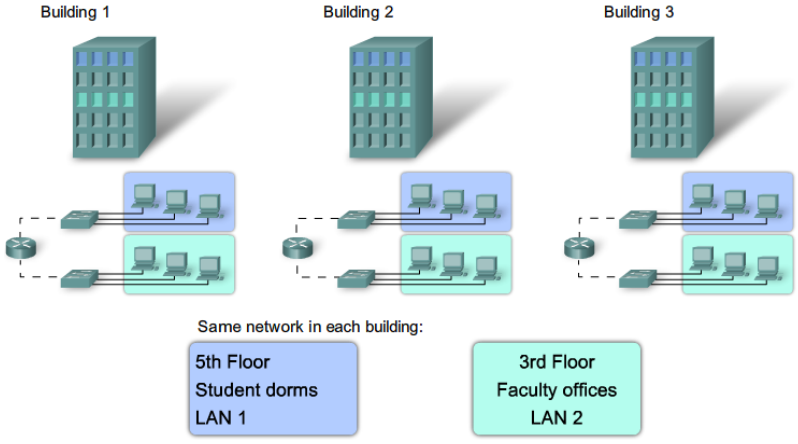
The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.1 Virtual Local Area Networks (VLANs)

Objectives

1. Explain the role of VLANs in a network.
2. Explain the role of trunking in a network.
3. Configure VLANs and trunks on switches.
4. Verify VLAN Information

Overview of VLANs

To appreciate why VLANs are being widely used today, consider a small community college with student dorms and the faculty offices all in one building (Building 1). The figure below shows the student computers in one LAN and the faculty computers in another LAN. This works fine because each department is physically together, so it is easy to provide them with their network resources. A year later, the college has grown and now has three buildings. In the figure, the original network is the same, but student and faculty computers are spread out across three buildings. The student dorms remain on the fifth floor and the faculty offices remain on the third floor. However, now the IT department wants to ensure that student computers all share the same security features and bandwidth controls. How can the network accommodate the shared needs of the geographically separated departments? Do you create a large LAN and wire each department together? How easy would it be to make changes to that network? It would be great to group the people with the resources they use regardless of their geographic location, and it would make it easier to manage their specific security and bandwidth needs.



The solution for the community college is to use a networking technology called a virtual LAN (VLAN). A VLAN allows a network administrator to create groups of logically networked devices that act as if they are on their own independent network, even if they share a common infrastructure with other VLANs. When you configure a VLAN, you can name it to describe the primary role of the users for that VLAN. As an example, all of the student computers in a school can be configured in the "Student" VLAN. Using VLANs, you can logically segment switched networks based on functions, departments, or project teams. These VLANs allow the network administrator to implement access and security policies to particular groups of users.

What are VLANs?

A VLAN (Virtual Local Area Network) is a logical network that groups devices regardless of their physical location. Each VLAN acts as a separate subnet. Access ports connect to one VLAN, while routers are needed for communication between VLANs.

Benefits of a VLAN Design

- Security: Isolate sensitive data (e.g., faculty vs students).
- Cost reduction: Better use of existing infrastructure.
- Performance: Reduce broadcast traffic and increase efficiency.
- Broadcast storm mitigation: Limit propagation across the network.
- Easier management: VLANs can be clearly named (e.g., Student, Faculty, Guest).
- Simplified project/application management: Easier to manage specialized teams.

VLAN ID Ranges

Access VLANs are divided into either a normal range or an extended range.

1. Normal Range VLANs
   - Used in small- and medium-sized business and enterprise networks.
   - Identified by a VLAN ID between 1 and 1005.
   - IDs 1002 through 1005 are reserved for Token Ring and FDDI VLANs.
   - IDs 1 and 1002 to 1005 are automatically created and cannot be removed.
   - Configurations are stored within a VLAN database file, called vlan.dat. The vlan.dat file is located in the flash memory of the switch.
2. Extended Range VLANs
   - Enable service providers to extend their infrastructure to a greater number of customers. Some global enterprises could be large enough to need extended range VLAN IDs.
   - Are identified by a VLAN ID between 1006 and 4094.
   - Support fewer VLAN features than normal range VLANs.
   - Are saved in the running configuration file.

Types of VLANs

1- Default VLAN

- All switch ports are automatically placed in VLAN 1.
- VLAN 1 is the **default native VLAN**.
- VLAN 1 is also the **default management VLAN**.
- VLAN 1 **cannot be renamed or deleted**.

2- Data VLAN

- Used to carry **user-generated traffic** only.
- Also called a **user VLAN** (groups users/devices).
- Modern networks often use **multiple data VLANs**.
- **Voice and management traffic are not allowed** on data VLANs.

3- Native VLAN

- Traffic from a VLAN must carry its VLAN ID when sent to another switch.
- **Trunk ports** connect switches and carry tagged traffic.
- **802.1Q trunks** add a 4-byte tag to Ethernet frames to indicate the VLAN.
- **Untagged traffic** can be sent over trunks, usually placed on the **native VLAN**.
- On Cisco switches, the default native VLAN is **VLAN 1**.
- Best practice: assign the native VLAN to an **unused VLAN**, different from VLAN 1 and other active VLANs.

4- Management VLAN

- A **management VLAN** handles network management traffic (SSH, Telnet, HTTPS, HTTP, SNMP).
- By default, **VLAN 1** is used for this on a Layer 2 switch.

5- Voice VLANs

A **Voice VLAN** is used to support VoIP traffic, ensuring low delay and priority to maintain call quality. The details of how to configure a network to support VoIP are beyond the scope of this experiment.

Defining VLAN Trunks

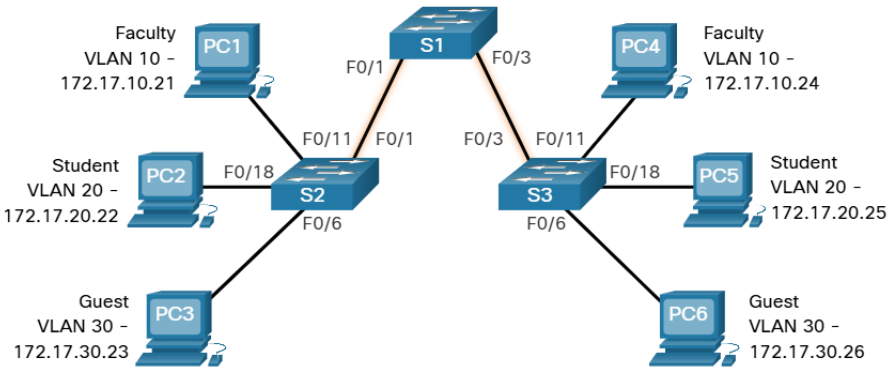
What is a trunk?

- A trunk is a connection (link) between two network devices (usually switches).
- It allows **multiple VLANs** to pass through the same cable.
- This makes it possible for devices on the same VLAN but on different switches to communicate **without a router**.

Trunk features:

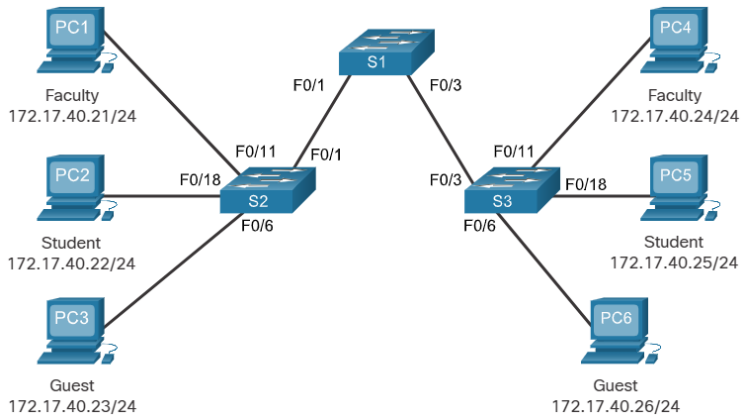
- Uses the **IEEE 802.1Q** standard to add a tag to each frame, showing which VLAN it belongs to.
- A trunk is not tied to one VLAN; it carries all the allowed VLANs.
- Can connect not only switches but also servers or devices with a network card that supports 802.1Q.
- By default, Cisco switches allow all VLANs on a trunk port.

In the figure below, the highlighted links between switches S1 and S2, and S1 and S3 are configured to transmit traffic coming from VLANs 10, 20, 30, and 99 (i.e., native VLAN) across the network. This network could not function without VLAN trunks.



Network without VLANs

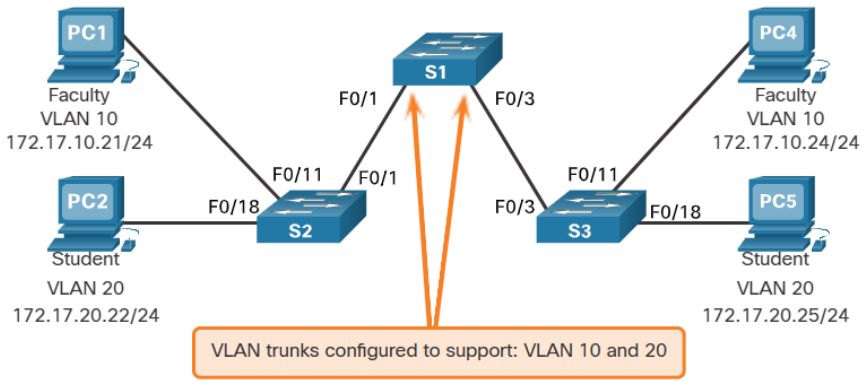
When a switch receives a broadcast frame on one of its ports, it forwards the frame out all other ports except the port where the broadcast was received. In the figure, the entire network is configured in the same subnet (172.17.40.0/24) and no VLANs are configured. As a result, when the faculty computer (PC1) sends out a broadcast frame, switch S2 sends that broadcast frame out all of its ports. Eventually the entire network receives the broadcast because the network is one broadcast domain.



Network with VLANs

VLANs are associated with and configured on individual switch ports. Devices attached to those ports have no concept of VLANs. However, these devices are configured with IP addressing and are members of a specific IP network. This is where the connection between VLAN and IP network is apparent. A VLAN is the equivalent to an IP network (or subnet). VLANs are configured on the switch, whereas IP addressing is configured on the device.

In the figure, the same network has now been segmented using two VLANs. Faculty devices are assigned to VLAN 10 and student devices are assigned to VLAN 20. When a broadcast frame is sent from the faculty computer, PC1, to switch S2, the switch forwards that broadcast frame only to those switch ports configured to support VLAN 10.



The ports that comprise the connection between switches S2 and S1 (ports F0/1), and between S1 and S3 (ports F0/3) are trunks and have been configured to support all the VLANs in the network.

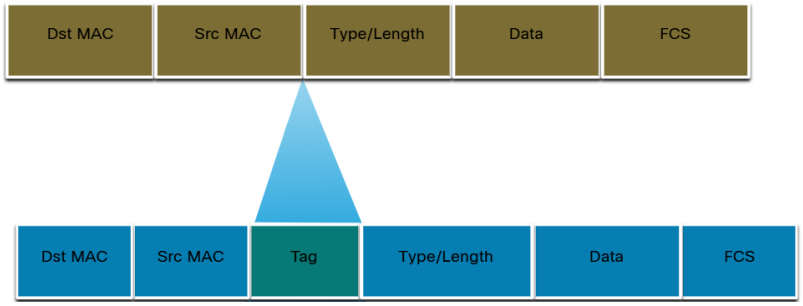
When S1 receives the broadcast frame on port F0/1, S1 forwards that broadcast frame out of the only other port configured to support VLAN 10, which is port F0/3. When S3 receives the broadcast frame on port F0/3, it forwards that broadcast frame out the only other port configured to support

VLAN 10, which is port F0/11. The broadcast frame arrives at the only other computer in the network configured in VLAN 10, which is faculty computer PC4.

VLAN Identification with a Tag

The standard Ethernet frame header does not contain information about the VLAN to which the frame belongs. Therefore, when Ethernet frames are placed on a trunk, information about the VLANs to which they belong must be added. This process, called tagging, is accomplished by using the IEEE 802.1Q header, specified in the IEEE 802.1Q standard. The 802.1Q header includes a 4-byte tag inserted within the original Ethernet frame header, specifying the VLAN to which the frame belongs.

When the switch receives a frame on a port configured in access mode and assigned a VLAN, the switch inserts a VLAN tag in the frame header, recalculates the Frame Check Sequence (FCS), and sends the tagged frame out of a trunk port.



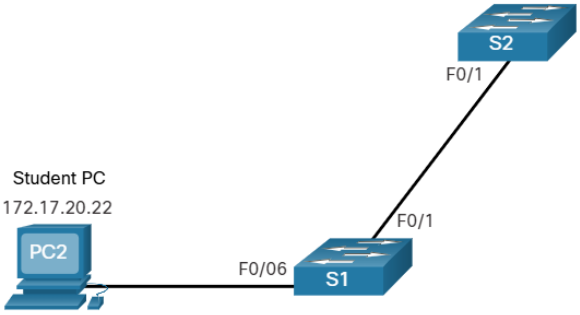
VLAN Creation Commands

When configuring normal range VLANs, the configuration details are stored in flash memory on the switch in a file called vlan.dat. Flash memory is persistent and does not require the **copy running-config startup-config** command. However, because other details are often configured on a Cisco switch at the same time that VLANs are created, it is good practice to save running configuration changes to the startup configuration.

The table displays the Cisco IOS command syntax used to add a VLAN to a switch and give it a name. Naming each VLAN is considered a best practice in switch configuration.

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`

In the topology, the student computer (PC2) has not been associated with a VLAN yet, but it does have an IP address of 172.17.20.22, which belongs to VLAN 20.



The example shows how the student VLAN (VLAN 20) is configured on switch S1.

```
S1# configure terminal
S1(config)# vlan 20
S1(config-vlan)# name student
S1(config-vlan)# end
```

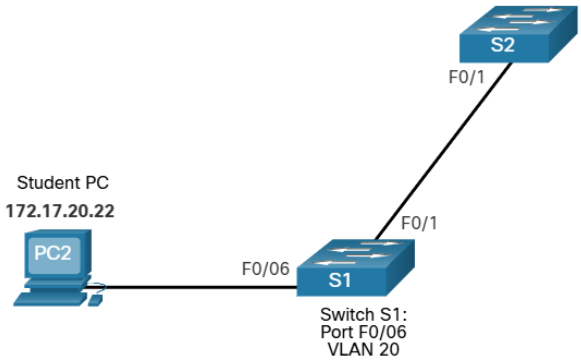
VLAN Port Assignment Commands

After creating a VLAN, the next step is to assign ports to the VLAN. The table displays the syntax for defining a port to be an access port and assigning it to a VLAN. The switchport mode access command is optional, but strongly recommended as a security best practice. With this command, the interface changes to strictly access mode. Access mode indicates that the port belongs to a single VLAN and will not negotiate to become a trunk link.

Task	IOS Command
Enter global configuration mode.	`Switch# configure terminal`
Enter interface configuration mode.	`Switch(config)# interface interface-id`
Set the port to access mode.	`Switch(config-if)# switchport mode access`
Assign the port to a VLAN.	`Switch(config-if)# switchport access vlan vlan-id`
Return to the privileged EXEC mode.	`Switch(config-if)# end`

Note: Use the **interface range** command to simultaneously configure multiple interfaces.

In the figure, port F0/6 on switch S1 is configured as an access port and assigned to VLAN 20. Any device connected to that port will be associated with VLAN 20. Therefore, in our example, PC2 is in VLAN 20.



In the figure above, port F0/6 on switch S1 is configured as an access port and assigned to VLAN 20. Any device connected to that port will be associated with VLAN 20. Therefore, in our example, PC2 is in VLAN 20.

```
S1# configure terminal
S1(config)# interface fa0/6
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 20
S1(config-if)# end
```

VLANs are configured on the switch port and not on the end device. PC2 is configured with an IPv4 address and subnet mask that is associated with the VLAN, which is configured on the switch port. In this example, it is VLAN 20. When VLAN 20 is configured on other switches, the network administrator must configure the other student computers to be in the same subnet as PC2 (172.17.20.0/24).

Verify VLAN Information

After a VLAN is configured, VLAN configurations can be validated using Cisco IOS **show** commands.

The **show vlan** command displays a list of all configured VLANs. The **show vlan** command can also be used with options. The complete syntax is **show vlan [brief** | **id** *vlan-id* | **name** *vlan-name* | **summary**].

The table describes the **show vlan** command options.

Task	Command Option
Display VLAN name, status, and its ports one VLAN per line.	`brief`
Display information about the identified VLAN ID number. For *vlan-id*, the range is 1 to 4094.	`id vlan-id`
Display information about the identified VLAN name. The *vlan-name* is an ASCII string from 1 to 32 characters.	`name vlan-name`
Display VLAN summary information.	`summary`

The **show vlan summary** command displays the count of all configured VLANs.

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

Other useful commands are the **show interfaces** *interface-id* **switchport** and the **show interfaces vlan** *vlan-id* command. For example, the **show interfaces fa0/18 switchport** command can be used to confirm that the FastEthernet 0/18 port has been correctly assigned to data and voice VLANs.

```
S1# show interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 20 (student)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: 150
Administrative private-vlan host-association: none
(Output omitted)
```

Change VLAN Port Membership

If the switch access port has been incorrectly assigned to a VLAN, then simply re-enter the **switchport access vlan** *vlan-id* interface configuration command with the correct VLAN ID. For instance, assume Fa0/18 was incorrectly configured to be on the default VLAN 1 instead of VLAN 20. To change the port to VLAN 20, simply enter **switchport access vlan 20**.
To change the membership of a port back to the default VLAN 1, use the **no switchport access vlan** interface configuration mode command as shown.
In the output for example, Fa0/18 is configured to be on the default VLAN 1 as confirmed by the **show vlan brief** command.

```
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
```

Notice that VLAN 20 is still active, even though no ports are assigned to it.

Delete VLANs

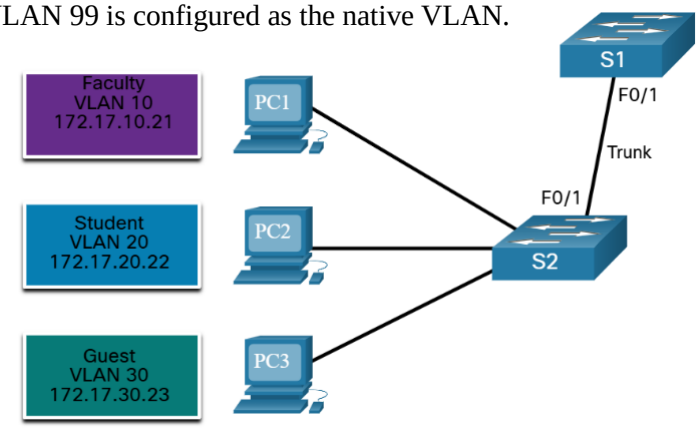
The **no vlan** *vlan-id* global configuration mode command is used to remove a VLAN from the switch vlan.dat file.
Caution: Before deleting a VLAN, reassign all member ports to a different VLAN first. Any ports that are not moved to an active VLAN are unable to communicate with other hosts after the VLAN is deleted and until they are assigned to an active VLAN.
The entire vlan.dat file can be deleted using the **delete flash:vlan.dat** privileged EXEC mode command. The abbreviated command version (**delete vlan.dat**) can be used if the vlan.dat file has not been moved from its default location. After issuing this command and reloading the switch, any previously configured VLANs are no longer present.

Trunk Configuration Commands

Now that you have configured and verified VLANs, it is time to configure and verify VLAN trunks. A VLAN trunk is a Layer 2 link between two switches that carries traffic for all VLANs (unless the allowed VLAN list is restricted manually or dynamically). To enable trunk links, configure the interconnecting ports with the set of interface configuration commands shown in the table.

Task	IOS Command
Enter global configuration mode.	`Switch# configure terminal`
Enter interface configuration mode.	`Switch(config)# interface interface-id`
Set the port to permanent trunking mode.	`Switch(config-if)# switchport mode trunk`
Sets the native VLAN to something other than VLAN 1.	`Switch(config-if)# switchport trunk native vlan vlan-id`
Specify the list of VLANs to be allowed on the trunk link.	`Switch(config-if)# switchport trunk allowed vlan vlan-list`
Return to the privileged EXEC mode.	`Switch(config-if)# end`

In the figure, VLANs 10, 20, and 30 support the Faculty, Student, and Guest computers (PC1, PC2, and PC3). The F0/1 port on switch S1 is configured as a trunk port and forwards traffic for VLANs 10, 20, and 30. VLAN 99 is configured as the native VLAN.



The example shows the configuration of port F0/1 on switch S1 as a trunk port. The native VLAN is changed to VLAN 99 and the allowed VLAN list is restricted to 10, 20, 30, and 99.

```
S1(config)# interface fastEthernet 0/1
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
```

Verify Trunk Configuration

The switch output displays the configuration of switch port F0/1 on switch S1. The configuration is verified with the **show interfaces** *interface-ID* **switchport** command.

```
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (VLAN0099)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: 10,20,30,99
Pruning VLANs Enabled: 2-1001
(output omitted)
```

The top highlighted area shows that port F0/1 has its administrative mode set to **trunk**. The port is in trunking mode. The next highlighted area verifies that the native VLAN is VLAN 99. Further down in the output, the bottom highlighted area shows that VLANs 10, 20, 30, and 99 are enabled on the trunk. **Note:** Another useful command for veryfing trunk interfaces is the **show interface trunk** command.

Procedures:

Dear students, please note that the lab problems sheet, the packet tracer activities and the practical discussion videos have been uploaded on your Microsoft Teams group. You are required to carefully study this experiment and then complete the lab sheet.

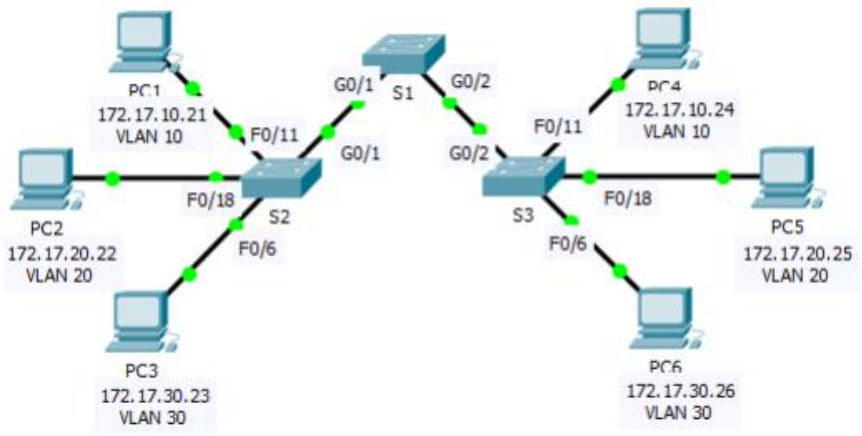
References

Cisco Networking Academy - CCNA: Switching, Routing, and Wireless Essentials.
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.1 Virtual Local Area Networks (VLANs).
Lab sheet

Problem 1: Configuring VLANs

In this activity, you will practice creating, naming VLANs, and assigning access ports to specific VLANs.



Task 1: View the Default VLAN Configuration

Step 1. Verify VLAN configuration.

Step 2. Verify connectivity between PCs on the same network.

Task 2: Configure VLANs

Step 1. Create and name VLANs on S1.

Step 2. Verify the VLAN configuration.

Step 3. Create the VLANs on S2 and S3.

Step 4. Verify the VLAN configuration.

Task 3: Assign VLANs to Ports

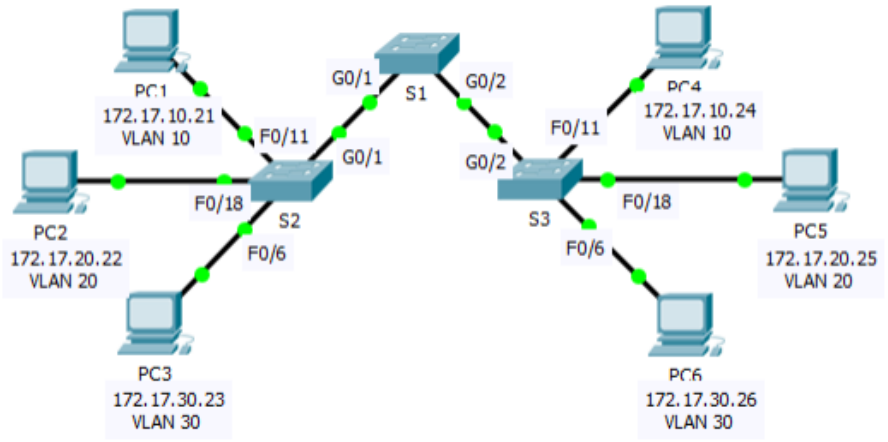
Step 1. Assign VLANs to the active ports on S2.

Step 2. Assign VLANs to the active ports on S3.

Step 3. Verify loss of connectivity.

Problem 2: Configuring Trunks

Trunks are required to pass VLAN information between switches. A port on a switch is either an access port or a trunk port. Access ports carry traffic from a specific VLAN assigned to the port. A trunk port by default is a member of all VLANs; therefore, it carries traffic for all VLANs. This activity focuses on creating trunk ports, and assigning them to a native VLAN other than the default.



.

Task 1: Verify VLANs

Step 1. Display the current VLANs.

Step 2. Verify loss of connectivity between PCs on the same network.

Task 2: Configure Trunks

Step 1. Configure trunking on S1 and use VLAN 99 as the native VLAN.

Step 2. Verify trunking is enabled on S2 and S3.

Step 3. Correct the native VLAN mismatch on S2 and S3.

Step 4. Verify configurations on S2 and S3.

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.2 Inter-VLAN Routing

Objectives

1. Describe options for configuring inter-VLAN routing.
2. Configure legacy inter-VLAN routing.
3. Configure router-on-a-Stick Inter-VLAN routing.
4. Configure inter-VLAN routing using Layer 3 switching.

Introduction

Now that you know how to configure VLANs on a network switch, the next step is to allow devices connected to the various VLANs to communicate with each other. In a previous experiment, you learned that each VLAN is a unique broadcast domain, so computers on separate VLANs are, by default, not able to communicate. There is a way to permit these end stations to communicate; it is called inter-VLAN routing. In this experiment, you will learn what inter-VLAN routing is and some of the different ways to accomplish inter-VLAN routing on a network.

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.

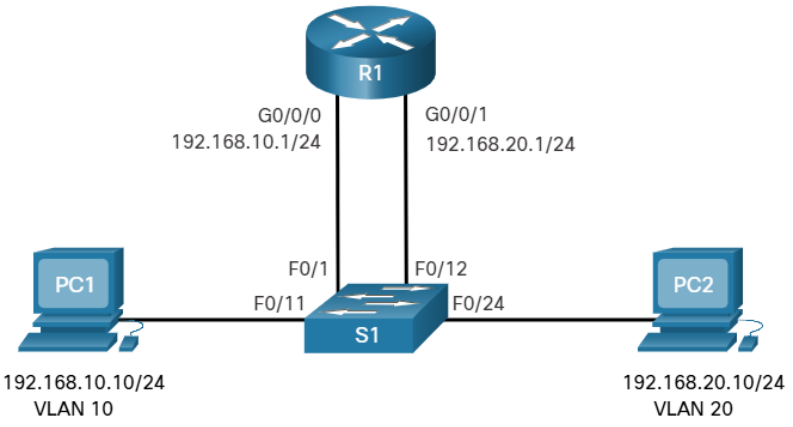
Legacy Inter-VLAN routing

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.

For example, refer to the topology where R1 has two interfaces connected to switch S1.

Notice in the example MAC address table of S1 is populated as follows:

- Fa0/1 port is assigned to VLAN 10 and is connected to the R1 G0/0/0 interface.
- Fa0/11 port is assigned to VLAN 10 and is connected to PC1.
- Fa0/12 port is assigned to VLAN 20 and is connected to the R1 G0/0/1 interface.
- Fa0/24 port is assigned to VLAN 20 and is connected to PC2.



When PC1 sends a packet to PC2 on another network, it forwards it to its default gateway 192.168.10.1. R1 receives the packet on its G0/0/0 interface and examines the destination address

of the packet. R1 then routes the packet out its G0/0/1 interface to the F0/12 port in VLAN 20 on S1. Finally, S1 forwards the frame to PC2.

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.

In our example, R1 required two separate Ethernet interfaces to route between VLAN 10 and VLAN 20. What if there were six (or more) VLANs to interconnect? A separate interface would be required for each VLAN. Obviously, this solution is not scalable.

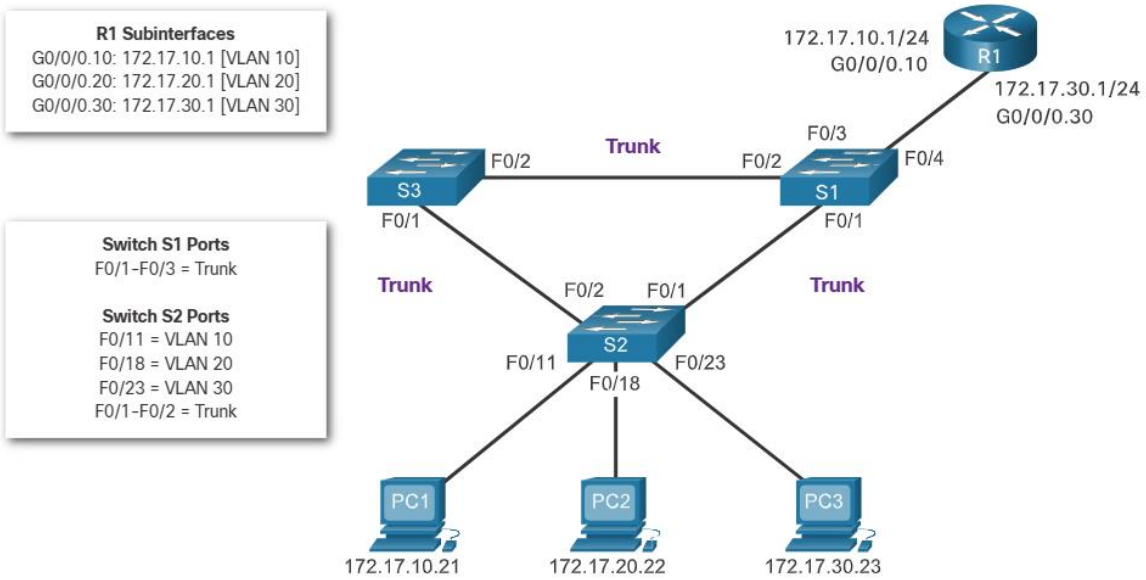
Router-on-a-Stick Inter-VLAN Routing

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.Q subinterface, the data frames are VLAN-tagged with the new VLAN and sent back out the physical interface.

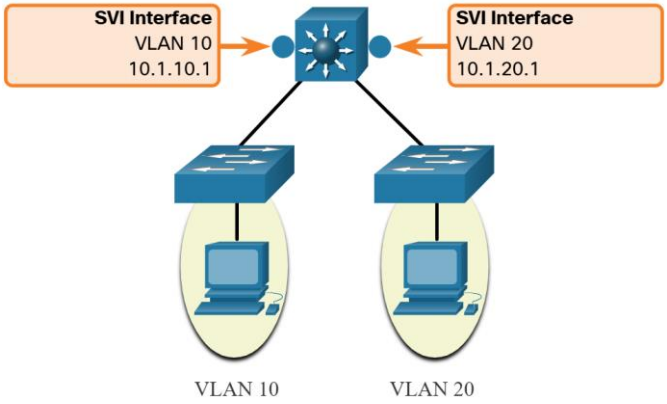


As seen in the animation, PC1 on VLAN 10 is communicating with PC3 on VLAN 30. When R1 accepts the tagged unicast traffic on VLAN 10, it routes that traffic to VLAN 30, using its

configured subinterfaces. Switch S2 removes the VLAN tag of the unicast frame and forwards the frame out to PC3 on port F0/23.
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.



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.

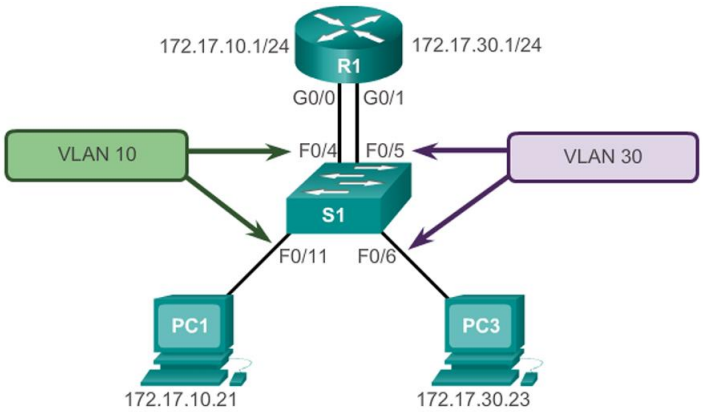
The following are advantages of using Layer 3 switches for inter-VLAN routing:

- 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.

Legacy Inter-VLAN Routing Configuration

Legacy inter-VLAN routing requires a router to have multiple physical interfaces, with each interface connected to a unique VLAN. Every interface is assigned an IP address corresponding to the subnet of its VLAN. Devices within each VLAN use the router as their default gateway to communicate with devices in other VLANs.



Switch Configuration:

```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/11
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/4
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/6
S1(config-if)# switchport access vlan 30
S1(config-if)# interface f0/5
S1(config-if)# switchport access vlan 30
S1(config-if)# end
*Mar 20 01:22:56.751: %SYS-5-CONFIG_I: Configured from console by
console
S1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

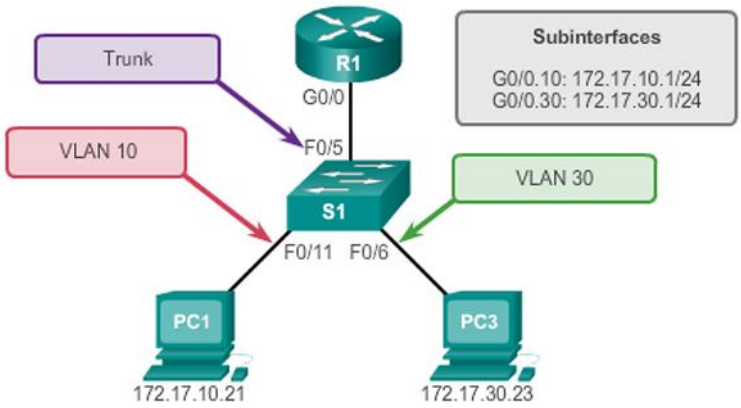
Router Interface Configuration:

```
R1(config)# interface g0/0
R1(config-if)# ip address 172.17.10.1 255.255.255.0
R1(config-if)# no shutdown
*Mar 20 01:42:12.951: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to up
*Mar 20 01:42:13.951: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up
R1(config-if)# interface g0/1
R1(config-if)# ip address 172.17.30.1 255.255.255.0
R1(config-if)# no shutdown
*Mar 20 01:42:54.951: %LINK-3-UPDOWN: Interface GigabitEthernet0/1,
changed state to up
*Mar 20 01:42:55.951: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed state to up
R1(config-if)# end
R1# copy running-config startup-config
```

Router-on-a-Stick Configuration

An alternative to legacy inter-VLAN routing is the use of VLAN trunking and subinterfaces. VLAN trunking enables a single physical router interface to handle traffic for multiple VLANs. The router's physical interface is connected to a trunk link on the adjacent switch, and subinterfaces are

then created on the router for each VLAN. Each subinterface is assigned an IP address corresponding to its VLAN's subnet and is configured to tag frames for that VLAN.



Switch Configuration:

```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/5
S1(config-if)# switchport mode trunk
S1(config-if)# end
S1#
```

Router Subinterface Configuration:

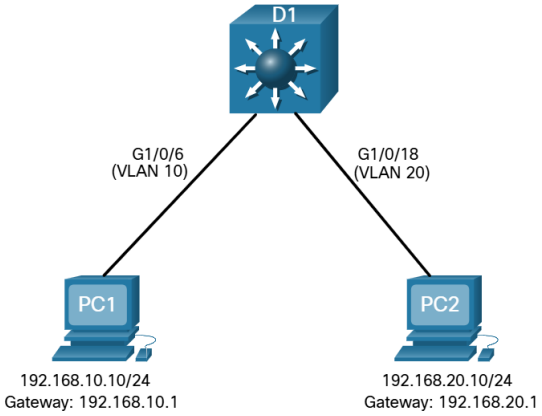
```
R1(config)# interface g0/0.10
R1(config-subif)# encapsulation dot1q 10
R1(config-subif)# ip address 172.17.10.1 255.255.255.0
R1(config-subif)# interface g0/0.30
R1(config-subif)# encapsulation dot1q 30
R1(config-subif)# ip address 172.17.30.1 255.255.255.0
R1(config)# interface g0/0
R1(config-if)# no shutdown
*Mar 20 00:20:59.299: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
 changed state to down
*Mar 20 00:21:02.919: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
 changed state to up
*Mar 20 00:21:03.919: %LINEPROTO-5-UPDOWN: Line protocol on
 changed state to down
*Mar 20 00:21:02.919: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
 changed state to up
*Mar 20 00:21:03.919: %LINEPROTO-5-UPDOWN: Line protocol on
 Interface GigabitEthernet0/0, changed state to up
```

Layer 3 Switch Configuration

Inter-VLAN routing using the router-on-a-stick method is simple to implement for a small to medium-sized organization. However, a large enterprise requires a faster, much more scalable method to provide inter-VLAN routing.

To provide inter-VLAN routing, Layer 3 switches use SVIs. SVIs are configured using the same **interface vlan** *vlan-id* command used to create the management SVI on a Layer 2 switch. A Layer 3 SVI must be created for each of the routable VLANs.

In the figure, the Layer 3 switch, D1, is connected to two hosts on different VLANs. PC1 is in VLAN 10 and PC2 is in VLAN 20, as shown.



Complete the following steps to configure S1 with VLANs and trunking:
Step 1. Create the VLANs.
Step 2. Create the SVI VLAN interfaces.
Step 3. Configure access ports.
Step 4. Enable IP routing.

1. Create the VLANs.
First, create the two VLANs as shown in the output.

```
D1(config)# vlan 10
D1(config-vlan)# name LAN10
D1(config-vlan)# vlan 20
D1(config-vlan)# name LAN20
D1(config-vlan)# exit
D1(config)#
```

2. Create the SVI VLAN interfaces.
Configure the SVI for VLANs 10 and 20. The IP addresses that are configured will serve as the default gateways to the hosts in the respective VLANs. Notice the informational messages showing the line protocol on both SVIs changed to up.

```
D1(config)# interface vlan 10
D1(config-if)# description Default Gateway SVI for 192.168.10.0/24
D1(config-if)# ip add 192.168.10.1 255.255.255.0
D1(config-if)# no shut
D1(config-if)# exit
D1(config)#
D1(config)# int vlan 20
D1(config-if)# description Default Gateway SVI for 192.168.20.0/24
D1(config-if)# ip add 192.168.20.1 255.255.255.0
D1(config-if)# no shut
D1(config-if)# exit
D1(config)#
*Sep 17 13:52:16.053: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
*Sep 17 13:52:16.160: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
```

3. Configure access ports.

Next, configure the access ports connecting to the hosts and assign them to their respective VLANs.

```
D1(config)# interface GigabitEthernet1/0/6
D1(config-if)# description Access port to PC1
D1(config-if)# switchport mode access
D1(config-if)# switchport access vlan 10
D1(config-if)# exit
D1(config)#
D1(config)# interface GigabitEthernet1/0/18
D1(config-if)# description Access port to PC2
D1(config-if)# switchport mode access
D1(config-if)# switchport access vlan 20
D1(config-if)# exit
```

4. Enable IP routing.

Finally, enable IPv4 routing with the **ip routing** global configuration command to allow traffic to be exchanged between VLANs 10 and 20. This command must be configured to enable inter-VAN routing on a Layer 3 switch for IPv4.

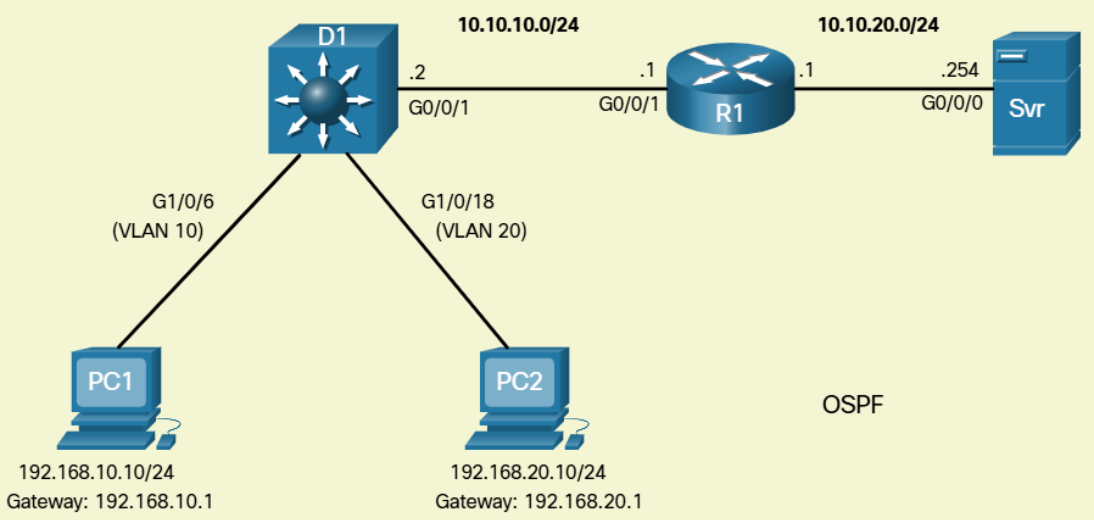
```
D1(config)# ip routing
D1(config)#
```

Routing on a Layer 3 Switch

If VLANs are to be reachable by other Layer 3 devices, then they must be advertised using static or dynamic routing. To enable routing on a Layer 3 switch, a routed port must be configured.

A routed port is created on a Layer 3 switch by disabling the switchport feature on a Layer 2 port that is connected to another Layer 3 device. Specifically, configuring the **no switchport** interface configuration command on a Layer 2 port converts it into a Layer 3 interface. Then the interface can be configured with an IPv4 configuration to connect to a router or another Layer 3 switch.

In the figure, the previously configured D1 Layer 3 switch is now connected to R1. R1 and D1 are both in an Open Shortest Path First (OSPF) routing protocol domain. Assume inter-VLAN has been successfully implemented on D1. The G0/0/1 interface of R1 has also been configured and enabled. Additionally, R1 is using OSPF to advertise its two networks, 10.10.10.0/24 and 10.20.20.0/24.



Complete the following steps to configure D1 to route with R1:
Step 1. Configure the routed port.
Step 2. Enable routing.
Step 3. Configure routing.
Step 4. Verify routing.
Step 5. Verify connectivity.

1. Configure the routed port.
Configure G0/0/1 to be a routed port, assign it an IPv4 address, and enable it.

```
D1(config)# interface GigabitEthernet0/0/1
D1(config-if)# description routed Port Link to R1
D1(config-if)# no switchport
D1(config-if)# ip address 10.10.10.2 255.255.255.0
D1(config-if)# no shut
D1(config-if)# exit
D1(config)#
```

2. Enable routing.
Ensure IPv4 routing is enabled with the **ip routing** global configuration command.

```
D1(config)# ip routing
D1(config)#
```

3. Configure routing.
Configure the OSPF routing protocol to advertise the VLAN 10 and VLAN 20 networks, along with the network that is connected to R1. Notice the message informing you that an adjacency has been established with R1.

```
D1(config)# router ospf 10
D1(config-router)# network 192.168.10.0 0.0.0.255 area 0
D1(config-router)# network 192.168.20.0 0.0.0.255 area 0
D1(config-router)# network 10.10.10.0 0.0.0.3 area 0
D1(config-router)# ^Z
D1#
*Sep 17 13:52:51.163: %OSPF-5-ADJCHG: Process 10, Nbr 10.20.20.1 on GigabitEthernet0/0/1 from LOADING to
FULL, Loading Done
D1#
```

4. Verify routing.
Verify the routing table on D1. Notice that D1 now has a route to the 10.20.20.0/24 network.

```
D1# show ip route | begin Gateway
Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C        10.10.10.0/30 is directly connected, GigabitEthernet0/0/1
L        10.10.10.2/32 is directly connected, GigabitEthernet0/0/1
O        10.10.20.0/24 [110/2] via 10.10.10.1, 00:00:06, GigabitEthernet0/0/1
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, Vlan10
L        192.168.10.1/32 is directly connected, Vlan10
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.20.0/24 is directly connected, Vlan20
L        192.168.20.1/32 is directly connected, Vlan20
D1#
```

5. Verify connectivity.

At this time, PC1 and PC2 are able to ping the server connected to R1.

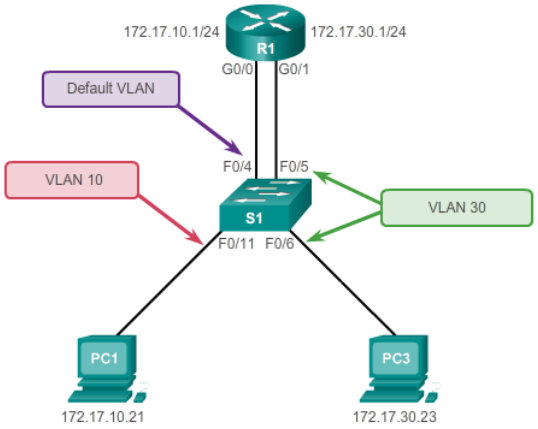
```
C:\Users\PC1> ping 10.20.20.254
Pinging 10.20.20.254 with 32 bytes of data:
Request timed out.
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Ping statistics for 10.20.20.254:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss).
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Users\PC1>
!================================================
C:\Users\PC2> ping 10.20.20.254
Pinging 10.20.20.254 with 32 bytes of data:
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Ping statistics for 10.20.20.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Users\PC2>
```

Inter-VLAN Configuration Issues

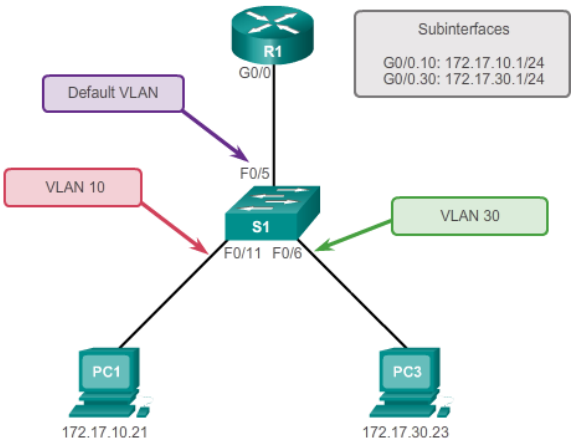
There are several common switch misconfigurations that can arise when configuring routing between multiple VLANs.

When using the traditional routing model for inter-VLAN routing, ensure that the switch ports that connect to the router interfaces are configured with the correct VLANs. If a switch port is not configured for the correct VLAN, devices configured on that VLAN cannot connect to the router interface; therefore, those devices are unable to send data to the other VLANs.

As shown in the Figure below, PC1 and router R1 interface G0/0 are configured to be on the same logical subnet, as indicated by their IP address assignment. However, the switch port F0/4 that connects to router R1 interface G0/0 has not been configured and remains in the default VLAN. Because router R1 is on a different VLAN than PC1, they are unable to communicate. To correct this problem, execute the **switchport access vlan 10** interface configuration mode command on switch port F0/4 on switch S1. When the switch port is configured for the correct VLAN, PC1 can communicate with router R1 interface G0/0, which allows it to access the other VLANs connected to router R1.

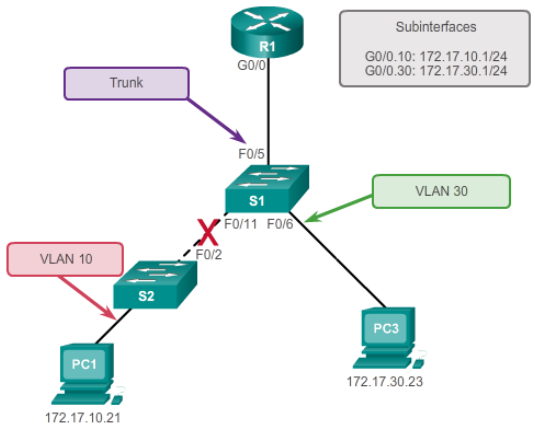


The following Figure topology shows the router-on-a-stick routing model. However, interface F0/5 on switch S1 is not configured as a trunk and is left in the default VLAN for the port. As a result, the router is unable to route between VLANs because each of its configured subinterfaces is unable to send or receive VLAN-tagged traffic.

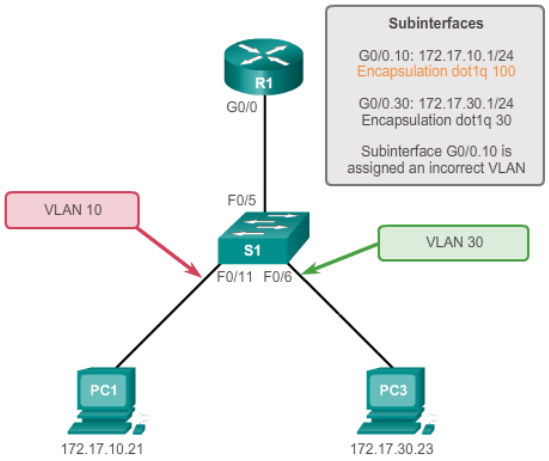


To correct this problem, issue the **switchport mode trunk** interface configuration mode command on switch port F0/5 on S1. This converts the interface to a trunk port, allowing a trunk to be established between R1 and S1. When the trunk is successfully established, devices connected to each of the VLANs are able to communicate with the subinterface assigned to their VLAN, thus enabling inter-VLAN routing.

The Figure topology below shows the trunk link between S1 and S2 is down. Because there is no redundant connection or path between the devices, all devices connected to S2 are unable to reach router R1. As a result, all devices connected to S2 are unable to route to other VLANs through R1. To reduce the risk of a failed inter-switch link disrupting inter-VLAN routing, redundant links and alternate paths should be accounted for within the network design.



With router-on-a-stick configurations, a common problem is assigning the wrong VLAN ID to the subinterface as shown in the Figure below. Router R1 has been configured with the wrong VLAN on subinterface G0/0.10, preventing devices configured on VLAN 10 from communicating with subinterface G0/0.10. This subsequently prevents those devices from being able to send data to other VLANs on the network. Using the **show interface** and the **show running-config** commands can be useful in troubleshooting this issue.



To correct this problem, configure subinterface G0/0.10 to be on the correct VLAN using the **encapsulation dot1q 10** subinterface configuration mode command. When the subinterface has been assigned to the correct VLAN, it is accessible by devices on that VLAN and the router can perform inter-VLAN routing. With proper verification, router configuration problems are quickly addressed, allowing inter-VLAN routing to function properly.

Procedures:

Dear students, please note that the lab problems **sheet, the packet tracer activities and the practical discussion videos have** been uploaded **on your Microsoft Teams group**. You are required to carefully study this experiment and then complete the lab sheet.

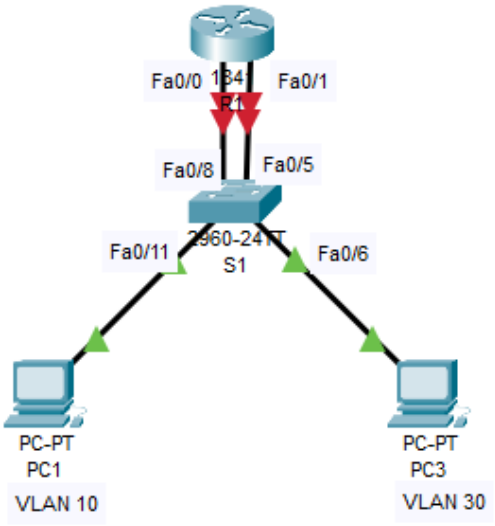
References

Cisco Networking Academy - CCNA: Switching, Routing, and Wireless Essentials.
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.2 Inter-VLAN Routing
Lab sheet

Problem 1: Configuring Traditional Inter-VLAN Routing

In this activity, you will configure traditional inter-VLAN routing simply by configuring two Fast Ethernet interfaces on a router. You will complete the configuration by adding VLANs to S1 and assigning VLANs to the correct ports. Then you will configure R1 with IP addressing. In traditional inter-VLAN routing, there are no additional, VLAN-related configurations needed on R1.



Task 1: Test Connectivity Without Inter-VLAN Routing

Step 1. Ping between PC1 and PC3.

Step 2. Switch to Simulation mode to monitor pings.

Task 2: Add VLANs

Step 1. Create VLANs on S1.

Step 2. Assign the VLANs to ports.

Step 3. Test connectivity between PC1 and PC3.

Step 4. Check results.

Task 3: Configure IP Addressing

Step 1. Configure IP addressing on R1.

Step 2. Check results.

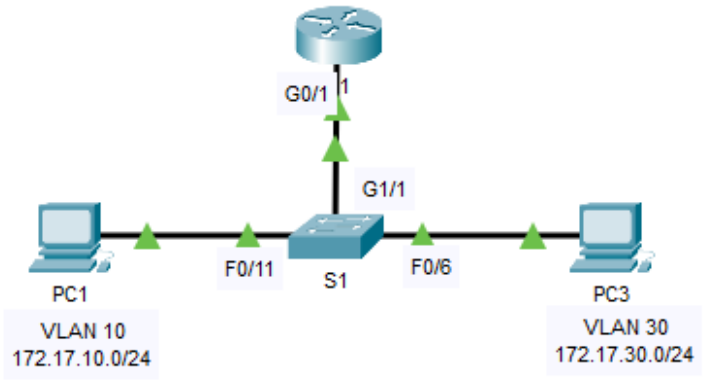
Task 4: Test Connectivity Again

Step 1. Ping between PC1 and PC3.

Step 2. Switch to simulation mode to monitor pings.

Problem 2: Configuring Router-on-a-Stick Inter-VLAN Routing

In this activity, you will configure Router-on-a-Stick inter-VLAN routing. R1 has one connection to S1. You will complete the configuration by adding VLANs to S1 and assigning VLANs to the correct ports. Then you will configure R1 with subinterfaces, 802.1Q encapsulation, and IP addressing.



Task 1: Test Connectivity Without Inter-VLAN Routing

Step 1. Ping between PC1 and PC3.

Step 2. Switch to Simulation mode to monitor pings.

Task 2: Add VLANs

Step 1. Create VLANs on S1.

Step 2. Assign the VLANs to ports.

Step 3. Test connectivity between PC1 and PC3.

Step 4. Check results.

Task 3: Configure IP Addressing

Step 1: Configure subinterfaces with 802.1Q encapsulation.

Step 2. Check results.

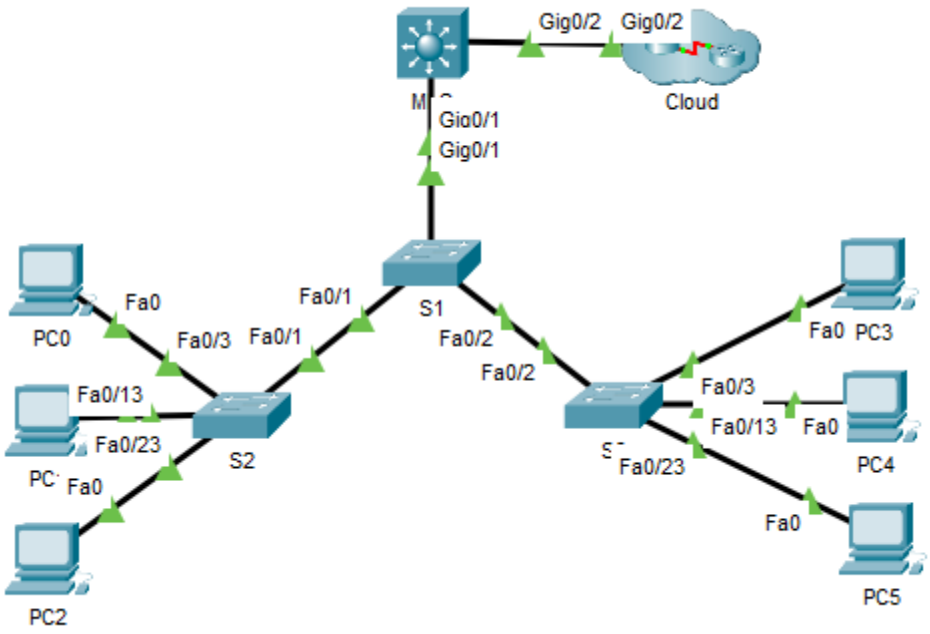
Task 4: Test Connectivity Again

Step 1. Ping between PC1 and PC3.

Step 2. Switch to Simulation mode to monitor pings.

Problem 3: Configure Layer 3 Switching and Inter-VLAN Routing

A multilayer switch like the Cisco Catalyst 3650 is capable of both Layer 2 switching and Layer 3 routing. One of the advantages of using a multilayer switch is this dual functionality. A benefit for a small to medium-sized company would be the ability to purchase a single multilayer switch instead of separate switching and routing network devices. Capabilities of a multilayer switch include the ability to route from one VLAN to another using multiple switched virtual interfaces (SVIs), as well as the ability to convert a Layer 2 switchport to a Layer 3 interface.



Task 1: Configure Layer 3 Switching

Task 2: Configure Inter-VLAN Routing

Step 1: Add VLANs.

Step 2: Configure SVI on MLS.

Step 3: Configure Trunking on MLS.

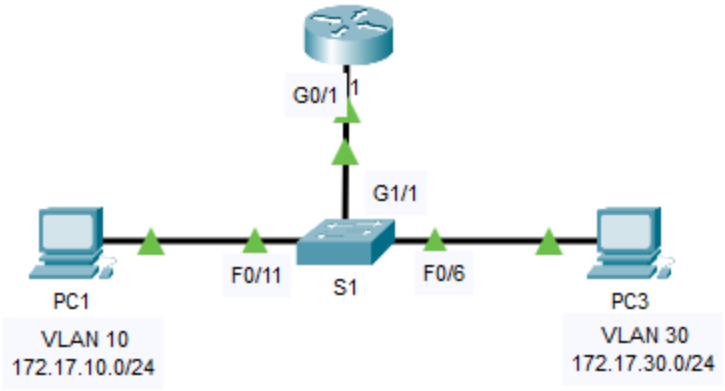
Step 4: Configure trunking on S1.

Step 5: Enable routing.

Step 6: Verify end-to-end connectivity.

Problem 4: Troubleshooting Inter-VLAN Routing

In this activity, you will troubleshoot connectivity problems caused by improper configurations related to VLANs and inter-VLAN routing.



Task 1: Locate the Network Problems

Task 2: Implement the Solutions

Task 3: Verify Network Connectivity

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.3 Scalling VlANs (DTP, VTP, STP)

Objectives

1. Illustrates the negotiation options of DTP.
2. Explain the need for VTP and its configuration.
3. Explain the role of redundancy in a converged network.
4. Summarize how STP works to eliminate Layer 2 loops in a converged network.

Dynamic Trunking Protocol (DTP)

Some Cisco switches have a proprietary protocol that lets them automatically negotiate trunking with a neighboring device. This protocol is called Dynamic Trunking Protocol (DTP). DTP can speed up the configuration process for a network administrator. Ethernet trunk interfaces support different trunking modes. An interface can be set to trunking or nontrunking, or to negotiate trunking with the neighbor interface. Trunk negotiation is managed by DTP, which operates on a point-to-point basis only, between network devices.

DTP is a Cisco proprietary protocol that is automatically enabled on Catalyst 2960 and Catalyst 3650 Series switches. DTP manages trunk negotiation only if the port on the neighbor switch is configured in a trunk mode that supports DTP. Switches from other vendors do not support DTP.
Caution: Some internetworking devices might forward DTP frames improperly, which can cause misconfigurations. To avoid this, turn off DTP on Cisco switch interfaces that are connected to devices that do not support DTP.

The default DTP configuration for Cisco Catalyst 2960 and 3650 switches is dynamic auto.

To enable trunking from a Cisco switch to a device that does not support DTP, use the *switchport mode trunk* and *switchport nonegotiate* interface configuration mode commands. This causes the interface to become a trunk, but it will not generate DTP frames.

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

To re-enable dynamic trunking protocol use the switchport mode dynamic auto command.

S1(config-if)# switchport mode dynamic auto

The switchport mode command has additional options for negotiating the interface mode. The full command syntax is the following:

Switch(config-if)# switchport mode { access | dynamic { auto | desirable } | trunk }

The options are described in the table below.

Option	Description
access	• Puts the interface (access port) into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. • The interface becomes a nontrunk interface, regardless of whether the neighboring interface is a trunk interface.
dynamic auto	• Makes the interface able to convert the link to a trunk link. • The interface becomes a trunk interface if the neighboring interface is set to trunk or desirable mode. • The default switchport mode for all Ethernet interfaces is dynamic auto.
dynamic desirable	• Makes the interface actively attempt to convert the link to a trunk link. • The interface becomes a trunk interface if the neighboring interface is set to trunk, desirable, or dynamic auto mode.
trunk	• Puts the interface into permanent trunking mode and negotiates to convert the neighboring link into a trunk link. • The interface becomes a trunk interface even if the neighboring interface is not a trunk interface.

The table below illustrates the results of the DTP configuration options on opposite ends of a trunk link connected to Catalyst 2960 switch ports. Best practice is to configure trunk links statically whenever possible.

	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

To determine and verify the current DTP mode, issue the *show dtp interface* command.

VLAN Trunking Protocol (VTP)

To carry traffic of a VLAN, it must be first configured on the switch. Suppose, if the user wants to send a frame from source to destination and the shortest path between them contains 1000 switches. To process a frame of any VLAN, VLANs should be configured first so, have to configure the same VLANs on all the 1000 switches manually. It will not be possible for the administrator to do that. Here comes VTP to the rescue.

VTP is CISCO proprietary protocol used to maintain consistency throughout the network or the user can say that synchronizing the VLAN information in the same VTP domain. VTP allows you

to add, delete and rename VLANs which is then propagated to other switches in the VTP domain. VTP advertisements can be sent over 802.1Q, and ISL trunks.

There are some requirements for VTP to communicate VLAN information between switches. These are:

- The VTP version must be same on the switches user wants to configure
- VTP domain name must be same on the switches
- One of the switches must be a server
- Authentication should match if applied

VTP modes – There are 3 modes:

- Server
- Client
- Transparent

Server – The switches are set to this mode by default. This mode allows you to create, add and delete VLANs. The changes you want to make should be done in this mode. Any changes that are done on this mode(on a particular switch) will be advertised to all the switches that are in the same VTP domain. In this mode, the configuration are saved in NVRAM.
Configuration:
Switch(config)#vtp mode server
Switch(config)#vtp domain ju
Switch(config)#vtp password cisco

User can verify the configuration by:
Switch(config)#do should vtp password
Switch(config)#do show vtp

Client – In this mode, the switches receive the updates and can also forward the updates to other switches (which are in the same VTP domain). The updates received here are not saved in NVRAM so all the configuration will be deleted if the switch is reset or reloaded i.e the switches will only learn and pass the VTP summary advertisements to the other switches.
Configuration:
Switch(config)#vtp mode client
Switch(config)#vtp domain ju
Switch(config)#vtp password cisco

Transparent – This mode only forwards the VTP summary advertisements through trunk link. The transparent mode switches can make their own local database which keep secret from other switches. The whole purpose of transparent mode is to forward the VTP summary advertisements but not to take part in the VLAN assignments.
Configuration:
Switch(config)#vtp mode transparent
Switch(config)#vtp domain ju
Switch(config)#vtp password cisco

Configuration Revision Number – The configuration revision number is a 32-bit number that indicates the level of revision for a VTP packet. This configuration number is tracked by every switch in order to find that the received information is more recent than the current version. Everytime one modification is done on the VLANs by the server switch, and the configuration revision number increases by one. The client mode devices receive it and check if the

configuration revision number that they received is latest or not by comparing their own configuration number with the number received. If the configuration number is greater than their own number then the devices update their configuration and pass it to other clients of the same VTP domain. If the configuration number is the same then the devices just pass it to other clients of the same VTP domain. User can check the configuration revision number by:
switch(config)#do show vtp status

Spanning Tree Protocol (STP)

Redundancy is an important part of the hierarchical design for eliminating single points of failure and preventing disruption of network services to users. Redundant networks require the addition of physical paths, but logical redundancy must also be part of the design. Having alternate physical paths for data to traverse the network makes it possible for users to access network resources, despite path disruption. However, redundant paths in a switched Ethernet network may cause both physical and logical Layer 2 loops.

Ethernet LANs require a loop-free topology with a single path between any two devices. A loop in an Ethernet LAN can cause continued propagation of Ethernet frames until a link is disrupted and breaks the loop. This Looping of frames causes three problems as stated below:

- MAC table instability – Due to looping of frame around LAN, MAC-Table of switch get changed frequently. Looping causes incorrect MAC-table entries resulting in incorrect frame delivery.
- Broadcast Storm – Repeated forwarding of frames around links in LAN causes the inefficient use of links.
- Multiple Frame Transmission – A very serious negative effect of looping is that multiple copies of same frame are delivered to host. This process left host with confusion.

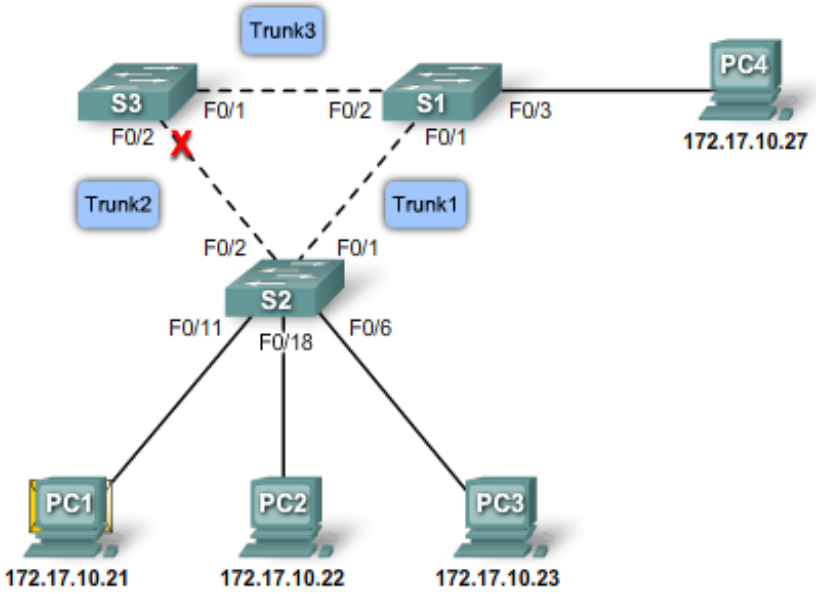
Unlike the Layer 3 protocols, IPv4 and IPv6, Layer 2 Ethernet does not include a mechanism to recognize and eliminate endlessly looping frames. Both IPv4 and IPv6 include a mechanism that limits the number of times a Layer 3 networking device can retransmit a packet. A router will decrement the TTL (Time to Live) in every IPv4 packet, and the Hop Limit field in every IPv6 packet. When these fields are decremented to 0, a router will drop the packet.

Spanning Tree Protocol (STP) is a loop-prevention network protocol that allows for redundancy while creating a loop-free Layer 2 topology. It ensures that there is only one logical path between all destinations on the network by intentionally blocking redundant paths that could cause a loop. A port is considered blocked when network traffic is prevented from entering or leaving that port. This does not include Bridge Protocol Data Unit (BPDU)frames that are used by STP to prevent loops. You will learn more about STP BPDU frames later in the experiment. Blocking the redundant paths is critical to preventing loops on the network. The physical paths still exist to provide redundancy, but these paths are disabled to prevent the loops from occurring. If the path is ever needed to compensate for a network cable or switch failure, STP recalculates the paths and unblocks the necessary ports to allow the redundant path to become active.

In the figure below, all switches now have STP enabled:

1. PC1 sends a broadcast out onto the network.
2. Switch S3 is configured with STP and has set the port for Trunk2 to a blocking state. The blocking state prevents ports from being used to forward switch traffic, preventing a loop from occurring. Switch S2 forwards a broadcast frame out all switch ports, except the originating port from PC1, and the port on Trunk2, which leads to the blocked port on S3.

3. Switch S1 receives the broadcast frame and forwards it out all of its switch ports, where it reaches PC4 and S3. S3 does not forward the frame back to S2 over Trunk2 because of the blocked port. The Layer 2 loop is prevented.



If the trunk link between switch S2 and switch S1 fails. Switch S3 unblocks the previously blocked port for Trunk2 and allows the broadcast traffic to traverse the alternate path around the network, permitting communication to continue. If this link comes back up, STP reconverges and the port on S3 is again blocked.

How STP Works?

STP uses the Spanning Tree Algorithm (STA) to determine which switch ports on a network need to be configured for blocking to prevent loops from occurring. The STA designates a single switch as the root bridge and uses it as the reference point for all path calculations. In the figure below the root bridge, switch S1, is chosen through an election process. All switches participating in STP exchange BPDU frames to determine which switch has the lowest bridge ID (BID) on the network. The switch with the lowest BID automatically becomes the root bridge for the STA calculations.

The BPDU is the message frame exchanged by switches for STP. Each BPDU contains a BID that identifies the switch that sent the BPDU. The BID contains a priority value, the MAC address of the sending switch, and an optional extended system ID. The lowest BID value is determined by the combination of these three fields. Example of BID: {32768 - 0002.4A7C.4BC4}.

After the root bridge has been determined, the STA calculates the shortest path to the root bridge. Each switch uses the STA to determine which ports to block. While the STA determines the best paths to the root bridge for all destinations in the broadcast domain, all traffic is prevented from forwarding through the network. The STA considers both path and port costs when determining which path to leave unblocked. The path costs are calculated using port cost values associated with port speeds for each switch port along a given path. The sum of the port cost values determines the overall path cost to the root bridge. If there is more than one path to choose from, STA chooses the path with the lowest path cost.

When the STA has determined which paths are to be left available, it configures the switch ports into distinct port roles. The port roles describe their relation in the network to the root bridge and whether they are allowed to forward traffic.

Port Roles

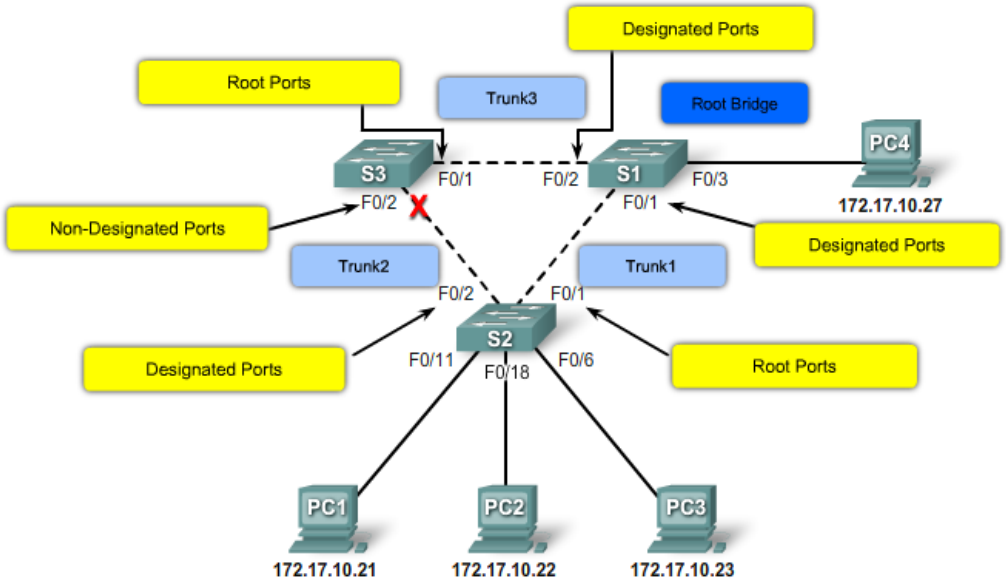
There are four distinct port roles that switch ports are automatically configured for during the spanning-tree process.

Root Port - The root port exists on non-root bridges and is the switch port with the best path to the root bridge. Root ports forward traffic toward the root bridge. One root port is allowed per bridge. In the example below, switch S1 is the root bridge and switches S2 and S3 have root ports defined on the trunk links connecting back to S1.

Designated Port - The designated port exists on root and non-root bridges. For root bridges, all switch ports are designated ports. For non-root bridges, a designated port is the switch port that receives and forwards frames toward the root bridge as needed. Only one designated port is allowed per segment. If multiple switches exist on the same segment, an election process determines the designated switch, and the corresponding switch port begins forwarding frames for the segment. Designated ports are capable of populating the MAC table.
In the example below, switch S1 has both sets of ports for its two trunk links configured as designated ports. Switch S2 also has a designated port configured on the trunk link going toward switch S3.

Non-designated Port - The non-designated port is a switch port that is blocked, so it is not forwarding data frames and not populating the MAC address table with source addresses. A non-designated port is not a root port or a designated port. For some variants of STP, the non-designated port is called an alternate port.
In the example below, switch S3 has the only non-designated ports in the topology. The non-designated ports prevent the loop from occurring.

Disabled Port - The disabled port is a switch port that is administratively shut down. A disabled port does not function in the spanning-tree process. There are no disabled ports in this example.



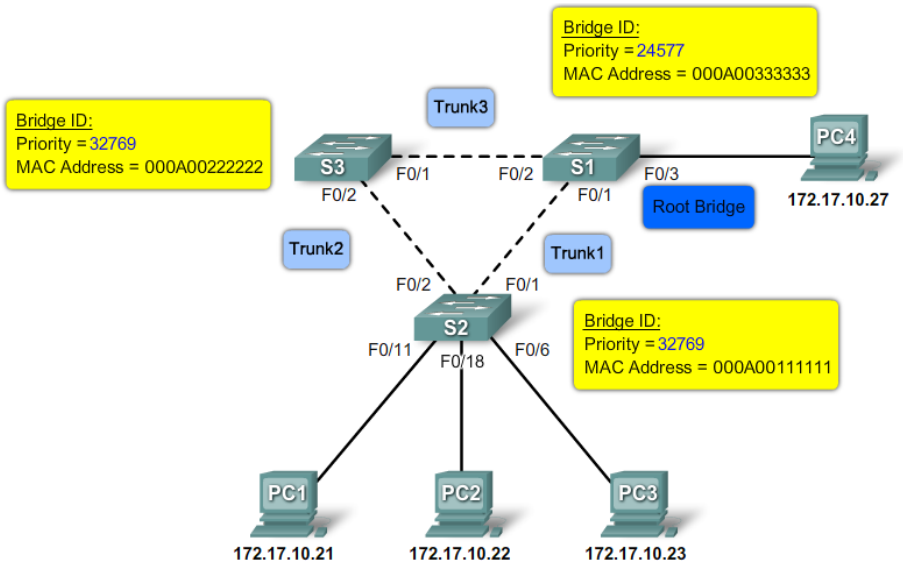
The Root Bridge

Every spanning-tree instance (switched LAN or broadcast domain) has a switch designated as the root bridge. The root bridge serves as a reference point for all spanning-tree calculations to determine which redundant paths to block.

An election process determines which switch becomes the root bridge. The figure below shows the BID fields. The BID is made up of a priority value, an extended system ID, and the MAC address of the switch.

All switches in the broadcast domain participate in the election process. After a switch boots, it sends out BPDU frames containing the switch BID and the root ID every 2 seconds. By default, the root ID matches the local BID for all switches on the network. The root ID identifies the root bridge on the network. Initially, each switch identifies itself as the root bridge after bootup.

As the switches forward their BPDU frames, adjacent switches in the broadcast domain read the root ID information from the BPDU frame. If the root ID from the BPDU received is lower than the root ID on the receiving switch, the receiving switch updates its root ID identifying the adjacent switch as the root bridge. Note: It may not be an adjacent switch, but any other switch in the broadcast domain. The switch then forwards new BPDU frames with the lower root ID to the other adjacent switches. Eventually, the switch with the lowest BID ends up being identified as the root bridge for the spanning-tree instance.



Best Paths to the Root Bridge

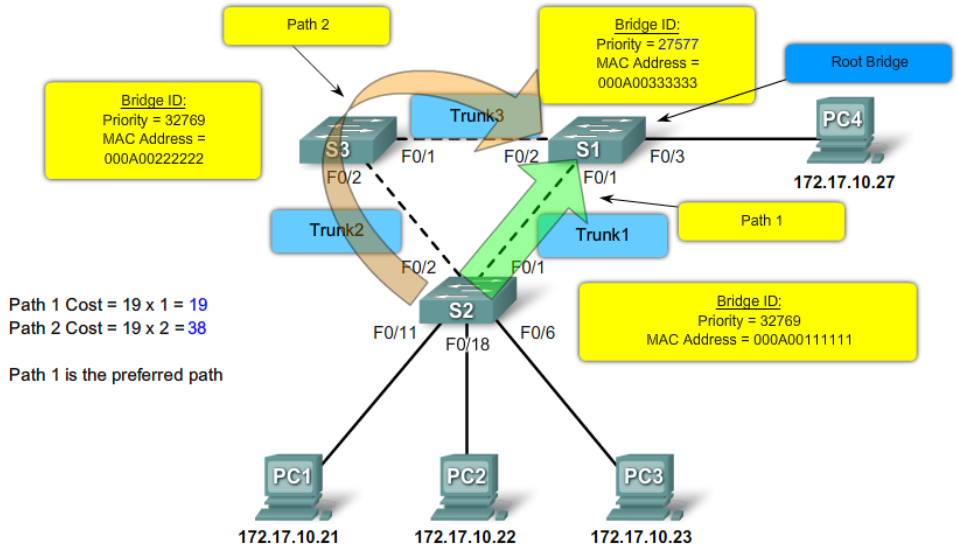
When the root bridge has been designated for the spanning-tree instance, the STA starts the process of determining the best paths to the root bridge from all destinations in the broadcast domain. The path information is determined by summing up the individual port costs along the path from the destination to the root bridge.

The default port costs are defined by the speed at which the port operates, 10-Gb/s Ethernet ports have a port cost of 2, 1-Gb/s Ethernet ports have a port cost of 4, 100-Mb/s Fast Ethernet ports have a port cost of 19, and 10-Mb/s Ethernet ports have a port cost of 100.

Although switch ports have a default port cost associated with them, the port cost is configurable as shown in the figure below. The ability to configure individual port costs gives the administrator the flexibility to control the spanning-tree paths to the root bridge.

```
S2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
S2(config)#interface f0/1
S2(config-if)#spanning-tree cost 25
S2(config-if)#end
S2#
```

Path cost is the sum of all the port costs along the path to the root bridge. The paths with the lowest path cost become the preferred path, and all other redundant paths are blocked. In the example, the path cost from switch S2 to the root bridge switch S1, over path 1 is 19 (based on the IEEE-specified individual port cost), while the path cost over path 2 is 38. Because path 1 has a lower overall path cost to the root bridge, it is the preferred path. STP then configures the redundant path to be blocked, preventing a loop from occurring.



To verify the port and path cost to the root bridge, enter the show spanning-tree privileged EXEC mode command. The Cost field in the output is the total path cost to the root bridge. This value changes depending on how many switch ports need to be traversed to get to the root bridge. In the output, each interface is also identified with an individual port cost of 19.

```
S2#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority 27577
              Address     000A.0033.3333
              Cost        19
              Port        1
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     000A.0011.1111
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
F0/1             Root FWD 19         128.1    Edge P2p
F0/2             Desg FWD 19         128.2    Edge P2p
```

Configure and Verify the BID

When a specific switch is to become a root bridge, the bridge priority value needs to be adjusted to ensure it is lower than the bridge priority values of all the other switches on the network. There are two different configuration methods that you can use to configure the bridge priority value on a Cisco Catalyst switch.

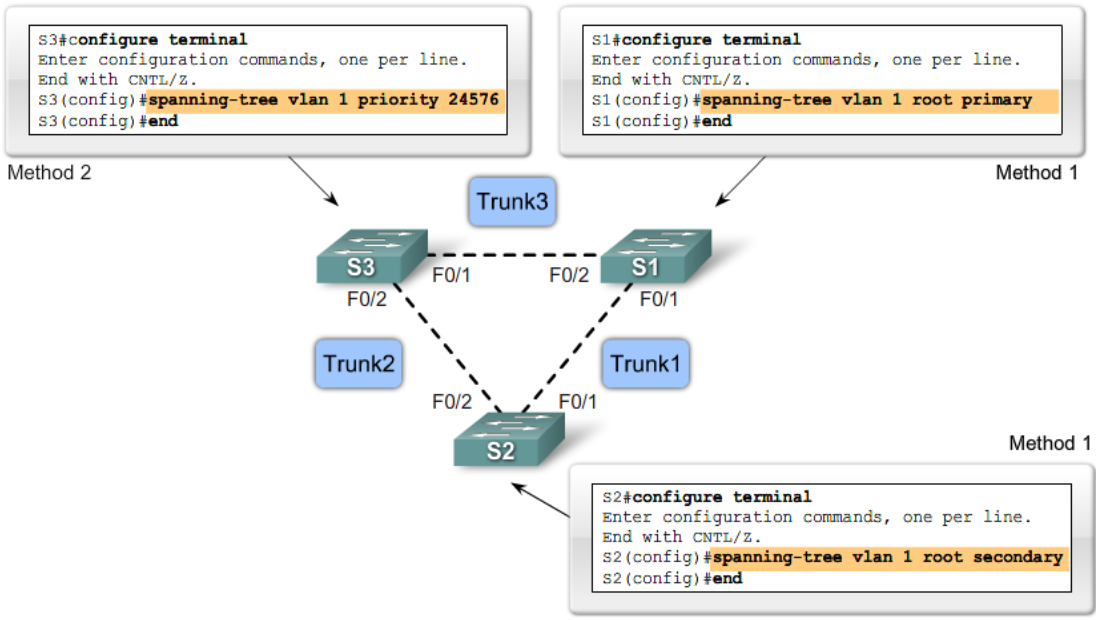
Method 1 - To ensure that the switch has the lowest bridge priority value, use the *spanning-tree vlan vlan-id root primary* command in global configuration mode. The priority for the switch is set to the predefined value of 24576 or to the next 4096 increment value below the lowest bridge priority detected on the network.

If an alternate root bridge is desired, use the *spanning-tree vlan vlan-id root secondary* global configuration mode command. This command sets the priority for the switch to the predefined value of 28672. This ensures that this switch becomes the root bridge if the primary root bridge fails and a new root bridge election occurs and assuming that the rest of the switches in the network have the default 32768 priority value defined.

In the example below, switch S1 has been assigned as the primary root bridge using the *spanning-tree vlan 1 root primary* global configuration mode command, and switch S2 has been configured as the secondary root bridge using the *spanning-tree vlan 1 root secondary* global configuration mode command.

Method 2 - Another method for configuring the bridge priority value is using the *spanning-tree vlan vlan-id priority value* global configuration mode command. This command gives you more granular control over the bridge priority value. The priority value is configured in increments of 4096 between 0 and 65536.

In the example below, switch S3 has been assigned a bridge priority value of 24576 using the *spanning-tree vlan 1 priority 24576* global configuration mode command.



To verify the bridge priority of a switch, use the *show spanning-tree* privileged EXEC mode command. In the example, the priority of the switch has been set to 24576. Also notice that the switch is designated as the root bridge for the spanning-tree instance.

```
S1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
             Address     000A.0033.3333
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
             Address     000A.0033.3333
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Desg FWD 4         128.1    Shr
Fa0/2            Desg FWD 4         128.2    Shr
S1#
```

Port States

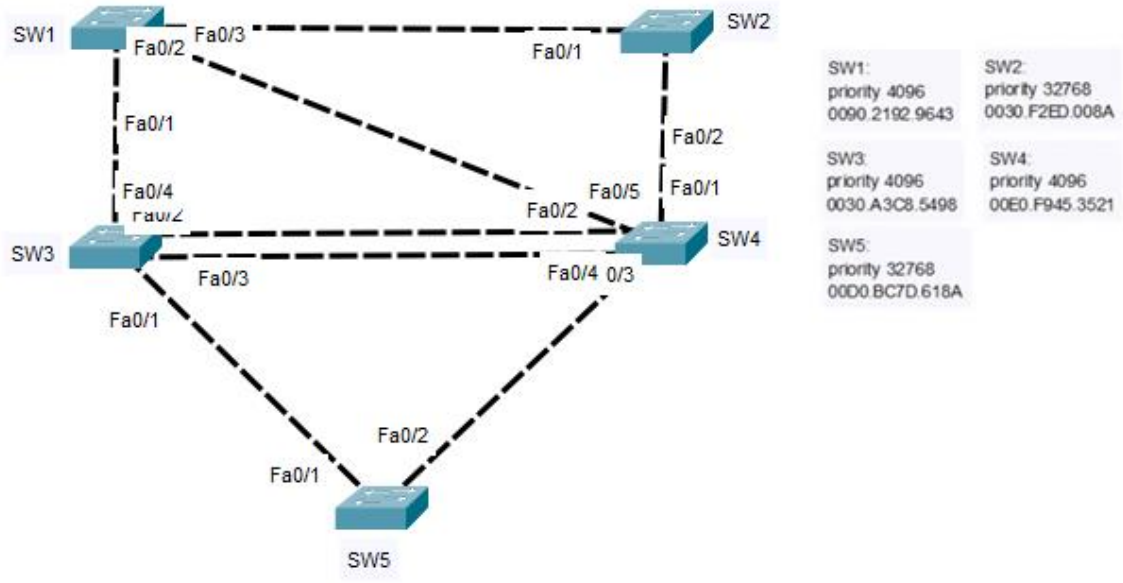
STP determines the logical loop-free path throughout the broadcast domain. The spanning tree is determined through the information learned by the exchange of the BPDU frames between the interconnected switches. To facilitate the learning of the logical spanning tree, each switch port transitions through five possible port states and three BPDU timers.

The spanning tree is determined immediately after a switch is finished booting up. If a switch port were to transition directly from the blocking to the forwarding state, the port could temporarily create a data loop if the switch was not aware of all topology information at the time. For this reason, STP introduces five port states. The table below summarizes what each port state does. The following provides some additional information on how the port states ensure that no loops are created during the creation of the logical spanning tree.

Processes	Blocking	Listening	Learning	Forwarding	Disable
Receives and process BPDUs	[1]YES	YES	YES	YES	NO
Forward data frames received on interface	NO	NO	NO	YES	NO
Forward data frames switched from another interface	NO	NO	NO	YES	NO
Learn MAC addresses	NO	NO	YES	YES	NO

Blocking - The port is a non-designated port and does not participate in frame forwarding. The port receives BPDU frames to determine the location and root ID of the root bridge switch and what port roles each switch port should assume in the final active STP topology.
Listening - STP has determined that the port can participate in frame forwarding according to the BPDU frames that the switch has received thus far. At this point, the switch port is not only receiving BPDU frames, it is also transmitting its own BPDU frames and informing adjacent switches that the switch port is preparing to participate in the active topology.
Learning - The port prepares to participate in frame forwarding and begins to populate the MAC address table.
Forwarding - The port is considered part of the active topology and forwards frames and also sends and receives BPDU frames.
Disabled - The Layer 2 port does not participate in spanning tree and does not forward frames. The disabled state is set when the switch port is administratively disabled.

As a summary on how STP Works, consider the following topology and determine the port state of all interfaces:



1- Elect a root bridge: The switch with the lowest BID (1- Priority, 2- MAC Address)

2- Elect a root port for each switch except the RB: The port that has the lowest cost path to the root bridge (Highest Bandwidth & Lowest number of links)

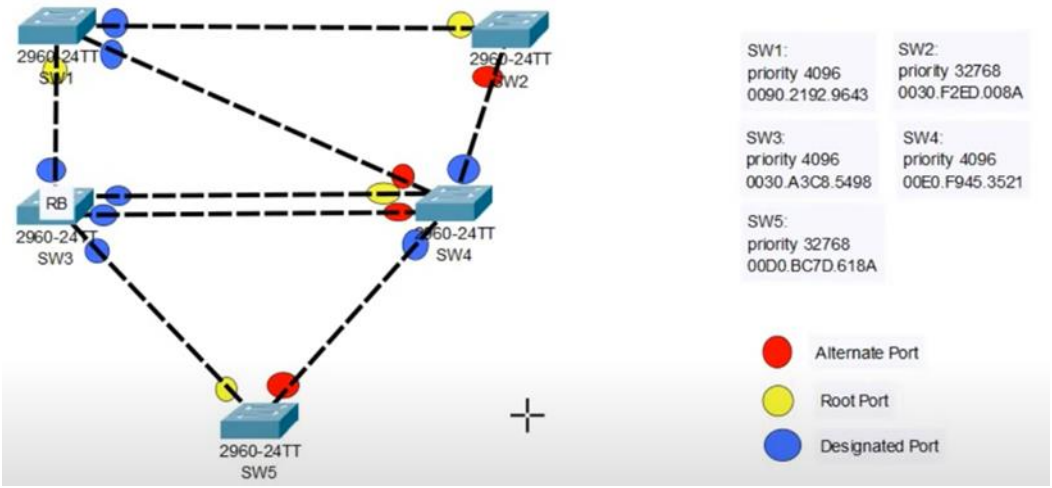
- If a switch has multiple equal cost paths to the RB, it will select the path with the lowest neighbour's BID (If it is connected directly to the RB or has multiple links connected to that neighbour, it will select the lowest Port ID (Example of Port ID: Fa0/1)

3- All the ports that are opposite the root ports and all the root bridge's ports will be designated ports.

4- Elect a designated port for each remaining link: The switch that has the lowest cost path to the root bridge

- If both switches have the same cost path, it will select the switch with lowest BID.

5- All the other ports will be alternate ports (Blocking state)



Different Versions of STP

Up to now, we have used the term Spanning Tree Protocol and the acronym STP, which can be misleading. Many professionals generically use these to refer to the various implementations of spanning tree, such as Rapid Spanning Tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MSTP). In order to communicate spanning tree concepts correctly, it is important to refer to the implementation or standard of spanning tree in context.
Several varieties of spanning tree protocols have emerged since the original IEEE 802.1D specification, here is some examples:

STP: Spanning Tree Protocol

- The original specification of STP
- Developed by IEEE
- Defined in 802.1D
- Slow: uses a slow convergence algorithm that can take up to 30 seconds.
- STP has five port states: blocking, listening, learning, forwarding, and disabled.
- Provides a spanning tree instance for all VLANs

RSTP: Rapid Spanning Tree Protocol

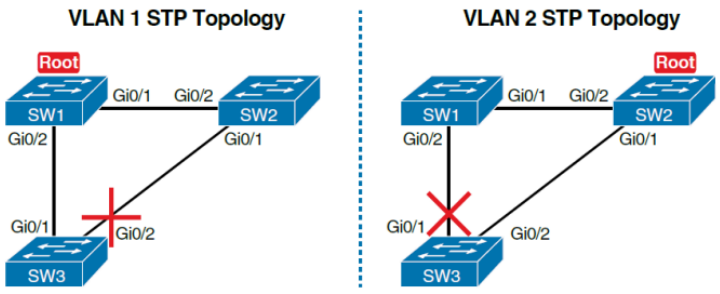
- An evolution of STP
- Developed by IEEE
- Defined in 802.1W
- Fast: employs a rapid convergence algorithm that can converge in less than one second.
- RSTP has only three port states: discarding, learning, and forwarding.
- Provides a spanning tree instance for all VLANs

PVST+: Per-VLAN Spanning Tree Plus

- Based on STP 802.1D
- Developed by Cisco
- Used by default on Cisco Switches
- Slow
- Provides a spanning tree instance for each configured VLAN

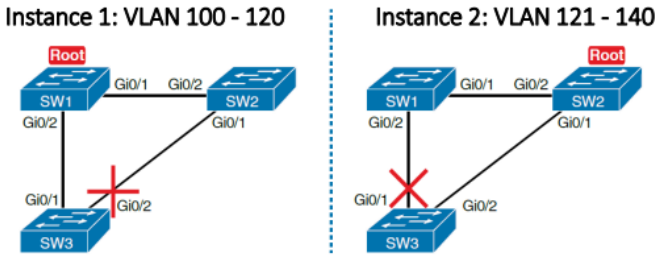
RPVST+: Rapid Per-VLAN Spanning Tree Plus

- Based on RSTP 802.1W
- Developed by Cisco
- Fast
- Provides a spanning tree instance for each configured VLAN



MSTP: Multiple Spanning Tree Protocol

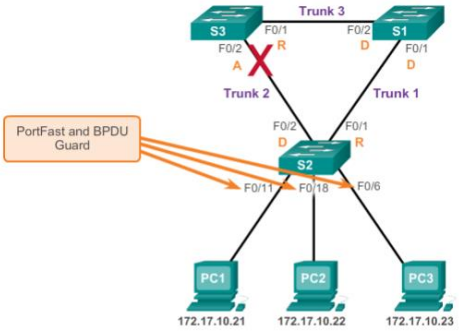
- Developed by Cisco and IEEE, defined in 802.1S
- Fast
- Provides a spanning tree instance for group of VLANs



PortFast and BPDU Guard

When a device is connected to a switch port or when a switch powers up, the switch port goes through both the listening and learning states, each time waiting for the Forward Delay timer to expire. This delay is 15 seconds for each state, listening and learning, for a total of 30 seconds. This delay can present a problem for DHCP clients trying to discover a DHCP server. DHCP messages from the connected host will not be forwarded for the 30 seconds of Forward Delay timers and the DHCP process may timeout. The result is that an IPv4 client will not receive a valid IPv4 address.

When a switch port is configured with PortFast, that port transitions from blocking to forwarding state immediately, bypassing the usual 802.1D STP transition states (the listening and learning states) and avoiding a 30 second delay. Because the purpose of PortFast is to minimize the time that access ports must wait for spanning tree to converge, it should only be used on access ports. If you enable PortFast on a port connecting to another switch, you risk creating a spanning tree loop.




```
S2(config)# interface FastEthernet 0/11
S2(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 FastEthernet0/11 but will only
 have effect when the interface is in a non-trunking mode.
S2(config-if)# spanning-tree bpduguard enable
S2(config-if)# end
```

In a valid PortFast configuration, BPDUs should never be received on PortFast-enabled switch ports because that would indicate that another bridge or switch is connected to the port. This potentially causes a spanning tree loop. To prevent this type of scenario from occurring, Cisco switches support a feature called BPDU guard. When enabled, BPDU guard immediately puts the switch port in an errdisabled (error-disabled) state on receipt of any BPDU. This protects against potential loops by effectively shutting down the port.

Procedures:

Dear students, please note that the lab problems **sheet, the packet tracer activities and the practical discussion videos have** been uploaded **on your Microsoft Teams group**. You are required to carefully study this experiment and then complete the lab sheet.

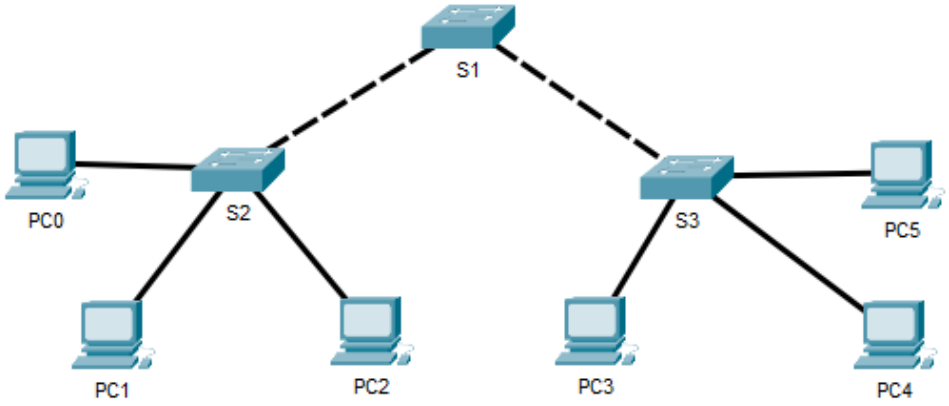
References

Cisco Networking Academy - CCNA: Switching, Routing, and Wireless Essentials.
https://www.netacad.com

Advanced Networks Lab 0907529
Exp3. Scalling VlANs (DTP, VTP, STP)
Lab sheet

Problem 1: Configure VLANs, VTP and DTP

In this activity, you will configure trunk links between the switches. You will configure a VTP server and VTP clients in the same VTP domain. You will also observe the VTP behavior when a switch is in VTP transparent mode. You will assign ports to VLANs and verify end-to-end connectivity with the same VLAN.



Task 1: Configure and Verify DTP

Step 1. Verify VLAN configuration.

Step 2. Configure Trunks on S1, S2, and S3.

Task 2: Configure and Verify VTP

Step 1. Configure S1 as VTP server.

Step 2. Verify VTP on S1.

Step 3. Add S2 and S3 to the VTP domain.

Step 4. Create more VLANs on S1.

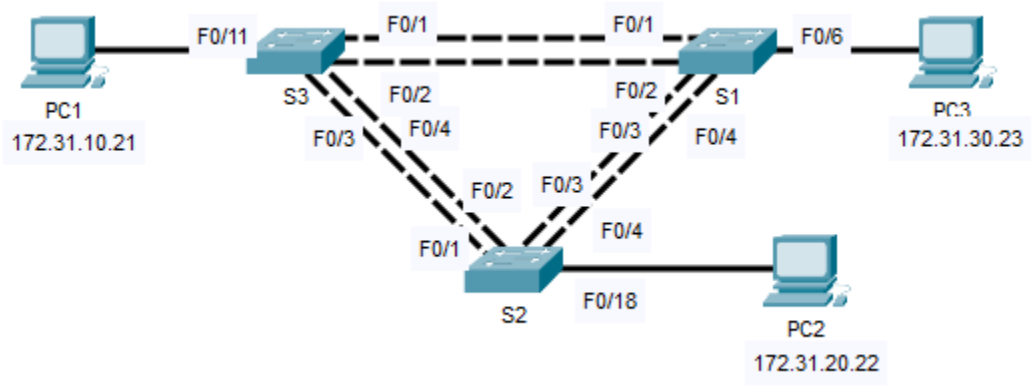
Step 5. Observe VTP transparent mode.

Step 6. Assign VLANs to Ports

Step 7. Verify end to end connectivity.

Problem 2: Configuring PVST+

In this activity, you will configure VLANs and trunks, and examine and configure the Spanning Tree Protocol primary and secondary root bridges. You will also optimize the switched topology using PVST+, PortFast, and BPDU guard.



Task 1: Configure VLANs

Step 1. Enable the user ports on S1, S2, and S3 in access mode.

Step 2. Create VLANs.

Step 3. Assign VLANs to switch ports.

Step 4. Verify the VLANs.

Step 5. Assign the trunks to native VLAN 99.

Task 2: Configure Spanning Tree PVST+ and Load Balancing

Step 1: Configure STP mode.

Step 2. Configure Spanning Tree PVST+ load balancing.

Task 3: Configure PortFast and BPDU Guard

Step 1. Configure PortFast on the switches.

Step 2. Configure BPDU guard on the switches.

Step 3. Verify your configuration.

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.4 Introduction to Wireless LANs

Objectives

1. Review types of wireless networks.
2. Discuss 802.11 standards.
3. Demonstrate WLAN components.
4. Illustrate wireless topology modes.
5. Illustrate the process of wireless client and AP Association.
6. Configure WLAN

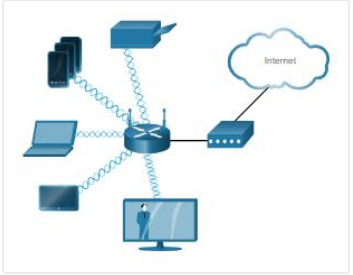
Introduction

A Wireless LAN (WLAN) is a type of wireless network that is commonly used in homes, offices, and campus environments. Networks must support people who are on the move. People connect using computers, laptops, tablets, and smart phones. There are many different network infrastructures that provide network access, such as wired LANs, service provider networks, and cell phone networks. But it's the WLAN that makes mobility possible within the home and business environments.

Types of Wireless Networks

Wireless networks are based on the Institute of Electrical and Electronics Engineers (IEEE) standards and can be classified broadly into four main types: WPAN, WLAN, WMAN, and WWAN.

Wireless Personal-Area Networks (WPAN) - Uses low powered transmitters for a short-range network, usually 20 to 30 ft. (6 to 9 meters). Bluetooth and ZigBee based devices are commonly used in WPANs. WPANs are based on the 802.15 standard and a 2.4-GHz radio frequency.



Wireless LANs (WLAN) - Uses transmitters to cover a medium-sized network, usually up to 300 feet. WLANs are suitable for use in a home, office, and even a campus environment. WLANs are based on the 802.11 standard and a 2.4-GHz or 5-GHz radio frequency.

Wireless MANs (WMAN) - Uses transmitters to provide wireless service over a larger geographic area. WMANs are suitable for providing wireless access to a metropolitan city or specific district. WMANs use specific licensed frequencies.

Wireless Wide-Area Networks (WWANs) - Uses transmitters to provide coverage over an extensive geographic area. WWANs are suitable for national and global communications. WWANs also use specific licensed frequencies.

802.11 Standards

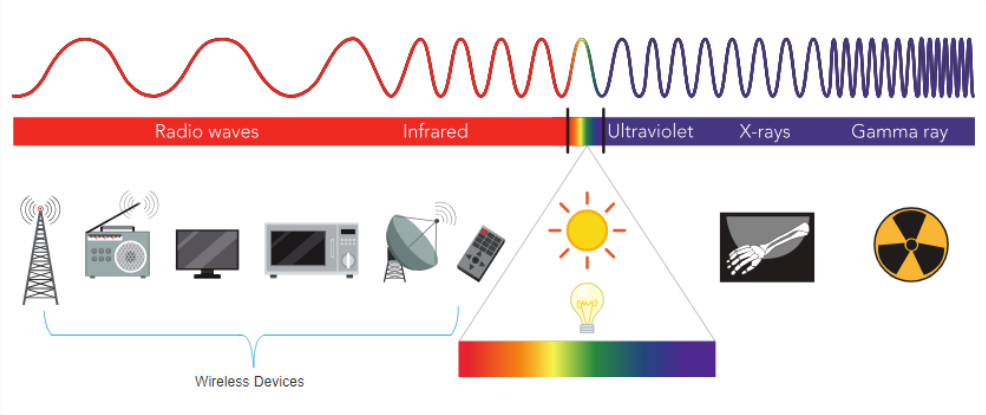
Various implementations of the IEEE 802.11 standard have been developed over the years. The table highlights these standards.

IEEE WLAN Standard	Radio Frequency	Description
802.11	2.4 GHz	• speeds of up to 2 Mbps
802.11a	5 GHz	• speeds of up to 54 Mbps • small coverage area • less effective at penetrating building structures • not interoperable with the 802.11b and 802.11g
802.11b	2.4 GHz	• speeds of up to 11 Mbps • longer range than 802.11a • better able to penetrate building structures
802.11g	2.4 GHz	• speeds of up to 54 Mbps • backward compatible with 802.11b with reduced bandwidth capacity
802.11n	2.4 GHz 5 GHz	• data rates range from 150 Mbps to 600 Mbps with a distance range of up to 70 m (230 feet) • APs and wireless clients require multiple antennas using MIMO technology • backward compatible with 802.11a/b/g devices with limiting data rates
802.11ac	5 GHz	• provides data rates ranging from 450 Mbps to 1.3 Gbps (1300 Mbps) using MIMO technology • Up to eight antennas can be supported • backwards compatible with 802.11a/n devices with limiting data rates
802.11ax	2.4 GHz 5 GHz	• latest standard released in 2019 • also known as Wi-Fi 6 or High-Efficiency Wireless (HEW)

IEEE WLAN Standard	Radio Frequency	Description
		• provides improved power efficiency, higher data rates, increased capacity, and handles many connected devices • currently operates using 2.4 GHz and 5 GHz but will use 1 GHz and 7 GHz when those frequencies become available • Search the internet for Wi-Fi Generation 6 for more information

All wireless devices operate in the radio waves range of the electromagnetic spectrum. WLAN networks operate in the 2.4 GHz frequency band and the 5 GHz band. Wireless LAN devices have transmitters and receivers tuned to specific frequencies of the radio waves range, as shown in the figure. Specifically, the following frequency bands are allocated to 802.11 wireless LANs:

- 2.4 GHz (UHF) - 802.11b/g/n/ax
- 5 GHz (SHF) - 802.11a/n/ac/ax



WLAN Components

The physical WLAN architecture is fairly simple. Basic components of WLAN are typically: network interface cards (NICs) or client adaptors, wireless home routers, and wireless access points. You can use other components, such as wireless bridges and repeaters, to extend the reach of your network.

Wireless NICs - Wireless deployments require a minimum of two devices that have a radio transmitter and a radio receiver tuned to the same radio frequencies:

- End devices with wireless NICs
- A network device, such as a wireless router or wireless AP

To communicate wirelessly, laptops, tablets, smart phones, and even the latest automobiles include integrated wireless NICs that incorporate a radio transmitter/receiver. However, if a device does not have an integrated wireless NIC, then a USB wireless adapter can be used, as shown in the figure.

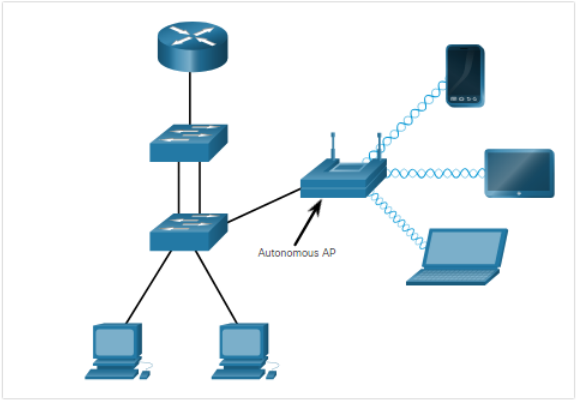
Wireless Home Router - The type of infrastructure device that an end device associates and authenticates with varies based on the size and requirement of the WLAN. For example, a home user typically interconnects wireless devices using a small, wireless router, as shown in the figure. The wireless router serves as an:

- Access point - This provides 802.11a/b/g/n/ac wireless access.
- Switch - This provides a four-port, full-duplex, 10/100/1000 Ethernet switch to interconnect wired devices.
- Router - This provides a default gateway for connecting to other network infrastructures, such as the internet.

Wireless Access Points - While range extenders are easy to set up and configure, the best solution would be to install another wireless access point to provide dedicated wireless access to the user devices. Wireless clients use their wireless NIC to discover nearby APs advertising their SSID. Clients then attempt to associate and authenticate with an AP. After being authenticated, wireless users have access to network resources. The Cisco Meraki Go APs are shown in the figure.

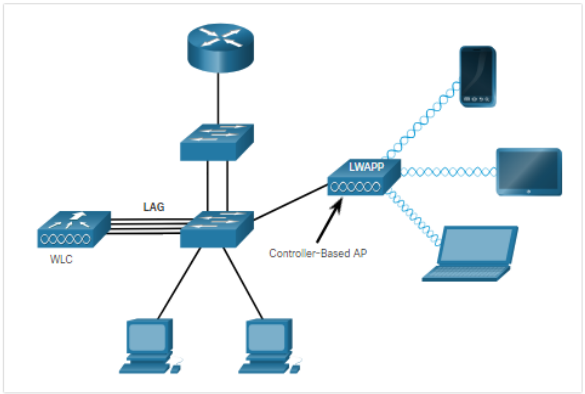
APs can be categorized as either autonomous APs or controller-based APs.

1. Autonomous APs



These are standalone devices configured using a command line interface or a GUI, as shown in the figure. Autonomous APs are useful in situations where only a couple of APs are required in the organization. A home router is an example of an autonomous AP because the entire AP configuration resides on the device. If the wireless demands increase, more APs would be required. Each AP would operate independent of other APs and each AP would require manual configuration and management. This would become overwhelming if many APs were needed.

2. Controller-based APs



These devices require no initial configuration and are often called lightweight APs (LAPs). LAPs use the Lightweight Access Point Protocol (LWAPP) to communicate with a WLAN controller (WLC), as shown in the next figure. Controller-based APs are useful in situations where many APs are required in the network. As more APs are added, each AP is automatically configured and managed by the WLC.

Most business class APs require external antennas to make them fully functioning units.

Omnidirectional antennas such as the one shown in the figure provide 360-degree coverage and are ideal in houses, open office areas, conference rooms, and outside areas.

Directional antennas focus the radio signal in a given direction. This enhances the signal to and from the AP in the direction the antenna is pointing This provides a stronger signal strength in one direction and reduced signal strength in all other directions. Examples of directional Wi-Fi antennas include Yagi and parabolic dish antennas.

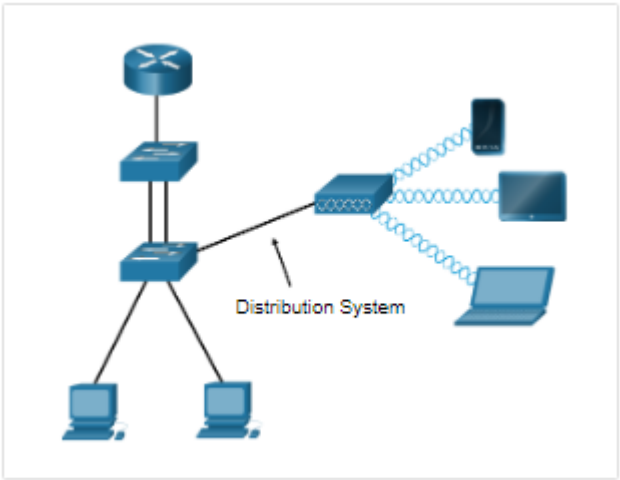
Multiple Input Multiple Output (MIMO) uses multiple antennas to increase available bandwidth for IEEE 802.11n/ac/ax wireless networks. Up to eight transmit and receive antennas can be used to increase throughput.

Wireless Topology Modes

Wireless LANs can accommodate various network topologies. The 802.11 standard identifies two main wireless topology modes: Ad hoc mode and Infrastructure mode.

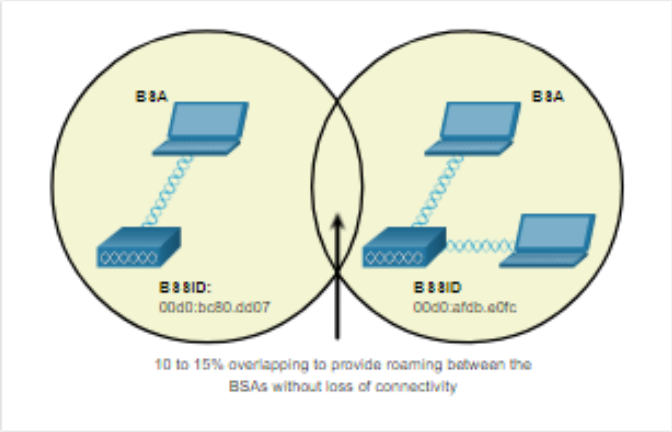
Ad hoc mode - This is when two devices connect wirelessly in a peer-to-peer (P2P) manner without using APs or wireless routers. Examples include wireless clients connecting directly to each other using Bluetooth or Wi-Fi Direct. The IEEE 802.11 standard refers to an ad hoc network as an independent basic service set (IBSS).

Infrastructure mode - This is when wireless clients interconnect via a wireless router or AP, such as in WLANs. APs connect to the network infrastructure using the wired distribution system, such as Ethernet.



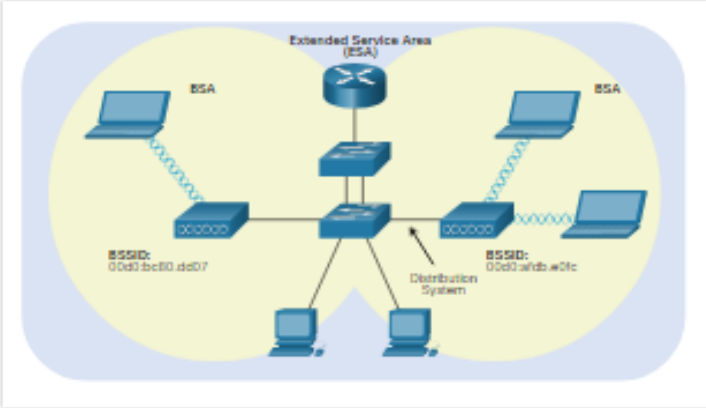
Infrastructure mode defines two topology building blocks: A Basic Service Set (BSS) and an Extended Service Set (ESS).

Basic Service Set - A BSS consists of a single AP interconnecting all associated wireless clients. Two BSSs are shown in the figure. The circles depict the coverage area for the BSS, which is called the Basic Service Area (BSA). If a wireless client moves out of its BSA, it can no longer directly communicate with other wireless clients within the BSA.



The Layer 2 MAC address of the AP is used to uniquely identify each BSS, which is called the Basic Service Set Identifier (BSSID). Therefore, the BSSID is the formal name of the BSS and is always associated with only one AP.

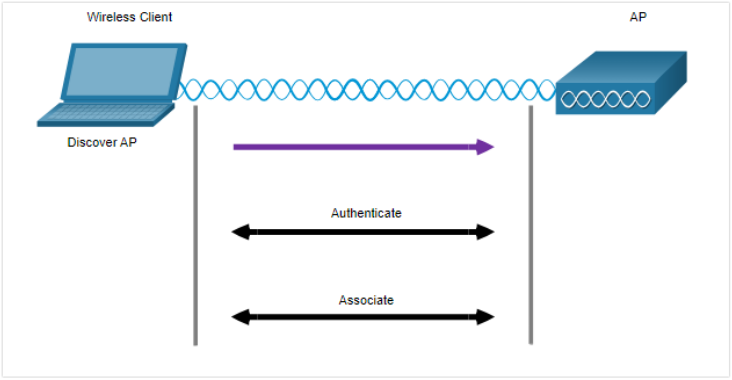
Extended Service Set - When a single BSS provides insufficient coverage, two or more BSSs can be joined through a common distribution system (DS) into an ESS. An ESS is the union of two or more BSSs interconnected by a wired DS. Each ESS is identified by a SSID and each BSS is identified by its BSSID.



Wireless clients in one BSA can now communicate with wireless clients in another BSA within the same ESS. Roaming mobile wireless clients may move from one BSA to another (within the same ESS) and seamlessly connect. The rectangular area in the figure depicts the coverage area within which members of an ESS may communicate. This area is called the Extended Service Area (ESA).

Wireless Client and AP Association

For wireless devices to communicate over a network, they must first associate with an AP or wireless router. An important part of the 802.11 process is discovering a WLAN and subsequently connecting to it. Wireless devices complete the following three stage process, as shown in the figure:



- Discover a wireless AP
- Authenticate with AP
- Associate with AP

In order to have a successful association, a wireless client and an AP must agree on specific parameters. Parameters must then be configured on the AP and subsequently on the client to enable the negotiation of a successful association.

SSID -The SSID name appears in the list of available wireless networks on a client. In larger organizations that use multiple VLANs to segment traffic, each SSID is mapped to one VLAN. Depending on the network configuration, several APs on a network can share a common SSID.

Password - This is required from the wireless client to authenticate to the AP.

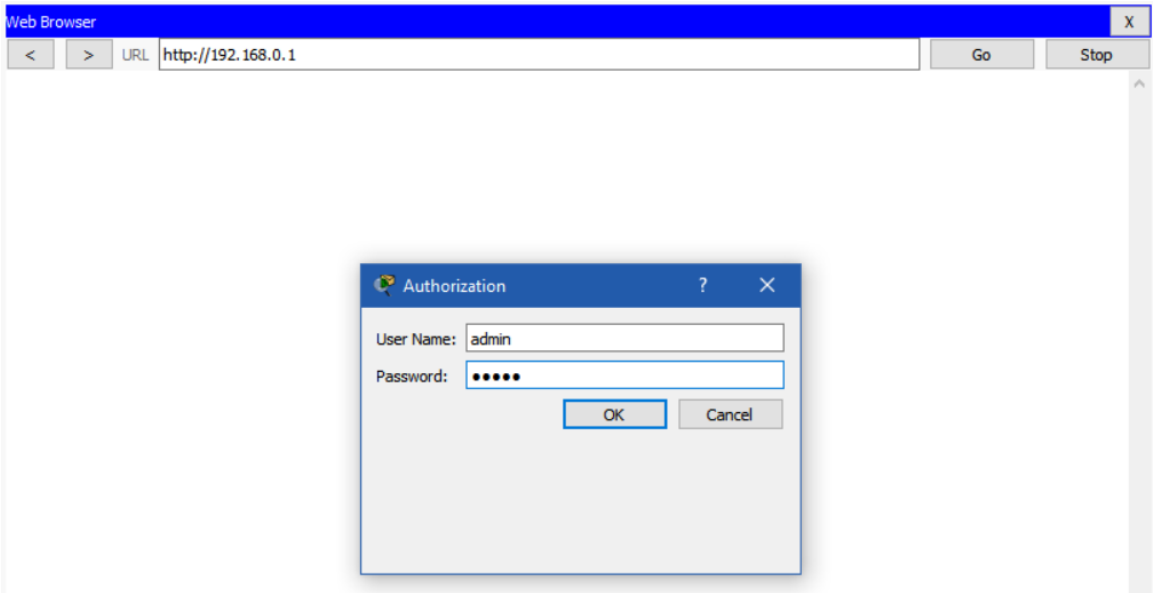
Network mode - This refers to the 802.11a/b/g/n/ac/ad WLAN standards. APs and wireless routers can operate in a Mixed mode meaning that they can simultaneously support clients connecting via multiple standards.

Security mode - This refers to the security parameter settings, such as WEP, WPA, or WPA2. Always enable the highest security level supported.

Channel settings - This refers to the frequency bands used to transmit wireless data. Wireless routers and APs can scan the radio frequency channels and automatically select an appropriate channel setting. The channel can also be set manually if there is interference with another AP or wireless device.

Basic Wireless Configuration

Most wireless routers are ready for service out of the box. They are preconfigured to be connected to the network and provide services. For example, the wireless router uses DHCP to automatically provide addressing information to connected devices. However, wireless router default IP addresses, usernames, and passwords can easily be found on the internet. Just enter the search phrase “default wireless router IP address” or “default wireless router passwords” to see a listing of many websites that provide this information. For example, username and password for the wireless router in the figure is “admin”. Therefore, your first priority should be to change these defaults for security reasons.



To gain access to the wireless router’s configuration GUI, open a web browser. In the address field, enter the default IP address for your wireless router. The default IP address can be found in the documentation that came with the wireless router or you can search the internet. The figure above shows the IPv4 address 192.168.0.1, which is a common default for many manufacturers. A

security window prompts for authorization to access the router GUI. The word admin is commonly used as the default username and password. Again, check your wireless router's documentation or search the internet.

Basic network setup includes the following steps:

1. Log in to the router from a web browser.
2. Change the default administrative password.
3. Log in with the new administrative password.
4. Change the default DHCP IPv4 addresses.
5. Renew the IP address.
6. Log in to the router with the new IP address.

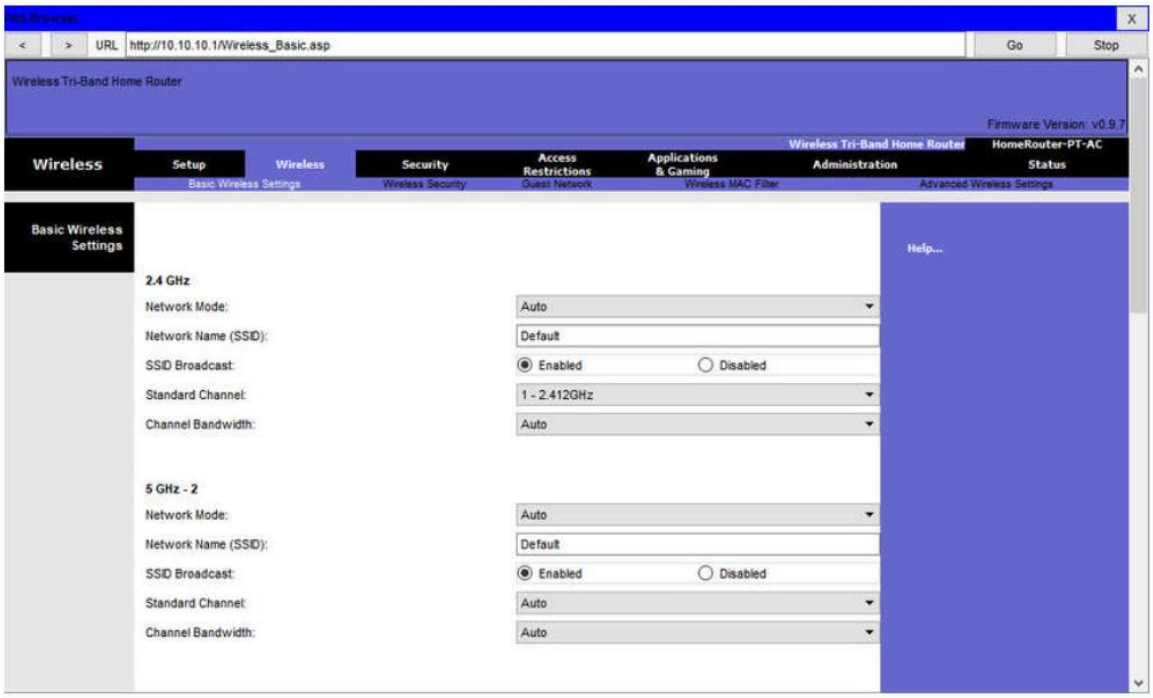
Basic wireless network configuration

Basic wireless configuration includes the following steps:

1. View the WLAN defaults.
2. Change the network mode.
3. Configure the SSID.
4. Configure the channel.
5. Configure the security mode.
6. Configure the passphrase.

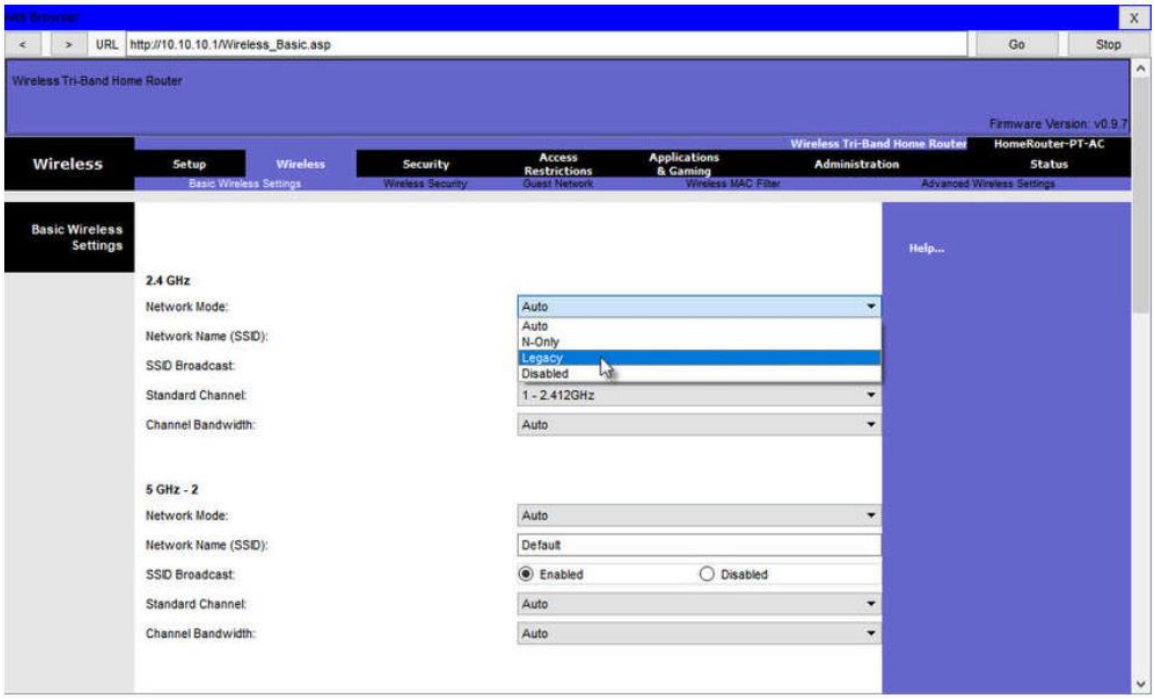
1. View the WLAN defaults.

Out of the box, a wireless router provides wireless access to devices using a default wireless network name and password. The network name is called the Service Set Identified (SSID). Locate the basic wireless settings for your router to change these defaults, as shown in the example.



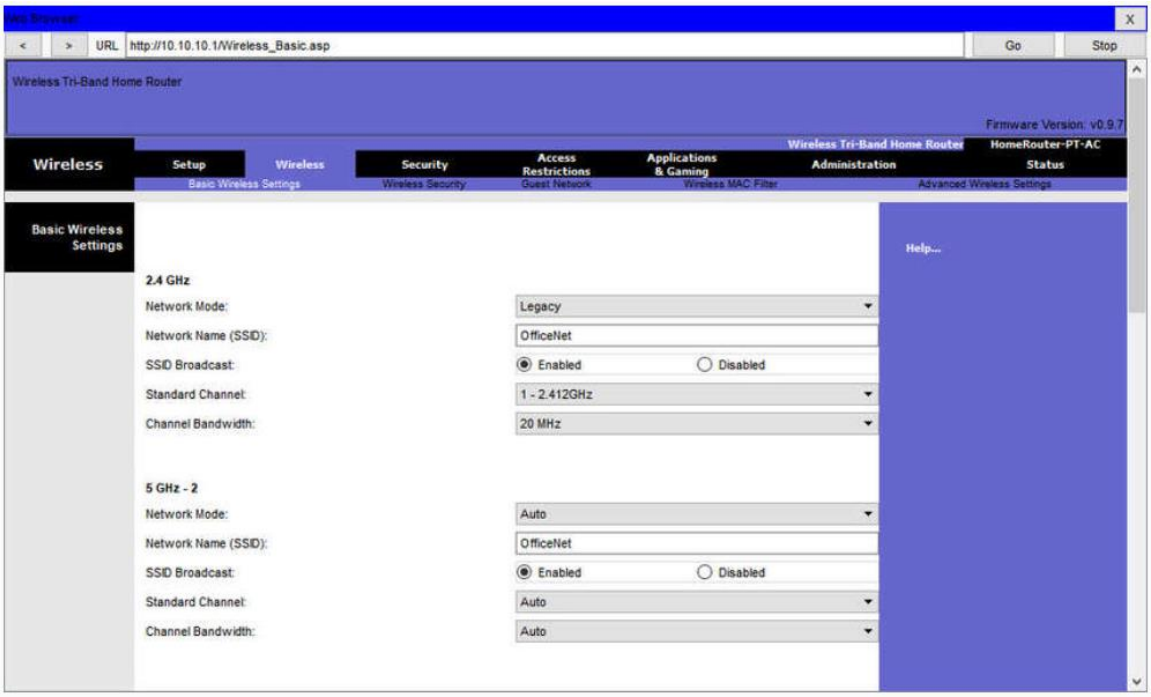
2. Change the network mode.

Some wireless routers allow you to select which 802.11 standard to implement. The example shows that “Legacy” has been selected. This means wireless devices connecting to the wireless router can have a variety of wireless NICs installed. Today’s wireless routers configured for legacy or mixed mode most likely support 802.11a, 802.11n, and 802.11ac NICs.



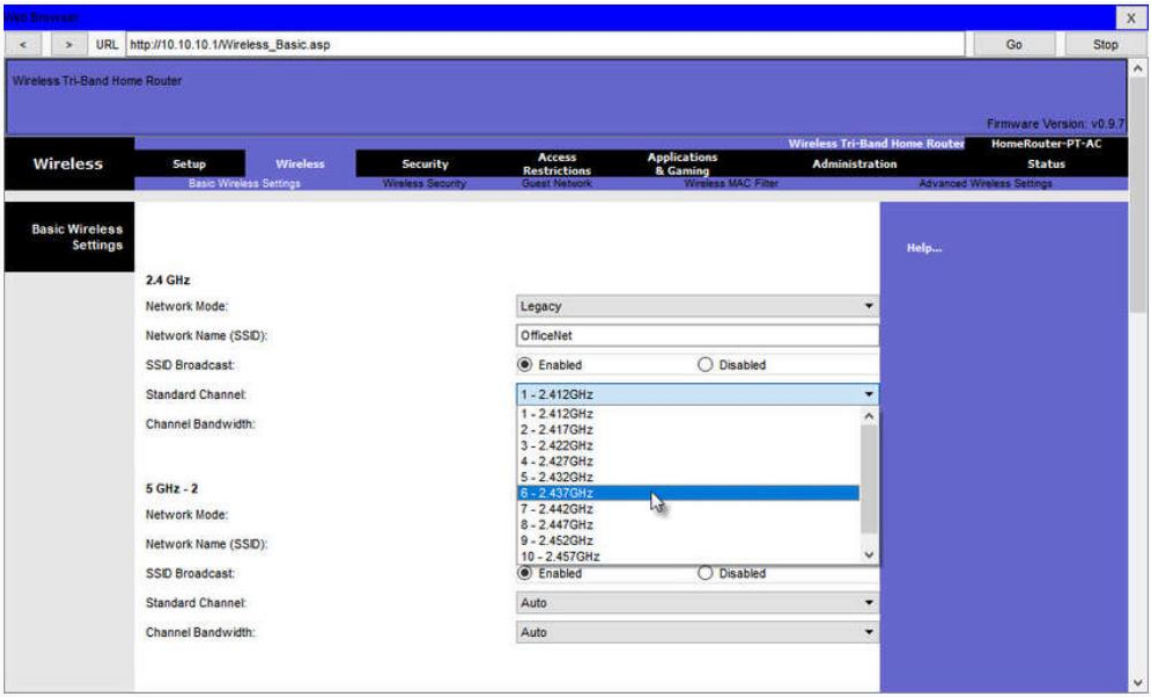
3. Configure the SSID.

Assign an SSID to the WLANs. OfficeNet is used in the example for all three WLANs (the third WLAN is not shown). The wireless router announces its presence by sending broadcasts advertising its SSID. This allows wireless hosts to automatically discover the name of the wireless network. If the SSID broadcast is disabled, you must manually enter the SSID on each wireless device that connects to the WLAN.



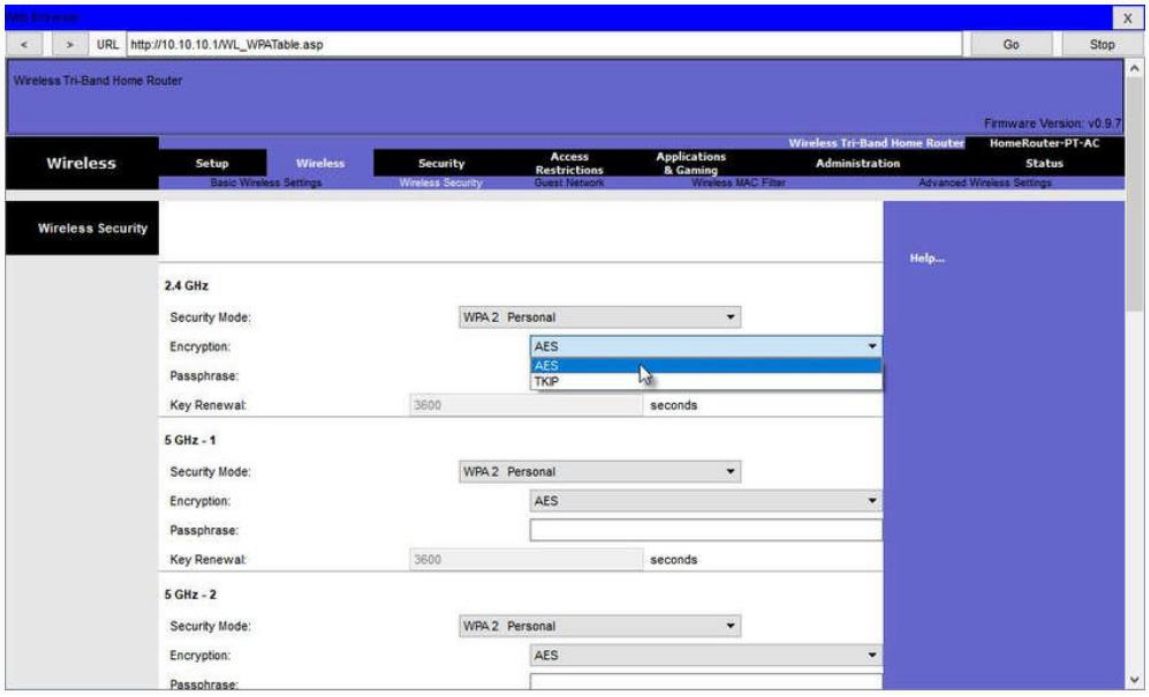
4. Configure the channel.

Devices configured with the same channel within the 2.4GHz band may overlap and cause distortion, slowing down the wireless performance and potentially break network connections. The solution to avoid interference is to configure non-overlapping channels on the wireless routers and access points that are near to each other. Specifically, channels 1, 6, and 11 are non-overlapping. In the example, the wireless router is configured to use channel 6.



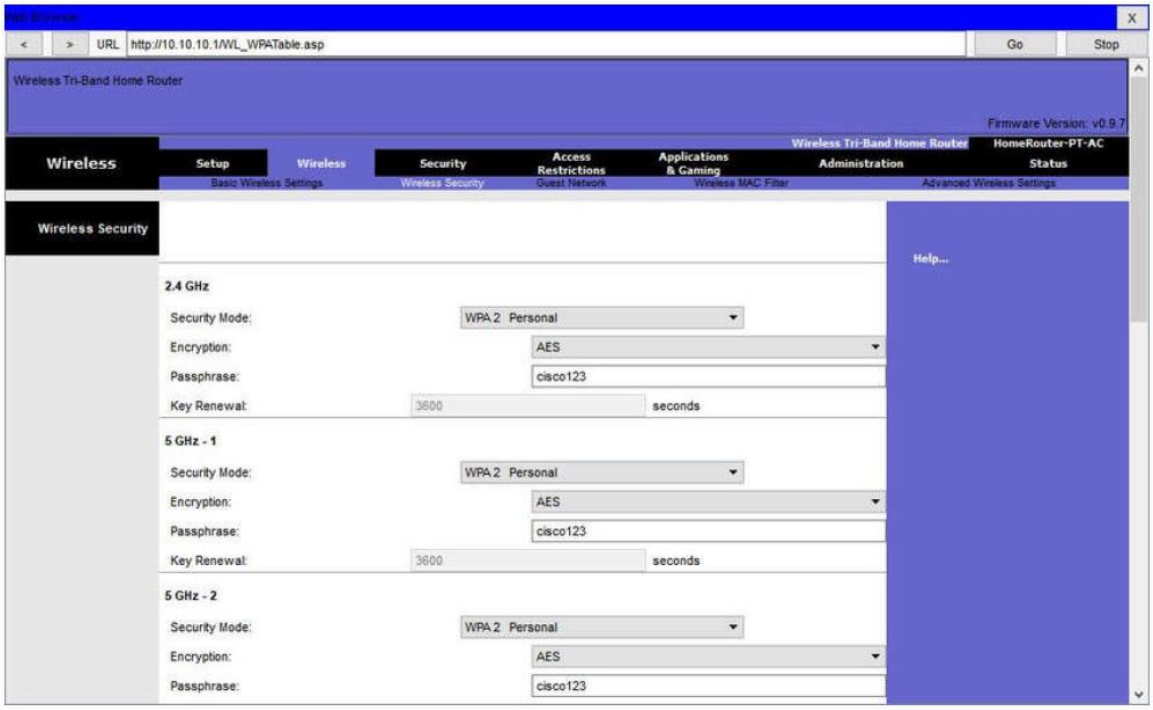
5. Configure the security mode.

Out of the box, a wireless router may have no WLAN security configured. In the example, the personal version of Wi-Fi Protected Access version 2 (WPA2 Personal) is selected for all three WLANs. WPA2 with Advanced Encryption Standard (AES) encryption is currently the strongest security mode.



6. Configure the passphrase.

WPA2 personal uses a passphrase to authenticate wireless clients. WPA2 personal is easier to use in a small office or home environment because it does not require an authentication server. Larger organizations implement WPA2 enterprise and require wireless clients to authenticate with a username and password.



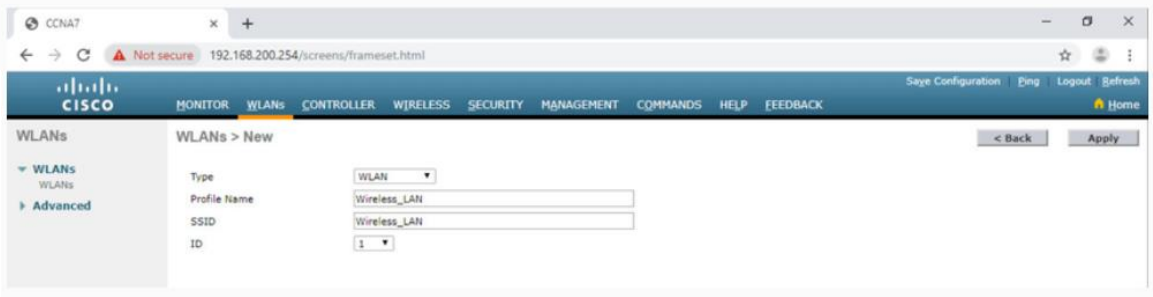
Configure a Basic WLAN on the WLC

Basic WLAN configuration on the WLC includes the following steps:

1. Create the WLAN
2. Apply and Enable the WLAN
3. Select the Interface
4. Secure the WLAN
5. Verify the WLAN is Operational
6. Monitor the WLAN
7. View Wireless Client Information

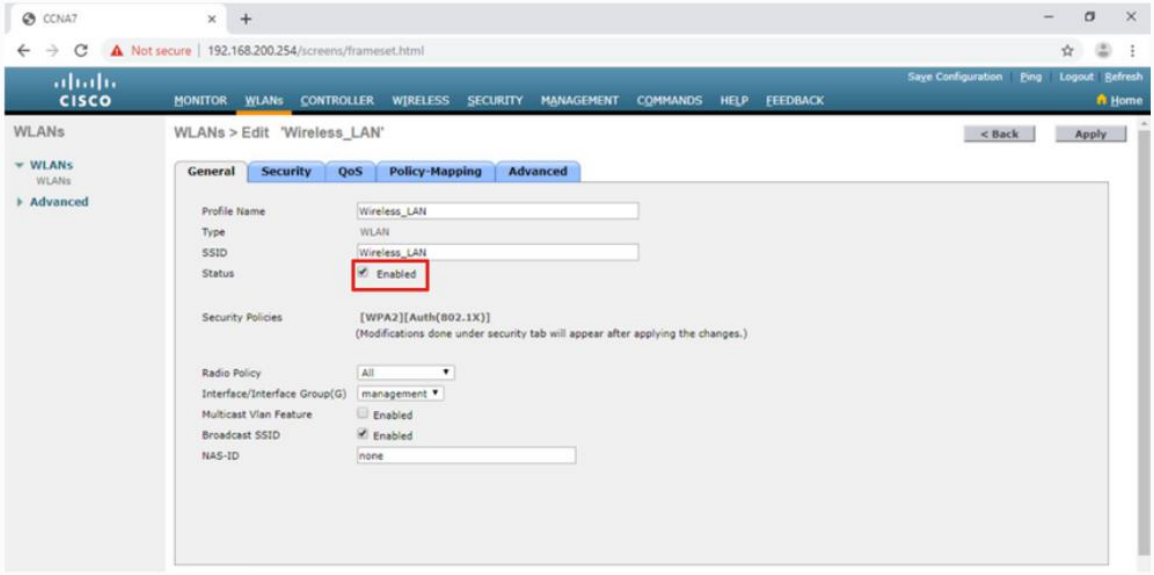
1. Create the WLAN

In the figure, the administrator is creating a new WLAN that will use **Wireless_LAN** as the name and service set identifier (SSID). The ID is an arbitrary value that is used to identify the WLAN in display output on the WLC.



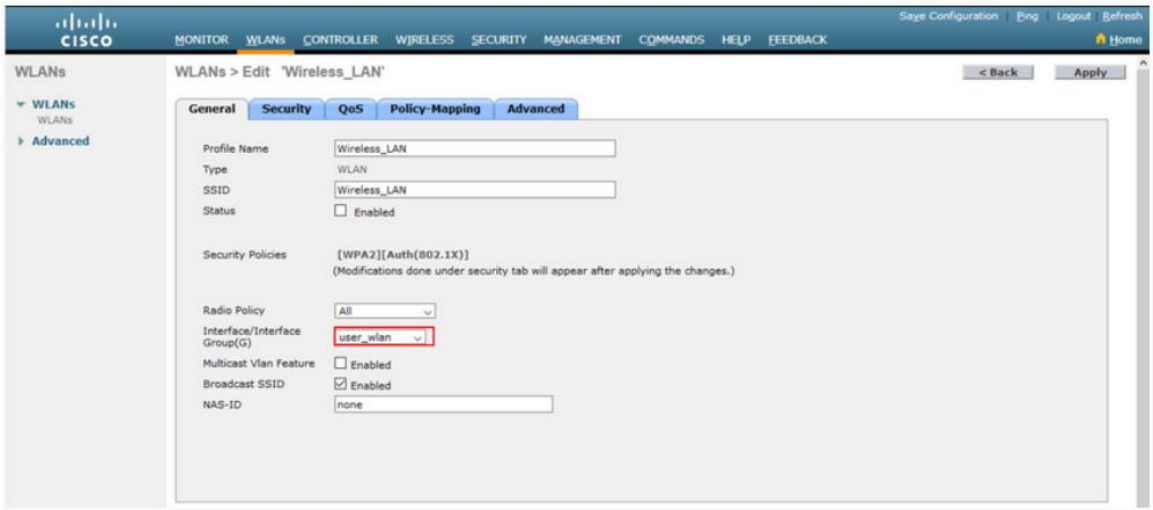
2. Apply and Enable the WLAN

After clicking **Apply**, the network administrator must enable the WLAN before it can be accessed by users, as shown in the figure. The Enable checkbox allows the network administrator to configure a variety of features for the WLAN, as well as additional WLANs, before enabling them for wireless client access. From here, the network administrator can configure a variety of settings for the WLAN including security, QoS, policies, and other advanced settings.



3. Select the Interface

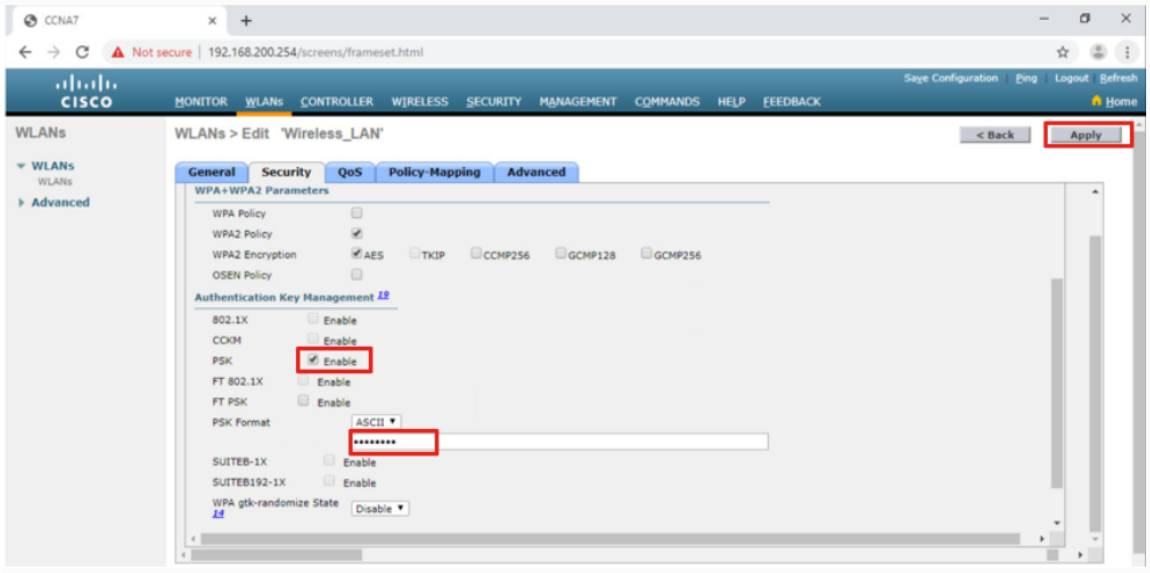
When you create a WLAN, you must select the interface that will carry the WLAN traffic. The next figure shows the selection of an interface that has already been created on the WLC. We will learn how to create interfaces later in this module.



4. Secure the WLAN

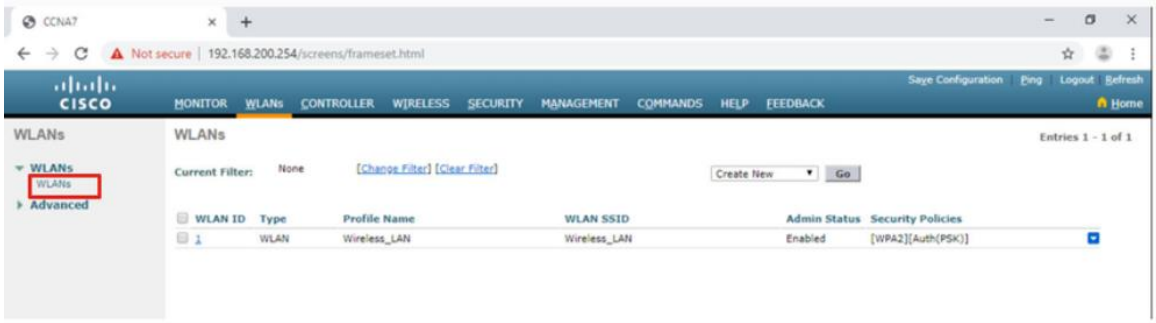
Click the Security tab to access all the available options for securing the LAN. The network administrator wants to secure Layer 2 with WPA2-PSK. WPA2 and 802.1X are set by default. In

the Layer 2 Security drop down box, verify that **WPA+WPA2** is selected (not shown). Click PSK and enter the pre-shared key, as shown in the figure. Then click **Apply**. This will enable the WLAN with WPA2-PSK authentication. Wireless clients that know the pre-shared key can now associate and authenticate with the AP.



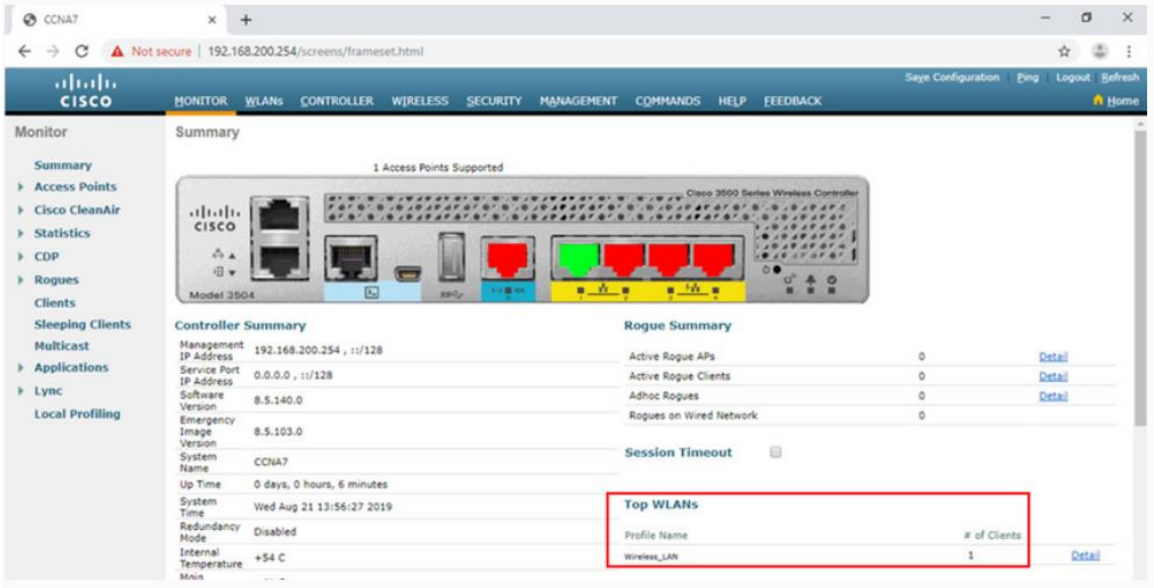
5. Verify the WLAN is Operational

Click **WLANs** in the menu on the left to view the newly configured WLAN. In the figure, you can verify that WLAN ID 1 is configured with **Wireless_LAN** as the name and SSID, it is enabled, and is using WPA2 PSK security.



6. Monitor the WLAN

Click the **Monitor** tab at the top to access the advanced **Summary** page again. Here you can see that the **Wireless_LAN** now has one client using its services, as shown in the figure.



7. View Wireless Client Details

Click **Clients** in the left menu to view more information about the clients connected to the WLAN, as shown in the figure. One client is attached to **Wireless_LAN** through AP1 and was given the IP address 192.168.5.2. DHCP services in this topology are provided by the router.

Procedures:

Dear students, please note that the lab problems sheet, the packet tracer activities and the practical discussion videos have been uploaded on your Microsoft Teams group. You are required to carefully study this experiment and then complete the lab sheet.

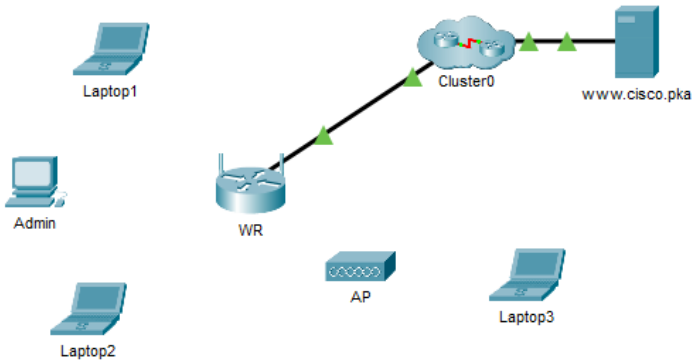
References

Cisco Networking Academy - CCNA: Switching, Routing, and Wireless Essentials.
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.4 Introduction to Wireless LANs
Lab sheet

Problem 1: Configure a Wireless Network

In this activity, you will configure a wireless router and an access point to accept wireless clients and route IP packets. Furthermore, you will also update some of the default settings.



Task 1: Connect to a Wireless Router

Step 1. Connect Admin to WR.

Step 2. Configure Admin to use DHCP.

Step 3. Connect to the WR Web Interface.

Step 4. Configure the Internet Port of WR.

Task 2: Configure the Wireless Settings

Step 1. Configure the WR SSID.

Step 2. Configure wireless security settings.

Step 3. Connect the Wireless Clients.

Task 3: Connect Wireless Clients to an Access Point

Step 1. Configure the Access Point.

Step 2. Connect the Wireless Clients.

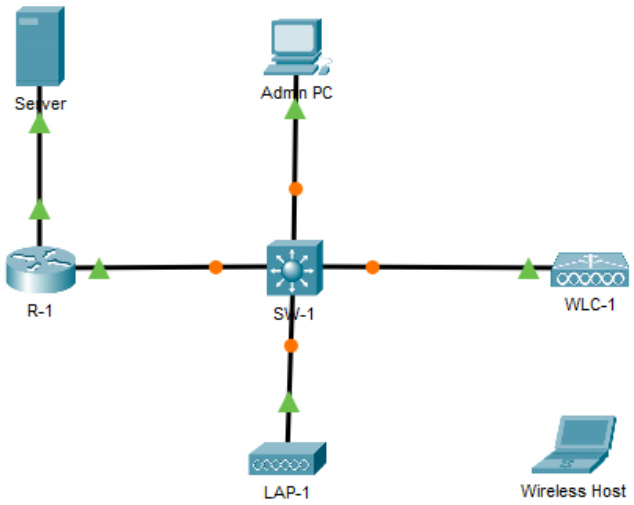
Task 4: Other Administrative Tasks

Step 1. Change the WR Access Password.

Step 2. Change the DHCP address range in WR.

Problem 2: Configure a Basic WLAN on the WLC

In this lab, you will explore some of the features of a wireless LAN controller. You will create a new WLAN on the controller and implement security on that LAN. Then you will configure a wireless host to connect to the new WLAN through an AP that is under the control of the WLC. Finally, you will verify connectivity.



Task 1: Monitor the WLC

Task 2: Create a Wireless LAN

Step 1: Create and enable the WLAN.

Step 2. Secure the WLAN.

Step 3. Verify the Settings

Task 3: Connect a Host to the WLAN

Step 1. Connect to the network and verify connectivity.

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.5 Port Security on Switches

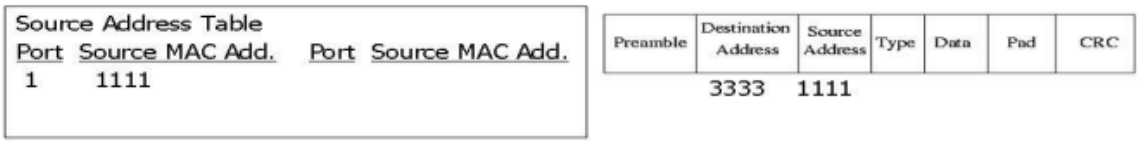
Objectives

1. Illustrate the switch normal operation and examples on switch attacks.
2. Demonstrate a complete port security configuration.
3. Illustrate Port Security Violation Modes.

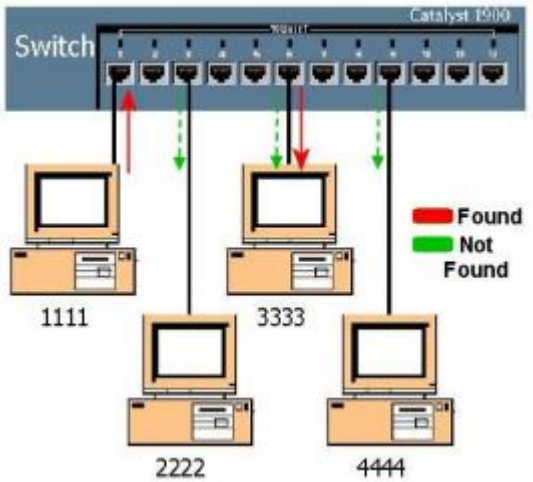
Introduction

An important part of your responsibility as a network professional is to keep the network secure. Most of the time we only think about security attacks coming from outside the network, but threats can come from within the network as well. These threats can range anywhere from an employee innocently adding an Ethernet switch to the corporate network so they can have more ports, to malicious attacks caused by a disgruntled employee. It is your job to keep the network safe and ensuring that business operations continue uncompromised.

The switch or bridge is a layer 2 learning device that examines the MAC address in the layer 2 frames and make the forwarding decisions based in it. Every switch has Content Addressable memory (CAM) table or MAC table that the MAC addresses of all devices that previously communicate via this switch are stored. This table contain the MAC address of the device and the number of switch port that this device of this MAC send its frame as shown in the following figure.



When a certain switch receives an Ethernet frame, the following procedures are followed:
1. The switch searches the MAC address table for the source MAC address. If it finds a match, it resets the aging timer. This timer is used to keep entries up-to-date and remove the expired entries from the MAC table. If the device with a certain MAC address doesn't communicate via the switch for 5-Mintus, the entry of these MAC is removed from the table. If it doesn't find a match, the MAC address is learned as new MAC address and added as a valid address on that port.
2. The switch search the MAC table for the destination MAC address. If it finds a match, it forwards the frame by only sending it out that port. If the destination address is not in the table, the switch acts like a hub and floods it out all ports.



Organizations commonly implement security solutions using routers, firewalls, Intrusion Prevention System (IPSs), and VPN devices from Layer 3 up to Layer 7. Layer 2 LANs are often considered to be a safe and secure environment. However, if Layer 2 is compromised then all layers above it are also affected. The first step in mitigating attacks on the Layer 2 infrastructure is to understand the underlying operation of Layer 2 and the threats posed by the Layer 2 infrastructure.

MAC address table overflow attacks or MAC flooding attacks is a common attack that could be happened by the exploiting the dynamic learning process that used on the switches by default. MAC address tables are limited in size, only 4096 entry is stored. The attacker uses this limitation by sending fake source MAC addresses to the switch, while the switch learned this MACs and store it in the MAC table with the port that were received from, until the switch MAC address table is full. The switch then enters the fail-open mode and starts acting as a hub, and broadcasts packets to all the machines on the network rather than searching in the MAC table for the requested destination MAC address. As a result, the attacker can see all of the frames sent from a victim host to another host without a MAC address table entry. MAC flooding is usually performed using a network attack tool. To mitigate some switch attacks, do the procedures discussed below.

Secure Unused Ports

Layer 2 devices are considered to be the weakest link in a company's security infrastructure. 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. For example, if a Catalyst 2960 switch has 24 ports and there are three Fast Ethernet connections in use, it is good practice to disable the 21 unused ports. 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
```

For example, to shutdown ports for Fa0/8 through Fa0/24 on S1, you would enter the following command.

```
S1(config)# interface range fa0/8 - 24
S1(config-if-range)# shutdown
%LINK-5-CHANGED: Interface FastEthernet0/8, changed state to administratively down
(output omitted)
%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to administratively down
S1(config-if-range)#
```

Enable Port Security

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.

```
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#
```

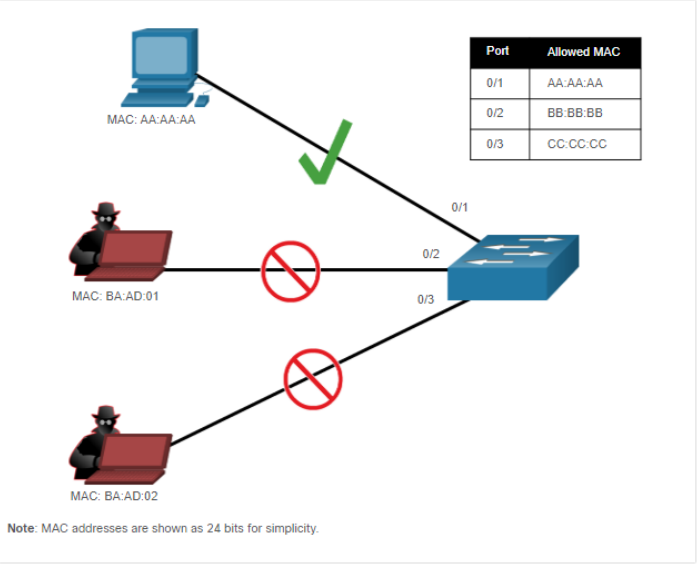
Use the **show port-security interface** command to display the current port security settings for FastEthernet 0/1, as shown in the example. Notice how port security is enabled, port status is Secure-down which means there are no devices attached and no violation has occurred, the violation mode is Shutdown, and how the maximum number of MAC addresses is 1. If a device is connected to the port, the switch port status would display Secure-up and the switch will automatically add the device's MAC address as a secure MAC. In this example, no device is connected to the port.

```
S1# show port-security interface f0/1
Port Security              : Enabled
Port Status                : Secure-down
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#
```

Limit and Learn MAC Addresses

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, as shown in the figure.



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
```

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 startup 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.

The following example demonstrates a complete port security configuration for FastEthernet 0/1 with a host connected to port Fa0/1. The administrator specifies a maximum of 2 MAC addresses, manually configures one secure MAC address, and then configures the port to dynamically learn additional secure MAC addresses up to the 2 secure MAC address maximum. Use the **show port-security interface** and the **show port-security address** command to verify the configuration. The output of the **show port-security interface** command verifies that port security is enabled, there is a host connected to the port (i.e., Secure-up), a total of 2 MAC addresses will be allowed, and S1 has learned one MAC address statically and one MAC address dynamically (i.e., sticky). The output of the **show port-security address** command lists the two learned MAC addresses.

```
S1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)#
S1(config)# interface fa0/1
S1(config-if)# switchport mode access
S1(config-if)# switchport port-security
S1(config-if)# switchport port-security maximum 2
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      : 2
Total MAC Addresses        : 2
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : a41f.7272.676a:1
Security Violation Count   : 0
S1# show port-security address
              Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                          Ports   Remaining Age
                                                                    (mins)
----    -----------       ----                          -----   -------------
1    a41f.7272.676a    SecureSticky                Fa0/1       -
1    aaaa.bbbb.1234    SecureConfigured            Fa0/1       -
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 1
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. 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 only if they are inactive for the specified aging time.
Use aging to remove secure MAC addresses on a secure port without manually deleting the existing secure MAC addresses. Aging time limits can also be increased to ensure past secure MAC addresses remain, even while new MAC addresses are added. Aging of statically configured secure addresses can be enabled or disabled on a per-port basis.

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.

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

The parameters for the command are described in the table.

Parameter	Description
static	Enable aging for statically configured secure addresses on this port.
time *time*	Specify the aging time for this port. The range is 0 to 1440 minutes. If the time is 0, aging is disabled for this port.
type absolute	Set the absolute aging time. All the secure addresses on this port age out exactly after the time (in minutes) specified and are removed from the secure address list.
type inactivity	Set the inactivity aging type. The secure addresses on this port age out only if there is no data traffic from the secure source address for the specified time period.

Port Security Violation Modes

If the MAC address of a device attached to the port differs from the list of secure addresses, then a port violation occurs. By default, 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 { protect | restrict | shutdown}
```

The following tables show 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.

Violation Mode	Discards Offending Traffic	Sends Syslog Message	Increase Violation Counter	Shuts Down Port
Protect	Yes	No	No	No
Restrict	Yes	Yes	Yes	No
Shutdown	Yes	Yes	Yes	Yes

What happens when the port security violation is shutdown and a port violation occurs? The port is physically shutdown and placed in the error-disabled state, and no traffic is sent or received on that port.

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

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 instead of Secure-up. The Security Violation counter increments by 1.

```
S1# show interface fa0/1 | include down
FastEthernet0/18 is down, line protocol is down (err-disabled)
(output omitted)
S1# show port-security interface fa0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 10 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 2
Total MAC Addresses        : 2
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : a41f.7273.018c:1
Security Violation Count   : 1
S1#
```

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.

In the next example, the first host is reconnected to Fa0/1. To re-enable the port, first use the **shutdown** command, then, use the **no shutdown** command to make the port operational, as shown in the example.

Procedures:

Dear students, please note that the lab problems sheet, the packet tracer activities and the practical discussion videos have been uploaded on your Microsoft Teams group. You are required to carefully study this experiment and then complete the lab sheet.

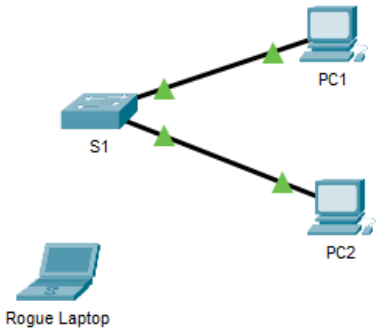
References

Cisco Networking Academy - CCNA: Switching, Routing, and Wireless Essentials.
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.5 Port Security on Switches
Lab sheet

Problem 1: Configuring Switch Port Security

In this activity, you will configure and verify port security on a switch. Port security allows you to restrict a port's ingress traffic by limiting the MAC addresses that are allowed to send traffic into the port.

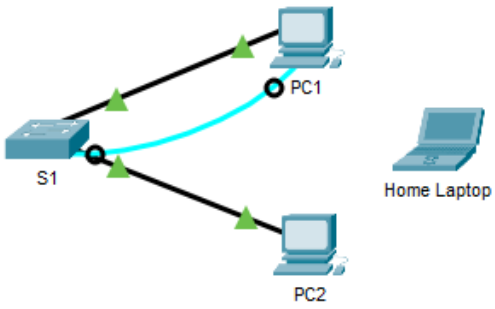


Task 1: Configure Port Security

Task 2: Verify Port Security

Problem 2: Troubleshooting Switch Port Security

In this activity, the employee who normally uses PC1 brought his laptop from home, disconnected PC1 and connected the laptop to the telecommunication outlet. After reminding him of the security policy that does not allow personal devices on the network, you now must reconnect PC1 and re-enable the port.



Task 1: Disconnect Home Laptop and reconnect PC1 to the appropriate port.

Task 2: Enable the port using the necessary command.

Task 3: Verify connectivity. PC1 should now be able to ping PC2.

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.6 Access Control Lists (ACLs)

Objectives

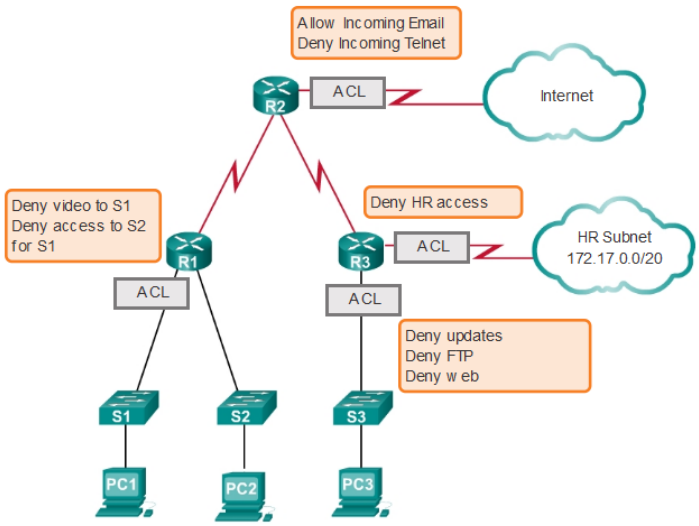
1. Explain how ACLs are used to filter traffic.
2. Compare standard and extended IPv4 ACLs.
3. Explain the guidelines for creating and placement of ACLs.
4. Modify ACLs.
5. Explain how a router processes packets when an ACL is applied.
6. Troubleshoot common ACL errors.

Purpose of ACLs:

Network security is a huge subject, and one of the most important skills a network administrator needs is mastery of access control lists (ACLs).
Network designers use firewalls to protect networks from unauthorized use. Firewalls are hardware or software solutions that enforce network security policies. Consider a lock on a door to a room inside a building. The lock allows only authorized users with a key or access card to pass through the door. Similarly, a firewall filters unauthorized or potentially dangerous packets from entering the network. On a Cisco router, you can configure a simple firewall that provides basic traffic filtering capabilities using ACLs. Administrators use ACLs to stop traffic or permit only specified traffic on their networks.

An ACL is a sequential list of permit or deny statements that apply to addresses or upper-layer protocols based on information found in the packet header. ACLs provide a powerful way to control traffic into and out of a network. ACLs can be configured for all routed network protocols.
When configured, ACLs perform the following tasks as shown in the figure:

- Limit network traffic to increase network performance. For example, if corporate policy does not allow video traffic on the network, ACLs that block video traffic could be configured and applied. This would greatly reduce the network load and increase network performance.
- Provide traffic flow control. ACLs can restrict the delivery of routing updates. If updates are not required because of network conditions, bandwidth is preserved.
- Provide a basic level of security for network access. ACLs can allow one host to access a part of the network and prevent another host from accessing the same area. For example, access to the Human Resources network can be restricted to authorized users.
- Filter traffic based on traffic type. For example, an ACL can permit email traffic, but block all Telnet traffic.
- Screen hosts to permit or deny access to network services. ACLs can permit or deny a user to access file types, such as FTP or HTTP.



By default, a router does not have ACLs configured; therefore, by default a router does not filter traffic. Traffic that enters the router is routed solely based on information within the routing table. However, 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 is a sequential list of permit or deny statements, known as access control entries (ACEs). ACEs are also commonly called ACL statements. ACEs can be created to filter traffic based on certain criteria such as: the source address, destination address, the protocol, and port numbers. 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 statements. If a match is found, the packet is processed accordingly. In this way, ACLs can be configured to control access to a network or subnet.

To evaluate network traffic, the ACL extracts the following information from the Layer 3 packet header:

- Source IP address
- Destination IP address
- ICMP message type

The ACL can also extract upper layer information from the Layer 4 header, including:

- TCP/UDP source port
- TCP/UDP destination port

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

ACLs are configured to apply to inbound traffic or to apply to outbound traffic as shown in the figure.

- Inbound ACLs - Incoming packets are processed 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. If the packet is permitted by the tests, it is then processed for routing. Inbound ACLs are best used to filter packets when the network attached to an inbound interface is the only source of the packets needed to be examined.
- Outbound ACLs - Incoming packets are routed to the outbound interface, and then they are processed through the outbound ACL. Outbound ACLs are best used when the same filter will be applied to packets coming from multiple inbound interfaces before exiting the same outbound interface.

The last statement of an ACL is always an implicit deny. This statement is automatically inserted at the end of each ACL even though it is not physically present. The implicit deny blocks all traffic. Because of this implicit deny, an ACL that does not have at least one permit statement will block all traffic.

Standard versus Extended IPv4 ACLs:

The two types of Cisco IPv4 ACLs are standard and extended. Standard ACLs can be used to permit or deny traffic only from source IPv4 addresses. The destination of the packet and the ports involved are not evaluated. The example below allows all traffic from the 192.168.30.0/24 network. Because of the implied "deny any" at the end, all other traffic is blocked with this ACL. Standard ACLs are created in global configuration mode.

R1(config)# access-list 10 permit 192.168.30.0 0.0.0.255

Extended ACLs filter IPv4 packets based on several attributes:

- Protocol type
- Source IPv4 address
- Destination IPv4 address
- Source TCP or UDP ports
- Destination TCP or UDP ports

In the example below, ACL 103 permits traffic originating from any address on the 192.168.30.0/24 network to any IPv4 network if the destination host port is 80 (HTTP). Extended ACLs are created in global configuration mode.
R1(config)# access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq80

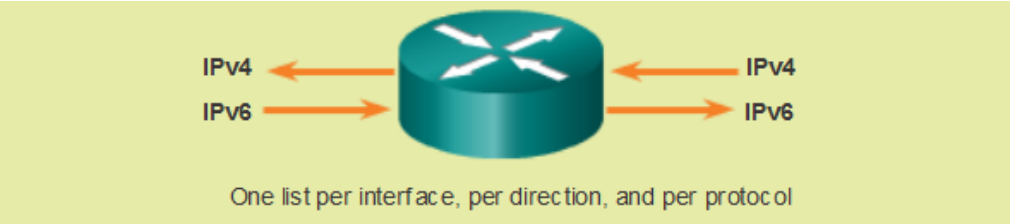
The commands for standard and extended ACLs are explained in more details later in this experiment.

Standard and extended ACLs can be created using either a number or a name to identify the ACL and its list of statements. Using numbered ACLs is an effective method for determining the ACL type on smaller networks with more homogeneously defined traffic. However, a number does not provide information about the purpose of the ACL.

Numbered ACL: Assign a number based on protocol to be filtered.
- (1 to 99) and (1300 to 1999): Standard IP ACL
- (100 to 199) and (2000 to 2699): Extended IP ACL

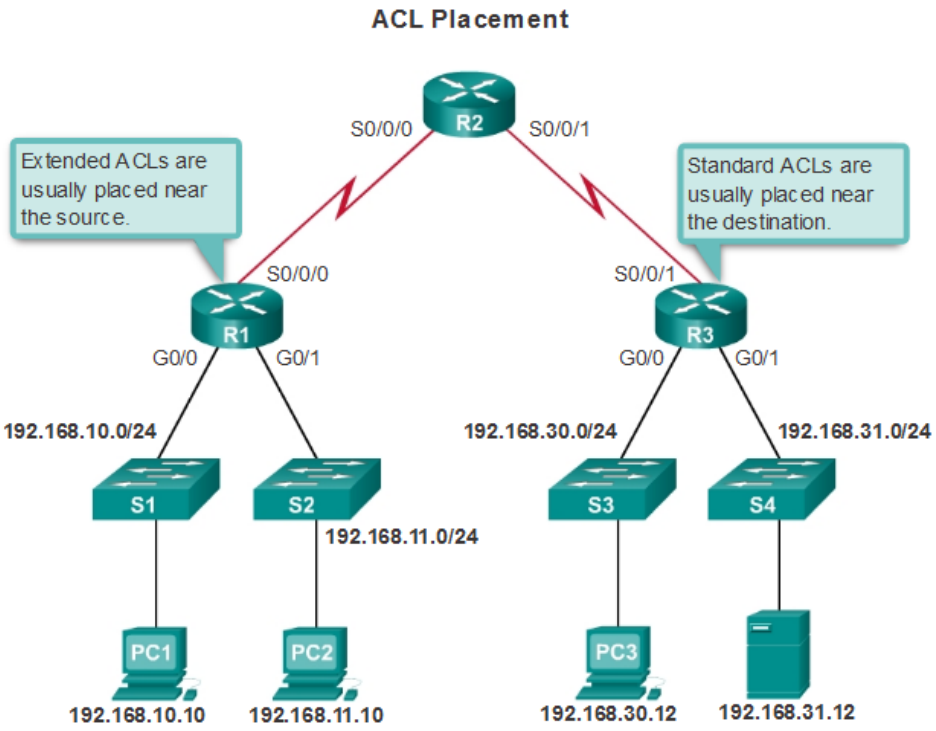
Named ACL: Assign a name to identify the ACL.
- Names can contain alphanumeric characters.
- It is suggested that the name be written in CAPITAL LETTERS.
- Names cannot contain spaces or punctuation.
- Entries can be added or deleted within the ACL.

Writing ACLs can be a complex task. For every interface there may be multiple policies needed to manage the type of traffic allowed to enter or exit that interface. The router in the figure has two interfaces configured for IPv4 and IPv6. If we needed ACLs for both protocols, on both interfaces and in both directions, this would require eight separate ACLs. Each interface would have four ACLs; two ACLs for IPv4 and two ACLs for IPv6. For each protocol, one ACL is for inbound traffic and one for outbound traffic.
The Three Ps:
A general rule for applying ACLs on a router can be recalled by remembering the three Ps. You can configure one ACL per protocol, per direction, per interface:
- One ACL per protocol - To control traffic flow on an interface, an ACL must be defined for each protocol enabled on the interface.
- One ACL per direction - ACLs control traffic in one direction at a time on an interface. Two separate ACLs must be created to control inbound and outbound traffic.
- One ACL per interface - ACLs control traffic for an interface, for example, GigabitEthernet 0/0.



The proper placement of an ACL can make the network operate more efficiently. An ACL can be placed to reduce unnecessary traffic. For example, traffic that will be denied at a remote destination should not be forwarded using network resources along the route to that destination.
Every ACL should be placed where it has the greatest impact on efficiency. As shown in the figure below, the basic rules are:

- **Extended ACLs** - Locate extended ACLs as close as possible to the source of the traffic to be filtered. This way, undesirable traffic is denied close to the source network without crossing the network infrastructure.
- **Standard ACLs** - Because standard ACLs do not specify destination addresses, place them as close to the destination as possible. Placing a standard ACL at the source of the traffic will effectively prevent that traffic from reaching any other networks through the interface where the ACL is applied.



Wildcard Masks in ACLs

IPv4 ACEs include the use of wildcard masks. A wildcard mask is a string of 32 binary digits used by the router to determine which bits of the address to examine for a match.

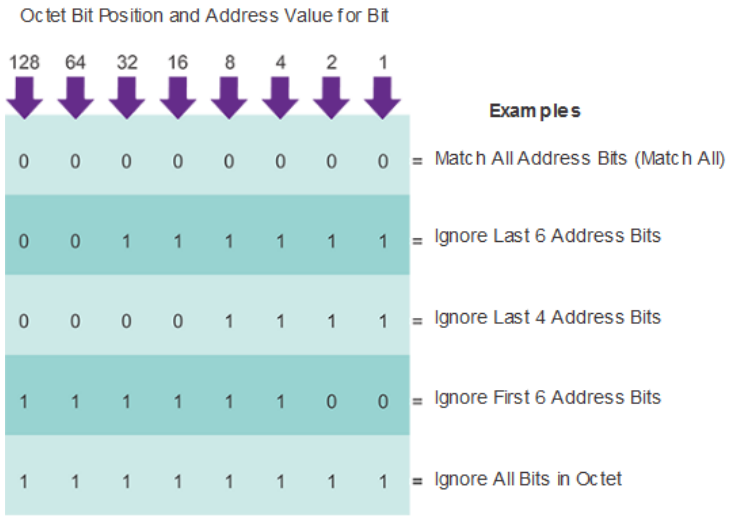
Subnet masks use binary 1s and 0s to identify the network, subnet, and host portion of an IP address. Wildcard masks use binary 1s and 0s to filter individual IP addresses or groups of IP addresses to permit or deny access to resources.

Wildcard masks and subnet masks differ in the way they match binary 1s and 0s. 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.

The following is an example on how to deal with wildcard masks:

	Decimal Address	Binary Address
IP Address to be processed	192.168.10.0	11000000.10101000.00001010.00000000
Wild Mask	0.0.255.255	00000000. 00000000.11111111. 11111111
Resulting IP Address	192.168.0.0	11000000.10101000. 00000000. 00000000



The ***any*** and *host* keywords: ***any*** keyword to substitute for the IPv4 address 0.0.0.0 with a wildcard mask of 255.255.255.255 as shown in the example below:
R1(config)# access-list 1 permit 0.0.0.0 255.255. 255.255
R1(config)# access-list 1 permit any
The ***host*** keyword to substitute for the wildcard mask when identifying a single host as shown in the example below.
R1(config)# access-list 1 permit 192.168.10.10 0.0.0.0
R1(config)# access-list 1 permit host 192.168.10.10

ACL Creation and Placement:

Configuring Standard ACLs:

To use numbered standard ACLs on a Cisco router, you must first create the standard ACL and then activate the ACL on an interface. The access-list global configuration command defines a standard ACL with a number in the range of 1 through 99. The full syntax of the standard ACL command is as follows:

Router(config)# **access-list** *access-list-number* { **deny** | **permit** } *source* [*source-wildcard*]

ACEs can deny or permit an individual host or a range of host addresses. If you want to remove the ACL, the global configuration **no access-list** command is used. Issuing the **show access-list** command confirms that access list 10 has been removed.

After a standard ACL is configured, it is linked to an interface using the **ip access-group** command in interface configuration mode:

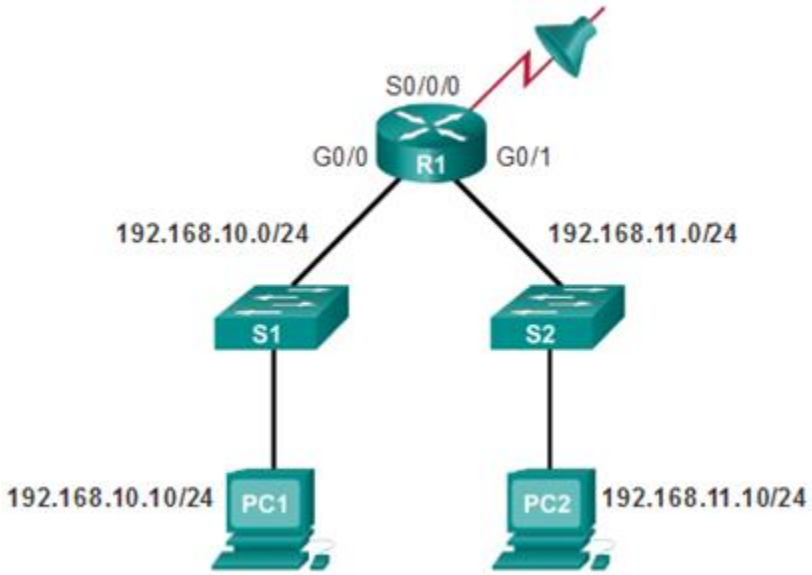
Router(config-if)# **ip access-group** { *access-list-number* | *access-list-name* } { **in** | **out** }

To remove an ACL from an interface, first enter the **no ip access-group** command on the interface, and then enter the global **no access-list** command to remove the entire ACL.

To create a statement that will permit a range of IPv4 addresses in a numbered ACL 10 that permits all IPv4 addresses in the network 192.168.10.0/24, you would enter:
R1(config)# access-list 10 permit 192.168.10.10 0.0.0.255

To apply a numbered standard ACL 10 on a router, you would enter:

R1(config)#interface serial 0/0/0
R1(config-if)# ip access-group 10 out



This ACL allows only traffic from source network 192.168.10.0 to be forwarded out of interface S0/0/0. Traffic from networks other than 192.168.10.0 is blocked.

Naming an ACL makes it easier to understand its function. For example, the above ACL could be called ***VLAN10_ALLOW***. When you identify your ACL with a name instead of with a number, the configuration mode and command syntax are slightly different as shown below.
R1(config)# ip access-list standard VLAN10_ALLOW
R1(config-std-nacl)# permit 192.168.10.10 0.0.0.255
R1(config-std-nacl)#exit
R1(config)#interface serial 0/0/0
R1(config-if)# ip access-group VLAN10_ALLOW out

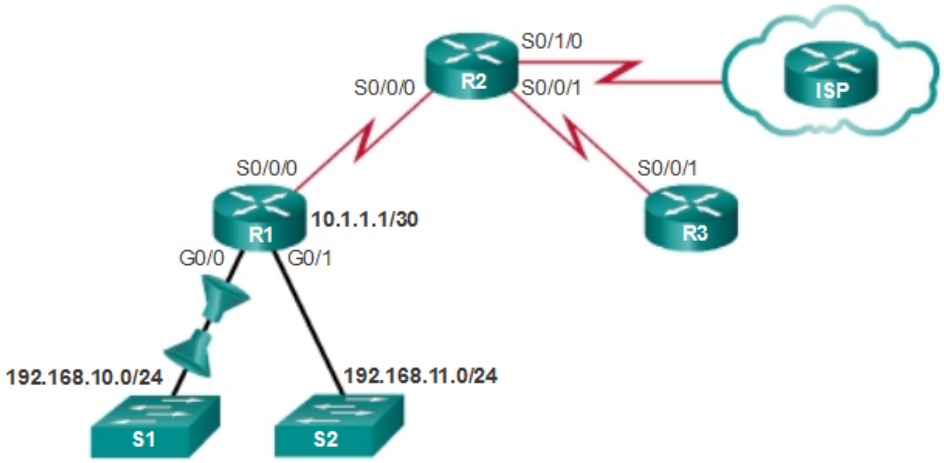
Using an ACL to Control VTY Access:
Restricting VTY access is a technique that allows you to define which IP addresses are allowed Telnet access to the router EXEC process. You can control which administrative workstation or network manages your router with an ACL and an **access-class** statement configured on your VTY lines.
An example allowing a range of addresses to access VTY lines 0 - 4 is shown below.
Configure the vty lines to accept incoming telnet connections using access list 21.
R1(config)# line vty 0 4
R1(config-line)# access-class 21 in
Create access list 21 to permit the 192.168.10.0/24 network to access VTY lines 0 - 4 and explicitly deny all others.
R1(config)# access-list 21 permit 192.168.10.0 0.0.0.255
R1(config)# access-list 21 deny any

Configuring Extended ACLs:
The procedural steps for configuring extended ACLs are the same as for standard ACLs. The extended ACL is first configured, and then it is activated on an interface. However, the command syntax and parameters are more complex to support the additional features provided by extended ACLs. The common command syntax for extended IPv4 ACLs is shown below. Note that there are many keywords and parameters for extended ACLs. It is not necessary to use all of the keywords and parameters when configuring an extended ACL.

Router(config)# ***access-list*** *access-list-number { **deny** | **permit** } protocol source [source-wildcard] [**port** port-number or name] destination [destination-wildcard] [**port** port-number or name] [**established**]*

The following is an example of an extended ACL. In this example, the network administrator has configured ACLs to restrict network access to allow website browsing only from the LAN attached to interface G0/0 to any external network. ACL 103 allows traffic coming from any address on the 192.168.10.0 network to go to any destination, subject to the limitation that the traffic is using ports 80 (HTTP) and 443 (HTTPS) only so that 192.168.10.0/24 network to browse both insecure and secure websites.

The nature of HTTP requires that traffic flow back into the network from websites accessed from internal clients. The network administrator wants to restrict that return traffic to HTTP exchanges from requested websites, while denying all other traffic. ACL 104 does that by blocking all incoming traffic, except for previously established connections. The permit statement in ACL 104 allows inbound traffic using the **established** parameter. Without the **established** parameter in the ACL statement, clients could send traffic to a web server, but not receive traffic returning from the web server.

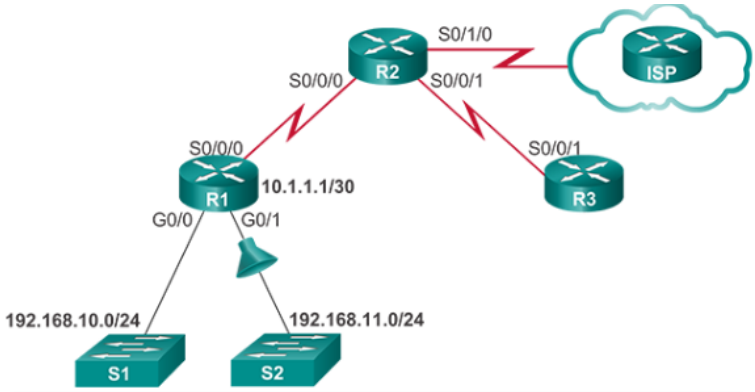



```
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)#access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)#interface g0/0
R1(config-if)#ip access-group 103 in
R1(config-if)#ip access-group 104 out
```

- ACL 103 allows requests to ports 80 and 443.
- ACL 104 allows established HTTP and HTTPS replies.

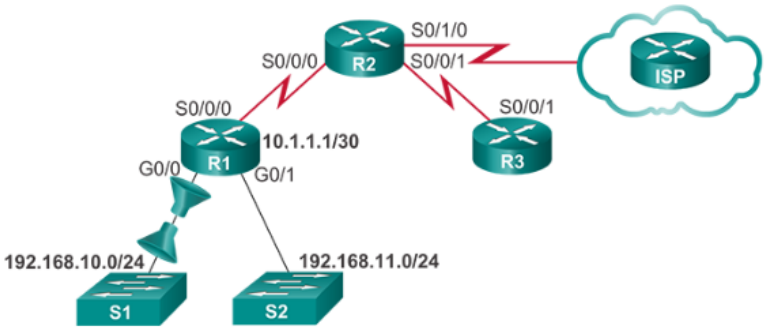
The example shown in the figure below denies FTP traffic from subnet 192.168.11.0 that is going to subnet 192.168.10.0, but permits all other traffic. Note the use of wildcard masks and the explicit deny any statement. Remember that FTP uses TCP ports 20 and 21; therefore the ACL requires both port name keywords **ftp** and **ftp-data** or **eq 20** and **eq 21** to deny FTP.
If using port numbers instead of port names, the commands would be written as:
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 20
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21

To prevent the implied deny any statement at the end of the ACL from blocking all traffic, the **permit ip any any** statement is added. Without at least one **permit** statement in an ACL, all traffic on the interface where that ACL was applied would be dropped. The ACL should be applied inbound on the G0/1 interface so that traffic from the 192.168.11.0/24 LAN is filtered as it enters the router interface.




```
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255
192.168.10.0 0.0.0.255 eq ftp
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255
192.168.10.0 0.0.0.255 eq ftp-data
R1(config)# access-list 101 permit ip any any
R1(config)# interface g0/1
R1(config-if)#ip access-group 101 in
```

Named extended ACLs are created in essentially the same way that named standard ACLs are created. The figure below shows the named versions of the ACLs created in a previous example.




```
R1(config)#ip access-list extended SURFING
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)#ip access-list extended BROWSING
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
R1(config-if)#ip access-group SURFING in
R1(config-if)#ip access-group BROWSING out
```

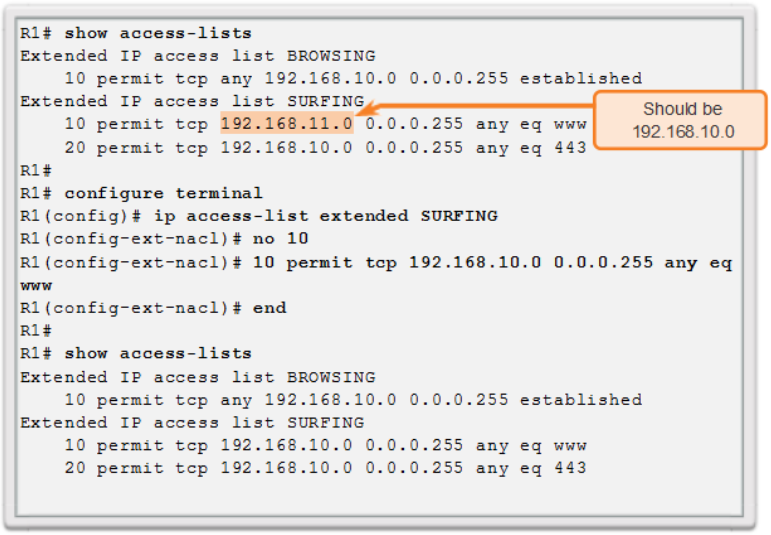
Modify ACLs.

Editing an extended ACL can be accomplished using the same process as editing a standard ACL. An ACL can be modified using:

- **Method 1 Text editor** - Using this method, the ACL is copied and pasted into the text editor where the changes are made. The current access list is removed using the **no access-list** command. The modified ACL is then pasted back into the configuration.

- **Method 2 Sequence numbers** - Sequence numbers can be used to delete or insert an ACL statement. The **ip access-list {standard | extended}** *name* command is used to enter named-ACL configuration mode. If the ACL is numbered instead of named, the ACL number is used in the *name* parameter. ACEs can be inserted or removed.

In the figure below the administrator needs to edit the ACL named SURFING to correct a typo in the source network statement. To view the current sequence numbers, the **show access-lists** command is used. The statement to be edited is identified as statement 10. The original statement is removed with the **no** *sequence_#* command. The corrected statement is added replacing the original statement.



Processing Packets with ACLs

Inbound ACL Logic

If the information in a packet header and the first ACL statement match, the rest of the statements in the list are skipped, and the packet is permitted or denied as specified by the matched statement. If a packet header does not match an ACL statement, the packet is tested against the next statement in the list. This matching process continues until the end of the list is reached.

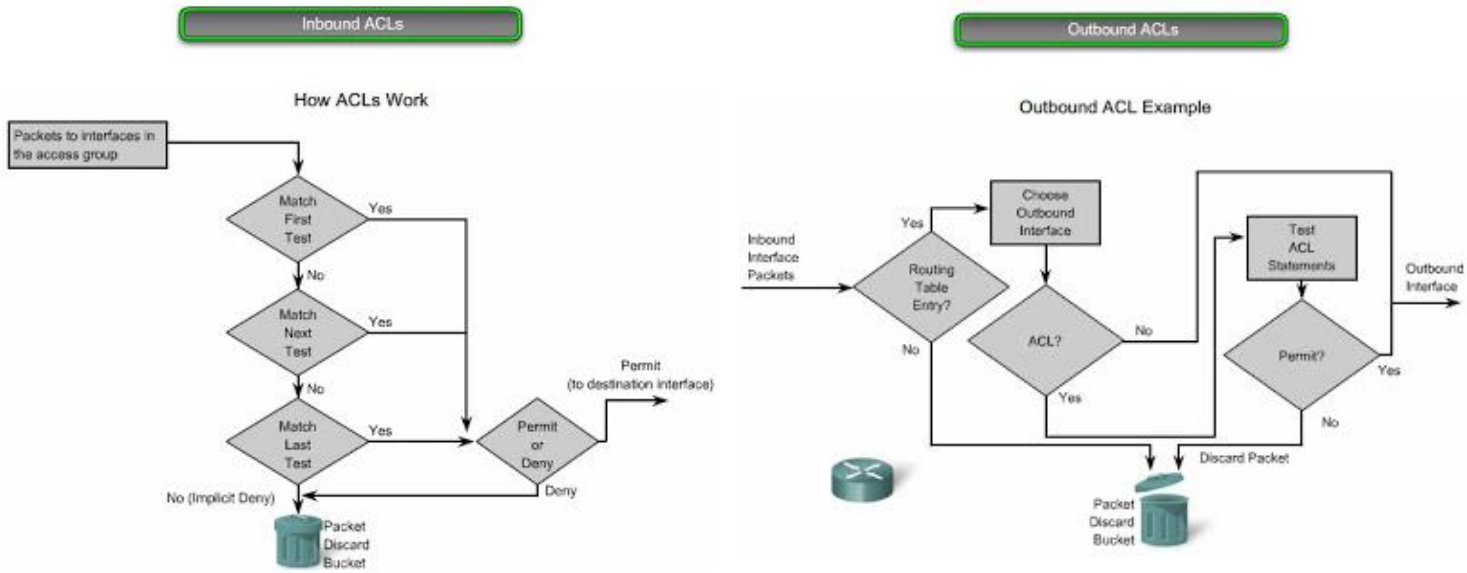
At the end of every ACL is a statement is an implicit *deny any* statement. This statement is not shown in output. This final implied statement applied to all packets for which conditions did not test true. This final test condition matches all other packets and results in a "deny" action. Instead of proceeding into or out of an interface, the router drops all of these remaining packets. This final statement is often referred to as the "implicit deny any statement" or the "deny all traffic" statement. Because of this statement, an ACL should have at least one permit statement in it; otherwise, the ACL blocks all traffic.

Outbound ACL Logic

Before a packet is forwarded to an outbound interface, the router checks the routing table to see if the packet is routable. If the packet is not routable, it is dropped and is not tested against the ACEs. Next, the router checks to see whether the outbound interface is grouped to an ACL. If the outbound interface is not grouped to an ACL, the packet can be sent to the output buffer. Examples of outbound ACL operation are as follows:

- **No ACL applied to the interface:** If the outbound interface is not grouped to an outbound ACL, the packet is sent directly to the outbound interface.
- **ACL applied to the interface:** If the outbound interface is grouped to an outbound ACL, the packet is not sent out on the outbound interface until it is tested by the combination of ACEs that are associated with that interface. Based on the ACL tests, the packet is permitted or denied.

For outbound lists, "permit" means to send the packet to the output buffer, and "deny" means to discard the packet.



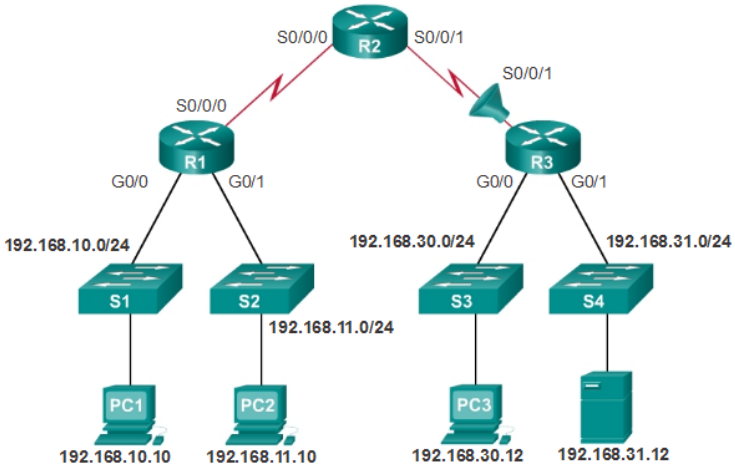
Troubleshoot ACLs

Using the **show** commands reveals most of the more common ACL errors. The most common errors are entering ACEs in the wrong order and not applying adequate criteria to the ACL rules.

Error Example 1

In the figure below, host 192.168.10.10 has no connectivity with 192.168.30.12. When viewing the output of the **show access-lists** command, matches are shown for the first deny statement. This is an indicator that this statement has been matched by traffic.

Solution - Look at the order of the ACEs (ACL Entries). Host 192.168.10.10 has no connectivity with 192.168.30.12 because of the order of rule 10 in the access list. Because the router processes ACLs from the top down, statement 10 denies host 192.168.10.10, so statement 20 can never be matched. Statements 10 and 20 should be reversed. The last line allows all other non-TCP traffic that falls under IP (ICMP, UDP, etc.).



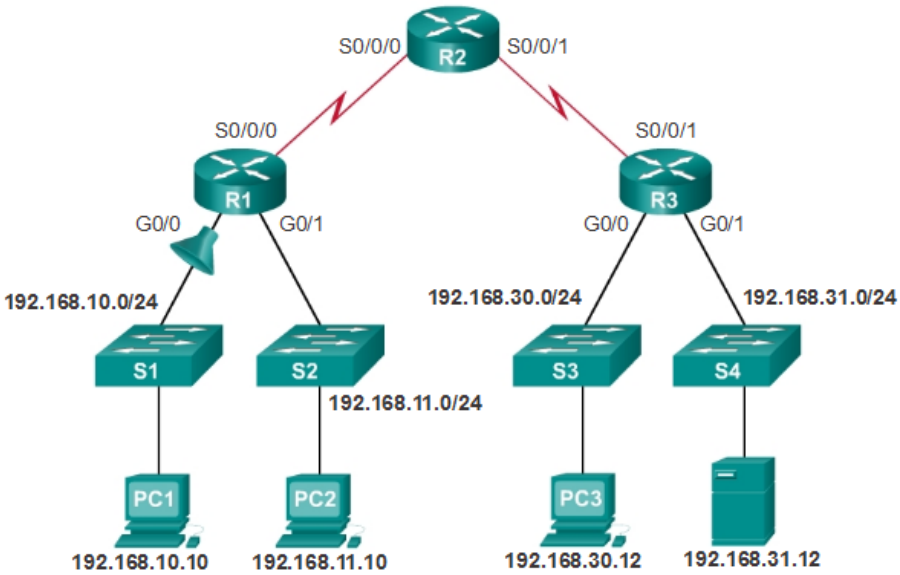

```
R3# show access-lists
Extended IP access list 110
    10 deny tcp 192.168.10.0 0.0.0.255 any (12 match(es))
    20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
    30 permit ip any any
```

Error Example 2

In the figure above, the 192.168.10.0/24 network cannot use TFTP to connect to the 192.168.30.0/24 network.

Solution - The 192.168.10.0/24 network cannot use TFTP to connect to the 192.168.30.0/24 network because TFTP uses the transport protocol UDP. Statement 30 in access list 120 allows all other TCP traffic. However, because TFTP uses UDP instead of TCP, it is implicitly denied. Recall that the implied deny any statement does not appear in **show access-lists** output and therefore matches are not shown. Statement 30 should be **ip any any**.

This ACL works whether it is applied to G0/0 of R1, or S0/0/1 of R3, or S0/0/0 of R2 in the incoming direction. However, based on the rule about placing extended ACLs closest to the source, the best option is to place it inbound on G0/0 of R1 because it allows undesirable traffic to be filtered without crossing the network infrastructure.



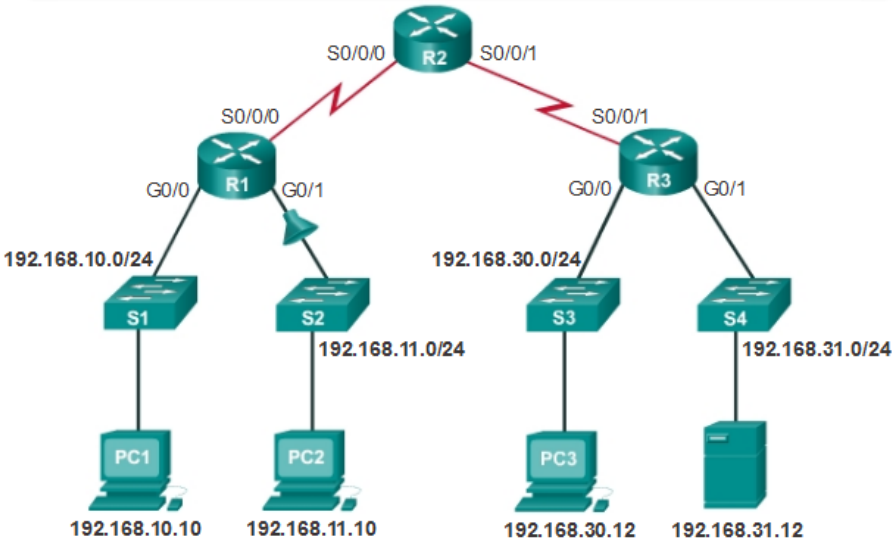

```
R3# show access-lists 120
Extended IP access list 120
    10 deny tcp 192.168.10.0 0.0.0.255 any eq telnet
    20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any
```

Error Example 3

In the figure above, the 192.168.11.0/24 network can use Telnet to connect to 192.168.30.0/24, but according to company policy, this connection should not be allowed. The results of the **show access-lists 130** command indicate that the permit statement has been matched.

Solution - The 192.168.11.0/24 network can use Telnet to connect to the 192.168.30.0/24 network, because the Telnet port number in statement 10 of access list 130 is listed in the wrong position in the ACL statement. Statement 10 currently denies any source packet with a port number that is equal to Telnet. To deny Telnet traffic inbound on G0/1, deny the destination port number that is equal to Telnet, for example, **deny tcp any any eq telnet**.

```
R1#show access-lists 130
Extended IP access list 130
  10 deny tcp any eq telnet any
  20 deny tcp 192.168.11.0 0.0.0.255 host 192.168.31.12 eq smtp
  30 permit tcp any any (12 match(es))
```

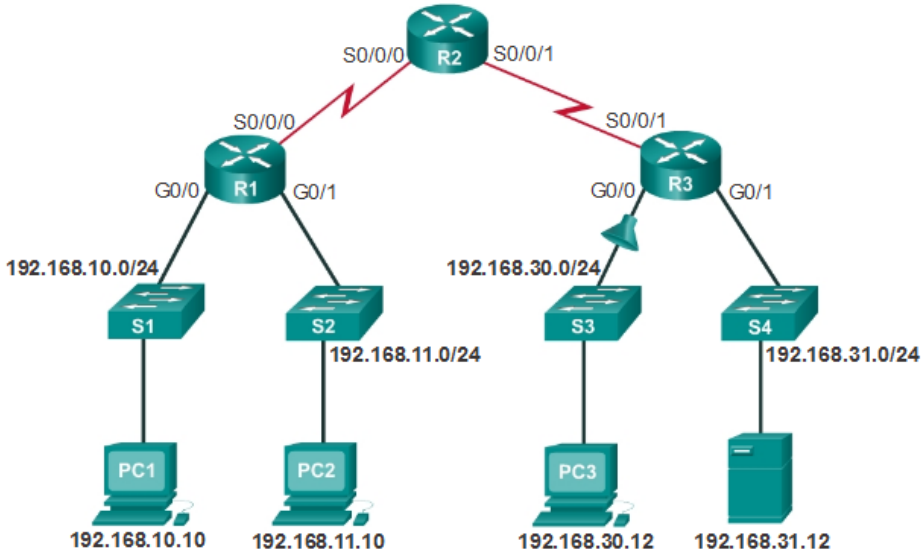


Error Example 4

In the figure below, host 192.168.30.12 is able to Telnet to connect to 192.168.31.12, but company policy states that this connection should not be allowed. Output from the **show access-lists 140** command indicate that the permit statement has been matched.

Solution - Host 192.168.30.12 can use Telnet to connect to 192.168.31.12 because there are no rules that deny host 192.168.30.12 or its network as the source. Statement 10 of access list 140 denies the router interface on which traffic enters the router. The host IPv4 address in statement 10 should be 192.168.30.12.

```
R3#show access-lists 140
Extended IP access list 140
         10 deny tcp host 192.168.30.1 any eq telnet
         20 permit ip any any (5 match(es))
```

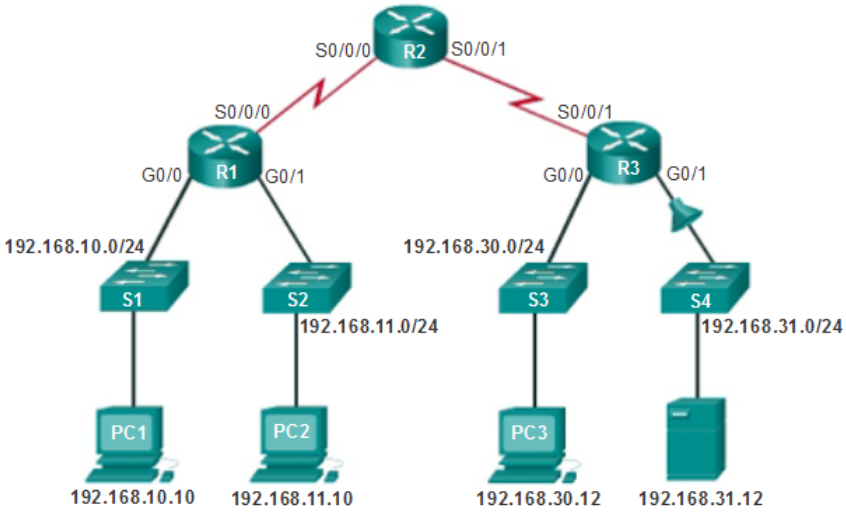


Error Example 5

In the figure below, host 192.168.30.12 can use Telnet to connect to 192.168.31.12, but according to the security policy, this connection should not be allowed. Output from the **show access-lists 150** command indicate that no matches have occurred for the deny statement as expected.

Solution - Host 192.168.30.12 can use Telnet to connect to 192.168.31.12 because of the direction in which access list 150 is applied to the G0/1 interface. Statement 10 denies any source address to connect to host 192.168.31.12 using telnet. However, this filter should be applied outbound on G0/1 to filter correctly.

```
R2#show access-lists 150
Extended IP access list 150
  10 deny tcp any host 192.168.31.12 eq telnet
  20 permit ip any any
```



Procedures:

Dear students, please note that the lab problems sheet, the packet tracer activities and the practical discussion videos have been uploaded on your Microsoft Teams group. You are required to carefully study this experiment and then complete the lab sheet.

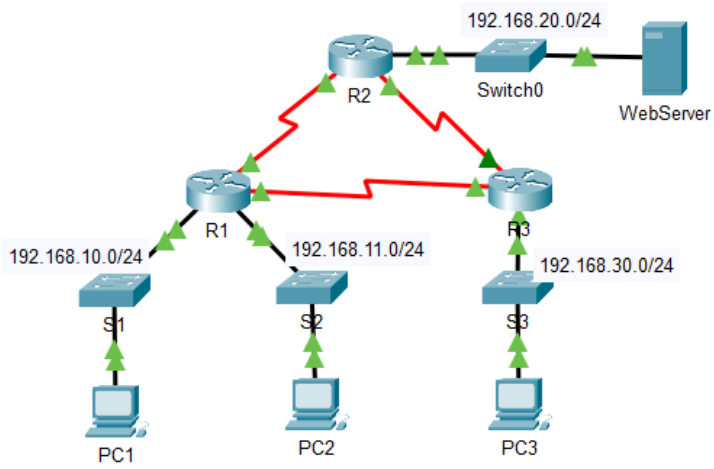
References

Enterprise Networking, Security, and Automation - Cisco Networking Academy
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.6 Access Control Lists (ACLs)
Lab sheet

Problem 1: Configuring Standard ACLs

This activity focuses on defining filtering criteria, configuring standard ACLs, applying ACLs to router interfaces, and verifying and testing the ACL implementation. The routers are already configured, including IP addresses and Enhanced Interior Gateway Routing Protocol (EIGRP) routing.

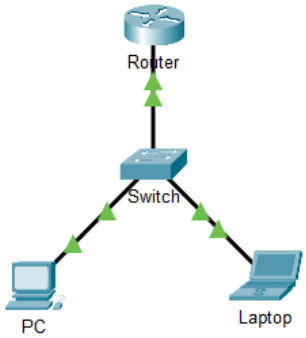


Task 1: Plan an ACL Implementation

Task 2: Configure, Apply, and Verify a Standard ACL

Problem 2: Configuring an ACL on VTY Lines

As network administrator, you must have remote access to your router. This access should not be available to other users of the network. Therefore, you will configure and apply an access control list (ACL) that allows PC access to the Telnet lines, but denies all other source IP addresses.

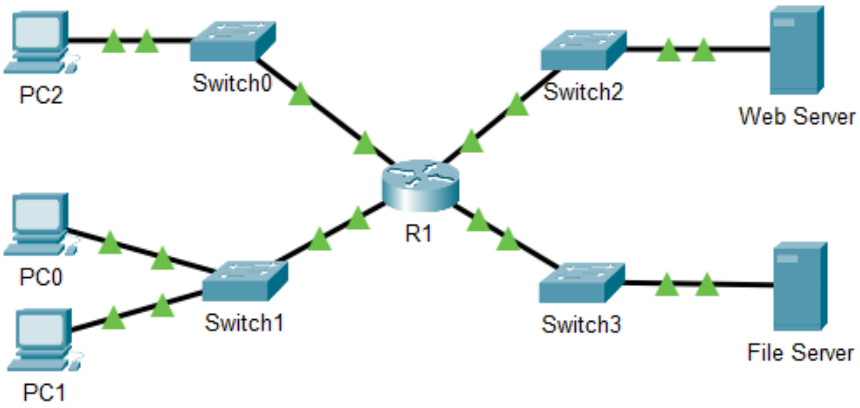


Task 1: Configure and Apply an ACL to VTY Lines.

Task 2: Verify the ACL Implementation.

Problem 3: Configuring Named Standard ACLs

The senior network administrator has tasked you to create a standard named ACL to prevent access to a file server. All clients from one network and one specific workstation from a different network should be denied access.

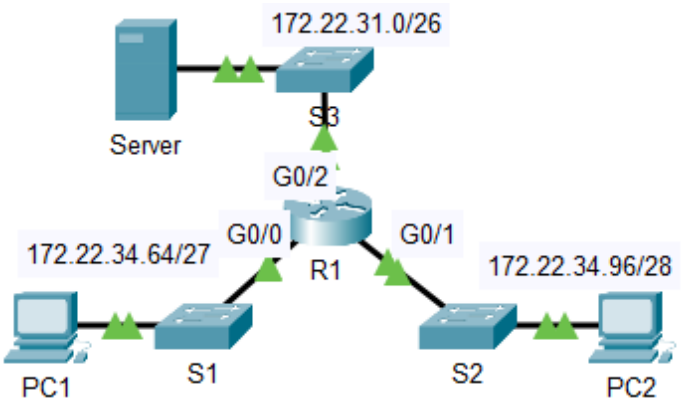


Task 1: Configure and Apply a Named Standard ACL.

Task 2: Verify the ACL Implementation.

Problem 4: Configuring Extended ACLs - Scenario 1

In this scenario, two employees need access to services provided by the server. PC1 only needs FTP access while PC2 only needs web access. Both computers are able to ping the server, but not each other.

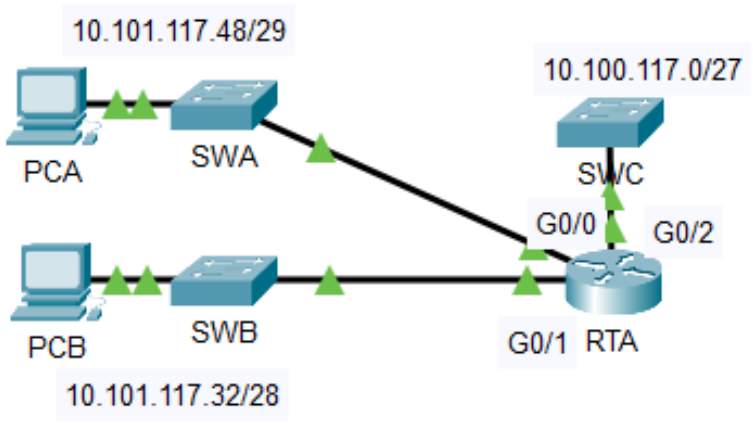


Task 1: Configure, Apply and Verify an Extended Numbered ACL.

Task 2: Configure, Apply and Verify an Extended Named ACL.

Problem 5: Configuring Extended ACLs - Scenario 2

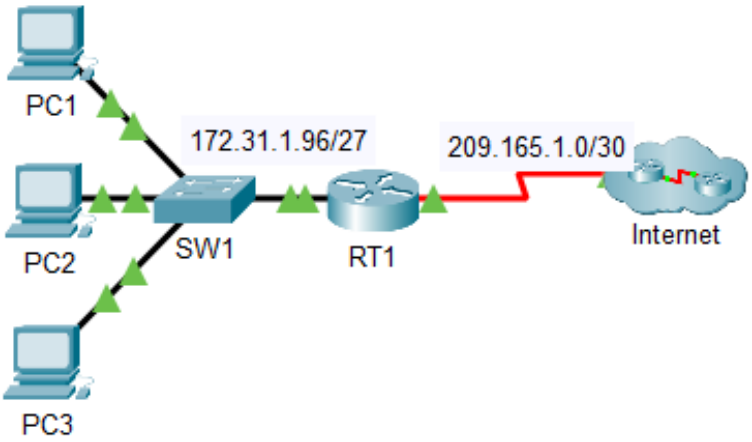
In this scenario, devices on one LAN are allowed to remotely access devices in another LAN using the Telnet protocol. Besides ICMP, all traffic from other networks is denied.



Task 1: Configure, Apply and Verify an Extended Numbered ACL.

Problem 6: Configuring Extended ACLs - Scenario 3

In this scenario, specific devices on the LAN are allowed to various services on servers located on the Internet.



Task 1: Configure a Named Extended ACL.

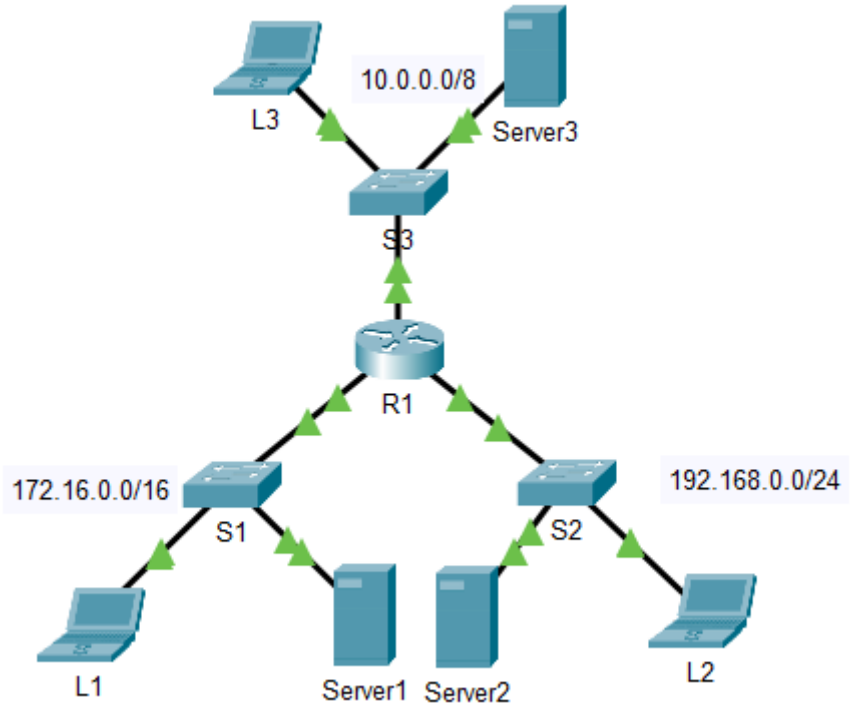
Task 2: Apply and Verify the Extended ACL.

Problem 7: Troubleshooting ACLs

This network is meant to have the following three policies implemented:

- Hosts from the 192.168.0.0/24 network are unable to access any TCP service of Server3.
- Hosts from the 10.0.0.0/8 network are unable to access the HTTP service of Server1.
- Hosts from the 172.16.0.0/16 network are unable to access the FTP service of Server2.

No other restrictions should be in place. Unfortunately, the rules that have been implemented are not working correctly. Your task is to find and fix the errors related to the access lists on R1.



Task 1: Troubleshoot ACL Issue 1.

Task 2: Troubleshoot ACL Issue 2.

Task 3: Troubleshoot ACL Issue 3.

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.7 Network Address Translation for IPv4 (NAT)

Objectives

1. Describe NAT operation.
2. Describe the benefits and drawbacks of NAT.
3. Configure static NAT using the CLI.
4. Configure dynamic NAT using the CLI.
5. Configure PAT using the CLI.
6. Use show commands to verify NAT operation.

NAT Operation:

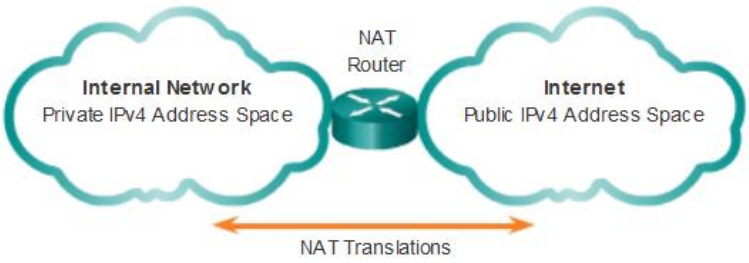
All public IPv4 addresses that transverse the Internet must be registered with a Regional Internet Registry (RIR). Organizations can lease public addresses from a service provider (SP), but only the registered holder of a public Internet address can assign that address to a network device. However, with a theoretical maximum of 4.3 billion addresses (2^{32}), IPv4 address space is severely limited. When Bob Kahn and Vint Cerf first developed the suite of TCP/IP protocols including IPv4 in 1981, they never envisioned what the Internet would become. At the time, the personal computer was mostly a curiosity for hobbyists and the World Wide Web was still more than a decade away. With the proliferation of personal computing and the advent of the World Wide Web, it soon became obvious that 4.3 billion IPv4 addresses would not be enough. The long term solution was IPv6, but more immediate solutions to address exhaustion were required. For the short term, several solutions were implemented by the IETF including Network Address Translation (NAT) and RFC 1918 private IPv4 addresses. This experiment discusses how NAT, combined with the use of private address space, is used to both conserve and more efficiently use IPv4 addresses to provide networks of all sizes access to the Internet.
Networks are commonly implemented using private IPv4 addresses, as defined in RFC 1918. The figure below shows the range of addresses included in RFC 1918.

Private Internet addresses are defined in RFC 1918:

Class	RFC 1918 Internal Address Range	CIDR Prefix
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

These private addresses are used within an organization or site to allow devices to communicate locally. However, because these addresses do not identify any single company or organization, private IPv4 addresses cannot be routed over the Internet. 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. This allows a device with a private IPv4 address to access resources outside of their private network, such as those found on the Internet. NAT combined with private IPv4 addresses, has proven to be a useful method of preserving public IPv4 addresses. A single, public IPv4 address can be shared by hundreds, even thousands of devices, each configured with a unique private IPv4 address.

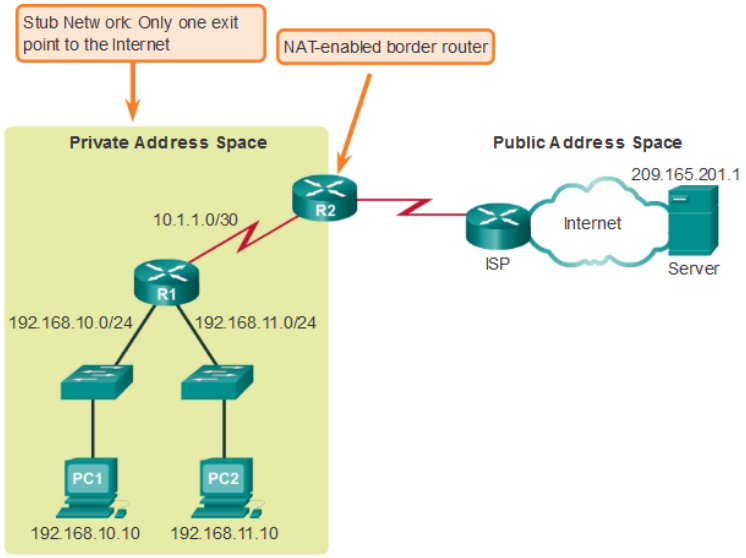
Without NAT, the exhaustion of the IPv4 address space would have occurred well before the year 2000. However, NAT has certain limitations, which will be explored later. The solution to the exhaustion of IPv4 address space and the limitations of NAT is the eventual transition to IPv6.



NAT has many uses, but its primary use is to conserve public IPv4 addresses. NAT has an added benefit of adding a degree of privacy and security to a network, because it hides internal IPv4 addresses from outside networks.

NAT-enabled routers can be configured with one or more valid public IPv4 addresses. These public addresses are known as the NAT pool. When an internal device sends traffic out of the network, the NAT-enabled router translates the internal IPv4 address of the device to a public address from the NAT pool. To outside devices, all traffic entering and exiting the network appears to have a public IPv4 address from the provided pool of addresses.

A NAT router typically operates at the border of a stub network. A stub network is a network that has a single connection to its neighboring network, one way in and one way out of the network. In the example below, R2 is a border router. As seen from the ISP, R2 forms a stub network.



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

When using NAT, IPv4 addresses have different designations based on whether they are on the private network, or on the public network (Internet), and whether the traffic is incoming or outgoing.

NAT includes four types of addresses:

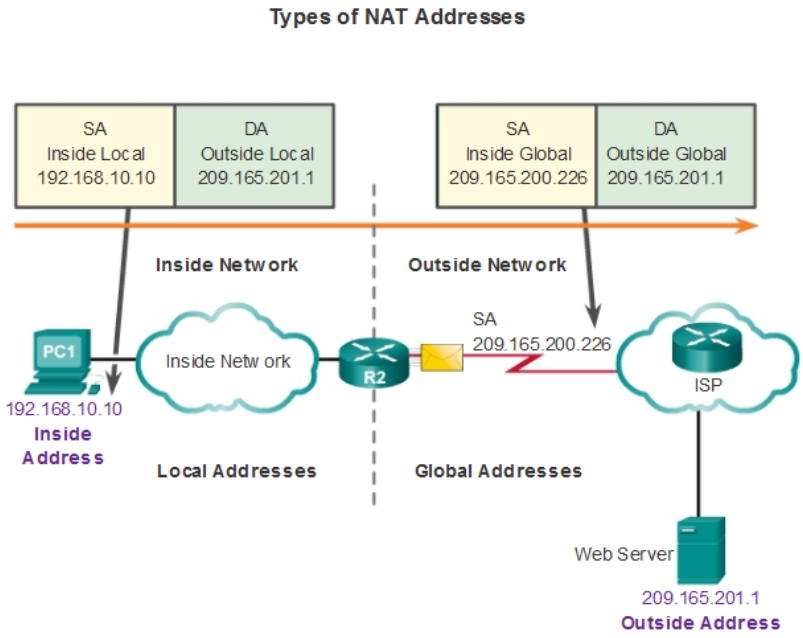
- Inside local address
- Inside global address
- Outside local address
- Outside global address

When determining which type of address is used, it is important to remember that 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.

NAT also uses the concept of local or global with respect to addresses:

- **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.



Static NAT

Static NAT uses a one-to-one mapping of local and global addresses. These mappings are configured by the network administrator and remain constant.

In the figure, R2 is configured with static mappings for the inside local addresses of Svr1, PC2, and PC3. When these devices send traffic to the Internet, their inside local addresses are translated to the configured inside global addresses. To outside networks, these devices have public IPv4 addresses.

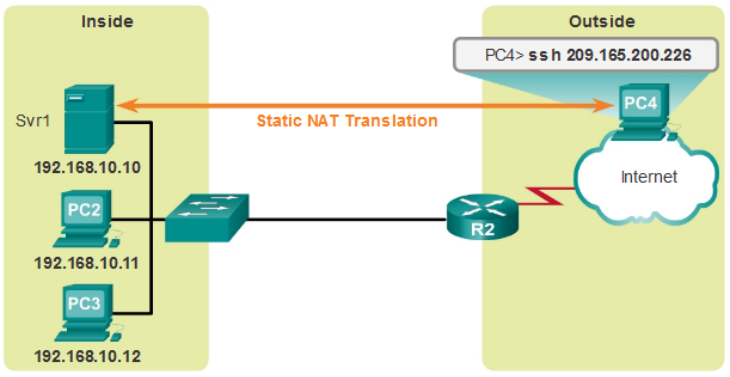
Static NAT is particularly useful for web servers or devices that must have a consistent address that is accessible from the Internet, such as a company web server. It is also useful for devices that must be accessible by authorized personnel when offsite, but not by the general public on the Internet. For example, a network administrator from PC4 can SSH to Svr1's inside global address (209.165.200.226). R2 translates this inside global address to the inside local address and connects the administrator's session to Svr1.

Static NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

Static NAT

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



Dynamic NAT

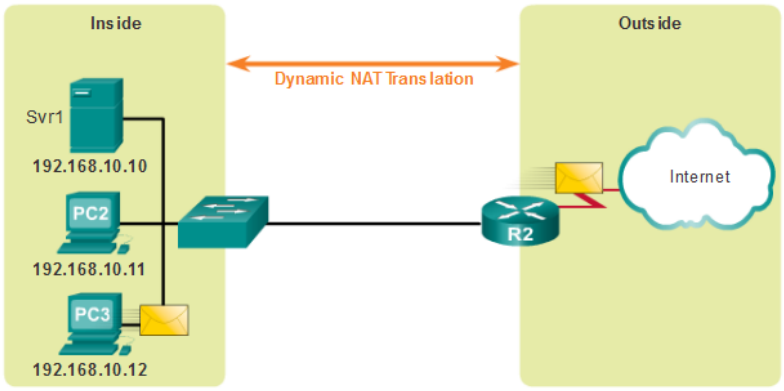
Dynamic NAT uses a pool of public addresses and assigns them on a first-come, first-served basis. When an inside device requests access to an outside network, dynamic NAT assigns an available public IPv4 address from the pool.

In the figure, PC3 has accessed the Internet using the first available address in the dynamic NAT pool. The other addresses are still available for use. Similar to static NAT, dynamic NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

Dynamic NAT

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 (PAT)

Port Address Translation (PAT), also known as NAT overloading, maps multiple private IPv4 addresses to a single public IPv4 address or a few addresses. This is what most home routers do. The ISP assigns one address to the router, yet several members of the household can simultaneously access the Internet. This is the most common form of NAT.

With PAT, multiple addresses can be mapped to one or to a few addresses, because each private address is also tracked by a port number. When a device initiates a TCP/IP session, it generates a TCP or UDP source port value to uniquely identify the session. When the NAT router receives a packet from the client, it uses its source port number to uniquely identify the specific NAT translation.

PAT ensures that devices use a different TCP port number for each session with a server on the Internet. When a response comes back from the server, the source port number, which becomes the destination port number on the return trip, determines to which device the router forwards the packets. The PAT process also validates that the incoming packets were requested, thus adding a degree of security to the session.

PAT

Inside Global Address	Inside Local Address
209.165.200.226:1444	192.168.10.10:1444
209.165.200.226:1445	192.168.10.11:1444
209.165.200.226:1555	192.168.10.12:1555
209.165.200.226:1556	192.168.10.13:1555

Note that PAT attempts to preserve the original source port. However, if the original source port is already used, PAT assigns the first available port number.

Benefits of NAT

NAT provides many benefits, including:

- NAT conserves the legally registered addressing scheme by allowing the privatization of intranets. NAT conserves addresses through application port-level multiplexing. With NAT overload, internal hosts can share a single public IPv4 address for all external communications. In this type of configuration, very few external addresses are required to support many internal hosts.
- NAT increases the flexibility of connections to the public network. Multiple pools, backup pools, and load-balancing pools can be implemented to ensure reliable public network connections.
- NAT provides consistency for internal network addressing schemes. On a network not using private IPv4 addresses and NAT, changing the public IPv4 address scheme requires the readdressing of all hosts on the existing network. The costs of readdressing hosts can be significant. NAT allows the existing private IPv4 address scheme to remain while allowing for easy change to a new public addressing scheme. This means an organization could change ISPs and not need to change any of its inside clients.
- NAT provides network security. Because private networks do not advertise their addresses or internal topology, they remain reasonably secure when used in conjunction with NAT to gain controlled external access. However, NAT does not replace firewalls.

NAT does have some drawbacks. The fact that hosts on the Internet appear to communicate directly with the NAT-enabled device, rather than with the actual host inside the private network, creates a number of issues.

One disadvantage of using NAT is related to network performance, particularly for real time protocols such as VoIP. NAT increases switching delays because the translation of each IPv4 address within the packet headers takes time. The first packet is process-switched; it always goes through the slower path. The router must look at every packet to decide whether it needs translation. The router must alter the IPv4 header, and possibly alter the TCP or UDP header. The IPv4 header checksum, along with the TCP or UDP checksum must be recalculated each time a translation is made. Remaining packets go through the fast-switched path if a cache entry exists; otherwise, they too are delayed.

Another disadvantage of using NAT is that end-to-end addressing is lost. Many Internet protocols and applications depend on end-to-end addressing from the source to the destination. Some applications do not work with NAT. For example, some security applications, such as digital signatures, fail because the source IPv4 address changes before reaching the destination. Applications that use physical addresses, instead of a qualified domain name, do not reach destinations that are translated across the NAT router. Sometimes, this problem can be avoided by implementing static NAT mappings.

End-to-end IPv4 traceability is also lost. It becomes much more difficult to trace packets that undergo numerous packet address changes over multiple NAT hops, making troubleshooting challenging.

Configuring Static NAT

Static NAT is a one-to-one mapping between an inside address and an outside address. There are two basic tasks when configuring static NAT translations.

Step 1. The first task is to create a mapping between the inside local address and the inside global addresses. For example, the 192.168.10.254 inside local address and the 209.165.201.5 inside global address in the figure below are configured as a static NAT translation.

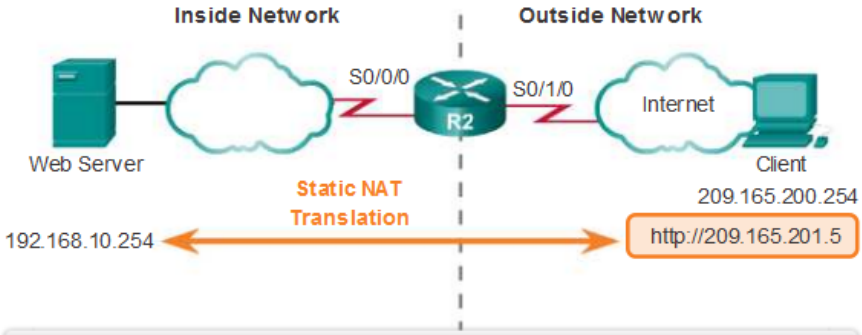
*Router(config)# **ip nat inside source static** local-ip global-ip*

Step 2. After the mapping is configured, the interfaces participating in the translation are configured as inside or outside relative to NAT. In the example, the Serial 0/0/0 interface of R2 is an inside interface and Serial 0/1/0 is an outside interface.

*Router(config)# **interface** type number*
*Router(config-if)# **ip nat {inside | outside}***

The figure below shows an inside network containing a web server with a private IPv4 address. Router R2 is configured with static NAT to allow devices on the outside network (Internet) to access the web server. The client on the outside network accesses the web server using a public IPv4 address. Static NAT translates the public IPv4 address to the private IPv4 address.

Example Static NAT Configuration




```
Establishes static translation between an inside local address and
an inside global address.
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5

R2(config)# interface Serial0/0/0
R2(config-if)# ip address 10.1.1.2 255.255.255.252
Identifies interface serial 0/0/0 as an inside NAT interface.
R2(config-if)# ip nat inside
R2(config-if)# exit

R2(config)# interface Serial0/1/0
R2(config-if)# ip address 209.165.200.225 255.255.255.224
Identifies interface serial 0/1/0 as the outside NAT interface.
R2(config-if)# ip nat outside
```

A useful command to verify NAT operation is the **show ip nat translations** command. This command shows active NAT translations. Static translations, unlike dynamic translations, are always in the NAT table. The figure below shows the output from this command using the previous configuration example. 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 outside device as shown below.

The static translation is always present in the NAT table.

```
R2# show ip nat translations
Pro Inside global  Inside local   Outside local  Outside global
--- 209.165.201.5  192.168.10.254  ---            ---
R2#
```

The static translation during an active session.

```
R2# show ip nat translations
Pro Inside global Inside local   Outside local    Outside global
--- 209.165.201.5 192.168.10.254 209.165.200.254  209.165.200.254
R2#
```

Another useful command is the **show ip nat statistics** command. As shown in the figure below, 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 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.
Prior to any communications with the web server, the **show ip nat statistics** command shows no current hits. After the client establishes a session with the web server, the **show ip nat statistics** command has been incremented to five hits. This verifies that the static NAT translation is taking place on R2.

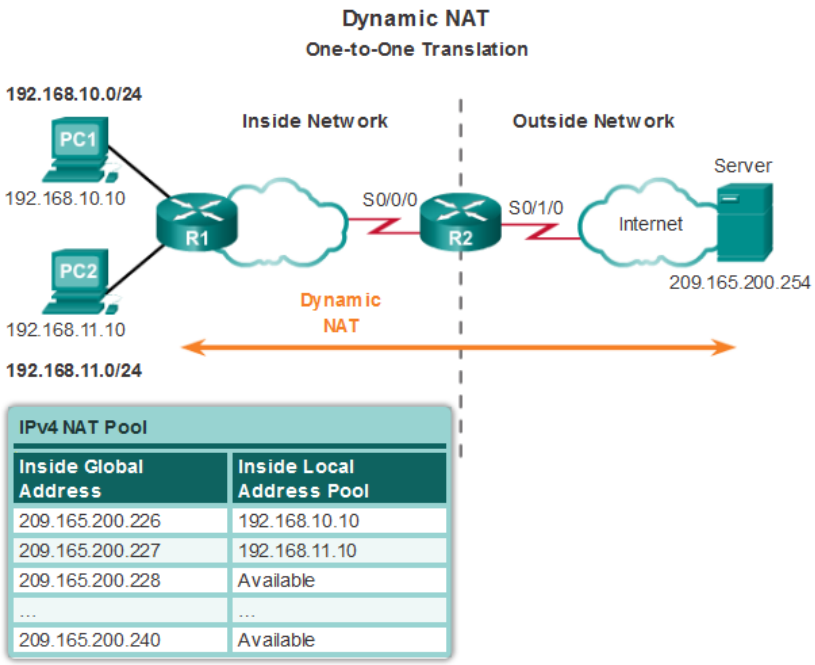
```
R2# clear ip nat statistics

R2# show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Peak translations: 0
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  Serial0/0/0
Hits: 0  Misses: 0
<output omitted>

Client PC establishes a session with the web server

R2# show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Peak translations: 2, occurred 00:00:14 ago
Outside interfaces:
  Serial0/1/0
Inside interfaces:
  Serial0/0/0
Hits: 5  Misses: 0


<output omitted>
```

Configuring Dynamic NAT

While static NAT provides a permanent mapping between an inside local address and an inside global address, dynamic NAT allows the automatic mapping of inside local addresses to inside global addresses. These inside global addresses are typically public IPv4 addresses. Dynamic NAT uses a group, or pool of public IPv4 addresses for translation.
Dynamic NAT, like static NAT, requires the configuration of the inside and outside interfaces participating in NAT.

The example topology shown below has an inside network using addresses from the RFC 1918 private address space. Attached to router R1 are two LANs, 192.168.10.0/24 and 192.168.11.0/24. Router R2, the border router, is configured for dynamic NAT using a pool of public IPv4 addresses 209.165.200.226 through 209.165.200.240.

The pool of public IPv4 addresses (inside global address pool) is available to any device on the inside network on a first-come first-served basis. With dynamic NAT, a single inside address is translated to a single outside address. With this type of translation there must be enough addresses in the pool to accommodate all the inside devices needing access to the outside network at the same time. If all of the addresses in the pool have been used, a device must wait for an available address before it can access the outside network.



IPv4 NAT Pool

Inside Global Address	Inside Local Address Pool
209.165.200.226	192.168.10.10
209.165.200.227	192.168.11.10
209.165.200.228	Available
…	…
209.165.200.240	Available

The steps and the commands used to configure dynamic NAT are as follows:

Step 1. Define the pool of addresses that will be used for translation using the **ip nat pool** command. This pool of addresses is typically a group of public addresses. The addresses are defined by indicating the starting IP address and the ending IP address of the pool. The **netmask** or **prefix-length** keyword indicates which address bits belong to the network and which bits belong to the host for the range of addresses.

Step 2. Configure a standard ACL to identify (permit) only those addresses that are to be translated. An ACL that is too permissive can lead to unpredictable results. Remember there is an implicit **deny all** statement at the end of each ACL.

Step 3. Bind the ACL to the pool. The **ip nat inside source list** *access-list-number* **number pool** *pool name* command is used to bind the ACL to the pool. This configuration is used by the router to identify which devices (**list**) receive which addresses (**pool**).

Step 4. Identify which interfaces are inside, in relation to NAT; that is, any interface that connects to the inside network.

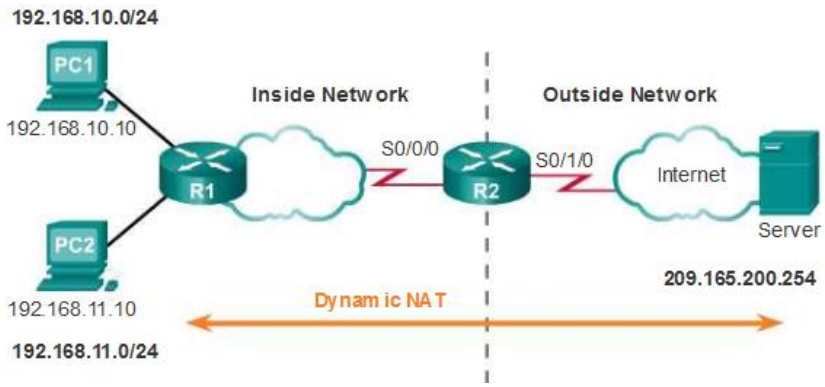
Step 5. Identify which interfaces are outside, in relation to NAT; that is, any interface that connects to the outside network.

Dynamic NAT Configuration Steps

Step 1	Define a pool of global addresses to be used for translation. `ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length}`
Step 2	Configure a standard access list permitting the addresses that should be translated. `access-list access-list-number permit source [source-wildcard]`
Step 3	Establish dynamic source translation, specifying the access list and pool defined in prior steps. `ip nat inside source list access-list-number pool name`

Step 4	Identify the inside interface.
	`interface type number`
	`ip nat inside`

Step 5	Identify the outside interface.
	`interface type number`
	`ip nat outside`



Define a pool of public IPv4 addresses 209.165.200.241 to 209.165.200.250 with pool name PUBLIC-POOL.
R2(config)# ip nat pool PUBLIC-POOL 209.165.200.241 209.165.200.250 netmask 255.255.255.224
Configure ACL 2 to permit devices from 192.168.10.0/24 network to be translated by NAT.
R2(config)# access-list 2 permit 192.168.10.0 0.0.0.255
Bind PUBLIC-POOL with ACL 2.
R2(config)# ip nat inside source list 2 pool PUBLIC-POOL
Configure the proper inside NAT interface.
R2(config)# interface Serial0/0/0
R2(config-if)# ip nat inside
Configure the proper outside NAT interface.
R2(config)# interface Serial0/1/0
R2(config-if)# ip nat outside
You successfully configured dynamic NAT.

The output of the **show ip nat translations** command shown below displays the details of the two previous NAT assignments. The 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
---  209.165.200.226     192.168.10.10  ---           ---
---  209.165.200.227     192.168.11.10  ---           ---
```

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 global configuration mode command. It is useful to clear the dynamic entries when testing the NAT configuration.
Note:Only the dynamic translations are cleared from the table. Static translations cannot be cleared from the translation table.
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.
Alternatively, use the **show running-config** command and look for NAT, ACL, interface, or pool commands with the required values. Examine these carefully and correct any errors discovered.

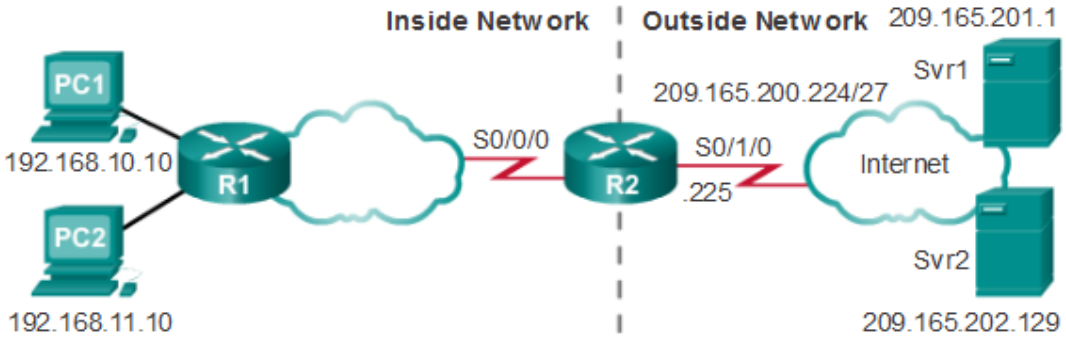
```
R2# clear ip nat statistics

PC1 and PC2 establish sessions with the server

R2# show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 0
extended)
Peak translations: 6, occurred 00:27:07 ago
Outside interfaces:
   Serial0/0/1
Inside interfaces:
   Serial0/1/0
Hits: 24 Misses: 0
CEF Translated packets: 24, CEF Punted packets: 0
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool NAT-POOL1 refcount 2
  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

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
R2#
```

Configuring Port Address Translation (PAT)

PAT (also called NAT overload) conserves addresses in the inside global address pool by allowing the router to use one inside global address for many inside local addresses. In other words, a single public IPv4 address can be used for hundreds, even thousands of internal private IPv4 addresses. When this type of translation is configured, the router maintains enough information from higher-level protocols, TCP or UDP port numbers, for example, to translate the inside global address back into the correct inside local address. When multiple inside local addresses map to one inside global address, the TCP or UDP port numbers of each inside host distinguish between the local addresses.
Note: The total number of internal addresses that can be translated to one external address could theoretically be as high as 65,536 per IP address. However, the number of internal addresses that can be assigned a single IP address is around 4,000.
There are two ways to configure PAT, depending on how the ISP allocates public IPv4 addresses. In the first instance, the ISP allocates more than one public IPv4 address to the organization, and in the other, it allocates a single public IPv4 address that is required for the organization to connect to the ISP.

Configuring PAT for a Pool of Public IP Addresses

If a site has been issued more than one public IPv4 address, these addresses can be part of a pool that is used by PAT. This is similar to dynamic NAT, except that there are not enough public addresses for a one-to-one mapping of inside to outside addresses. The small pool of addresses is shared among a larger number of devices. The primary difference between this configuration and the configuration for dynamic, one-to-one NAT is that the **overload** keyword is used. The **overload** keyword enables PAT.
The example configuration shown below establishes overload translation for the NAT pool named NAT-POOL2. NAT-POOL2 contains addresses 209.165.200.226 to 209.165.200.240. Hosts in the 192.168.0.0/16 network are subject to translation. The S0/0/0 interface is identified as an inside interface and the S0/1/0 interface is identified as an outside interface.

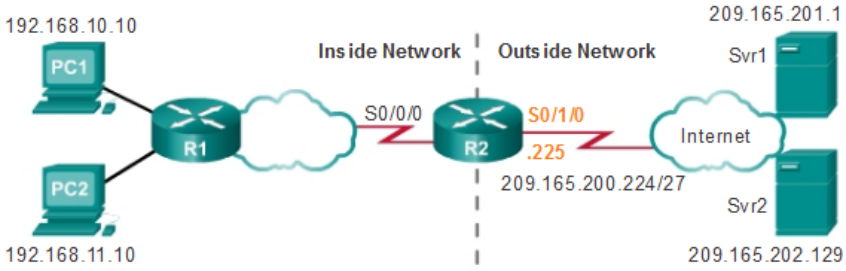


R2(config)# ip nat pool NAT-POOL2 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 2 pool NAT-POOL2 overload
R2(config)# interface Serial0/0/0
R2(config-if)# ip nat inside
R2(config)# interface Serial0/1/0
R2(config-if)# ip nat outside

Configuring PAT for a Single Public IPv4 Address

The following figure shows the topology of a PAT implementation for a single public IPv4 address translation. 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 S0/1/0). The traffic flows will be identified by port numbers in the NAT table, because the **overload** keyword was used.

With only a single public IPv4 address available, the overload configuration typically assigns the public address to the outside interface that connects to the ISP. All inside addresses are translated to the single IPv4 address when leaving the outside interface.



NAT Table

Inside Global Address	Inside Local Address	Outside Local Address	Outside Global Address
209.165.200.225:1444	192.168.10.10:1444	209.165.201.1:80	209.165.201.1:80
209.165.200.225:1445	192.168.10.11:1444	209.165.202.129:80	209.165.202.129:80

R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat source list 1 interface serial 0/1/0 overload
R2(config)# interface serial0/0/0
R2(config-if)# ip nat inside
R2(config)# interface serial0/1/0
R2(config-if)# ip nat outside

The configuration is similar to dynamic NAT, except that instead of a pool of addresses, the **interface** keyword is used to identify the outside IPv4 address. Therefore, no NAT pool is defined.

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
tcp 209.165.200.226:51839  192.168.10.10:51839  209.165.201.1:80
tcp 209.165.200.226:42558  192.168.11.10:42558  209.165.202.129:8
R2#
```

The **show ip nat statistics** command verifies that NAT-POOL2 has allocated a single address for both translations. Included in the output is information about the number and type of active translations, NAT configuration parameters, the number of addresses in the pool, and how many have been allocated.

```
R2# clear ip nat statistics

R2# show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:00:05 ago
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  Serial0/1/0
Hits: 4  Misses: 0
CEF Translated packets: 4, 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 209.165.200.226 end 209.165.200.240
     type generic, total addresses 15, allocated 1 (6%), misses 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
R2#
```

Procedures:

Dear students, please note that the lab problems sheet, the packet tracer activities and the practical discussion videos have been uploaded on your Microsoft Teams group. You are required to carefully study this experiment and then complete the lab sheet.

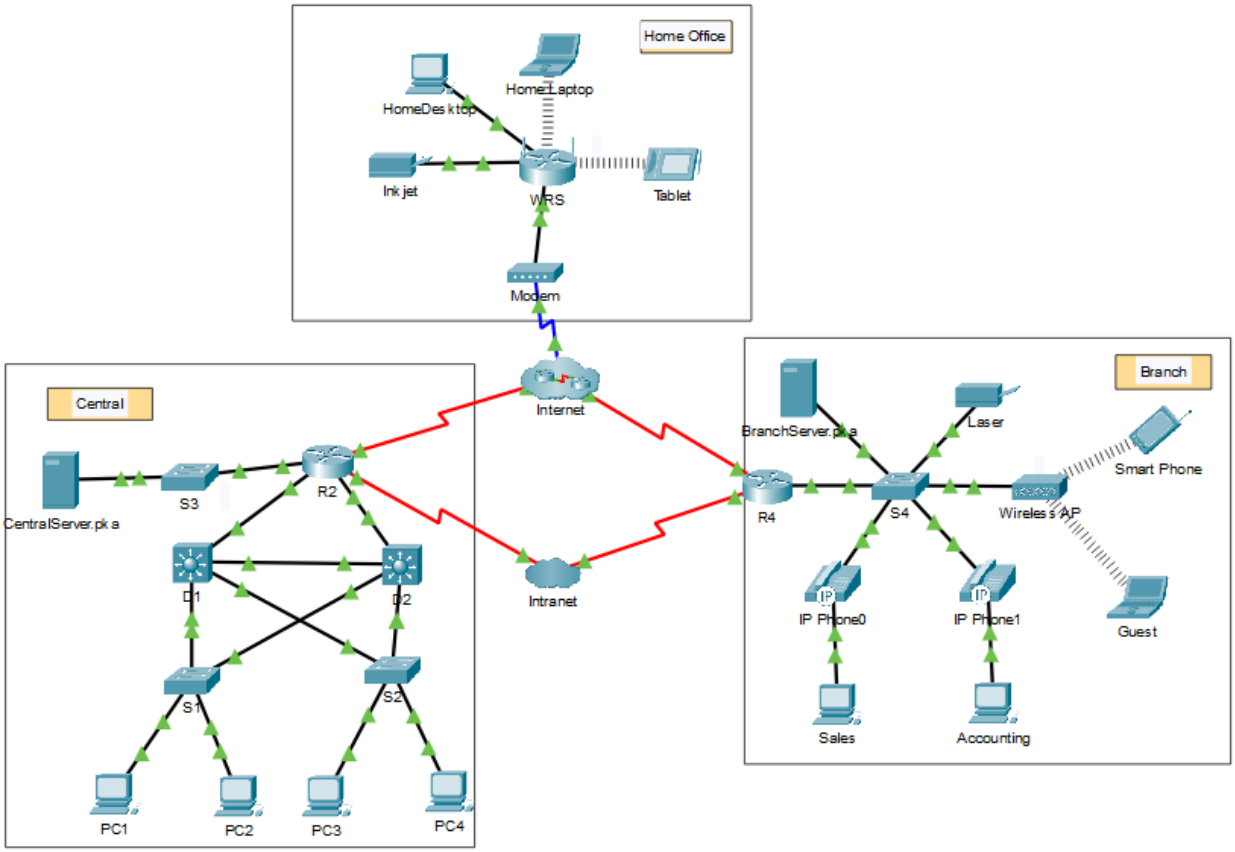
References

Enterprise Networking, Security, and Automation - Cisco Networking Academy
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.7 Network Address Translation for IPv4 (NAT)
Lab sheet

Problem 1: Investigate NAT Operations

As a frame travels across a network, the MAC addresses may change. IP addresses can also change when a packet is forwarded by a device configured with NAT. In this activity, we will investigate what happens to IP addresses during the NAT process.



Task 1: Investigate NAT Operation Across the <u>Intranet</u>

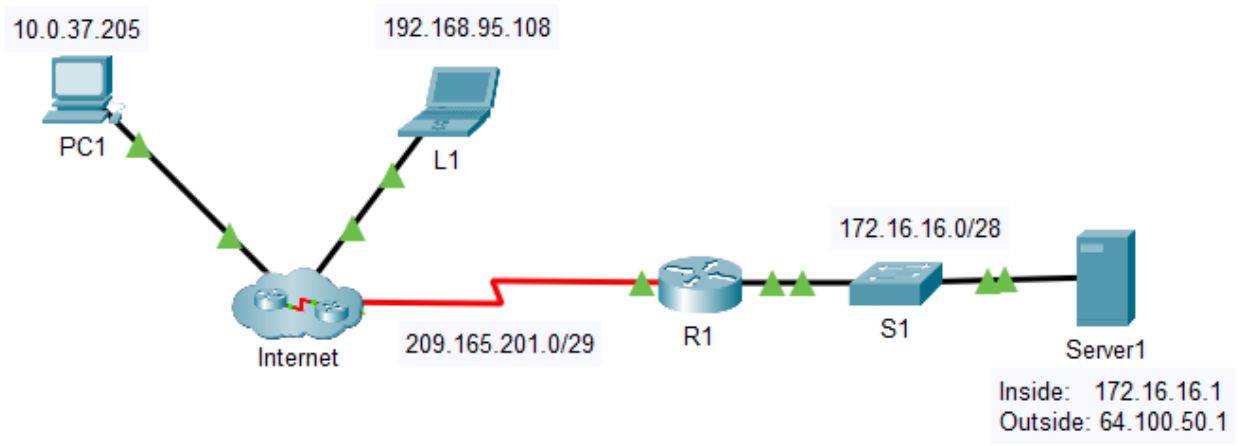
Task 2: Investigate NAT Operation Across the <u>Internet</u>

Task 3: Conduct Further Investigations

Problem 2: Configure Static NAT

In IPv4 configured networks, clients and servers use private addressing. Before packets with private addressing can cross the internet, they need to be translated to public addressing. Servers that are accessed from outside the organization are usually assigned both a public and a private static IP address. In this

activity, you will configure static NAT so that outside devices can access an inside server at its public address.



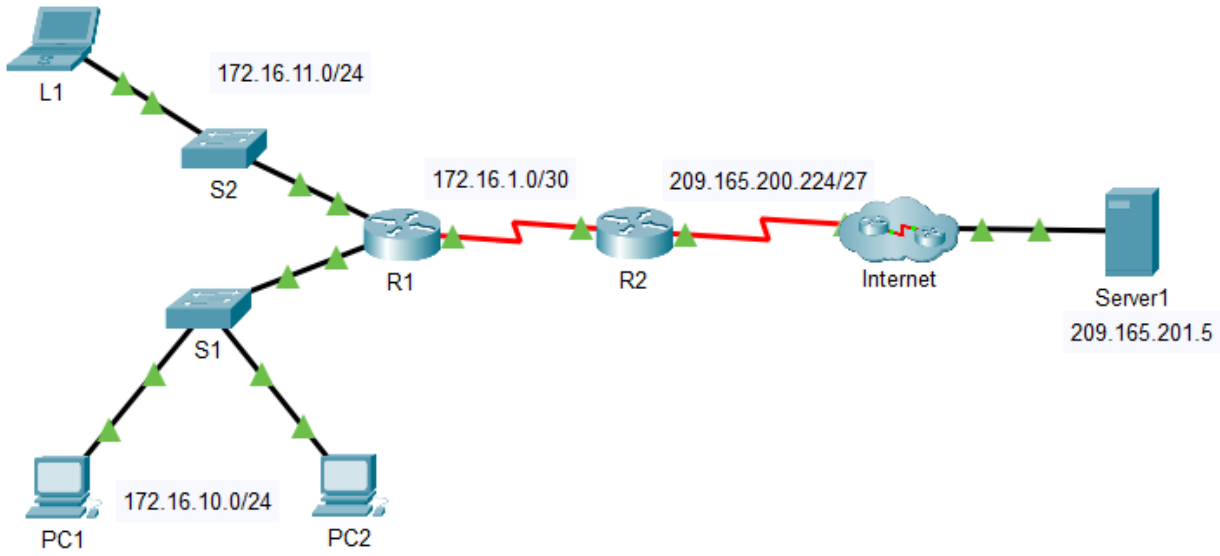
Task 1: Test Access without NAT

Task 2: Configure Static NAT

Task 3: Test Access with NAT

Problem 3: Configure Dynamic NAT

In this activity, you will learn how to configure and verify dynamic NAT. Dynamic NAT uses a pool of addresses and automatically maps inside local addresses to inside global addresses.

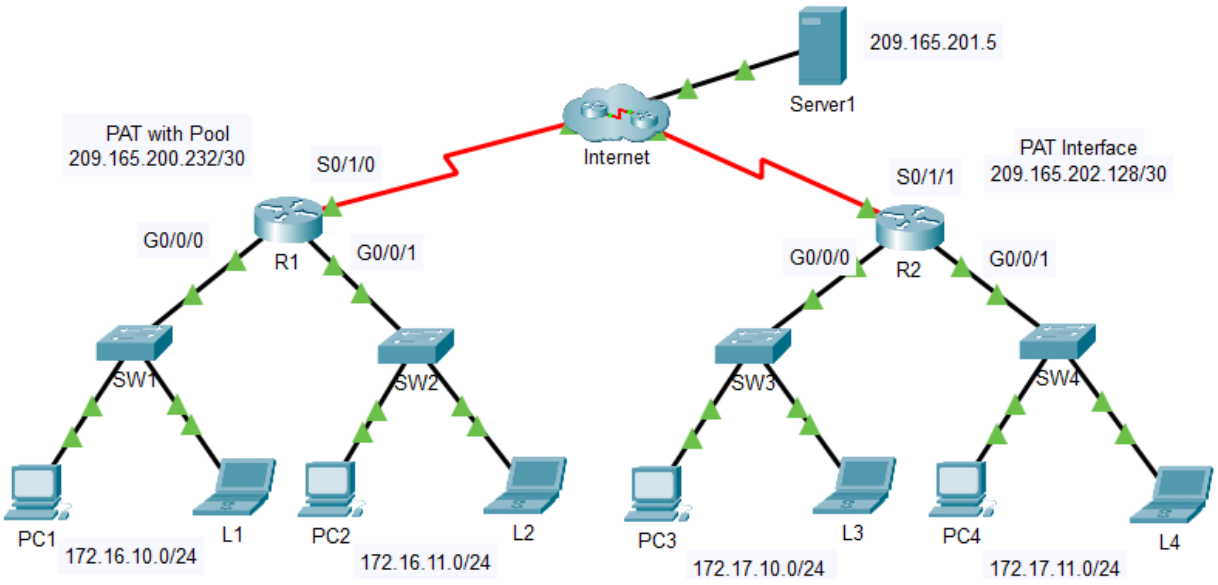


Task 1: Configure Dynamic NAT

Task 2: Verify NAT Implementation

Problem 4: Configure PAT

In this activity, you will learn how to configure and verify dynamic NAT with Overload and how to configure PAT using an Interface.



Task 1: Configure Dynamic NAT with Overload

Task 2: Verify Dynamic NAT with Overload Implementation

Task 3: Configure PAT using an Interface

Task 4: Verify PAT Interface Implementation

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.8 Dynamic Host Configuration Protocol (DHCP)

Objectives

1. Describe the operation of DHCP in a small-to-medium-sized business network.
2. Configure a router as a DHCP server.
3. Configure a router as a DHCP client.
4. Troubleshoot a DHCP configuration in a switched network.

Introduction

Every device that connects to a network needs a unique IP address. Network administrators assign static IP addresses to routers, servers, printers, and other network devices whose locations (physical and logical) are not likely to change. These are usually devices that provide services to users and devices on the network; therefore, the addresses assigned to them should remain constant. Additionally, static addresses enable administrators to manage these devices remotely. It is easier for network administrators to access a device when they can easily determine its IP address.

However, computers and users in an organization often change locations, physically and logically. It can be difficult and time consuming for administrators to assign new IP addresses every time an employee moves. Additionally, for mobile employees working from remote locations, manually setting the correct network parameters can be challenging. Even for desktop clients, the manual assignment of IP addresses and other addressing information presents an administrative burden, especially as the network grows.

Introducing a Dynamic Host Configuration Protocol (DHCP) server to the local network simplifies IP address assignment to both desktop and mobile devices. Using a centralized DHCP server enables organizations to administer all dynamic IP address assignments from a single server. This practice makes IP address management more effective and ensures consistency across the organization, including branch offices.

DHCPv4 Operation

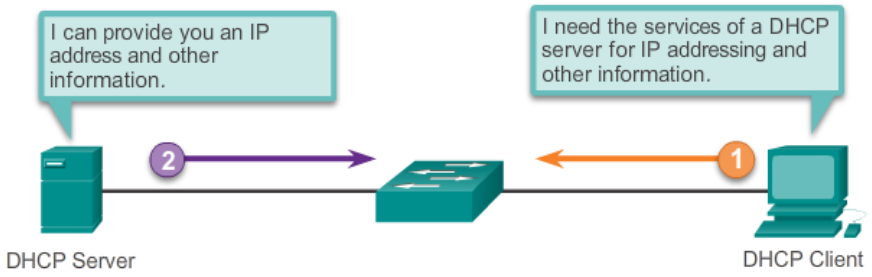
DHCPv4 assigns IPv4 addresses and other network configuration information dynamically. Because desktop clients typically make up the bulk of network nodes, DHCPv4 is an extremely useful and timesaving tool for network administrators.

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. A Cisco IOS feature set (called ”Easy IP)” offers an optional, full-featured DHCPv4 server.

DHCPv4 includes three different address allocation mechanisms to provide flexibility when assigning IP addresses:

- **Manual Allocation** - The administrator assigns a pre-allocated IPv4 address to the client, and DHCPv4 communicates only the IPv4 address to the device.
- **Automatic Allocation** - DHCPv4 automatically assigns a static IPv4 address permanently to a device, selecting it from a pool of available addresses. There is no lease and the address is permanently assigned to the device.
- **Dynamic Allocation** - DHCPv4 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.

Dynamic allocation is the most commonly used DHCPv4 mechanism, as shown in the figure below. 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.



Lease Origination

When the client boots (or otherwise wants to join a network), it begins a four step process to obtain a lease. As shown in the Figure below , a client starts the process with a broadcast DHCPDISCOVER message with its own MAC address to discover available DHCPv4 servers.

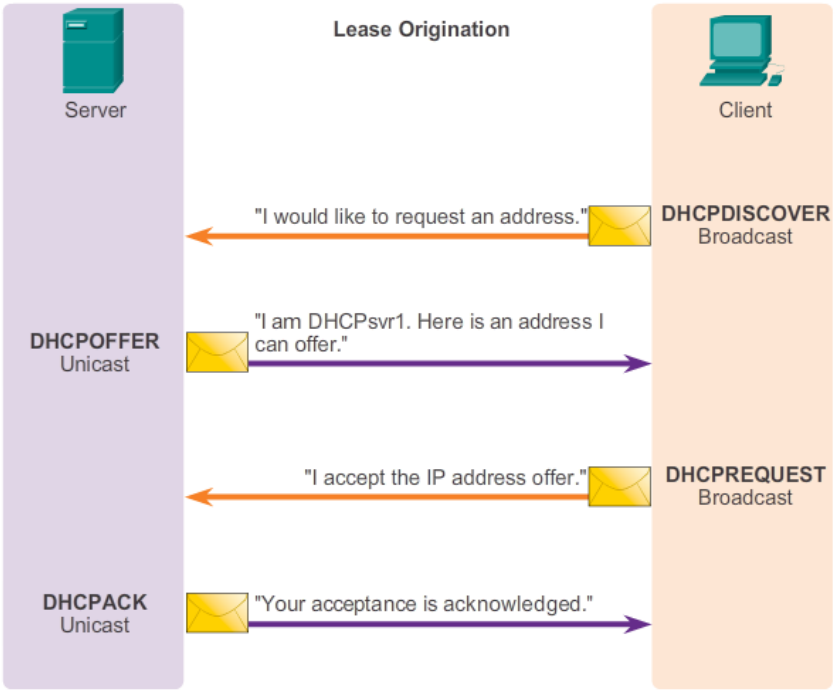
DHCP Discover (DHCPDISCOVER)
The DHCPDISCOVER message finds DHCPv4 servers on the network. Because the client has no valid IPv4 information at bootup, it uses Layer 2 and Layer 3 broadcast addresses to communicate with the server.

DHCP Offer (DHCPOFFER)
When the DHCPv4 server receives a DHCPDISCOVER message, it reserves an available IPv4 address to lease to the client. The server also creates an ARP entry consisting of the MAC address of the requesting client and the leased IPv4 address of the client. As shown in the Figure below, the DHCPv4 server sends the binding DHCPOFFER message to the requesting client. The DHCPOFFER message is sent as a unicast, using the Layer 2 MAC address of the server as the source address and the Layer 2 MAC address of the client as the destination.

DHCP Request (DHCPREQUEST)
When the client receives the DHCPOFFER from the server, it sends back a DHCPREQUEST message as shown in the Figure below. This message is used for both lease origination and lease renewal. When used for lease origination, the DHCPREQUEST serves as a binding acceptance notice to the selected server for the parameters it has offered and an implicit decline to any other servers that may have provided the client a binding offer.

DHCP Acknowledgment (DHCPACK)

On receiving the DHCPREQUEST message, the server verifies the lease information with an ICMP ping to that address to ensure it is not being used already, creates a new ARP entry for the client lease, and replies with a unicast DHCPACK message as shown in the Figure below.



Lease Renewal

DHCP Request (DHCPREQUEST)

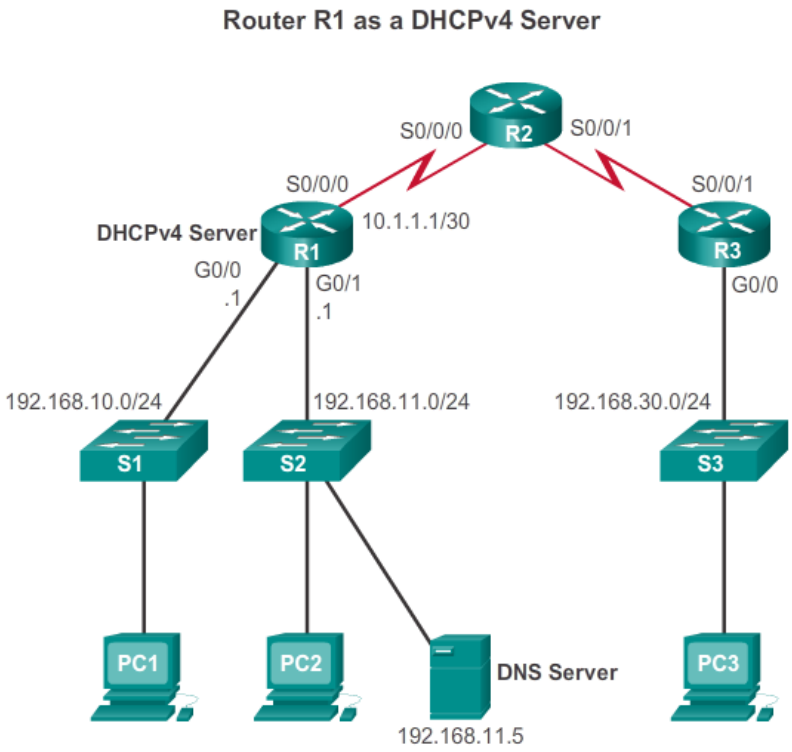
As shown in the Figure below, when the lease has expired, the client sends a DHCPREQUEST message directly to the DHCPv4 server that originally offered the IPv4 address. If a DHCPACK is not received within a specified amount of time, the client broadcasts another DHCPREQUEST so that one of the other DHCPv4 servers can extend the lease.

DHCP Acknowledgment (DHCPACK)

On receiving the DHCPREQUEST message, the server verifies the lease information by returning a DHCPACK, as shown below.

Configuring a Basic DHCPv4 Server

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. The topology shown below is used to illustrate this functionality.



Step 1. Excluding IPv4 Addresses

The router functioning as the DHCPv4 server assigns all IPv4 addresses in a DHCPv4 address pool unless configured to exclude specific addresses. Typically, some IPv4 addresses in a pool are assigned to network devices that require static address assignments. Therefore, these IPv4 addresses should not be assigned to other devices. To exclude specific addresses, use the **ip dhcp excluded-address** command.

A single address or a range of addresses can be excluded by specifying the low-address and high-address of the range. Excluded addresses should include the addresses assigned to routers, servers, printers, and other devices that have been manually configured.

Step 2. Configuring a DHCPv4 Pool

Configuring a DHCPv4 server involves defining a pool of addresses to assign. The **ip dhcp pool** *pool-name* command creates a pool with the specified name and puts the router in DHCPv4 configuration mode, which is identified by this prompt Router(dhcp-config)#.

Step 3. Configuring Specific Tasks

The figure below lists the tasks to complete the DHCPv4 pool configuration. Some of these are optional, while others must be configured. The address pool and default gateway router must be configured. Use the **network** statement to define the range of available addresses.

Use the **default-router** command to define the default gateway router. Typically, the gateway is the LAN interface of the router closest to the client devices. One gateway is required, but you can list up to eight addresses if there are multiple gateways.

Other DHCPv4 pool commands are optional. For example, the IPv4 address of the DNS server that is available to a DHCPv4 client is configured using the **dns-server** command. The **domain-name** *domain* command is used to define the domain name. The duration of the DHCPv4 lease can be changed using the **lease** command. The default lease value is one day.

```
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#
```

Disabling DHCPv4

The DHCPv4 service is enabled, by default, on versions of Cisco IOS software that support it. 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. Enabling the service has no effect if the parameters are not configured.

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. The second command, **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. As seen in the figure below in the output for these commands, currently there are no bindings and the statistics indicate no messages sent or received. At this point no devices have requested DHCPv4 services from router R1.

Before DHCPv4: `show ip dhcp` Commands

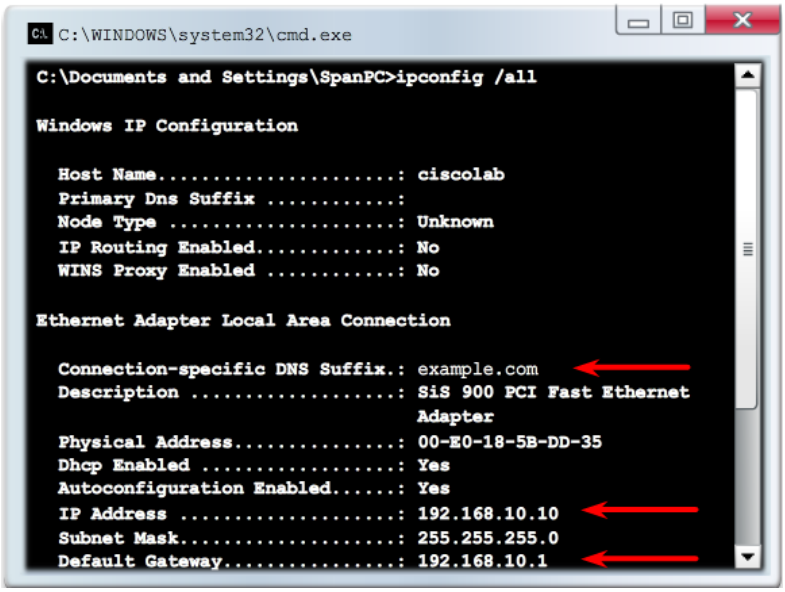
```
R1# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address     Client-ID/          Lease expiration   Type
               Hardware address/
               User name

R1# show ip dhcp server statistics
Memory usage         23543
Address pools        1
Database agents      0
Automatic bindings   0
Manual bindings      0
Expired bindings     0
Malformed messages   0
Secure arp entries   0

Message              Received
BOOTREQUEST          0
DHCPDISCOVER         0
DHCPREQUEST          0
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0

Message              Sent
BOOTREPLY            0
DHCPOFFER            0
```

The commands are issued after PC1 and PC2 have been powered on and have completed the booting process. Notice that the binding information now displays that the IPv4 addresses of 192.168.10.10 and 192.168.11.10 have been bound to MAC addresses. The statistics are also displaying DHCPDISCOVER, DHCPREQUEST, DHCPOFFER, and DHCPACK activity.

After DHCPv4: `show ip dhcp` Commands

```
R1# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/          Lease expiration      Type
                Hardware address/
                User name
192.168.10.10   0100.e018.5bdd.35   May 28 2013 01:06 PM Automatic
192.168.11.10   0100.b0d0.d817.e6   May 28 2013 01:10 PM Automatic


R1# show ip dhcp server statistics
Memory usage          25307
Address pools         2
Database agents       0
Automatic bindings    2
Manual bindings       0
Expired bindings      0
Malformed messages    0
Secure arp entries    0

Message               Received
BOOTREQUEST           0
DHCPDISCOVER          8
DHCPREQUEST           3
DHCPDECLINE           0
DHCPRELEASE           0
DHCPINFORM            0
```

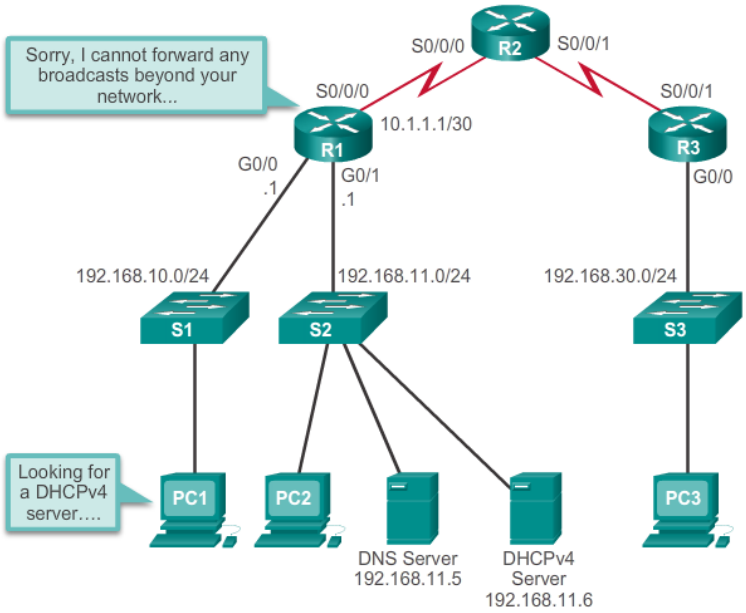
The **ipconfig /all** command shown below, when issued on PC1, displays the TCP/IP parameters. Because PC1 was connected to the network segment 192.168.10.0/24, it automatically received a DNS suffix, IPv4 address, subnet mask, default gateway, and DNS server address from that pool. No router interface 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.



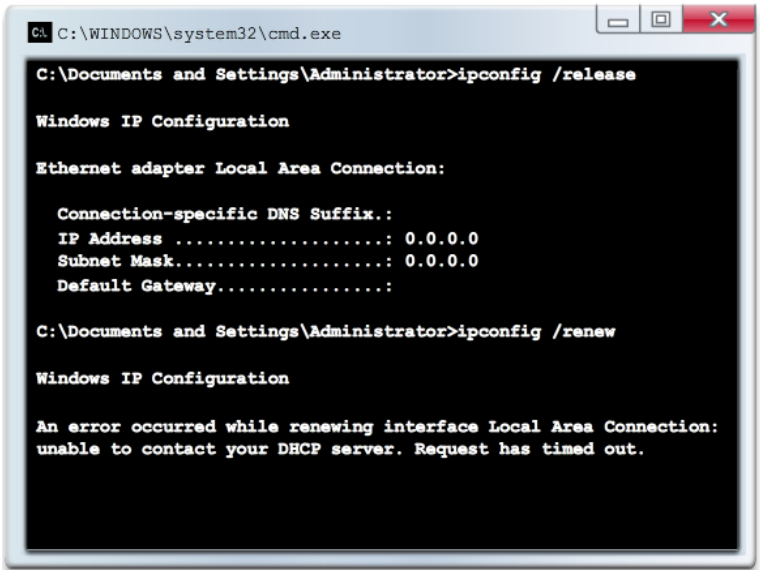
What is DHCP Relay?

In a complex hierarchical network, enterprise servers are usually located in a server farm. 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 below, PC1 is attempting to acquire an IPv4 address from a DHCP server using a broadcast message. In this scenario, router 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.



In the figure below, PC1 is attempting to renew its IPv4 address. To do so, the **ipconfig /release** command is issued. Notice that the IPv4 address is released and the address is shown to be 0.0.0.0. Next, the **ipconfig /renew** command is issued. This command causes PC1 to broadcast a DHCPDISCOVER message. The output shows that PC1 is unable to locate the DHCPv4 server. Because routers do not forward broadcasts, the request is not successful.



As a solution to this problem, an administrator can add DHCPv4 servers on all the subnets. However, running these services on several computers creates additional cost and administrative overhead.

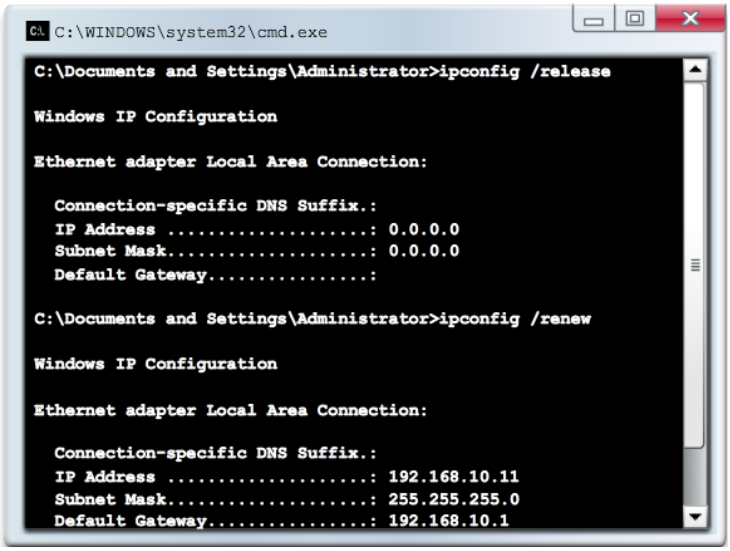
A better solution is to configure a Cisco IOS helper address. This solution enables a router to forward DHCPv4 broadcasts to the DHCPv4 server. When a router forwards address assignment/parameter requests, it is acting as a DHCPv4 relay agent. In the example topology, PC1 would broadcast a request to locate a DHCPv4 server. If R1 was configured as a DHCPv4 relay agent, it would forward the request to the DHCPv4 server located on subnet 192.168.11.0.

As shown below, the interface on R1 receiving the broadcast is configured with the **ip helper-address** interface configuration mode command. The address of the DHCPv4 server is configured as the only parameter.

```
R1(config)# interface g0/0
R1(config-if)# ip helper-address 192.168.11.6
R1(config-if)# end
R1# show ip interface g0/0
GigabitEthernet0/0 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>
```

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 **show ip interface** command is used to verify the configuration.

As shown below, PC1 is now able to acquire an IPv4 address from the DHCPv4 server.



Configure DHCPv4 Client

Sometimes, Cisco routers in small office/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 below, 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 G0/1 interface is configured with the **ip address dhcp** command, 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 g0/1
SOHO(config-if)# ip address dhcp
SOHO(config-if)# no shutdown
SOHO(config-if)#
*Jan 31 17:31:11.507: %DHCP-6-ADDRESS_ASSIGN: Interface
GigabitEthernet0/1 assigned DHCP address 209.165.201.12, mask
255.255.255.224, hostname SOHO
SOHO(config-if)# end
SOHO# show ip interface g0/1
GigabitEthernet0/1 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>
```

Procedures:

Dear students, please note that the lab problems sheet, the packet tracer activities and the practical discussion videos have been uploaded on your Microsoft Teams group. You are required to carefully study this experiment and then complete the lab sheet.

References

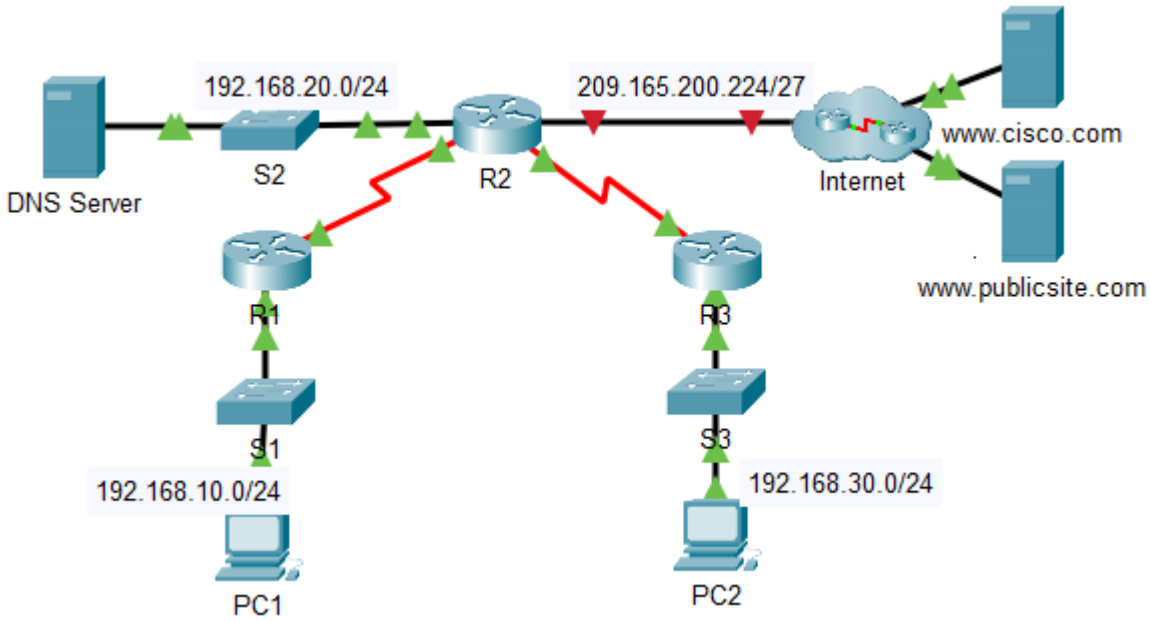
Cisco Networking Academy - CCNA: Switching, Routing, and Wireless Essentials.
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.8 Dynamic Host Configuration Protocol (DHCP)
Lab sheet

Problem 1: Configuring DHCP Using Cisco IOS

A dedicated DHCP server is scalable and relatively easy to manage, but can be costly to have one at every location in a network. However, a Cisco router can be configured to provide DHCP services without the need for a dedicated server.

As the network technician for your company, you are tasked with configuring a Cisco router as a DHCP server to provide dynamic allocation of addresses to clients on the network. You are also required to configure the edge router as a DHCP client so that it receives an IP address from the ISP network.



Task 1: Configure a Router as a DHCP Server

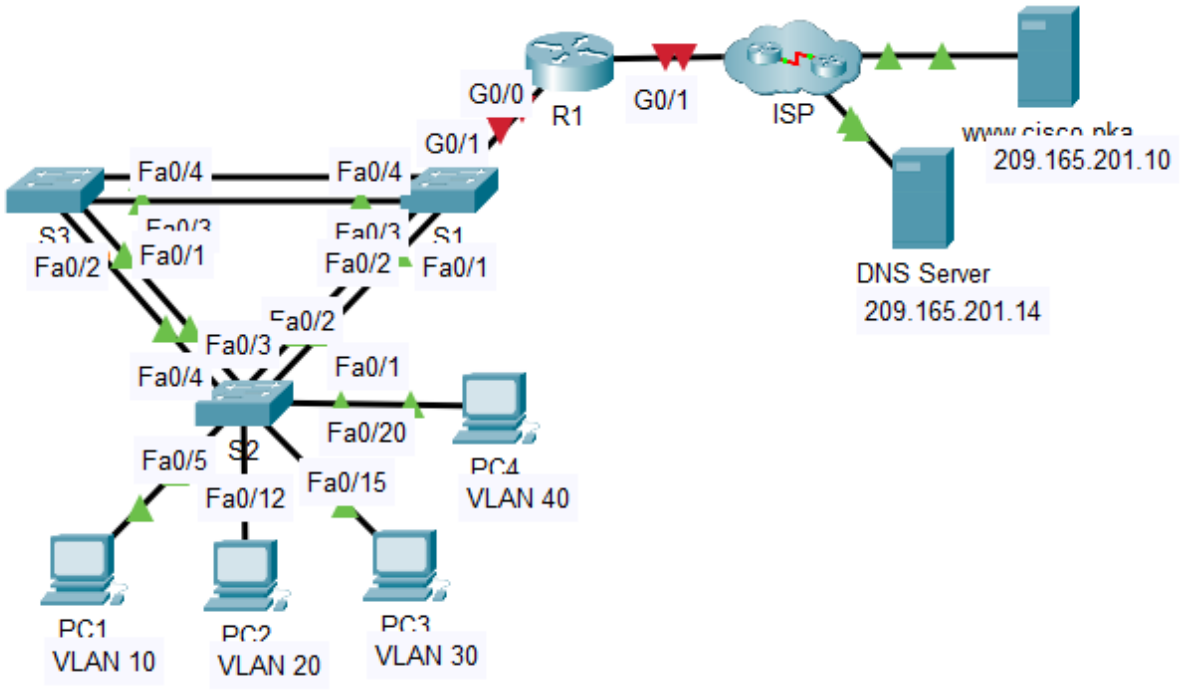
Task 2: Configure DHCP Relay

Task 3: Configure R2 as a DHCP Client

Task 4: Verify DHCP and Connectivity

Problem 2: Skills Integration Challenge

In this culminating activity, you will configure VLANs, trunks, DHCP Server, DHCP relay agents, and configure a router as a DHCP client.



Task 1: configure VLANs and trunks

Task 2: Configure DHCP Server, DHCP relay agents, and configure a router as a DHCP client.

The University of Jordan (UJ)
School of Engineering
Department of Computer Engineering
Advanced Networks Lab 0907529
Exp.9 Secure Remote Access and Virtual Private Network (VPN)

Objectives

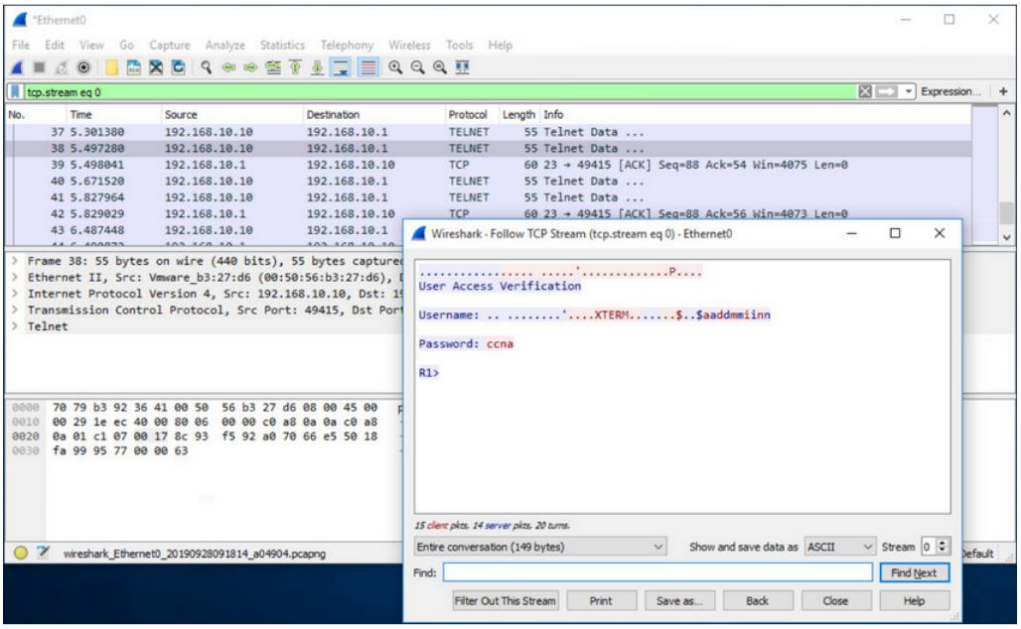
1. Discuss how to configure Secure Shell (SSH) for remote access.
2. Describe benefits of VPN technology.
3. Describe different types of VPNs.
4. Explain how the IPsec framework is used to secure network traffic.

Secure Remote Access using SSH

1. Introduction

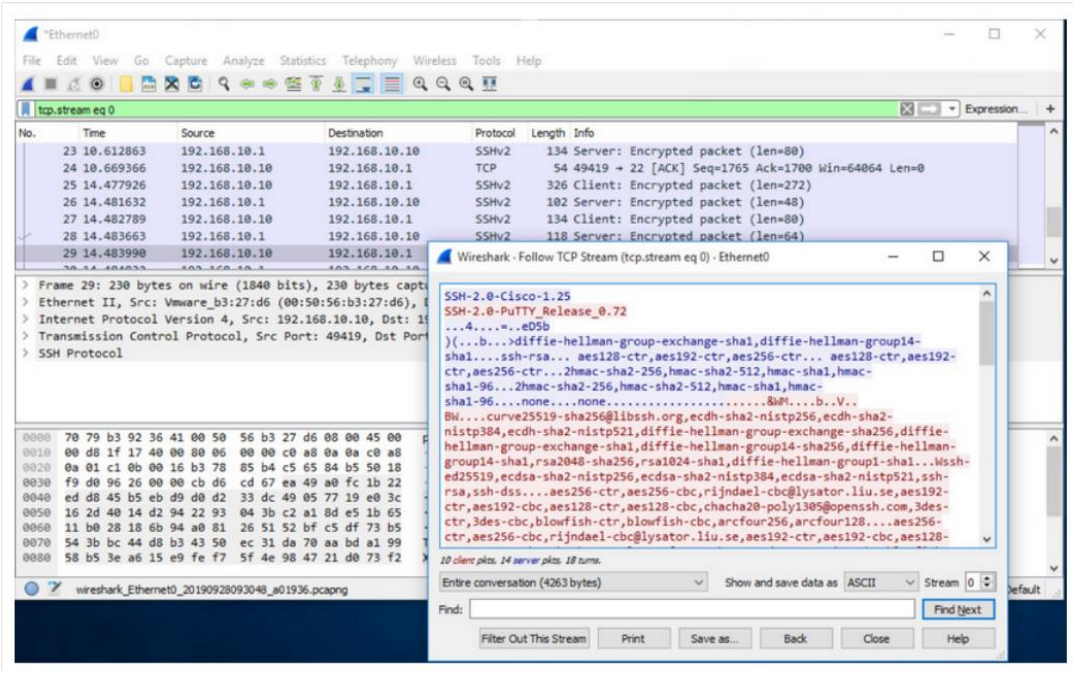
You might not always have direct access to your switch or router when you need to configure it. You need to be able to access it remotely and it is imperative that your access is secure. This topic discusses how to configure Secure Shell (SSH) for remote access.

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 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. For example, 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.



2. SSH Configuration

Before configuring SSH, the switch must be minimally configured with a unique hostname and the correct network connectivity settings.

Step 1:Verify SSH support.
Use the ***show ip ssh*** command to verify that the switch supports SSH. If the switch is not running an IOS that supports cryptographic features, this command is unrecognized. If the IOS filename includes the combination "k9" then it supports cryptographic (encrypted) features and capabilities.

```
S1# show ip ssh
```

Step 2: Configure the IP domain.
Configure the IP domain name of the network using the ***ip domain-name domain-name*** global configuration mode command. In the figure, the domain-name value is ***cisco.com***.

```
S1(config)# ip domain-name cisco.com
```

Step 3:Generate RSA key pairs.
Not all versions of the IOS default to SSH version 2, and SSH version 1 has known security flaws. To configure SSH version 2, issue the ***ip ssh version 2*** global configuration mode command. Generating an RSA key pair automatically enables SSH. Use the ***crypto key generate rsa*** global configuration mode command to enable the SSH server on the switch and generate an RSA key pair. When generating RSA keys, the administrator is prompted to enter a modulus length. The sample configuration in the figure uses a modulus size of 1,024 bits. A longer modulus length is more secure, but it takes longer to generate and to use.

```
S1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
```

Step 4: Configure user authentication.
The SSH server can authenticate users locally or using an authentication server. To use the local authentication method, create a username and password pair using the ***username*** *username* ***secret*** *password* global configuration mode command. In the example, the user admin is assigned the password ccna.

```
S1(config)# username admin secret ccna
```

Step 5: Configure the vty lines.
Enable the SSH protocol on the vty lines by using the ***transport input ssh*** line configuration mode command. The Catalyst 2960 has vty lines ranging from 0 to 15. This configuration prevents non-SSH (such as Telnet) connections and limits the switch to accept only SSH connections. Use the ***line vty*** global configuration mode command and then the ***login local*** line configuration mode command to require local authentication for SSH connections from the local username database.

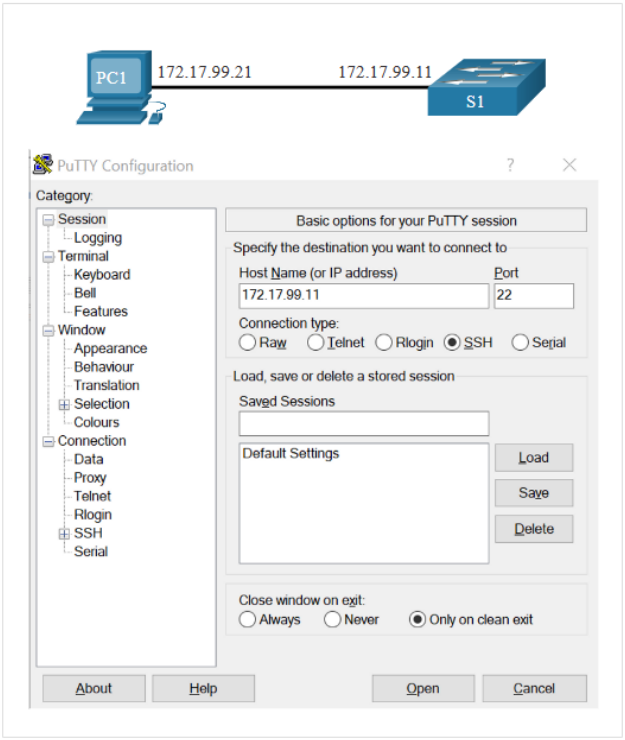
```
S1(config)# line vty 0 15
S1(config-line)# transport input ssh
S1(config-line)# login local
S1(config-line)# exit
```

Step 6: Enable SSH version 2.
By default, SSH supports both versions 1 and 2. When supporting both versions, this is shown in the ***show ip ssh*** output as supporting version 2. Enable SSH version using the ***ip ssh version 2*** global configuration command.

```
S1(config)# ip ssh version 2
```

3. Verify SSH is Operational

On a PC, an SSH client such as PuTTY, is used to connect to an SSH server. For example, assume the following is configured: SSH is enabled on switch S1, Interface VLAN 99 (SVI) with IPv4 address 172.17.99.11 on switch S1 and PC1 with IPv4 address 172.17.99.21
The figure shows the PuTTY settings for PC1 to initiate an SSH connection to the SVI VLAN IPv4 address of S1.



When connected, the user is prompted for a username and password as shown in the example. Using the configuration from the previous example, the username admin and password ccna are entered. After entering the correct combination, the user is connected via SSH to the command line interface (CLI) on the Catalyst 2960 switch.

```
Login as: admin
Using keyboard-interactive
Authentication.
Password:
S1> enable
Password:
S1#
```

To display the version and configuration data for SSH on the device that you configured as an SSH server, use the show ip ssh command. In the example, SSH version 2 is enabled.

```
S1# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
To check the SSH connections to the device, use the show ssh command as shown.
S1# show ssh
%No SSHv1 server connections running.
Connection Version Mode Encryption  Hmac               State          Username
0          2.0     IN   aes256-cbc  hmac-sha1    Session started       admin
0          2.0     OUT  aes256-cbc  hmac-sha1    Session started       admin
S1#
```

Virtual Private Network (VPN)

1. Introduction

Have you, or someone you know, ever been hacked while using public WiFi? It's surprisingly easy to do. But there is a solution to this problem: Virtual Private Networks (VPNs) and the additional protection of IP Security (IPsec). VPNs are commonly used by remote workers around the globe. There are also personal VPNs that you can use when you are on public WiFi. In fact, there are many different kinds of VPNs using IPsec to protect and authenticate IP packets between their source and destination.

2. VPN Benefits

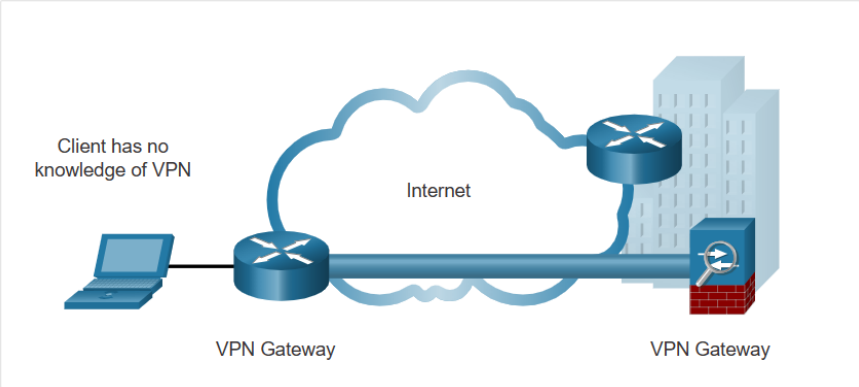
Major benefits of VPNs are shown in the table.

Benefit	Description
Cost Savings	With the advent of cost-effective, high-bandwidth technologies, organizations can use VPNs to reduce their connectivity costs while simultaneously increasing remote connection bandwidth.
Security	VPNs provide the highest level of security available, by using advanced encryption and authentication protocols that 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 all the popular broadband technologies. Remote workers can take advantage of these high-speed connections to gain secure access to their corporate networks.

3. VPN Types

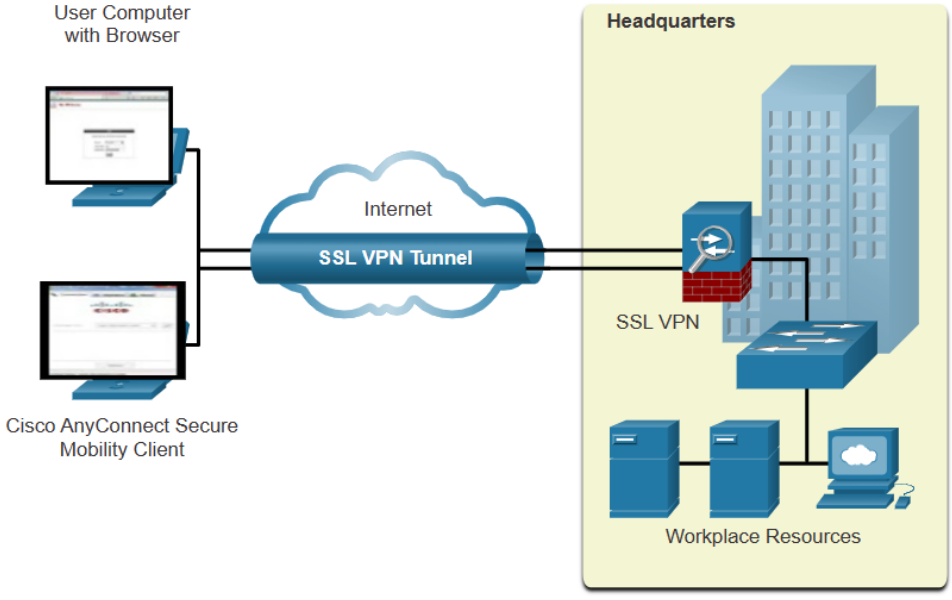
VPNs are commonly deployed in one of the following configurations: site-to-site or remote-access.

A site-to-site VPN is created when VPN terminating devices, also called VPN gateways, are preconfigured with information to establish a secure tunnel. VPN traffic is only encrypted between these devices. Internal hosts have no knowledge that a VPN is being used.



A remote-access VPN is dynamically created to establish a secure connection between a client and a VPN terminating device. As shown in the figure, remote-access VPNs let remote and mobile users securely connect to the enterprise by creating an encrypted tunnel. Remote users can securely replicate their enterprise security access including email and network applications. Remote-access VPNs also allow contractors and partners to have limited access to the specific servers, web pages, or files as required. This means that these users can contribute to business productivity without compromising network security.

Remote-access VPNs are typically enabled dynamically by the user when required. Remote access VPNs can be created using either IPsec or SSL. The figure displays two ways that a remote user can initiate a remote access VPN connection: clientless VPN (using web browser) and client-based VPN (using Cisco AnyConnect software).



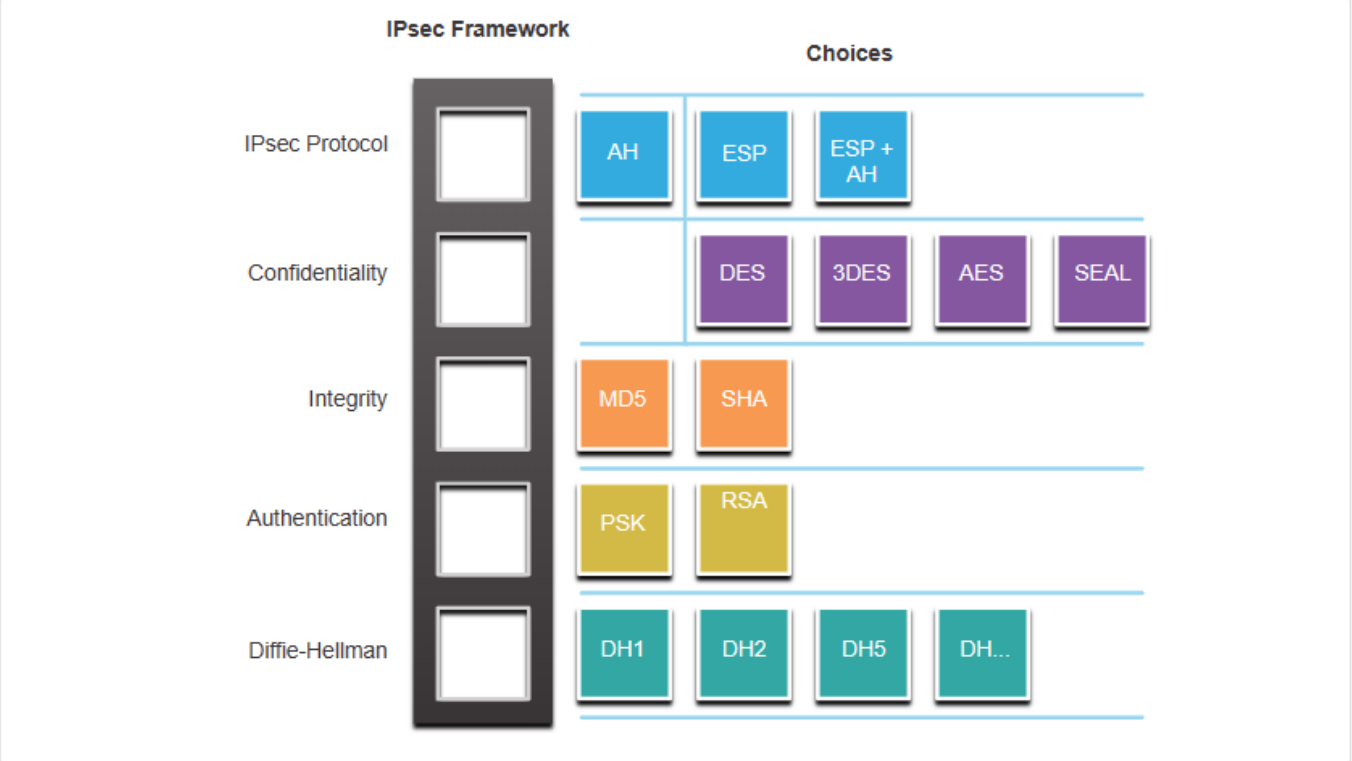
4. Site-to-Site IPsec VPNs

Site-to-site VPNs are used to connect networks across another untrusted network such as the internet. The VPN gateway encapsulates and encrypts outbound traffic. It then sends the traffic through a VPN tunnel over the internet to a VPN gateway at the target site. Upon receipt, the receiving VPN gateway strips the headers, decrypts the content, and relays the packet toward the target host inside its private network. Site-to-site VPNs are typically created and secured using IP security (IPsec).

Other VPN tunneling protocols are Generic Routing Encapsulation (GRE), Dynamic Multipoint VPN (DMVPN) and IPsec Virtual Tunnel Interface (VTI). We will not discuss these types in this experiment.

IPsec protects and authenticates IP packets between source and destination. IPsec can protect traffic from Layer 4 through Layer 7. Using the IPsec framework, IPsec provides these essential security functions:

- **Confidentiality** - IPsec uses encryption algorithms to prevent cybercriminals from reading the packet contents.
- **Integrity** - IPsec uses hashing algorithms to ensure that packets have not been altered between source and destination.
- **Origin authentication** - IPsec uses the Internet Key Exchange (IKE) protocol to authenticate source and destination. Methods of authentication including using pre-shared keys (passwords), digital certificates, or RSA certificates.
- **Diffie-Hellman** - Secure key exchange typically using various groups of the DH algorithm.



IPsec is not bound to any specific rules for secure communications. This flexibility of the framework allows IPsec to easily integrate new security technologies without updating the existing IPsec standards. The currently available technologies are aligned to their specific security function. The open slots shown in the IPsec framework 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).

The security functions are listed in the table.

IPsec Function	Description
IPsec Protocol	The choices for IPsec Protocol include Authentication Header (AH) or Encapsulation Security Protocol (ESP). AH authenticates the Layer 3 packet but doesn't use any encryption method. ESP encrypts the Layer 3 packet. **Note**: ESP+AH is rarely used as this combination will not successfully traverse a NAT device.
Confidentiality	Encryption ensures confidentiality of the Layer 3 packet. Choices include Data Encryption Standard (DES), Triple DES (3DES), Advanced Encryption Standard (AES), or Software-Optimized Encryption Algorithm (SEAL). No encryption is also an option.
Integrity	Ensures that data arrives unchanged at the destination using a hash algorithm, such as message-digest 5 (MD5) or Secure Hash Algorithm (SHA).
Authentication	IPsec uses Internet Key Exchange (IKE) to authenticate users and devices that can carry out communication independently. IKE uses several types of authentication, including username and password, one-time password, biometrics, pre-shared keys (PSKs), and digital certificates using the Rivest, Shamir, and Adleman (RSA) algorithm.
Diffie-Hellman	IPsec uses the DH algorithm to provide a public key exchange method for two peers to establish a shared secret key. There are several different groups to choose from including DH14, 15, 16 and DH 19, 20, 21 and 24. DH1, 2 and 5 are no longer recommended.

5. IPsec Configuration

In this part, we will show you how to configure two Cisco IOS routers to use IPSec in Tunnel mode.

Step 1: ACL

To define interesting traffic, configure each router with an ACL to permit traffic from the local LAN to the remote LAN, as shown in the following examples for R1 and R2. The ACL will be used in the crypto map configuration to specify what traffic will trigger the start of IKE Phase 1.


```
R1(config)# access-list 101 permit ip 10.0.1.0 0.0.0.255 192.168.1.0 0.0.0.255
R2(config)# access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
```

Step 2: Configure ISAKMP Policy (Phase 1)

To configure a new ISAKMP policy, use the **crypto isakmp policy** command, as shown in the figure. The only argument for the command is to set a priority for the policy (from 1 to 10000). Peers will attempt to negotiate using the policy with the lowest number (highest priority). Peers do not require matching priority numbers.

When in ISAKMP policy configuration mode, the SAs for the IKE Phase 1 tunnel can be configured. Use the mnemonic **HAGLE** to remember the five SAs to configure:

Hash
Authentication
Group
Lifetime
Encryption

The example below shows the ISAKMP policy configuration. Use the **show crypto isakmp policy** command to verify the configuration. R2 has an equivalent configuration.

```
R1(config)# crypto isakmp policy 1
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# hash sha
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 24
R1(config-isakmp)# lifetime 3600
R1(config-isakmp)# end
R1# show crypto isakmp policy

Global IKE policy
Protection suite of priority 1
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #24 (2048 bit, 256 bit subgroup)
        lifetime:               3600 seconds, no volume limit
R1#
```

```
R2(config)# crypto isakmp policy 1
R2(config-isakmp)# hash sha
R2(config-isakmp)# authentication pre-share
R2(config-isakmp)# group 24
R2(config-isakmp)# lifetime 3600
R2(config-isakmp)# encryption aes 256
```

Security policy requires that a pre-shared key be used for authentication between the peers. The administrator can either specify a host name or an IP address for the peer. The command syntax is shown below.

```
Router(config)# crypto isakmp key keystring address peer-address
Router(config)# crypto isakmp key keystring hostname peer-hostname
```

```
R1# conf t
R1(config)# crypto isakmp key cisco12345 address 172.30.2.2
R1(config)#
```

```
R2# conf t
R2(config)# crypto isakmp key cisco12345 address 172.30.2.1
R2(config)#
```

Step 3: Configure the Transform Set (Phase 2)

The next step is to configure the set of encryption and hashing algorithms that will be used to transform the data sent through the IPsec tunnel. This is called the transform set. During IKE Phase 2 negotiations, the peers agree on the IPsec transform set to be used for protecting interesting traffic.

Configure a transform set using the **crypto ipsec transform-set** command, as shown here. First, specify a name for the transform set. After the transform set is named, the encryption and hashing algorithm can be configured in either order.

```
R1(config)# crypto ipsec transform-set R1-R2 esp-aes esp-sha-hmac
R1(config)#
```

```
R2(config)# crypto ipsec transform-set R1-R2 esp-aes esp-sha-hmac
R2(config)#
```

Step 4: Configure IPsec Crypto Map (tie it together)

Now that the interesting traffic is defined, and an IPsec transform set is configured, it is time to bind those configurations with the rest of the IPsec policy in a crypto map. The syntax to start a crypto map set is shown below. The sequence number is important when configuring multiple crypto map entries.

To finish the configuration to meet the IPsec security policy, complete the following:

Step 1. Bind the ACL and the transform set to the map.
Step 2. Specify the peer's IP address.
Step 3. Configure the DH group.
Step 4. Configure the IPsec tunnel lifetime.

The crypto map configurations for R1 and R2 are shown below. The map name is **R1-R2_MAP**, and the sequence number is **10**.

```
R1(config)# crypto map R1-R2_MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R1(config-crypto-map)# match address 101
R1(config-crypto-map)# set transform-set R1-R2
R1(config-crypto-map)# set peer 172.30.2.2
R1(config-crypto-map)# set pfs group24
R1(config-crypto-map)# set security-association lifetime seconds 900
R1(config-crypto-map)# exit
R1(config)#
```

```
R2(config)# crypto map R1-R2_MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R2(config-crypto-map)# match address 102
R2(config-crypto-map)# set transform-set R1-R2
R2(config-crypto-map)# set peer 172.30.2.1
R2(config-crypto-map)# set pfs group24
R2(config-crypto-map)# set security-association lifetime seconds 900
R2(config-crypto-map)# exit
R2(config)#
```

Use the **show crypto map** command to verify the crypto map configuration, as shown below for R1. All the required SAs should be in place. Notice that the output shows that no interfaces are currently using the crypto map.

```
R1# show crypto map
Crypto Map IPv4 "R1-R2_MAP" 10 ipsec-isakmp
    Peer = 172.30.2.2
    Extended IP access list 101
        access-list 101 permit ip 10.0.1.0 0.0.0.255 192.168.1.0 0.0.0.255
    Security association lifetime: 4608000 kilobytes/900 seconds
    Responder-Only (Y/N): N
    PFS (Y/N): Y
    DH group:  group24
    Mixed-mode : Disabled
    Transform sets={
            R1-R2:  { esp-aes esp-sha-hmac  } ,
    }
    Interfaces using crypto map R1-R2_MAP:

R1#
```

Step 5: Applying Crypto Maps to Interfaces

To apply the crypto map, enter interface configuration mode for the outbound interface and configure the **crypto map** *map-name* command. Below is the configuration. Notice the **show crypto map** output now displays that the Serial 0/0/0 interface is using the crypto map. R2 is configured with the same command on its Serial 0/0/0 interface.

```
R1(config)# interface serial0/0/0
R1(config-if)# crypto map R1-R2_MAP
R1(config-if)#
*Mar 19 19:36:36.273: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config-if)# end
R1# show crypto map
Crypto Map IPv4 "R1-R2_MAP" 10 ipsec-isakmp
          Peer = 172.30.2.2
          Extended IP access list 101
              access-list 101 permit ip 10.0.1.0 0.0.0.255 192.168.1.0 0.0.0.255
          Security association lifetime: 4608000 kilobytes/900 seconds
          Responder-Only (Y/N): N
          PFS (Y/N): Y
          DH group:  group24
          Mixed-mode : Disabled
          Transform sets={
                  R1-R2:  { esp-aes esp-sha-hmac  } ,
          }
          Interfaces using crypto map R1-R2_MAP:
                  Serial0/0/0
```

Now that both the ISAKMP and IPsec policies are configured, and the crypto map is applied to the appropriate outbound interfaces, test the two tunnels by sending interesting traffic across the link.

Traffic from the LAN interface on R1 that is destined for the LAN interface on R2 is considered interesting traffic because it matches the ACLs configured on both routers. An extended ping from R1 will effectively test the VPN configuration. The extended ping command syntax and results are shown below. The first ping failed because it takes a few milliseconds to establish the ISAKMP and IPsec tunnels.

```
R1# ping 192.168.1.1 source 10.0.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.1
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/4 ms
R1#
```

Sending interesting traffic does not actually mean that the tunnels are established. R1 and R2 will route traffic between the two LANs even if the ISAKMP and IPsec policy configurations are wrong. To verify that tunnels have been established, use the **show crypto isakmp sa** and **show crypto ipsec sa** commands. In the output below, notice that the tunnel is active between the two peers, 172.30.2.1 and 172.30.2.2, and that they are using the R1-R2_MAP crypto map.

```
R1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state      conn-id status
172.30.2.2   172.30.2.1   QM_IDLE       1005 ACTIVE

R1# show crypto ipsec sa

interface: Serial0/0/0
    Crypto map tag: R1-R2_MAP, local addr 172.30.2.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   current_peer 172.30.2.2 port 500
```

Procedures:

Dear students, please note that the lab problems sheet, the packet tracer activities and the practical discussion videos have been uploaded on your Microsoft Teams group. You are required to carefully study this experiment and then complete the lab sheet.

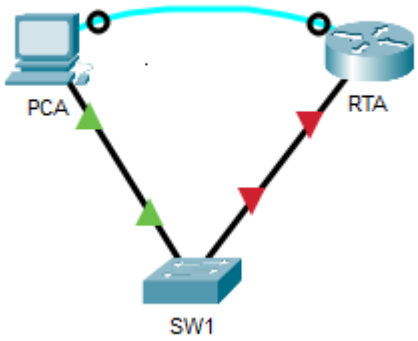
References

Network Security - Cisco Networking Academy
https://www.netacad.com

Advanced Networks Lab 0907529
Exp.9 Secure Remote Access and Virtual Private Network (VPN)
Lab sheet

Problem 1: Configure Secure Passwords and SSH

SSH should replace Telnet for management connections. Telnet uses insecure plain text communications. SSH provides security for remote connections by providing strong encryption of all transmitted data between devices. In this activity, the network administrator has asked you to prepare RTA and SW1 for deployment. Before they can be connected to the network, security measures must be enabled.



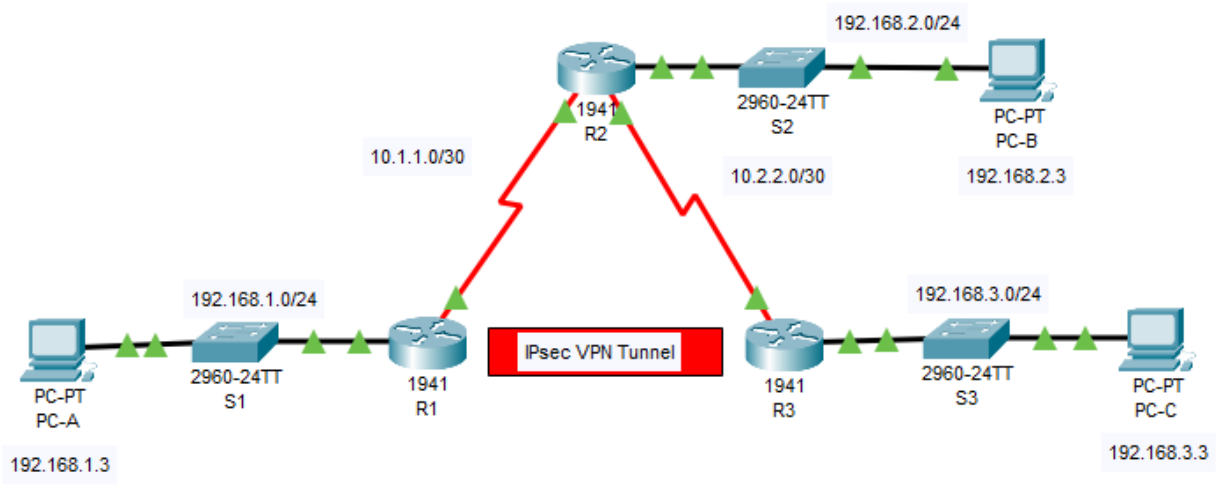
Task 1: Configure Basic Security on the Router

Task 2: Configure Basic Security on the Switch

Task 3: Verify SSH Implementation

Problem 2: Configure and Verify a Site-to-Site IPsec VPN

The network topology shows three routers. Your task is to configure R1 and R3 to support a site-to-site IPsec VPN when traffic flows between their respective LANs. The IPsec VPN tunnel is from R1 to R3 via R2. R2 acts as a pass-through and has no knowledge of the VPN. IPsec provides secure transmission of sensitive information over unprotected networks, such as the Internet. IPsec operates at the network layer and protects and authenticates IP packets between participating IPsec devices (peers), such as Cisco routers.



Task 1: Configure IPsec parameters on R1

Task 2: Configure IPsec Parameters on R3

Task 3: Verify the IPsec VPN