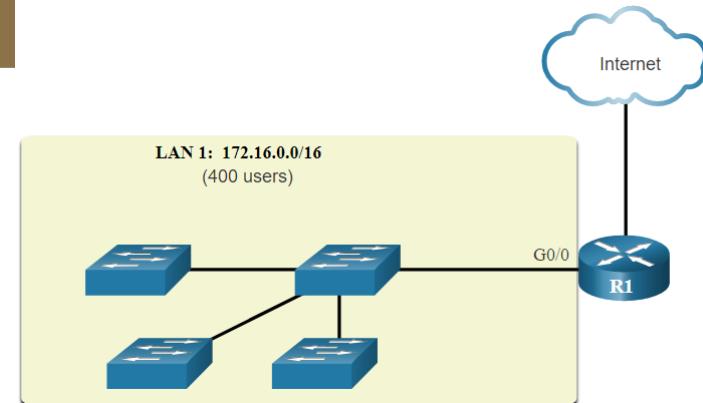


- The only device that stops broadcasts is a router.
- Routers do not propagate broadcasts.
- Each router interface connects to a broadcast domain and broadcasts are only propagated within that specific broadcast domain.

Problems with Large Broadcast Domains



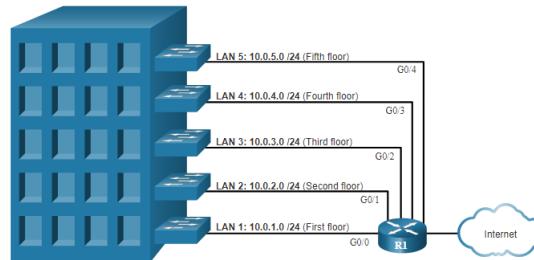
- A problem with a large broadcast domain is that these hosts can generate excessive broadcasts and negatively affect the network.
- The solution is to reduce the size of the network to create smaller broadcast domains in a process called subnetting.
- Dividing the network address 172.16.0.0 /16 into two subnets of 200 users each: 172.16.0.0 /24 and 172.16.1.0 /24.
- Broadcasts are only propagated within the smaller broadcast domains.



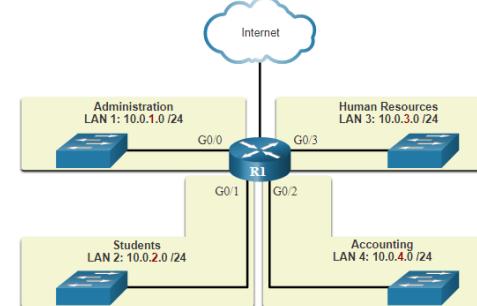
Reasons for Segmenting Networks

- Subnetting reduces overall network traffic and improves network performance.
- It can be used to implement security policies between subnets.
- Subnetting reduces the number of devices affected by abnormal broadcast traffic.
- Subnets are used for a variety of reasons including by:

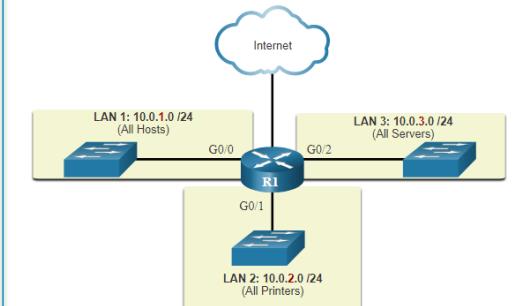
Location



Group or Function



Device Type



Subnet on an Octet Boundary

- Networks are most easily subnetted at the octet boundary of /8, /16, and /24.
- Notice that using longer prefix lengths decreases the number of hosts per subnet.

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of hosts
/8	255.0.0.0	nnnnnnnn . hhhhhh . hhhhhh . hhhhhh 11111111 . 00000000 . 00000000 . 00000000	16,777,214
/16	255.255.0.0	nnnnnnnn . nnnnnnnn . hhhhhh . hhhhhh 11111111 . 11111111 . 00000000 . 00000000	65,534
/24	255.255.255.0	nnnnnnnn . nnnnnnnn . nnnnnnnn . hhhhhh 11111111 . 11111111 . 11111111 . 00000000	254

Subnet on an Octet Boundary (Cont.)

- In the first table 10.0.0.0/8 is subnetted using /16 and in the second table, a /24 mask.

Subnet Address (256 Possible Subnets)	Host Range (65,534 possible hosts per subnet)	Broadcast
10.0.0.0/16	10.0.0.1 - 10.0.255.254	10.0.255.255
10.1.0.0/16	10.1.0.1 - 10.1.255.254	10.1.255.255
10.2.0.0/16	10.2.0.1 - 10.2.255.254	10.2.255.255
10.3.0.0/16	10.3.0.1 - 10.3.255.254	10.3.255.255
10.4.0.0/16	10.4.0.1 - 10.4.255.254	10.4.255.255
10.5.0.0/16	10.5.0.1 - 10.5.255.254	10.5.255.255
10.6.0.0/16	10.6.0.1 - 10.6.255.254	10.6.255.255
10.7.0.0/16	10.7.0.1 - 10.7.255.254	10.7.255.255
...
10.255.0.0/16	10.255.0.1 - 10.255.255.254	10.255.255.255

Subnet Address (65,536 Possible Subnets)	Host Range (254 possible hosts per subnet)	Broadcast
10.0.0.0/24	10.0.0.1 - 10.0.0.254	10.0.0.255
10.0.1.0/24	10.0.1.1 - 10.0.1.254	10.0.1.255
10.0.2.0/24	10.0.2.1 - 10.0.2.254	10.0.2.255
...
10.0.255.0/24	10.0.255.1 - 10.0.255.254	10.0.255.255
10.1.0.0/24	10.1.0.1 - 10.1.0.254	10.1.0.255
10.1.1.0/24	10.1.1.1 - 10.1.1.254	10.1.1.255
10.1.2.0/24	10.1.2.1 - 10.1.2.254	10.1.2.255
...
10.100.0.0/24	10.100.0.1 - 10.100.0.254	10.100.0.255
...
10.255.255.0/24	10.255.255.1 - 10.255.255.254	10.255.255.255

Subnet within an Octet Boundary

- Refer to the table to see six ways to subnet a /24 network.

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of subnets	# of hosts
/25	255.255.255.128	nnnnnnnn.nnnnnnnn.nnnnnnnn. n hhhhh 11111111.11111111.11111111. 1 0000000	2	126
/26	255.255.255.192	nnnnnnnn.nnnnnnnn.nnnnnnnn. nn hhhhh 11111111.11111111.11111111. 11 000000	4	62
/27	255.255.255.224	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnn hhh 11111111.11111111.11111111. 111 00000	8	30
/28	255.255.255.240	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnnn hh 11111111.11111111.11111111. 1111 0000	16	14
/29	255.255.255.248	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnnnn hh 11111111.11111111.11111111. 11111 000	32	6
/30	255.255.255.252	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnnnnn hh 11111111.11111111.11111111. 111111 00	64	2

Create Subnets with a Slash 16 prefix



- The table highlights all the possible scenarios for subnetting a /16 prefix.

Prefix Length	Subnet Mask	Network Address (n = network, h = host)	# of subnets	# of hosts
/17	255.255. 128 .0	nnnnnnnn.nnnnnnnn. n hhhhh.hhhhhh 11111111.11111111. 1 0000000.00000000	2	32766
/18	255.255. 192 .0	nnnnnnnn.nnnnnnnn. nn hhhhh.hhhhhh 11111111.11111111. 11 000000.00000000	4	16382
/19	255.255. 224 .0	nnnnnnnn.nnnnnnnn. nnn hhh.hhhhhh 11111111.11111111. 111 00000.00000000	8	8190
/20	255.255. 240 .0	nnnnnnnn.nnnnnnnn. nnnn hh.hhhhhh 11111111.11111111. 1111 0000.00000000	16	4094
/21	255.255. 248 .0	nnnnnnnn.nnnnnnnn. nnnnn hh.hhhhhh 11111111.11111111. 11111 000.00000000	32	2046
/22	255.255. 252 .0	nnnnnnnn.nnnnnnnn. nnnnn h.hhhhhh 11111111.11111111. 111111 00.00000000	64	1022
/23	255.255. 254 .0	nnnnnnnn.nnnnnnnn. nnnnnnn h.hhhhhh 11111111.11111111. 1111111 0.00000000	128	510
/24	255.255. 255 .0	nnnnnnnn.nnnnnnnn. nnnnnnn n.hhhhhh 11111111.11111111. 11111111 .00000000	256	254
/25	255.255. 255.128	nnnnnnnn.nnnnnnnn. nnnnnnn n. n hhhhh 11111111.11111111. 11111111 . 1 0000000	512	126
/26	255.255. 255.192	nnnnnnnn.nnnnnnnn. nnnnnnn n. nn hhhhh 11111111.11111111. 11111111 . 11 0000000	1024	62
/27	255.255. 255.224	nnnnnnnn.nnnnnnnn. nnnnnnn n. nnn hhh 11111111.11111111. 11111111 . 111 000000	2048	30
/28	255.255. 255.240	nnnnnnnn.nnnnnnnn. nnnnnnn n. nnnn hhh 11111111.11111111. 11111111 . 1111 00000	4096	14
/29	255.255. 255.248	nnnnnnnn.nnnnnnnn. nnnnnnn n. nnnnn hh 11111111.11111111. 11111111 . 11111 00000	8192	6
/30	255.255. 255.252	nnnnnnnn.nnnnnnnn. nnnnnnn n. nnnnn hh 11111111.11111111. 11111111 . 111111 0000	16384	2

Create 100 Subnets with a Slash 16 prefix

Consider a large enterprise that requires at least 100 subnets and has chosen the private address 172.16.0.0/16 as its internal network address.

- The figure displays the number of subnets that can be created when borrowing bits from the third octet and the fourth octet.
- Notice there are now up to 14 host bits that can be borrowed (i.e., last two bits cannot be borrowed).

To satisfy the requirement of 100 subnets for the enterprise, 7 bits (i.e., $2^7 = 128$ subnets) would need to be borrowed (for a total of 128 subnets).

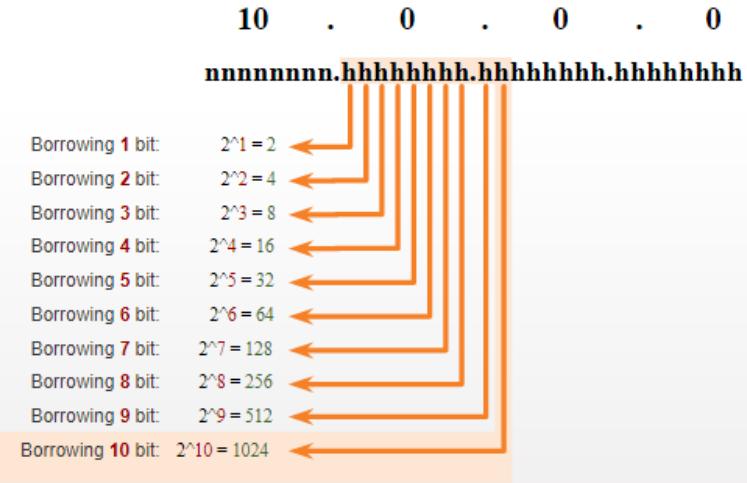


Create 1000 Subnets with a Slash 8 prefix

Consider a small ISP that requires 1000 subnets for its clients using network address 10.0.0.0/8 which means there are 8 bits in the network portion and 24 host bits available to borrow toward subnetting.

- The figure displays the number of subnets that can be created when borrowing bits from the second and third.
- Notice there are now up to 22 host bits that can be borrowed (i.e., last two bits cannot be borrowed).

To satisfy the requirement of 1000 subnets for the enterprise, 10 bits (i.e., $2^{10}=1024$ subnets) would need to be borrowed (for a total of 128 subnets)



Minimize Unused Host IPv4

Addresses and Maximize Subnets

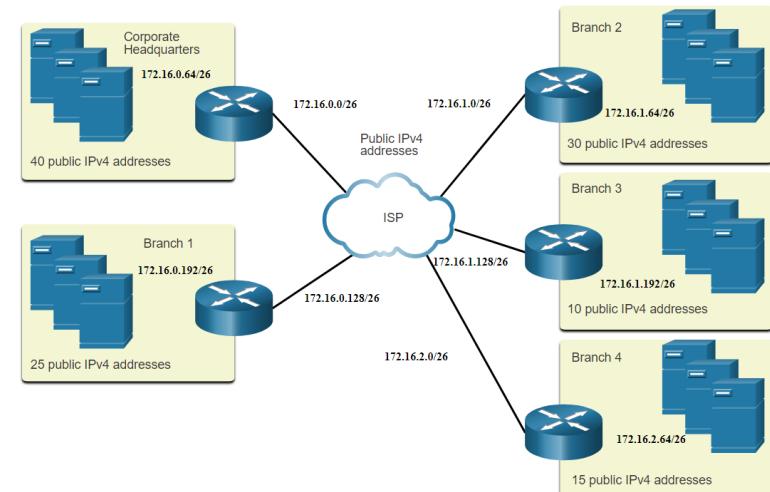
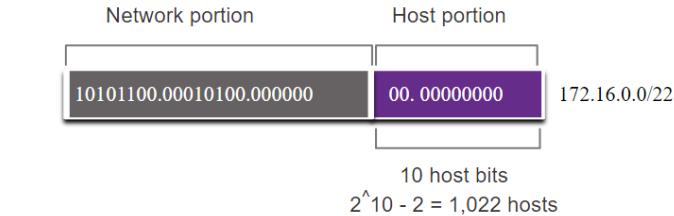
There are two considerations when planning subnets:

- The number of host addresses required for each network
- The number of individual subnets needed

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of subnets	# of hosts
/25	255.255.255.128	nnnnnnnn.nnnnnnnn.nnnnnnnn. n hhhhhhh 11111111.11111111.11111111. 1 0000000	2	126
/26	255.255.255.192	nnnnnnnn.nnnnnnnn.nnnnnnnn. nn hhhhhhh 11111111.11111111.11111111. 11 000000	4	62
/27	255.255.255.224	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnn hhhhhhh 11111111.11111111.11111111. 111 00000	8	30
/28	255.255.255.240	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnnn hhh 11111111.11111111.11111111. 1111 0000	16	14
/29	255.255.255.248	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnnnn hhh 11111111.11111111.11111111. 11111 000	32	6
/30	255.255.255.252	nnnnnnnn.nnnnnnnn.nnnnnnnn. nnnnnn hh 11111111.11111111.11111111. 111111 00	64	2

Example: Efficient IPv4 Subnetting

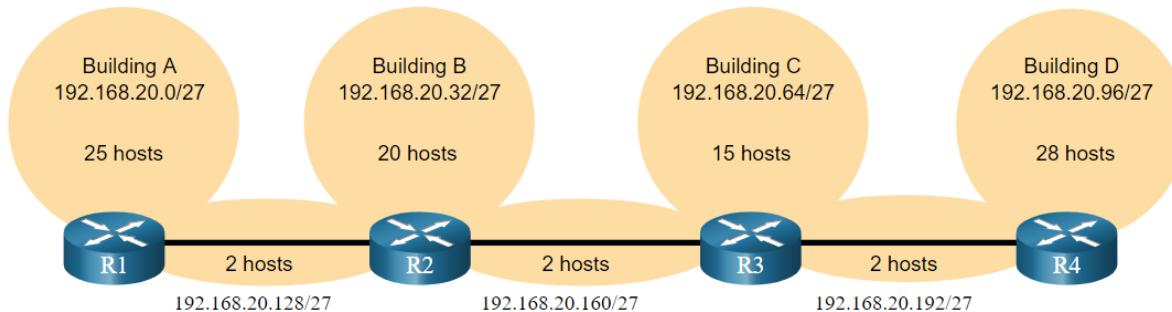
- In this example, corporate headquarters has been allocated a public network address of 172.16.0.0/22 (10 host bits) by its ISP providing 1,022 host addresses.
- There are five sites and therefore five internet connections which means the organization requires 10 subnets with the largest subnet requires 40 addresses.
- It allocated 10 subnets with a /26 (i.e., 255.255.255.192) subnet mask.



IPv4 Address Conservation

Given the topology, 7 subnets are required (i.e, four LANs and three WAN links) and the largest number of host is in Building D with 28 hosts.

- A /27 mask would provide 8 subnets of 30 host IP addresses and therefore support this topology.



IPv4 Address Conservation (Cont.)

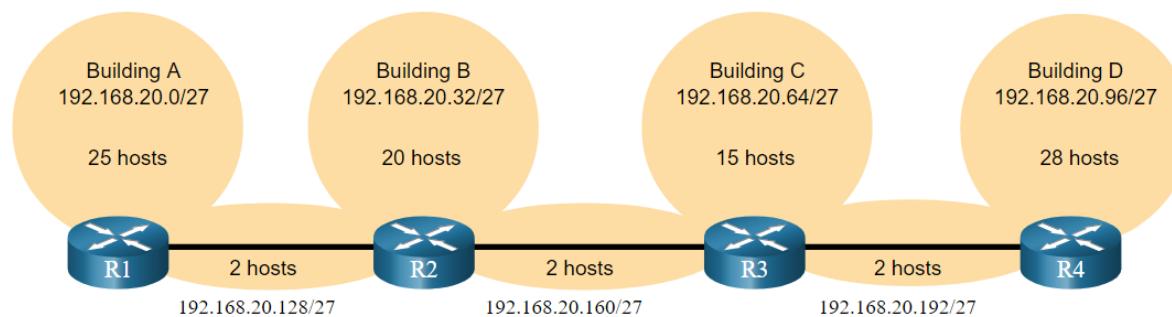
However, the point-to-point WAN links only require two addresses and therefore waste 28 addresses each for a total of 84 unused addresses.



Host portion
 $2^5 - 2 = 30$ host IP addresses per subnet

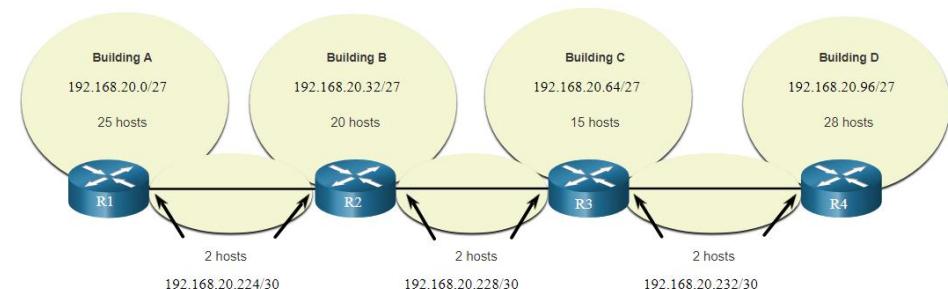
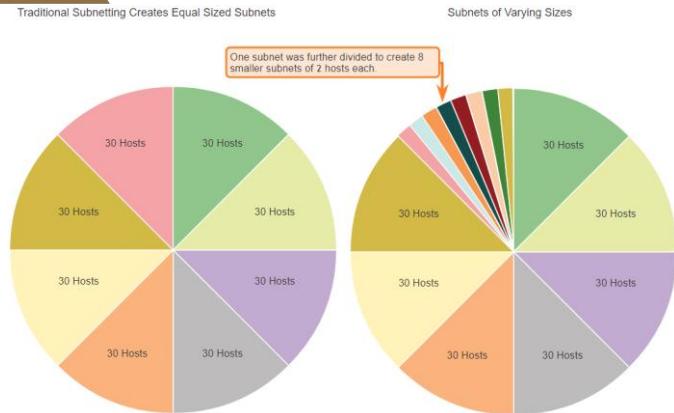
$30 - 2 = 28$
Each WAN subnet wastes 28 addresses

$28 \times 3 = 84$
84 addresses are unused



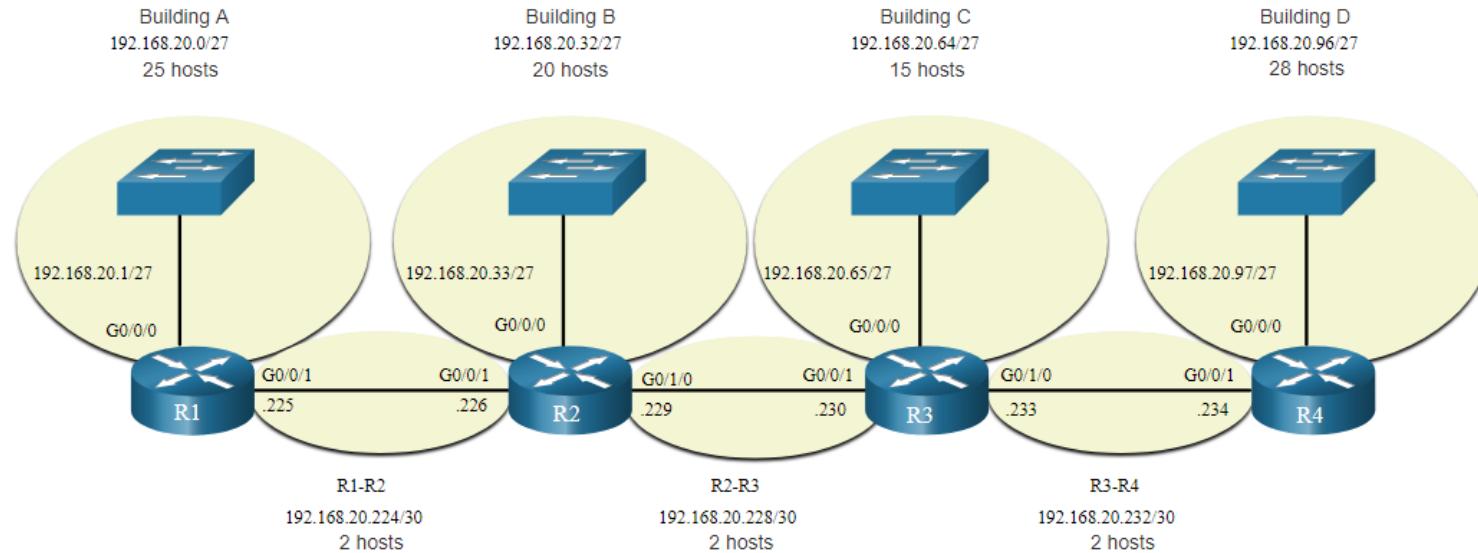
- Applying a traditional subnetting scheme to this scenario is not very efficient and is wasteful.
- VLSM was developed to avoid wasting addresses by enabling us to subnet a subnet.

- The left side displays the traditional subnetting scheme (i.e., the same subnet mask) while the right side illustrates how VLSM can be used to subnet a subnet and divided the last subnet into eight /30 subnets.
- When using VLSM, always begin by satisfying the host requirements of the largest subnet and continue subnetting until the host requirements of the smallest subnet are satisfied.
- The resulting topology with VLSM applied.



VLSM Topology Address Assignment

- Using VLSM subnets, the LAN and inter-router networks can be addressed without unnecessary waste as shown in the logical topology diagram.



Basic Router Configuration

Basic Router Configuration Steps

- Configure the device name.
- Secure privileged EXEC mode.
- Secure user EXEC mode.
- Secure remote Telnet / SSH access.
- Encrypt all plaintext passwords.
- Provide legal notification and save the configuration.

```
Router(config)# hostname hostname
```

```
Router(config)# enable secret password
```

```
Router(config)# line console 0
Router(config-line)# password password
Router(config-line)# login
```

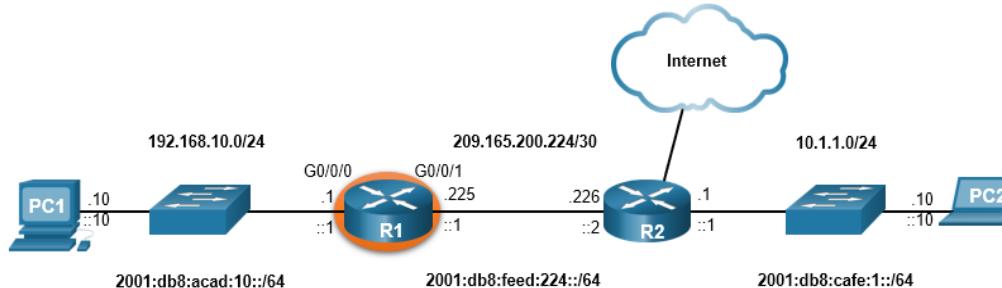
```
Router(config)# line vty 0 4
Router(config-line)# password password
Router(config-line)# login
Router(config-line)# transport input {ssh | telnet}
```

```
Router(config)# service password encryption
```

```
Router(config)# banner motd # message #
Router(config)# end
Router# copy running-config startup-config
```

Configure Router Interfaces Exam

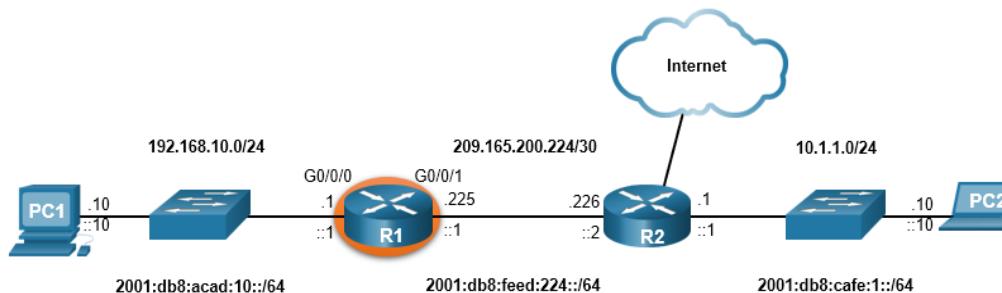
The commands to configure interface G0/0/0 on R1 are shown here:



```
R1(config)# interface gigabitEthernet 0/0/0
R1(config-if)# description Link to LAN
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:10::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
*Aug  1 01:43:53.435: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/0, changed state to down
*Aug  1 01:43:56.447: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/0, changed state to up
*Aug  1 01:43:57.447: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0,
changed state to up
```

Configure Router Interfaces Example (

The commands to configure interface G0/0/1 on R1 are shown here:



```
R1(config)# interface gigabitEthernet 0/0/1
R1(config-if)# description Link to R2
R1(config-if)# ip address 209.165.200.225 255.255.255.252
R1(config-if)# ipv6 address 2001:db8:feed:224::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
*Aug  1 01:46:29.170: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to down
*Aug  1 01:46:32.171: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to up
*Aug  1 01:46:33.171: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/1,
changed state to up
```

Verify Interface Configuration

To verify interface configuration use the **show ip interface brief** command shown here:

```
R1# show ip interface brief
Interface          IP-Address      OK? Method Status          Protocol
GigabitEthernet0/0/0 192.168.10.1   YES manual up           up
GigabitEthernet0/0/1 209.165.200.225 YES manual up           up
Vlan1              unassigned     YES unset administratively down down
```

Configure Verification Commands

The table summarizes show commands used to verify interface configuration.

Commands	Description
show ip interface brief show ipv6 interface brief	Displays all interfaces, their IP addresses, and their current status.
show ip route show ipv6 route	Displays the contents of the IP routing tables stored in RAM.
show interfaces	Displays statistics for all interfaces on the device. Only displays the IPv4 addressing information.
show ip interfaces	Displays the IPv4 statistics for all interfaces on a router.
show ipv6 interfaces	Displays the IPv6 statistics for all interfaces on a router.

Configure Verification Commands (Cont.)

View status of all interfaces with the **show ip interface brief** and **show ipv6 interface brief** commands, shown here:

```
R1# show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0/0 192.168.10.1   YES manual up        up
GigabitEthernet0/0/1 209.165.200.225 YES manual up        up
Vlan1              unassigned     YES unset administratively down down
R1#
```

```
R1# show ipv6 interface brief
GigabitEthernet0/0/0      [up/up]
  FE80::201:C9FF:FE89:4501
  2001:DB8:ACAD:10::1
GigabitEthernet0/0/1      [up/up]
  FE80::201:C9FF:FE89:4502
  2001:DB8:FEED:224::1
Vlan1                  [administratively down/down]
  unassigned
R1#
```

Configure Verification Commands

Display the contents of the IP routing tables with the **show ip route** command as shown here:

```
R1# show ip route
< output omitted>
Gateway of last resort is not set
    192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, GigabitEthernet0/0/0
L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/0
    209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.200.224/30 is directly connected, GigabitEthernet0/0/1
L        209.165.200.225/32 is directly connected, GigabitEthernet0/0/1
R1#
```

Configure Verification Commands

Display statistics for all interfaces with the **show interfaces** command, as shown here:

```
R1# show interfaces gig0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Hardware is ISR4321-2x1GE, address is a0e0.af0d.e140 (bia a0e0.af0d.e140)
  Description: Link to LAN
  Internet address is 192.168.10.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full Duplex, 100Mbps, link type is auto, media type is RJ45
  output flow-control is off, input flow-control is off
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:35, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (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
    1180 packets input, 109486 bytes, 0 no buffer
    Received 84 broadcasts (0 IP multicasts)
      0 runts, 0 giants, 0 throttles

<output omitted>

R1#
```

ICMP

ICMPv4 Messages



- Internet Control Message Protocol (ICMP) provides feedback about issues related to the processing of IP packets under certain conditions.
- ICMPv4 is the messaging protocol for IPv4. ICMPv6 is the messaging protocol for IPv6 and includes additional functionality.
- The ICMP messages common to both ICMPv4 and ICMPv6 include:
 - Host reachability
 - Destination or Service Unreachable
 - Time exceeded

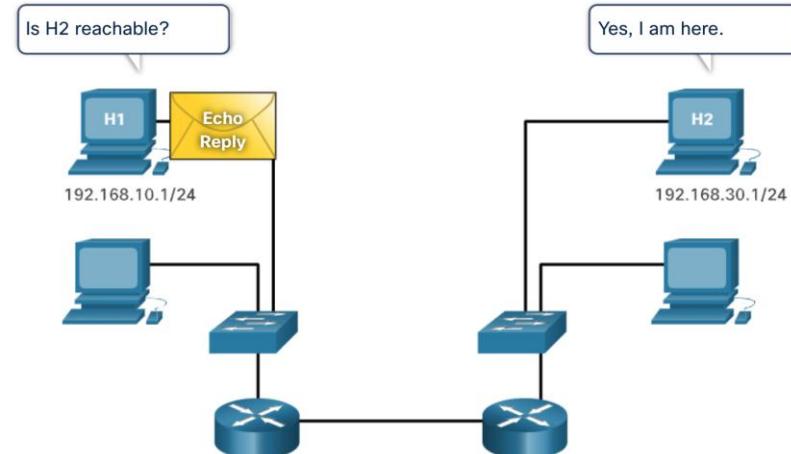
Note: ICMPv4 messages are not required and are often not allowed within a network for security reasons.

Host Reachability

ICMP Echo Message can be used to test the reachability of a host on an IP network.

In the example:

- The local host sends an ICMP Echo Request to a host.
- If the host is available, the destination host responds with an Echo Reply.



Destination or Service Unreachable



- An ICMP Destination Unreachable message can be used to notify the source that a destination or service is unreachable.
- The ICMP message will include a code indicating why the packet could not be delivered.

A few Destination Unreachable codes for ICMPv4 are as follows:

- 0 - Net unreachable
- 1 - Host unreachable
- 2 - Protocol unreachable
- 3 - Port unreachable

A few Destination Unreachable codes for ICMPv6 are as follows:

- 0 - No route to destination
- 1 - Communication with the destination is administratively prohibited (e.g., firewall)
- 2 – Beyond scope of the source address
- 3 - Address unreachable
- 4 - Port unreachable

Note: ICMPv6 has similar but slightly different codes for Destination Unreachable messages.

Time Exceeded



- When the Time to Live (TTL) field in a packet is decremented to 0, an ICMPv4 Time Exceeded message will be sent to the source host.
- ICMPv6 also sends a Time Exceeded message. Instead of the IPv4 TTL field, ICMPv6 uses the IPv6 Hop Limit field to determine if the packet has expired.

```
Pinging 8.8.8.8 with 32 bytes of data:  
Reply from 192.168.1.1: TTL expired in transit.  
  
Ping statistics for 8.8.8.8:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Note: Time Exceeded messages are used by the **traceroute** tool.

Traceroute – Test the Path



- Traceroute (**tracert**) is a utility that is used to test the path between two hosts and provide a list of hops that were successfully reached along that path.
- Traceroute provides round-trip time for each hop along the path and indicates if a hop fails to respond. An asterisk (*) is used to indicate a lost or unrepplied packet.
- This information can be used to locate a problematic router in the path or may indicate that the router is configured not to reply.

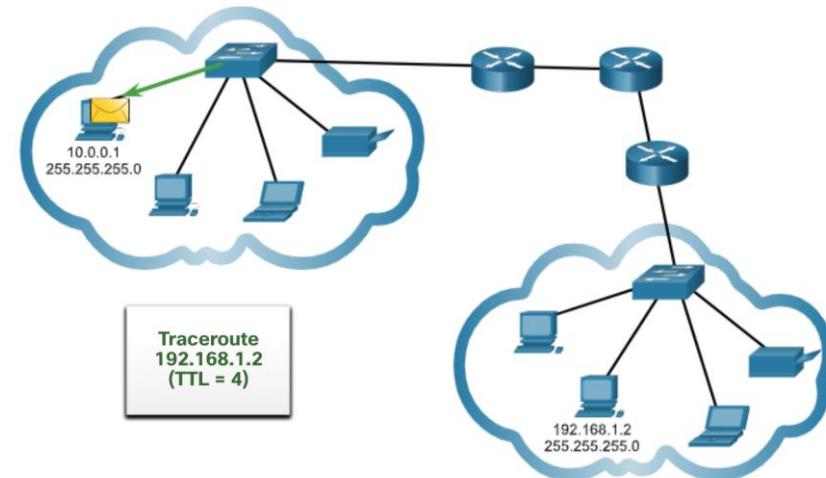
```
R1#traceroute 192.168.40.2
Type escape sequence to abort.
Tracing the route to 192.168.40.2
```

1	192.168.10.2	1 msec	0 msec	0 msec	
2	192.168.20.2	2 msec	1 msec	0 msec	
3	192.168.30.2	1 msec	0 msec	0 msec	
4	192.168.40.2	0 msec	0 msec	0 msec	

Note: Traceroute makes use of a function of the TTL field in IPv4 and the Hop Limit field in IPv6 in the Layer 3 headers, along with the ICMP Time Exceeded message.

Traceroute – Test the Path (Cont.)

- The first message sent from traceroute will have a TTL field value of 1. This causes the TTL to time out at the first router. This router then responds with a ICMPv4 Time Exceeded message.
- Traceroute then progressively increments the TTL field (2, 3, 4...) for each sequence of messages. This provides the trace with the address of each hop as the packets time out further down the path.
- The TTL field continues to be increased until the destination is reached, or it is incremented to a predefined maximum.



Steps for Configuring SSH

Example SSH Configuration

```
R1# conf t
R1(config)# ip domain-name span.com
R1(config)# crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.span.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)

R1(config)#
*Feb 16 21:18:41.977: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)# ip ssh version 2
R1(config)# username Bob algorithm-type scrypt secret cisco54321
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# end
R1#
```

Example Verification of SSH

```
R1# show crypto key mypubkey rsa
% Key pair was generated at: 21:18:41 UTC Feb 16 2015
Key name: R1.span.com
Key type: RSA KEYS
Storage Device: not specified
Usage: General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CF35DB
A58A1BDB F7C7E600 F189C2F3 2EC6E584 D923EE5B 71841D98 B5472A03 D19CD620
ED125825 5A58412B B7F29234 DE2A1809 6C421AC3 07F298E6 80BE149D 2A262E13
74888DAF CACF187 B11111AF A413E76F 6C157CDF DFEF0D82 2961B58C BE1CAD21
176E82B9 6D81F893 06E66C93 94E1C508 887462F6 90AC63CE 5E169845 C1020301 0001
% Key pair was generated at: 21:18:42 UTC Feb 16 2015
Key name: R1.span.com.server
Key type: RSA KEYS
Temporary key
Usage: Encryption Key
Key is not exportable.
Key Data:
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00AB914D 8172DFBE
DE57ACAA9 78844239 1F3B5942 3943AC0D F54E7746 3895CF54 606C3961 8A44FEB3
1A019F27 D9E71AAE FC73F423 A59CB8F5 50289272 3392CEBC 4C3CBD6D DB9233DE
9DD9DAD 79D56165 4293AA62 FD1CBAB2 7AB859DC 2890C795 ED020301 0001
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# crypto key zeroize rsa
% All keys will be removed.
% All router certs issued using these keys will also be removed.
Do you really want to remove these keys? [yes/no]: yes
R1(config)#

```

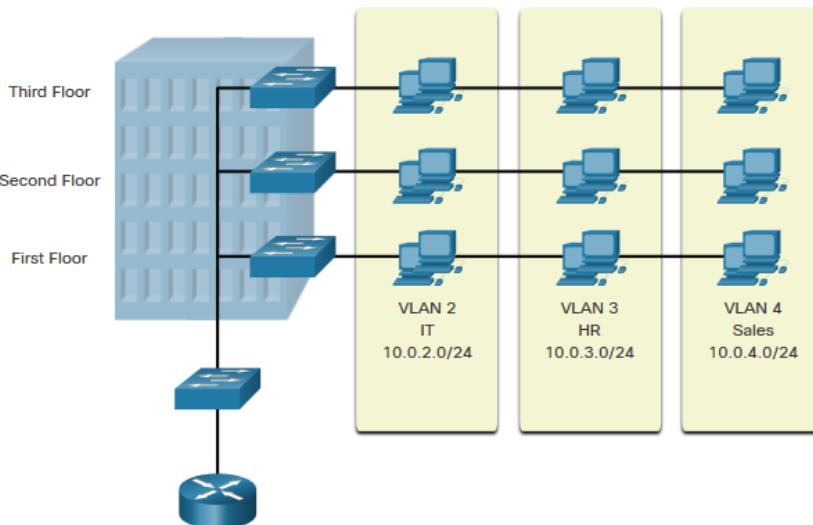
Password recovery exact steps for routers



1. Connect your console to the router (password recovery can't be done using telnet, it must be done using console connection)
2. Turnoff the router then turn it on again
3. During 60 seconds of router startup → press “control + break” buttons
4. You will enter the rommon mode
5. Change the configuration register to 0x2142 using this command :
 1. Confreg 0x2142
 2. Reset
6. After the router startup again, you will find that there is no configuration on it and you can access it easily without any passwords.
7. From privilege mode copy the startup config to running config
 1. Router# copy start run
 2. Router# conf t
 3. Router(config)# configuration-register 0x2102
 4. Router(config)# Exit
 5. Router# Copy run start

VLANs

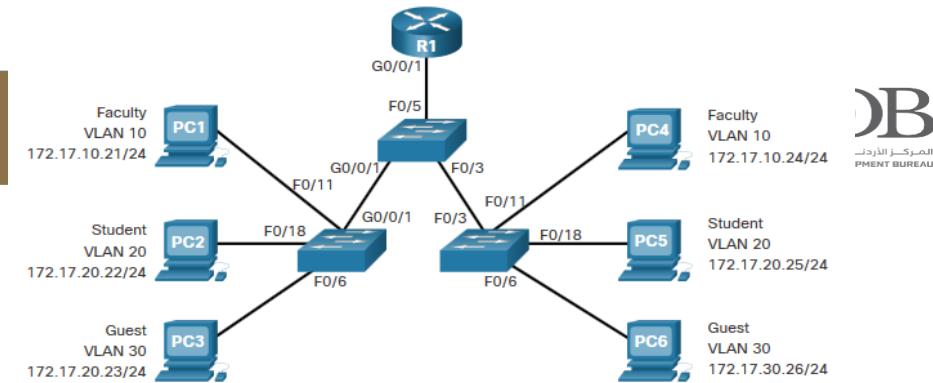
VLAN Definitions



VLANs are logical connections with other similar devices. Placing devices into various VLANs have the following characteristics:

- Provides segmentation of the various groups of devices on the same switches
- Provide organization that is more manageable
 - Broadcasts, multicasts and unicasts are isolated in the individual VLAN
 - Each VLAN will have its own unique range of IP addressing
 - Smaller broadcast domains

Benefits of a LAN Design



Benefits	Description
Smaller Broadcast Domains	Dividing the LAN reduces the number of broadcast domains
Improved Security	Only users in the same VLAN can communicate together
Improved IT Efficiency	VLANs can group devices with similar requirements, e.g. faculty vs. students
Reduced Cost	One switch can support multiple groups or VLANs
Better Performance	Small broadcast domains reduce traffic, improving bandwidth
Simpler	Similar groups will need similar applications and other network

Types of VLANs



Default VLAN

VLAN 1 is the following:

- The default VLAN
- The default Native VLAN
- The default Management VLAN
- Cannot be deleted or renamed

Note: While we cannot delete VLAN1 Cisco will recommend that we assign these default features to other VLANs

```
Switch# show vlan brief
VLAN Name          Status    Ports
---- -----
1     default      active   Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12
                           Fa0/13, Fa0/14, Fa0/15, Fa0/16
                           Fa0/17, Fa0/18, Fa0/19, Fa0/20
                           Fa0/21, Fa0/22, Fa0/23, Fa0/24
                           Gi0/1, Gi0/2
1002 fddi-default    act/unsup
1003 token-ring-default  act/unsup
1004 fddinet-default   act/unsup
1005 trnet-default     act/unsup
```

Types of VLANs (Cont.)



Data VLAN

- Dedicated to user-generated traffic (email and web traffic).
- VLAN 1 is the default data VLAN because all interfaces are assigned to this VLAN.

Native VLAN

- This is used for trunk links only.
- All frames are tagged on an 802.1Q trunk link except for those on the native VLAN.

Management VLAN

- This is used for SSH/Telnet VTY traffic and should not be carried with end user traffic.
- Typically, the VLAN that is the SVI for the Layer 2 switch.

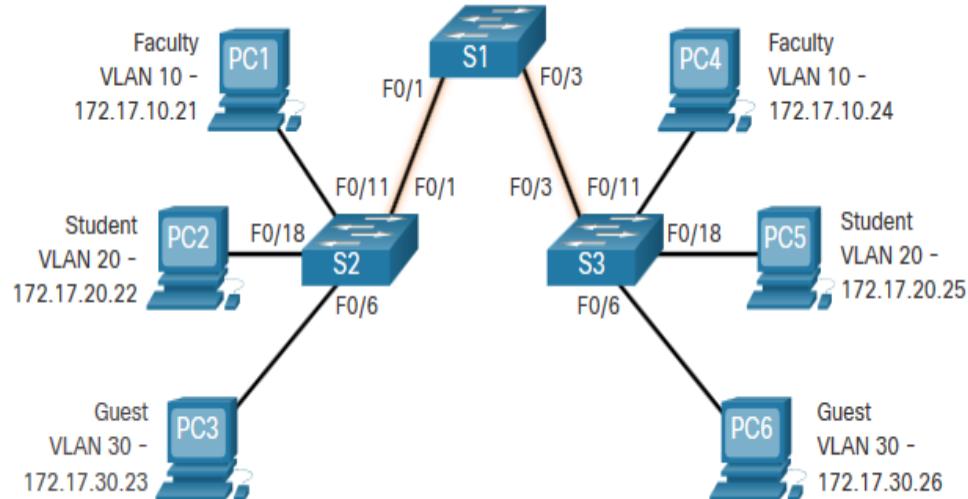
Defining VLAN Trunks



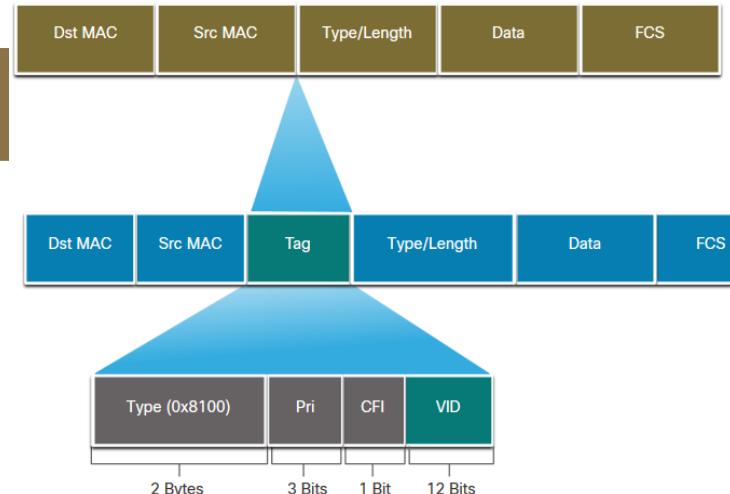
A trunk is a point-to-point link between two network devices.

Cisco trunk functions:

- Allow more than one VLAN
- Extend the VLAN across the entire network
- By default, supports all VLANs
- Supports 802.1Q trunking



VLAN Identification with a Tag



- The IEEE 802.1Q header is 4 Bytes
- When the tag is created the FCS must be recalculated.
- When sent to end devices, this tag must be removed and the FCS recalculated back to its original number.

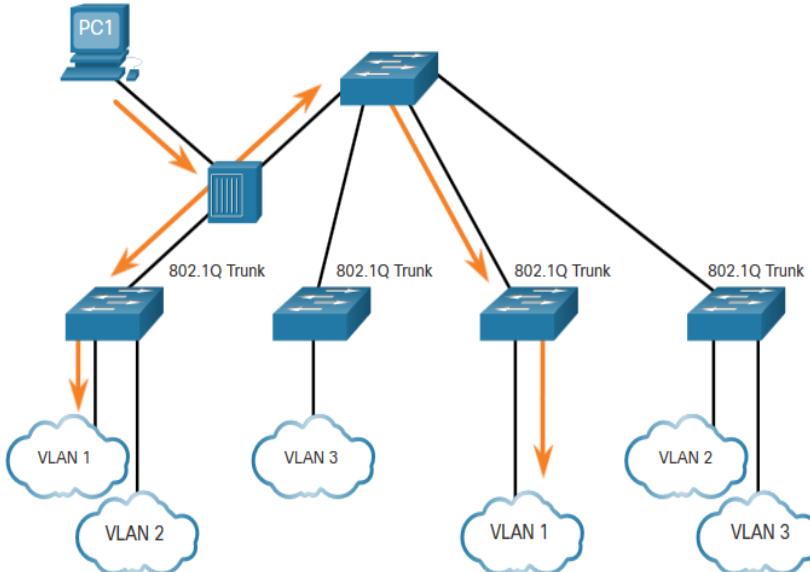
802.1Q VLAN Tag Field	Function
Type	<ul style="list-style-type: none"> 2-Byte field with hexadecimal 0x8100 This is referred to as Tag Protocol ID (TPID)
User Priority	<ul style="list-style-type: none"> 3-bit value that supports
Canonical Format Identifier (CFI)	<ul style="list-style-type: none"> 1-bit value that can support token ring frames on Ethernet
VLAN ID (VID)	<ul style="list-style-type: none"> 12-bit VLAN identifier that can support up to 4096 VLANs

Native VLANs and 802.1Q Tagging



802.1Q trunk basics:

- Tagging is typically done on all VLANs.
- The use of a native VLAN was designed for legacy use, like the hub in the example.
- Unless changed, VLAN1 is the native VLAN.
- Both ends of a trunk link must be configured with the same native VLAN.
- Each trunk is configured separately, so it is possible to have a different native VLANs on separate trunks.



VLAN Ranges on Catalyst Switches

Catalyst switches 2960 and 3650 support over 4000 VLANs.

Switch# show vlan brief			
VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

Normal Range VLAN 1 – 1005

Used in Small to Medium sized businesses

1002 – 1005 are reserved for legacy VLANs

1, 1002 – 1005 are auto created and cannot be deleted

Stored in the `vlan.dat` file in flash

VTP can synchronize between switches

Extended Range VLAN 1006 - 4095

Used by Service Providers

Are in Running-Config

Supports fewer VLAN features

Requires VTP configurations

VLAN Creation Command

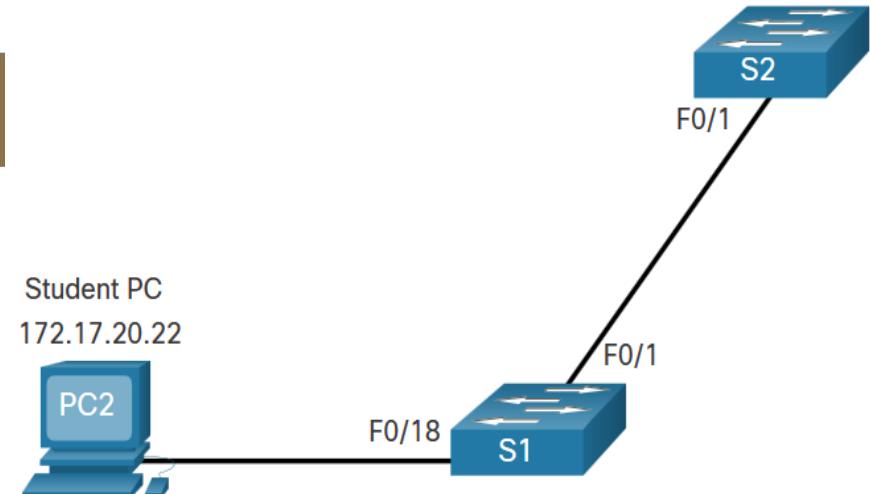


VLAN details are stored in the `vlan.dat` file. You create VLANs in the global configuration mode.

Task	IOS Command
Enter global configuration mode.	Switch# configure terminal
Create a VLAN with a valid ID number.	Switch(config)# vlan vlan-id
Specify a unique name to identify the VLAN.	Switch(config-vlan)# name vlan-name
Return to the privileged EXEC mode.	Switch(config-vlan)# end
Enter global configuration mode.	Switch# configure terminal

VLAN Creation Example

- If the Student PC is going to be in VLAN 20, we will create the VLAN first and then name it.
- If you do not name it, the Cisco IOS will give it a default name of vlan and the four digit number of the VLAN. E.g. vlan0020 for VLAN 20.

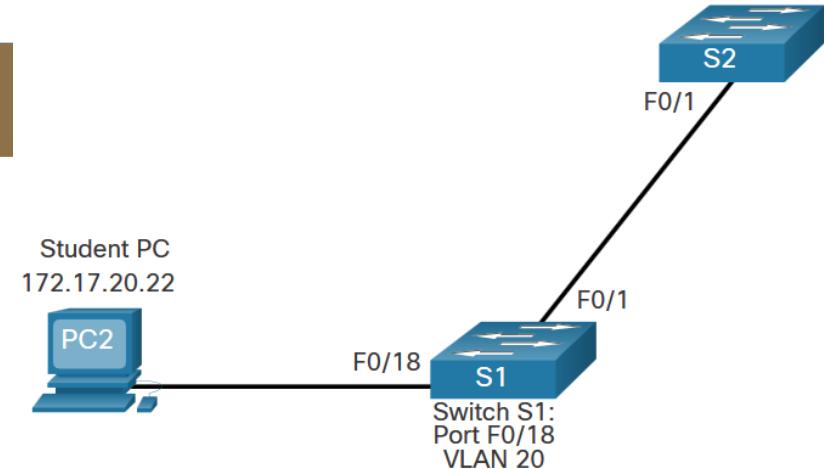


Prompt	Command
S1#	Configure terminal
S1(config)#	vlan 20
S1(config-vlan)#	name student
S1(config-vlan)#	end

VLAN Port Assignment

We can assign the VLAN to the port interface.

- Once the device is assigned the VLAN, then the end device will need the IP address information for that VLAN
- Here, Student PC receives 172.17.20.22



Prompt	Command
S1#	Configure terminal
S1(config)#	Interface fa0/18
S1(config-if)#	Switchport mode access
S1(config-if)#	Switchport access vlan 20
S1(config-if)#	end

Verify VLAN Information

Use the **show vlan** command. The complete syntax is:

show vlan [brief | id *vlan-id* | name *vlan-name* | summary]

```
S1# show vlan summary
Number of existing VLANs : 7
Number of existing VTP VLANs : 7
Number of existing extended VLANs : 0
```

ACADEMY

```
S1# show interface vlan 20
Vlan20 is up, line protocol is up
  Hardware is EtherSVI, address is 001f.6ddb.3ec1 (bia 001f.6ddb.3ec1)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set

(Output omitted)
```

Task	Command Option
Display VLAN name, status, and its ports one VLAN per line.	brief
Display information about the identified VLAN ID number.	id <i>vlan-id</i>
Display information about the identified VLAN name. The <i>vlan-name</i> is an ASCII string from 1 to 32 characters.	name <i>vlan-name</i>
Display VLAN summary information.	summary

VLAN Configuration

Change VLAN Port Membership

There are a number of ways to change VLAN membership:

- re-enter **switchport access vlan *vlan-id*** command
- use the **no switchport access vlan** to place interface back in VLAN 1

Use the **show vlan brief** or the **show interface fa0/18 switchport** commands to verify the correct VLAN association.

```
S1(config)# interface fa0/18
S1(config-if)# no switchport access vlan
S1(config-if)# end
S1#
S1# show vlan brief
VLAN Name          Status    Ports
----  -----
1    default        active   Fa0/1, Fa0/2, Fa0/3, Fa0/4
                               Fa0/5, Fa0/6, Fa0/7, Fa0/8
                               Fa0/9, Fa0/10, Fa0/11, Fa0/12
                               Fa0/13, Fa0/14, Fa0/15, Fa0/16
                               Fa0/17, Fa0/18, Fa0/19, Fa0/20
                               Fa0/21, Fa0/22, Fa0/23, Fa0/24
                               Gi0/1, Gi0/2
20   student         active
1002 fddi-default    act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default  act/unsup
1005 trnet-default    act/unsup
```

```
S1# show interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
```

Delete VLANs



Delete VLANs with the **no vlan *vlan-id*** command.

Caution: Before deleting a VLAN, reassign all member ports to a different VLAN.

- Delete all VLANs with the **delete flash:vlan.dat** or **delete vlan.dat** commands.
- Reload the switch when deleting all VLANs.

Note: To restore to factory default – unplug all data cables, erase the startup-configuration and delete the **vlan.dat** file, then reload the device.

VLAN Trunks

Trunk Configuration Example

The subnets associated with each VLAN are:

VLAN 10 - Faculty/Staff - 172.17.10.0/24

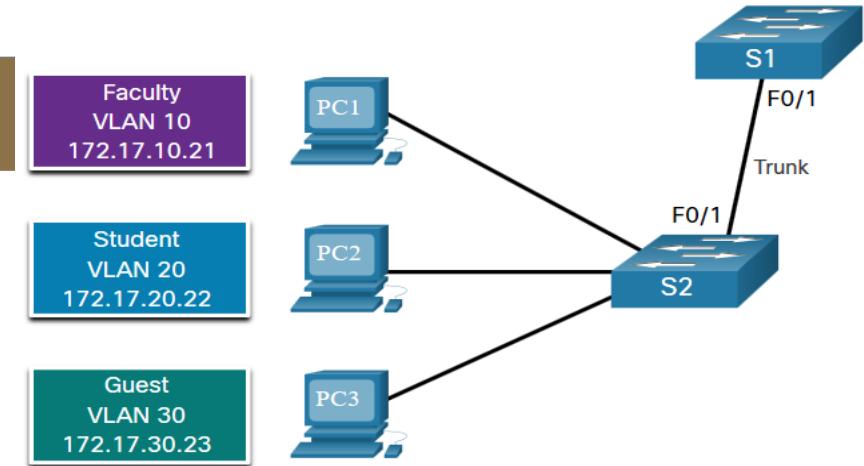
VLAN 20 - Students - 172.17.20.0/24

VLAN 30 - Guests - 172.17.30.0/24

VLAN 99 - Native - 172.17.99.0/24

F0/1 port on S1 is configured as a trunk port.

Note: This assumes a 2960 switch using 802.1q tagging. Layer 3 switches require the encapsulation to be configured before the trunk mode.



Prompt	Command
S1(config)#	Interface fa0/1
S1(config-if)#	Switchport mode trunk
S1(config-if)#	Switchport trunk native vlan 99
S1(config-if)#	Switchport trunk allowed vlan 10,20,30,99
S1(config-if)#	end

Introduction to DTP



Dynamic Trunking Protocol (DTP) is a proprietary Cisco protocol.

DTP characteristics are as follows:

- On by default on Catalyst 2960 and 2950 switches
- Dynamic-auto is default on the 2960 and 2950 switches
- May be turned off with the nonegotiate command
- May be turned back on by setting the interface to dynamic-auto
- Setting a switch to a static trunk or static access will avoid negotiation issues with the **switchport mode trunk** or the **switchport mode access** commands.

```
S1(config-if)# switchport mode trunk  
S1(config-if)# switchport nonegotiate
```

```
S1(config-if)# switchport mode dynamic auto
```

Negotiated Interface Modes



The **switchport mode** command has additional options.

Use the **switchport nonegotiate** interface configuration command to stop DTP negotiation.

Option	Description
access	Permanent access mode and negotiates to convert the neighboring link into an access link
dynamic auto	Will becomes a trunk interface if the neighboring interface is set to trunk or desirable mode
dynamic desirable	Actively seeks to become a trunk by negotiating with other auto or desirable interfaces
trunk	Permanent trunking mode and negotiates to convert the neighboring link into a trunk link

Results of a DTP Configuration



	Dynamic Auto	Dynamic Desirable	Trunk	Access
Dynamic Auto	Access	Trunk	Trunk	Access
Dynamic Desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Limited connectivity
Access	Access	Access	Limited connectivity	Access

Verify DTP Mode



The default DTP configuration is dependent on the Cisco IOS version and platform.

Use the **show dtp interface** command to determine the current DTP mode.

Best practice recommends that the interfaces be set to access or trunk and to turnoff DTP

```
S1# show dtp interface fa0/1
DTP information for FastEthernet0/1:
TOS/TAS/TNS: ACCESS/AUTO/ACCESS
TOT/TAT/TNT: NATIVE/NEGOTIATE/NATIVE
Neighbor address 1: C80084AEF101
Neighbor address 2: 000000000000
Hello timer expiration (sec/state): 11/RUNNING
Access timer expiration (sec/state): never/STOPPED
Negotiation timer expiration (sec/state): never/STOPPED
Multidrop timer expiration (sec/state): never/STOPPED
FSM state: S2:ACCESS
# times multi & trunk 0
Enabled: yes
In STP: no
```

Inter-VLAN Routing

What is Inter-VLAN Routing?

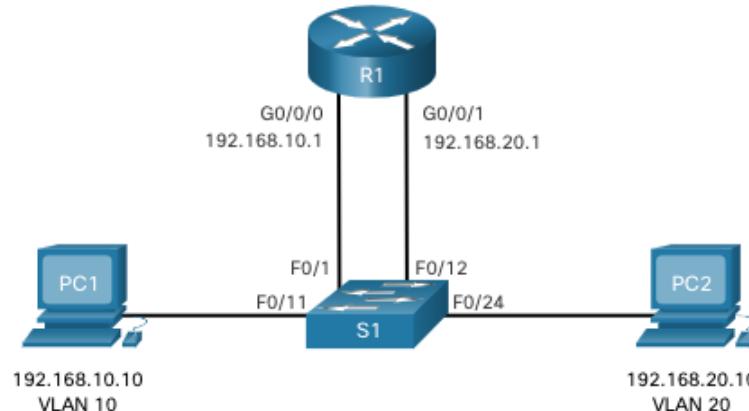
Inter-VLAN routing is the process of forwarding network traffic from one VLAN to another VLAN.

There are three inter-VLAN routing options:

- **Legacy Inter-VLAN routing** - This is a legacy solution. It does not scale well.
- **Router-on-a-Stick** - This is an acceptable solution for a small to medium-sized network.
- **Layer 3 switch using switched virtual interfaces (SVIs)** - This is the most scalable solution for medium to large organizations.

Legacy Inter-VLAN Routing

- The first inter-VLAN routing solution relied on using a router with multiple Ethernet interfaces. Each router interface was connected to a switch port in different VLANs. The router interfaces served as the default gateways to the local hosts on the VLAN subnet.
- Legacy inter-VLAN routing using physical interfaces works, but it has a significant limitation. It is not reasonably scalable because routers have a limited number of physical interfaces. Requiring one physical router interface per VLAN quickly exhausts the physical interface capacity of a router.
- Note:** This method of inter-VLAN routing is no longer implemented in switched networks and is included for explanation purposes only.



Router-on-a-Stick Inter-VLAN Rou

The 'router-on-a-stick' inter-VLAN routing method overcomes the limitation of the legacy inter-VLAN routing method. It only requires one physical Ethernet interface to route traffic between multiple VLANs on a network.

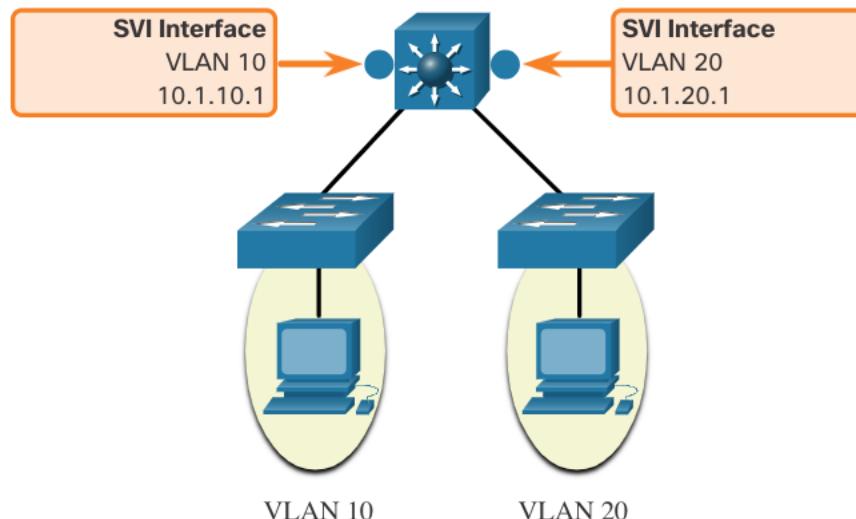
- A Cisco IOS router Ethernet interface is configured as an 802.1Q trunk and connected to a trunk port on a Layer 2 switch. Specifically, the router interface is configured using subinterfaces to identify routable VLANs.
- The configured subinterfaces are software-based virtual interfaces. Each is associated with a single physical Ethernet interface. Subinterfaces are configured in software on a router. Each subinterface is independently configured with an IP address and VLAN assignment. Subinterfaces are configured for different subnets that correspond to their VLAN assignment. This facilitates logical routing.
- When VLAN-tagged traffic enters the router interface, it is forwarded to the VLAN subinterface. After a routing decision is made based on the destination IP network address, the router determines the exit interface for the traffic. If the exit interface is configured as an 802.1q subinterface, the data frames are VLAN-tagged with the new VLAN and sent back out the physical interface

Note: The router-on-a-stick method of inter-VLAN routing does not scale beyond 50 VLANs.

Inter-VLAN Routing on a Layer 3 Switch

The modern method of performing inter-VLAN routing is to use Layer 3 switches and switched virtual interfaces (SVI). An SVI is a virtual interface that is configured on a Layer 3 switch, as shown in the figure.

Note: A Layer 3 switch is also called a multilayer switch as it operates at Layer 2 and Layer 3. However, in this course we use the term Layer 3 switch.



Inter-VLAN Routing on a Layer 3 Switch

Inter-VLAN SVIs are created the same way that the management VLAN interface is configured. The SVI is created for a VLAN that exists on the switch. Although virtual, the SVI performs the same functions for the VLAN as a router interface would. Specifically, it provides Layer 3 processing for packets that are sent to or from all switch ports associated with that VLAN.

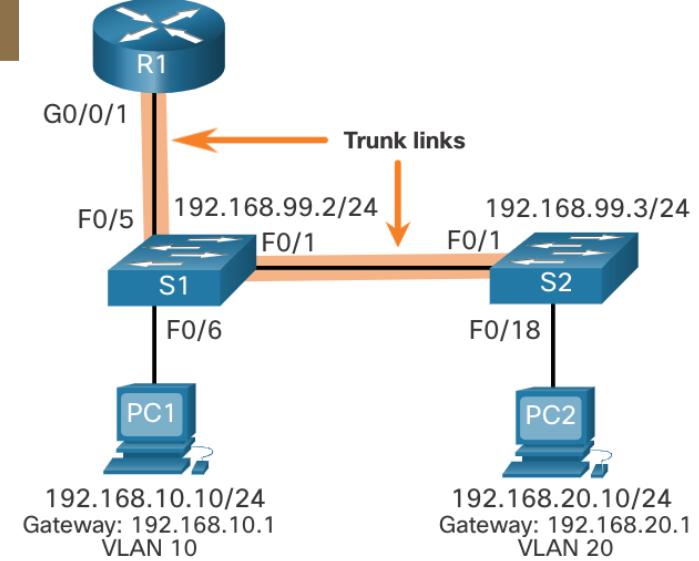
The following are advantages of using Layer 3 switches for inter-VLAN routing:

- They are much faster than router-on-a-stick because everything is hardware switched and routed.
- There is no need for external links from the switch to the router for routing.
- They are not limited to one link because Layer 2 EtherChannels can be used as trunk links between the switches to increase bandwidth.
- Latency is much lower because data does not need to leave the switch in order to be routed to a different network.
- They more commonly deployed in a campus LAN than routers.
- The only disadvantage is that Layer 3 switches are more expensive.

Router-on-a-Stick Scenario



- In the figure, the R1 GigabitEthernet 0/0/1 interface is connected to the S1 FastEthernet 0/5 port. The S1 FastEthernet 0/1 port is connected to the S2 FastEthernet 0/1 port. These are trunk links that are required to forward traffic within and between VLANs.
- To route between VLANs, the R1 GigabitEthernet 0/0/1 interface is logically divided into three subinterfaces, as shown in the table. The table also shows the three VLANs that will be configured on the switches.
- Assume that R1, S1, and S2 have initial basic configurations. Currently, PC1 and PC2 cannot **ping** each other because they are on separate networks. Only S1 and S2 can **ping** each other, but they but are unreachable by PC1 or PC2 because they are also on different networks.
- To enable devices to ping each other, the switches must be configured with VLANs and trunking, and the router must be configured for inter-VLAN routing.



Subinterface	VLAN	IP Address
G0/0/1.10	10	192.168.10.1/24
G0/0/1.20	20	192.168.20.1/24
G0/0/1.30	99	192.168.99.1/24

S1 VLAN and Trunking Configuration

Complete the following steps to configure S1 with VLANs and trunking:

- **Step 1.** Create and name the VLANs.
- **Step 2.** Create the management interface.
- **Step 3.** Configure access ports.
- **Step 4.** Configure trunking ports.

S2 VLAN and Trunking Configuration

The configuration for S2 is similar to S1.

```
S2(config)# vlan 10
S2(config-vlan)# name LAN10
S2(config-vlan)# exit
S2(config)# vlan 20
S2(config-vlan)# name LAN20
S2(config-vlan)# exit
S2(config)# vlan 99
S2(config-vlan)# name Management
S2(config-vlan)# exit
S2(config)#
S2(config)# interface vlan 99
S2(config-if)# ip add 192.168.99.3 255.255.255.0
S2(config-if)# no shut
S2(config-if)# exit
S2(config)# ip default-gateway 192.168.99.1
S2(config)# interface fa0/18
S2(config-if)# switchport mode access
S2(config-if)# switchport access vlan 20
S2(config-if)# no shut
S2(config-if)# exit
S2(config)# interface fa0/1
S2(config-if)# switchport mode trunk
S2(config-if)# no shut
S2(config-if)# exit
S2(config-if)# end
*Mar  1 00:23:52.137: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to up
```

R1 Subinterface Configuration

The router-on-a-stick method requires you to create a subinterface for each VLAN to be routed. A subinterface is created using the **interface *interface_id* *subinterface_id*** global configuration mode command. The subinterface syntax is the physical interface followed by a period and a subinterface number. Although not required, it is customary to match the subinterface number with the VLAN number.

Each subinterface is then configured with the following two commands:

- **encapsulation dot1q *vlan_id* [native]** - This command configures the subinterface to respond to 802.1Q encapsulated traffic from the specified *vlan-id*. The **native** keyword option is only appended to set the native VLAN to something other than VLAN 1.
- **ip address *ip-address* *subnet-mask*** - This command configures the IPv4 address of the subinterface. This address typically serves as the default gateway for the identified VLAN.

Repeat the process for each VLAN to be routed. Each router subinterface must be assigned an IP address on a unique subnet for routing to occur. When all subinterfaces have been created, enable the physical interface using the **no shutdown** interface configuration command. If the physical interface is disabled, all subinterfaces are disabled.

R1 Subinterface Configuration (Config Mode)

In the configuration, the R1 G0/0/1 subinterfaces are configured for VLANs 10, 20, and 99.

```
R1(config)# interface G0/0/1.10
R1(config-subif)# Description Default Gateway for VLAN 10
R1(config-subif)# encapsulation dot1Q 10
R1(config-subif)# ip add 192.168.10.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1.20
R1(config-subif)# Description Default Gateway for VLAN 20
R1(config-subif)# encapsulation dot1Q 20
R1(config-subif)# ip add 192.168.20.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1.99
R1(config-subif)# Description Default Gateway for VLAN 99
R1(config-subif)# encapsulation dot1Q 99
R1(config-subif)# ip add 192.168.99.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1
R1(config-if)# Description Trunk link to S1
R1(config-if)# no shut
R1(config-if)# end
R1#
*Sep 15 19:08:47.015: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to down
*Sep 15 19:08:50.071: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to up
*Sep 15 19:08:51.071: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/1, changed state to up
R1#
```

Verify Connectivity Between PC1 and PC2

The router-on-a-stick configuration is complete after the switch trunk and the router subinterfaces have been configured. The configuration can be verified from the hosts, router, and switch.

From a host, verify connectivity to a host in another VLAN using the **ping** command. It is a good idea to first verify the current host IP configuration using the **ipconfig** Windows host command.

Next, use **ping** to verify connectivity with PC2 and S1, as shown in the figure. The **ping** output successfully confirms inter-VLAN routing is operating.

```
C:\Users\PC1> ping 192.168.20.10
Pinging 192.168.20.10 with 32 bytes of data:
Reply from 192.168.20.10: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.20.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\PC1>
C:\Users\PC1> ping 192.168.99.2
Pinging 192.168.99.2 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.99.2: bytes=32 time=2ms TTL=254
Reply from 192.168.99.2: bytes=32 time=1ms TTL=254
Ping statistics for 192.168.99.2:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss).
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Users\PC1>
```

Router-on-a-Stick Inter-VLAN Routing

In addition to using **ping** between devices, the following **show** commands can be used to verify and troubleshoot the router-on-a-stick configuration.

- **show ip route**
- **show ip interface brief**
- **show interfaces**
- **show interfaces trunk**

Routing Concepts

Two Functions of a Router

When a router receives an IP packet on one interface, it determines which interface to use to forward the packet to the destination. This is known as routing. The interface that the router uses to forward the packet may be the final destination, or it may be a network connected to another router that is used to reach the destination network. Each network that a router connects to typically requires a separate interface, but this may not always be the case.

The primary functions of a router are to determine the best path to forward packets based on the information in its routing table, and to forward packets toward their destination.

Build the Routing Table

Directly Connected Networks: Added to the routing table when a local interface is configured with an IP address and subnet mask (prefix length) and is active (up and up).

Remote Networks: Networks that are not directly connected to the router. Routers learn about remote networks in two ways:

- **Static routes** - Added to the routing table when a route is manually configured.
- **Dynamic routing protocols** - Added to the routing table when routing protocols dynamically learn about the remote network.

Default Route: Specifies a next-hop router to use when the routing table does not contain a specific route that matches the destination IP address. The default route can be entered manually as a static route, or learned automatically from a dynamic routing protocol.

- A default route has a /0 prefix length. This means that no bits need to match the destination IP address for this route entry to be used. If there are no routes with a match longer than 0 bits, the default route is used to forward the packet. The default

Basic Router Configuration Review

Configuration Commands

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with
CNTL/Z.
Router(config)# hostname R1
R1(config)# enable secret class
R1(config)# line console 0
R1(config-line)# logging synchronous
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# transport input ssh telnet
R1(config-line)# exit
R1(config)# service password-encryption
R1(config)#
banner motd #
Enter TEXT message. End with a new line and the #
*****
WARNING: Unauthorized access is prohibited!
*****
#
#
```

```
R1(config)# ipv6 unicast-routing
R1(config)# interface gigabitethernet 0/0/0
R1(config-if)# description Link to LAN 1
R1(config-if)# ip address 10.0.1.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# ipv6 address fe80::1:a link-local
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface gigabitethernet 0/0/1
R1(config-if)# description Link to LAN 2
R1(config-if)# ip address 10.0.2.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:2::1/64
R1(config-if)# ipv6 address fe80::1:b link-local
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface serial 0/1/1
R1(config-if)# description Link to R2
R1(config-if)# ip address 10.0.3.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:3::1/64
R1(config-if)# ipv6 address fe80::1:c link-local
R1(config-if)# no shutdown
R1(config-if)# exit
R1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R1#
```

Basic Router Configuration Review

Verification Commands

Common verification commands include the following:

- **show ip interface brief**
- **show running-config interface** *interface-type number*
- **show interfaces**
- **show ip interface**
- **show ip route**
- **ping**

In each case, replace **ip** with **ipv6** for the IPv6 version of the command.

Filter Command Output

Filtering commands can be used to display specific sections of output. To enable the filtering command, enter a pipe (|) character after the **show** command and then enter a filtering parameter and a filtering expression.

The filtering parameters that can be configured after the pipe include:

- **section** - This displays the entire section that starts with the filtering expression.
- **include** - This includes all output lines that match the filtering expression.
- **exclude** - This excludes all output lines that match the filtering expression.
- **begin** - This displays all the output lines from a certain point, starting with the line that matches the filtering expression.

Note: Output filters can be used in combination with any **show** command.

Route Sources

A routing table contains a list of routes to known networks (prefixes and prefix lengths). The source of this information is derived from the following:

- Directly connected networks
- Static routes
- Dynamic routing protocols

The source for each route in the routing table is identified by a code. Common codes include the following:

- **L** - Identifies the address assigned to a router interface.
- **C** - Identifies a directly connected network.
- **S** - Identifies a static route created to reach a specific network.
- **O** - Identifies a dynamically learned network from another router using the OSPF routing protocol.
- ***** - This route is a candidate for a default route.

Routing Table Principles

There are three routing table principles as described in the table. These are issues that are addressed by the proper configuration of dynamic routing protocols or static routes on all the routers between the source and destination devices.

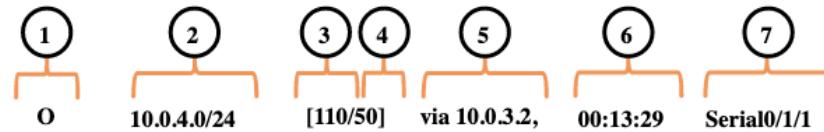
Routing Table Principle	Example
Every router makes its decision alone, based on the information it has in its own routing table.	<ul style="list-style-type: none">R1 can only forward packets using its own routing table.R1 does not know what routes are in the routing tables of other routers (e.g., R2).
The information in a routing table of one router does not necessarily match the routing table of another router.	Just because R1 has route in its routing table to a network in the internet via R2, that does not mean that R2 knows about that same network.
Routing information about a path does not provide return routing information.	R1 receives a packet with the destination IP address of PC1 and the source IP address of PC3. Just because R1 knows to forward the packet out its G0/0/0 interface, doesn't necessarily mean that it knows how to forward packets originating from PC1 back to the remote network of PC3

Routing Table Entries

In the figure, the numbers identify the following information:

- **Route source** - This identifies how the route was learned.
- **Destination network (prefix and prefix length)** - This identifies the address of the remote network.
- **Administrative distance** - This identifies the trustworthiness of the route source. Lower values indicate preferred route source.
- **Metric** - This identifies the value assigned to reach the remote network. Lower values indicate preferred routes.
- **Next-hop** - This identifies the IP address of the next router to which the packet would be forwarded.
- **Route timestamp** - This identifies how much time has passed since the route was learned.
- **Exit interface** - This identifies the egress interface to use for outgoing packets to reach their final destination.

IPv4 Routing Table



IPv6 Routing Table



Note: The prefix length of the destination network specifies the minimum number of far-left bits that must match between the IP address of the packet and the destination network (prefix) for this route to be used.

Directly Connected Networks

To learn about any remote networks, the router must have at least one active interface configured with an IP address and subnet mask (prefix length). This is known as a directly connected network or a directly connected route. Routers add a directly connected route to its routing table when an interface is configured with an IP address and is activated.

- A directly connected network is denoted by a status code of **C** in the routing table. The route contains a network prefix and prefix length.
- The routing table also contains a local route for each of its directly connected networks, indicated by the status code of **L**.
- For IPv4 local routes the prefix length is /32 and for IPv6 local routes the prefix length is /128. This means the destination IP address of the packet must match all the bits in the local route for this route to be a match. The purpose of the local route is to efficiently determine when it receives a packet for the interface instead of a packet that needs to be forwarded.

Static Routes

After directly connected interfaces are configured and added to the routing table, static or dynamic routing can be implemented for accessing remote networks. Static routes are manually configured. They define an explicit path between two networking devices. They are not automatically updated and must be manually reconfigured if the network topology changes.

Static routing has three primary uses:

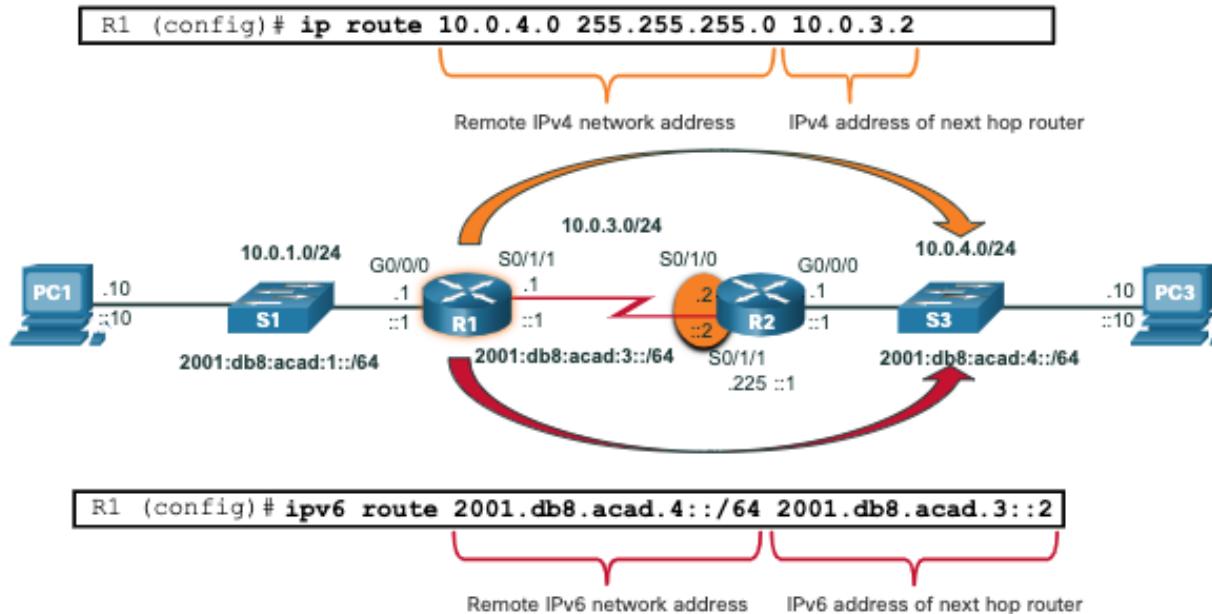
It provides ease of routing table maintenance in smaller networks that are not expected to grow significantly.

It uses a single default route to represent a path to any network that does not have a more specific match with another route in the routing table. Default routes are used to send traffic to any destination beyond the next upstream router.

It routes to and from stub networks. A stub network is a network accessed by a single route, and the router has only one neighbor.

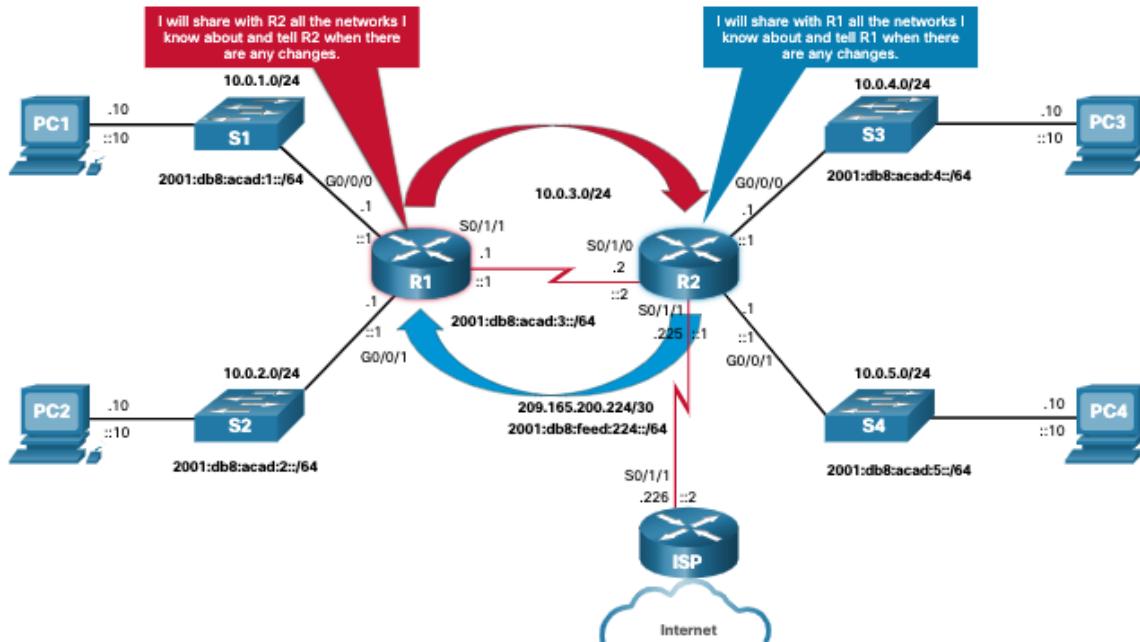
Static Routes in the IP Routing Table

The topology in the figure is simplified to show only one LAN attached to each router. The figure shows IPv4 and IPv6 static routes configured on R1 to reach the 10.0.4.0/24 and 2001:db8:acad:4::/64 networks on R2.



Dynamic Routing Protocols

Dynamic routing protocols are used by routers to automatically share information about the reachability and status of remote networks. Dynamic routing protocols perform several activities, including network discovery and maintaining routing tables.



Dynamic Routes in the Routing Table

OSPF is now being used in our sample topology to dynamically learn all the networks connected to R1 and R2. The routing table entries use the status code of **O** to indicate the route was learned by the OSPF routing protocol. Both entries also include the IP address of the next-hop router, via *ip-address*.

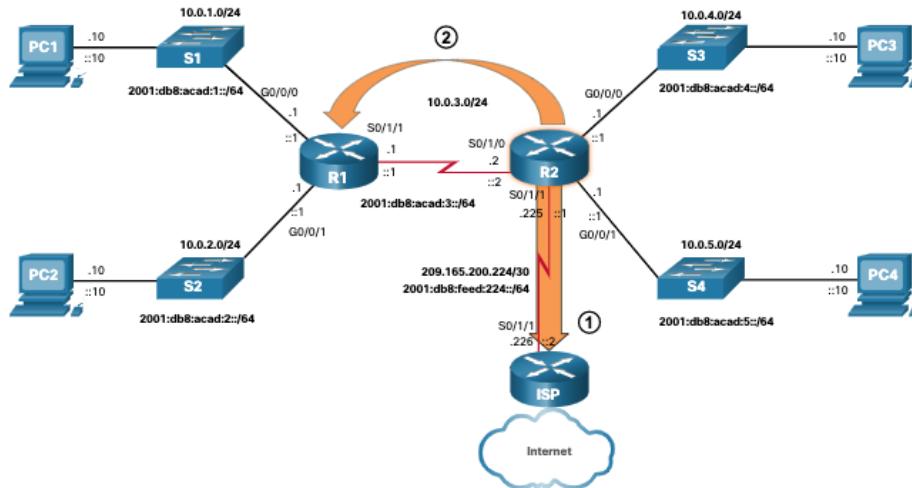
Note: IPv6 routing protocols use the link-local address of the next-hop router.

Note: OSPF routing configuration for IPv4 and IPv6 is beyond the scope of this course.

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP D - EIGRP,
EX - EIGRP external, O - OSPF, IA - OSPF inter area
(output omitted for brevity)
O 10.0.4.0/24 [110/50] via 10.0.3.2, 00:24:22, Serial0/1/1
O 10.0.5.0/24 [110/50] via 10.0.3.2, 00:24:15, Serial0/1/1
R1# show ipv6 route
IPv6 Routing Table - default - 10 entries
(Output omitted)
NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
O 2001:DB8:ACAD:4::/64 [110/50]
    via FE80::2:C, Serial0/1/1
O 2001:DB8:ACAD:5::/64 [110/50]
    via FE80::2:C, Serial0/1/1
```

Default Route

The default route specifies a next-hop router to use when the routing table does not contain a specific route that matches the destination IP address. A default route can be either a static route or learned automatically from a dynamic routing protocol. A default route has an IPv4 route entry of 0.0.0.0/0 or an IPv6 route entry of ::/0. This means that zero or no bits need to match between the destination IP address and the default route.



Structure of an IPv4 Routing Table

IPv4 was standardized using the now obsolete classful addressing architecture. The IPv4 routing table is organized using this same classful structure. Although the lookup process no longer uses classes, the structure of the IPv4 routing table still retains in this format.

An indented entry is known as a child route. A route entry is indented if it is the subnet of a classful address (class A, B or C network). Directly connected networks will always be indented (child routes) because the local address of the interface is always entered in the routing table as a /32. The child route will include the route source and all the forwarding information such as the next-hop address. The classful network address of this subnet will be shown above the route entry, less indented, and without a source code. That route is known as a parent route.

Structure of an IPv4 Routing Table

- An indented entry is known as a **child route**. A route entry is indented if it is the subnet of a classful address (class A, B or C network).
- Directly connected networks will always be indented (child routes) because the local address of the interface is always entered in the routing table as a /32.
- The child route will include the route source and all the forwarding information such as the next-hop address.
- The classful network address of this subnet will be shown above the route entry, less indented, and without a source code. That route is known as a **parent route**.

```
Router# show ip route
```

(Output omitted)

```
 192.168.1.0/24 is variably..
C   192.168.1.0/24 is direct..
L   192.168.1.1/32 is direct..
O   192.168.2.0/24 [110/65]..
O   192.168.3.0/24 [110/65]..
  192.168.12.0/24 is variab..
C   192.168.12.0/30 is direct..
L   192.168.12.1/32 is direct..
  192.168.13.0/24 is variably..
C   192.168.13.0/30 is direct..
L   192.168.13.1/32 is direct..
  192.168.23.0/30 is subnette..
O   192.168.23.0/30 [110/128]..
```

```
Router#
```

Administrative Distance

A route entry for a specific network address (prefix and prefix length) can only appear once in the routing table. However, it is possible that the routing table learns about the same network address from more than one routing source. Except for very specific circumstances, only one dynamic routing protocol should be implemented on a router. Each routing protocol may decide on a different path to reach the destination based on the metric of that routing protocol.

This raises a few questions, such as the following:

- How does the router know which source to use?
- Which route should it install in the routing table?

Cisco IOS 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.

Administrative Distance (Cont.)

The table lists various routing protocols and their associated ADs.

Route Source	Administrative Distance
Directly connected	0
Static route	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal BGP	200

Static or Dynamic?

Static and dynamic routing are not mutually exclusive. Rather, most networks use a combination of dynamic routing protocols and static routes.

Static routes are commonly used in the following scenarios:

- As a default route forwarding packets to a service provider
- For routes outside the routing domain and not learned by the dynamic routing protocol
- When the network administrator wants to explicitly define the path for a specific network
- For routing between stub networks

Static routes are useful for smaller networks with only one path to an outside network. They also provide security in a larger network for certain types of traffic, or links to other networks that need more control.

Static or Dynamic? (Cont.)

Dynamic routing protocols are implemented in any type of network consisting of more than just a few routers. Dynamic routing protocols are scalable and automatically determine better routes if there is a change in the topology.

Dynamic routing protocols are commonly used in the following scenarios:

- In networks consisting of more than just a few routers
- When a change in the network topology requires the network to automatically determine another path
- For scalability. As the network grows, the dynamic routing protocol automatically learns about any new networks.

Static or Dynamic? (Cont.)

The table shows a comparison of some the differences between dynamic and static routing.

Feature	Dynamic Routing	Static Routing
Configuration complexity	Independent of network size	Increases with network size
Topology changes	Automatically adapts to topology changes	Administrator intervention required
Scalability	Suitable for simple to complex network topologies	Suitable for simple topologies
Security	Security must be configured	Security is inherent
Resource Usage	Uses CPU, memory, and link bandwidth	No additional resources needed
Path Predictability	Route depends on topology and routing protocol used	Explicitly defined by the administrator

Best Path

The best path is selected by a routing protocol based on the value or metric it uses to determine the distance to reach a network. 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.

Dynamic routing protocols typically use their own rules and metrics to build and update routing tables. The following table lists common dynamic protocols and their metrics.

Routing Protocol	Metric
Routing Information Protocol (RIP)	<ul style="list-style-type: none">The metric is “hop count”.Each router along a path adds a hop to the hop count.A maximum of 15 hops allowed.
Open Shortest Path First (OSPF)	<ul style="list-style-type: none">The metric is “cost” which is based on the cumulative bandwidth from source to destination.Faster links are assigned lower costs compared to slower (higher cost) links.
Enhanced Interior Gateway Routing Protocol (EIGRP)	<ul style="list-style-type: none">It calculates a metric based on the slowest bandwidth and delay values.It could also include load and reliability into the metric calculation.

Load Balancing

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. The router forwards packets using the multiple exit interfaces listed in the routing table.
- If configured correctly, load balancing can increase the effectiveness and performance of the network.
- Equal cost load balancing is implemented automatically by dynamic routing protocols. It is enabled with static routes when there are multiple static routes to the same destination network using different next-hop routers.

Note: Only EIGRP supports unequal cost load balancing.

IP Static Routing

IPv4 Static Route Command

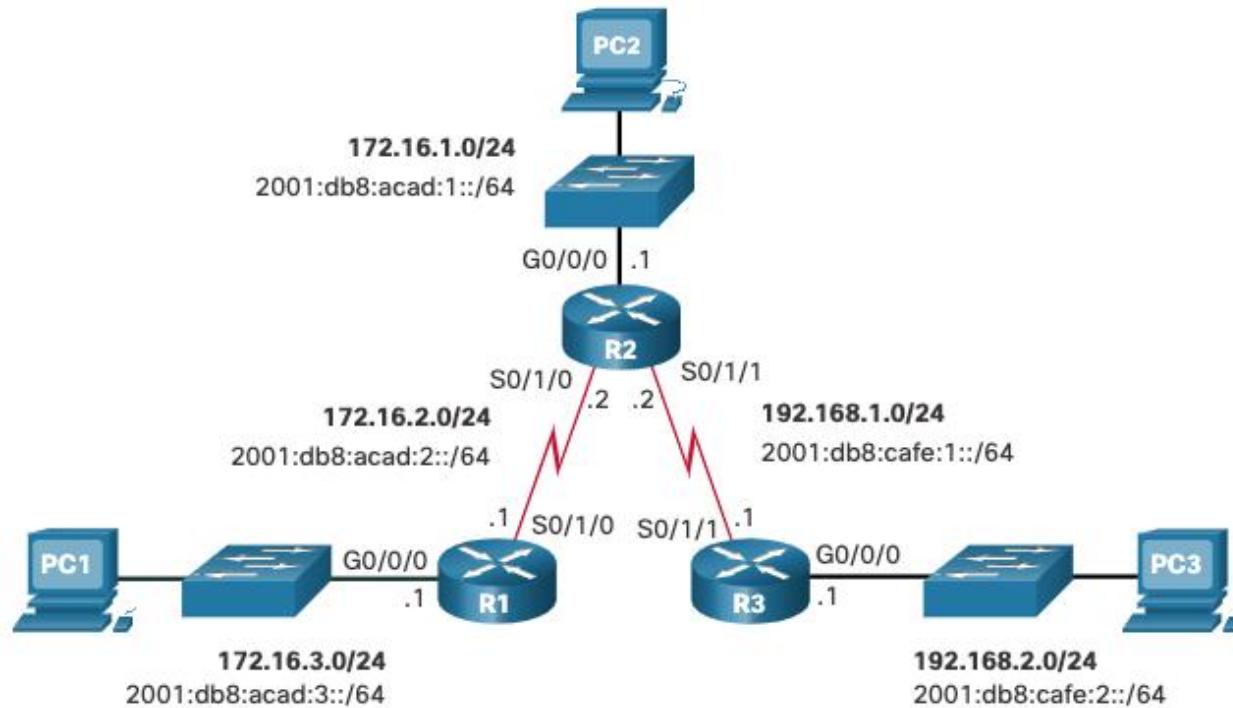
IPv4 static routes are configured using the following global configuration command:

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

Note: Either the *ip-address*, *exit-intf*, or the *ip-address* and *exit-intf* parameters must be configured.

Dual-Stack Topology

The figure shows a dual-stack network topology. Currently, no static routes are configured for either IPv4 or IPv6.



IPv4 Starting Routing Tables

- Each router has entries only for directly connected networks and associated local addresses.
- R1 can ping R2, but cannot ping the R3 LAN

```
R1# show ip route | begin Gateway
Gateway of last resort is not set
    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C        172.16.2.0/24 is directly connected, Serial0/1/0
L        172.16.2.1/32 is directly connected, Serial0/1/0
C        172.16.3.0/24 is directly connected, GigabitEthernet0/0/0
L        172.16.3.1/32 is directly connected, GigabitEthernet0/0/0
R1#
R1# ping 172.16.2.2
Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 172.16.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
R1# ping 192.168.2.1
Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

IPv4 Next-Hop Static Route

In a next-hop static route, only the next-hop IP address is specified. The exit interface is derived from the next hop. For example, three next-hop IPv4 static routes are configured on R1 using the IP address of the next hop, R2.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.2
R1(config)# ip route 192.168.1.0 255.255.255.0 172.16.2.2
R1(config)# ip route 192.168.2.0 255.255.255.0 172.16.2.2
```

The resulting routing table entries on R1:

```
R1# show ip route | begin Gateway
Gateway of last resort is not set
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
S      172.16.1.0/24 [1/0] via 172.16.2.2
C      172.16.2.0/24 is directly connected, Serial0/1/0
L      172.16.2.1/32 is directly connected, Serial0/1/0
C      172.16.3.0/24 is directly connected, GigabitEthernet0/0/0
L      172.16.3.1/32 is directly connected, GigabitEthernet0/0/0
S      192.168.1.0/24 [1/0] via 172.16.2.2
S      192.168.2.0/24 [1/0] via 172.16.2.2
```

IPv4 Directly Connected Static Routes

When configuring a static route, another option is to use the exit interface to specify the next-hop address. Three directly connected IPv4 static routes are configured on R1 using the exit interface.

Note: Using a next-hop address is generally recommended. Directly connected static routes should only be used with point-to-point serial interfaces.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 s0/1/0
R1(config)# ip route 192.168.1.0 255.255.255.0 s0/1/0
R1(config)# ip route 192.168.2.0 255.255.255.0 s0/1/0
```

```
R1# show ip route | begin Gateway
Gateway of last resort is not set
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
      S    172.16.1.0/24 is directly connected, Serial0/1/0
      C    172.16.2.0/24 is directly connected, Serial0/1/0
      L    172.16.2.1/32 is directly connected, Serial0/1/0
      C    172.16.3.0/24 is directly connected, GigabitEthernet0/0/0
      L    172.16.3.1/32 is directly connected, GigabitEthernet0/0/0
      S    192.168.1.0/24 is directly connected, Serial0/1/0
      S    192.168.2.0/24 is directly connected, Serial0/1/0
```

IPv4 Fully Specified Static Route

- In a fully specified static route, both the exit interface and the next-hop IP address are specified. This form of static route is used when the exit interface is a multi-access interface and it is necessary to explicitly identify the next hop. The next hop must be directly connected to the specified exit interface. Using an exit interface is optional, however it is necessary to use a next-hop address.
- It is recommended that when the exit interface is an Ethernet network, that the static route includes a next-hop address. You can also use a fully specified static route that includes both the exit interface and the next-hop address.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 GigabitEthernet 0/0/1 172.16.2.2
R1(config)# ip route 192.168.1.0 255.255.255.0 GigabitEthernet 0/0/1 172.16.2.2
R1(config)# ip route 192.168.2.0 255.255.255.0 GigabitEthernet 0/0/1 172.16.2.2
```

```
R1# show ip route | begin Gateway
Gateway of last resort is not set
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
S      172.16.1.0/24 [1/0] via 172.16.2.2, GigabitEthernet0/0/1
C      172.16.2.0/24 is directly connected, GigabitEthernet0/0/1
L      172.16.2.1/32 is directly connected, GigabitEthernet0/0/1
C      172.16.3.0/24 is directly connected, GigabitEthernet0/0/0
L      172.16.3.1/32 is directly connected, GigabitEthernet0/0/0
S      192.168.1.0/24 [1/0] via 172.16.2.2, GigabitEthernet0/0/1
S      192.168.2.0/24 [1/0] via 172.16.2.2, GigabitEthernet0/0/1
```

Verify a Static Route

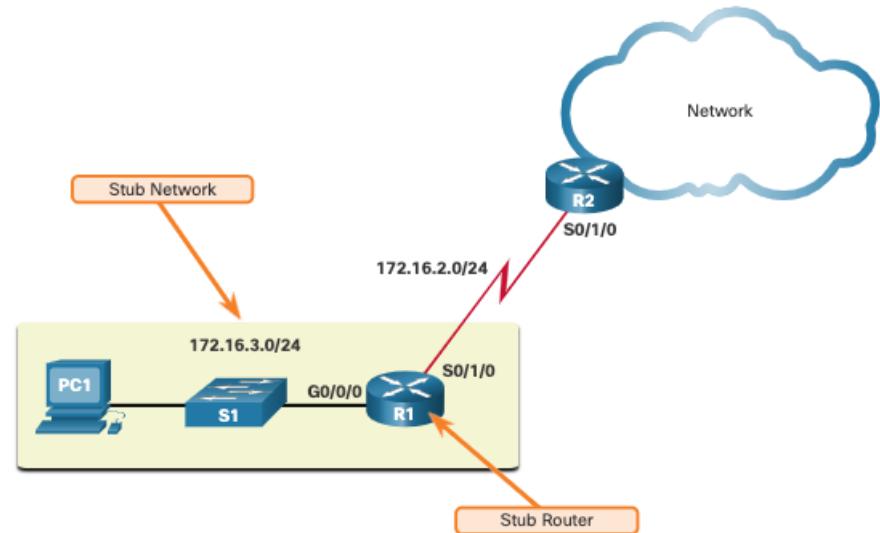
Along with **show ip route**, **show ipv6 route**, **ping** and **traceroute**, other useful commands to verify static routes include the following:

- **show ip route static**
- **show ip route network**
- **show running-config | section ip route**

Replace **ip** with **ipv6** for the IPv6 versions of the command.

Default Static Route

- A default route is a static route that matches all packets. A single default route represents any network that is not in the routing table.
- Routers commonly use default routes that are either configured locally or learned from another router. The default route is used as the Gateway of Last Resort.
- Default static routes are commonly used when connecting an edge router to a service provider network, or a stub router (a router with only one upstream neighbor router).
- The figure shows a typical default static route scenario.



Default Static Route (Cont.)

IPv4 Default Static Route: The command syntax for an IPv4 default static route is similar to any other IPv4 static route, except that the network address is **0.0.0.0** and the subnet mask is **0.0.0.0**. The **0.0.0.0 0.0.0.0** in the route will match any network address.

Note: An IPv4 default static route is commonly referred to as a quad-zero route.

The basic command syntax for an IPv4 default static route is as follows:

```
Router(config)# ip route 0.0.0.0 0.0.0.0 {ip-address | exit-intf}
```

IPv6 Default Static Route: The command syntax for an IPv6 default static route is similar to any other IPv6 static route, except that the ipv6-prefix/prefix-length is **::/0**, which matches all routes.

The basic command syntax for an IPv6 default static route is as follows:

```
Router(config)# ipv6 route ::/0 {ipv6-address | exit-intf}
```

Configure a Default Static Route

The example shows an IPv4 default static route configured on R1. With the configuration shown in the example, any packets not matching more specific route entries are forwarded to R2 at 172.16.2.2.

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

An IPv6 default static route is configured in similar fashion. With this configuration any packets not matching more specific IPv6 route entries are forwarded to R2 at 2001:db8:acad:2::2

```
R1 (config)# ipv6 route ::/0 2001:db8:acad:2::2
```

Verify a Default Static Route

The **show ip route static** command output from R1 displays the contents of the static routes in the routing table. Note the asterisk (*) next to the route with code 'S'. The asterisk indicates that this static route is a candidate default route, which is why it is selected as the Gateway of Last Resort.

Notice that the static default route configuration uses the /0 mask for IPv4 default routes. Remember that the IPv4 subnet mask in a routing table determines how many bits must match between the destination IP address of the packet and the route in the routing table. A /0 mask indicates that none of the bits are required to match. As long as a more specific match does not exist, the default static route matches all packets.

```
R1# show ip route static
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

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
```

Configure Floating Static R Floating Static Routes

- Another type of static route is a floating static route. Floating static routes are static routes that are used to provide a backup path to a primary static or dynamic route. The floating static route is only used when the primary route is not available.
- To accomplish this, the floating static route is configured with a higher administrative distance than the primary route. The administrative distance represents the trustworthiness of a route. If multiple paths to the destination exist, the router will choose the path with the lowest administrative distance.
- By default, static routes have an administrative distance of 1, making them preferable to routes learned from dynamic routing protocols.
- The administrative distance of a static route can be increased to make the route less desirable than that of another static route or a route learned through a dynamic routing protocol. In this way, the static route “floats” and is not used when the route with the better administrative distance is active.

Configure IPv4 Floating Static Route

The commands to configure default and floating IP default routes are as follows:

```
R1(config) # ip route 0.0.0.0 0.0.0.0 172.16.2.2
R1(config) # ip route 0.0.0.0 0.0.0.0 10.10.10.2 5
R1(config) # ipv6 route ::/0 2001:db8:acad:2::2
R1(config) # ipv6 route ::/0 2001:db8:feed:10::2 5
```

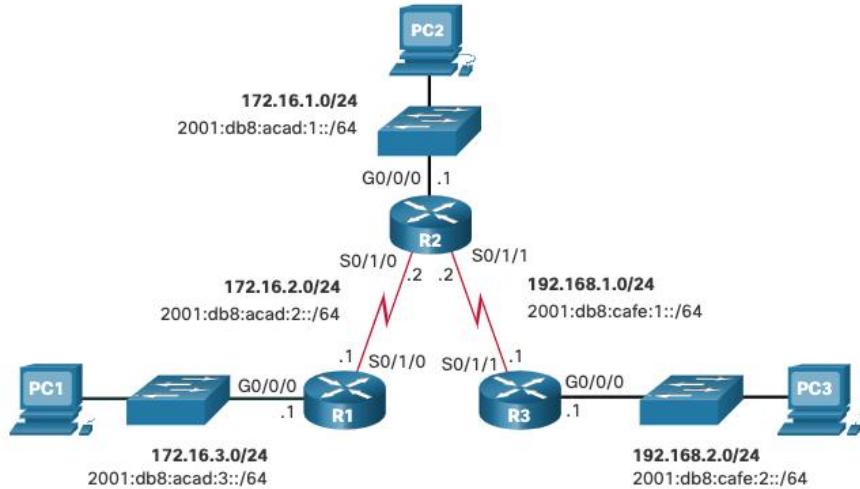
The **show ip route** and **show ipv6 route** output verifies that the default routes to R2 are installed in the routing table. Note that the IPv4 floating static route to R3 is not present in the routing table.

```
R1# show ip route static | begin Gateway
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
R1# show ipv6 route static | begin S :
S ::/0 [1/0]
    via 2001:DB8:ACAD:2::2
R1#
```

Test the Floating Static R

- What would happen if R2 failed? To simulate this, R2 shuts down both of its serial interfaces.
- R1 automatically generates syslog messages for the link going down.
- A look at R1's routing table would show the secondary route being used.



```
R1# show ip route static | begin Gateway
Gateway of last resort is 10.10.10.2 to network 0.0.0.0
S*    0.0.0.0/0 [5/0] via 10.10.10.2
R1# show ipv6 route static | begin :::
S    ::/0 [5/0]
      via 2001:DB8:FEED:10::2
R1#
```

Host Routes

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

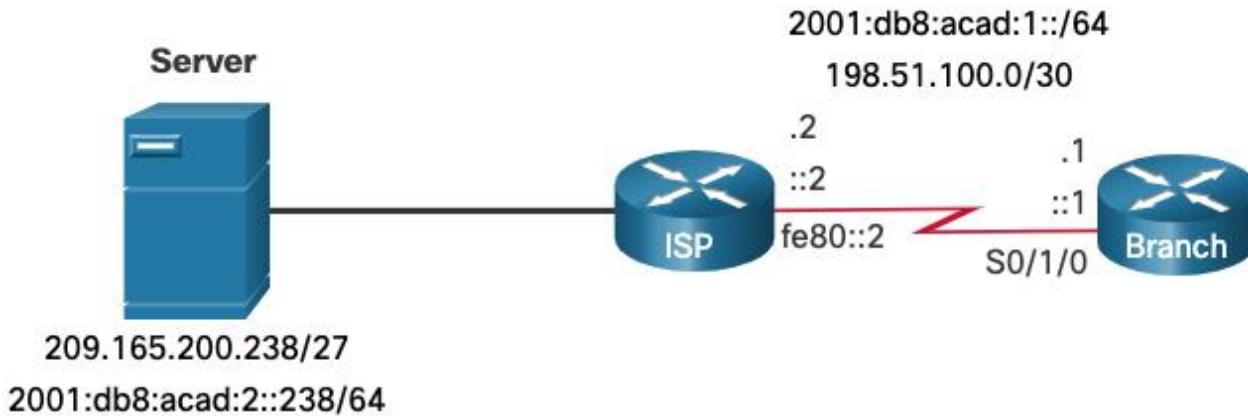
- Automatically installed when an IP address is configured on the router
- Configured as a static host route
- Host route automatically obtained through other methods (discussed in later courses)

Automatically Installed Host Route

- 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.
- The local routes are marked with **L** in the output of the routing table.

Static Host Routes

A host route can be a manually configured static route to direct traffic to a specific destination device, such as the server shown in the figure. The static route uses a destination IP address and a 255.255.255.255 (/32) mask for IPv4 host routes, and a /128 prefix length for IPv6 host routes.



Configure Static Host Routes

The example shows the IPv4 and IPv6 static host route configuration on the Branch router to access the 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::238/128 2001:db8:acad:1::2
Branch(config)# exit
Branch#
```

Verify Static Host Routes

A review of both the IPv4 and IPv6 route tables verifies that the routes are active.

```
Branch# show ip route | begin Gateway
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, Serial0/1/0
L        198.51.100.1/32 is directly connected, Serial0/1/0
    209.165.200.0/32 is subnetted, 1 subnets
S            209.165.200.238 [1/0] via 198.51.100.2
Branch# show ipv6 route
(Output omitted)
C    2001:DB8:ACAD:1::/64 [0/0]
    via Serial0/1/0, directly connected
L    2001:DB8:ACAD:1::1/128 [0/0]
    via Serial0/1/0, receive
S    2001:DB8:ACAD:2::238/128 [1/0]
    via 2001:DB8:ACAD:1::2
Branch#
```

Single-Area OSPFv2 Concepts

OSPF Features and Characteristics

Introduction to OSPF

- OSPF is a link-state routing protocol that was developed as an alternative for the distance vector Routing Information Protocol (RIP). OSPF has significant advantages over RIP in that it offers faster convergence and scales to much larger network implementations.
- OSPF is a link-state routing protocol that uses the concept of areas. A network administrator can divide the routing domain into distinct areas that help control routing update traffic.
- A link is an interface on a router, a network segment that connects two routers, or a stub network such as an Ethernet LAN that is connected to a single router.
- Information about the state of a link is known as a link-state. All link-state information includes the network prefix, prefix length, and cost.
- This module covers basic, single-area OSPF implementations and configurations.

OSPF Features and Characteristics

Components of OSPF

- All routing protocols share similar components. They all use routing protocol messages to exchange route information. The messages help build data structures, which are then processed using a routing algorithm.
- Routers running OSPF exchange messages to convey routing information using five types of packets:
 - Hello packet
 - Database description packet
 - Link-state request packet
 - Link-state update packet
 - Link-state acknowledgment packet
- These packets are used to discover neighboring routers and also to exchange routing information to maintain accurate information about the network.

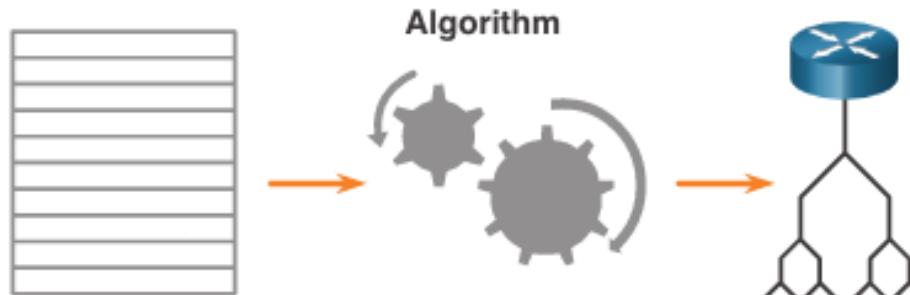
Components of OSPF (Cont.)

OSPF messages are used to create and maintain three OSPF databases, as follows:

Database	Table	Description
Adjacency Database	Neighbor Table	<ul style="list-style-type: none"> List of all neighbor routers to which a router has established bi-directional communication. This table is unique for each router. Can be viewed using the show ip ospf neighbor command.
Link-state Database (LSDB)	Topology Table	<ul style="list-style-type: none"> Lists information about all other routers in the network. The database represents the network LSDB. All routers within an area have identical LSDB. Can be viewed using the show ip ospf database command.
Forwarding Database	Routing Table	<ul style="list-style-type: none"> List of routes generated when an algorithm is run on the link-state database. Each router's routing table is unique and contains information on how and where to send packets to other routers. Can be viewed using the show ip route command.

Components of OSPF (Cont.)

- The router builds the topology table using results of calculations based on the Dijkstra shortest-path first (SPF) algorithm. The SPF algorithm is based on the cumulative cost to reach a destination.
- The SPF algorithm creates an SPF tree by placing each router at the root of the tree and calculating the shortest path to each node. The SPF tree is then used to calculate the best routes. OSPF places the best routes into the forwarding database, which is used to make the routing table.



Link-State Operation

To maintain routing information, OSPF routers complete a generic link-state routing process to reach a state of convergence. The following are the link-state routing steps that are completed by a router:

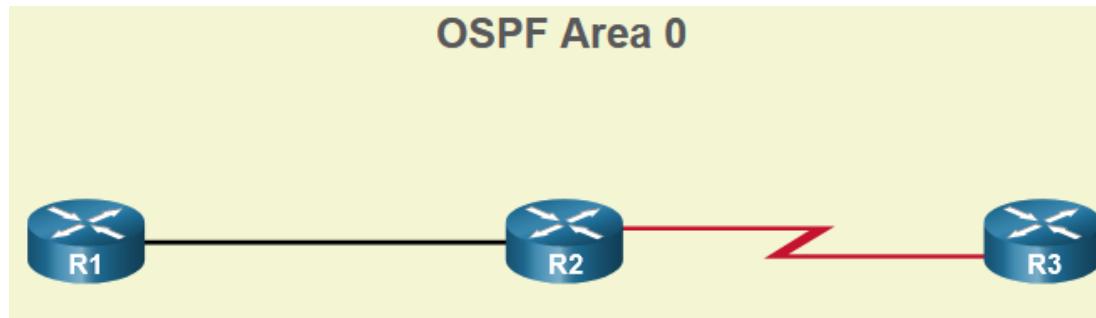
1. Establish Neighbor Adjacencies
2. Exchange Link-State Advertisements
3. Build the Link State Database
4. Execute the SPF Algorithm
5. Choose the Best Route

Single-Area and Multiarea OSPF

To make OSPF more efficient and scalable, OSPF supports hierarchical routing using areas. An OSPF area is a group of routers that share the same link-state information in their LSDBs. OSPF can be implemented in one of two ways, as follows:

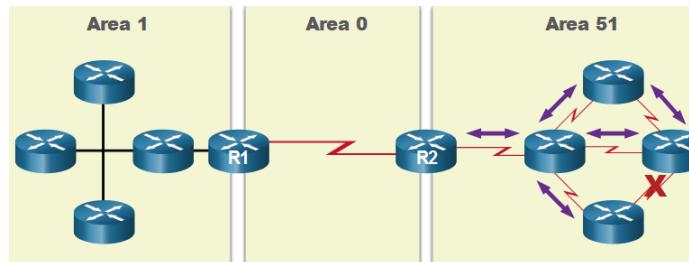
- **Single-Area OSPF** - All routers are in one area. Best practice is to use area 0.
- **Multiarea OSPF** - OSPF is implemented using multiple areas, in a hierarchical fashion. All areas must connect to the backbone area (area 0). Routers interconnecting the areas are referred to as Area Border Routers (ABRs).

The focus of this module is on single-area OSPFv2.



Multiarea OSPF

- The hierarchical-topology design options with multiarea OSPF can offer the following advantages.
- **Smaller routing tables** - Tables are smaller because there are fewer routing table entries. This is because network addresses can be summarized between areas. Route summarization is not enabled by default.
- **Reduced link-state update overhead** - Designing multiarea OSPF with smaller areas minimizes processing and memory requirements.
- **Reduced frequency of SPF calculations** — Multiarea OSPF localize the impact of a topology change within an area. For instance, it minimizes routing update impact because LSA flooding stops at the area boundary.



Types of OSPF Packets

The table summarizes the five different types of Link State Packets (LSPs) used by OSPFv2. OSPFv3 has similar packet types.

Type	Packet Name	Description
1	Hello	Discovers neighbors and builds adjacencies between them
2	Database Description (DBD)	Checks for database synchronization between routers
3	Link-State Request (LSR)	Requests specific link-state records from router to router
4	Link-State Update (LSU)	Sends specifically requested link-state records
5	Link-State Acknowledgment (LSAck)	Acknowledges the other packet types

Link-State Updates

- LSUs are also used to forward OSPF routing updates. An LSU packet can contain 11 different types of OSPFv2 LSAs. OSPFv3 renamed several of these LSAs and also contains two additional LSAs.
- LSU and LSA are often used interchangeably, but the correct hierarchy is LSU packets contain LSA messages.

LSUs		
Type	Packet Name	Description
1	Hello	Discovers neighbors and builds adjacencies between them
2	DBD	Checks for database synchronization between routers
3	LSR	Requests specific link-state records from router to router
4	LSU	Sends specifically requested link-state records
5	LSAck	Acknowledges the other packet types

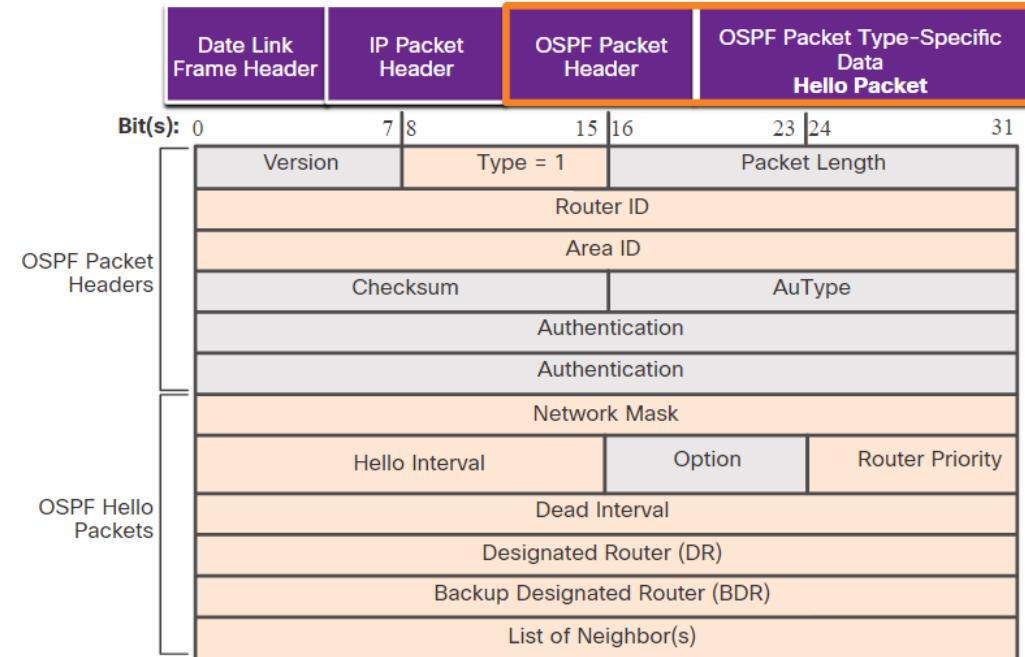


LSAs	
LSA Type	Description
1	Router LSAs
2	Checks for database synchronization between routers
3 or 4	Summary LSAs
5	Autonomous System External LSAs
6	Multicast OSPF LSAs
7	Defined for Not-So-Stubby Areas
8	External Attributes LSA for Border Gateway Patrol (BGP)

Hello Packet

The OSPF Type 1 packet is the Hello packet. Hello packets are used to do the following:

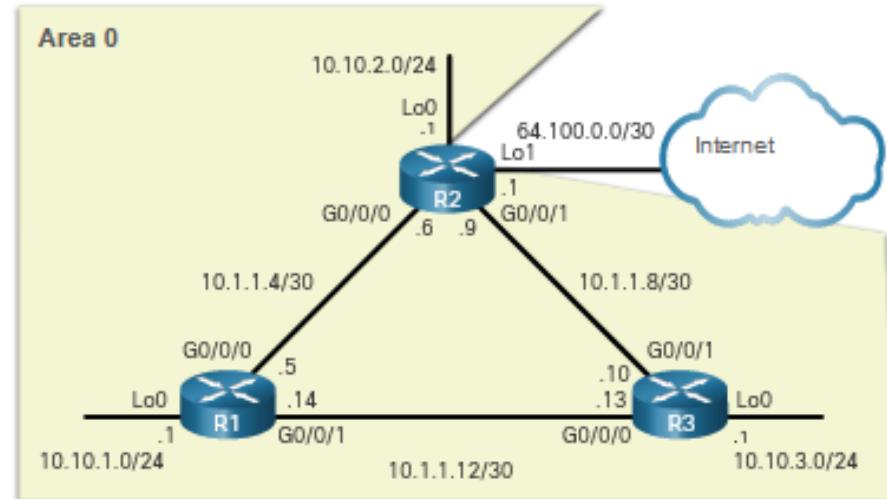
- Discover OSPF neighbors and establish neighbor adjacencies.
- Advertise parameters on which two routers must agree to become neighbors.
- Elect the Designated Router (DR) and Backup Designated Router (BDR) on multiaccess networks like Ethernet. Point-to-point links do not require DR or BDR.



Single-Area OSPFv2 Configuration

OSPF Reference Topology

The figure shows the topology used for configuring OSPFv2 in this module. The routers in the topology have a starting configuration, including interface addresses. There is currently no static routing or dynamic routing configured on any of the routers. All interfaces on R1, R2, and R3 (except the loopback 1 on R2) are within the OSPF backbone area. The ISP router is used as the gateway to the internet of the routing domain.



Router Configuration Mode for OSX

OSPFv2 is enabled using the **router ospf** *process-id* global configuration mode command. The *process-id* value represents a number between 1 and 65,535 and is selected by the network administrator. The *process-id* value is locally significant. It is considered best practice to use the same *process-id* on all OSPF routers.

Configure a Loopback Interface as the Router ID

Instead of relying on physical interface, the router ID can be assigned to a loopback interface. Typically, the IPv4 address for this type of loopback interface should be configured using a 32-bit subnet mask (255.255.255.255). This effectively creates a host route. A 32-bit host route would not get advertised as a route to other OSPF routers.

OSPF does not need to be enabled on an interface for that interface to be chosen as the router ID.

```
R1(config-if)# interface Loopback 1
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# end
R1# show ip protocols | include Router ID
  Router ID 1.1.1.1
R1#
```

The network Command Syntax

- You can specify the interfaces that belong to a point-to-point network by configuring the **network** command. You can also configure OSPF directly on the interface with the **ip ospf** command.
- The basic syntax for the **network** command is as follows:

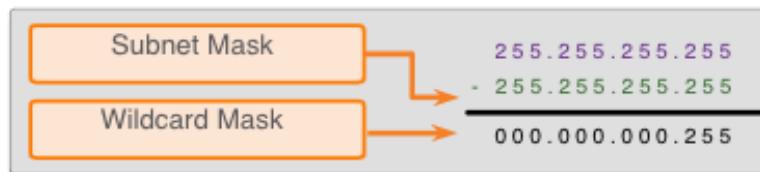
```
Router(config-router) # network network-address wildcard-mask area area-id
```

- The *network-address wildcard-mask* syntax is used to enable OSPF on interfaces. Any interfaces on a router that match this part of the command are enabled to send and receive OSPF packets.
- The **area area-id** syntax refers to the OSPF area. When configuring single-area OSPFv2, the **network** command must be configured with the same *area-id* value on all routers. Although any area ID can be used, it is good practice to use an area ID of 0 with single-area OSPFv2. This convention makes it easier if the network is later altered to support multiarea OSPFv2.

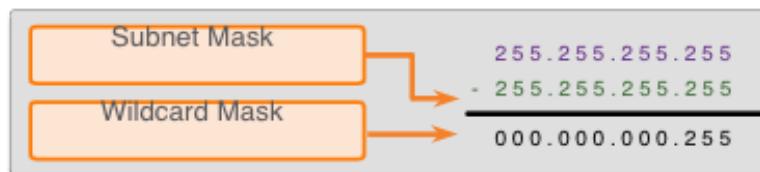
The Wildcard Mask

- The wildcard mask is typically the inverse of the subnet mask configured on that interface.
- The easiest method for calculating a wildcard mask is to subtract the network subnet mask from 255.255.255.255, as shown for /24 and /26 subnet masks in the figure.

Calculating a Wildcard Mask for /24



Calculating a Wildcard Mask for /26



Configure OSPF Using the network Command

Within routing configuration mode, there are two ways to identify the interfaces that will participate in the OSPFv2 routing process.

- In the first example, the wildcard mask identifies the interface based on the network addresses. Any active interface that is configured with an IPv4 address belonging to that network will participate in the OSPFv2 routing process.
- **Note:** Some IOS versions allow the subnet mask to be entered instead of the wildcard mask. The IOS then converts the subnet mask to the wildcard mask format.

```
R1(config)# router ospf 10
R1(config-router)# network 10.10.1.0 0.0.0.255 area 0
R1(config-router)# network 10.1.1.4 0.0.0.3 area 0
R1(config-router)# network 10.1.1.12 0.0.0.3 area 0
R1(config-router)#

```

Configure OSPF Using the network Command

- As an alternative, OSPFv2 can be enabled by specifying the exact interface IPv4 address using a quad zero wildcard mask. Entering **network 10.1.1.5 0.0.0.0 area 0** on R1 tells the router to enable interface Gigabit Ethernet 0/0/0 for the routing process.
- The advantage of specifying the interface is that the wildcard mask calculation is not necessary. Notice that in all cases, the **area** argument specifies area 0.

```
R1(config)# router ospf 10
R1(config-router)# network 10.10.1.1 0.0.0.0 area 0
R1(config-router)# network 10.1.1.5 0.0.0.0 area 0
R1(config-router)# network 10.1.1.14 0.0.0.0 area 0
R1(config-router)#

```

Configure OSPF Using the **ip ospf** Command

To configure OSPF directly on the interface, use the **ip ospf** interface configuration mode command. The syntax is as follows:

```
Router(config-if)# ip ospf process-id area area-id
```

Remove the network commands using the **no** form of the command. Then go to each interface and configure the **ip ospf** command

```
R1(config)# router ospf 10
R1(config-router)# no network 10.10.1.1 0.0.0.0 area 0
R1(config-router)# no network 10.1.1.5 0.0.0.0 area 0
R1(config-router)# no network 10.1.1.14 0.0.0.0 area 0
R1(config-router)# interface GigabitEthernet 0/0/0
R1(config-if)# ip ospf 10 area 0
R1(config-if)# interface GigabitEthernet 0/0/1
R1(config-if)# ip ospf 10 area 0
R1(config-if)# interface Loopback 0
R1(config-if)# ip ospf 10 area 0
R1(config-if)#

```

Cisco OSPF Cost Metric

- Routing protocols use a metric to determine the best path of a packet across a network. OSPF uses cost as a metric. A lower cost indicates a better path.
- The Cisco cost of an interface is inversely proportional to the bandwidth of the interface. Therefore, a higher bandwidth indicates a lower cost. The formula used to calculate the OSPF cost is:

$$\text{Cost} = \text{reference bandwidth} / \text{interface bandwidth}$$

- The default reference bandwidth is 10^8 (100,000,000); therefore, the formula is:

$$\text{Cost} = 100,000,000 \text{ bps} / \text{interface bandwidth in bps}$$

- Because the OSPF cost value must be an integer, FastEthernet, Gigabit Ethernet, and 10 GigE interfaces share the same cost. To correct this situation, you can:
 - Adjust the reference bandwidth with the **auto-cost reference-bandwidth** command on each OSPF router.
 - Manually set the OSPF cost value with the **ip ospf cost** command on necessary interfaces.

Cisco OSPF Cost Metric (Cont.)

Refer to the table for a breakdown of the cost calculation

Interface Type	Reference Bandwidth in bps	Default Bandwidth in bps	Cost
10 Gigabit Ethernet 10 Gbps	100,000,000	÷ 10,000,000,000	0.01 = 1
Gigabit Ethernet 1 Gbps	100,000,000	÷ 1,000,000,000	0.1 = 1
Fast Ethernet 100 Mbps	100,000,000	÷ 100,000,000	1
Ethernet 10 Mbps	100,000,000	÷ 10,000,000	1

Same Costs due to reference bandwidth

Adjust the Reference Bandwidth

- The cost value must be an integer. If something less than an integer is calculated, OSPF rounds up to the nearest integer. Therefore, the OSPF cost assigned to a Gigabit Ethernet interface with the default reference bandwidth of 100,000,000 bps would equal 1, because the nearest integer for 0.1 is 0 instead of 1.

$$\text{Cost} = 100,000,000 \text{ bps} / 1,000,000,000 = 1$$

- For this reason, all interfaces faster than Fast Ethernet will have the same cost value of 1 as a Fast Ethernet interface.
- To assist OSPF in making the correct path determination, the reference bandwidth must be changed to a higher value to accommodate networks with links faster than 100 Mbps.

Adjust the Reference Bandwidth (Cisco)

- Changing the reference bandwidth does not actually affect the bandwidth capacity on the link; rather, it simply affects the calculation used to determine the metric.
- To adjust the reference bandwidth, use the **auto-cost reference-bandwidth Mbps** router configuration command.
- This command must be configured on every router in the OSPF domain.
- Notice in the command that the value is expressed in Mbps; therefore, to adjust the costs for Gigabit Ethernet, use the command **auto-cost reference-bandwidth 1000**. For 10 Gigabit Ethernet, use the command **auto-cost reference-bandwidth 10000**.
- To return to the default reference bandwidth, use the **auto-cost reference-bandwidth 100** command.
- Another option is to change the cost on one specific interface using the **ip ospf cost cost** command.

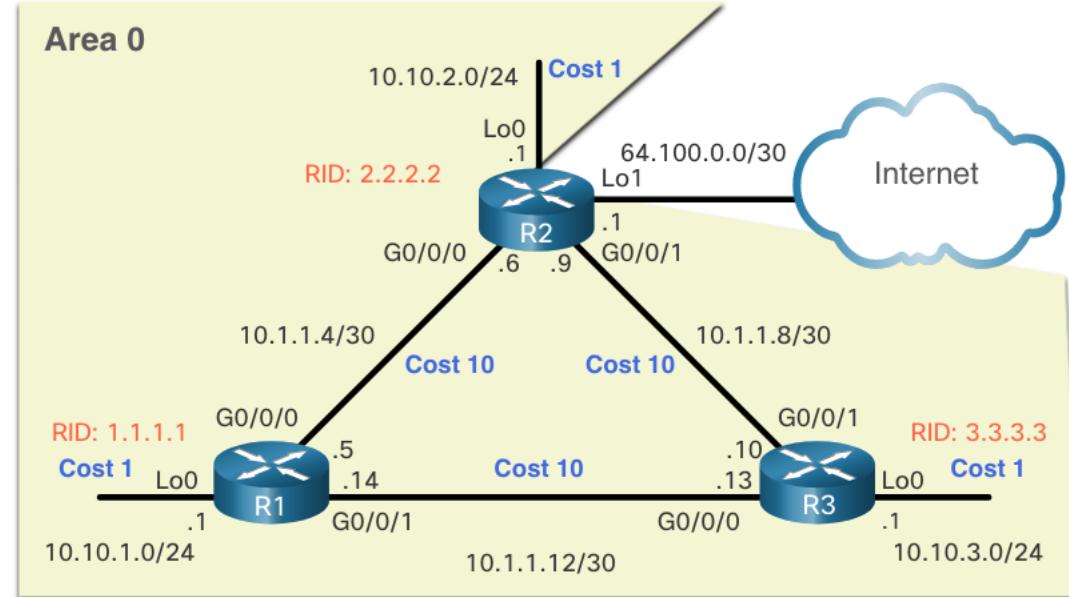
Adjust the Reference Bandwidth (Cost)

- Whichever method is used, it is important to apply the configuration to all routers in the OSPF routing domain.
- The table shows the OSPF cost if the reference bandwidth is adjusted to accommodate 10 Gigabit Ethernet links. The reference bandwidth should be adjusted anytime there are links faster than FastEthernet (100 Mbps).
- Use the **show ip ospf interface** command to verify the current OSPFv2 cost assigned to the interface.

Interface Type	Reference Bandwidth in bps	÷	Default Bandwidth in bps	Cost
10 Gigabit Ethernet 10 Gbps	10,000,000,000	÷	10,000,000,000	1
Gigabit Ethernet 1 Gbps	10,000,000,000	÷	1,000,000,000	10
Fast Ethernet 100 Mbps	10,000,000,000	÷	100,000,000	100
Ethernet 10 Mbps	10,000,000,000	÷	10,000,000	1000

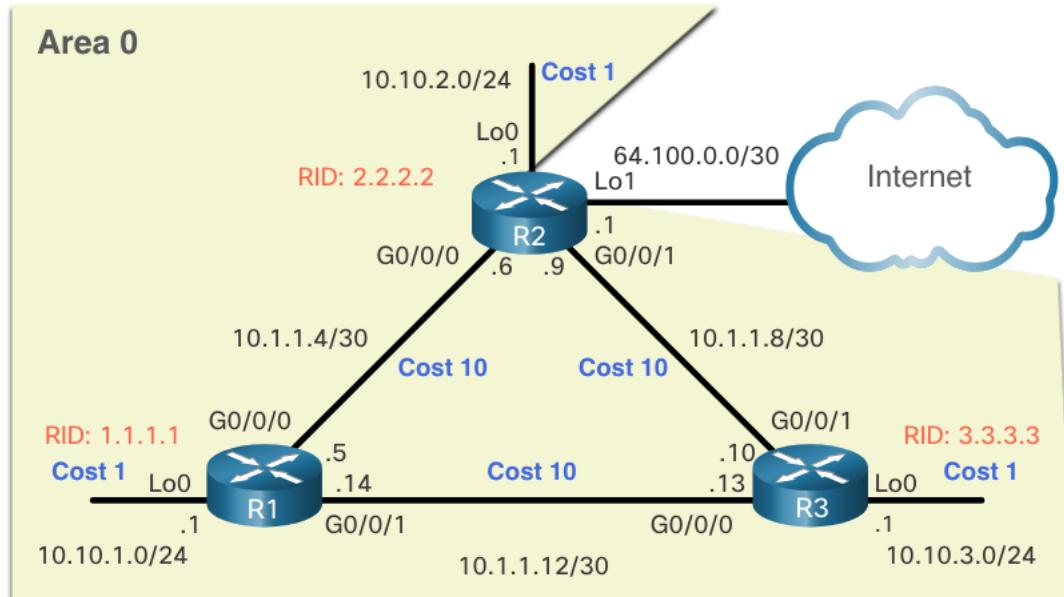
OSPF Accumulates Cost

- The cost of an OSPF route is the accumulated value from one router to the destination network.
- Assuming the **auto-cost reference-bandwidth 10000** command has been configured on all three routers, the cost of the links between each router is now 10. The loopback interfaces have a default cost of 1.



OSPF Accumulates Cost (Cont.)

- You can calculate the cost for each router to reach each network.
- For example, the total cost for R1 to reach the 10.10.2.0/24 network is 11. This is because the link to R2 cost = 10 and the loopback default cost = 1. $10 + 1 = 11$.
- You can verify this with the **show ip route** command.



OSPF Accumulates Cost (Cont.)

Verifying the accumulated cost for the path to the 10.10.2.0/24 network:

```
R1# show ip route | include 10.10.2.0
O          10.10.2.0/24 [110/11] via 10.1.1.6, 01:05:02, GigabitEthernet0/0/0
R1# show ip route 10.10.2.0
Routing entry for 10.10.2.0/24
  Known via "ospf 10", distance 110, metric 11, type intra area
  Last update from 10.1.1.6 on GigabitEthernet0/0/0, 01:05:13 ago
  Routing Descriptor Blocks:
    * 10.1.1.6, from 2.2.2.2, 01:05:13 ago, via GigabitEthernet0/0/0
      Route metric is 11, traffic share count is 1
R1#
```

Manually Set OSPF Cost Value

Reasons to manually set the cost value include:

- The Administrator may want to influence path selection within OSPF, causing different paths to be selected than what normally would given default costs and cost accumulation.
- Connections to equipment from other vendors who use a different formula to calculate OSPF cost.

To change the cost value reported by the local OSPF router to other OSPF routers, use the interface configuration command **ip ospf cost value**.

```
R1(config)# interface g0/0/1 R1(config-if)# ip
ospf cost 30 R1(config-if)# interface lo0
R1(config-if)# ip ospf cost 10 R1(config-if)#
end
R1#
```

Propagate a Default Static Route

To propagate a default route, the edge router must be configured with the following:

- A default static route using the **ip route 0.0.0.0 0.0.0.0 [next-hop-address | exit-intf]** command.
- The **default-information originate** router configuration command. This instructs R2 to be the source of the default route information and propagate the default static route in OSPF updates.

In the example, R2 is configured with a loopback to simulate a connection to the internet. A default route is configured and propagated to all other OSPF routers in the routing domain.

Note: When configuring static routes, best practice is to use the next-hop IP address. However, when simulating a connection to the internet, there is no next-hop IP address. Therefore, we use the *exit-intf* argument.

```
R2(config)# interface lo1
R2(config-if)# ip address 64.100.0.1 255.255.255.252
R2(config-if)# exit
R2(config)# ip route 0.0.0.0 0.0.0.0 loopback 1
%Default route without gateway, if not a point-to-point interface, may impact performance
R2(config)# router ospf 10
R2(config-router)# default-information originate
R2(config-router)# end
R2#
```

Verify the Propagated Default Rou

- You can verify the default route settings on R2 using the **show ip route** command. You can also verify that R1 and R3 received a default route.
- Notice that the route source on R1 is **O*E2**, signifying that it was learned using OSPFv2. The asterisk identifies this as a good candidate for the default route. The E2 designation identifies that it is an external route. The meaning of E1 and E2 is beyond the scope of this module.

```
R2# show ip route | begin Gateway
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
S*          0.0.0.0/0 is directly connected, Loopback1
            10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
(output omitted)
```

```
R1# show ip route | begin Gateway
Gateway of last resort is 10.1.1.6 to network 0.0.0.0
O*E2  0.0.0.0/0 [110/1] via 10.1.1.6, 00:11:08, GigabitEthernet0/0/0
            10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
(output omitted)
```

Verify OSPF Neighbors

After configuring single-area OSPFv2, you will need to verify your configurations. The following two commands are particularly useful for verifying routing:

- **show ip interface brief** - This verifies that the desired interfaces are active with correct IP addressing.
- **show ip route** - This verifies that the routing table contains all the expected routes.

Additional commands for determining that OSPF is operating as expected include the following:

- **show ip ospf neighbor**
- **show ip protocols**
- **show ip ospf**
- **show ip ospf interface**

Verify OSPF Neighbors (Cont.)

- Use the **show ip ospf neighbor** command to verify that the router has formed an adjacency with its neighboring routers. If the router ID of the neighboring router is not displayed, or if it does not show as being in a state of FULL, the two routers have not formed an OSPFv2 adjacency.

Note: A non-DR or BDR router that has a neighbor relationship with another non-DR or BDR router will display a two-way adjacency instead of full.

- The following command output displays the neighbor table of R1.

```
R1# show ip ospf neighbor
Neighbor ID          Pri      State        Dead Time           Address
                      Interface
3.3.3.3              0         FULL/ -    00:00:35     10.1.1.13
                      GigabitEthernet0/0/1
0.0.0.0              0         -           00:00:31     10.1.1.6
                      GigabitEthernet0/0/0
```

Verify OSPF Neighbors (Cont.)

Two routers may not form an OSPFv2 adjacency if the following occurs:

- The subnet masks do not match, causing the routers to be on separate networks.
- The OSPFv2 Hello or Dead Timers do not match.
- The OSPFv2 Network Types do not match.
- There is a missing or incorrect OSPFv2 network command.

Verify OSPF Protocol Settings

The **show ip protocols** command is a quick way to verify vital OSPF configuration information, as shown in the command output. This includes the OSPFv2 process ID, the router ID, interfaces explicitly configured to advertise OSPF routes, the neighbors the router is receiving updates from, and the default administrative distance, which is 110 for OSPF.

```
R1# show ip protocols
*** IP Routing is NSF aware ***
(output omitted)
Routing Protocol is "ospf 10"
    Outgoing update filter list for all interfaces is not set
    Incoming update filter list for all interfaces is not set
    Router ID 1.1.1.1
    Number of areas in this router is 1. 1 normal 0 stub 0 nssa
    Maximum path: 4
    Routing for Networks:
        Routing on Interfaces Configured Explicitly (Area 0):
            Loopback0
            GigabitEthernet0/0/1
            GigabitEthernet0/0/0
    Routing Information Sources:
        Gateway      Distance          Last Update
        3.3.3.3      110              00:09:30
        2.2.2.2      110              00:09:58
    Distance: (default is 110)
R1#
```

Verify OSPF Interface Settings

The **show ip ospf interface** command provides a detailed list for every OSPFv2-enabled interface. Specify an interface to display the settings of just that interface. This command shows the process ID, the local router ID, the type of network, OSPF cost, DR and BDR information on multiaccess links (not shown), and adjacent neighbors.

```
R1# show ip ospf interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet Address 10.1.1.5/30, Area 0, Attached via Interface Enable
  Process ID 10, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 10

<output omitted>

  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)
R1#
```

Verify OSPF Interface Settings (Cisco)

To get a quick summary of OSPFv2-enabled interfaces, use the **show ip ospf interface brief** command, as shown in the command output. This command is useful for seeing important information including:

- Interfaces are participating in OSPF
- Networks that are being advertised (IP Address/Mask)
- Cost of each link
- Network state
- Number of neighbors on each link

R1# show ip ospf interface brief					
Interface	Nbrs	PID	Area	IP Address/Mask	Cost
			F/C		State
Lo0	10		10	0	10.10.1.1/24
				P2P	0/0
Gi0/0/1		10	0	10.1.1.14/30	30

Implement Port Security

Secure Unused Ports

Layer 2 attacks are some of the easiest for hackers to deploy but these threats can also be mitigated with some common Layer 2 solutions.

- All switch ports (interfaces) should be secured before the switch is deployed for production use. How a port is secured depends on its function.
- A simple method that many administrators use to help secure the network from unauthorized access is to disable all unused ports on a switch. Navigate to each unused port and issue the Cisco IOS **shutdown** command. If a port must be reactivated at a later time, it can be enabled with the **no shutdown** command.
- Switch(config) # **interface range type module/first-number – last-number**

Mitigate MAC Address Table Attack

The simplest and most effective method to prevent MAC address table overflow attacks is to enable port security.

- Port security limits the number of valid MAC addresses allowed on a port. It allows an administrator to manually configure MAC addresses for a port or to permit the switch to dynamically learn a limited number of MAC addresses. When a port configured with port security receives a frame, the source MAC address of the frame is compared to the list of secure source MAC addresses that were manually configured or dynamically learned on the port.
- By limiting the number of permitted MAC addresses on a port to one, port security can be used to control unauthorized access to the network.

Enable Port Security

Port security is enabled with the **switchport port-security** interface configuration command.

Notice in the example, the **switchport port-security** command was rejected. This is because port security can only be configured on manually configured access ports or manually configured trunk ports. By default, Layer 2 switch ports are set to dynamic auto (trunking on). Therefore, in the example, the port is configured with the **switchport mode access** interface configuration command.

Note: Trunk port security is beyond the scope of this course.

```
S1(config)# interface f0/1
S1(config-if)# switchport port-security
Command rejected: FastEthernet0/1 is a dynamic port.
S1(config-if)# switchport mode access
S1(config-if)# switchport port-security
S1(config-if)# end
S1#
```

Enable Port Security (Cont.)

Use the **show port-security interface** command to display the current port security settings for FastEthernet 0/1.

- Notice how port security is enabled, the violation mode is shutdown, and how the maximum number of MAC addresses is 1.
- If a device is connected to the port, the switch will automatically add the device's MAC address as a secure MAC. In this example, no device is connected to the port.

Note: If an active port is configured with the **switchport port-security** command and more than one device is connected to that port, the port will transition to the error-disabled state.

```
S1# show port-security interface f0/1
Port Security           : Enabled
Port Status              : Secure-shutdown
Violation Mode          : Shutdown
Aging Time               : 0 mins
Aging Type               : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses    : 1
Total MAC Addresses       : 0
Configured MAC Addresses : 0
Sticky MAC Addresses      : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0
S1#
```

Enable Port Security (Cont.)

After port security is enabled, other port security specifics can be configured, as shown in the example.

```
S1(config-if)# switchport port-security ?
      aging          Port-security aging commands
      mac-address   Secure mac address
      maximum       Max secure addresses
      violation     Security violation mode
      <cr>
S1(config-if)# switchport port-security
```

Limit and Learn MAC Addresses

To set the maximum number of MAC addresses allowed on a port, use the following command:

```
Switch(config-if)# switchport port-security maximum value
```

- The default port security value is 1.
- The maximum number of secure MAC addresses that can be configured depends the switch and the IOS.
- In this example, the maximum is 8192.

```
S1(config)# interface f0/1
S1(config-if)# switchport port-security maximum ?
<1-8192> Maximum addresses
S1(config-if)# switchport port-security maximum
```

Limit and Learn MAC Addresses

The switch can be configured to learn about MAC addresses on a secure port in one of three ways:

1. Manually Configured: The administrator manually configures a static MAC address(es) by using the following command for each secure MAC address on the port:

```
Switch(config-if)# switchport port-security mac-address mac-address
```

2. Dynamically Learned: When the **switchport port-security** command is entered, the current source MAC for the device connected to the port is automatically secured but is not added to the running configuration. If the switch is rebooted, the port will have to re-learn the device's MAC address.

3. Dynamically Learned – Sticky: The administrator can enable the switch to dynamically learn the MAC address and “stick” them to the running configuration by using the following command:

```
Switch(config-if)# switchport port-security mac-address sticky
```

Saving the running configuration will commit the dynamically learned MAC address to NVRAM.

Limit and Learn MAC Addresses

The example demonstrates a complete port security configuration for FastEthernet 0/1.

- The administrator specifies a maximum of 4 MAC addresses, manually configures one secure MAC address, and then configures the port to dynamically learn additional secure MAC addresses up to the 4 secure MAC address maximum.
- Use the **show port-security interface** and the **show port-security address** command to verify the configuration.

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode access
S1(config-if)# switchport port-security
S1(config-if)# switchport port-security maximum 4
S1(config-if)# switchport port-security mac-address aaaa.bbbb.1234
S1(config-if)# switchport port-security mac-address sticky
S1(config-if)# end
S1# show port-security interface fa0/1
Port Security          : Enabled
Port Status             : Secure-up
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 4
Total MAC Addresses     : 1
Configured MAC Addresses : 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0
S1# show port-security address
               Secure Mac Address Table
-----+-----+-----+-----+-----+
Vlan  Mac Address      Type        Ports      Remaining Age
          (mins)
-----+-----+-----+-----+-----+
1     aaaa.bbbb.1234    SecureConfigured  Fa0/1      -
-----+-----+-----+-----+-----+
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 8192
S1#
```

Port Security Aging

Port security aging can be used to set the aging time for static and dynamic secure addresses on a port and two types of aging are supported per port:

- **Absolute** - The secure addresses on the port are deleted after the specified aging time.
- **Inactivity** - The secure addresses on the port are deleted if they are inactive for a specified time.

Use aging to remove secure MAC addresses on a secure port without manually deleting the existing secure MAC addresses.

- Aging of statically configured secure addresses can be enabled or disabled on a per-port basis.

```
Switch(config-if)# switchport port-security aging {static | time time | type {absolute | inactivity}}
```

Use the **switchport port-security aging** command to enable or disable static aging for the secure port, or to set the aging time or type.

Port Security Aging (Cont.)

The example shows an administrator configuring the aging type to 10 minutes of inactivity.

The **show port-security** command confirms the changes.

```
S1(config)# interface fa0/1
S1(config-if)# switchport port-security aging time 10
S1(config-if)# switchport port-security aging type inactivity
S1(config-if)# end
S1# show port-security interface fa0/1
Port Security          : Enabled
Port Status             : Secure-shutdown
Violation Mode         : Restrict
Aging Time              : 10 mins
Aging Type              : Inactivity
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 4
Total MAC Addresses     : 1
Configured MAC Addresses : 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0050.56be.e4dd:1
Security Violation Count : 1
```

Port Security Violation Modes

If the MAC address of a device attached to a port differs from the list of secure addresses, then a port violation occurs and the port enters the error-disabled state.

- To set the port security violation mode, use the following command:

```
Switch(config-if)# switchport port-security violation {shutdown | restrict | protect}
```

The following table shows how a switch reacts based on the configured violation mode.

Mode	Description
shutdown (default)	The port transitions to the error-disabled state immediately, turns off the port LED, and sends a syslog message. It increments the violation counter. When a secure port is in the error-disabled state, an administrator must re-enable it by entering the shutdown and no shutdown commands.
restrict	The port drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value or increase the maximum value. This mode causes the Security Violation counter to increment and generates a syslog message.
protect	This is the least secure of the security violation modes. The port drops packets with unknown MAC source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value or increase the maximum value. No syslog message is sent.

Port Security Violation Modes (Cont.)

The example shows an administrator changing the security violation to “Restrict”.

The output of the **show port-security interface** command confirms that the change has been made.

```
S1(config)# interface f0/1
S1(config-if)# switchport port-security violation restrict
S1(config-if)# end
S1#
S1# show port-security interface f0/1
Port Security          : Enabled
Port Status             : Secure-shutdown
Violation Mode         : Restrict
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 4
Total MAC Addresses     : 1
Configured MAC Addresses : 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0050.56be.e4dd:1
Security Violation Count : 1
S1#
```

Ports in error-disabled State

When a port is shutdown and placed in the error-disabled state, no traffic is sent or received on that port.

A series of port security related messages display on the console, as shown in the following example.

Note: The port protocol and link status are changed to down and the port LED is turned off.

```
*Sep 20 06:44:54.966: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/18, putting Fa0/18 in err-disable state
*Sep 20 06:44:54.966: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 000c.292b.4c75 on port FastEthernet0/18.
*Sep 20 06:44:55.973: %LINEPROTO-5-PPDOWN: Line protocol on Interface FastEthernet0/18, changed state to down
*Sep 20 06:44:56.971: %LINK-3-UPDOWN: Interface FastEthernet0/18, changed state to down
```

Ports in error-disabled State (Cont.)

- In the example, the **show interface** command identifies the port status as **err-disabled**. The output of the **show port-security interface** command now shows the port status as **secure-shutdown**. The Security Violation counter increments by 1.
- The administrator should determine what caused the security violation. If an unauthorized device is connected to a secure port, the security threat is eliminated before re-enabling the port.
- To re-enable the port, first use the **shutdown** command, then, use the **no shutdown** command.

```
S1# show interface fa0/18
FastEthernet0/18 is down, line protocol is down (err-disabled)
(output omitted)
S1# show port-security interface fa0/18
Port Security          : Enabled
Port Status             : Secure-shutdown
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses: 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : c025.5cd7.ef01:1
Security Violation Count: 1
S1#
```

Verify Port Security

After configuring port security on a switch, check each interface to verify that the port security is set correctly, and check to ensure that the static MAC addresses have been configured correctly.

To display port security settings for the switch, use the **show port-security** command.

- The example indicates that all 24 interfaces are configured with the **switchport port-security** command because the maximum allowed is 1 and the violation mode is shutdown.
- No devices are connected, therefore, the CurrentAddr (Count) is 0 for each interface.

Secure Port (Count)	MaxSecureAddr (Count)	CurrentAddr (Count)	SecurityViolation (Count)	Security Action
Fa0/1	1	0	0	Shutdown
Fa0/2	1	0	0	Shutdown
Fa0/3	1	0	0	Shutdown
(output omitted)				
Fa0/24	1	0	0	Shutdown
Total Addresses in System (excluding one mac per port) : 0				
Max Addresses limit in System (excluding one mac per port) : 4096				
Switch#				

Verify Port Security (Cont.)

Use the **show port-security interface** command to view details for a specific interface, as shown previously and in this example.

```
S1# show port-security interface fastethernet 0/18
Port Security          : Enabled
Port Status             : Secure-up
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0025.83e6.4b01:1
Security Violation Count : 0
S1#
```

Verify Port Security (Cont.)

To verify that MAC addresses are “sticking” to the configuration, use the **show run** command as shown in the example for FastEthernet 0/19.

```
S1# show run | begin interface FastEthernet0/19
interface FastEthernet0/19
switchport mode access
switchport port-security maximum 10
switchport port-security
switchport port-security mac-address sticky
switchport port-security mac-address sticky 0025.83e6.4b02
(output omitted)
S1#
```

Verify Port Security (Cont.)

To display all secure MAC addresses that are manually configured or dynamically learned on all switch interfaces, use the **show port-security address** command as shown in the example.

```
S1# show port-security address
```

```
Secure Mac Address Table
```

Vlan	Mac Address	Type	Ports	Remaining Age (mins)
---	-----	-----	-----	-----
1	0025.83e6.4b01	SecureDynamic	Fa0/18	-
1	0025.83e6.4b02	SecureSticky	Fa0/19	-

```
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 8192
S1#
```

ACL Concepts

What is an ACL?

An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header. By default, a router does not have any ACLs configured. When an ACL is applied to an interface, the router performs the additional task of evaluating all network packets as they pass through the interface to determine if the packet can be forwarded.

- An ACL uses a sequential list of permit or deny statements, known as access control entries (ACEs).

Note: ACEs are also commonly called ACL statements.

- When network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each ACE, in sequential order, to determine if the packet matches one of the ACEs. This process is called packet filtering.

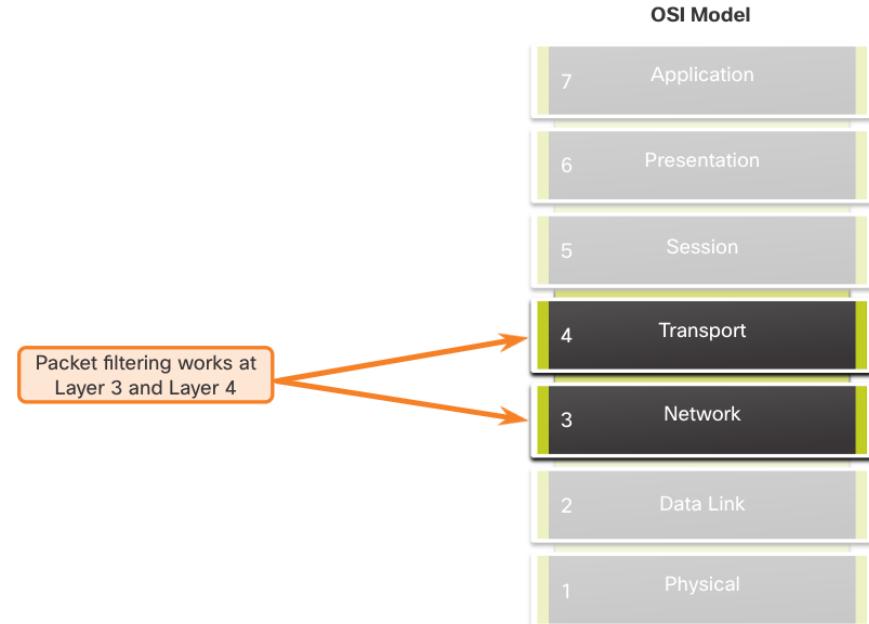
What is an ACL? (Cont.)

Several tasks performed by routers require the use of ACLs to identify traffic:

- Limit network traffic to increase network performance
- Provide traffic flow control
- Provide a basic level of security for network access
- Filter traffic based on traffic type
- Screen hosts to permit or deny access to network services
- Provide priority to certain classes of network traffic

Packet Filtering

- Packet filtering controls access to a network by analyzing the incoming and/or outgoing packets and forwarding them or discarding them based on given criteria.
- Packet filtering can occur at Layer 3 or Layer 4.
- Cisco routers support two types of ACLs:
 - **Standard ACLs** - ACLs only filter at Layer 3 using the source IPv4 address only.
 - **Extended ACLs** - ACLs filter at Layer 3 using the source and / or destination IPv4 address. They can also filter at Layer 4 using TCP, UDP ports, and optional protocol type information for finer control.



ACL Operation

- ACLs define the set of rules that give added control for packets that enter inbound interfaces, packets that relay through the router, and packets that exit outbound interfaces of the router.
- ACLs can be configured to apply to inbound traffic and outbound traffic.

Note: ACLs do not act on packets that originate from the router itself.

- An inbound ACL filters packets before they are routed to the outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the packet is discarded.
- An outbound ACL filters packets after being routed, regardless of the inbound interface.



ACL Operation (Cont.)

When an ACL is applied to an interface, it follows a specific operating procedure. Here are the operational steps used when traffic has entered a router interface with an inbound standard IPv4 ACL configured:

1. The router extracts the source IPv4 address from the packet header.
2. The router starts at the top of the ACL and compares the source IPv4 address to each ACE in a sequential order.
3. When a match is made, the router carries out the instruction, either permitting or denying the packet, and the remaining ACEs in the ACL, if any, are not analyzed.
4. If the source IPv4 address does not match any ACEs in the ACL, the packet is discarded because there is an implicit deny ACE automatically applied to all ACLs.

The last ACE statement of an ACL is always an implicit deny that blocks all traffic. It is hidden and not displayed in the configuration.

Note: An ACL must have at least one permit statement otherwise all traffic will be denied due to the implicit deny ACE statement.

Wildcard Mask Overview

A wildcard mask is similar to a subnet mask in that it uses the ANDing process to identify which bits in an IPv4 address to match. Unlike a subnet mask, in which binary 1 is equal to a match and binary 0 is not a match, in a wildcard mask, the reverse is true.

- An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match.
- Wildcard masks use the following rules to match binary 1s and 0s:
 - **Wildcard mask bit 0** - Match the corresponding bit value in the address
 - **Wildcard mask bit 1** - Ignore the corresponding bit value in the address

Wildcard Mask Overview (Cont.)

Wildcard Mask	Last Octet (in Binary)	Meaning (0 - match, 1 - ignore)
0.0.0.0	00000000	Match all octets.
0.0.0.63	00111111	<ul style="list-style-type: none"> • Match the first three octets • Match the two left most bits of the last octet • Ignore the last 6 bits
0.0.0.15	00001111	<ul style="list-style-type: none"> • Match the first three octets • Match the four left most bits of the last octet • Ignore the last 4 bits of the last octet
0.0.0.240	11111100	<ul style="list-style-type: none"> • Match the first three octets • Ignore the six left most bits of the last octet

Wildcard Mask Types

Wildcard to Match a Host:

- Assume ACL 10 needs an ACE that only permits the host with IPv4 address 192.168.1.1. Recall that “0” equals a match and “1” equals ignore. To match a specific host IPv4 address, a wildcard mask consisting of all zeroes (i.e., 0.0.0.0) is required.
- When the ACE is processed, the wildcard mask will permit only the 192.168.1.1 address. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.1.1 0.0.0.0**.

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.0	00000000.00000000.00000000.00000000
Permitted IPv4 Address	192.168.1.1	11000000.10101000.00000001.00000001

Wildcard Mask Types (Cont.)

Wildcard Mask to Match an IPv4 Subnet

- ACL 10 needs an ACE that permits all hosts in the 192.168.1.0/24 network. The wildcard mask 0.0.0.255 stipulates that the very first three octets must match exactly but the fourth octet does not.
- When processed, the wildcard mask 0.0.0.255 permits all hosts in the 192.168.1.0/24 network. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.1.0 0.0.0.255**.

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.255	00000000.00000000.00000000.11111111
Permitted IPv4 Address	192.168.1.0/24	11000000.10101000.00000001.00000000

Wildcard Mask Types (Cont.)

Wildcard Mask to Match an IPv4 Address Range

- ACL 10 needs an ACE that permits all hosts in the 192.168.16.0/24, 192.168.17.0/24, ..., 192.168.31.0/24 networks.
- When processed, the wildcard mask 0.0.15.255 permits all hosts in the 192.168.16.0/24 to 192.168.31.0/24 networks. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.16.0 0.0.15.255**.

	Decimal	Binary
IPv4 address	192.168.16.0	11000000.10101000.00010000.00000000
Wildcard Mask	0.0.15.255	00000000.00000000.00001111.11111111
Permitted IPv4 Address	192.168.16.0/24 to 192.168.31.0/24	11000000.10101000.00010000.00000000 11000000.10101000.00010000.00000000

Wildcard Mask Calculation

Calculating wildcard masks can be challenging. One shortcut method is to subtract the subnet mask from 255.255.255.255. Some examples:

- Assume you wanted an ACE in ACL 10 to permit access to all users in the 192.168.3.0/24 network. To calculate the wildcard mask, subtract the subnet mask (255.255.255.0) from 255.255.255.255. This produces the wildcard mask 0.0.0.255. The ACE would be **access-list 10 permit 192.168.3.0 0.0.0.255**.
- Assume you wanted an ACE in ACL 10 to permit network access for the 14 users in the subnet 192.168.3.32/28. Subtract the subnet (i.e., 255.255.255.240) from 255.255.255.255. This produces the wildcard mask 0.0.0.15. The ACE would be **access-list 10 permit 192.168.3.32 0.0.0.15**.
- Assume you needed an ACE in ACL 10 to permit only networks 192.168.10.0 and 192.168.11.0. These two networks could be summarized as 192.168.10.0/23 which is a subnet mask of 255.255.254.0. Subtract 255.255.254.0 subnet mask from 255.255.255.255. This produces the wildcard mask 0.0.1.255. The ACE would be **access-list 10 permit 192.168.10.0 0.0.1.255**.

Wildcard Mask Keywords

The Cisco IOS provides two keywords to identify the most common uses of wildcard masking. The two keywords are:

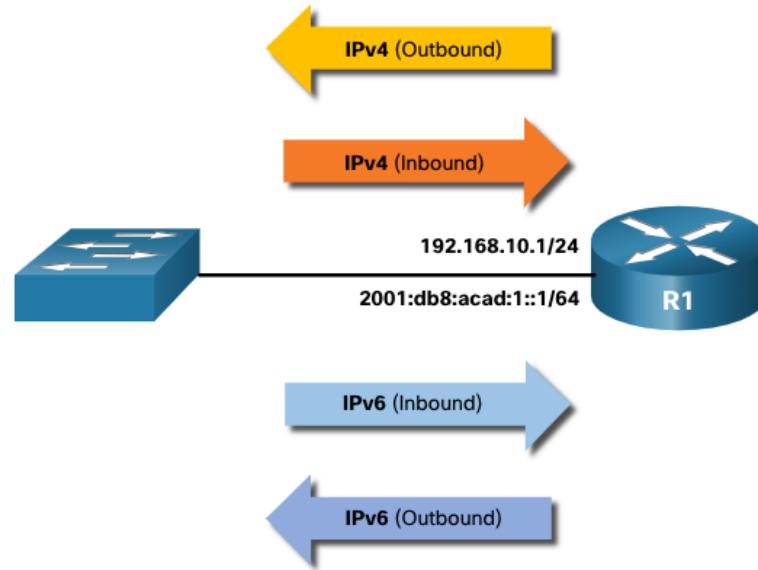
- **host** - This keyword substitutes for the 0.0.0.0 mask. This mask states that all IPv4 address bits must match to filter just one host address.
- **any** - This keyword substitutes for the 255.255.255.255 mask. This mask says to ignore the entire IPv4 address or to accept any addresses.

Limited Number of ACLs per Interface

There is a limit on the number of ACLs that can be applied on a router interface. For example, a dual-stacked (i.e., IPv4 and IPv6) router interface can have up to four ACLs applied, as shown in the figure.

Specifically, a router interface can have:

- One outbound IPv4 ACL.
- One inbound IPv4 ACL.
- One inbound IPv6 ACL.
- One outbound IPv6 ACL.



Note: ACLs do not have to be configured in both directions.

The number of ACLs and their direction applied to the interface will depend on the security policy of the organization.

Standard and Extended ACLs

There are two types of IPv4 ACLs:

- **Standard ACLs** - These permit or deny packets based only on the source IPv4 address.
- **Extended ACLs** - These permit or deny packets based on the source IPv4 address and destination IPv4 address, protocol type, source and destination TCP or UDP ports and more.

Numbered and Named ACLs

Numbered ACLs

- ACLs numbered 1-99, or 1300-1999 are standard ACLs, while ACLs numbered 100-199, or 2000-2699 are extended ACLs.

```
R1(config)# access-list ?
<1-99> IP standard access list
<100-199> IP extended access list
<1100-1199> Extended 48-bit MAC address access list
<1300-1999> IP standard access list (expanded range)
<200-299> Protocol type-code access list
<2000-2699> IP extended access list (expanded range)
<700-799> 48-bit MAC address access list
rate-limit Simple rate-limit specific access list
template Enable IP template acls
Router(config)# access-list
```

Numbered and Named ACLs (Con)

Named ACLs

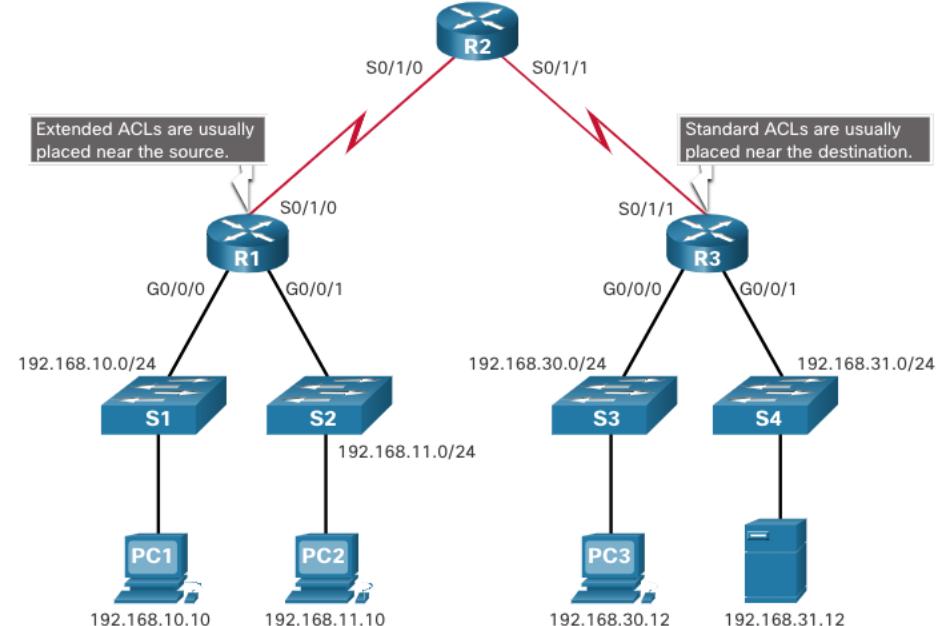
- Named ACLs are the preferred method to use when configuring ACLs. Specifically, standard and extended ACLs can be named to provide information about the purpose of the ACL. For example, naming an extended ACL FTP-FILTER is far better than having a numbered ACL 100.
- The **ip access-list** global configuration command is used to create a named ACL, as shown in the following example.

```
R1(config)# ip access-list extended FTP-FILTER
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp-data
R1(config-ext-nacl)#

```

Where to Place ACLs

- Every ACL should be placed where it has the greatest impact on efficiency.
- Extended ACLs should be located as close as possible to the source of the traffic to be filtered.
- Standard ACLs should be located as close to the destination as possible.



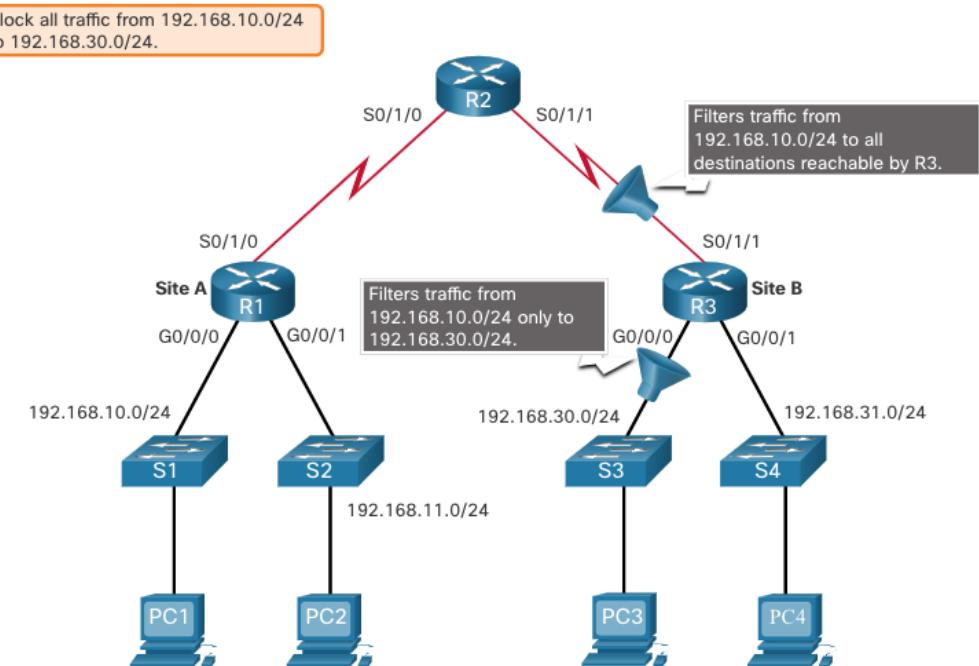
Where to Place ACLs (Cont.)

Factors Influencing ACL Placement	Explanation
The extent of organizational control	Placement of the ACL can depend on whether or not the organization has control of both the source and destination networks.
Bandwidth of the networks involved	It may be desirable to filter unwanted traffic at the source to prevent transmission of bandwidth-consuming traffic.
Ease of configuration	<ul style="list-style-type: none">• It may be easier to implement an ACL at the destination, but traffic will use bandwidth unnecessarily.• An extended ACL could be used on each router where the traffic originated. This would save bandwidth by filtering the traffic at the source, but it would require creating extended ACLs on multiple routers.

Standard ACL Placement Example

In the figure, the administrator wants to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.

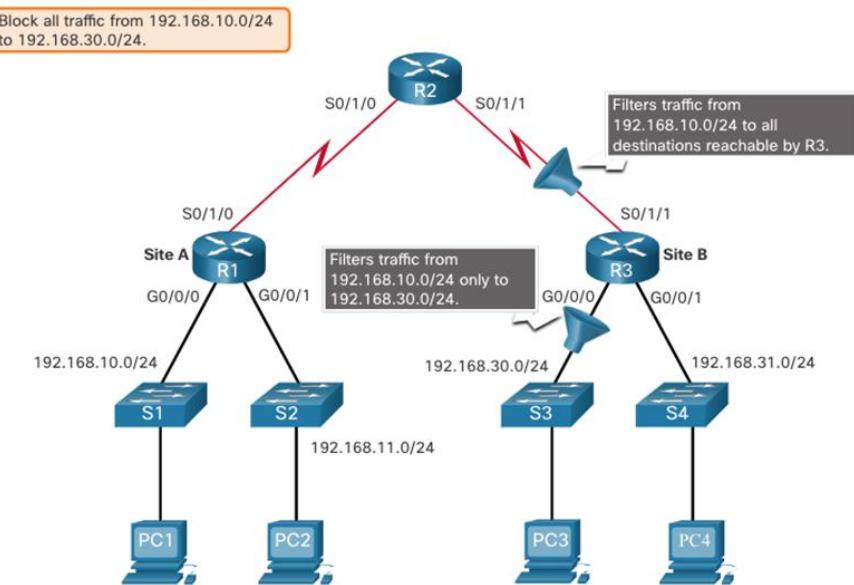
Following the basic placement guidelines, the administrator would place a standard ACL on router R3.



Standard ACL Placement Example (Cont.)

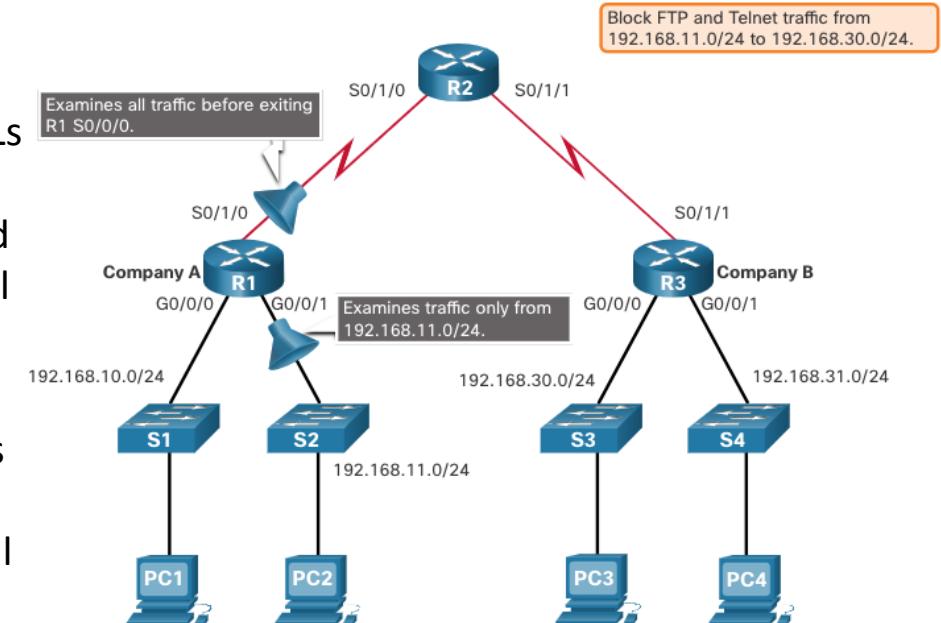
There are two possible interfaces on R3 to apply the standard ACL:

- R3 S0/1/1 interface (inbound)** - The standard ACL can be applied inbound on the R3 S0/1/1 interface to deny traffic from .10 network. However, it would also filter .10 traffic to the 192.168.31.0/24 (.31 in this example) network. Therefore, the standard ACL should not be applied to this interface.
- R3 G0/0 interface (outbound)** - The standard ACL can be applied outbound on the R3 G0/0/0 interface. This will not affect other networks that are reachable by R3. Packets from .10 network will still be able to reach the .31 network. This is the best interface to place the standard ACL to meet the traffic requirements.



Extended ACL Placement Example

- Extended ACLs should be located as close to the source as possible.
- However, the organization can only place ACLs on devices that they control. Therefore, the extended ACL placement must be determined in the context of where organizational control extends.
- In the figure, for example, Company A wants to deny Telnet and FTP traffic to Company B's 192.168.30.0/24 network from their 192.168.11.0/24 network, while permitting all other traffic.



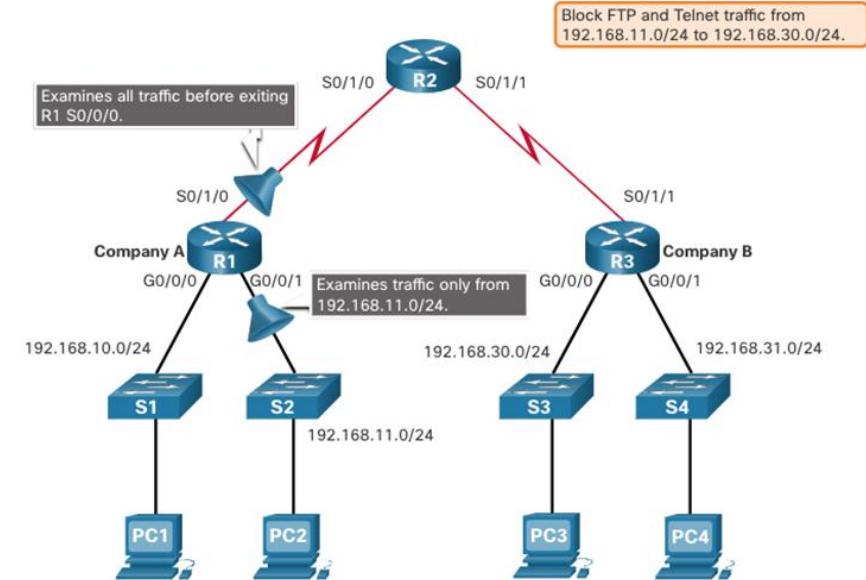
Extended ACL Placement Example (Cont.)

An extended ACL on R3 would accomplish the task, but the administrator does not control R3. In addition, this solution allows unwanted traffic to cross the entire network, only to be blocked at the destination.

The solution is to place an extended ACL on R1 that specifies both source and destination addresses.

There are two possible interfaces on R1 to apply the extended ACL:

- R1 S0/1/0 interface (outbound)** - The extended ACL can be applied outbound on the S0/1/0 interface. This solution will process all packets leaving R1 including packets from 192.168.10.0/24.
- R1 G0/0/1 interface (inbound)** - The extended ACL can be applied inbound on the G0/0/1 and 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.



Named Extended IPv4 ACL Syntax

Naming an ACL makes it easier to understand its function. To create a named extended ACL, use the **ip access-list extended** configuration command.

In the example, a named extended ACL called NO-FTP-ACCESS is created and the prompt changed to named extended ACL configuration mode. ACE statements are entered in the named extended ACL sub configuration mode.

```
Router(config)# ip access-list extended access-list-name
```

```
R1(config)# ip access-list extended NO-FTP-ACCESS
R1(config-ext-nacl)#

```

ACLs for IPv4 Configuration

Numbered Standard IPv4 ACL Syntax

To create a numbered standard ACL, use the **access-list** command.

```
Router(config)# access-list access-list-number {deny | permit | remark text} source [source-wildcard]  
[log]
```

Parameter	Description
<i>access-list-number</i>	Number range is 1 to 99 or 1300 to 1999
deny	Denies access if the condition is matched
permit	Permits access if the condition is matched
remark text	(Optional) text entry for documentation purposes
source	Identifies the source network or host address to filter
source-wildcard	(Optional) 32-bit wildcard mask that is applied to the source
log	(Optional) Generates and sends an informational message when the ACE is matched

Note: Use the **no access-list access-list-number** global configuration command to remove a numbered standard ACL.

Named Standard IPv4 ACL Syntax

To create a named standard ACL, use the **ip access-list standard** command.

- ACL names are alphanumeric, case sensitive, and must be unique.
- Capitalizing ACL names is not required but makes them stand out when viewing the running-config output.

```
Router(config)# ip access-list standard access-list-name
```

```
R1(config)# ip access-list standard NO-ACCESS
R1(config-std-nacl)# ?
Standard Access List configuration commands:
  <1-2147483647>  Sequence Number
  default          Set a command to its defaults
  deny             Specify packets to reject
  exit              Exit from access-list configuration mode
  no                Negate a command or set its defaults
  permit            Specify packets to forward
  remark            Access list entry comment
R1(config-std-nacl)#

```

Apply a Standard IPv4 ACL

After a standard IPv4 ACL is configured, it must be linked to an interface or feature.

- The **ip access-group** command is used to bind a numbered or named standard IPv4 ACL to an interface.
- To remove an ACL from an interface, first enter the **no ip access-group** interface configuration command.

```
Router(config-if) # ip access-group {access-list-number | access-list-name} {in | out}
```

Numbered Standard ACL Example

The example ACL permits traffic from host 192.168.10.10 and all hosts on the 192.168.20.0/24 network out interface serial 0/1/0 on router R1.

```
R1(config)# access-list 10 remark ACE permits ONLY host 192.168.10.10 to the internet
R1(config)# access-list 10 permit host 192.168.10.10
R1(config)# do show access-lists
Standard IP access list 10
    10 permit 192.168.10.10
R1(config)#
R1(config)# access-list 10 remark ACE permits all host in LAN 2
R1(config)# access-list 10 permit 192.168.20.0 0.0.0.255
R1(config)# do show access-lists
Standard IP access list 10
    10 permit 192.168.10.10
    20 permit 192.168.20.0, wildcard bits 0.0.0.255
R1(config)#
R1(config)# interface Serial 0/1/0
R1(config-if)# ip access-group 10 out
R1(config-if)# end
R1#
```

Numbered Standard ACL Example (Cont.)

- Use the **show running-config** command to review the ACL in the configuration.
- Use the **show ip interface** command to verify the ACL is applied to the interface.

```
R1# show run | section access-list
access-list 10 remark ACE permits host 192.168.10.10
access-list 10 permit 192.168.10.10
access-list 10 remark ACE permits all host in LAN 2
access-list 10 permit 192.168.20.0 0.0.0.255
R1#
```

```
R1# show ip int Serial 0/1/0 | include access list
Outgoing Common access list is not set
Outgoing access list is 10
Inbound Common access list is not set
Inbound access list is not set
R1#
```

Named Standard ACL Example

The example ACL permits traffic from host 192.168.10.10 and all hosts on the 192.168.20.0/24 network out interface serial 0/1/0 on router R1.

```
R1(config)# no access-list 10
R1(config)# ip access-list standard PERMIT-ACCESS
R1(config-std-nacl)# remark ACE permits host 192.168.10.10
R1(config-std-nacl)# permit host 192.168.10.10
R1(config-std-nacl)#

```

```
R1(config-std-nacl)# remark ACE permits host 192.168.10.10
R1(config-std-nacl)# permit host 192.168.10.10
R1(config-std-nacl)# remark ACE permits all hosts in LAN 2
R1(config-std-nacl)# permit 192.168.20.0 0.0.0.255
R1(config-std-nacl)# exit
R1(config)#

```

```
R1(config)# interface Serial 0/1/0
R1(config-if)# ip access-group PERMIT-ACCESS out
R1(config-if)# end
R1#

```

Named Standard ACL Example (Cont.)

- Use the **show access-list** command to review the ACL in the configuration.
- Use the **show ip interface** command to verify the ACL is applied to the interface.

```
R1# show access-lists
Standard IP access list PERMIT-ACCESS
    10 permit 192.168.10.10
    20 permit 192.168.20.0, wildcard bits 0.0.0.255
R1# show run | section ip access-list
ip access-list standard PERMIT-ACCESS
    remark ACE permits host 192.168.10.10
    permit 192.168.10.10
    remark ACE permits all hosts in LAN 2
    permit 192.168.20.0 0.0.0.255
R1#
R1# show ip int Serial 0/1/0 | include access list
    Outgoing Common access list is not set
    Outgoing access list is PERMIT-ACCESS
    Inbound Common access list is not set
    Inbound access list is not set
R1#
```

ACL Statistics

The **show access-lists** command in the example shows statistics for each statement that has been matched.

- The deny ACE has been matched 20 times and the permit ACE has been matched 64 times.
- Note that the implied deny any statement does not display any statistics. To track how many implicit denied packets have been matched, you must manually configure the **deny any** command.
- Use the **clear access-list counters** command to clear the ACL statistics.

```
R1# show access-lists
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10  (20 matches)
    20 permit 192.168.10.0, wildcard bits 0.0.0.255 (64 matches)
R1# clear access-list counters NO-ACCESS
R1# show access-lists
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

The access-class Command

A standard ACL can secure remote administrative access to a device using the vty lines by implementing the following two steps:

- Create an ACL to identify which administrative hosts should be allowed remote access.
- Apply the ACL to incoming traffic on the vty lines.

```
R1(config-line)# access-class {access-list-number | access-list-name} { in | out }
```

Secure VTY Access Example

This example demonstrates how to configure an ACL to filter vty traffic.

- First, a local database entry for a user **ADMIN** and password **class** is configured.
- The vty lines on R1 are configured to use the local database for authentication, permit SSH traffic, and use the ADMIN-HOST ACL to restrict traffic.

```
R1(config)# username ADMIN secret class
R1(config)# ip access-list standard ADMIN-HOST
R1(config-std-nacl)# remark This ACL secures incoming vty lines
R1(config-std-nacl)# permit 192.168.10.10
R1(config-std-nacl)# deny any
R1(config-std-nacl)# exit
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input telnet
R1(config-line)# access-class ADMIN-HOST in
R1(config-line)# end
R1#
```

Verify the VTY Port is Secured

After an ACL to restrict access to the vty lines is configured, it is important to verify it works as expected.

To verify the ACL statistics, issue the **show access-lists** command.

- The match in the permit line of the output is a result of a successful SSH connection by host with IP address 192.168.10.10.
- The match in the deny statement is due to the failed attempt to create a SSH connection from a device on another network.

```
R1#  
Oct  9 15:11:19.544: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 192.168.10.10]  
[localport: 23] at 15:11:19 UTC Wed Oct 9 2019  
R1# show access-lists  
Standard IP access list ADMIN-HOST  
    10 permit 192.168.10.10  (2 matches)  
    20 deny   any   (2 matches)  
R1#
```

Extended ACLs

Extended ACLs provide a greater degree of control. They can filter on source address, destination address, protocol (i.e., IP, TCP, UDP, ICMP), and port number.

Extended ACLs can be created as:

- **Numbered Extended ACL** - Created using the **access-list *access-list-number*** global configuration command.
- **Named Extended ACL** - Created using the **ip access-list extended *access-list-name***.

Configure Extended IPv4 Access Lists

Protocols and Ports

Protocol Options

Extended ACLs can filter on internet protocols and ports. Use the ? to get help when entering a complex ACE. The four highlighted protocols are the most popular options.

```
R1(config)# access-list 100 permit ?
<0-255>      An IP protocol number
ahp            Authentication Header Protocol
dvmrp          dvmrp
eigrp          Cisco's EIGRP routing protocol
esp             Encapsulation Security Payload
gre             Cisco's GRE tunneling
icmp            Internet Control Message Protocol
igmp            Internet Gateway Message Protocol
ip              Any Internet Protocol
ipinip          IP in IP tunneling
nos             KA9Q NOS compatible IP over IP tunneling
object-group   Service object group
ospf            OSPF routing protocol
pcp             Payload Compression Protocol
pim             Protocol Independent Multicast
tcp             Transmission Control Protocol
udp             User Datagram Protocol
R1(config)# access-list 100 permit
```

Protocols and Ports (Cont.)

Selecting a protocol influences port options. Many TCP port options are available, as shown in the output.

```
R1(config)# access-list 100 permit tcp any any eq ?  
<0-65535>  Port number  
bgp      Border Gateway Protocol (179)  
chargen  Character generator (19)  
cmd      Remote commands (rcmd, 514)  
daytime  Daytime (13)  
discard  Discard (9)  
domain   Domain Name Service (53)  
echo     Echo (7)  
exec    Exec (rsh, 512)  
finger   Finger (79)  
ftp      File Transfer Protocol (21)  
ftp-data FTP data connections (20)  
gopher   Gopher (70)  
hostname NIC hostname server (101)  
ident    Ident Protocol (113)  
irc      Internet Relay Chat (194)  
klogin   Kerberos login (543)  
kshell   Kerberos shell (544)  
login    Login (rlogin, 513)  
lpd      Printer service (515)  
msrpc   MS Remote Procedure Call (135)  
nntp    Network News Transport Protocol (119)  
onep-plain OneP Cleartext (15001)  
onep-tls  OneP TLS (15002)  
pim-auto-rp PIM Auto-RP (496)  
pop2    Post Office Protocol v2 (109)  
pop3    Post Office Protocol v3 (110)  
smtp    Simple Mail Transport Protocol (25)  
sunrpc  Sun Remote Procedure Call (111)  
syslog  Syslog (514)  
tacacs  TAC Access Control System (49)  
talk    Talk (517)  
telnet  Telnet (23)  
time    Time (37)  
uucp    Unix-to-Unix Copy Program (540)  
whos    Nicname (43)  
www    World Wide Web (HTTP - 80)
```

Protocols and Port Numbers

Configuration Examples

Extended ACLs can filter on different port number and port name options.

This example configures an extended ACL 100 to filter HTTP traffic. The first ACE uses the **www** port name. The second ACE uses the port number **80**. Both ACEs achieve exactly the same result.

```
R1(config)# access-list 100 permit tcp any any eq www
!or...
R1(config)# access-list 100 permit tcp any any eq 80
```

Configuring the port number is required when there is not a specific protocol name listed such as SSH (port number 22) or an HTTPS (port number 443), as shown in the next example.

```
R1(config)# access-list 100 permit tcp any any eq 22
R1(config)# access-list 100 permit tcp any any eq 443
R1(config)#
```

Apply a Numbered Extended IPv4 ACL

In this example, the ACL permits both HTTP and HTTPS traffic from the 192.168.10.0 network to go to any destination.

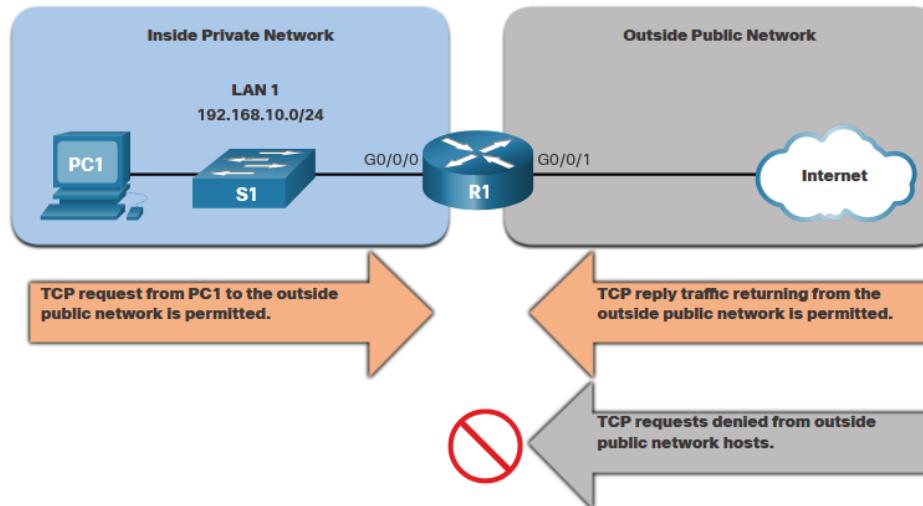
Extended ACLs can be applied in various locations. However, they are commonly applied close to the source. Here ACL 110 is applied inbound on the R1 G0/0/0 interface.

```
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq www
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# interface g0/0/0
R1(config-if)# ip access-group 110 in
R1(config-if)# exit
R1(config)#
```

TCP Established Extended ACL

TCP can also perform basic stateful firewall services using the TCP **established** keyword.

- The **established** keyword enables inside traffic to exit the inside private network and permits the returning reply traffic to enter the inside private network.
- TCP traffic generated by an outside host and attempting to communicate with an inside host is denied.



TCP Established Extended ACL (Cont.)

- ACL 120 is configured to only permit returning web traffic to the inside hosts. The ACL is then applied outbound on the R1 G0/0/0 interface.
- The **show access-lists** command shows that inside hosts are accessing the secure web resources from the internet.

Note: A match occurs if the returning TCP segment has the ACK or reset (RST) flag bits set, indicating that the packet belongs to an existing connection.

```
R1(config)# access-list 120 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)# interface g0/0/0
R1(config-if)# ip access-group 120 out
R1(config-if)# end
R1# show access-lists
Extended IP access list 110
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (657 matches)
Extended IP access list 120
    10 permit tcp any 192.168.10.0 0.0.0.255 established (1166 matches)
R1#
```

Named Extended IPv4 ACL Syntax

Naming an ACL makes it easier to understand its function. To create a named extended ACL, use the **ip access-list extended** configuration command.

In the example, a named extended ACL called NO-FTP-ACCESS is created and the prompt changed to named extended ACL configuration mode. ACE statements are entered in the named extended ACL sub configuration mode.

```
Router(config)# ip access-list extended access-list-name
```

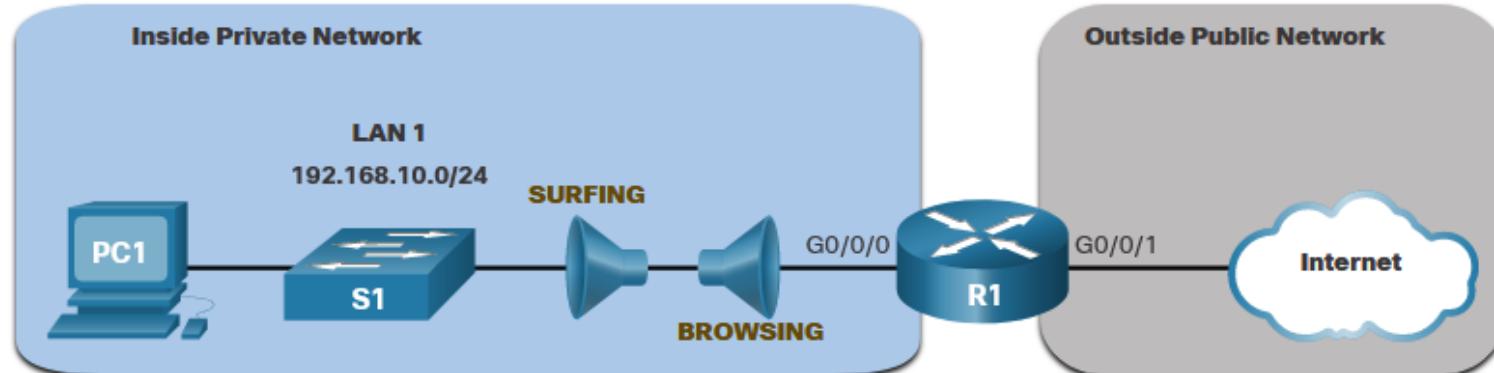
```
R1(config)# ip access-list extended NO-FTP-ACCESS
R1(config-ext-nacl)#

```

Named Extended IPv4 ACL Example

The topology below is used to demonstrate configuring and applying two named extended IPv4 ACLs to an interface:

- **SURFING** - This will permit inside HTTP and HTTPS traffic to exit to the internet.
- **BROWSING** - This will only permit returning web traffic to the inside hosts while all other traffic exiting the R1 G0/0/0 interface is implicitly denied.



Named Extended IPv4 ACL Example (

- The SURFING ACL permits HTTP and HTTPS traffic from inside users to exit the G0/0/1 interface connected to the internet. Web traffic returning from the internet is permitted back into the inside private network by the BROWSING ACL.
- The SURFING ACL is applied inbound and the BROWSING ACL is applied outbound on the R1 G0/0/0 interface.

```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# Remark Permits inside HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)#
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# Remark Only permit returning HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
R1(config-if)# end
R1# show access-lists
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (124 matches)
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established (369 matches)
R1#
```

Named Extended IPv4 ACL Example (1)

The **show access-lists** command is used to verify the ACL statistics. Notice that the permit secure HTTPS counters (i.e., eq 443) in the SURFING ACL and the return established counters in the BROWSING ACL have increased.

```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 19.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

Edit Extended ACLs

An extended ACL can be edited using a text editor when many changes are required. Or, if the edit applies to one or two ACEs, then sequence numbers can be used.

Example:

- The ACE sequence number 10 in the SURFING ACL has an incorrect source IP networks address.

```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 19.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

Edit Extended ACLs (Cont.)

- To correct this error the original statement is removed with the **no sequence_#** command and the corrected statement is added replacing the original statement.
- The **show access-lists** command output verifies the configuration change.

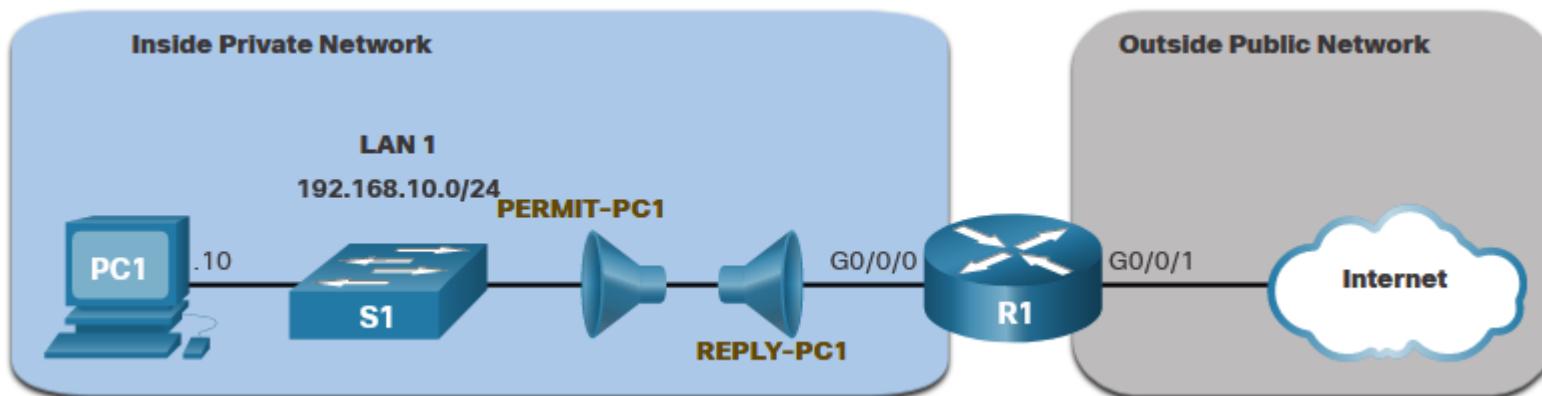
```
R1# configure terminal
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# no 10
R1(config-ext-nacl)# 10 permit tcp 192.168.10.0 0.0.0.255 any eq www
R1(config-ext-nacl)# end
```

```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

Another Extended IPv4 ACL Example

Two named extended ACLs will be created:

- **PERMIT-PC1** - This will only permit PC1 TCP access to the internet and deny all other hosts in the private network.
- **REPLY-PC1** - This will only permit specified returning TCP traffic to PC1 implicitly deny all other traffic.



Another Extended IPv4 ACL Example

- The **PERMIT-PC1** ACL permits PC1 (192.168.10.10) TCP access to the FTP, SSH, Telnet, DNS , HTTP, and HTTPS traffic.
- The **REPLY-PC1** ACL will permit return traffic to PC1.
- The **PERMIT-PC1** ACL is applied inbound and the **REPLY-PC1** ACL applied outbound on the R1 G0/0/0 interface.

```
R1(config)# ip access-list extended PERMIT-PC1
R1(config-ext-nacl)# Remark Permit PC1 TCP access to internet
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 20
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 21
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 22
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 23
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 53
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 80
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 443
R1(config-ext-nacl)# deny ip 192.168.10.0 0.0.0.255 any
R1(config-ext-nacl)# exit
R1(config)#
R1(config)# ip access-list extended REPLY-PC1
R1(config-ext-nacl)# Remark Only permit returning traffic to PC1
R1(config-ext-nacl)# permit tcp any host 192.168.10.10 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group PERMIT-PC1 in
R1(config-if)# ip access-group REPLY-PC1 out
R1(config-if)# end
R1#
```

Verify Extended ACLs

The **show ip interface** command is used to verify the ACL on the interface and the direction in which it was applied.

```
R1# show ip interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up (connected)
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is REPLY-PC1
  Inbound access list is PERMIT-PC1
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is disabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
R1#
R1# show ip interface g0/0/0 | include access list
Outgoing access list is REPLY-PC1
Inbound access list is PERMIT-PC1
R1#
```

Verify Extended ACLs (Cont.)

The **show access-lists** command can be used to confirm that the ACLs work as expected. The command displays statistic counters that increase whenever an ACE is matched.

Note: Traffic must be generated to verify the operation of the ACL.

```
R1# show access-lists
Extended IP access list PERMIT-PC1
10 permit tcp host 192.168.10.10 any eq 20
20 permit tcp host 192.168.10.10 any eq ftp
30 permit tcp host 192.168.10.10 any eq 22
40 permit tcp host 192.168.10.10 any eq telnet
50 permit tcp host 192.168.10.10 any eq domain
60 permit tcp host 192.168.10.10 any eq www
70 permit tcp host 192.168.10.10 any eq 443
80 deny ip 192.168.10.0 0.0.0.255 any
Extended IP access list REPLY-PC1
10 permit tcp any host 192.168.10.10 established
R1#
```

Verify Extended ACLs (Cont.)

The **show running-config** command can be used to validate what was configured. The command also displays configured remarks.

```
R1# show running-config | begin ip access-list
ip access-list extended PERMIT-PC1
remark Permit PC1 TCP access to internet
permit tcp host 192.168.10.10 any eq 20
permit tcp host 192.168.10.10 any eq ftp
permit tcp host 192.168.10.10 any eq 22
permit tcp host 192.168.10.10 any eq telnet
permit tcp host 192.168.10.10 any eq domain
permit tcp host 192.168.10.10 any eq www
permit tcp host 192.168.10.10 any eq 443
deny ip 192.168.10.0 0.0.0.255 any
ip access-list extended REPLY-PC1
remark Only permit returning traffic to PC1
permit tcp any host 192.168.10.10 established
!
```

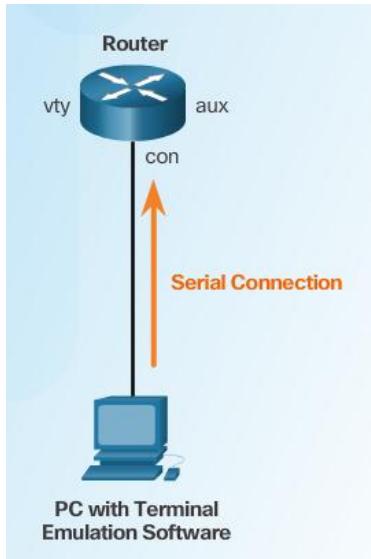
Securing Network Devices

Three Areas of Router Security

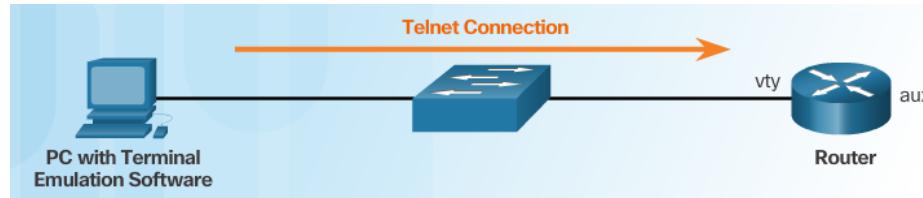


Secure Local and Remote Access

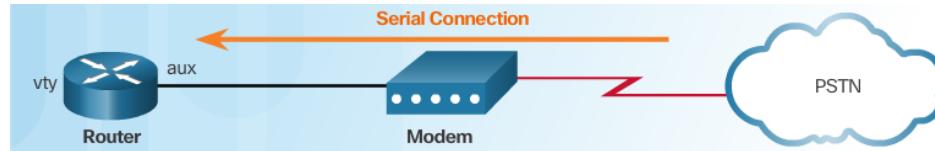
Local Access



Remote Access Using Telnet



Remote Access Using Modem and Aux Port



Increasing Access Security

```
R1(config)# security passwords min-length 10
R1(config)# service password-encryption
R1(config)# line vty 0 4
R1(config-line)# exec-timeout 3 30
R1(config-line)# line console 0
R1(config-line)# exec-timeout 3 30
```

```
R1(config)# service password-encryption
R1(config)# exit
R1# show running-config

<output omitted>

line con 0
exec-timeout 3 30
password 7 094F471A1A0A
login
line aux 0
exec-timeout 3 30
password 7 094F471A1A0A
login
line vty 0 4
password 7 094F471A1A0A
login
```

Cisco Cracker

094F471A1A0A

Crack it

Password = Cisco

Secret Password Algorithms

Guidelines:

- Configure all secret passwords using type 8 or type 9 passwords
- Use the enable algorithm-type command syntax to enter an unencrypted password

```
Router(config)#
enable algorithm-type {md5 | scrypt | sha256} secret unencrypted-password
```

- Use the username name algorithm-type command to specify type 9 encryption

```
Router(config)#
username name algorithm-type {md5 | scrypt | sha256} secret unencrypted-password
```

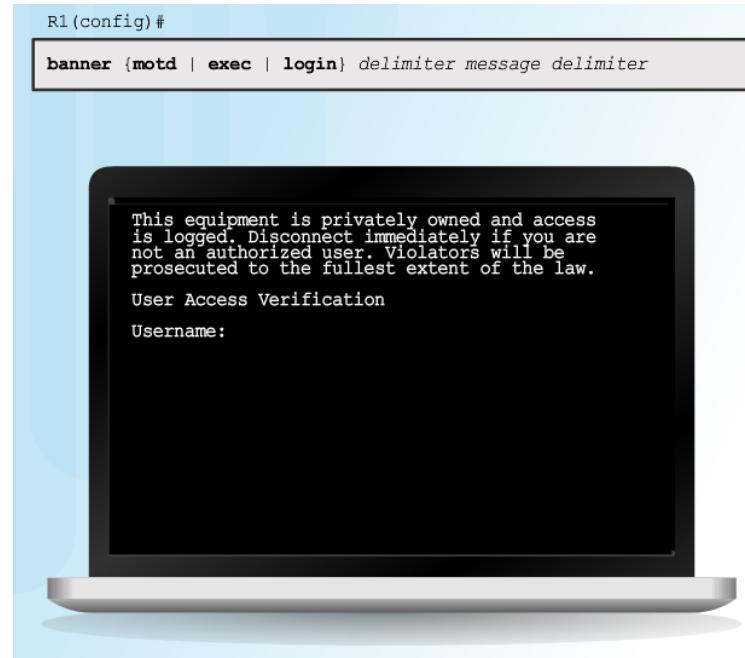
Securing Line Access

```
R1(config)# username Bob algorithm-type scrypt secret cisco54321
R1(config)# line con 0
R1(config-line)# no password
R1(config-line)# login local
R1(config-line)# exit
R1(config)# line aux 0
R1(config-line)# no password
R1(config-line)# login local
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
```

Enhancing the Login Process

Virtual login security enhancements:

- Implement delays between successive login attempts
- Enable login shutdown if DoS attacks are suspected
- Generate system-logging messages for login detection



Configuring Login Enhancement Features

```
R1(config)#
  login block-for seconds attempts tries within seconds
```

```
R1(config)#
  login quiet-mode access-class {acl-name|acl-number}
```

```
R1(config)#
  login delay seconds
```

```
R1(config)#
  login on-success log [every] login
```

```
R1(config)#
  login on-failure log [every] login
```

Enable Login Enhancements

Command Syntax: **login block-for**

```
router(config)#  
login block-for seconds attempts tries within seconds
```

```
R1(config)# login block-for 120 attempts 5 within 60
```

Example: **login quiet-mode access-class**

```
R1(config)# ip access-list standard PERMIT-ADMIN  
R1(config-std-nacl)# remark Permit only Administrative hosts  
R1(config-std-nacl)# permit 192.168.10.10  
R1(config-std-nacl)# permit 192.168.11.10  
R1(config-std-nacl)# exit  
R1(config)# login quiet-mode access-class PERMIT-ADMIN
```

Example: **login delay**

```
R1(config)# login delay 3
```

Logging Failed Attempts

Generate Login Syslog Messages

```
R1(config)# login on-success log [every login]
R1(config)# login on-failure log [every login]
R1(config)# security authentication failure rate threshold-rate log
```

Example: show login failures

```
R1# show login failures
Total failed logins: 22
Detailed information about last 50 failures

Username      SourceIPAddr      lPort Count TimeStamp
admin         1.1.2.1           23    5     15:38:54 UTC Wed Dec 10 2008
Admin         10.10.10.10       23   13     15:58:43 UTC Wed Dec 10 2008
admin         10.10.10.10       23    3     15:57:14 UTC Wed Dec 10 2008
cisco         10.10.10.10       23    1     15:57:21 UTC Wed Dec 10 2008

R1#
```

Steps for Configuring SSH

Example SSH Configuration

```
R1# conf t
R1(config)# ip domain-name span.com
R1(config)# crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.span.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)

R1(config)#
*Feb 16 21:18:41.977: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)# ip ssh version 2
R1(config)# username Bob algorithm-type scrypt secret cisco54321
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# end
R1#
```

Example Verification of SSH

```
R1# show crypto key mypubkey rsa
% Key pair was generated at: 21:18:41 UTC Feb 16 2015
Key name: R1.span.com
Key type: RSA KEYS
Storage Device: not specified
Usage: General Purpose Key
Key is not exportable.
Key Data:
 30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CF35DB
 A58A1BDB F7C7E600 F189C2F3 2EC6E584 D923EE5B 71841D98 B5472A03 D19CD620
 ED125825 5A58412B B7F29234 DE2A1809 6C421AC3 07F298E6 80BE149D 2A262E13
 7488SDAF CAC8F187 B11111AF A413E76F 6C157CDF DFEF0D82 2961B58C BE1CAD21
 176E82B9 6D81F893 06E66C93 94E1C508 887462F6 90AC63CE 5E169845 C1020301 0001
% Key pair was generated at: 21:18:42 UTC Feb 16 2015
Key name: R1.span.com.server
Key type: RSA KEYS
Temporary key
  Usage: Encryption Key
  Key is not exportable.
  Key Data:
    307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00AB914D 8172DFBE
    DE57ACAA 78844239 1F3B5942 3943AC0D F54E7746 3895CF54 606C3961 8A44FEB3
    1A019F27 D9E71AAE FC73F423 A59CB8F5 50289272 3392CEBC 4C3CBD6D DB9233DE
    9DDDD9DAD 79D56165 4293AA62 FD1CBAB2 7AB859DC 2890C795 ED020301 0001
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# crypto key zeroize rsa
% All keys will be removed.
% All router certs issued using these keys will also be removed.
Do you really want to remove these keys? [yes/no]: yes
R1(config)#

```

Modifying the SSH Configuration

```
R1# show ip ssh
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Authentication timeout: 120 secs; Authentication retries: 3
<output omitted>

R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip ssh time-out 60
R1(config)# ip ssh authentication-retries 2
R1(config)# ^Z
R1#
*Feb 16 21:23:51.237: %SYS-5-CONFIG_I: Configured from console by console
R1# show ip ssh
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Authentication timeout: 60 secs; Authentication retries: 2
<output omitted>
```

Connecting to an SSH-Enabled Router

Two ways to connect:

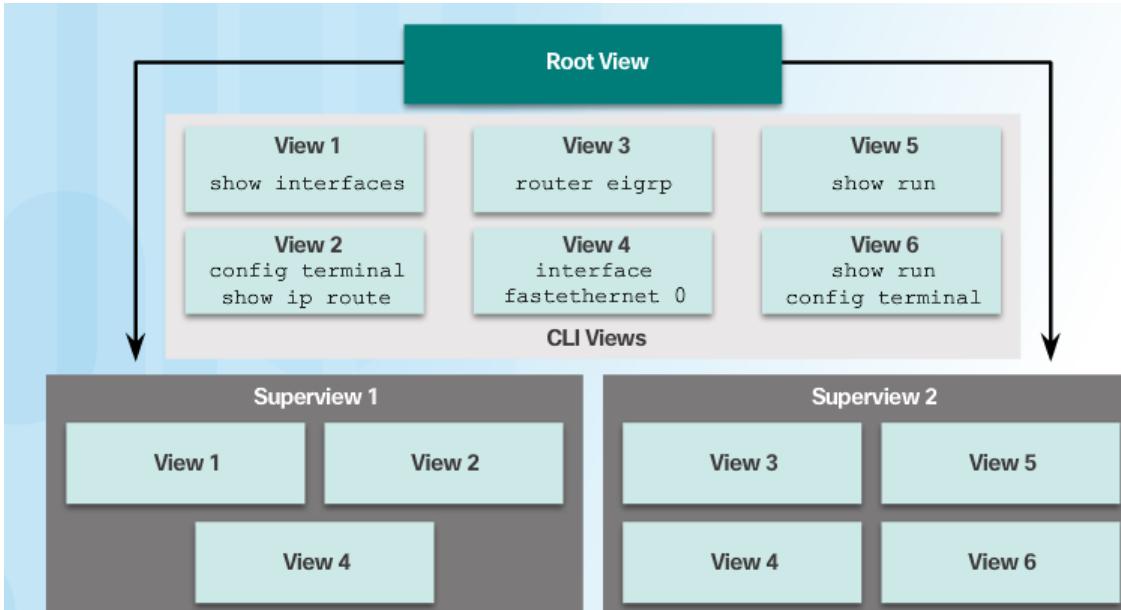
Enable SSH and use a Cisco router as an SSH server or SSH client.

As a server, the router can accept SSH client connections

As a client, the router can connect via SSH to another SSH-enabled router

Use an SSH client running on a host, such as PuTTY, OpenSSH, or TeraTerm.

Role-Based Views



Superviews contain Views but not commands. Two Superviews can use the same View.
 For example, both Superview 1 and Superview 2 can have CLI View 4 placed inside.

Configuring Role-Based Views

Step 1

```
Router#
```

```
enable [view [view-name]]
```

Step 2

```
Router(config)#
```

```
parser view view-name
```

Step 3

```
Router(config-view)#
```

```
secret encrypted-password
```

Step 4

```
Router(config-view)#
```

```
commands parser mode {include | include-exclusive | exclude} [all]
[interface interface-name | command]
```

Configuring Role-Based CLI Superviews

Step 1

```
Router(config)#
  parser view view-name superview
```

Step 2

```
Router(config-view)#
  secret encrypted-password
```

Step 3

```
Router(config-view)#
  view view-name
```

Verify Role-Based CLI Views

Enable Root View and Verify All Views

```
R1# show parser view
Current view is 'JR-ADMIN'

R1# enable view
Password:

R1# show parser view
Current view is 'root'

R1# show parser view all
Views/SuperViews Present in System:
SHOWVIEW
VERIFYVIEW
REBOOTVIEW
USER *
SUPPORT *

JR-ADMIN *

-----(*) represent superview-----
R1#
```

Recovering a Router Password

1. Connect to the console port.
2. Record the configuration register setting.
3. Power cycle the router.
4. Issue the break sequence.
5. Change the default configuration register with the confreg 0x2142 command.
6. Reboot the router.
7. Press Ctrl-C to skip the initial setup procedure.
8. Put the router into privileged EXEC mode.
9. Copy the startup configuration to the running configuration.
10. Verify the configuration.
11. Change the enable secret password.
12. Enable all interfaces.
13. Change the config-register with the config-register configuration_register_setting.
14. Save the configuration changes.

Password Recovery

```
R1(config)# no service password-recovery
WARNING:
Executing this command will disable password recovery
mechanism.
Do not execute this command without another plan for
password recovery.
Are you sure you want to continue? [yes/no]: yes
R1(config)#

```

Disable Password Recovery

No Service Password Recovery

```
R1# show running-config
Building configuration...

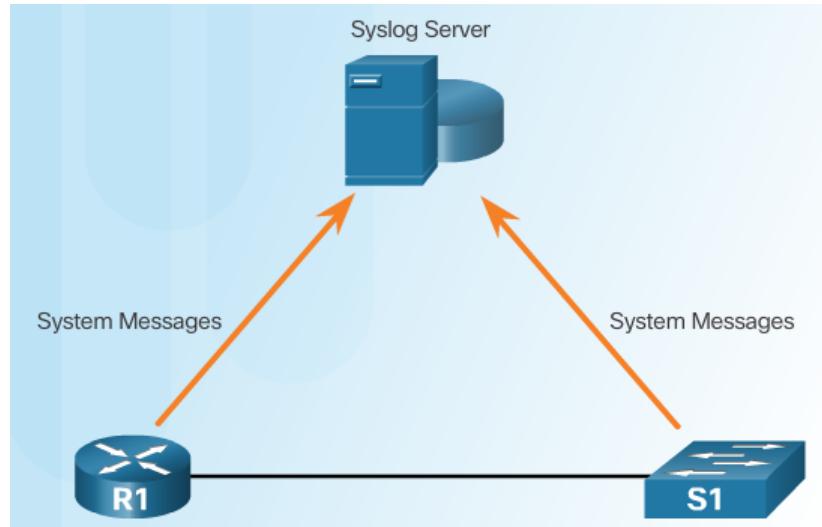
Current configuration : 836 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service password-recovery
```

```
System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2006 by cisco Systems, Inc.
PLD version 0x10
GIO ASIC version 0x127
c1841 platform with 131072 Kbytes of main memory
Main memory is configured to 64 bit mode with parity disabled

PASSWORD RECOVERY FUNCTIONALITY IS DISABLED
program load complete, entry point: 0x8000f000, size:0xcb80
```

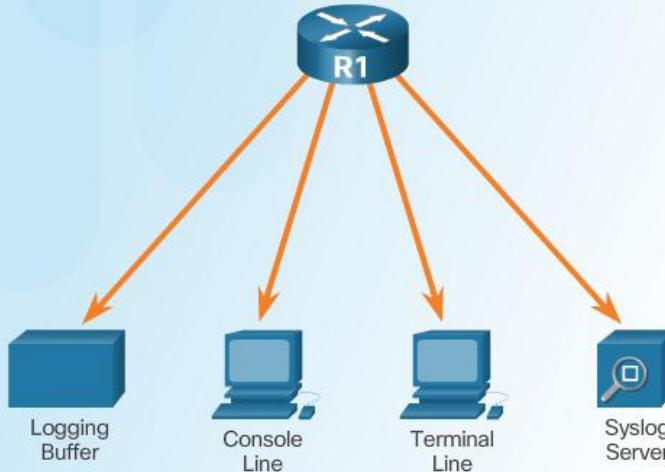
Password Recovery Functionality is Disabled

Introduction to Syslog



Syslog Operation

```
R1(config-if)# no shutdown
R1(config-if)#
000047: *Feb 19 11:36:47.779: %LINK-3-UPDOWN: Interface Serial0/0/0, changed
state to up
000048: *Feb 19 11:36:48.779: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/0, changed state to up
```



Syslog Message

Security Levels

Level	Keyword	Description	Definition
Highest Level	0	emergencies	System is unusable
	1	alerts	Immediate action is needed
	2	critical	Critical conditions exist
	3	errors	Error conditions exist
	4	warnings	Warning conditions exist
	5	notifications	Normal but significant condition
	6	informational	Informational messages only
Lowest Level	7	debugging	Debugging messages

Example Severity Levels

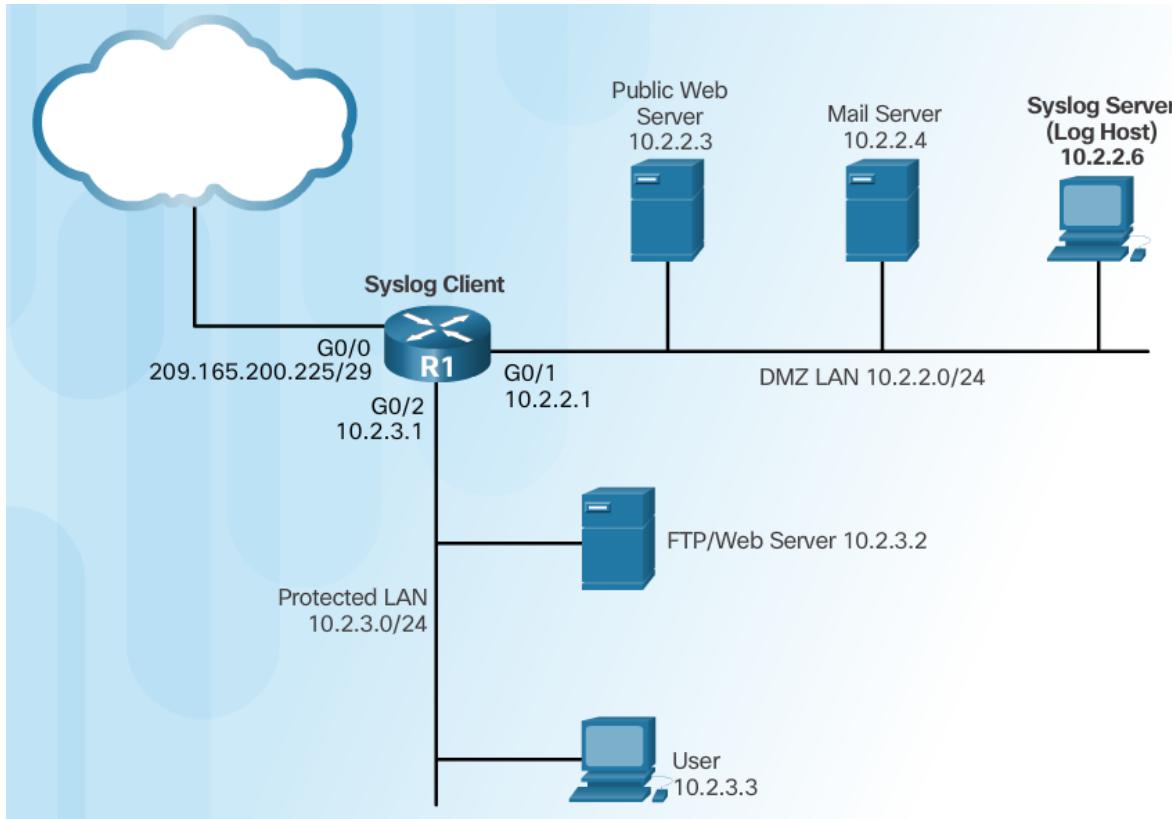
Syslog Level and Name	Definition	Example
0 LOG_EMERG	A panic condition normally broadcast to all users	Cisco IOS software could not load
1 LOG_ALERT	A condition that should be corrected immediately, such as a corrupted system database	Temperature too high
2 LOG_CRIT	Critical conditions; for example, device errors	Unable to allocate memory
3 LOG_ERR	Errors	Invalid memory size
4 LOG_WARNING	Warning messages	Crypto operation failed
5 LOG_NOTICE	Non-error conditions that may require special handling	Interface changed state, up or down
6 LOG_INFO	Informational messages	Packet denied by ACL
7 LOG_DEBUG	Messages that contain information that is normally used only when debugging a program	Packet type invalid

Syslog Message (Cont.)

1 2 3 4 5
000048: *Feb 19 11:36:48.779: %LINEPROTO-5-UPDOWN:
Line protocol on Interface Serial0/0/0, changed state to up
6

Column 1	Column 2
1	seq no Stamps log messages with a sequence number if service sequence-numbers is configured.
2	timestamp displays if service timestamps log is configured
3	facility denotes the source or the cause of the system message
4	severity levels 0 - 7
5	MNEMONIC text string that uniquely describes the message
6	description text string containing detailed information about the event being reported

Syslog Systems



Configuring System Logging

Step 1

```
Router(config) #
```

```
logging host [hostname | ip-address]
```

Step 2 (optional)

```
Router(config) #
```

```
logging trap level
```

Step 3

```
Router(config) #
```

```
logging source-interface interface-type interface-number
```

Step 4

```
Router(config) #
```

```
logging on
```

Cisco AutoSecure

```
R1# auto secure
--- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security
of the router but it will not make router
absolutely secure from all security attacks ***

All the configuration done as part of AutoSecure
will be shown here. For more details of why and
how this configuration is useful, and any possible
side effects, please refer to Cisco documentation of
AutoSecure.

At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for
AutoSecure

Is this router connected to internet? [no]:yes
```

Using the Cisco AutoSecure Feature

Router#

```
auto secure [no-interact | full] [forwarding | management]
[ntp | login | ssh | firewall | tcp-intercept]
```

Parameter	Description
no-interact	(Optional) The user will not be prompted for any interactive configurations. No interactive dialogue parameters will be configured, including usernames or passwords.
full	(Optional) The user will be prompted for all interactive questions. This is the default setting.
forwarding	(Optional) Only the forwarding plane will be secured.
management	(Optional) Only the management plane will be secured.
ntp	(Optional) Specifies the configuration of the NTP feature in the AutoSecure CLI.
login	(Optional) Specifies the configuration of the Login feature in the AutoSecure CLI.
ssh	(Optional) Specifies the configuration of the SSH feature in the AutoSecure CLI.
firewall	(Optional) Specifies the configuration of the Firewall feature in the AutoSecure CLI.
tcp-intercept	(Optional) Specifies the configuration of the TCP-Intercept feature in the AutoSecure CLI.

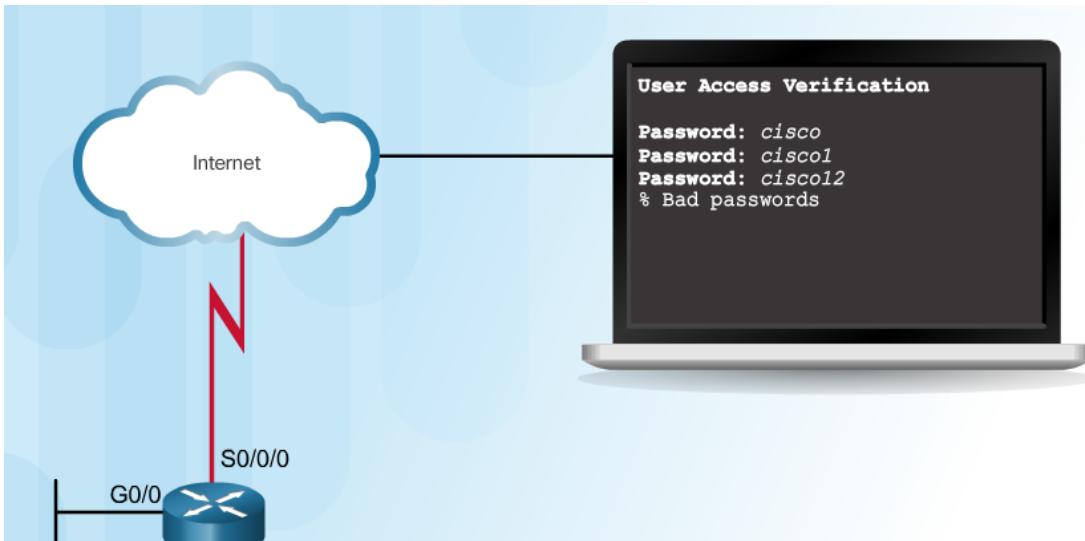
Using the auto secure Command

1. The auto secure command is entered
2. Wizard gathers information about the outside interfaces
3. AutoSecure secures the management plane by disabling unnecessary services
4. AutoSecure prompts for a banner
5. AutoSecure prompts for passwords and enables password and login features
6. Interfaces are secured
7. Forwarding plane is secured

Authentication, Authorization, and Accounting

Authentication without AAA

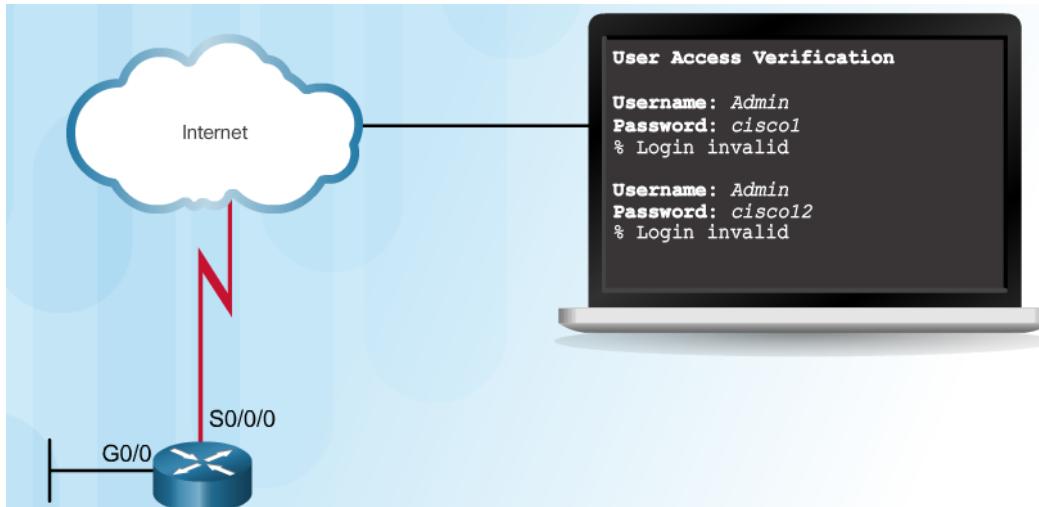
Telnet is Vulnerable to Brute-Force Attacks



```
R1(config)# line vty 0 4
R1(config-line)# password cis5cio
R1(config-line)# login
```

Authentication without AAA (Cont.)

SSH and Local Database Method



```
R1(config)# ip domain-name cisco-academy.com
R1(config)# crypto key generate rsa general-keys modulus 2048
R1(config)# username Admin algorithm-type scrypt secret Str0ng5rPa55w0rd
R1(config)# line vty 0 4
R1(config-line)# transport input ssh
R1(config-line)# login local
```

AAA Components

The diagram illustrates the three components of AAA (Authentication, Authorization, Accounting) using a credit card statement as an example.

Authentication: Who are you? (Associated with the top left box)

Authorization: How much can you spend? (Associated with the middle left box)

Accounting: What did you spend it on? (Associated with the bottom left box)

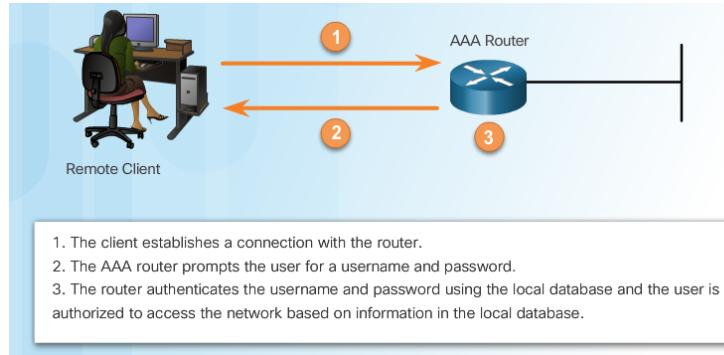
Credit Card Statement Details:

Account Number	Statement Closing Date	Current Amount Due		
1234-567-890	01-31-01	\$278.50		
MAIL PAYMENT TO: THE BANK 132 VINE STREET ANYTOWN, USA 67500-0010				
872919345 00178255000000003				
Detach here and return upper portion with check or money order. Do not staple or fold. Retain this portion for your files.				
Statement of Personal Credit Card Account				
Cardmember Name	Account Number	Statement Closing Date		
JOE EMPLOYEE	1234-456-890	01-31-01		
Statement Date:	02-01-01	Payment Due Date:	03-01-01	
Closing Date:	01-31-01	Credit Available:	\$1221.50	
Credit Limit	\$1,500.00	New Balance:	\$278.50	
Minimum Payment Due: \$20.00				
Account Summary				
Previous Balance:	+74.24	Transaction Fees:	+3.00	
Purchases:	+250.50	Annual Fees:	+25.00	
Cash Advances:	+0	Current Amount Due:	+250.50	
Payments:	-74.25	Amount Past Due:	+0	
Finance Charge:	+0	Amount Over Credit Line:	+0	
Late Charge:	+0	NEW BALANCE:	\$278.50	
Activity Since Last Statement				
Reference Number	Sold	Posted	Activity Since Last Statement	Amount
43210987	01-03	01-13	Payment, Thank You	-\$74.25
01234567	01-12	01-13	Wings 'N' Things Anytown, USA	\$25.25
78901234	01-14	01-17	Record Release Anytown, USA	\$40.00
45678901	01-14	01-17	Sports Stadium Anytown, USA	\$75.25
3210987	01-22	01-23	Tie Tack Anytown, USA	\$20.75
76543210	01-29	01-30	Electronic World Anytown, USA	\$89.25
23455678		01-30	Transaction Fees	\$3.00
34567890		01-01	Annual Fee	\$25.00

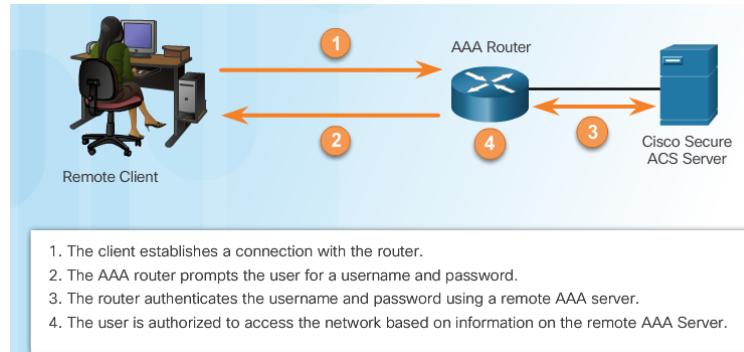
PAGE 1 OF 1

Authentication Modes

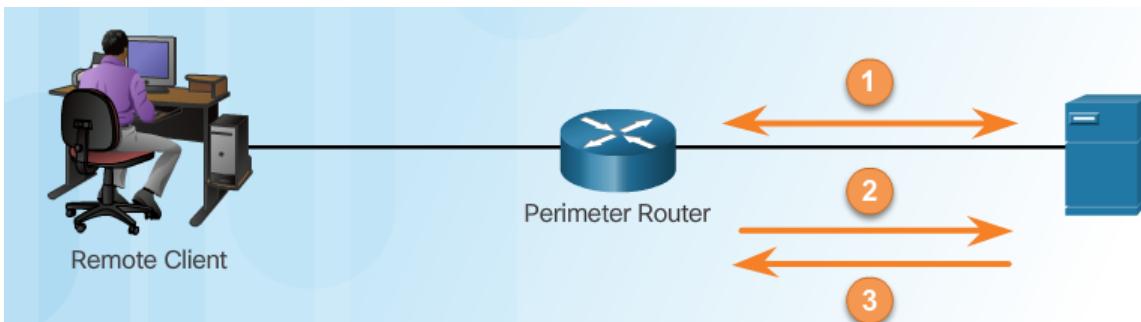
Local AAA Authentication



Server-Based AAA Authentication



AAA Authorization



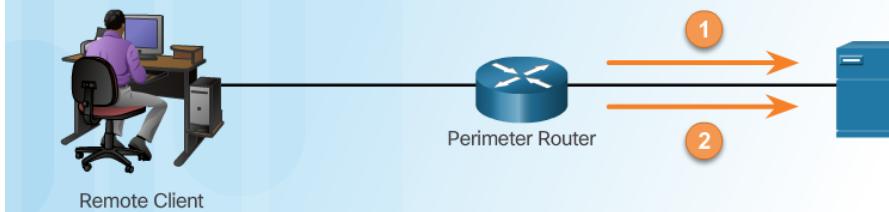
1. When a user has been authenticated, a session is established with the AAA server.
2. The router requests authorization for the requested service from the AAA server.
3. The AAA server returns a PASS/FAIL for authorization.

Accounting

Types of accounting information:

Network
Connection
EXEC
System
Command
Resource

AAA Accounting



1. When a user has been authenticated, the AAA accounting process generates a start message to begin the accounting process.
2. When the user finishes, a stop message is recorded and the accounting process ends.

Authenticating Administrative Access

1. Add usernames and passwords to the local router database for users that need administrative access to the router.
2. Enable AAA globally on the router.
3. Configure AAA parameters on the router.
4. Confirm and troubleshoot the AAA configuration.

```
R1(config)# username JR-ADMIN algorithm-type scrypt secret Str0ng5rPa55w0rd
R1(config)# username ADMIN algorithm-type scrypt secret Str0ng5rPa55w0rd
R1(config)# aaa new-model
R1(config)# aaa authentication login default local-case
R1(config)#

```

Authentication Methods

Method Type Keywords Description

enable	Uses the enable password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group group-name	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the aaa group server radius OR aaa group server tacacs+ command.

```
router(config-line)#
aaa authentication login {default | list-name} method1...[method4]
```

Command	Description
default	Uses the listed authentication methods that follow this keyword as the default list of methods when a user logs in.
list-name	Character string used to name the list of authentication methods activated when a user logs in.
method1... [method4]	Identifies the list of methods that the AAA authentication process will query in the given sequence. At least one method must be specified. A maximum of four methods may be specified.

Default and Named Methods

Example Local AAA Authentication

```
R1(config)# username JR-ADMIN algorithm-type scrypt secret Str0ng5rPa55w0rd
R1(config)# username ADMIN algorithm-type scrypt secret Str0ng5rPa55w0rd
R1(config)# aaa new-model
R1(config)# aaa authentication login default local-case enable
R1(config)# aaa authentication login SSH-LOGIN local-case
R1(config)# line vty 0 4
R1(config-line)# login authentication SSH-LOGIN
```

Fine-Tuning the Authentication Configuration

Command Syntax

```
Router(config)#  
aaa local authentication attempts max-fail [number-of-unsuccessful-attempts]
```

Command	Description
number-of-unsuccessful-attempts	Number of unsuccessful authentication attempts before a connection is dropped and the user account is locked.

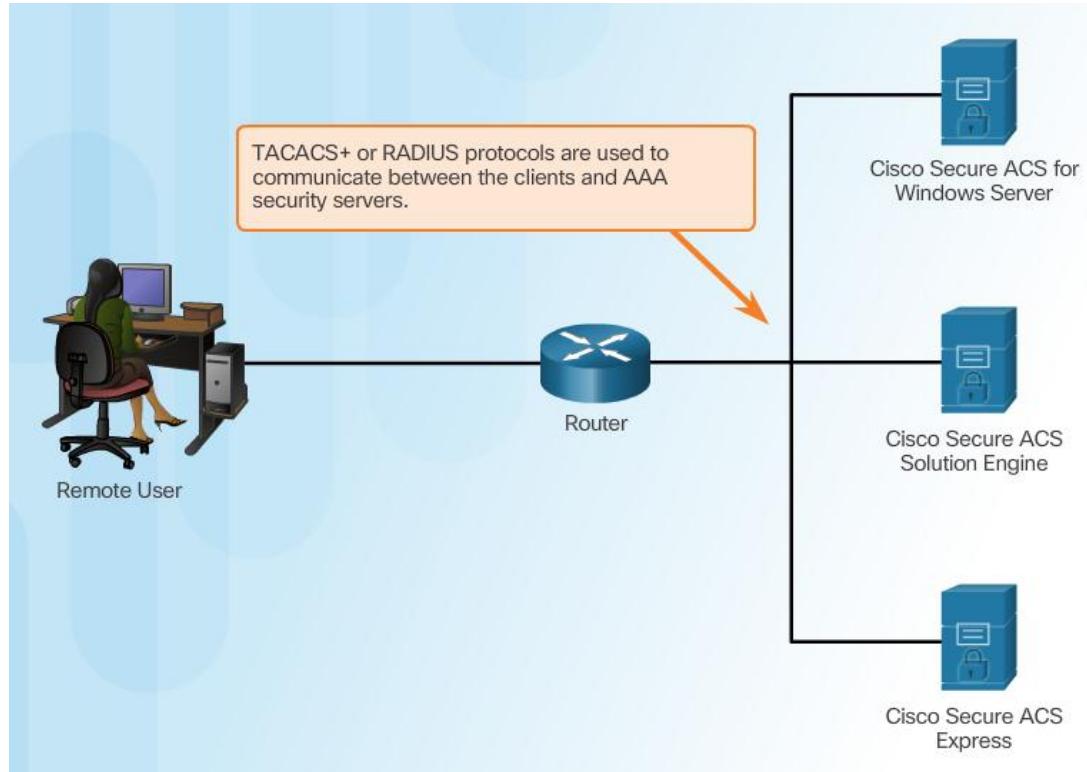
Display Locked Out Users

```
R1# show aaa local user lockout  
          Local-user      Lock time  
          JR-ADMIN       04:28:49 UTC Sat Dec 27 2015
```

Show Unique ID of a Session

```
R1# show aaa sessions  
Total sessions since last reload: 4  
Session Id: 1  
          Unique Id: 175  
          User Name: ADMIN  
          IP Address: 192.168.1.10  
          Idle Time: 0  
          CT Call Handle: 0
```

Introducing Cisco Secure Access Control System

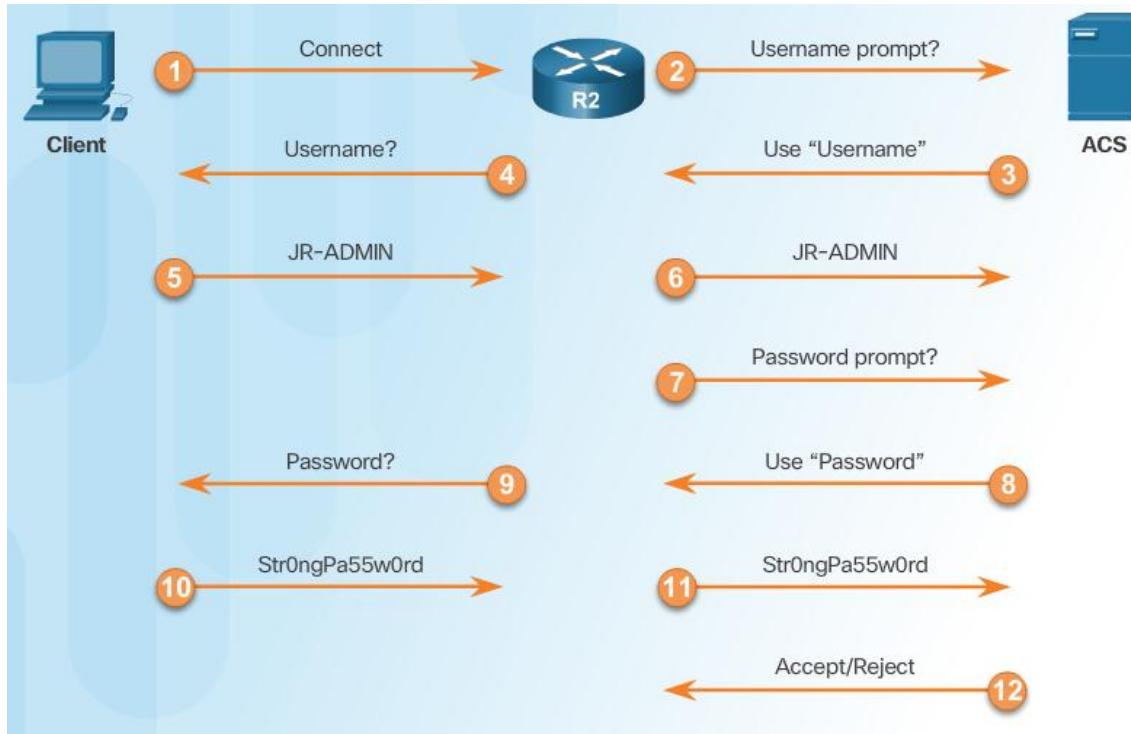


TACACS+ RADIUS

	TACACS+	RADIUS
Functionality	Separates AAA according to the AAA architecture, allowing modularity of the security server implementation	Combines authentication and authorization but separates accounting, allowing less flexibility in implementation than TACACS+
Standard	Mostly Cisco supported	Open/RFC standard
Transport Protocol	TCP	UDP
CHAP	Bidirectional challenge and response as used in Challenge Handshake Authentication Protocol (CHAP)	Unidirectional challenge and response from the RADIUS security server to the RADIUS client
Protocol Support	Multiprotocol support	No ARA, no NetBEUI
Confidentiality	Entire packet encrypted	Password encrypted
Customization	Provides authorization of router commands on a per-user or per-group basis	Has no option to authorize router commands on a per-user or per-group basis
Accounting	Limited	Extensive

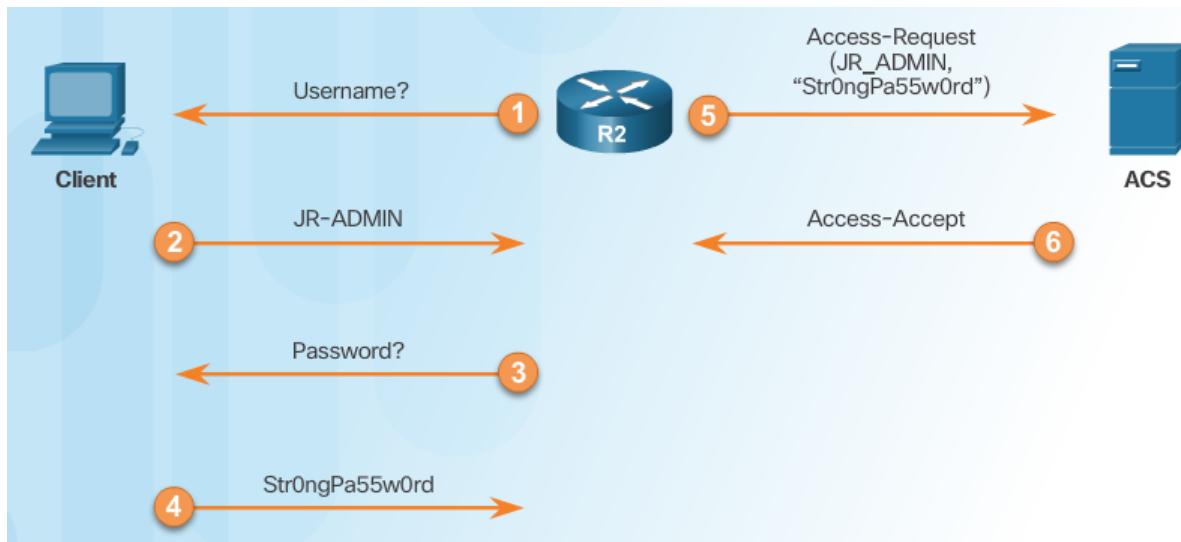
TACACS+ Authentication

TACACS+ Authentication Process



RADIUS Authentication

RADIUS Authentication Process

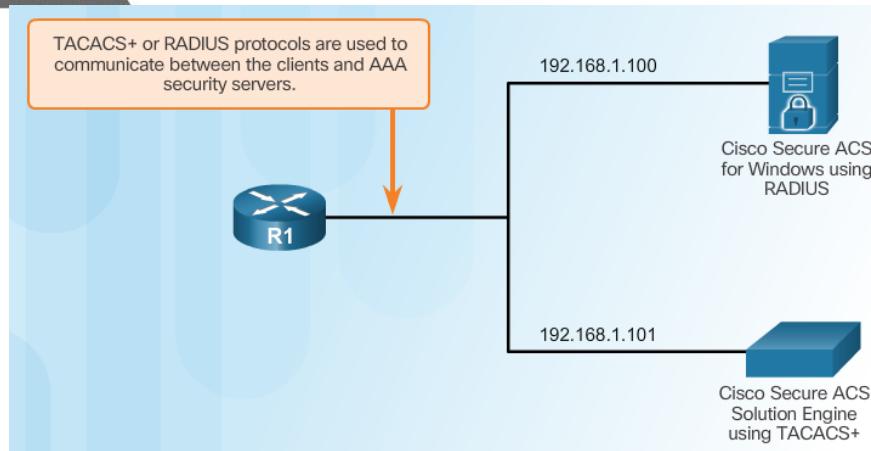


Steps for Configuring Server-Based AAA Authentication with CLI

1. Enable AAA.
2. Specify the IP address of the ACS server.
3. Configure the secret key.
4. Configure authentication to use either the RADIUS or TACACS+ server.

Configuring the CLI with TACACS+ Servers

Server-Based AAA Reference Topology



Configure a AAA TACACS+ Server

```
R1(config)# aaa new-model
R1(config)#
R1(config)# tacacs server Server-T
R1(config-server-tacacs)# address ipv4 192.168.1.101
R1(config-server-tacacs)# single-connection
R1(config-server-tacacs)# key TACACS-Pa55w0rd
R1(config-server-tacacs)# exit
R1(config)#
```

Configuring the CLI for RADIUS Servers

Configure a AAA RADIUS Server

```
R1(config)# aaa new-model
R1(config)#
R1(config)# radius server SERVER-R
R1(config-radius-server)# address ipv4 192.168.1.100 auth-port 1812 acct-port 1813
R1(config-radius-server)# key RADIUS-Pa55w0rd
R1(config-radius-server)# exit
R1(config)#
```

Configure Authentication to Use the AAA Server

Command Syntax

```
R1(config)# aaa authentication login default ?
  cache          Use Cached-group
  enable         Use enable password for authentication.
  group          Use Server-group
  krb5           Use Kerberos 5 authentication.
  krb5-telnet    Allow logins only if already authenticated via Kerberos V
                  Telnet.
  line           Use line password for authentication.
  local          Use local username authentication.
  local-case     Use case-sensitive local username authentication.
  none           NO authentication.
  passwd-expiry  enable the login list to provide password aging support

R1(config)# aaa authentication login default group ?
  WORD           Server-group name
  ldap           Use list of all LDAP hosts.
  radius         Use list of all Radius hosts.
  tacacs+        Use list of all Tacacs+ hosts.
```

Configure Server-Based AAA Authentication

```
R1(config)# aaa new-model
R1(config)#
R1(config)# tacacs server Server-T
R1(config-server-tacacs)# address ipv4 192.168.1.100
R1(config-server-tacacs)# single-connection
R1(config-server-tacacs)# key TACACS-Pa55w0rd
R1(config-server-tacacs)# exit
R1(config)#
R1(config)# radius server SERVER-R
R1(config-radius-server)# address ipv4 192.168.1.101 auth-port 1812 acct-port 1813
R1(config-radius-server)# key RADIUS-Pa55w0rd
R1(config-radius-server)# exit
R1(config)#
R1(config)# aaa authentication login default group tacacs+ group radius local-case
```

Introduction to Server-Based AAA Authorization

Authentication vs. Authorization

Authentication ensures a device or end-user is legitimate

Authorization allows or disallows authenticated users access to certain areas and programs on the network.

TACACS+ vs. RADIUS

TACACS+ separates authentication from authorization

RADIUS does **not** separate authentication from authorization

AAA Authorization Configuration with CLI

Command Syntax

```
R1(config)# aaa authorization {network | exec | commands level}  
{default | list-name} method1...[method4]
```

```
R1(config)# aaa authorization exec ?  
WORD      Named authorization list.  
default   The default authorization list.
```

Authorization Method Lists

```
R1(config)# aaa authorization {network | exec | commands level}  
{default | list-name} method1...[method4]
```

```
R1(config)# aaa authorization exec default ?  
cache      Use Cached-group  
group     Use server-group.  
if-authenticated  Succeed if user has authenticated.  
krb5-instance  Use Kerberos instance privilege maps.  
local       Use local database.  
none        No authorization (always succeeds).
```

```
R1(config)# aaa authorization exec default group ?  
WORD      Server-group name  
ldap     Use list of all LDAP hosts.  
radius    Use list of all Radius hosts.  
tacacs+   Use list of all Tacacs+ hosts.
```

Example AAA Authorization

```
R1(config)# username JR-ADMIN algorithm-type scrypt secret Str0ng5rPa55w0rd  
R1(config)# username ADMIN algorithm-type scrypt secret Str0ng5rPa55w0rd  
R1(config)# aaa new-model  
R1(config)# aaa authorization exec default group tacacs+  
R1(config)# aaa authorization network default group tacacs+
```

AAA Accounting Configuration with CLI

Command Syntax

```
R1(config)#
aaa accounting {network | exec | connection} {default | list-name}
{start-stop | stop-only | none } [broadcast] method1...[method4]
```

```
R1(config)#
aaa accounting exec?
WORD      Named Accounting list.
default   The default accounting list.
```

Accounting Method Lists

```
R1(config)#
aaa accounting {network | exec | connection} {default | list-name}
{start-stop | stop-only | none } [broadcast] method1...[method4]
```

```
R1(config)#
aaa accounting exec default start-stop?
broadcast Use Broadcast for Accounting
group     Use Server-group

R1(config)#
aaa accounting exec default start-stop group?
WORD      Server-group name
radius    Use list of all Radius hosts.
tacacs+   Use list of all Tacacs+ hosts.
```

Example AAA Accounting

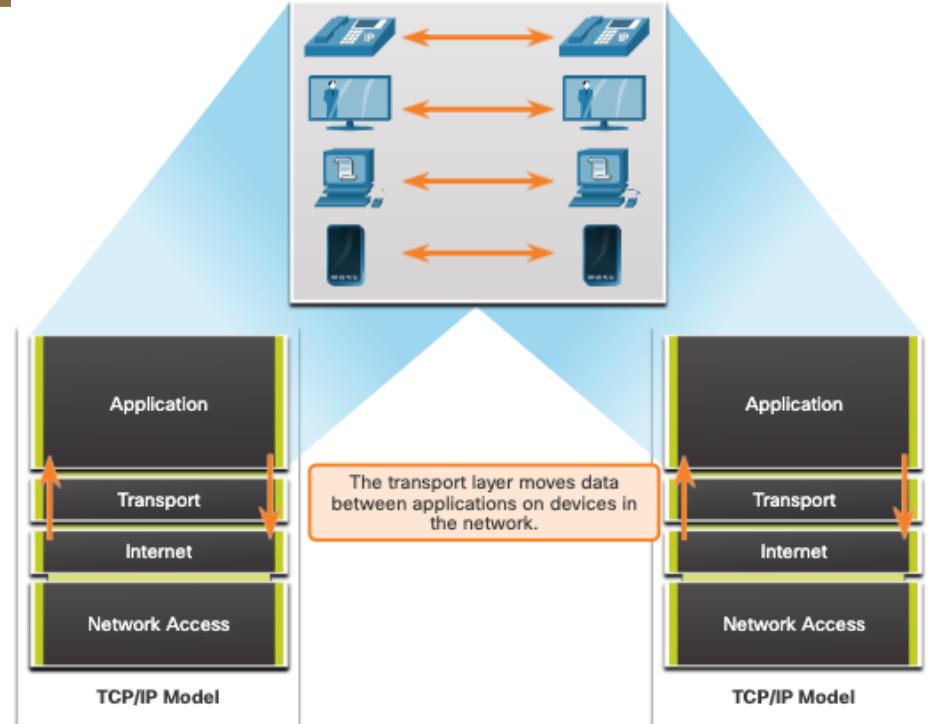
```
R1(config)#
username JR-ADMIN algorithm-type scrypt secret Str0ng5rPa5w0rd
R1(config)#
username ADMIN algorithm-type scrypt secret Str0ng5rPa55w0rd
R1(config)#
aaa new-model
R1(config)#
aaa authentication login default group tacacs+
R1(config)#
aaa authorization exec default group tacacs+
R1(config)#
aaa authorization network default group tacacs+
R1(config)#
aaa accounting exec default start-stop group tacacs+
R1(config)#
aaa accounting network default start-stop group tacacs+
```

Transport Layer

Role of the Transport Layer

The transport layer is:

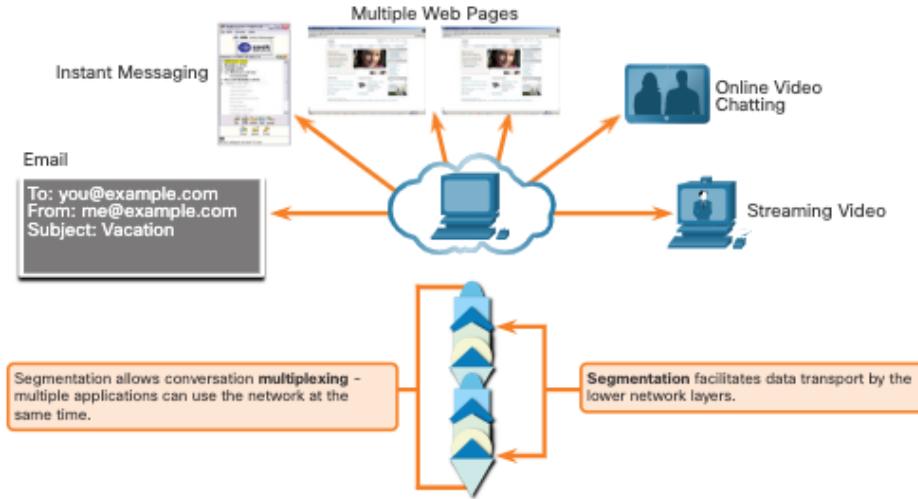
- responsible for logical communications between applications running on different hosts.
- The link between the application layer and the lower layers that are responsible for network transmission.



Transport Layer Responsibilities

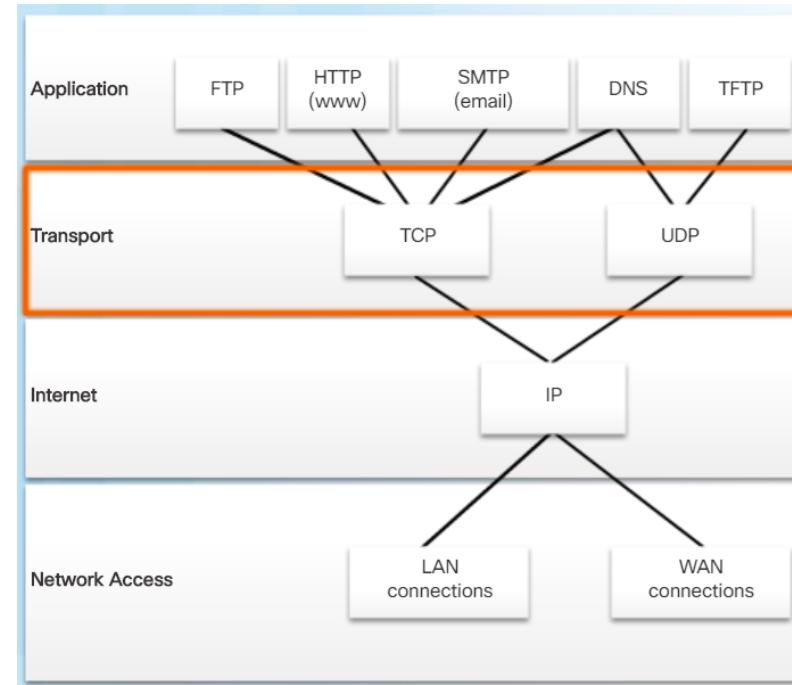
The transport layer has the following responsibilities:

- Tracking individual conversations
- Segmenting data and reassembling segments
- Adds header information
- Identify, separate, and manage multiple conversations
- Uses segmentation and multiplexing to enable different communication conversations to be interleaved on the same network



Transport Layer Protocols

- IP does not specify how the delivery or transportation of the packets takes place.
- Transport layer protocols specify how to transfer messages between hosts, and are responsible for managing reliability requirements of a conversation.
- The transport layer includes the TCP and UDP protocols.

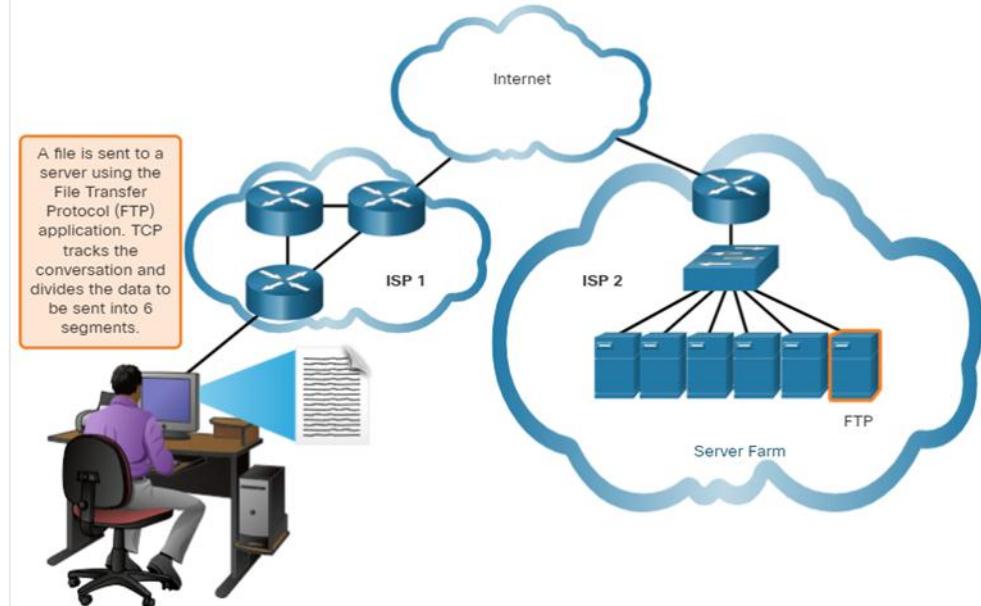


Transmission Control Protocol

TCP provides reliability and flow control.

TCP basic operations:

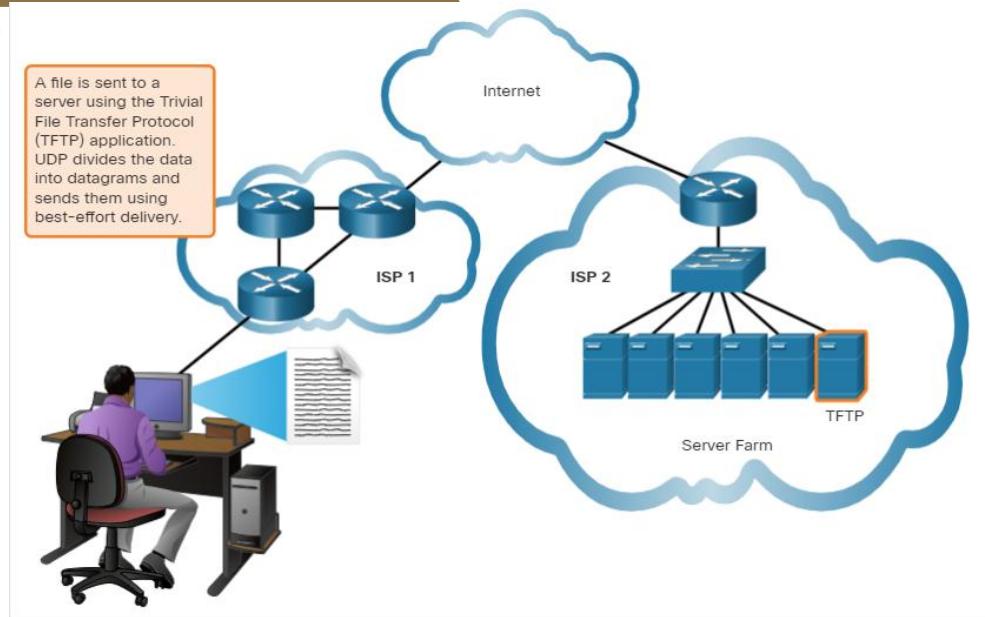
- Number and track data segments transmitted to a specific host from a specific application
- Acknowledge received data
- Retransmit any unacknowledged data after a certain amount of time
- Sequence data that might arrive in wrong order
- Send data at an efficient rate that is acceptable by the receiver



User Datagram Protocol (UDP)

UDP provides the basic functions for delivering datagrams between the appropriate applications, with very little overhead and data checking.

- UDP is a connectionless protocol.
- UDP is known as a best-effort delivery protocol because there is no acknowledgment that the data is received at the destination.



The Right Transport Layer Protocol for the Right Application

UDP is also used by request-and-reply applications where the data is minimal, and retransmission can be done quickly.

If it is important that all the data arrives and that it can be processed in its proper sequence, TCP is used as the transport protocol.

UDP



VoIP
(IP telephony)



DNS
(Domain Name Resolution)

TCP



SMTP/IMAP
(Email)



HTTP/HTTPS
(World Wide Web)

Required protocol properties:

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

Required protocol properties:

- Reliable
- Acknowledges data
- Resends lost data
- Delivers data in sequenced order

TCP Features



Establishes a Session - TCP is a connection-oriented protocol that negotiates and establishes a permanent connection (or session) between source and destination devices prior to forwarding any traffic.

Ensures Reliable Delivery - For many reasons, it is possible for a segment to become corrupted or lost completely, as it is transmitted over the network. TCP ensures that each segment that is sent by the source arrives at the destination.

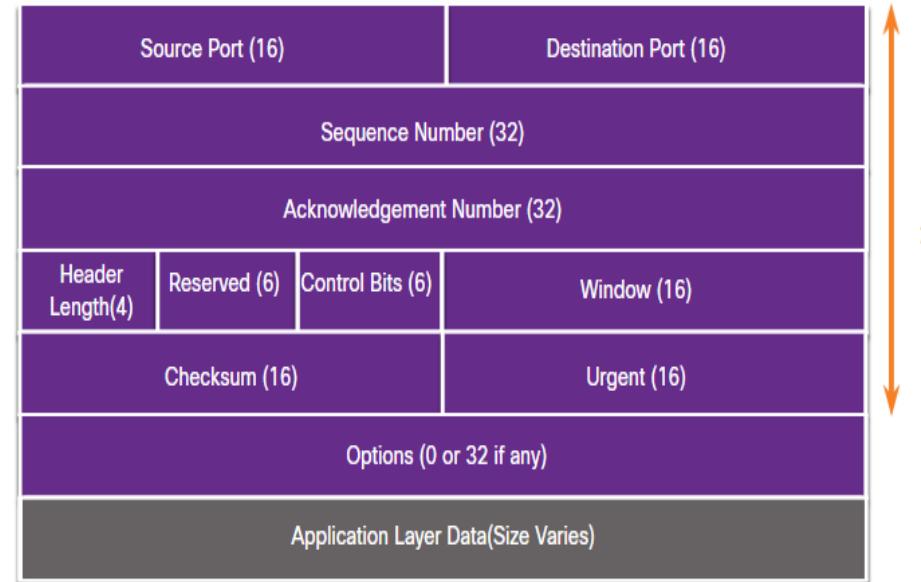
Provides Same-Order Delivery - Because networks may provide multiple routes that can have different transmission rates, data can arrive in the wrong order.

Supports Flow Control - Network hosts have limited resources (i.e., memory and processing power). When TCP is aware that these resources are overtaxed, it can request that the sending application reduce the rate of data flow.

TCP Header

TCP is a stateful protocol which means it keeps track of the state of the communication session.

TCP records which information it has sent, and which information has been acknowledged.



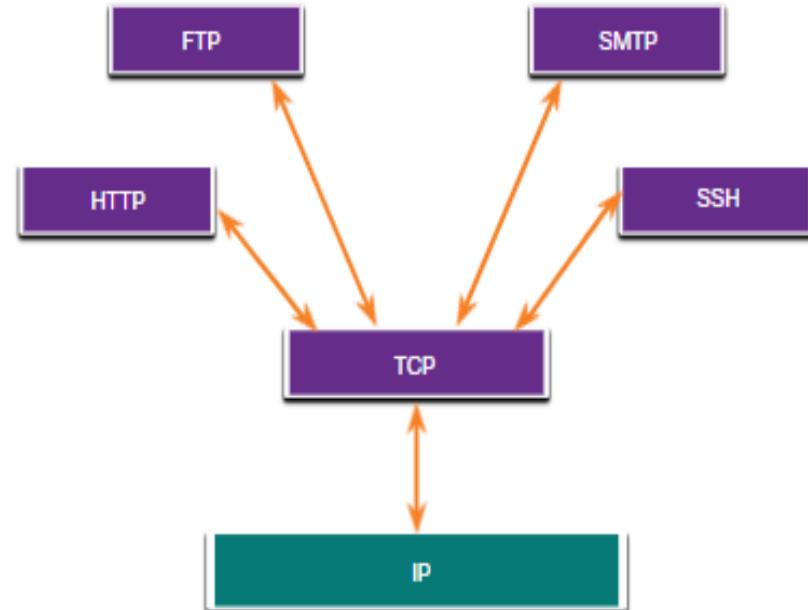
TCP Header Fields

TCP Header Field	Description
Source Port	A 16-bit field used to identify the source application by port number.
Destination Port	A 16-bit field used to identify the destination application by port number.
Sequence Number	A 32-bit field used for data reassembly purposes.
Acknowledgment Number	A 32-bit field used to indicate that data has been received and the next byte expected from the source.
Header Length	A 4-bit field known as "data offset" that indicates the length of the TCP segment header.
Reserved	A 6-bit field that is reserved for future use.
Control bits	A 6-bit field used that includes bit codes, or flags, which indicate the purpose and function of the TCP segment.
Window size	A 16-bit field used to indicate the number of bytes that can be accepted at one time.
Checksum	A 16-bit field used for error checking of the segment header and data.
Urgent	A 16-bit field used to indicate if the contained data is urgent.

Applications that use TCP



TCP handles all tasks associated with dividing the data stream into segments, providing reliability, controlling data flow, and reordering segments.



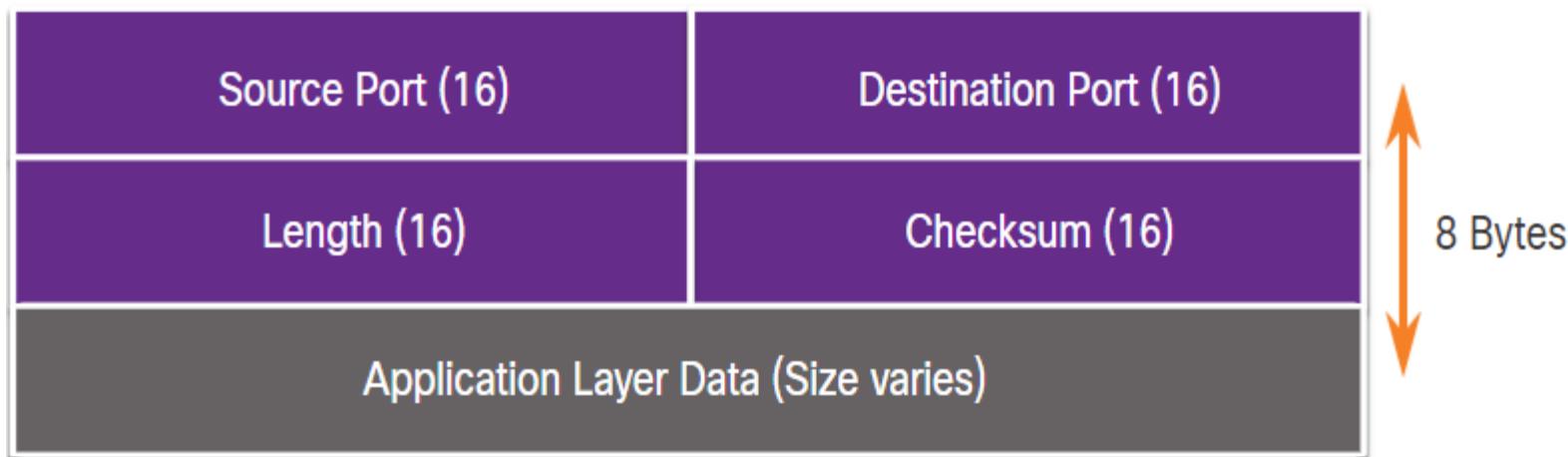
UDP Features

UDP features include the following:

- Data is reconstructed in the order that it is received.
- Any segments that are lost are not resent.
- There is no session establishment.
- The sending is not informed about resource availability.

UDP Header

The UDP header is far simpler than the TCP header because it only has four fields and requires 8 bytes (i.e. 64 bits).



UDP Header Fields

The table identifies and describes the four fields in a UDP header.

UDP Header Field	Description
Source Port	A 16-bit field used to identify the source application by port number.
Destination Port	A 16-bit field used to identify the destination application by port number.
Length	A 16-bit field that indicates the length of the UDP datagram header.
Checksum	A 16-bit field used for error checking of the datagram header and data.

Applications that use UDP

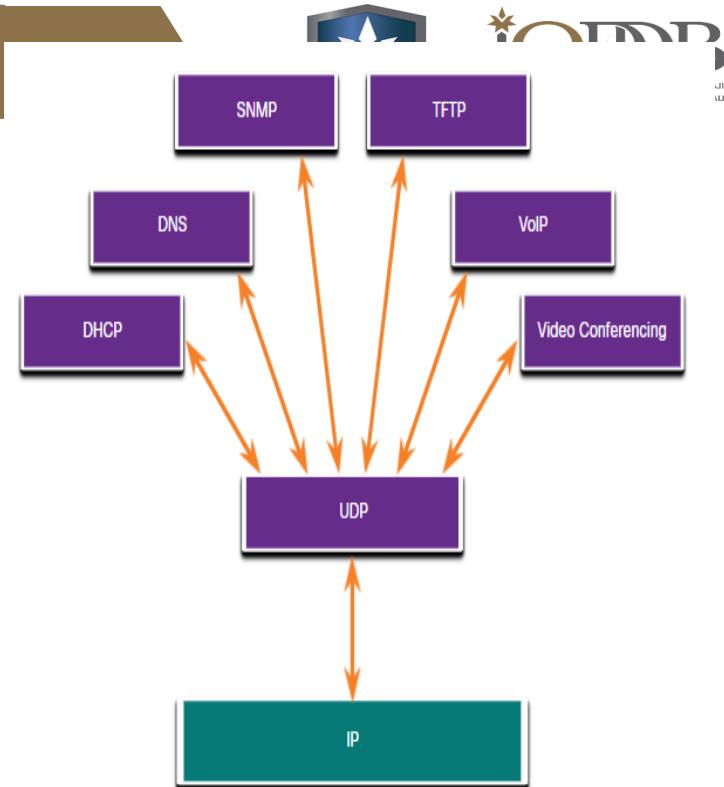
Live video and multimedia applications - These applications can tolerate some data loss but require little or no delay.

Examples include VoIP and live streaming video.

Simple request and reply applications - Applications with simple transactions where a host sends a request and may or may not receive a reply. Examples include DNS and DHCP.

Applications that handle reliability themselves -

Unidirectional communications where flow control, error detection, acknowledgments, and error recovery is not required, or can be handled by the application. Examples include SNMP and TFTP.



Multiple Separate Communications

TCP and UDP transport layer protocols use port numbers to manage multiple, simultaneous conversations.

The source port number is associated with the originating application on the local host whereas the destination port number is associated with the destination application on the remote host.

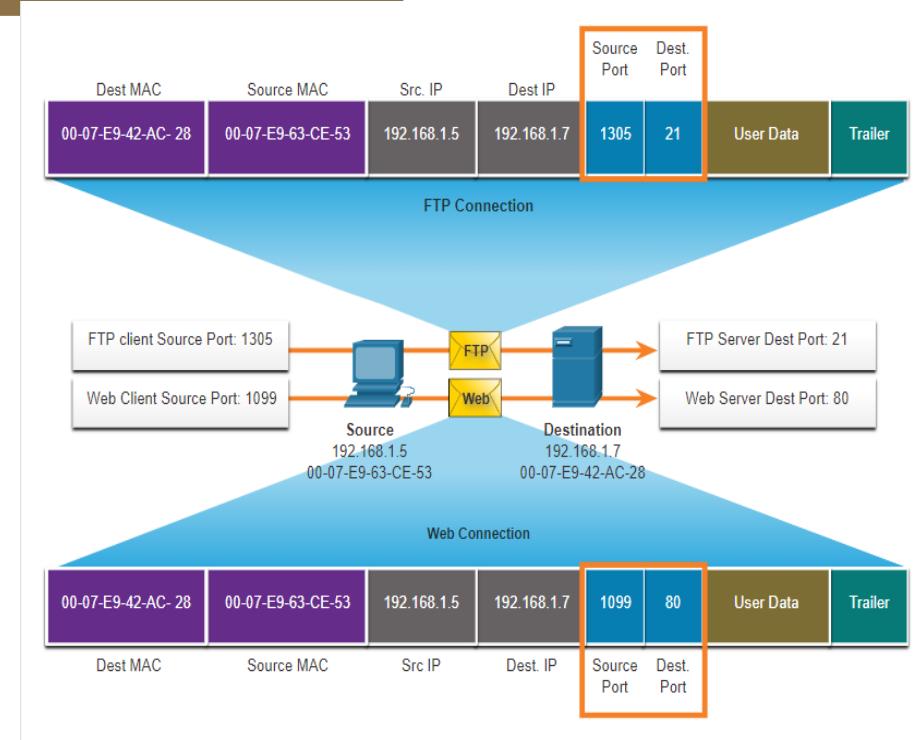
Source Port (16)

Destination Port (16)

Port numbers

Socket Pairs

- The source and destination ports are placed within the segment.
- The segments are then encapsulated within an IP packet.
- 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.
- Sockets enable multiple processes, running on a client, to distinguish themselves from each other, and multiple connections to a server process to be distinguished from each other.



Port Number Groups

Port Group	Number Range	Description
Well-known Ports	0 to 1,023	<ul style="list-style-type: none"> These port numbers are reserved for common or popular services and applications such as web browsers, email clients, and remote access clients. Defined well-known ports for common server applications enables clients to easily identify the associated service required.
Registered Ports	1,024 to 49,151	<ul style="list-style-type: none"> These port numbers are assigned by IANA to a requesting entity to use with specific processes or applications. These processes are primarily individual applications that a user has chosen to install, rather than common applications that would receive a well-known port number. For example, Cisco has registered port 1812 for its RADIUS server authentication process.
Private and/or Dynamic Ports	49,152 to 65,535	<ul style="list-style-type: none"> These ports are also known as <i>ephemeral ports</i>. The client's OS usually assign port numbers dynamically when a connection to a service is initiated. The dynamic port is then used to identify the client application during communication.

Port Number Groups (Cont.)

Well-Known Port Numbers

Port Number	Protocol	Application
20	TCP	File Transfer Protocol (FTP) - Data
21	TCP	File Transfer Protocol (FTP) - Control
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Simple Mail Transfer Protocol (SMTP)
53	UDP, TCP	Domain Name Service (DNS)
67	UDP	Dynamic Host Configuration Protocol (DHCP) - Server
68	UDP	Dynamic Host Configuration Protocol - Client
69	UDP	Trivial File Transfer Protocol (TFTP)
80	TCP	Hypertext Transfer Protocol (HTTP)
110	TCP	Post Office Protocol version 3 (POP3)
143	TCP	Internet Message Access Protocol (IMAP)
161	UDP	Simple Network Management Protocol (SNMP)
443	TCP	Hypertext Transfer Protocol Secure (HTTPS)

The netstat Command

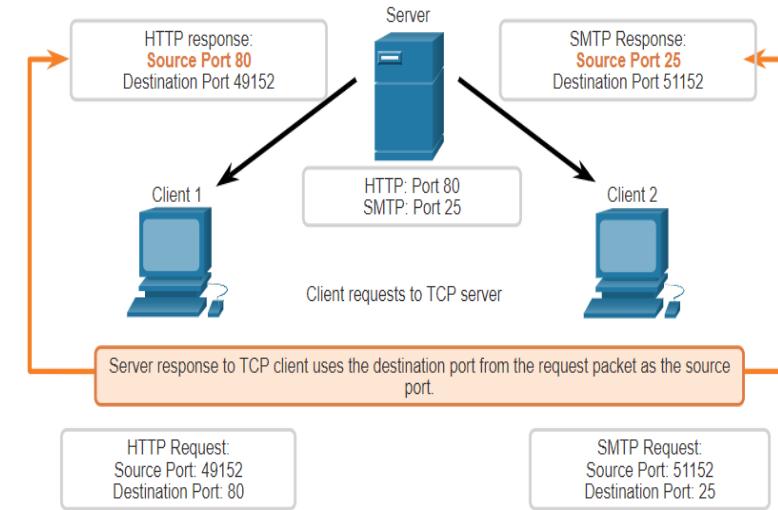
Unexplained TCP connections can pose a major security threat. Netstat is an important tool to verify connections.

```
C:\> netstat
Active Connections
Proto Local Address          Foreign Address
          State
TCP    192.168.1.124:3126      192.168.0.2:netbios-ssn
          ESTABLISHED
TCP    192.168.1.124:3158      207.138.126.152:http
          ESTABLISHED
TCP    192.168.1.124:3159      207.138.126.169:http
```

TCP Server Processes

Each application process running on a 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.
- An active server application assigned to a specific port is considered open, which means that the transport layer accepts, and processes segments addressed to that port.
- Any incoming client request addressed to the correct socket is accepted, and the data is passed to the server application.

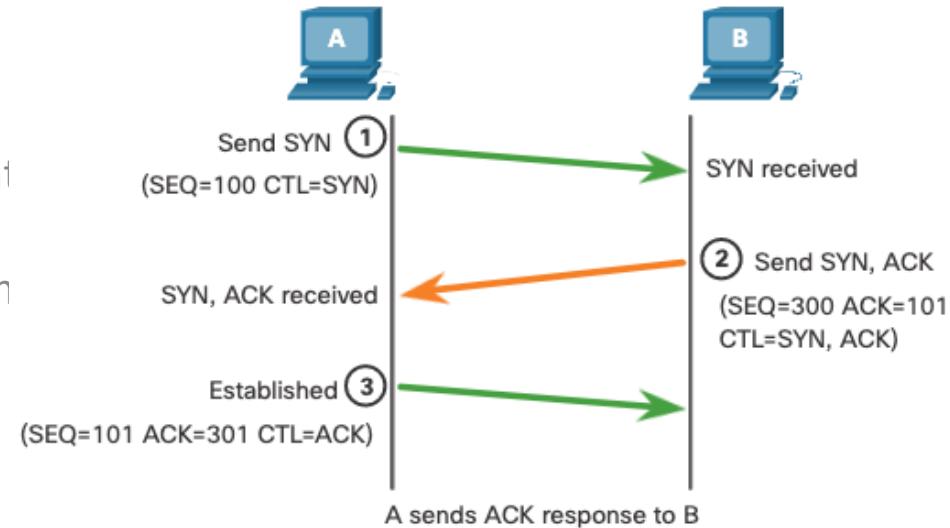


TCP Connection Establishment

Step 1: The initiating client requests a client-to-server communication session with the server.

Step 2: The server acknowledges the client-to-server communication session and requests a server-to-client communication session.

Step 3: The initiating client acknowledges the server-to-client communication session.



TCP Communication Process

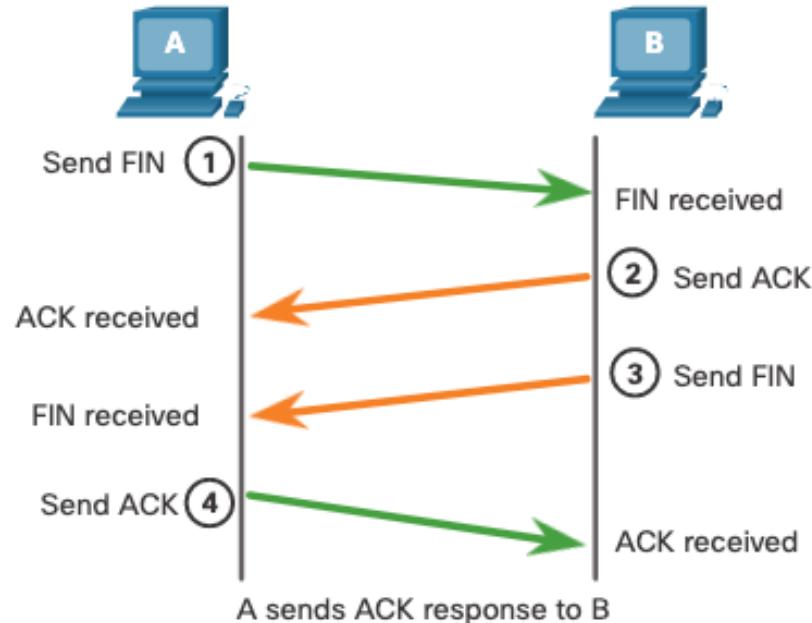
Session Termination

Step 1: When the client has no more data to send in the stream, it sends a segment with the FIN flag set.

Step 2: The server sends an ACK to acknowledge the receipt of the FIN to terminate the session from client to server.

Step 3: The server sends a FIN to the client to terminate the server-to-client session.

Step 4: The client responds with an ACK to acknowledge the FIN from the server.



TCP Three-Way Handshake Analysis



Functions of the Three-Way Handshake:

- It establishes that the destination device is present on the network.
- It verifies that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use.
- It informs the destination device that the source client intends to establish a communication session on that port number.

After the communication is completed the sessions are closed, and the connection is terminated. The connection and session mechanisms enable TCP reliability function.

TCP Three-Way Handshake Analysis



The six control bit flags are as follows:

URG - Urgent pointer field significant

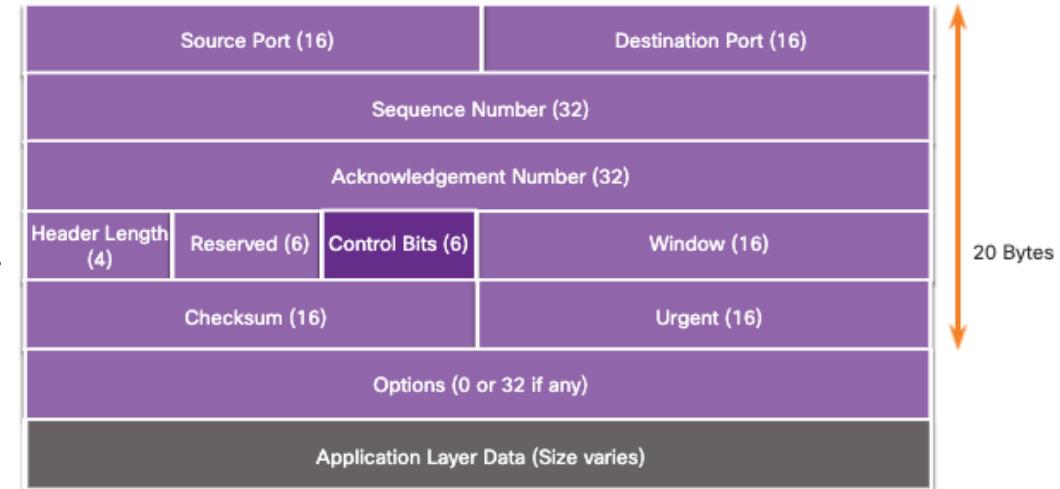
ACK - Acknowledgment flag used in connection establishment and session termination

PSH - Push function

RST - Reset the connection when an error or timeout occurs

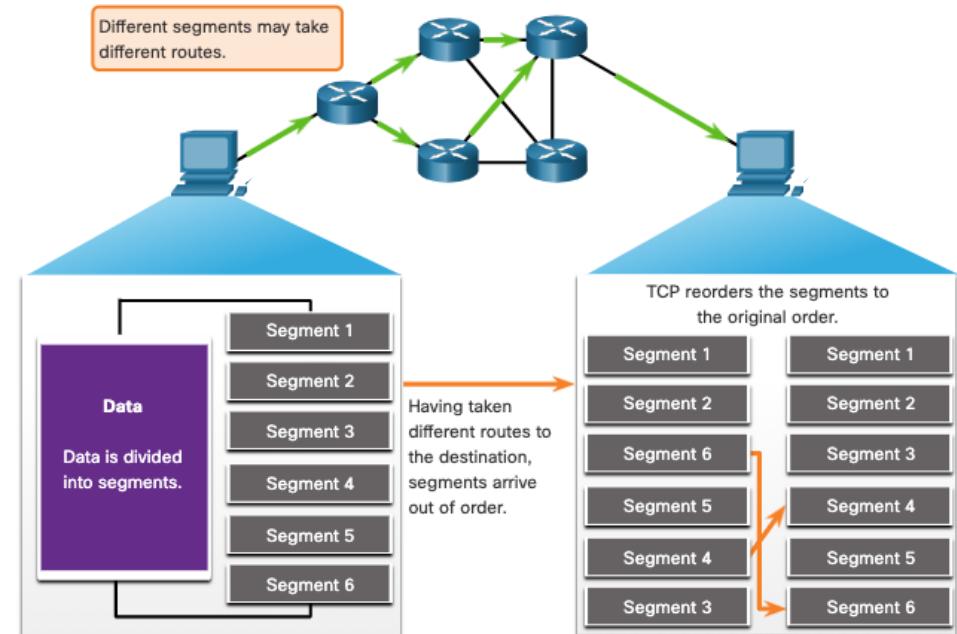
SYN - Synchronize sequence numbers used in connection establishment

FIN - No more data from sender and used in session termination



TCP Reliability- Guaranteed and Ordered Delivery

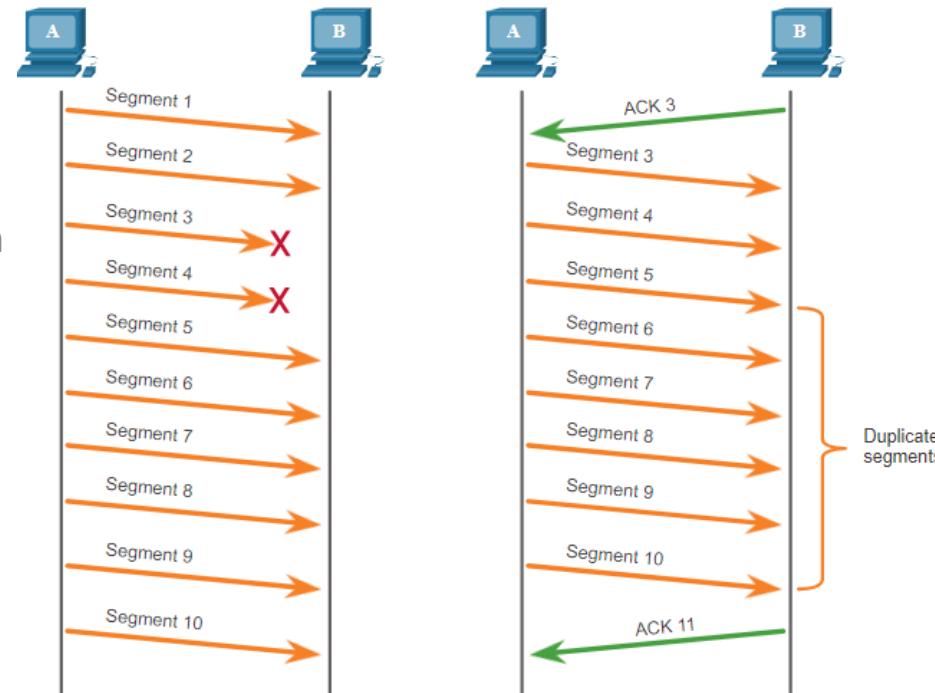
- TCP can also help maintain the flow of packets so that devices do not become overloaded.
- There may be times when TCP segments do not arrive at their destination or arrive out of order.
- All the data must be received and the data in these segments must be reassembled into the original order.
- Sequence numbers are assigned in the header of each packet to achieve this goal.



TCP Reliability – Data Loss and Retransmission

No matter how well designed a network is, data loss occasionally occurs.

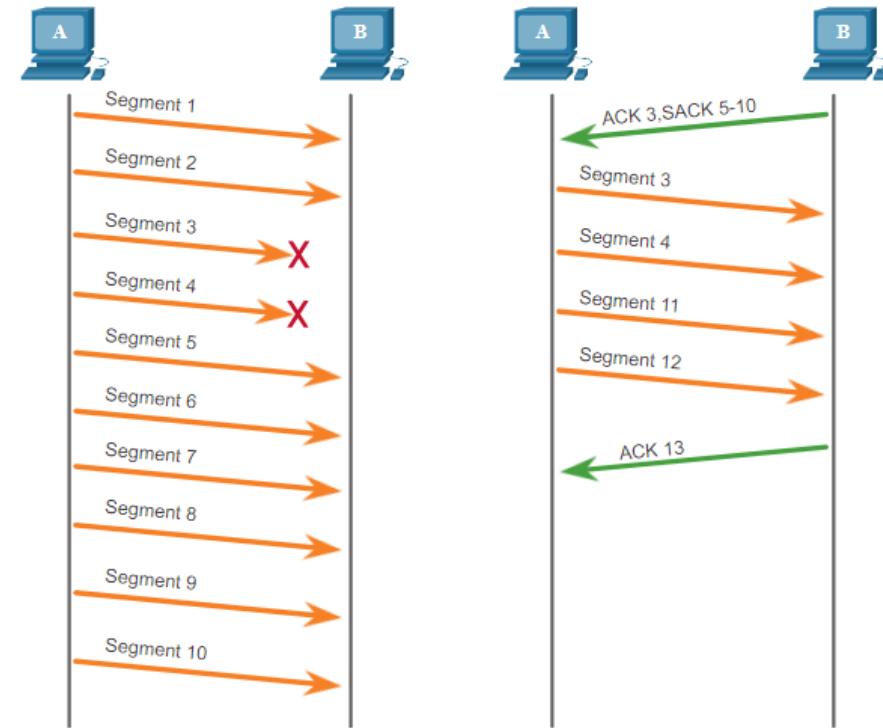
TCP provides methods of managing these segment losses. Among these is a mechanism to retransmit segments for unacknowledged data.



TCP Reliability – Data Loss and Retransmission (Cont.)

Host operating systems today typically employ an optional TCP feature called selective acknowledgment (SACK), negotiated during the three-way handshake.

If both hosts support SACK, the receiver can explicitly acknowledge which segments (bytes) were received including any discontinuous segments.



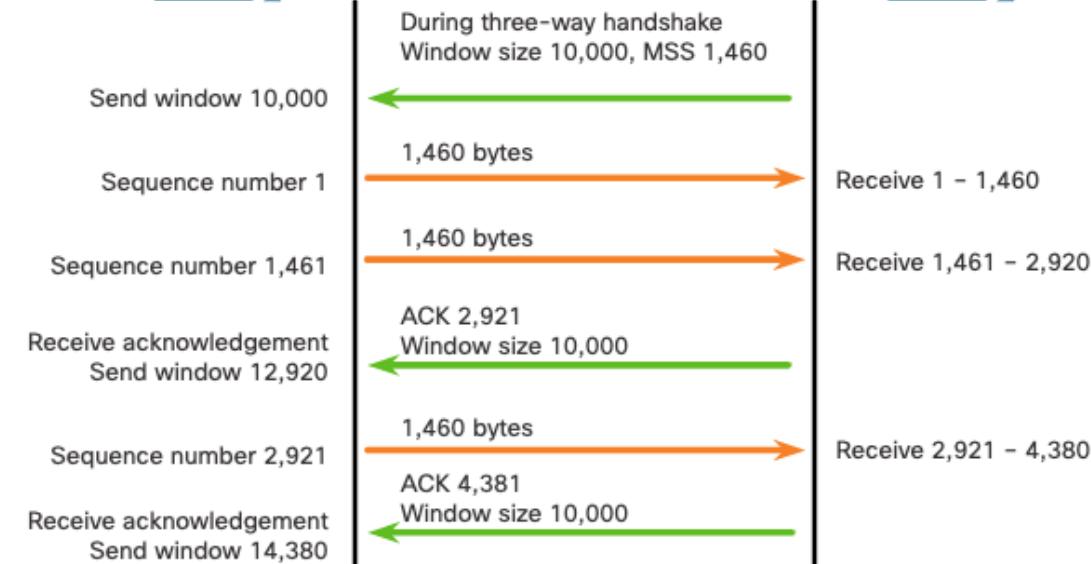
TCP Flow Control – Window Size and Acknowledgments

TCP also provides mechanisms for flow control as follows:

- Flow control is the amount of data that the destination can receive and process reliably.
- Flow control helps maintain the reliability of TCP transmission by adjusting the rate of data flow between source and destination for a given session.



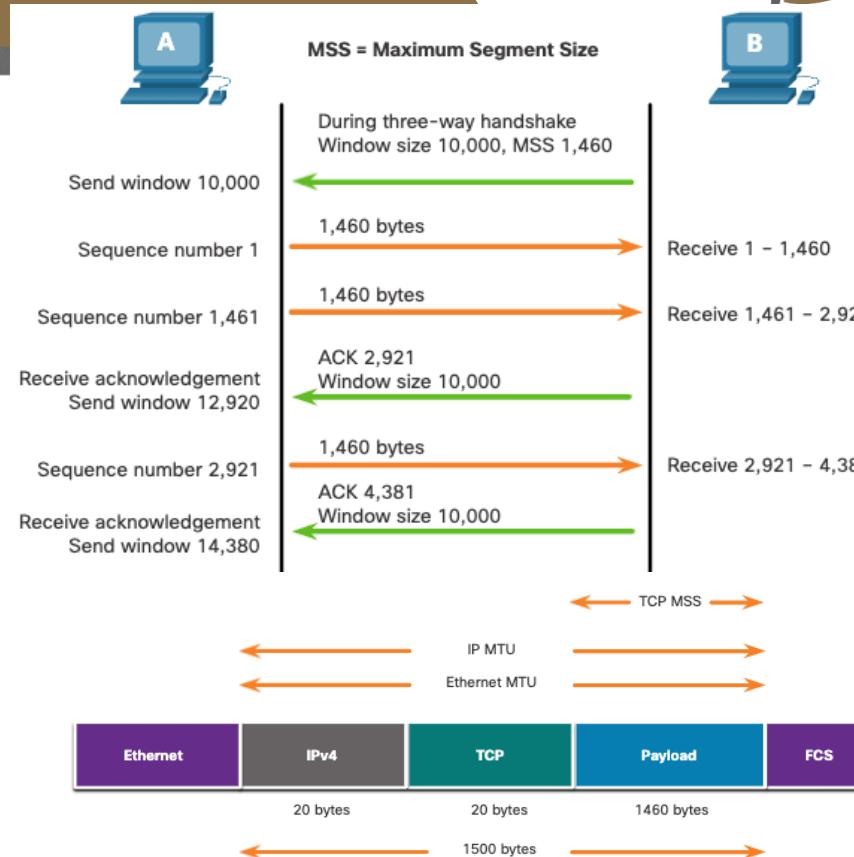
MSS = Maximum Segment Size



TCP Flow Control – Maximum Segment Size

Maximum Segment Size (MSS) is the maximum amount of data that the destination device can receive.

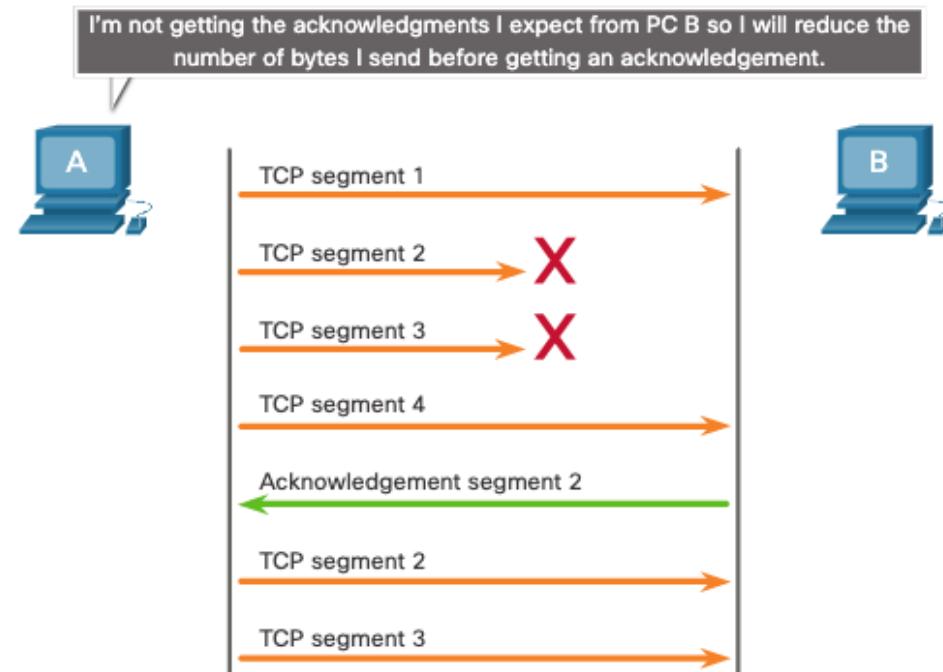
- A common MSS is 1,460 bytes when using IPv4.
- A host determines the value of its MSS field by subtracting the IP and TCP headers from the Ethernet maximum transmission unit (MTU), which is 1500 bytes be default.
- 1500 minus 40 (20 bytes for the IPv4 header and 20 bytes for the TCP header) leaves 1460 bytes.



TCP Flow Control – Congestion Avoidance

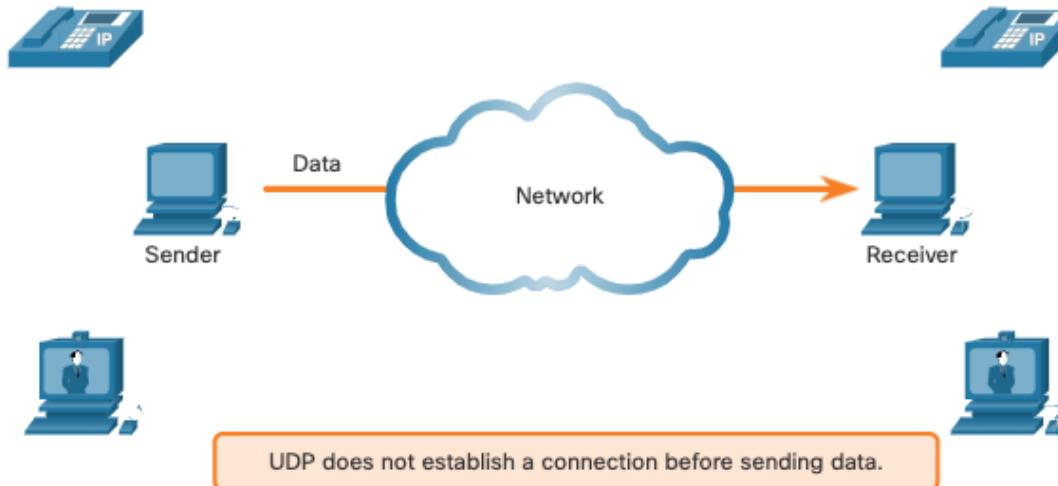
When congestion occurs on a network, it results in packets being discarded by the overloaded router.

To avoid and control congestion, TCP employs several congestion handling mechanisms, timers, and algorithms.



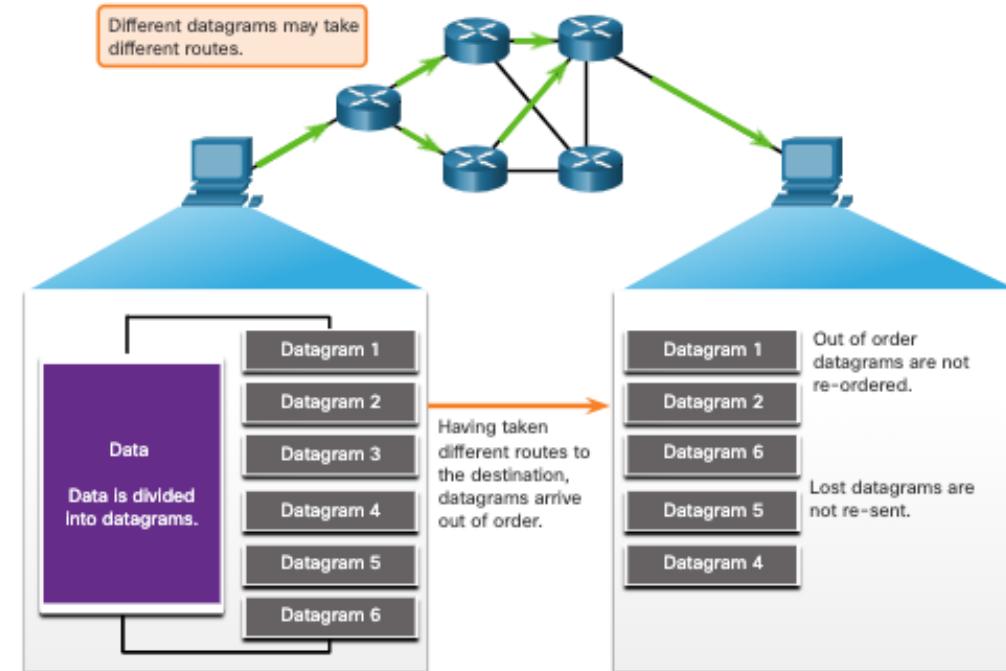
UDP Low Overhead versus Reliability

UDP does not establish a connection. UDP provides low overhead data transport because it has a small datagram header and no network management traffic.



UDP Datagram Reassembly

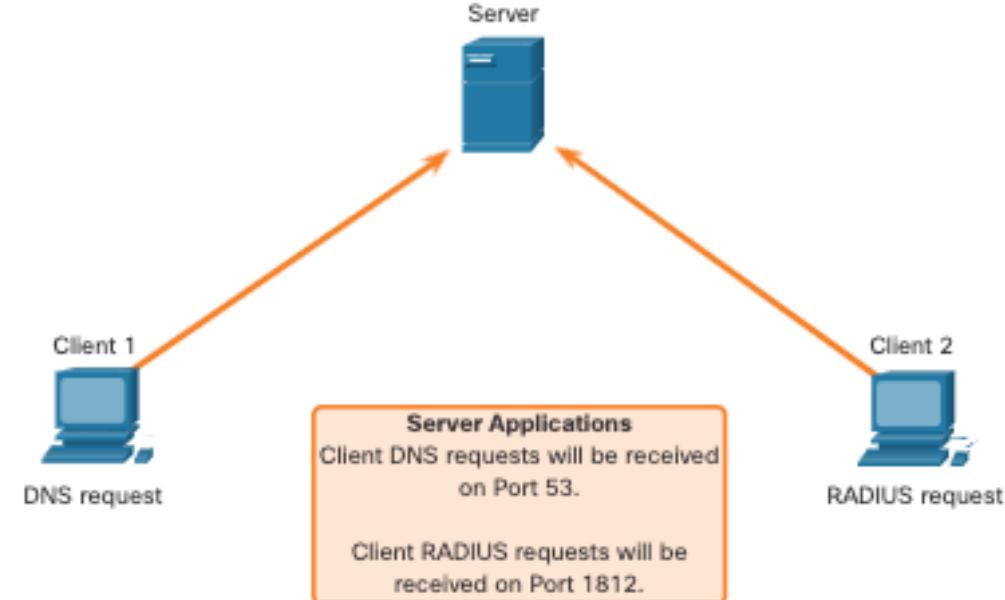
- UDP does not track sequence numbers the way TCP does.
- UDP has no way to reorder the datagrams into their transmission order.
- UDP simply reassembles the data in the order that it was received and forwards it to the application.



UDP Server Processes and Requests

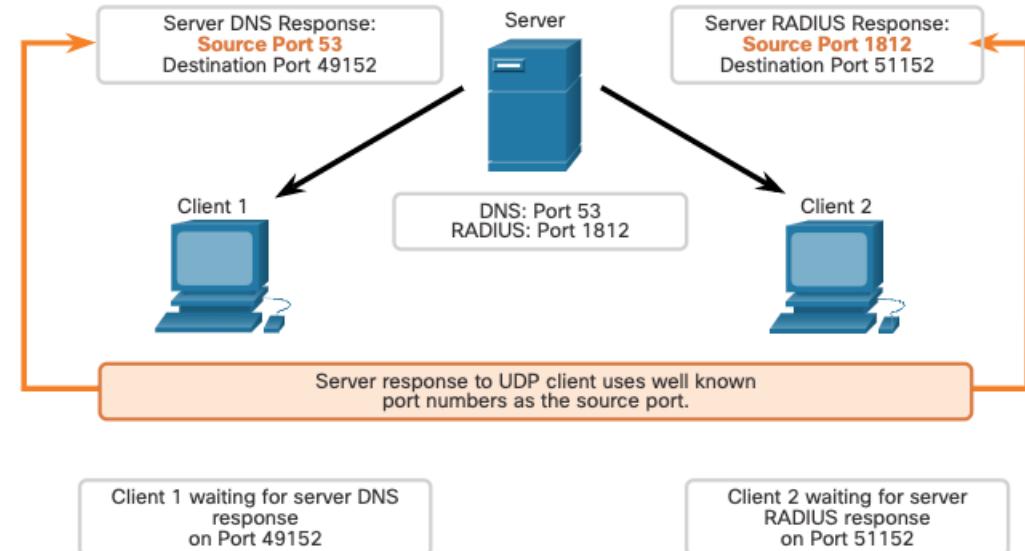
UDP-based server applications are assigned well-known or registered port numbers.

UDP receives a datagram destined for one of these ports, it forwards the application data to the appropriate application based on its port number.



UDP Client Processes

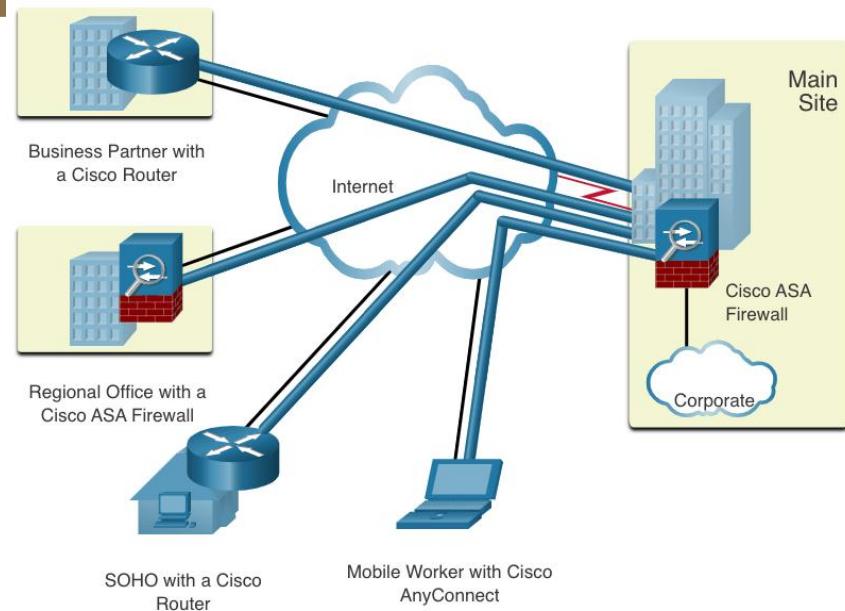
- The UDP client process dynamically selects a port number from the range of port numbers and uses this as the source port for the conversation.
- The destination port is usually the well-known or registered port number assigned to the server process.
- After a client has selected the source and destination ports, the same pair of ports are used in the header of all datagrams in the transaction.



VPN and IPsec Concepts

Virtual Private Networks

- Virtual private networks (VPNs) to create end-to-end private network connections.
- A VPN is virtual in that it carries information within a private network, but that information is actually transported over a public network.
- A VPN is private in that the traffic is encrypted to keep the data confidential while it is transported across the public network.



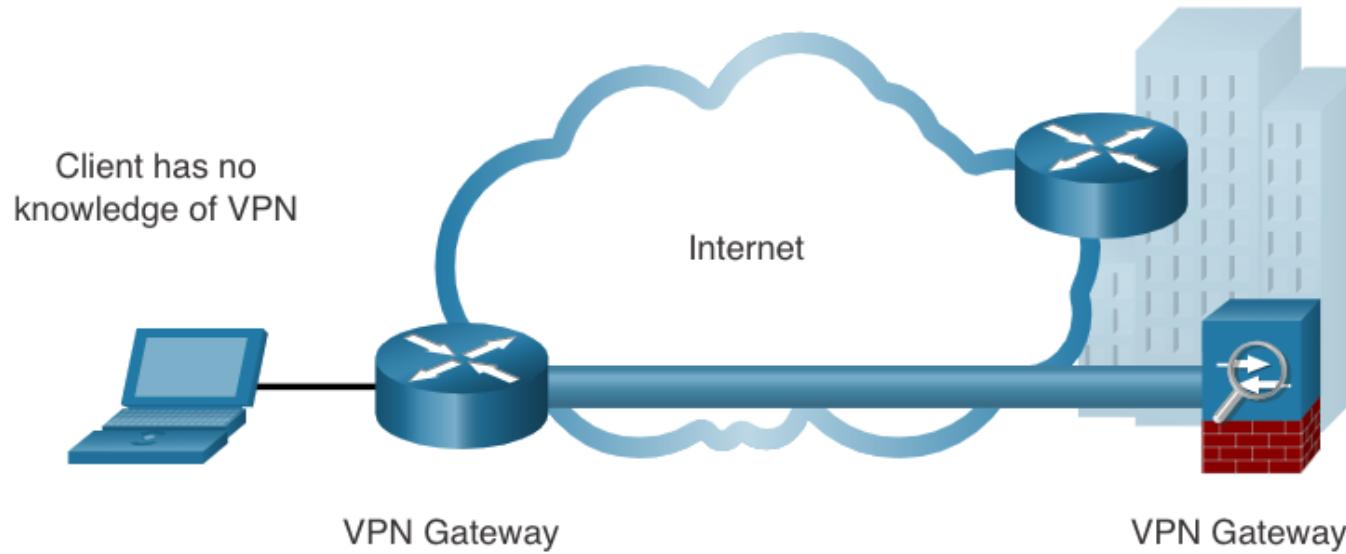
VPN Benefits

- Modern VPNs now support encryption features, such as Internet Protocol Security (IPsec) and Secure Sockets Layer (SSL) VPNs to secure network traffic between sites.
- Major benefits of VPNs are shown in the table:

Benefit	Description
Cost Savings	Organizations can use VPNs to reduce their connectivity costs while simultaneously increasing remote connection bandwidth.
Security	Encryption and authentication protocols protect data from unauthorized access.
Scalability	VPNs allow organizations to use the internet, making it easy to add new users without adding significant infrastructure.
Compatibility	VPNs can be implemented across a wide variety of WAN link options including broadband technologies. Remote workers can use these high-speed connections to gain secure access to corporate networks.

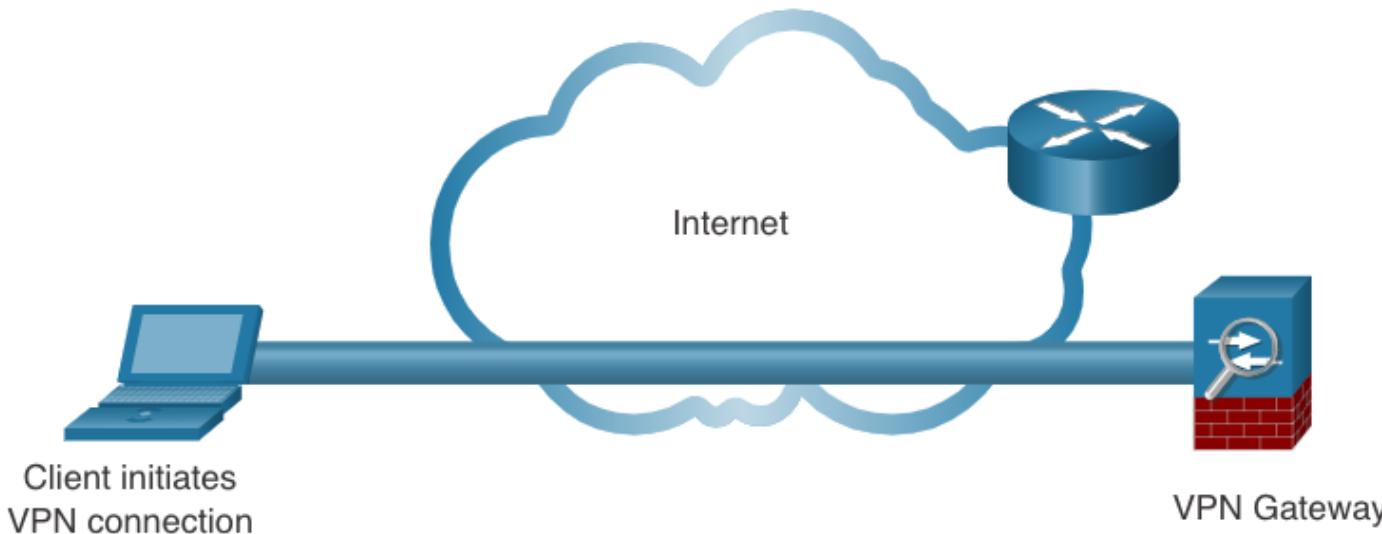
Site-to-Site and Remote Access VPNs

A site-to-site VPN is terminated on VPN gateways. VPN traffic is only encrypted between the gateways. Internal hosts have no knowledge that a VPN is being used.



Site-to-Site and Remote Access VPNs

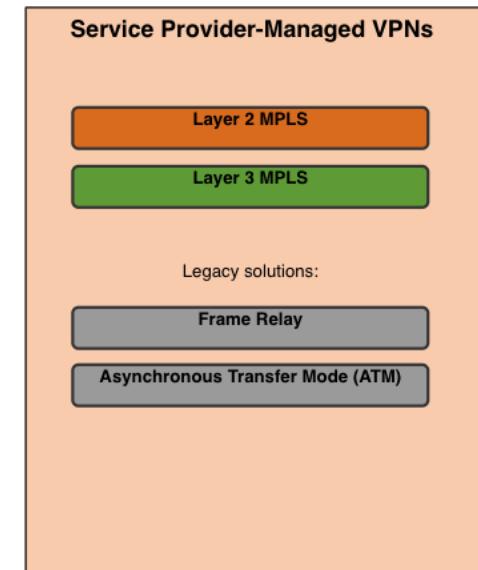
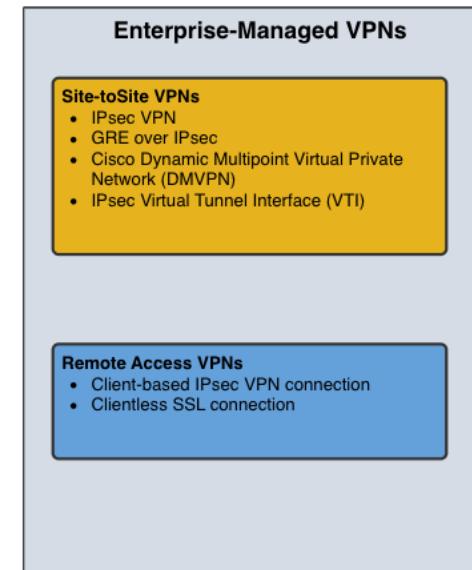
A remote-access VPN is dynamically created to establish a secure connection between a client and a VPN terminating device.



Enterprise and Service Provider VPNs

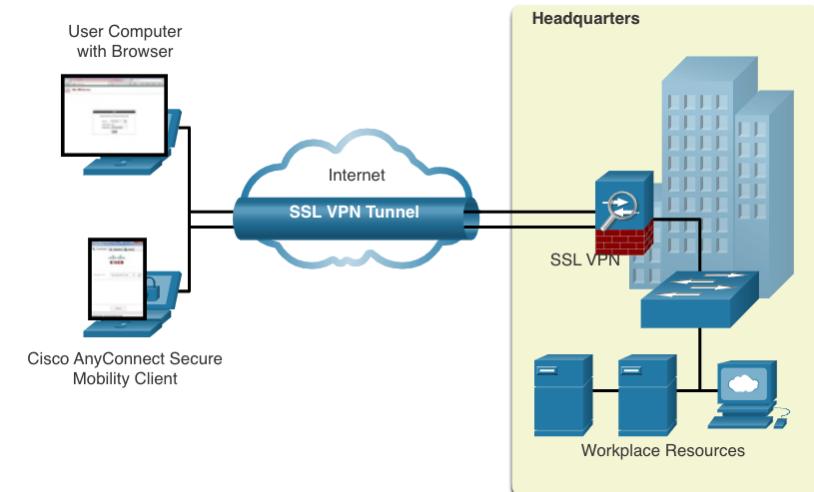
VPNs can be managed and deployed as:

- **Enterprise VPNs** - common solution for securing enterprise traffic across the internet. Site-to-site and remote access VPNs are created and managed by the enterprise using IPsec and SSL VPNs.
- **Service Provider VPNs** - created and managed by the provider network. The provider uses Multiprotocol Label Switching (MPLS) at Layer 2 or Layer 3 to create secure channels between an enterprise's sites, effectively segregating the traffic from other customer traffic.



Remote-Access VPNs

- Remote-access VPNs let remote and mobile users securely connect to the enterprise.
- Remote-access VPNs are typically enabled dynamically by the user when required and can be created using either IPsec or SSL.
- **Clientless VPN connection** -The connection is secured using a web browser SSL connection.
- **Client-based VPN connection** - VPN client software such as Cisco AnyConnect Secure Mobility Client must be installed on the remote user's end device.



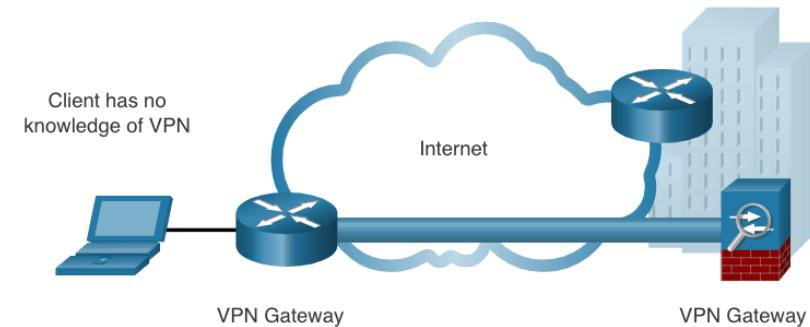
SSL VPNs

SSL uses the public key infrastructure and digital certificates to authenticate peers. The type of VPN method implemented is based on the access requirements of the users and the organization's IT processes. The table compares IPsec and SSL remote access deployments.

Feature	IPsec	SSL
Applications supported	Extensive – All IP-based applications	Limited – Only web-based applications and file sharing
Authentication strength	Strong – Two-way authentication with shared keys or digital certificates	Moderate – one-way or two-way authentication
Encryption strength	Strong – Key lengths 56 – 256 bits	Moderate to strong - Key lengths 40 – 256 bits
Connection complexity	Medium – Requires VPN client installed on a host	Low – Requires web browser on a host
Connection option	Limited – Only specific devices with specific configurations can connect	Extensive – Any device with a web browser can connect

Site-to-Site IPsec VPNs

- Site-to-site VPNs connect networks across an untrusted network such as the internet.
- End hosts send and receive normal unencrypted TCP/IP traffic through a VPN gateway.
- The VPN gateway encapsulates and encrypts outbound traffic from a site and sends the traffic through the VPN tunnel to the VPN gateway at the target site. The receiving VPN gateway strips the headers, decrypts the content, and relays the packet toward the target host inside its private network.



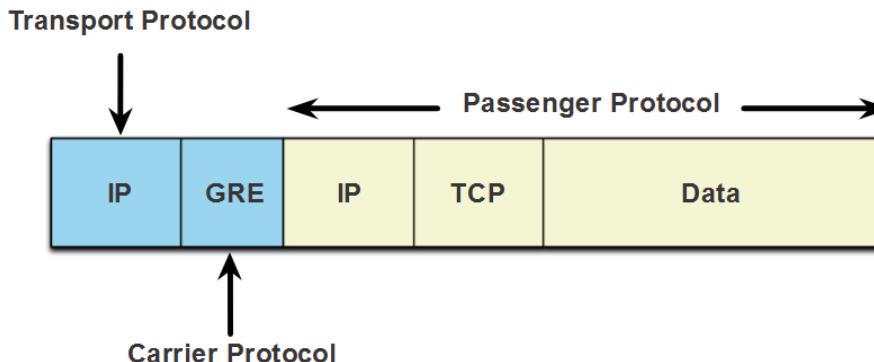
GRE over IPsec

- Generic Routing Encapsulation (GRE) is a non-secure site-to-site VPN tunneling protocol.
- A GRE tunnel can encapsulate various network layer protocols as well as multicast and broadcast traffic.
- GRE does not by default support encryption; and therefore, it does not provide a secure VPN tunnel.
- A GRE packet can be encapsulated into an IPsec packet to forward it securely to the destination VPN gateway.
- Standard IPsec VPNs (non-GRE) can only create secure tunnels for unicast traffic.
- Encapsulating GRE into IPsec allows multicast routing protocol updates to be secured through a VPN.

GRE over IPsec (Cont.)

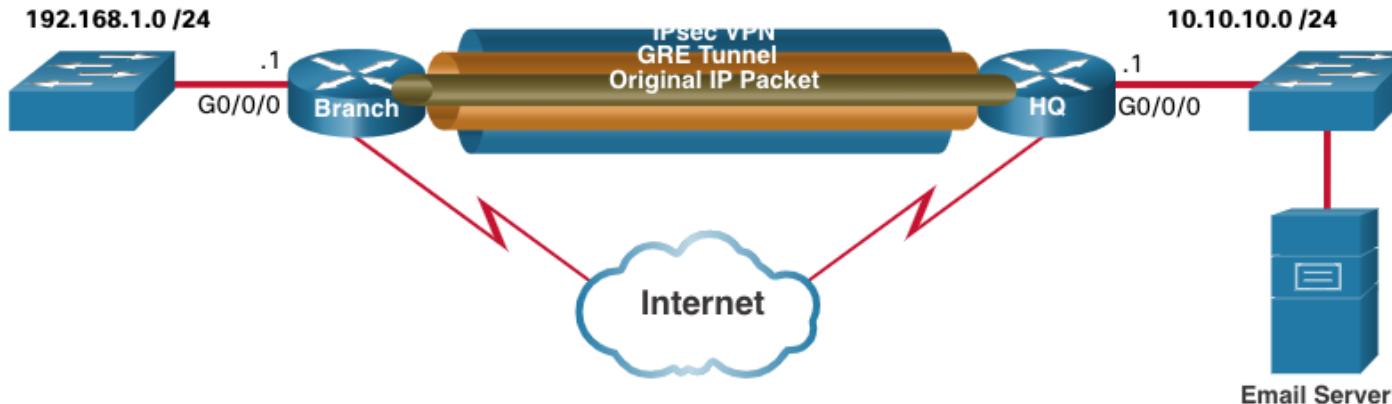
The terms used to describe the encapsulation of GRE over IPsec tunnel are passenger protocol, carrier protocol, and transport protocol.

- **Passenger protocol** – This is the original packet that is to be encapsulated by GRE. It could be an IPv4 or IPv6 packet, a routing update, and more.
- **Carrier protocol** – GRE is the carrier protocol that encapsulates the original passenger packet.
- **Transport protocol** – This is the protocol that will actually be used to forward the packet. This could be IPv4 or IPv6.



GRE over IPsec (Cont.)

For example, Branch and HQ need to exchange OSPF routing information over an IPsec VPN. GRE over IPsec is used to support the routing protocol traffic over the IPsec VPN. Specifically, the OSPF packets (i.e., passenger protocol) would be encapsulated by GRE (i.e., carrier protocol) and subsequently encapsulated in an IPsec VPN tunnel.



Dynamic Multipoint VPNs

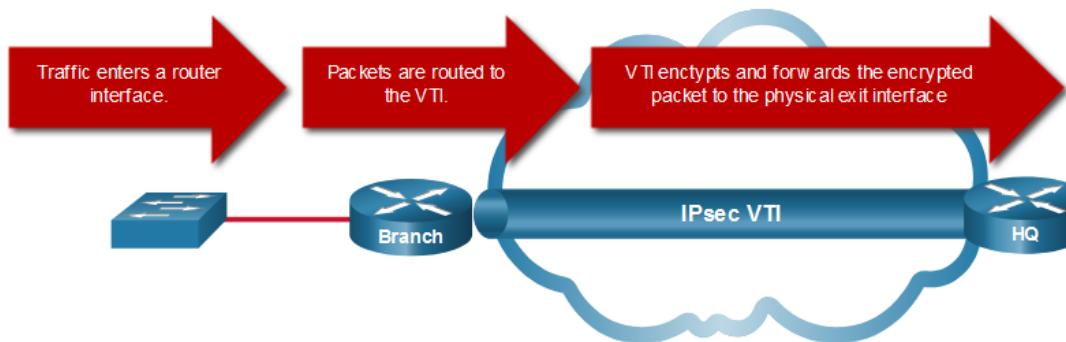
Site-to-site IPsec VPNs and GRE over IPsec are not sufficient when the enterprise adds many more sites. Dynamic Multipoint VPN (DMVPN) is a Cisco software solution for building multiple VPNs in an easy, dynamic, and scalable manner.

- DMVPN simplifies the VPN tunnel configuration and provides a flexible option to connect a central site with branch sites.
- It uses a hub-and-spoke configuration to establish a full mesh topology.
- Spoke sites establish secure VPN tunnels with the hub site.
- Each site is configured using Multipoint Generic Routing Encapsulation (mGRE). The mGRE tunnel interface allows a single GRE interface to dynamically support multiple IPsec tunnels.
- Spoke sites can also obtain information about each other, and alternatively build direct tunnels between themselves (spoke-to-spoke tunnels).

IPsec Virtual Tunnel Interface

IPsec Virtual Tunnel Interface (VTI) simplifies the configuration process required to support multiple sites and remote access.

- IPsec VTI configurations are applied to a virtual interface instead of static mapping the IPsec sessions to a physical interface.
- IPsec VTI is capable of sending and receiving both IP unicast and multicast encrypted traffic. Therefore, routing protocols are automatically supported without having to configure GRE tunnels.
- IPsec VTI can be configured between sites or in a hub-and-spoke topology.



Service Provider MPLS VPNs

Today, service providers use MPLS in their core network. Traffic is forwarded through the MPLS backbone using labels. Traffic is secure because service provider customers cannot see each other's traffic.

- MPLS can provide clients with managed VPN solutions; therefore, securing traffic between client sites is the responsibility of the service provider.
- There are two types of MPLS VPN solutions supported by service providers:
- **Layer 3 MPLS VPN** - The service provider participates in customer routing by establishing a peering between the customer's routers and the provider's routers.
- **Layer 2 MPLS VPN** - The service provider is not involved in the customer routing. Instead, the provider deploys a Virtual Private LAN Service (VPLS) to emulate an Ethernet multiaccess LAN segment over the MPLS network. No routing is involved. The customer's routers effectively belong to the same multiaccess network.

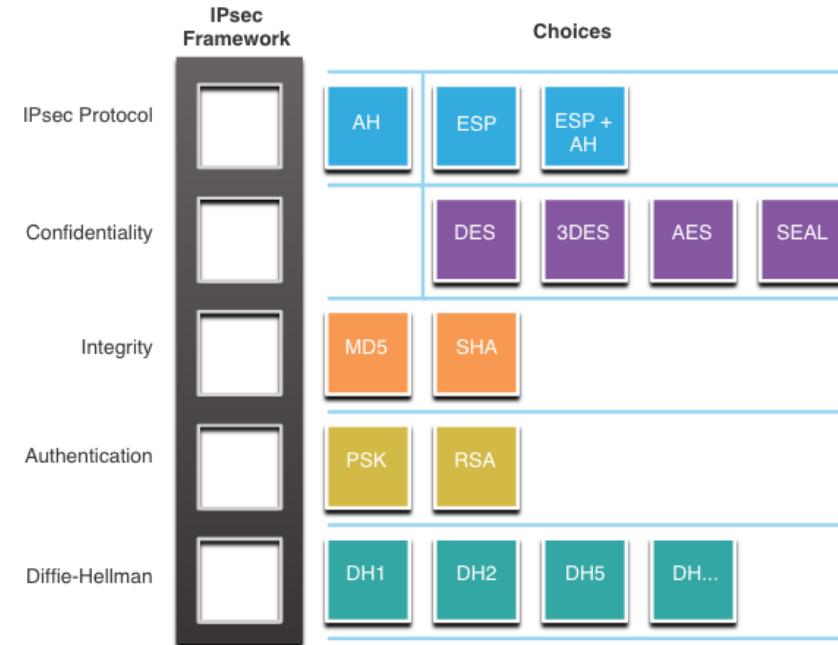
IPsec Technologies

IPsec is an IETF standard that defines how a VPN can be secured across IP networks. IPsec protects and authenticates IP packets between source and destination and provides these essential security functions:

- **Confidentiality** - Uses encryption algorithms to prevent cybercriminals from reading the packet contents.
- **Integrity** - Uses hashing algorithms to ensure that packets have not been altered between source and destination.
- **Origin authentication** - Uses the Internet Key Exchange (IKE) protocol to authenticate source and destination.
- **Diffie-Hellman** – Used to secure key exchange.

IPsec Technologies (Cont.)

- IPsec is not bound to any specific rules for secure communications.
- IPsec can easily integrate new security technologies without updating existing IPsec standards.
- The open slots in the IPsec framework shown in the figure can be filled with any of the choices that are available for that IPsec function to create a unique security association (SA).



IPsec Protocol Encapsulation

Choosing the IPsec protocol encapsulation is the first building block of the framework.

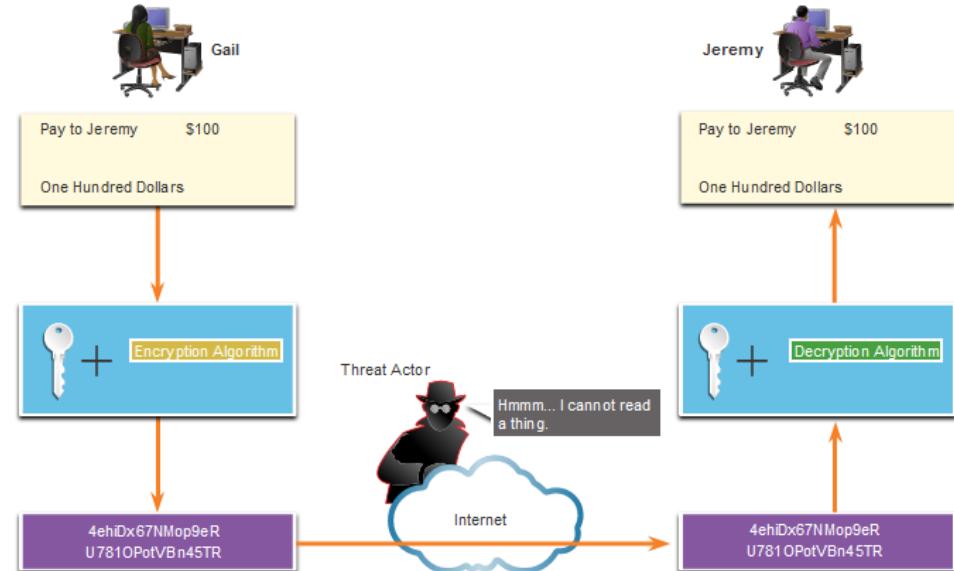
- IPsec encapsulates packets using Authentication Header (AH) or Encapsulation Security Protocol (ESP).
 - The choice of AH or ESP establishes which other building blocks are available.
 - AH is appropriate only when confidentiality is not required or permitted.
 - ESP provides both confidentiality and authentication.



Confidentiality

The degree of confidentiality depends on the encryption algorithm and the length of the key used in the encryption algorithm.

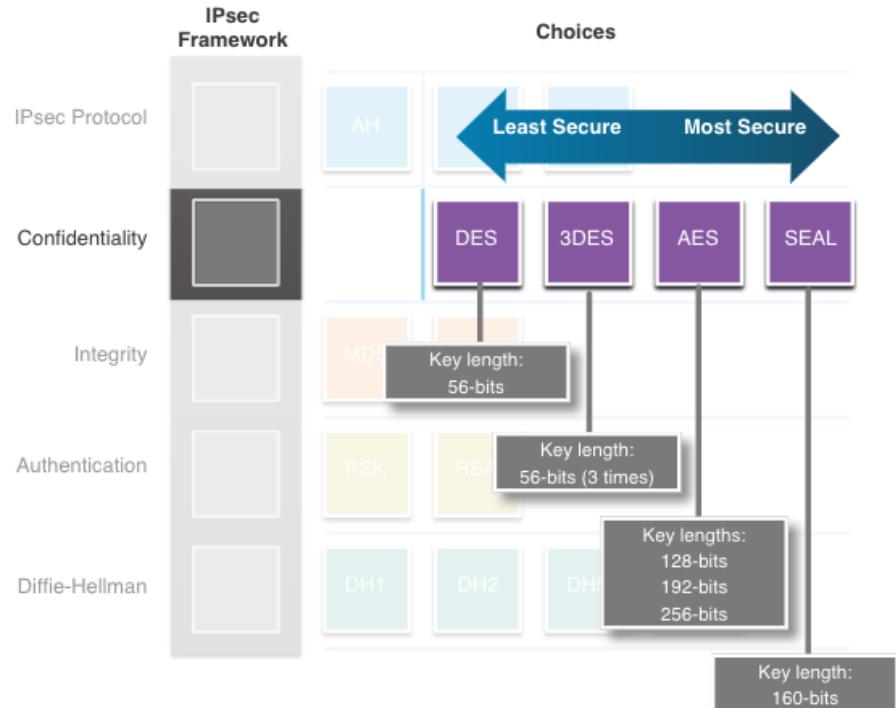
The number of possibilities to try to hack the key is a function of the length of the key - the shorter the key, the easier it is to break.



Confidentiality (Cont.)

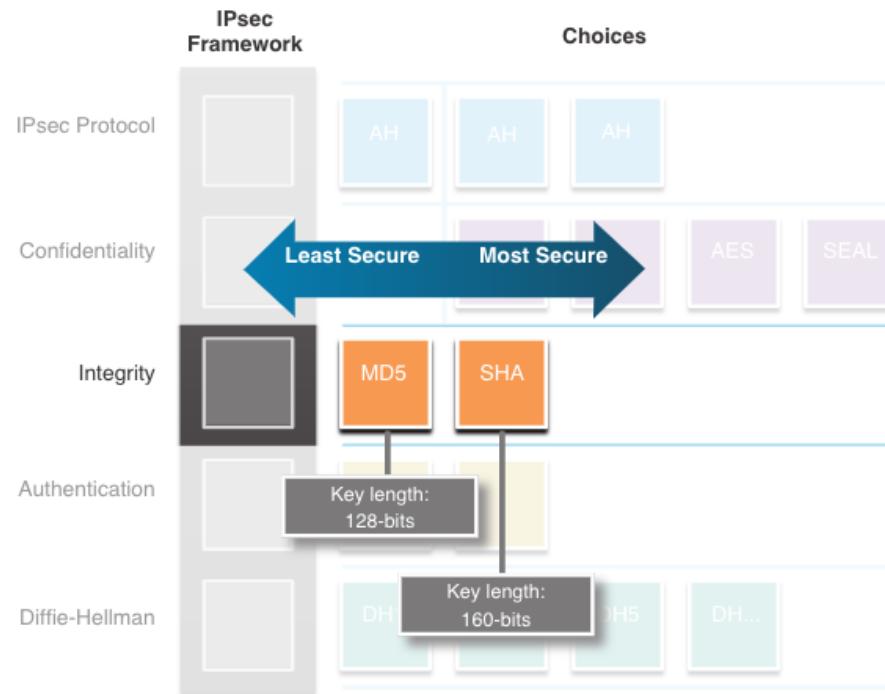
The encryption algorithms highlighted in the figure are all symmetric key cryptosystems:

- DES uses a 56-bit key.
- 3DES uses three independent 56-bit encryption keys per 64-bit block.
- AES offers three different key lengths: 128 bits, 192 bits, and 256 bits.
- SEAL is a stream cipher, which means it encrypts data continuously rather than encrypting blocks of data. SEAL uses a 160-bit key.



Integrity

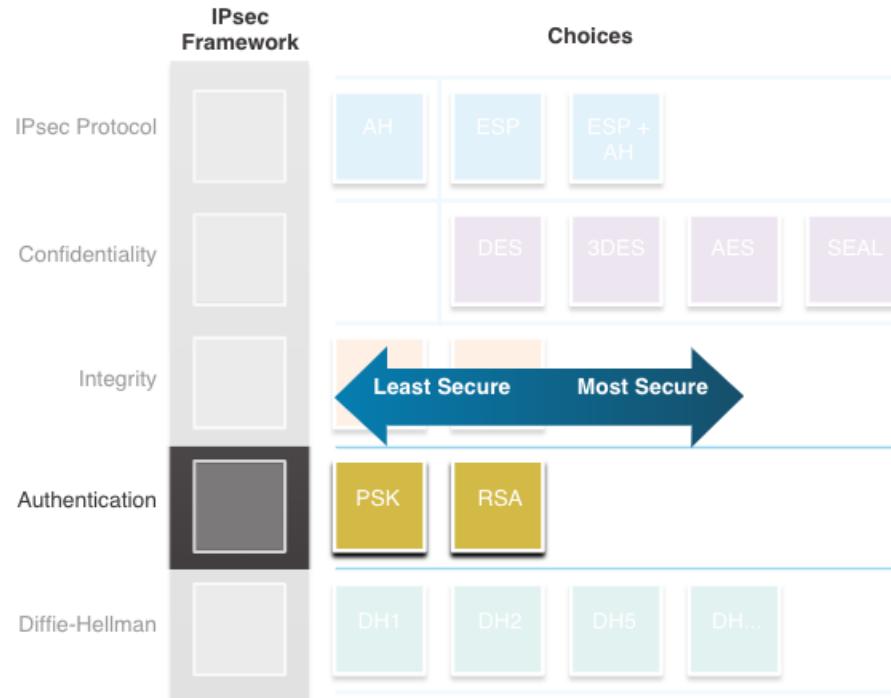
- Data integrity means that the data has not changed in transit.
- A method of proving data integrity is required.
- The Hashed Message Authentication Code (HMAC) is a data integrity algorithm that guarantees the integrity of the message using a hash value.
- Message-Digest 5 (MD5) uses a 128-bit shared-secret key.
- The Secure Hash Algorithm (SHA) uses a 160-bit secret key.



Authentication

There are two IPsec peer authentication methods:

1. **Pre-shared key (PSK)** - (PSK) value is entered into each peer manually.
 - Easy to configure manually
 - Does not scale well
 - Must be configured on every peer
2. **Rivest, Shamir, and Adleman (RSA)** - authentication uses digital certificates to authenticate the peers.
 - Each peer must authenticate its opposite peer before the tunnel is considered secure.



Secure Key Exchange with Diffie - Hellman

DH provides allows two peers to establish a shared secret key over an insecure channel.

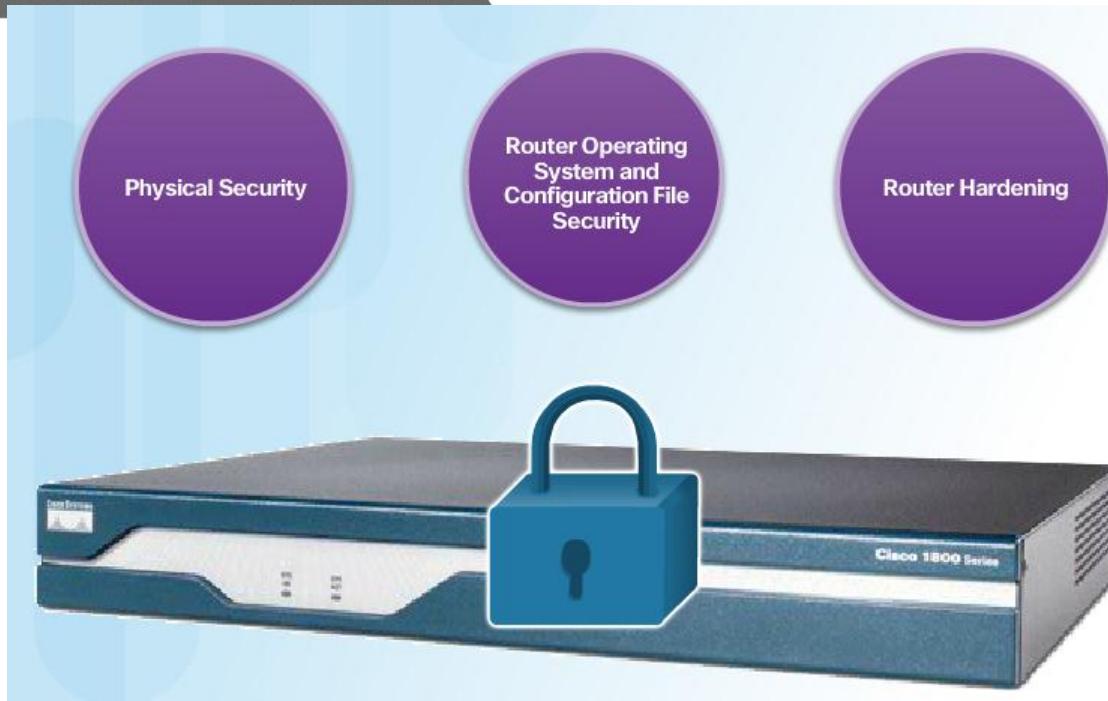
Variations of the DH key exchange are specified as DH groups:

- DH groups 1, 2, and 5 should no longer be used.
- DH groups 14, 15, and 16 use larger key sizes with 2048 bits, 3072 bits, and 4096 bits, respectively
- DH groups 19, 20, 21 and 24 with respective key sizes of 256 bits, 384 bits, 521 bits, and 2048 bits support Elliptical Curve Cryptography (ECC), which reduces the time needed to



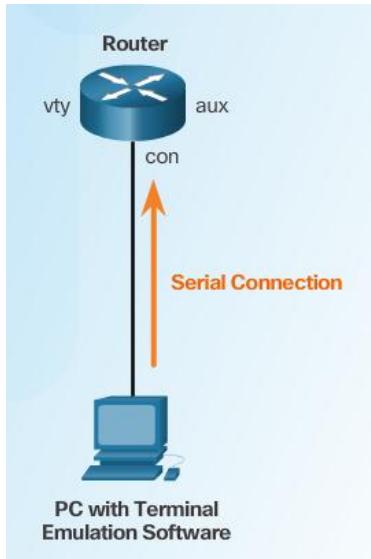
Securing Network Devices

Three Areas of Router Security

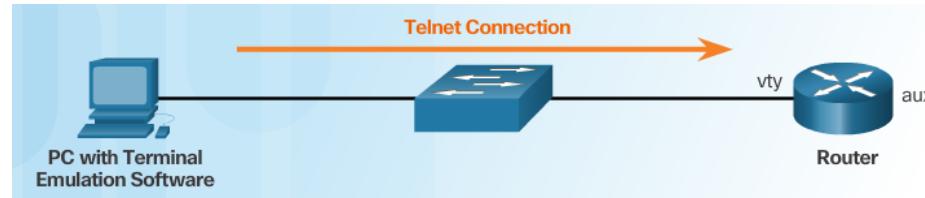


Secure Local and Remote Access

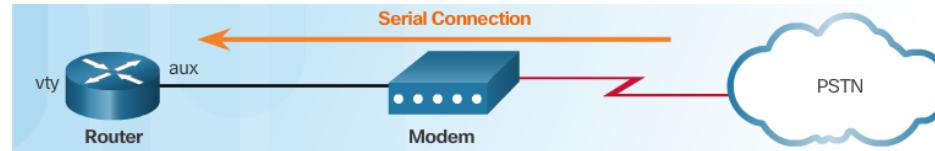
Local Access



Remote Access Using Telnet



Remote Access Using Modem and Aux Port



Increasing Access Security

```
R1(config)# security passwords min-length 10
R1(config)# service password-encryption
R1(config)# line vty 0 4
R1(config-line)# exec-timeout 3 30
R1(config-line)# line console 0
R1(config-line)# exec-timeout 3 30
```

```
R1(config)# service password-encryption
R1(config)# exit
R1# show running-config

<output omitted>

line con 0
exec-timeout 3 30
password 7 094F471A1A0A
login
line aux 0
exec-timeout 3 30
password 7 094F471A1A0A
login
line vty 0 4
password 7 094F471A1A0A
login
```

Cisco Cracker

094F471A1A0A

Crack it

Password = Cisco

Secret Password Algorithms

Guidelines:

- Configure all secret passwords using type 8 or type 9 passwords
- Use the enable algorithm-type command syntax to enter an unencrypted password

```
Router(config)#
enable algorithm-type {md5 | scrypt | sha256} secret unencrypted-password
```

- Use the username name algorithm-type command to specify type 9 encryption

```
Router(config)#
username name algorithm-type {md5 | scrypt | sha256} secret unencrypted-password
```

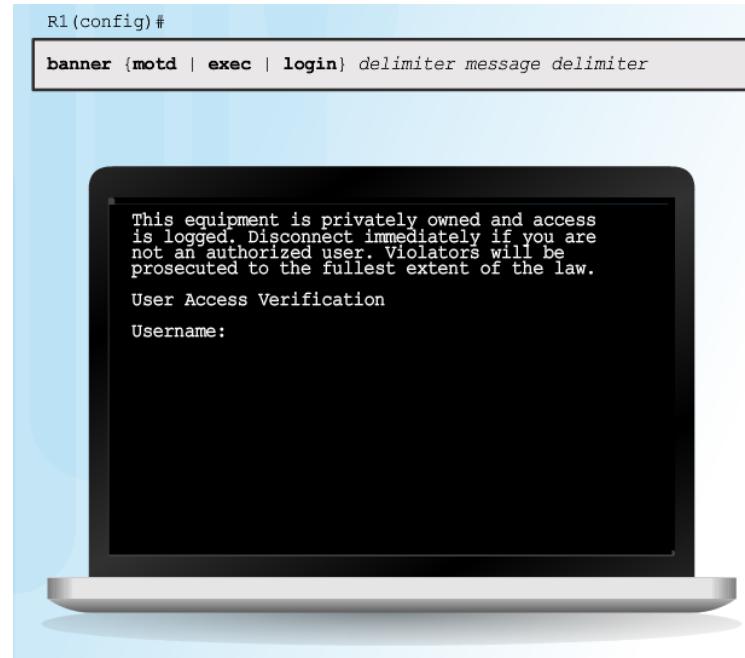
Securing Line Access

```
R1(config)# username Bob algorithm-type scrypt secret cisco54321
R1(config)# line con 0
R1(config-line)# no password
R1(config-line)# login local
R1(config-line)# exit
R1(config)# line aux 0
R1(config-line)# no password
R1(config-line)# login local
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
```

Enhancing the Login Process

Virtual login security enhancements:

- Implement delays between successive login attempts
- Enable login shutdown if DoS attacks are suspected
- Generate system-logging messages for login detection



Configuring Login Enhancement Features

```
R1(config)#
login block-for seconds attempts tries within seconds
```

```
R1(config)#
login quiet-mode access-class {acl-name|acl-number}
```

```
R1(config)#
login delay seconds
```

```
R1(config)#
login on-success log [every login]
```

```
R1(config)#
login on-failure log [every login]
```

Enable Login Enhancements

Command Syntax: **login block-for**

```
router(config)#  
login block-for seconds attempts tries within seconds
```

```
R1(config)# login block-for 120 attempts 5 within 60
```

Example: **login quiet-mode access-class**

```
R1(config)# ip access-list standard PERMIT-ADMIN  
R1(config-std-nacl)# remark Permit only Administrative hosts  
R1(config-std-nacl)# permit 192.168.10.10  
R1(config-std-nacl)# permit 192.168.11.10  
R1(config-std-nacl)# exit  
R1(config)# login quiet-mode access-class PERMIT-ADMIN
```

Example: **login delay**

```
R1(config)# login delay 3
```

Logging Failed Attempts

Generate Login Syslog Messages

```
R1(config)# login on-success log [every login]
R1(config)# login on-failure log [every login]
R1(config)# security authentication failure rate threshold-rate log
```

Example: show login failures

```
R1# show login failures
Total failed logins: 22
Detailed information about last 50 failures

Username      SourceIPAddr      lPort Count TimeStamp
admin         1.1.2.1           23    5     15:38:54 UTC Wed Dec 10 2008
Admin         10.10.10.10       23   13     15:58:43 UTC Wed Dec 10 2008
admin         10.10.10.10       23    3     15:57:14 UTC Wed Dec 10 2008
cisco         10.10.10.10       23    1     15:57:21 UTC Wed Dec 10 2008

R1#
```

Steps for Configuring SSH

Example SSH Configuration

```
R1# conf t
R1(config)# ip domain-name span.com
R1(config)# crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.span.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)

R1(config)#
*Feb 16 21:18:41.977: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)# ip ssh version 2
R1(config)# username Bob algorithm-type scrypt secret cisco54321
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# end
R1#
```

Example Verification of SSH

```
R1# show crypto key mypubkey rsa
% Key pair was generated at: 21:18:41 UTC Feb 16 2015
Key name: R1.span.com
Key type: RSA KEYS
Storage Device: not specified
Usage: General Purpose Key
Key is not exportable.
Key Data:
 30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CF35DB
 A58A1BDB F7C7E600 F189C2F3 2EC6E584 D923EE5B 71841D98 B5472A03 D19CD620
 ED125825 5A58412B B7F29234 DE2A1809 6C421AC3 07F298E6 80BE149D 2A262E13
 7488SDAF CAC8F187 B11111AF A413E76F 6C157CDF DFEF0D82 2961B58C BE1CAD21
 176E82B9 6D81F893 06E66C93 94E1C508 887462F6 90AC63CE 5E169845 C1020301 0001
% Key pair was generated at: 21:18:42 UTC Feb 16 2015
Key name: R1.span.com.server
Key type: RSA KEYS
Temporary key
  Usage: Encryption Key
  Key is not exportable.
  Key Data:
    307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00AB914D 8172DFBE
    DE57ACAA 78844239 1F3B5942 3943AC0D F54E7746 3895CF54 606C3961 8A44FEB3
    1A019F27 D9E71AAE FC73F423 A59CB8F5 50289272 3392CEBC 4C3CBD6D DB9233DE
    9DDDD9DAD 79D56165 4293AA62 FD1CBAB2 7AB859DC 2890C795 ED020301 0001
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# crypto key zeroize rsa
% All keys will be removed.
% All router certs issued using these keys will also be removed.
Do you really want to remove these keys? [yes/no]: yes
R1(config)#

```

Modifying the SSH Configuration

```
R1# show ip ssh
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Authentication timeout: 120 secs; Authentication retries: 3
<output omitted>

R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip ssh time-out 60
R1(config)# ip ssh authentication-retries 2
R1(config)# ^Z
R1#
*Feb 16 21:23:51.237: %SYS-5-CONFIG_I: Configured from console by console
R1# show ip ssh
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Authentication timeout: 60 secs; Authentication retries: 2
<output omitted>
```

Connecting to an SSH-Enabled Router

Two ways to connect:

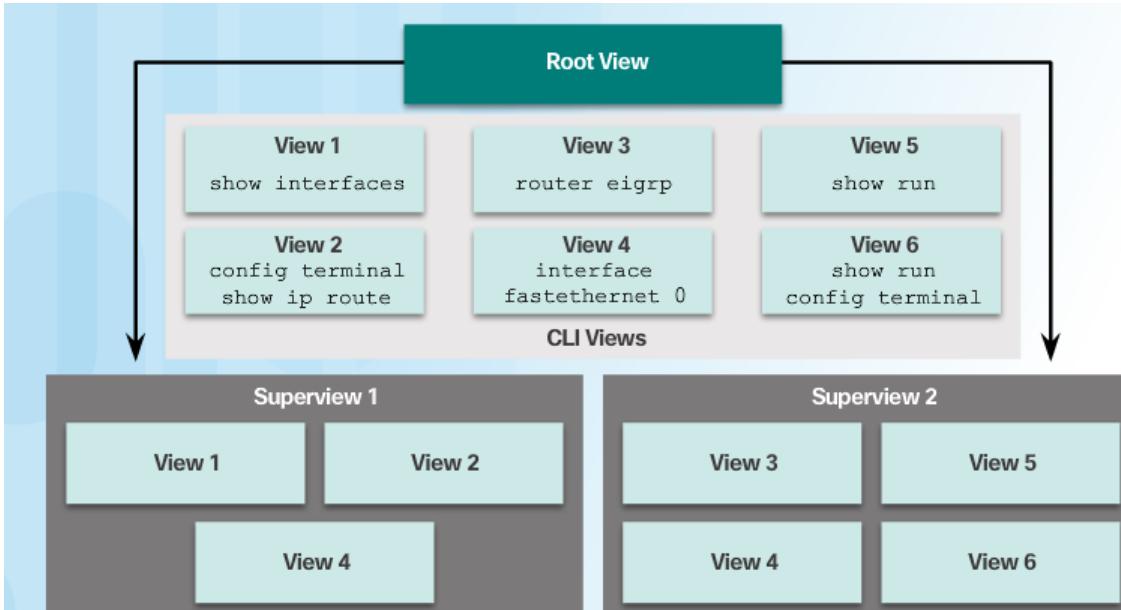
Enable SSH and use a Cisco router as an SSH server or SSH client.

As a server, the router can accept SSH client connections

As a client, the router can connect via SSH to another SSH-enabled router

Use an SSH client running on a host, such as PuTTY, OpenSSH, or TeraTerm.

Role-Based Views



Superviews contain Views but not commands. Two Superviews can use the same View.
 For example, both Superview 1 and Superview 2 can have CLI View 4 placed inside.

Configuring Role-Based Views

Step 1

```
Router#
```

```
enable [view [view-name]]
```

Step 2

```
Router(config)#
```

```
parser view view-name
```

Step 3

```
Router(config-view)#
```

```
secret encrypted-password
```

Step 4

```
Router(config-view)#
```

```
commands parser mode {include | include-exclusive | exclude} [all]  
[interface interface-name | command]
```

Configuring Role-Based CLI Superviews

Step 1

```
Router(config)#
  parser view view-name superview
```

Step 2

```
Router(config-view)#
  secret encrypted-password
```

Step 3

```
Router(config-view)#
  view view-name
```

Verify Role-Based CLI Views

Enable Root View and Verify All Views

```
R1# show parser view
Current view is 'JR-ADMIN'

R1# enable view
Password:

R1# show parser view
Current view is 'root'

R1# show parser view all
Views/SuperViews Present in System:
SHOWVIEW
VERIFYVIEW
REBOOTVIEW
USER *
SUPPORT *

JR-ADMIN *

-----(*) represent superview-----
R1#
```

Recovering a Router Password



1. Connect to the console port.
2. Record the configuration register setting.
3. Power cycle the router.
4. Issue the break sequence.
5. Change the default configuration register with the confreg 0x2142 command.
6. Reboot the router.
7. Press Ctrl-C to skip the initial setup procedure.
8. Put the router into privileged EXEC mode.
9. Copy the startup configuration to the running configuration.
10. Verify the configuration.
11. Change the enable secret password.
12. Enable all interfaces.
13. Change the config-register with the config-register configuration_register_setting.
14. Save the configuration changes.

Password Recovery

```
R1(config)# no service password-recovery
WARNING:
Executing this command will disable password recovery
mechanism.
Do not execute this command without another plan for
password recovery.
Are you sure you want to continue? [yes/no]: yes
R1(config)#

```

Disable Password Recovery

No Service Password Recovery

```
R1# show running-config
Building configuration...

Current configuration : 836 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service password-recovery

```

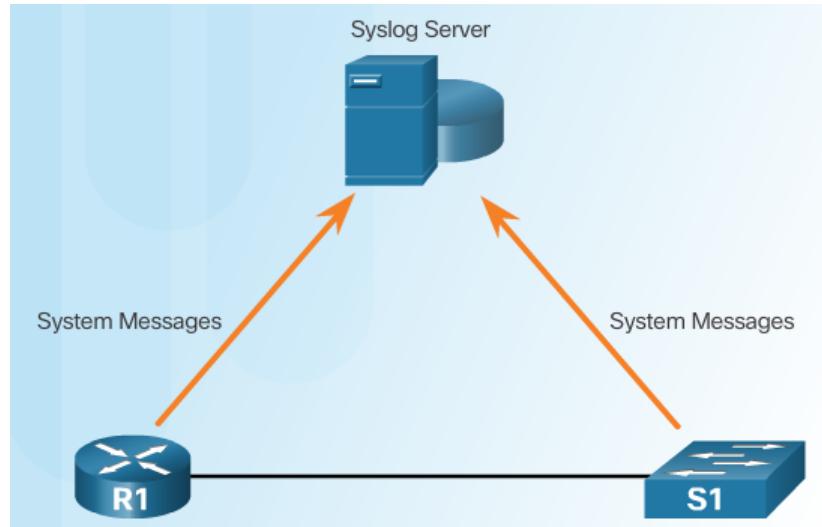
```
System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2006 by cisco Systems, Inc.
PLD version 0x10
GIO ASIC version 0x127
c1841 platform with 131072 Kbytes of main memory
Main memory is configured to 64 bit mode with parity disabled

PASSWORD RECOVERY FUNCTIONALITY IS DISABLED
program load complete, entry point: 0x8000f000, size:0xcb80

```

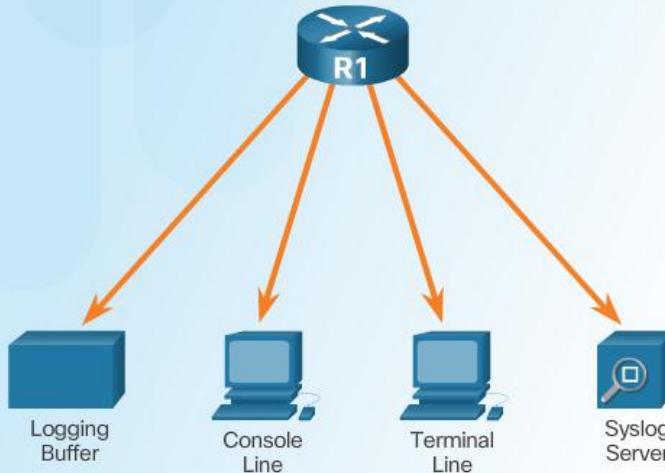
Password Recovery Functionality is Disabled

Introduction to Syslog



Syslog Operation

```
R1(config-if)# no shutdown
R1(config-if)#
000047: *Feb 19 11:36:47.779: %LINK-3-UPDOWN: Interface Serial0/0/0, changed
state to up
000048: *Feb 19 11:36:48.779: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/0, changed state to up
```



Syslog Message

Security Levels

Level	Keyword	Description	Definition
Highest Level	0	emergencies	System is unusable
	1	alerts	Immediate action is needed
	2	critical	Critical conditions exist
	3	errors	Error conditions exist
	4	warnings	Warning conditions exist
	5	notifications	Normal but significant condition
	6	informational	Informational messages only
Lowest Level	7	debugging	Debugging messages

Example Severity Levels

Syslog Level and Name	Definition	Example
0 LOG_EMERG	A panic condition normally broadcast to all users	Cisco IOS software could not load
1 LOG_ALERT	A condition that should be corrected immediately, such as a corrupted system database	Temperature too high
2 LOG_CRIT	Critical conditions; for example, device errors	Unable to allocate memory
3 LOG_ERR	Errors	Invalid memory size
4 LOG_WARNING	Warning messages	Crypto operation failed
5 LOG_NOTICE	Non-error conditions that may require special handling	Interface changed state, up or down
6 LOG_INFO	Informational messages	Packet denied by ACL
7 LOG_DEBUG	Messages that contain information that is normally used only when debugging a program	Packet type invalid

Syslog Message (Cont.)

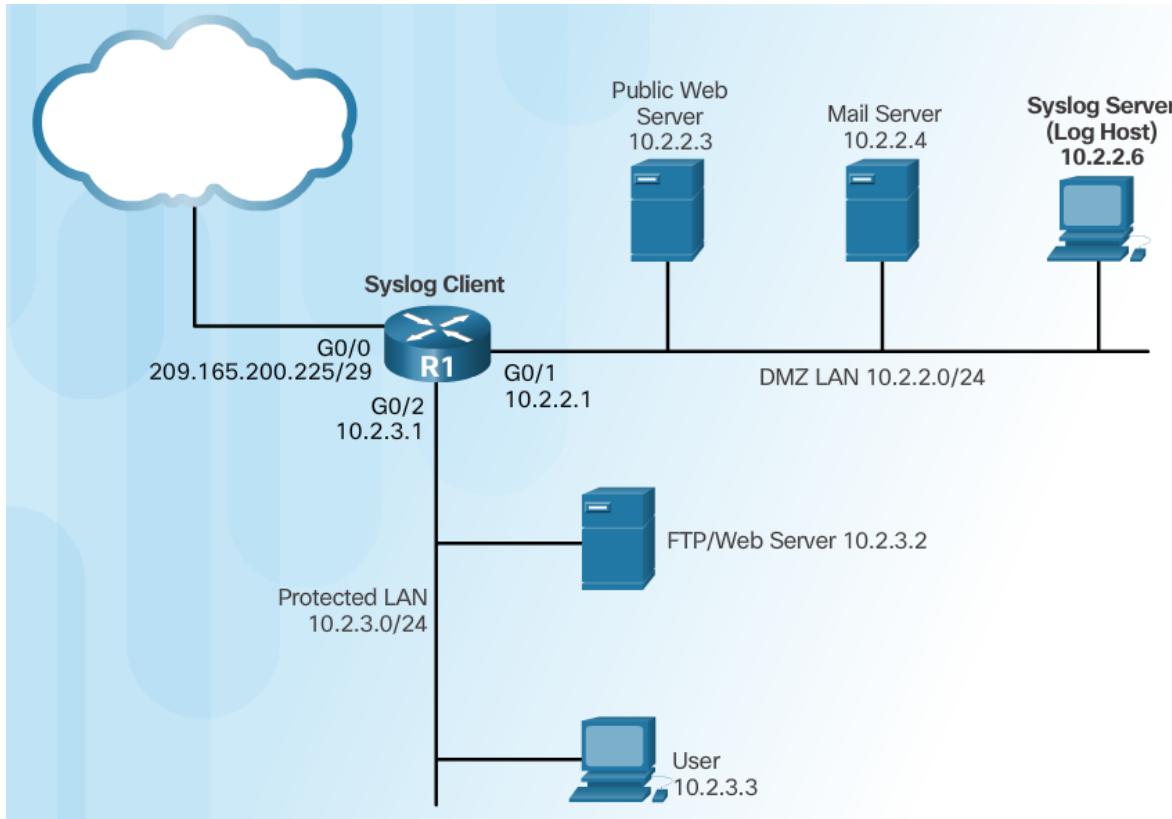
1 2 3 4 5
000048: *Feb 19 11:36:48.779: %LINEPROTO-5-UPDOWN:

Line protocol on Interface Serial0/0/0, changed state to up

6

Column 1	Column 2
1	seq no
	Stamps log messages with a sequence number if service sequence-numbers is configured.
2	timestamp
	displays if service timestamps log is configured
3	facility
	denotes the source or the cause of the system message
4	severity
	levels 0 – 7
5	MNEMONIC
	text string that uniquely describes the message
6	description
	text string containing detailed information about the event being reported

Syslog Systems



Configuring System Logging

Step 1

```
Router(config) #
```

```
logging host [hostname | ip-address]
```

Step 2 (optional)

```
Router(config) #
```

```
logging trap level
```

Step 3

```
Router(config) #
```

```
logging source-interface interface-type interface-number
```

Step 4

```
Router(config) #
```

```
logging on
```

Using NTP



Cisco AutoSecure

```
R1# auto secure
--- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security
of the router but it will not make router
absolutely secure from all security attacks ***

All the configuration done as part of AutoSecure
will be shown here. For more details of why and
how this configuration is useful, and any possible
side effects, please refer to Cisco documentation of
AutoSecure.

At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for
AutoSecure

Is this router connected to internet? [no]:yes
```

Using the Cisco AutoSecure Feature

Router#

```
auto secure [no-interact | full] [forwarding | management]
[ntp | login | ssh | firewall | tcp-intercept]
```

Parameter	Description
no-interact	(Optional) The user will not be prompted for any interactive configurations. No interactive dialogue parameters will be configured, including usernames or passwords.
full	(Optional) The user will be prompted for all interactive questions. This is the default setting.
forwarding	(Optional) Only the forwarding plane will be secured.
management	(Optional) Only the management plane will be secured.
ntp	(Optional) Specifies the configuration of the NTP feature in the AutoSecure CLI.
login	(Optional) Specifies the configuration of the Login feature in the AutoSecure CLI.
ssh	(Optional) Specifies the configuration of the SSH feature in the AutoSecure CLI.
firewall	(Optional) Specifies the configuration of the Firewall feature in the AutoSecure CLI.
tcp-intercept	(Optional) Specifies the configuration of the TCP-Intercept feature in the AutoSecure CLI.

Using the auto secure Command

1. The auto secure command is entered
2. Wizard gathers information about the outside interfaces
3. AutoSecure secures the management plane by disabling unnecessary services
4. AutoSecure prompts for a banner
5. AutoSecure prompts for passwords and enables password and login features
6. Interfaces are secured
7. Forwarding plane is secured

DHCPv4

DHCPv4 Server and Client

- Dynamic Host Configuration Protocol v4 (DHCPv4) assigns IPv4 addresses and other network configuration information dynamically. Because desktop clients typically make up the bulk of network nodes, DHCPv4 is an extremely useful and timesaving tool for network administrators.
- A dedicated DHCPv4 server is scalable and relatively easy to manage. However, in a small branch or SOHO location, a Cisco router can be configured to provide DHCPv4 services without the need for a dedicated server. Cisco IOS software supports an optional, full-featured DHCPv4 server.
- The DHCPv4 server dynamically assigns, or leases, an IPv4 address from a pool of addresses for a limited period of time chosen by the server, or until the client no longer needs the address.
- Clients lease the information from the server for an administratively defined period. Administrators configure DHCPv4 servers to set the leases to time out at different intervals. The lease is typically anywhere from 24 hours to a week or more. When the lease expires, the client must ask for another address, although the client is typically reassigned the same address.

DHCPv4 Operation

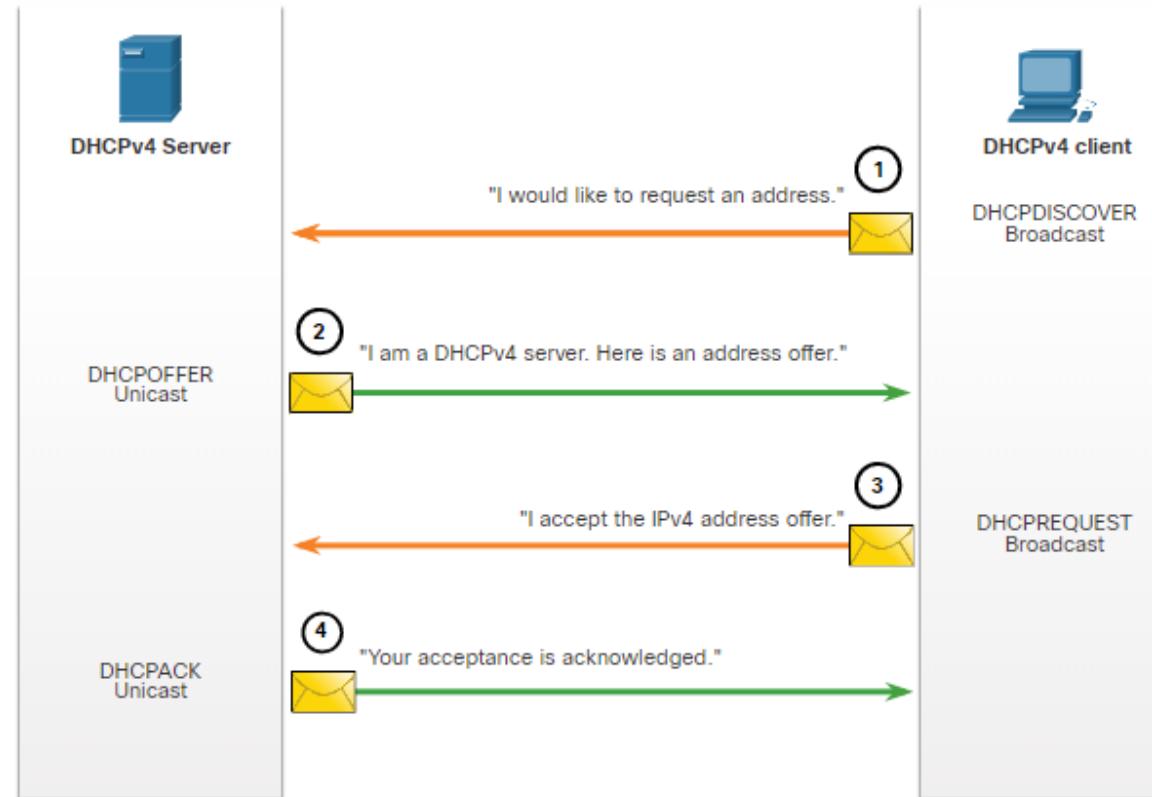
DHCPv4 works in a client/server mode. When a client communicates with a DHCPv4 server, the server assigns or leases an IPv4 address to that client.

- The client connects to the network with that leased IPv4 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.

Steps to Obtain a Lease

When the client boots (or otherwise wants to join a network), it begins a four-step process to obtain a lease:

1. DHCP Discover (DHCPDISCOVER)
2. DHCP Offer (DHCPOFFER)
3. DHCP Request (DHCPREQUEST)
4. DHCP Acknowledgment (DHCPACK)



Steps to Renew a Lease

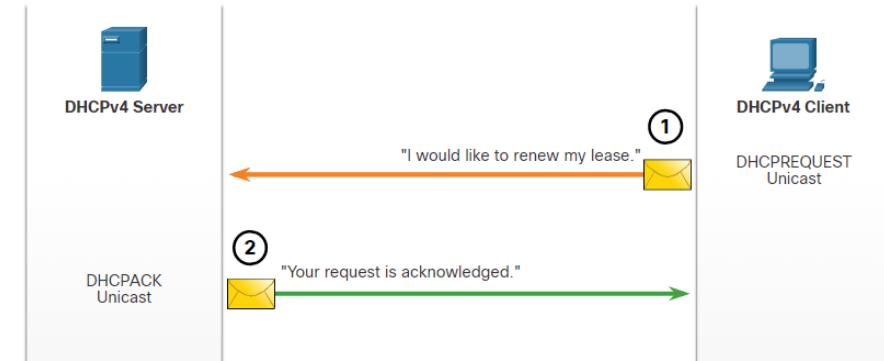
Prior to lease expiration, the client begins a two-step process to renew the lease with the DHCPv4 server, as shown in the figure:

1. DHCP Request (DHCPREQUEST)

Before the lease expires, the client sends a DHCPREQUEST message directly to the DHCPv4 server that originally offered the IPv4 address. If a DHCPACK is not received within a specified amount of time, the client broadcasts another DHCPREQUEST so that one of the other DHCPv4 servers can extend the lease.

2. DHCP Acknowledgment (DHCPACK)

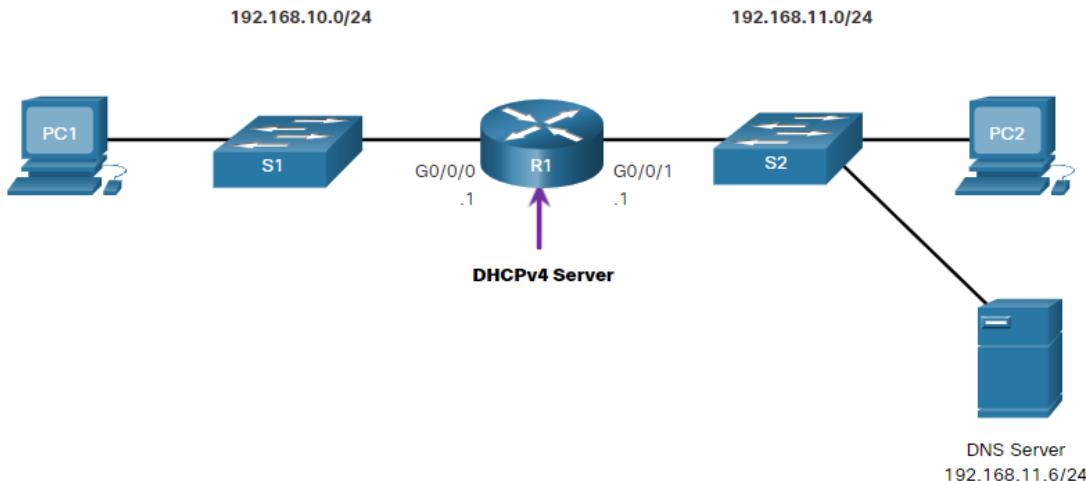
On receiving the DHCPREQUEST message, the server verifies the lease information by returning a DHCPACK.



Note: These messages (primarily the DHCPOFFER and DHCPACK) can be sent as unicast or broadcast according to IETF RFC 2131.

Cisco IOS DHCPv4 Server

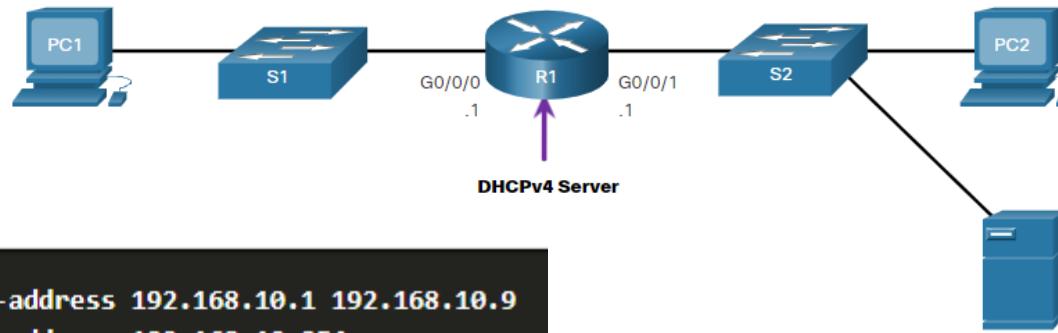
Now you have a basic understanding of how DHCPv4 works and how it can make your job a bit easier. A Cisco router running Cisco IOS software can be configured to act as a DHCPv4 server. The Cisco IOS DHCPv4 server assigns and manages IPv4 addresses from specified address pools within the router to DHCPv4 clients.



Configuration Example

192.168.10.0/24

192.168.11.0/24



```
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
R1(dhcp-config)# end
R1#
```

DHCPv4 Verification

Use the commands in the table to verify that the Cisco IOS DHCPv4 server is operational.

Command	Description
show running-config section dhcp	Displays the DHCPv4 commands configured on the router.
show ip dhcp binding	Displays a list of all IPv4 address to MAC address bindings provided by the DHCPv4 service.
show ip dhcp server statistics	Displays count information regarding the number of DHCPv4 messages that have been sent and received

Verify DHCPv4 is Operational

Verify the DHCPv4 Configuration: As shown in the example, the **show running-config | section dhcp** command output displays the DHCPv4 commands configured on R1. The **| section** parameter displays only the commands associated with DHCPv4 configuration.

```
R1# show running-config | section dhcp
ip dhcp excluded-address 192.168.10.1 192.168.10.9
ip dhcp excluded-address 192.168.10.254
ip dhcp pool LAN-POOL-1
network 192.168.10.0 255.255.255.0
default-router 192.168.10.1
dns-server 192.168.11.5
domain-name example.com
```

Verify DHCPv4 is Operational

Verify DHCPv4 Bindings: As shown in the example, the operation of DHCPv4 can be verified using the **show ip dhcp binding** command. This command displays a list of all IPv4 address to MAC address bindings that have been provided by the DHCPv4 service.

```
R1# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/          Lease expiration      Type      State       Interface
                  Hardware address/
                  User name
192.168.10.10   0100.5056.b3ed.d8    Sep 15 2019 8:42 AM  Automatic  Active
GigabitEthernet0/0/0
```

Verify DHCPv4 is Operational (Continued)

Verify DHCPv4 Statistics: The output of the **show ip dhcp server statistics** is used to verify that messages are being received or sent by the router. This command displays count information regarding the number of DHCPv4 messages that have been sent and received.

```
R1# show ip dhcp server statistics
```

Memory usage	19465
Address pools	1
Database agents	0
Automatic bindings	2
Manual bindings	0
Expired bindings	0
Malformed messages	0
Secure arp entries	0
Renew messages	0
Workspace timeouts	0
Static routes	0
Relay bindings	0
Relay bindings active	0
Relay bindings terminated	0
Relay bindings selecting	0
Message	Received
BOOTREQUEST	0
DHCPDISCOVER	4
DHCPOFFER	2
DHCPOFFER	0
DHCPOFFER	0

Verify DHCPv4 is Operational (Continued)

Verify DHCPv4 Client Received IPv4

Addressing: The **ipconfig /all** command, when issued on PC1, displays the TCP/IP parameters, as shown in the example. Because PC1 was connected to the network segment 192.168.10.0/24, it automatically received a DNS suffix, IPv4 address, subnet mask, default gateway, and DNS server address from that pool. No DHCP-specific router interface configuration is required. If a PC is connected to a network segment that has a DHCPv4 pool available, the PC can obtain an IPv4 address from the appropriate pool automatically.

```
C:\Users\Student> ipconfig /all
Windows IP Configuration
  Host Name . . . . . : ciscolab
  Primary Dns Suffix . . . . . :
  Node Type . . . . . : Hybrid
  IP Routing Enabled. . . . . : No
  WINS Proxy Enabled. . . . . : No
  Ethernet adapter Ethernet0:
    Connection-specific DNS Suffix . : example.com
    Description . . . . . : Realtek PCIe GBE Family Controller
    Physical Address. . . . . : 00-05-9A-3C-7A-00
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . . : Yes
    IPv4 Address. . . . . : 192.168.10.10
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained . . . . . : Saturday, September 14, 2019 8:42:22AM
    Lease Expires . . . . . : Sunday, September 15, 2019 8:42:22AM
    Default Gateway . . . . . : 192.168.10.1
    DHCP Server . . . . . : 192.168.10.1
    DNS Servers . . . . . : 192.168.11.5
```

Disable the Cisco IOS DHCPv4 Service

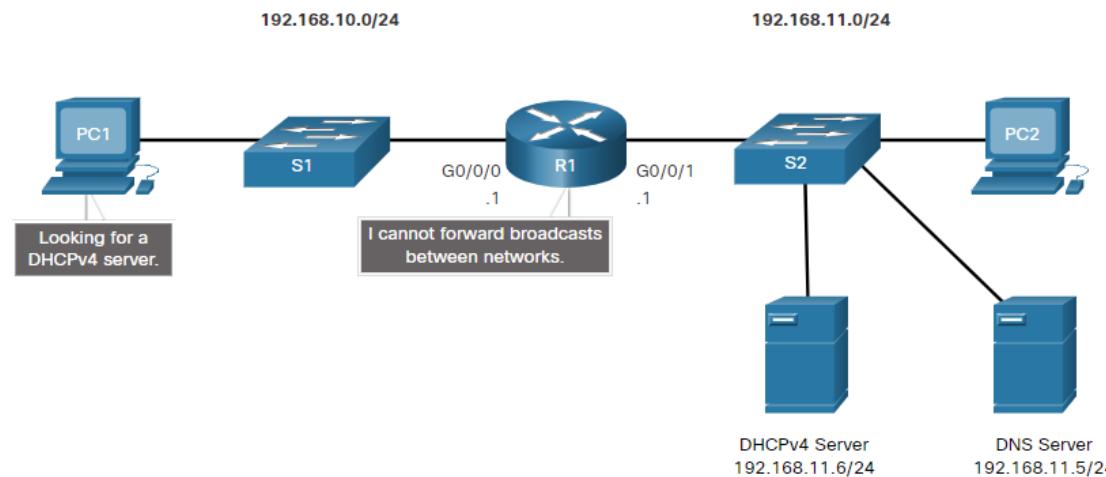
The DHCPv4 service is enabled by default. To disable the service, use the **no service dhcp** global configuration mode command. Use the **service dhcp** global configuration mode command to re-enable the DHCPv4 server process, as shown in the example. Enabling the service has no effect if the parameters are not configured.

Note: Clearing the DHCP bindings or stopping and restarting the DHCP service may result in duplicate IP addresses being temporarily assigned on the network.

```
R1(config)# no service dhcp
R1(config)# service dhcp
R1(config)#
```

DHCPv4 Relay

- In a complex hierarchical network, enterprise servers are usually located centrally. These servers may provide DHCP, DNS, TFTP, and FTP services for the network. Network clients are not typically on the same subnet as those servers. In order to locate the servers and receive services, clients often use broadcast messages.
- In the figure, PC1 is attempting to acquire an IPv4 address from a DHCPv4 server using a broadcast message. In this scenario, R1 is not configured as a DHCPv4 server and does not forward the broadcast. Because the DHCPv4 server is located on a different network, PC1 cannot receive an IP address using DHCP. R1 must be configured to relay DHCPv4 messages to the DHCPv4 server.



DHCPv4 Relay (Cont.)

- Configure R1 with the **ip helper-address address** interface configuration command. This will cause R1 to relay DHCPv4 broadcasts to the DHCPv4 server. As shown in the example, the interface on R1 receiving the broadcast from PC1 is configured to relay DHCPv4 address to the DHCPv4 server at 192.168.11.6.
- When R1 has been configured as a DHCPv4 relay agent, it accepts broadcast requests for the DHCPv4 service and then forwards those requests as a unicast to the IPv4 address 192.168.11.6. The network administrator can use the **show ip interface** command to verify the configuration.

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

```
R1# show ip interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is 192.168.11.6
  (output omitted)
```

Other Service Broadcasts Relayed

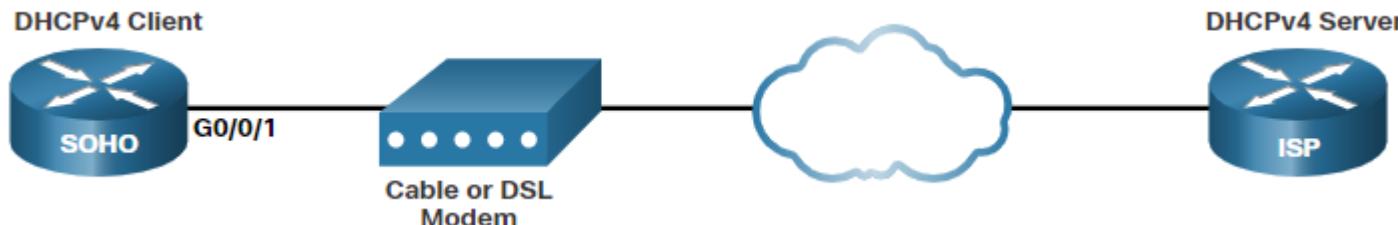
DHCPv4 is not the only service that the router can be configured to relay. By default, the **ip helper-address** command forwards the following eight UDP services:

- Port 37: Time
- Port 53: DNS
- Port 49: TACACS
- Port 67: DHCP/BOOTP server
- Port 68: DHCP/BOOTP client
- Port 69: TFTP
- Port 137: NetBIOS name service
- Port 138: NetBIOS datagram service

Cisco Router as a DHCPv4 Client

There are scenarios where you might have access to a DHCP server through your ISP. In these instances, you can configure a Cisco IOS router as a DHCPv4 client.

- Sometimes, Cisco routers in a small office or home office (SOHO) and branch sites have to be configured as DHCPv4 clients in a similar manner to client computers. The method used depends on the ISP. However, in its simplest configuration, the Ethernet interface is used to connect to a cable or DSL modem.
- To configure an Ethernet interface as a DHCP client, use the **ip address dhcp interface** configuration mode command.
- In the figure, assume that an ISP has been configured to provide select customers with IP addresses from the 209.165.201.0/27 network range after the G0/0/1 interface is configured with the **ip address dhcp**



Configuration Example

- To configure an Ethernet interface as a DHCP client, use the **ip address dhcp** interface configuration mode command, as shown in the example. This configuration assumes that the ISP has been configured to provide select customers with IPv4 addressing information.
- The **show ip interface g0/1** command confirms that the interface is up and that the address was allocated by a DHCPv4 server.

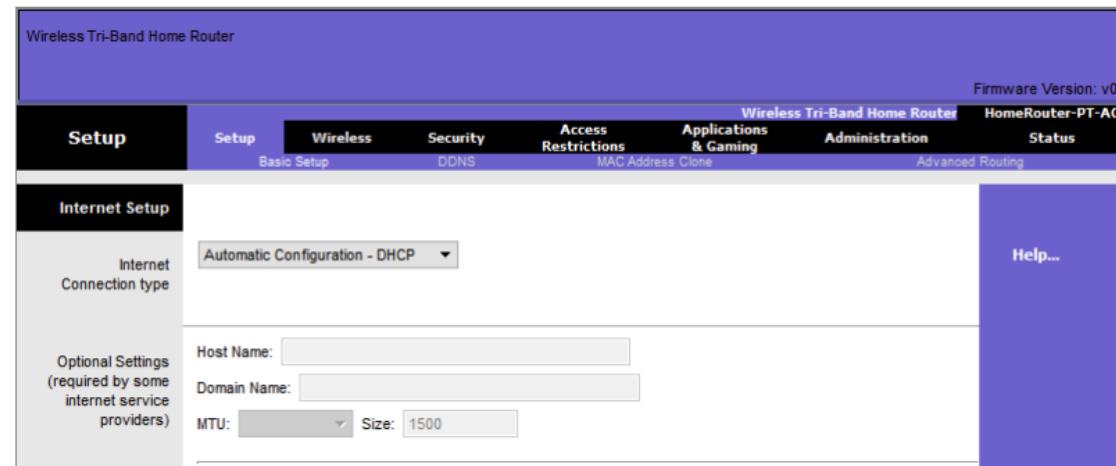
```
SOHO(config)# interface G0/0/1
SOHO(config-if)# ip address dhcp
SOHO(config-if)# no shutdown
Sep 12 10:01:25.773: %DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet0/0/1 assigned DHCP address
209.165.201.12, mask 255.255.255.224, hostname SOHO
```

```
SOHO# show ip interface g0/0/1
GigabitEthernet0/0/1 is up, line protocol is up
  Internet address is 209.165.201.12/27
  Broadcast address is 255.255.255.255
  Address determined by DHCP
  (output omitted)
```

Home Router as a DHCPv4 Client

Home routers are typically already set to receive IPv4 addressing information automatically from the ISP. This is so that customers can easily set up the router and connect to the internet.

- For example, the figure shows the default WAN setup page for a Packet Tracer wireless router. Notice that the internet connection type is set to **Automatic Configuration - DHCP**. This selection is used when the router is connected to a DSL or cable modem and acts as a DHCPv4 client, requesting an IPv4 address from the ISP.
- Various manufacturers of home routers will have a similar setup.



Switch Security Configuration

Secure Unused Ports

Layer 2 attacks are some of the easiest for hackers to deploy but these threats can also be mitigated with some common Layer 2 solutions.

- All switch ports (interfaces) should be secured before the switch is deployed for production use. How a port is secured depends on its function.
- A simple method that many administrators use to help secure the network from unauthorized access is to disable all unused ports on a switch. Navigate to each unused port and issue the Cisco IOS **shutdown** command. If a port must be reactivated at a later time, it can be enabled with the **no shutdown** command.
- To configure a range of ports, use the **interface range** command.

```
Switch(config)# interface range type module/first-number - last-number
```

Mitigate MAC Address Table Attack

The simplest and most effective method to prevent MAC address table overflow attacks is to enable port security.

- Port security limits the number of valid MAC addresses allowed on a port. It allows an administrator to manually configure MAC addresses for a port or to permit the switch to dynamically learn a limited number of MAC addresses. When a port configured with port security receives a frame, the source MAC address of the frame is compared to the list of secure source MAC addresses that were manually configured or dynamically learned on the port.
- By limiting the number of permitted MAC addresses on a port to one, port security can be used to control unauthorized access to the network.

Enable Port Security

Port security is enabled with the **switchport port-security** interface configuration command.

Notice in the example, the **switchport port-security** command was rejected. This is because port security can only be configured on manually configured access ports or manually configured trunk ports. By default, Layer 2 switch ports are set to dynamic auto (trunking on). Therefore, in the example, the port is configured with the **switchport mode access** interface configuration command.

Note: Trunk port security is beyond the scope of this course.

```
S1(config)# interface f0/1
S1(config-if)# switchport port-security
Command rejected: FastEthernet0/1 is a dynamic port.
S1(config-if)# switchport mode access
S1(config-if)# switchport port-security
S1(config-if)# end
S1#
```

Enable Port Security (Cont.)

Use the **show port-security interface** command to display the current port security settings for FastEthernet 0/1.

- Notice how port security is enabled, the violation mode is shutdown, and how the maximum number of MAC addresses is 1.
- If a device is connected to the port, the switch will automatically add the device's MAC address as a secure MAC. In this example, no device is connected to the port.

Note: If an active port is configured with the **switchport port-security** command and more than one device is connected to that port, the port will transition to the error-disabled state.

```
S1# show port-security interface f0/1
Port Security          : Enabled
Port Status             : Secure-shutdown
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 0
Configured MAC Addresses : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0
S1#
```

Enable Port Security (Cont.)

After port security is enabled, other port security specifics can be configured, as shown in the example.

```
S1(config-if)# switchport port-security ?
  aging          Port-security aging commands
  mac-address   Secure mac address
  maximum        Max secure addresses
  violation      Security violation mode
  <cr>
S1(config-if)# switchport port-security
```

Limit and Learn MAC Addresses

To set the maximum number of MAC addresses allowed on a port, use the following command:

```
Switch(config-if)# switchport port-security maximum value
```

- The default port security value is 1.
- The maximum number of secure MAC addresses that can be configured depends the switch and the IOS.
- In this example, the maximum is 8192.

```
S1(config)# interface f0/1
S1(config-if)# switchport port-security maximum ?
<1-8192> Maximum addresses
S1(config-if)# switchport port-security maximum
```

Limit and Learn MAC Addresses (Cont.)

The switch can be configured to learn about MAC addresses on a secure port in one of three ways:

1. Manually Configured: The administrator manually configures a static MAC address(es) by using the following command for each secure MAC address on the port:

```
Switch(config-if)# switchport port-security mac-address mac-address
```

2. Dynamically Learned: When the **switchport port-security** command is entered, the current source MAC for the device connected to the port is automatically secured but is not added to the running configuration. If the switch is rebooted, the port will have to re-learn the device's MAC address.

3. Dynamically Learned – Sticky: The administrator can enable the switch to dynamically learn the MAC address and “stick” them to the running configuration by using the following command:

```
Switch(config-if)# switchport port-security mac-address sticky
```

Saving the running configuration will commit the dynamically learned MAC address to NVRAM.

Limit and Learn MAC Addresses

The example demonstrates a complete port security configuration for FastEthernet 0/1.

- The administrator specifies a maximum of 4 MAC addresses, manually configures one secure MAC address, and then configures the port to dynamically learn additional secure MAC addresses up to the 4 secure MAC address maximum.
- Use the **show port-security interface** and the **show port-security address** command to verify the configuration.

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode access
S1(config-if)# switchport port-security
S1(config-if)# switchport port-security maximum 4
S1(config-if)# switchport port-security mac-address aaaa.bbbb.1234
S1(config-if)# switchport port-security mac-address sticky
S1(config-if)# end
S1# show port-security interface fa0/1
Port Security           : Enabled
Port Status              : Secure-up
Violation Mode          : Shutdown
Aging Time               : 0 mins
Aging Type               : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 4
Total MAC Addresses     : 1
Configured MAC Addresses : 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0
S1# show port-security address
                Secure Mac Address Table
-----+-----+-----+-----+-----+
Vlan  Mac Address      Type        Ports      Remaining Age
-----+-----+-----+-----+-----+
---   -----+-----+-----+-----+-----+
     1   aaaa.bbbb.1234  SecureConfigured  Fa0/1      -
-----+-----+-----+-----+-----+
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 8192
S1#
```

Port Security Aging

Port security aging can be used to set the aging time for static and dynamic secure addresses on a port and two types of aging are supported per port:

- **Absolute** - The secure addresses on the port are deleted after the specified aging time.
- **Inactivity** - The secure addresses on the port are deleted if they are inactive for a specified time.

Use aging to remove secure MAC addresses on a secure port without manually deleting the existing secure MAC addresses.

- Aging of statically configured secure addresses can be enabled or disabled on a per-port basis.

```
Switch(config-if)# switchport port-security aging {static | time time | type {absolute | inactivity}}
```

Use the **switchport port-security aging** command to enable or disable static aging for the secure port, or to set the aging time or type.

Port Security Aging (Cont.)

The example shows an administrator configuring the aging type to 10 minutes of inactivity.

The **show port-security** command confirms the changes.

```
S1(config)# interface fa0/1
S1(config-if)# switchport port-security aging time 10
S1(config-if)# switchport port-security aging type inactivity
S1(config-if)# end
S1# show port-security interface fa0/1
Port Security          : Enabled
Port Status             : Secure-shutdown
Violation Mode         : Restrict
Aging Time              : 10 mins
Aging Type              : Inactivity
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 4
Total MAC Addresses     : 1
Configured MAC Addresses : 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0050.56be.e4dd:1
Security Violation Count : 1
```

Port Security Violation Modes

If the MAC address of a device attached to a port differs from the list of secure addresses, then a port violation occurs and the port enters the error-disabled state.

- To set the port security violation mode, use the following command:

```
Switch(config-if)# switchport port-security violation {shutdown | restrict | protect}
```

The following table shows how a switch reacts based on the configured violation mode.

Mode	Description
shutdown (default)	The port transitions to the error-disabled state immediately, turns off the port LED, and sends a syslog message. It increments the violation counter. When a secure port is in the error-disabled state, an administrator must re-enable it by entering the shutdown and no shutdown commands.
restrict	The port drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value or increase the maximum value. This mode causes the Security Violation counter to increment and generates a syslog message.
protect	This is the least secure of the security violation modes. The port drops packets with unknown MAC source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value or increase the maximum value. No syslog message is sent.

Port Security Violation Modes (Cont.)

The example shows an administrator changing the security violation to “Restrict”.

The output of the **show port-security interface** command confirms that the change has been made.

```
S1(config)# interface f0/1
S1(config-if)# switchport port-security violation restrict
S1(config-if)# end
S1#
S1# show port-security interface f0/1
Port Security           : Enabled
Port Status              : Secure-shutdown
Violation Mode          : Restrict
Aging Time               : 0 mins
Aging Type               : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 4
Total MAC Addresses      : 1
Configured MAC Addresses : 1
Sticky MAC Addresses     : 0
Last Source Address:Vlan : 0050.56be.e4dd:1
Security Violation Count : 1
S1#
```

Ports in error-disabled State

When a port is shutdown and placed in the error-disabled state, no traffic is sent or received on that port.

A series of port security related messages display on the console, as shown in the following example.

Note: The port protocol and link status are changed to down and the port LED is turned off.

```
*Sep 20 06:44:54.966: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/18, putting Fa0/18 in err-disable state
*Sep 20 06:44:54.966: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 000c.292b.4c75 on port FastEthernet0/18.
*Sep 20 06:44:55.973: %LINEPROTO-5-PPDOWN: Line protocol on Interface FastEthernet0/18, changed state to down
*Sep 20 06:44:56.971: %LINK-3-UPDOWN: Interface FastEthernet0/18, changed state to down
```

Ports in error-disabled State (Cont.)

- In the example, the **show interface** command identifies the port status as **err-disabled**. The output of the **show port-security interface** command now shows the port status as **secure-shutdown**. The Security Violation counter increments by 1.
- The administrator should determine what caused the security violation. If an unauthorized device is connected to a secure port, the security threat is eliminated before re-enabling the port.
- To re-enable the port, first use the **shutdown** command, then, use the **no shutdown** command.

```
S1# show interface fa0/18
FastEthernet0/18 is down, line protocol is down (err-disabled)
(output omitted)
S1# show port-security interface fa0/18
Port Security          : Enabled
Port Status             : Secure-shutdown
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses: 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : c025.5cd7.ef01:1
Security Violation Count : 1
S1#
```

Verify Port Security

After configuring port security on a switch, check each interface to verify that the port security is set correctly, and check to ensure that the static MAC addresses have been configured correctly.

To display port security settings for the switch, use the **show port-security** command.

- The example indicates that all 24 interfaces are configured with the **switchport port-security** command because the maximum allowed is 1 and the violation mode is shutdown.
- No devices are connected, therefore, the CurrentAddr (Count) is 0 for each interface.

```
S1# show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
          (Count)      (Count)      (Count)
-----
 Fa0/1           1            0            0       Shutdown
 Fa0/2           1            0            0       Shutdown
 Fa0/3           1            0            0       Shutdown
 (output omitted)
 Fa0/24          1            0            0       Shutdown
-----
 Total Addresses in System (excluding one mac per port) : 0
 Max Addresses limit in System (excluding one mac per port) : 4096
Switch#
```

Verify Port Security (Cont.)

Use the **show port-security interface** command to view details for a specific interface, as shown previously and in this example.

```
S1# show port-security interface fastethernet 0/18
Port Security          : Enabled
Port Status             : Secure-up
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0025.83e6.4b01:1
Security Violation Count : 0
S1#
```

Verify Port Security (Cont.)

To verify that MAC addresses are “sticking” to the configuration, use the **show run** command as shown in the example for FastEthernet 0/19.

```
S1# show run | begin interface FastEthernet0/19
interface FastEthernet0/19
switchport mode access
switchport port-security maximum 10
switchport port-security
switchport port-security mac-address sticky
switchport port-security mac-address sticky 0025.83e6.4b02
(output omitted)
S1#
```

Verify Port Security (Cont.)

To display all secure MAC addresses that are manually configured or dynamically learned on all switch interfaces, use the **show port-security address** command as shown in the example.

```
S1# show port-security address
```

```
Secure Mac Address Table
```

Vlan	Mac Address	Type	Ports	Remaining Age (mins)
---	-----	-----	-----	-----
1	0025.83e6.4b01	SecureDynamic	Fa0/18	-
1	0025.83e6.4b02	SecureSticky	Fa0/19	-

```
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 8192
S1#
```

VLAN Attacks Review

A VLAN hopping attack can be launched in one of three ways:

- Spoofing DTP messages from the attacking host to cause the switch to enter trunking mode. From here, the attacker can send traffic tagged with the target VLAN, and the switch then delivers the packets to the destination.
- Introducing a rogue switch and enabling trunking. The attacker can then access all the VLANs on the victim switch from the rogue switch.
- Another type of VLAN hopping attack is a double-tagging (or double-encapsulated) attack. This attack takes advantage of the way hardware on most switches operate.

Steps to Mitigate VLAN Hopping Attack

Use the following steps to mitigate VLAN hopping attacks:

Step 1: Disable DTP (auto trunking) negotiations on non-trunking ports by using the **switchport mode access** interface configuration command.

Step 2: Disable unused ports and put them in an unused VLAN.

Step 3: Manually enable the trunk link on a trunking port by using the **switchport mode trunk** command.

Step 4: Disable DTP (auto trunking) negotiations on trunking ports by using the **switchport nonegotiate** command.

Step 5: Set the native VLAN to a VLAN other than VLAN 1 by using the **switchport trunk native**

```
S1(config)# interface range fa0/1 - 16
S1(config-if-range)# switchport mode access
S1(config-if-range)# exit
S1(config)#
S1(config)# interface range fa0/17 - 20
S1(config-if-range)# switchport mode access
S1(config-if-range)# switchport access vlan 1000
S1(config-if-range)# exit
S1(config)#
S1(config)# interface range fa0/21 - 24
S1(config-if-range)# switchport mode trunk
S1(config-if-range)# switchport nonegotiate
S1(config-if-range)# switchport trunk native vlan 999
S1(config-if-range)# end
S1#
```

DHCP Attack Review

The goal of a DHCP starvation attack is to use an attack tool such as Gobbler to create a Denial of Service (DoS) for connecting clients.

Recall that DHCP starvation attacks can be effectively mitigated by using port security because Gobbler uses a unique source MAC address for each DHCP request sent. However, mitigating DHCP spoofing attacks requires more protection.

Gobbler could be configured to use the actual interface MAC address as the source Ethernet address, but specify a different Ethernet address in the DHCP payload. This would render port security ineffective because the source MAC address would be legitimate.

DHCP spoofing attacks can be mitigated by using DHCP snooping on trusted ports.

DHCP Snooping

DHCP snooping filters DHCP messages and rate-limits DHCP traffic on untrusted ports.

- Devices under administrative control (e.g., switches, routers, and servers) are trusted sources.
- Trusted interfaces (e.g., trunk links, server ports) must be explicitly configured as trusted.
- Devices outside the network and all access ports are generally treated as untrusted sources.

A DHCP table is built that includes the source MAC address of a device on an untrusted port and the IP address assigned by the DHCP server to that device.

- The MAC address and IP address are bound together.
- Therefore, this table is called the DHCP snooping binding table.

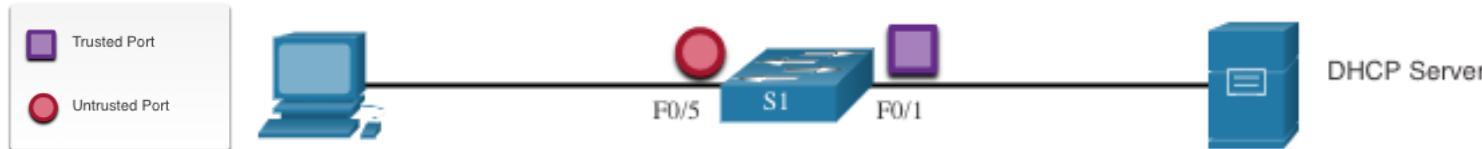
Steps to Implement DHCP Snooping

Use the following steps to enable DHCP snooping:

- Step 1.** Enable DHCP snooping by using the **ip dhcp snooping** global configuration command.
- Step 2.** On trusted ports, use the **ip dhcp snooping trust** interface configuration command.
- Step 3:** On untrusted interfaces, limit the number of DHCP discovery messages that can be received using the **ip dhcp snooping limit rate packets-per-second** interface configuration command.
- Step 4.** Enable DHCP snooping by VLAN, or by a range of VLANs, by using the **ip dhcp snooping vlan** global configuration command.

DHCP Snooping Configuration Example

Refer to the DHCP snooping sample topology with trusted and untrusted ports.



- DHCP snooping is first enabled on S1.
- The upstream interface to the DHCP server is explicitly trusted.
- F0/5 to F0/24 are untrusted and are, therefore, rate limited to six packets per second.
- Finally, DHCP snooping is enabled on VLANS 5, 10, 50, 51, and 52.

```
S1(config)# ip dhcp snooping
S1(config)# interface f0/1
S1(config-if)# ip dhcp snooping trust
S1(config-if)# exit
S1(config)# interface range f0/5 - 24
S1(config-if-range)# ip dhcp snooping limit rate 6
S1(config-if)# exit
S1(config)# ip dhcp snooping vlan 5,10,50-52
S1(config)# end
S1#
```

DHCP Snooping Configuration Example

Use the **show ip dhcp snooping** privileged EXEC command to verify DHCP snooping settings.

Use the **show ip dhcp snooping binding** command to view the clients that have received DHCP information.

Note: DHCP snooping is also required by Dynamic ARP Inspection (DAI).

```
S1# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
5,10,50-52
DHCP snooping is operational on following VLANs:
none
DHCP snooping is configured on the following L3 Interfaces:
Insertion of option 82 is enabled
  circuit-id default format: vlan-mod-port
  remote-id: 0cd9.96d2.3f80 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:
Interface      Trusted     Allow option   Rate limit (pps)
-----        -----        -----
FastEthernet0/1    yes        yes          unlimited
  Custom circuit-ids:
FastEthernet0/5     no         no            6
  Custom circuit-ids:
FastEthernet0/6     no         no            6
  Custom circuit-ids:
S1# show ip dhcp snooping binding
MacAddress      IpAddress       Lease(sec)  Type           VLAN Interface
-----        -----
00:03:47:B5:9F:AD  192.168.10.10  193185    dhcp-snooping  5    FastEthernet0/5
```

Dynamic ARP Inspection

In a typical ARP attack, a threat actor can send unsolicited ARP replies to other hosts on the subnet with the MAC Address of the threat actor and the IP address of the default gateway. To prevent ARP spoofing and the resulting ARP poisoning, a switch must ensure that only valid ARP Requests and Replies are relayed.

Dynamic ARP inspection (DAI) requires DHCP snooping and helps prevent ARP attacks by:

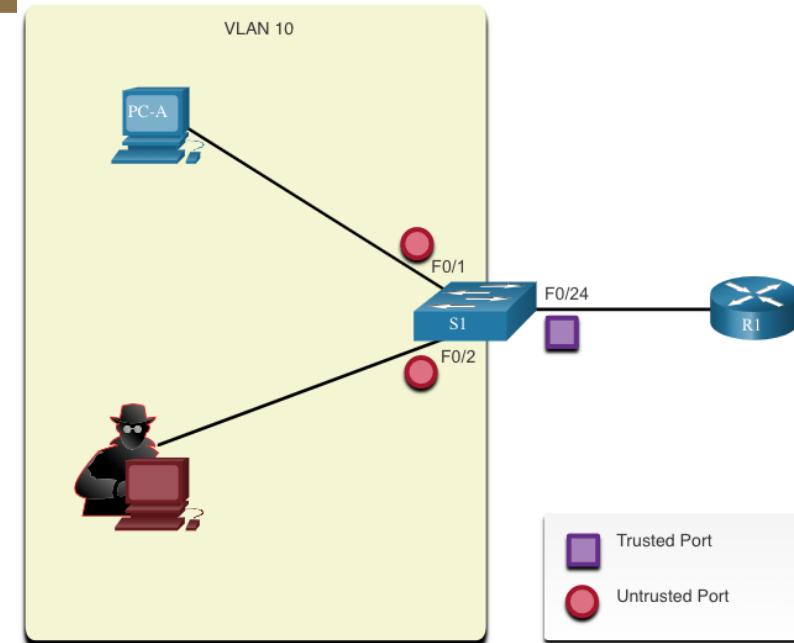
- Not relaying invalid or gratuitous ARP Replies out to other ports in the same VLAN.
- Intercepting all ARP Requests and Replies on untrusted ports.
- Verifying each intercepted packet for a valid IP-to-MAC binding.
- Dropping and logging ARP Replies coming from invalid to prevent ARP poisoning.
- Error-disabling the interface if the configured DAI number of ARP packets is exceeded.

DAI Implementation Guidelines

To mitigate the chances of ARP spoofing and ARP poisoning, follow these DAI implementation guidelines:

- Enable DHCP snooping globally.
- Enable DHCP snooping on selected VLANs.
- Enable DAI on selected VLANs.
- Configure trusted interfaces for DHCP snooping and ARP inspection.

It is generally advisable to configure all access switch ports as untrusted and to configure all uplink ports that are connected to other switches as trusted.



DAI Configuration Example

In the previous topology, S1 is connecting two users or

- DAI will be configured to mitigate against ARP spoofing.
- DHCP snooping is enabled because DAI requires the DHCP snooping binding table to operate.
- Next, DHCP snooping and ARP inspection are enabled for the PCs on VLAN10.
- The uplink port to the router is trusted, and therefore, is configured as trusted for DHCP snooping and ARP inspection.

```
S1(config)# ip dhcp snooping
S1(config)# ip dhcp snooping vlan 10
S1(config)# ip arp inspection vlan 10
S1(config)# interface fa0/24
S1(config-if)# ip dhcp snooping trust
S1(config-if)# ip arp inspection trust
```

DAI Configuration Example (Cont.)

DAI can also be configured to check for both destination or source MAC and IP addresses:

- **Destination MAC** - Checks the destination MAC address in the Ethernet header against the target MAC address in ARP body.
- **Source MAC** - Checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body.
- **IP address** - Checks the ARP body for invalid and unexpected IP addresses including addresses 0.0.0.0, 255.255.255.255, and all IP multicast addresses.

DAI Configuration Example (Cont.)

The **ip arp inspection validate {[src-mac] [dst-mac] [ip]}** global configuration command is used to configure DAI to drop ARP packets when the IP addresses are invalid.

- It can be used when the MAC addresses in the body of the ARP packets do not match the addresses that are specified in the Ethernet header.
- Notice in the following example how only one configuration command is present.
 - Therefore, entering multiple **ip arp inspection validate** commands overwrites the previous command.
 - To include more than one validation method, enter them on the same command line as shown in the output.

```
S1(config)# ip arp inspection validate ?
      dst-mac  Validate destination MAC address
      ip       Validate IP addresses
      src-mac  Validate source MAC address
S1(config)# ip arp inspection validate src-mac
S1(config)# ip arp inspection validate dst-mac
S1(config)# ip arp inspection validate ip
S1(config)# do show run | include validate
ip arp inspection validate ip
S1(config)# ip arp inspection validate src-mac dst-mac ip
S1(config)# do show run | include validate
ip arp inspection validate src-mac dst-mac ip
S1(config)#

```

PortFast and BPDU Guard

Recall that network attackers can manipulate the Spanning Tree Protocol (STP) to conduct an attack by spoofing the root bridge and changing the topology of a network.

To mitigate STP attacks, use PortFast and Bridge Protocol Data Unit (BPDU) Guard:

PortFast

- PortFast immediately brings a port to the forwarding state from a blocking state, bypassing the listening and learning states.
- Apply to all end-user access ports.

BPDU Guard

- BPDU guard immediately errors disables a port that receives a BPDU.
- Like PortFast, BPDU guard should only be configured on interfaces attached to end devices.

Configure PortFast

PortFast bypasses the STP listening and learning states to minimize the time that access ports must wait for STP to converge.

- Only enable PortFast on access ports.
- PortFast on inter switch links can create a spanning-tree loop.

PortFast can be enabled:

- **On an interface** – Use the **spanning-tree portfast** interface configuration command.
- **Globally** – Use the **spanning-tree portfast default** global configuration command to enable PortFast on all access ports.

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode access
S1(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION
%Portfast has been configured on FastEthernet0/1 but will only
have effect when the interface is in a non-trunking mode.
S1(config-if)# exit
S1(config)# spanning-tree portfast default
%Warning: this command enables portfast by default on all interfaces. You
should now disable portfast explicitly on switched ports leading to hubs,
switches and bridges as they may create temporary bridging loops.
S1(config)# exit
```

Configure PortFast (Cont.)

To verify whether PortFast is enabled globally you can use either the:

- **show running-config | begin span** command
- **show spanning-tree summary** command

To verify if PortFast is enabled an interface, use the **show running-config interface type/number** command.

The **show spanning-tree interface type/number detail** command can also be used for verification.

Configure BPDU Guard

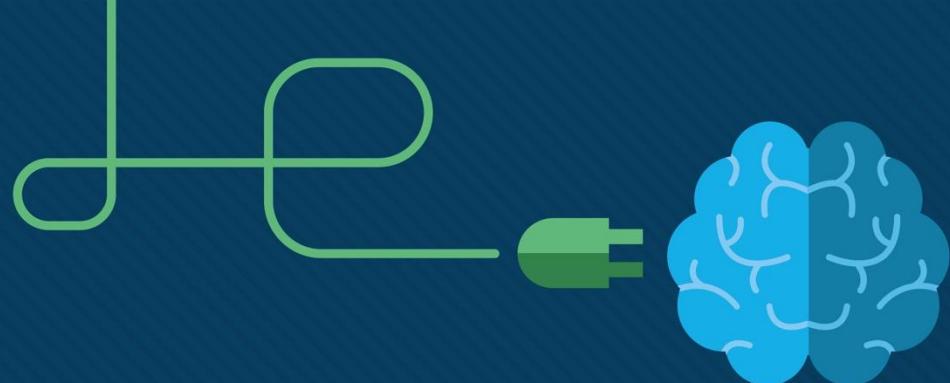
An access port could receive an unexpected BPDUs accidentally or because a user connected an unauthorized switch to the access port.

- If a BPDU is received on a BPDU Guard enabled access port, the port is put into error-disabled state.
- This means the port is shut down and must be manually re-enabled or automatically recovered through the **errdisable recovery cause psecureViolation** global command.

BPDU Guard can be enabled:

- **On an interface** – Use the **spanning-tree bpduguard enable** interface configuration command.
- **Globally** – Use the **spanning-tree portfast bpduguard default** global configuration command to enable BPDU Guard on all access ports.

```
S1(config)# interface fa0/1
S1(config-if)# spanning-tree bpduguard enable
S1(config-if)# exit
S1(config)# spanning-tree portfast bpduguard default
S1(config)# end
S1# show spanning-tree summary
Switch is in pvst mode
Root bridge for: none
Extended system ID      is enabled
Portfast Default        is enabled
PortFast BPDU Guard Default  is enabled
Portfast BPDU Filter Default is disabled
Loopguard Default       is disabled
EtherChannel misconfig guard is enabled
UplinkFast              is disabled
BackboneFast             is disabled
Configured Pathcost method used is short
(output omitted)
S1#
```



Module 6: NAT for IPv4

Instructor Materials

Enterprise Networking, Security,
and Automation v7.0 (ENSA)

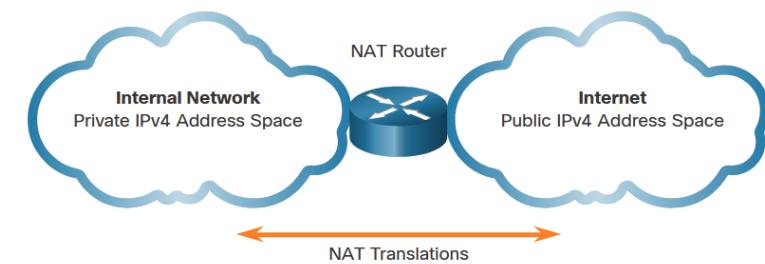


NAT for IPv4

IPv4 Address Space

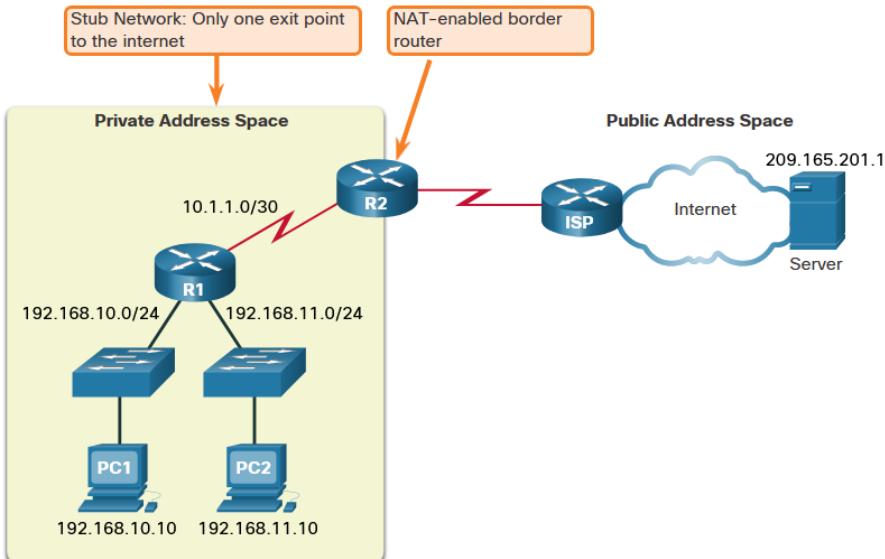
- Networks are commonly implemented using private IPv4 addresses, as defined in RFC 1918.
- Private IPv4 addresses cannot be routed over the internet and are used within an organization or site to allow devices to communicate locally.
- To allow a device with a private IPv4 address to access devices and resources outside of the local network, the private address must first be translated to a public address.
- NAT provides the translation of private addresses to public addresses.

Class	Activity Type	Activity Name
A	10.0.0.0 – 10.255.255.255	10.0.0.0/8
B	172.16.0.0 – 172.31.255.255	172.16.0.0/12
C	192.168.0.0 – 192.168.255.255	192.168.0.0/16



What is NAT

- The primary use of NAT is to conserve public IPv4 addresses.
- NAT allows networks to use private IPv4 addresses internally and translates them to a public address when needed.
- A NAT router typically operates at the border of a stub network.
- When a device inside the stub network wants to communicate with a device outside of its network, the packet is forwarded to the border router which performs the NAT process, translating the internal private address of the device to a public, outside, routable address.

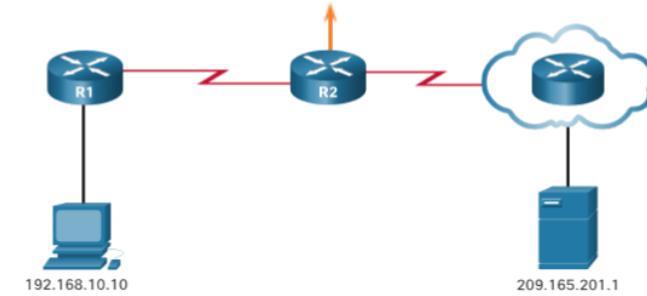


How NAT Works

PC1 wants to communicate with an outside web server with public address 209.165.201.1.

1. PC1 sends a packet addressed to the web server.
2. R2 receives the packet and reads the source IPv4 address to determine if it needs translation.
3. R2 adds mapping of the local to global address to the NAT table.
4. R2 sends the packet with the translated source address toward the destination.
5. The web server responds with a packet addressed to the inside global address of PC1 (209.165.200.226).
6. 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.

NAT Table			
Inside Local	Inside Global	Outside Local	Outside Global
192.168.10.10	209.165.200.226	209.165.201.1	209.165.201.1



NAT Terminology

NAT includes four types of addresses:

- Inside local address
- Inside global address
- Outside local address
- Outside global address

NAT terminology is always applied from the perspective of the device with the translated address:

- **Inside address** - The address of the device which is being translated by NAT.
- **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.

NAT Terminology (Cont.)

Inside local address

The address of the source as seen from inside the network. This is typically a private IPv4 address. The inside local address of PC1 is 192.168.10.10.

Inside global addresses

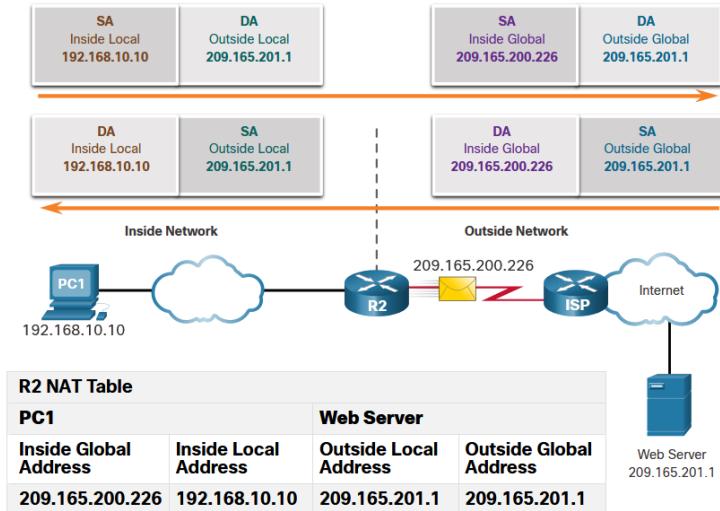
The address of source as seen from the outside network. The inside global address of PC1 is 209.165.200.226

Outside global address

The address of the destination as seen from the outside network. The outside global address of the web server is 209.165.201.1

Outside local address

The address of the destination as seen from the inside network. PC1 sends traffic to the web server at the IPv4 address 209.165.201.1. While uncommon, this address could be different than the globally routable address of the destination.



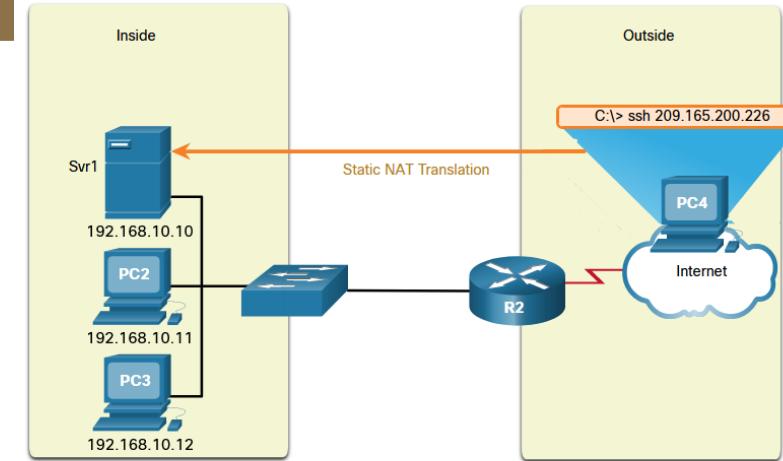
Types of NAT

Static NAT



Static NAT uses a one-to-one mapping of local and global addresses configured by the network administrator that remain constant.

- Static NAT is useful for web servers or devices that must have a consistent address that is accessible from the internet, such as a company web server.
- It is also useful for devices that must be accessible by authorized personnel when offsite, but not by the general public on the internet.



Static NAT Table

Inside Local Address	Inside Global Address - Addresses reachable via R2
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228

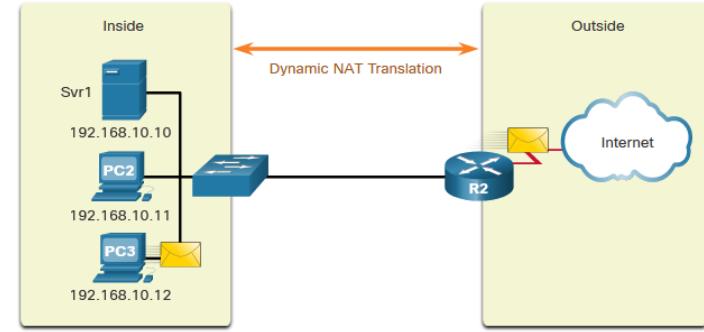
Note: Static NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

Dynamic NAT

Dynamic NAT uses a pool of public addresses and assigns them on a first-come, first-served basis.

- When an inside device requests access to an outside network, dynamic NAT assigns an available public IPv4 address from the pool.
- The other addresses in the pool are still available for use.

Note: Dynamic NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

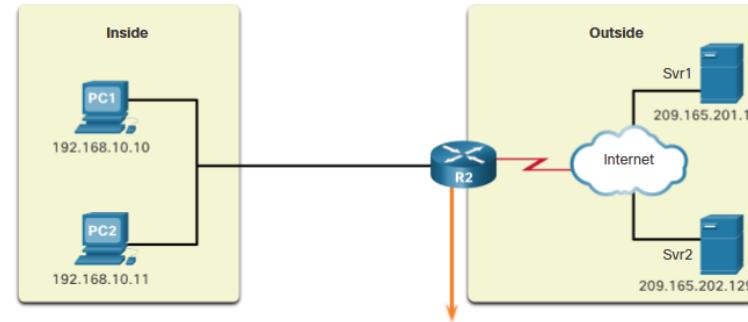


IPv4 NAT Pool	
Inside Local Address	Inside Global Address Pool - Addresses reachable via R2
192.168.10.12	209.165.200.226
Available	209.165.200.227
Available	209.165.200.228
Available	209.165.200.229
Available	209.165.200.230

Port Address Translation

Port Address Translation (PAT), also known as NAT overload, maps multiple private IPv4 addresses to a single public IPv4 address or a few addresses.

- With PAT, when the NAT router receives a packet from the client, it uses the 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.



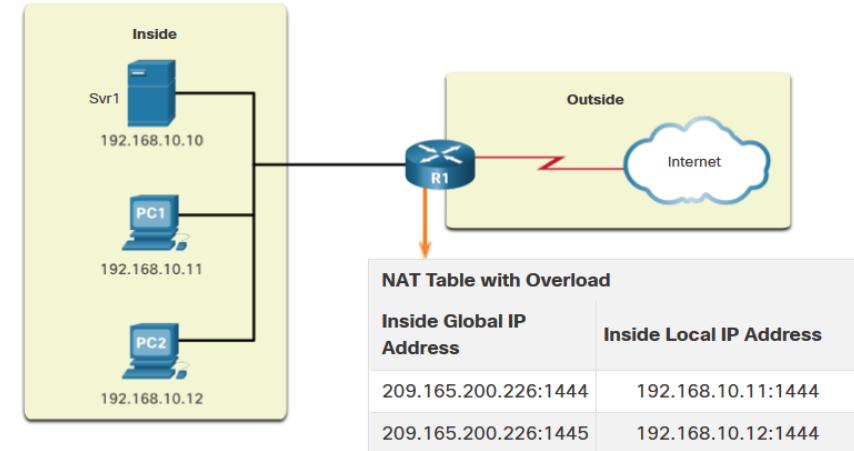
NAT Table with Overload

Inside Local IP Address	Inside Global IP Address	Outside Local IP Address	Outside Global IP Address
192.168.10.10:1555	209.165.200.226:1555	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1331	209.165.200.226:1331	209.165.202.129:80	209.165.202.129:80

Next Available Port

PAT attempts to preserve the original source port. If the original source port is already used, PAT assigns the first available port number starting from the beginning of the appropriate port group 0-511, 512-1,023, or 1,024-65,535.

- 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.
- The process continues until there are no more available ports or external IPv4 addresses in the address pool.



NAT and PAT Comparison

Summary of the differences between NAT and PAT.

NAT - Only modifies the IPv4 addresses

Inside Global Address	Inside Local Address
209.165.200.226	192.168.10.10

PAT - PAT modifies both the IPv4 address and the port number.

Inside Global Address	Inside Local Address
209.165.200.226:2031	192.168.10.10:2031

NAT	PAT
One-to-one mapping between Inside Local and Inside Global addresses.	One Inside Global address can be mapped to many Inside Local addresses.
Uses only IPv4 addresses in translation process.	Uses IPv4 addresses and TCP or UDP source port numbers in translation process.
A unique Inside Global address is required for each inside host accessing the outside network.	A single unique Inside Global address can be shared by many inside hosts accessing the outside network.

Packets without a Layer 4 Segment

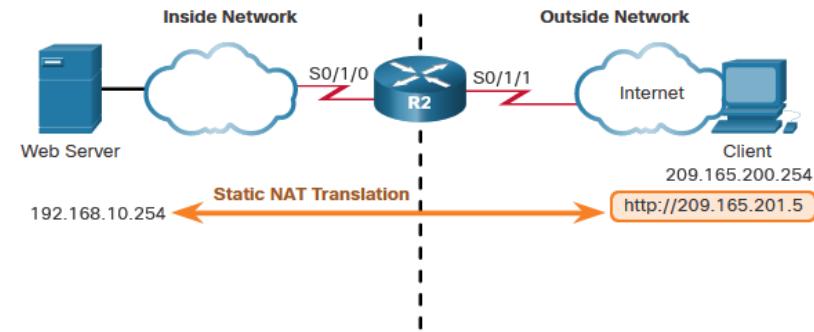
Some packets do not contain a Layer 4 port number, such as ICMPv4 messages. Each of these types of protocols is handled differently by PAT.

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.

Note: Other ICMPv4 messages do not use the Query ID. These messages and other protocols that do not use TCP or UDP port numbers vary and are beyond the scope of this curriculum.

Static NAT Scenario

- Static NAT is a one-to-one mapping between an inside address and an outside address.
- Static NAT allows external devices to initiate connections to internal devices using the statically assigned public address.
- For instance, an internal web server may be mapped to a specific inside global address so that it is accessible from outside networks.



Configure Static NAT

There are two basic tasks when configuring static NAT translations:

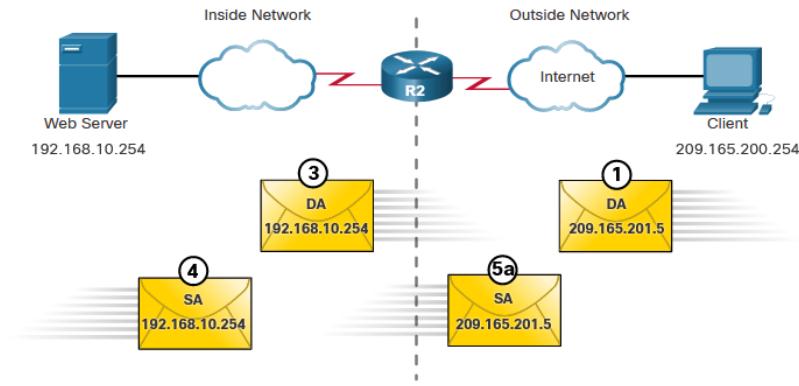
- **Step 1** - Create a mapping between the inside local address and the inside global addresses using the **ip nat inside source static** command.
- **Step 2** - The interfaces participating in the translation are configured as inside or outside relative to NAT with the **ip nat inside** and **ip nat outside** commands.

```
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5
R2(config)#
R2(config)# interface serial 0/1/0
R2(config-if)# ip address 192.168.1.2 255.255.255.252
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface serial 0/1/1
R2(config-if)# ip address 209.165.200.1 255.255.255.252
R2(config-if)# ip nat outside
```

Analyze Static NAT

The static NAT translation process between the client and the web server:

1. The client sends a packet to the web server.
2. R2 receives packets from the client on its NAT outside interface and checks its NAT table.
3. R2 translates the inside global address of to the inside local address and forwards the packet towards the web server.
4. The web server receives the packet and responds to the client using its inside local address.
5. (a) R2 receives the packet from the web server on its NAT inside interface with source address of the inside local address of the web server and (b) translates the source address to the inside global address.



Inside Local Address	Inside Global Address	Outside Local Address	Outside Global Address
192.168.10.254	209.165.201.5	209.165.200.254	209.165.200.254

Verify Static NAT

To verify NAT operation, issue the **show ip nat translations** command.

- This command shows active NAT translations.
- Because the example is a static NAT configuration, the translation is always present in the NAT table regardless of any active communications.
- If the command is issued during an active session, the output also indicates the address of the

```
R2# show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 209.165.201.5      192.168.10.254    ---                  ---
Total number of translations: 1
```

```
R2# show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
tcp 209.165.201.5      192.168.10.254    209.165.200.254   209.165.200.254
--- 209.165.201.5      192.168.10.254    ---                  ---
Total number of translations: 2
```

Verify Static NAT (Cont.)

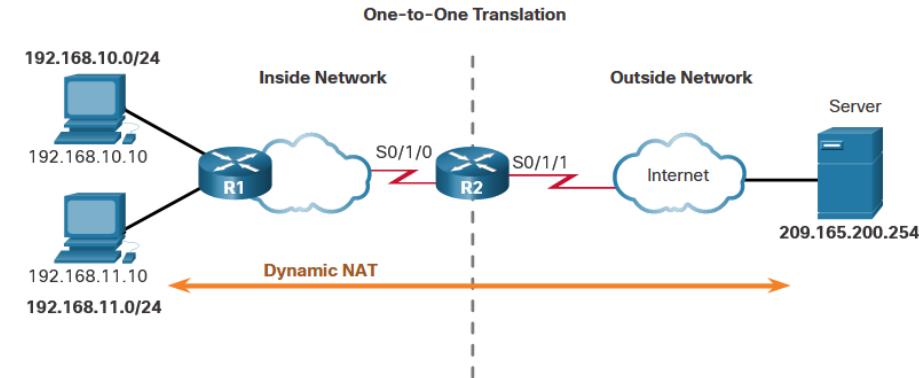
Another useful command is **show ip nat statistics**.

- It displays information about the total number of active translations, NAT configuration parameters, the number of addresses in the pool, and the number of addresses that have been allocated.
- To verify that the NAT translation is working, it is best to clear statistics from any past translations using the **clear ip nat statistics** command before testing.

```
R2# show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Outside interfaces:
    Serial0/1/1
Inside interfaces:
    Serial0/1/0
Hits: 4  Misses: 1
(output omitted)
```

Dynamic NAT Scenario

- Dynamic NAT automatically maps inside local addresses to inside global addresses.
- Dynamic NAT uses a pool of inside global addresses.
- The pool of inside global addresses is available to any device on the inside network on a first-come first-served basis.
- If all addresses in the pool are in use, a device must wait for an available address before it can access the outside network.



Configure Dynamic NAT

There are five tasks when configuring dynamic NAT translations:

- **Step 1** - Define the pool of addresses that will be used for translation using the **ip nat pool** command.
- **Step 2** - Configure a standard ACL to identify (permit) only those addresses that are to be translated.
- **Step 3** - Bind the ACL to the pool, using the **ip nat inside source list** command.

```
R2(config)# ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL1
```

Configure Dynamic NAT (Cont.)

There are five tasks when configuring dynamic NAT translations:

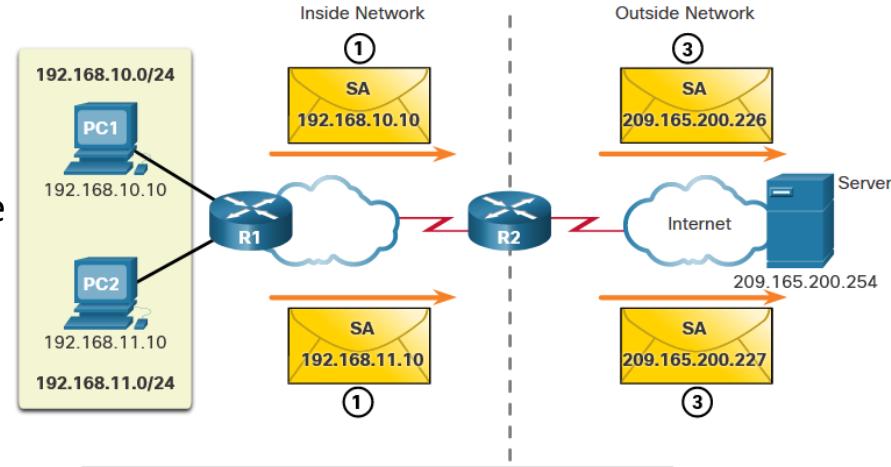
- **Step 4** - Identify which interfaces are inside.
- **Step 5** - Identify which interfaces are outside.

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

Analyze Dynamic NAT – Inside to Outs

Dynamic NAT translation process:

1. PC1 and PC2 send packets requesting a connection to the server.
2. R2 receives the first packet from PC1, checks the ALC to determine if the packet should be translated, selects an available global address, and creates a translation entry in the NAT table.
3. R2 replaces the inside local source address of PC1, 192.168.10.10, with the translated inside global address of 209.165.200.226 and forwards the packet. (The same process occurs for the packet from PC2 using the translated address of 209.165.200.227.)

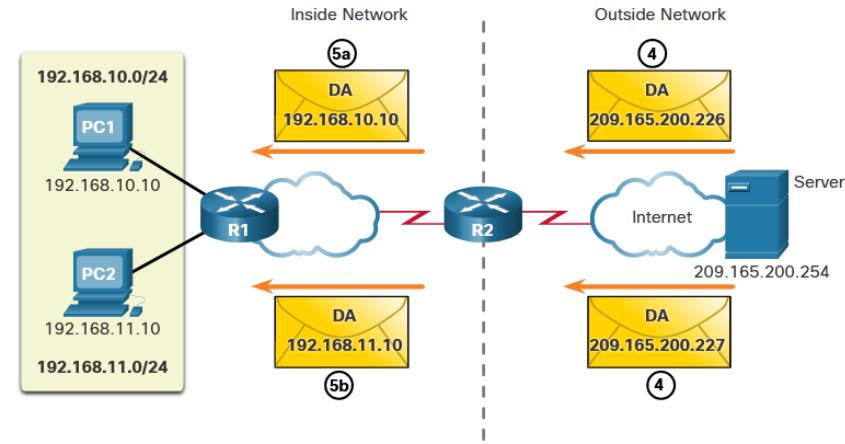


IPv4 NAT Pool	
Inside Local Address Pool	Inside Global Address
② 192.168.10.10	209.165.200.226
② 192.168.11.10	209.165.200.227

Analyze Dynamic NAT – Outside to Inside

Dynamic NAT translation process:

4. The server receives the packet from PC1 and responds using the destination address of 209.165.200.226. The server receives the packet from PC2, it responds to using the destination address of 209.165.200.227.
5. (a) When R2 receives the packet with the destination address of 209.165.200.226; it performs a NAT table lookup and translates the address back to the inside local address and forwards the packet toward PC1.
 (b) When R2 receives the packet with the destination address of 209.165.200.227; it performs a NAT table lookup and translates the address back to the inside local address 192.168.11.10 and forwards the packet toward PC2.



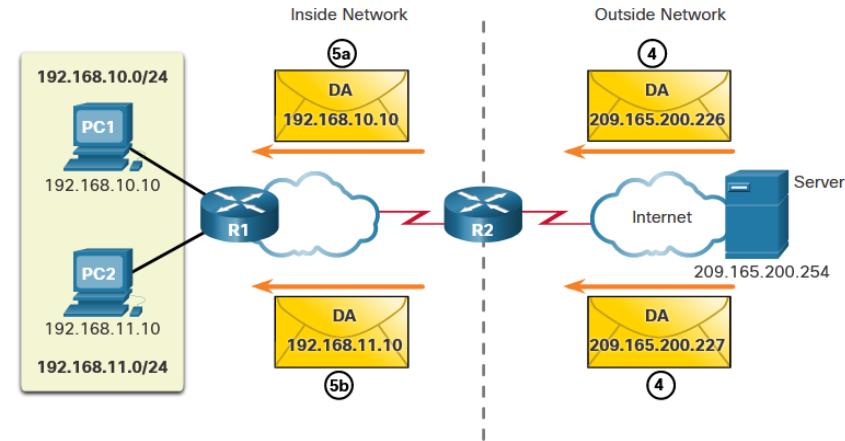
IPv4 NAT Pool	
Inside Local Address Pool	Inside Global Address
⑤a 192.168.10.10	209.165.200.226
⑤b 192.168.11.10	209.165.200.227

Analyze Dynamic NAT

– Outside to Inside (Cont.)

Dynamic NAT translation process:

6. PC1 and PC2 receive the packets and continue the conversation. The router performs Steps 2 to 5 for each packet.



IPv4 NAT Pool	
Inside Local Address Pool	Inside Global Address
⑤a 192.168.10.10	209.165.200.226
⑤b 192.168.11.10	209.165.200.227

Verify Dynamic NAT

The output of the **show ip nat translations** command displays all static translations that have been configured and any dynamic translations that have been created by traffic.

```
R2# show ip nat translations
Pro Inside global      Inside local        Outside local        Outside global
--- 209.165.200.228    192.168.10.10      ---                 ---
--- 209.165.200.229    192.168.11.10      ---                 ---
```

R2#

Verify Dynamic NAT (Cont.)

Adding the **verbose** keyword displays additional information about each translation, including how long ago the entry was created and used.

```
R2# show ip nat translation verbose
Pro Inside global      Inside local      Outside local      Outside global
tcp 209.165.200.228    192.168.10.10    ---              ---
create 00:02:11, use 00:02:11 timeout:86400000, left 23:57:48, Map-Id(In): 1,
flags:
none, use_count: 0, entry-id: 10, lc_entries: 0
tcp 209.165.200.229    192.168.11.10    ---              ---
create 00:02:10, use 00:02:10 timeout:86400000, left 23:57:49, Map-Id(In): 1,
flags:
none, use_count: 0, entry-id: 12, lc_entries: 0
R2#
```

Verify Dynamic NAT (Cont.)

By default, translation entries time out after 24 hours, unless the timers have been reconfigured with the **ip nat translation timeout *timeout-seconds*** command in global configuration mode. To clear dynamic entries before the timeout has expired, use the **clear ip nat translation** privileged EXEC mode command.

```
R2# clear ip nat translation *
R2# show ip nat translation
```

Command	Description
clear ip nat translation *	Clears all dynamic address translation entries from the NAT translation table.
clear ip nat translation inside <i>global-ip local-ip</i> [outside <i>local-ip global-ip</i>]	Clears a simple dynamic translation entry containing an inside translation or both inside and outside translation.
clear ip nat translation protocol inside <i>global-ip global-port local-ip local-port</i> [outside <i>local-ip local-port global-ip global-port</i>]	Clears an extended dynamic translation entry.

Verify Dynamic NAT (Cont.)

The **show ip nat statistics** command displays information about the total number of active translations, NAT configuration parameters, the number of addresses in the pool, and how many of the addresses have been allocated.

```
R2# show ip nat statistics
Total active translations: 4 (0 static, 4 dynamic; 0 extended)
Peak translations: 4, occurred 00:31:43 ago
Outside interfaces:
  Serial0/1/1
Inside interfaces:
  Serial0/1/0
Hits: 47 Misses: 0
CEF Translated packets: 47, CEF Punted packets: 0
Expired translations: 5
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool NAT-POOL1 refcount 4
  pool NAT-POOL1: netmask 255.255.255.224
    start 209.165.200.226 end 209.165.200.240
    type generic, total addresses 15, allocated 2 (13%), misses 0
(output omitted)
R2#
```

Verify Dynamic NAT (Cont.)

The **show running-config** command and show s the NAT, ACL, interface, or pool commands with the required values.

```
R2# show running-config | include NAT
ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224
ip nat inside source list 1 pool NAT-POOL1
```

Configure PAT to Use a Single IPv4 Address

To configure PAT to use a single IPv4 address, add the keyword **overload** to the **ip nat inside source** command.

In the example, all hosts from network 192.168.0.0/16 (matching ACL 1) that send traffic through router R2 to the internet will be translated to IPv4 address 209.165.200.225 (IPv4 address of interface S0/1/1). The traffic flows will be identified by port numbers in the NAT table because the **overload** keyword is configured.

```
R2(config)# ip nat inside source list 1 interface serial 0/1/0 overload
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# interface serial0/1/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface Serial0/1/1
R2(config-if)# ip nat outside
```

Configure PAT to Use an Address Pool

An ISP may allocate more than one public IPv4 address to an organization. In this scenario the organization can configure PAT to use a pool of IPv4 public addresses for translation.

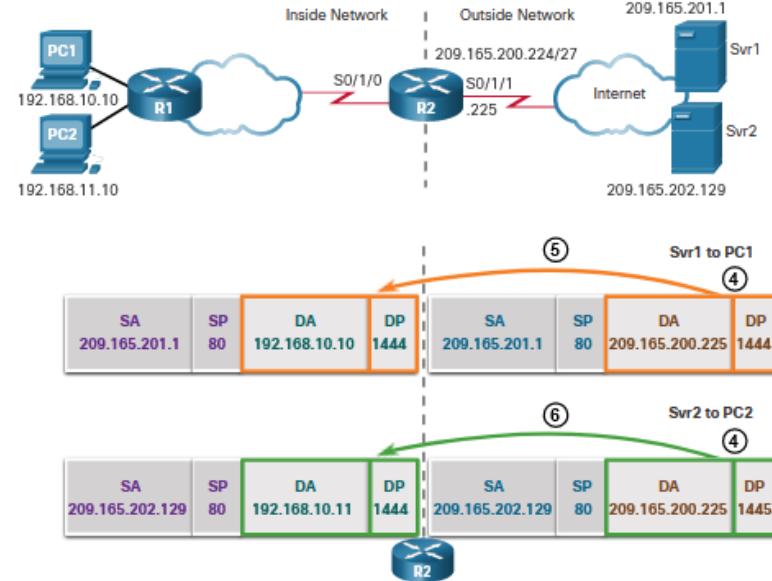
To configure PAT for a dynamic NAT address pool, simply add the keyword **overload** to the **ip nat inside source** command.

In the example, NAT-POOL2 is bound to an ACL to permit 192.168.0.0/16 to be translated. These hosts can share an IPv4 address from the pool because PAT is enabled with the keyword **overload**.

```
R2(config)# ip nat pool NAT-POOL2 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL2 overload
R2(config)# interface serial0/1/0
R2(config-if)# ip nat inside
R2(config-if)# interface serial0/1/0
R2(config-if)# ip nat outside
```

Analyze PAT – Server to PC

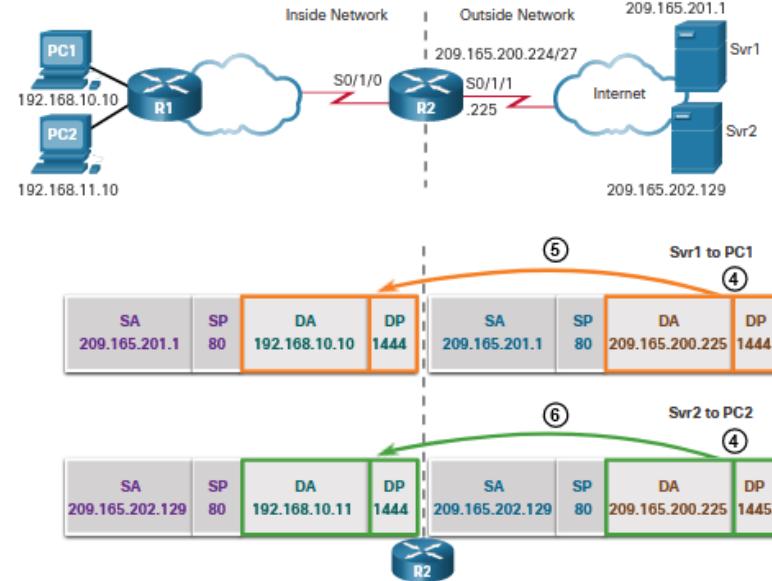
1. PC1 and PC2 send packets to Svr1 and Svr2.
2. The packet from PC1 reaches R2 first. R2 modifies the source IPv4 address to 209.165.200.225 (inside global address). The packet is then forwarded towards Svr1.
3. The packet from PC2 arrives at R2. PAT changes the source IPv4 address of PC2 to the inside global address 209.165.200.225. PC2 has the same source port number as the translation for PC1. PAT increments the source port number until it is a unique value in its table. In this instance, 1445.



NAT Table			
Inside Local Address	Inside Global Address	Outside Global Address	Outside Local Address
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.201.129:80	209.165.202.129:80

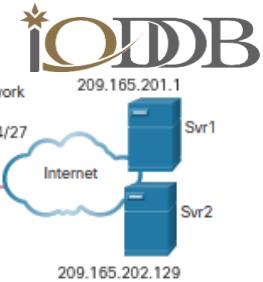
Analyze PAT – PC to Server

1. PC1 and PC2 send packets to Svr1 and Svr2.
2. The packet from PC1 reaches R2 first. R2 modifies the source IPv4 address to 209.165.200.225 (inside global address). The packet is then forwarded towards Svr1.
3. The packet from PC2 arrives at R2. PAT changes the source IPv4 address of PC2 to the inside global address 209.165.200.225. PC2 has the same source port number as the translation for PC1. PAT increments the source port number until it is a unique value in its table. In this instance, it is 1445.

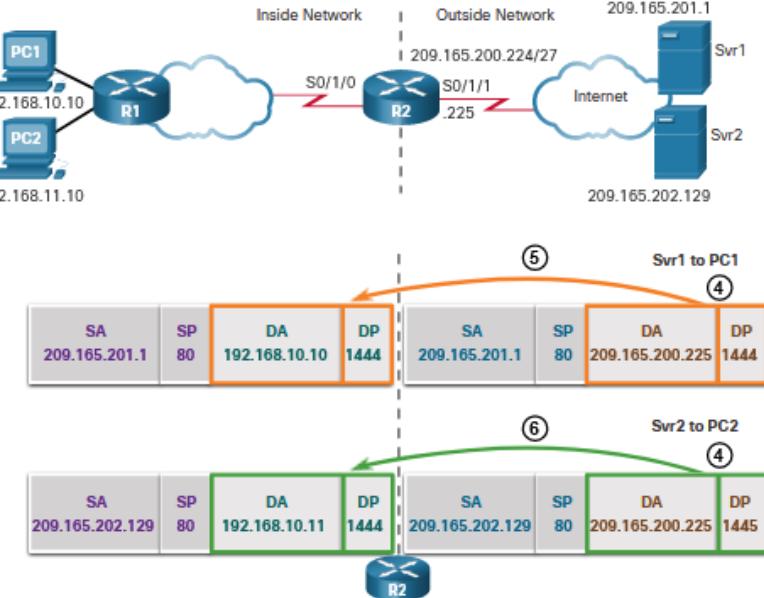


NAT Table			
Inside Local Address	Inside Global Address	Outside Global Address	Outside Local Address
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.201.129:80	209.165.202.129:80

Analyze PAT – Server to PC



1. The servers use the source port from the received packet as the destination port, and the source address as the destination address for the return traffic.
2. R2 changes the destination IPv4 address of the packet from Srv1 from 209.165.200.225 to 192.168.10.10, and forwards the packet toward PC1.
3. R2 changes the destination address of packet from Srv2. from 209.165.200.225 to 192.168.10.11. and modifies the destinations port back to its original value of 1444. The packet is then forwarded toward PC2.



NAT Table			
Inside Local Address	Inside Global Address	Outside Global Address	Outside Local Address
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.201.129:80	209.165.202.129:80

Verify PAT

The same commands used to verify static and dynamic NAT are used to verify PAT. The **show ip nat translations** command displays the translations from two different hosts to different web servers. Notice that two different inside hosts are allocated the same IPv4 address of 209.165.200.226 (inside global address). The source port numbers in the NAT table differentiate the two transactions.

```
R2# show ip nat translations
Pro Inside global           Inside local        Outside local      Outside global
tcp 209.165.200.225:1444  192.168.10.10:1444  209.165.201.1:80  209.165.201.1:80
tcp 209.165.200.225:1445  192.168.11.10:1444  209.165.202.129:80 209.165.202.129:80
R2#
```

Verify PAT (Cont.)

The **show ip nat statistics** command verifies that NAT-POOL2 has allocated a single address for both translations. Also shown are the number and type of active translations, NAT configuration parameters, the number of addresses in the pool, and how many have been allocated.

```
R2# show ip nat statistics
Total active translations: 4 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:31:43 ago
Outside interfaces:
    Serial0/1/1
Inside interfaces:
    Serial0/1/0
Hits: 4 Misses: 0
CEF Translated packets: 47, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 3] access-list 1 pool NAT-POOL2 refcount 2
    pool NAT-POOL2: netmask 255.255.255.224
        start 209.165.200.225 end 209.165.200.240
        type generic, total addresses 15, allocated 1 (6%), misses 0
(output omitted)
R2#
```