Chapter 1
Basic Device Configuration

Device Names

By default, all devices are assigned a factory default name. The problem is if all switches in a network were left with their default names, it would be difficult to identify a specific device. For instance, how would you know that you are connected to the right device when accessing it remotely using SSH? The hostname provides confirmation that you are connected to the correct device. The default name should be changed to something more descriptive. By choosing names wisely, it is easier to remember, document, and identify network devices.

Configure Hostname for Cisco

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

Configure Hostname for Juniper

[edit]
user@host# set system host-name Sw-Floor-1
user@host# commit

Configure Passwords

Configure Passwords for Cisco

When you initially connect to a device, you are in user EXEC mode. This mode is secured using the console.

To secure user EXEC mode access, enter line console configuration mode using the **line console 0** global configuration command, as shown in the example. The zero is used to represent the first (and in most cases the only) console interface. Next, specify the user EXEC mode password using the **password** *password* command. Finally, enable user EXEC access using the **login** command.

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

Console access will now require a password before allowing access to the user EXEC mode.

To have administrator access to all IOS commands including configuring a device, you must gain privileged EXEC mode access. It is the most important access method because it provides complete access to the device.

To secure privileged EXEC access, use the **enable secret** *password* global config command, as shown in the example.

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

Configure Passwords for Juniper

On Juniper devices (Junos), you don't configure a password **directly on the console port** like in Cisco. Instead, the console uses the root **or local user authentication**, so you secure console access by setting a password for a user (usually root).

When the device is powered on for the first time, it is ready to be configured. Initially, you log in as the user root with no password. You must configure a plain-text password for the root-level user (whose username is *root*) the first time you modify and commit the configuration. Configuring a plain-text password is one way to protect access to the root level by unauthorized users. If you forget the root password for the device, you can use the password recovery procedure to reset the root password.

Configure the Root Password

When you power on the router or switch, it is ready to be configured. Initially, you log in as the user root with no password. The root directory is the entry point to all other folders and files on that device. As a result, access to the root directory is restricted by default to a predefined user account known as the *root user*. The root user (also referred to as *superuser*) has unrestricted access and full permissions within the system. The expression "log in as root" is commonly used when an action requires the user to log in to the device as the root user.

```
[edit]
user@host# set system root-authentication plain-text-password
New password: type password here
Retype new password: retype password here
```

The default requirements for plain-text passwords are:

- The password must be between 6 and 128 characters long.
- You can include most character classes in a password (uppercase letters, lowercase letters, numbers, punctuation marks, and other special characters). Control characters are not recommended.
- Valid passwords must contain at least one uppercase letter or one lowercase letter, or one character class.

Save Configurations for Cisco

There are two system files that store the device configuration:

- **startup-config** - This is the saved configuration file that is stored in NVRAM. It contains all the commands that will be used by the device upon startup or reboot. Flash does not lose its contents when the device is powered off.
- **running-config** - This is stored in Random Access Memory (RAM). It reflects the current configuration. Modifying a running configuration affects the operation of a Cisco device immediately. RAM is volatile memory. It loses all of its content when the device is powered off or restarted.

The **show running-config** privileged EXEC mode command is used to view the running config. As shown in the example, the command will list the complete configuration currently stored in RAM.

To view the startup configuration file, use the **show startup-config** privileged EXEC command.

If power to the device is lost, or if the device is restarted, all configuration changes will be lost unless they have been saved. To save changes made to the running configuration to the startup configuration file, use the **copy running-config startup-config** privileged EXEC mode command.

Save Configurations for Juniper

Unlike Cisco, when you make configuration changes in Juniper, it is not automatically active and you need to use commit in configuration mode to activate changes, which saves them to juniper.conf.gz.

Configure IP Addresses

End devices in your network need an IP address so that they can communicate with other devices on your network. IPv4 address information can be entered into end devices manually, or automatically using Dynamic Host Configuration Protocol (DHCP).

Switch Virtual Interface Configuration on Cisco Devices

To access the switch remotely, an IP address and a subnet mask must be configured on the SVI. To configure an SVI on a switch, use the **interface vlan 1** global configuration command. Vlan 1 is not an actual physical interface but a virtual one. Next assign an IPv4 address using the **ip address** *ip-address subnet-mask* interface configuration command. Finally, enable the virtual interface using the **no shutdown** interface configuration command.

After these commands are configured, the switch has all the IPv4 elements ready for communication over the network.

Note: Similar to a Windows hosts, switches configured with an IPv4 address will typically also need to have a default gateway assigned. This can be done using the **ip default-gateway** *ip-address* global configuration command. The *ip-address* parameter would be the IPv4 address of the local router on the network, as shown in the example.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# interface vlan 1
Sw-Floor-1(config-if)# ip address 192.168.1.20 255.255.255.0
Sw-Floor-1(config-if)# no shutdown
Sw-Floor-1(config-if)# exit
Sw-Floor-1(config)# ip default-gateway 192.168.1.1
```

Configure Router Interfaces on Cisco Devices

There are many different types of interfaces available on Cisco routers. For example, the Cisco ISR 4321 router is equipped with two Gigabit Ethernet interfaces:

- **GigabitEthernet 0/0/0 (G0/0/0)**
- **GigabitEthernet 0/0/1 (G0/0/1)**

The task to configure a router interface is very similar to a management SVI on a switch. Specifically, it includes issuing the following commands:

```
Router(config)# interface type-and-number
Router(config-if)# ip address 192.168.1.20 255.255.255.0
Router(config-if)# no shutdown
```

There are several commands that can be used to verify interface configuration. The most useful of these is the **show ip interface brief**

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

Switch Virtual Interface Configuration on Juniper Devices

Configuring a **Switch Virtual Interface (SVI)** on a **Juniper switch** is different from Cisco, but the concept is similar: you assign an IP address to a VLAN so the switch can route traffic or be managed on that subnet. On Juniper, this is done by creating a **VLAN**, assigning it a VLAN interface (irb interface), and configuring the IP address

You first define the VLAN and associate it with L3 interface

set vlans VLAN10 vlan-id 10
set vlans VLAN10 l3-interface irb.10

Juniper uses irb interfaces for VLAN IP interfaces

set interfaces irb unit 10 family inet address 192.168.10.1/24

On a **Layer 2 switch**, the switch itself does not perform routing between VLANs (unless you configure it with Layer 3 capabilities). The **default gateway** is simply the IP address the switch uses to reach devices outside its local subnet—for management purposes.

To configure Default gateway on Juniper switches:

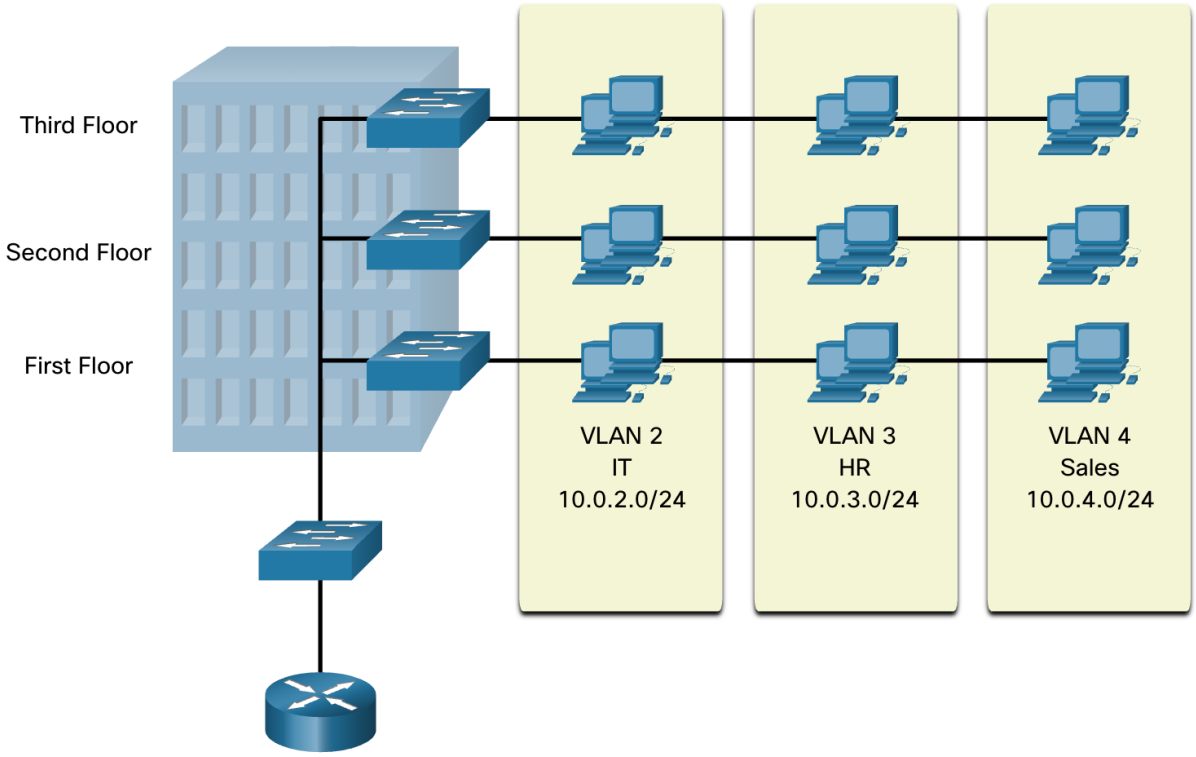
set routing-options static route 0.0.0.0/0 next-hop 192.168.10.1

On **Layer 2 switches**, the default gateway is only for **management traffic**. It doesn't make the switch route traffic between VLANs.

Chapter 2
Vlans

Segmenting a network into smaller parts isn't as straightforward as sorting screws into jars, but it significantly improves manageability. Virtual LANs (VLANs) enable both segmentation and flexible organization within a switched network. Devices in the same VLAN communicate as though they are connected to a single shared medium, even if they are physically distributed. This is because VLANs rely on logical grouping rather than physical connections.

As illustrated, VLANs allow users from different departments—such as IT, HR, and Sales—to be part of the same network, regardless of which switch they are connected to or where they are located within a campus environment.



VLANs enable administrators to divide a network based on criteria like function, department, or application, without being limited by the physical location of users or devices. Each VLAN operates as its own logical network, where devices behave as though they are part of a separate infrastructure, even when sharing the same physical hardware. Any switch port can be assigned to a VLAN.

Within a VLAN, unicast, broadcast, and multicast traffic is forwarded only to devices that belong to that VLAN. Traffic intended for devices outside the VLAN must pass through a routing-capable device.

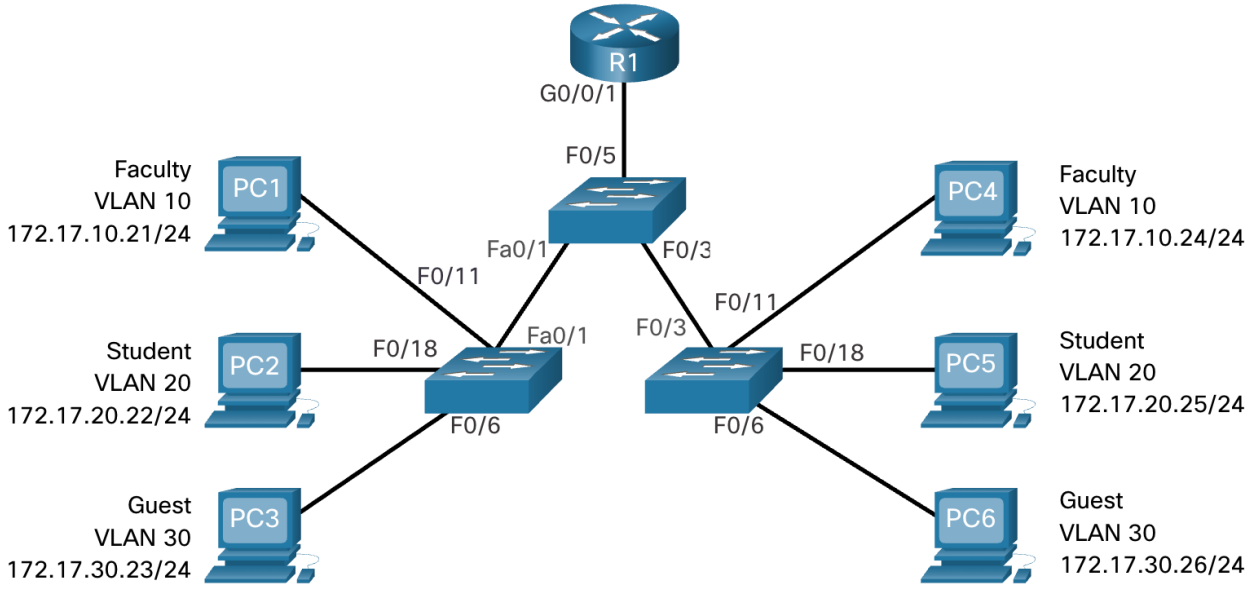
Although multiple IP subnets can exist on a switched network without VLANs, all devices would still share the same Layer 2 broadcast domain. As a result, broadcast traffic—such as ARP requests—would reach every device on the network, including those for which the traffic is irrelevant.

By creating separate logical broadcast domains, VLANs can span multiple physical network segments while improving overall performance. Broadcast traffic generated within one VLAN is confined to that VLAN, preventing it from reaching devices in other VLANs.

By using VLANs, network administrators can enforce access control and security policies based on defined user groups. Typically, each switch port is assigned to a single VLAN, with exceptions such as ports connected to IP phones or other switches.

Benefits of VLAN Design

In a switched network, each VLAN maps to a separate IP network. For this reason, VLAN design should align with a hierarchical IP addressing scheme. This approach assigns IP address ranges to specific network segments or VLANs in a structured way, considering the network as a whole. Contiguous blocks of IP addresses are reserved and applied to devices within particular areas of the network, as illustrated in the figure.



Benefit	Description
Smaller broadcast domains	• Dividing a network into VLANs reduces the number of devices in the broadcast domain. • In the figure, there are six computers in the network but only three broadcast domains (i.e., Faculty, Student, and Guest).
Improved security	• Only users in the same VLAN can communicate together. • Only users in the same VLAN can communicate without the services of a router. The router may have a security feature such as an access control list to restrict communications between VLANs.
Improved IT efficiency	• VLANs simplify network management because users with similar network requirements can be configured on the same VLAN. • VLANs can be named to make them easier to identify. • In the figure, VLAN 10 was named “Faculty”, VLAN 20 “Student”, and VLAN 30 “Guest.”
Reduced cost	VLANs reduce the need for expensive network upgrades and use the existing bandwidth and uplinks more efficiently, resulting in cost savings.
Better performance	Smaller broadcast domains reduce unnecessary traffic on the network and improve performance.
Simpler project and application management	• VLANs aggregate users and network devices to support business or geographic requirements. • Having separate functions makes managing a project or working with a specialized application easier; an example of such an application is an e-learning development platform for faculty.

VLAN Types in Cisco vs Juniper

VLAN Type	Cisco	Juniper (Junos)	Notes / Comments
Default VLAN	VLAN 1 (default VLAN on all ports)	VLAN `default` (ID 1 by default, but can be deleted or renamed)	Cisco VLAN 1 is special and cannot be deleted; Juniper treats it as normal VLAN
Data VLAN	User-defined VLANs for user data traffic	User-defined VLANs under `vlans` hierarchy	Both platforms use these for separating user traffic
Management VLAN	VLAN dedicated to management (e.g., VLAN 99)	VLAN assigned to management interface or IRB	Juniper assigns management IP to IRB interface within a VLAN
Native VLAN	Untagged VLAN on trunk ports (default VLAN 1)	VLAN configured via `native-vlan-id` on trunk ports	Untagged frames assigned to native VLAN; Juniper has no default native VLAN
Voice VLAN	Special VLAN assigned for VoIP traffic (`switchport voice vlan <id>`)	Voice VLAN configured with `voice-vlan` feature on interfaces (EX/QFX)	Both allow voice traffic separation and QoS
Extended VLANs	VLAN IDs 1006–4094 (requires VTP transparent mode)	VLAN IDs 1–4094 without distinction	Juniper treats all VLAN IDs the same; Cisco splits normal vs extended
Reserved VLANs	VLAN 0 and 4095 (reserved)	VLAN 0 and 4095 (reserved)	Both reserved per IEEE 802.1Q standard
Routed VLAN (SVI/IRB)	Switched VLAN with Layer 3 interface (SVI)	VLAN with associated IRB interface for routing	Layer 3 gateway for VLAN traffic

 Cisco explicitly defines VLAN types for operational clarity (default, voice, management, native, extended).

 Juniper treats VLANs more as generic Layer 2 domains, with specific functions assigned by configuration context (e.g., voice VLAN enabled per interface).

 Both platforms support **voice VLANs**, **management VLANs**, and **routed VLAN interfaces** (SVI in Cisco, IRB in Juniper).

 Native VLAN behavior exists on both, but Juniper requires explicit configuration.

 Extended VLANs are a Cisco-specific concept due to VTP design; Juniper does not differentiate

VLAN Trunks

A VLAN trunk is a network link that carries traffic for multiple VLANs at the same time between switches, routers, or other devices. Unlike an access port, which handles a single VLAN, a trunk port can transport frames from multiple VLANs by tagging them so the receiving device can identify the correct VLAN for each frame.

A trunk is not assigned to any single VLAN; instead, it acts as a pathway for multiple VLANs between network devices. It can also be used to connect to servers or other devices equipped with

an 802.1Q-capable network interface. On Cisco Catalyst switches, all VLANs are allowed on a trunk port by default. In contrast, on Juniper devices, you must explicitly configure which VLANs are permitted on the trunk interface, or allow all VLANs.

VLANs themselves are configured on switch ports, while the connected devices are unaware of VLANs. These devices are instead configured with IP addresses and belong to specific IP networks. This highlights the relationship between VLANs and IP networks: a VLAN effectively corresponds to an IP subnet. VLAN configuration is done on the switch, whereas IP addressing is configured on the end devices.

Vlan configuration on Cisco devices

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

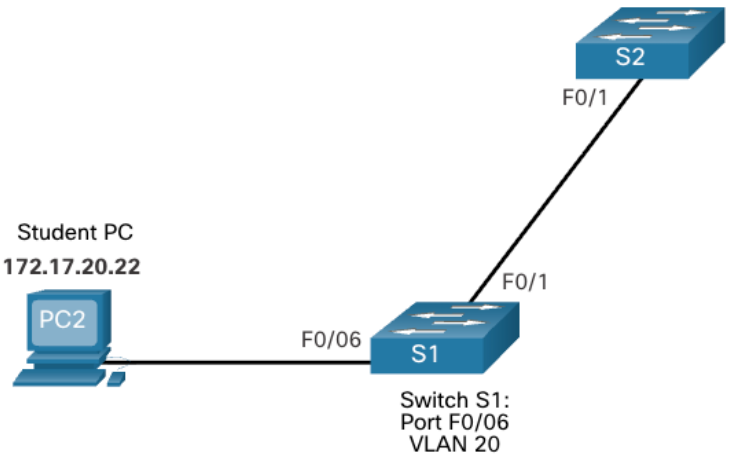
Note: In addition to entering a single VLAN ID, a series of VLAN IDs can be entered separated by commas, or a range of VLAN IDs separated by hyphens using the **vlan** *vlan-id* command. For example, entering the **vlan 100,102,105-107** global configuration command would create VLANs 100, 102, 105, 106, and 107.

After creating a VLAN, the next step is to assign ports to the VLAN.

The image 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.

```
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
```

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.



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).

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. This effectively places the switch into its factory default condition with regard to VLAN configurations.

Note: To restore a Catalyst switch to its factory default condition, unplug all cables except the console and power cable from the switch. Then enter the **erase startup-config** privileged EXEC mode command followed by the **delete vlan.dat** command.

Vlan configuration on Juniper devices

Create VLAN 10 (data) and assign it to an interface

```
root> configure
[edit]
root# set vlans data vlan-id 10
root# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
root# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members data
root# commit
```

delete vlans data vlan-id 10

```
root> configure
[edit]
root# delete vlans data vlan-id 10
root# commit
```

you can check vlans with commands:

show vlans
show vlans brief

Vlan Trunk configuration for Cisco

```
S1(config)# interface fastEthernet 0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk allowed vlan 10,20,30,99
S1(config-if)# end
```

If allowed vlan command is not issued, then all vlans will be permited on an interface.

Vlan Trunk configuration for Juniper

```
root> configure
[edit]
root# set vlans data vlan-id 10
root# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
root# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members all
root# commit
```

Chapter 3
Inter Vlan Routing

VLANs are used to segment switched Layer 2 networks for a variety of reasons. Regardless of the reason, hosts in one VLAN cannot communicate with hosts in another VLAN unless there is a router or a Layer 3 switch to provide routing services.

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

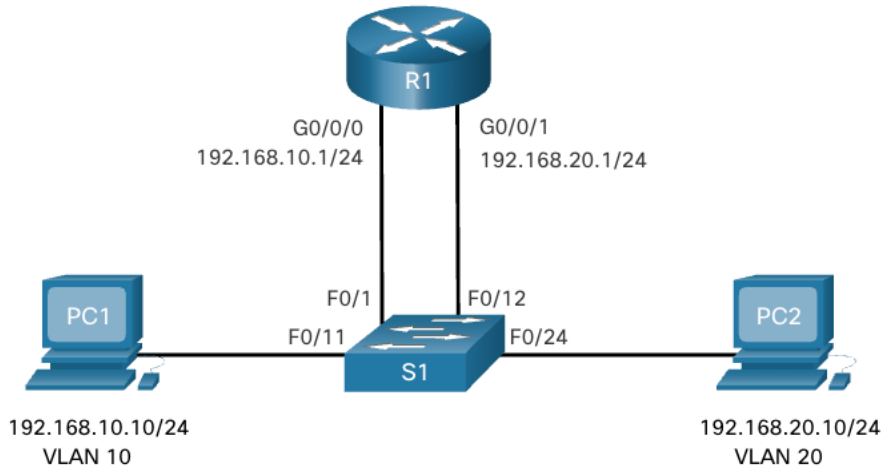
There are three inter-VLAN routing options:

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

Legacy Inter-VLAN Routing

The first inter-VLAN routing solution relied on using a router with multiple Ethernet interfaces. Each router interface was connected to a switch port in different VLANs. The router interfaces served as the default gateways to the local hosts on the VLAN subnet.

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.

MAC Address table for S1

Port	MAC Address	VLAN
F0/1	R1 G0/0/0 MAC	10
F0/11	PC1 MAC	10
F0/12	R1 G0/0/1 MAC	20
F0/24	PC2 MAC	20

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.

Note: This method of inter-VLAN routing is no longer implemented in switched networks and is included for explanation purposes only.

Router-on-a-Stick on Cisco devices

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.

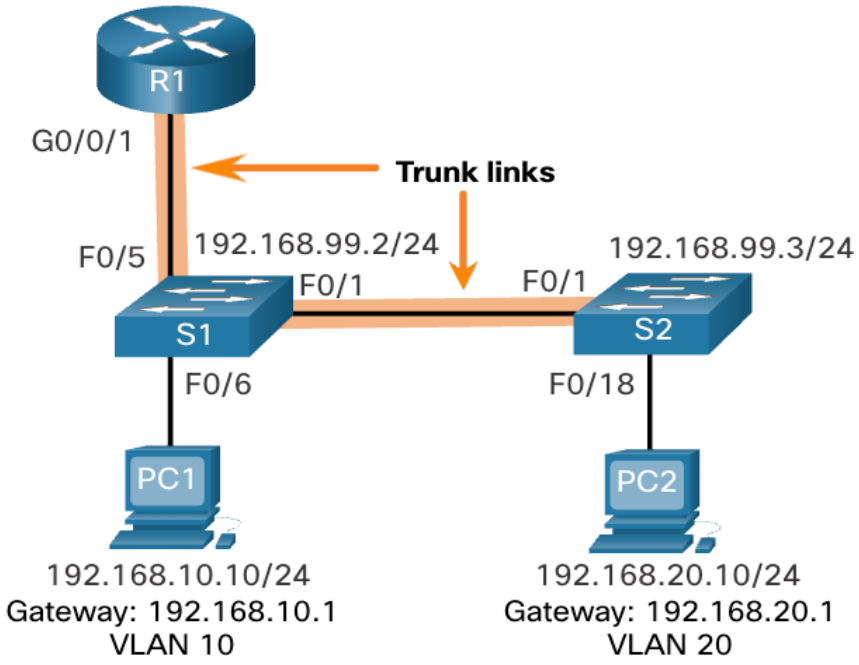
The configured subinterfaces are software-based virtual interfaces. Each is associated with a single physical Ethernet interface. Subinterfaces are configured in software on a router. Each subinterface is independently configured with an IP address and VLAN assignment. Subinterfaces are configured for 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.

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

This topic details how to configure router-on-a-stick inter-VLAN routing. You can see in the figure that the router is not in the center of the topology but instead, appears to be on a stick near the border, hence the name.

In the figure, the R1 GigabitEthernet 0/0/1 interface is connected to the S1 FastEthernet 0/5 port. The S1 FastEthernet 0/1 port is connected to the S2 FastEthernet 0/1 port. These are trunk links that are required to forward traffic within and between VLANs.



To route between VLANs, the R1 GigabitEthernet 0/0/1 interface is logically divided into three subinterfaces, as shown in the table. The table also shows the three VLANs that will be configured on the switches.

Router R1 Subinterfaces

Subinterface	VLAN	IP Address
G0/0/1.10	10	192.168.10.1/24
G0/0/1.20	20	192.168.20.1/24
G0/0/1.99	99	192.168.99.1/24

Assume that R1, S1, and S2 have initial basic configurations. Currently, PC1 and PC2 cannot **ping** each other because they are on separate networks. Only S1 and S2 can **ping** each other, but they but are unreachable by PC1 or PC2 because they are also on different networks.

To enable devices to ping each other, the switches must be configured with VLANs and trunking, and the router must be configured for inter-VLAN routing.

S1 switch port connected to R1 router and S2 switch should be configures as trunk. S1 switch port connected to PC1 should be configured as access port in VLAN 10
S2 Switch port connected to S1 switch should be configured and trunk
S2 switch port connected to PC2 should be configured as access port in VLAN 20
R1 Router G0/0/0 port should be configured with sub interfaces.
See example below:

S1(config)# **interface fa0/1**
S1(config-if)# **switchport mode trunk**

S1(config)# **interface fa0/5**
S1(config-if)# **switchport mode trunk**

S1(config)# **interface fa0/6**
S1(config-if)# **switchport mode access**
S1(config-if)# **switchport access vlan 10**

S2(config)# **interface fa0/1**
S2(config-if)# **switchport mode trunk**

S2(config)# **interface fa0/18**
S2(config-if)# **switchport mode access**
S2(config-if)# **switchport access vlan 20**

The router-on-a-stick method requires you to create a subinterface for each VLAN to be routed.

A subinterface is created using the **interface** *interface_id.subinterface_id* global configuration mode command. The subinterface syntax is the physical interface followed by a period and a

subinterface number. Although not required, it is customary to match the subinterface number with the VLAN number.

Each subinterface is then configured with the following two commands:

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

Repeat the process for each VLAN to be routed. Each router subinterface must be assigned an IP address on a unique subnet for routing to occur.

When all subinterfaces have been created, enable the physical interface using the **no shutdown** interface configuration command. If the physical interface is disabled, all subinterfaces are disabled.

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

```
R1(config)# interface G0/0/1.10
R1(config-subif)# description Default Gateway for VLAN 10
R1(config-subif)# encapsulation dot1Q 10
R1(config-subif)# ip add 192.168.10.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1.20
R1(config-subif)# description Default Gateway for VLAN 20
R1(config-subif)# encapsulation dot1Q 20
R1(config-subif)# ip add 192.168.20.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1
R1(config-if)# description Trunk link to S1
R1(config-if)# no shut
```

Let’s make the same configuration for Juniper devices. The topology is the same.

First create Vlans:

Switch S1

set vlans Vlan10 vlan-id 10
set vlans Vlan20 vlan-id 20

configure access port:

set interfaces f-0/6 unit 0 family ethernet-switching port-mode access
set interfaces f-0/6 unit 0 family ethernet-switching vlan members Vlan10

configure trunk port:

set interfaces f-0/5 unit 0 family ethernet-switching port-mode trunk
set interfaces f-0/5 unit 0 family ethernet-switching vlan all

set interfaces f-0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces f-0/1 unit 0 family ethernet-switching vlan members all

instead of “all” keyword, here you can indicate which vlans to be permitted on trunk interface. For example:
set interfaces f-0/0/1 unit 0 family ethernet-switching vlan members Vlan10, Vlan20. In this case only vlan10 and vlan20 will be permitted on this trunk interface.

Switch S2

set vlans Vlan10 vlan-id 10
set vlans Vlan20 vlan-id 20

configure access port:

set interfaces f-0/18 unit 0 family ethernet-switching port-mode access
set interfaces f-0/18 unit 0 family ethernet-switching vlan members Vlan10

configure trunk port:

set interfaces f-0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces f-0/1 unit 0 family ethernet-switching vlan all

Router R1

First enable VLAN tagging on physical interface:

set interfaces g-0/0/1 vlan-tagging

create subinterface for vlan10:

set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/0 unit 10 family inet address 192.168.10.1/24

create subinterface for vlan20:

set interfaces ge-0/0/0 unit 20 vlan-id 20
set interfaces ge-0/0/0 unit 20 family inet address 192.168.20.1/24

Layer 3 Switch Inter-VLAN Routing

Modern, enterprise networks rarely use router-on-a-stick because it does not scale easily to meet requirements. In these very large networks, network administrators use Layer 3 switches to configure inter-VLAN routing.

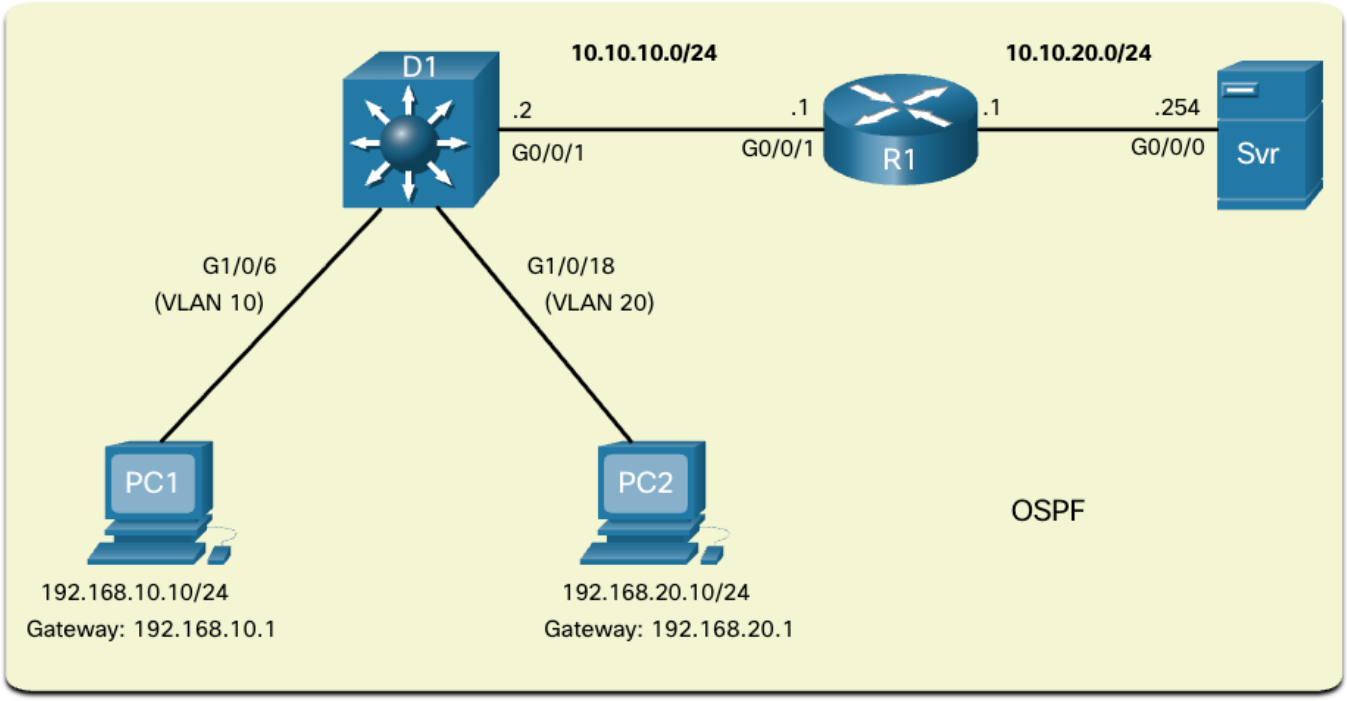
Inter-VLAN routing using the router-on-a-stick method is simple to implement for a small to medium-sized organization. However, a large enterprise requires a faster, much more scalable method to provide inter-VLAN routing.

Enterprise campus LANs use Layer 3 switches to provide inter-VLAN routing. Layer 3 switches use hardware-based switching to achieve higher-packet processing rates than routers. Layer 3 switches are also commonly implemented in enterprise distribution layer wiring closets.

Capabilities of a Layer 3 switch include the ability to do the following:

- Route from one VLAN to another using multiple switched virtual interfaces (SVIs).
- Convert a Layer 2 switchport to a Layer 3 interface (i.e., a routed port). A routed port is similar to a physical interface on a router.

Layer 3 Switch Scenario



D1 VLAN IP Addresses

VLAN Interface	IP Address
10	192.168.10.1/24
20	192.168.20.1/24

Configuration for Cisco devices

1. Create the VLANs.

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

D1(config-vlan)# **exit**

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)#

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.

5. 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)#

Configuration for Juniper devices

1. Create the VLANs.

set vlans Vlan10 vlan-id 10
set vlans Vlan20 vlan-id 20

2. Create the IRB interfaces.
In Junos switch virtual interfaces are called irb (integrated routing and bridging) interfaces.

set interfaces irb unit 10 description “Default Gateway for 192.168.10.0/24”
set interfaces irb unit 10 family inet address 192.168.10.1/24

set interfaces irb unit 20 description “Default Gateway for 192.168.20.0/24”
set interfaces irb unit 20 family inet address 192.168.20.1/24

3. Configure access ports.

set interfaces GigabitEthernet1/0/6 description “Access port to PC1”
set interfaces GigabitEthernet1/0/6 unit 0 family ethernet-switching port-mode access
set interfaces GigabitEthernet1/0/6 unit 0 family ethernet-switching vlan members Vlan10

set interfaces GigabitEthernet1/0/18 description “Access port to PC2”
set interfaces GigabitEthernet1/0/18 unit 0 family ethernet-switching port-mode access
set interfaces GigabitEthernet1/0/18 unit 0 family ethernet-switching vlan members Vlan20

4. Enable IP routing.
Juniper devices don't require special command to enable routing on L3 switches.

5. Configure the routed port.

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

set interfaces G0/0/1 description "routed Port Link to R1"
set interfaces G0/0/1 unit 0 family inet address 10.10.10.2/24

Chapter 4
Spanning Tree Protocol (STP)

This topic covers the causes of loops in a Layer 2 network and briefly explains how spanning tree protocol works. 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.

Spanning Tree Protocol (STP) is a loop-prevention network protocol that allows for redundancy while creating a loop-free Layer 2 topology. IEEE 802.1D is the original IEEE MAC Bridging standard for STP.

Path redundancy provides multiple network services by eliminating the possibility of a single point of failure. When multiple paths exist between two devices on an Ethernet network, and there is no spanning tree implementation on the switches, a Layer 2 loop occurs. A Layer 2 loop can result in MAC address table instability, link saturation, and high CPU utilization on switches and end-devices, resulting in the network becoming unusable.

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. Ethernet and Ethernet switches have no comparable mechanism for limiting the number of times a switch retransmits a Layer 2 frame. STP was developed specifically as a loop prevention mechanism for Layer 2 Ethernet.

Layer 2 Loops

Without STP enabled, Layer 2 loops can form, causing broadcast, multicast and unknown unicast frames to loop endlessly. This can bring down a network within a very short amount of time, sometimes in just a few seconds. For example, broadcast frames, such as an ARP Request are forwarded out all of the switch ports, except the original ingress port. This ensures that all devices in a broadcast domain are able to receive the frame. If there is more than one path for the frame to be forwarded out of, an endless loop can result. When a loop occurs, the MAC address table on a switch will constantly change with the updates from the broadcast frames, which results in MAC database instability. This can cause high CPU utilization, which makes the switch unable to forward frames.

Broadcast frames are not the only type of frames that are affected by loops. Unknown unicast frames sent onto a looped network can result in duplicate frames arriving at the destination device. An unknown unicast frame is when the switch does not have the destination MAC address in its MAC address table and must forward the frame out all ports, except the ingress port.

Broadcast Storm

A broadcast storm is an abnormally high number of broadcasts overwhelming the network during a specific amount of time. Broadcast storms can disable a network within seconds by overwhelming switches and end devices. Broadcast storms can be caused by a hardware problem such as a faulty NIC or from a Layer 2 loop in the network.

Layer 2 broadcasts in a network, such as ARP Requests are very common. A Layer 2 loop is likely to have immediate and disabling consequences on the network. Layer 2 multicasts are typically forwarded the same way as a broadcast by the switch. So, although IPv6 packets are never forwarded as a Layer 2 broadcast, ICMPv6 Neighbor Discovery uses Layer 2 multicasts.

The Spanning Tree Algorithm

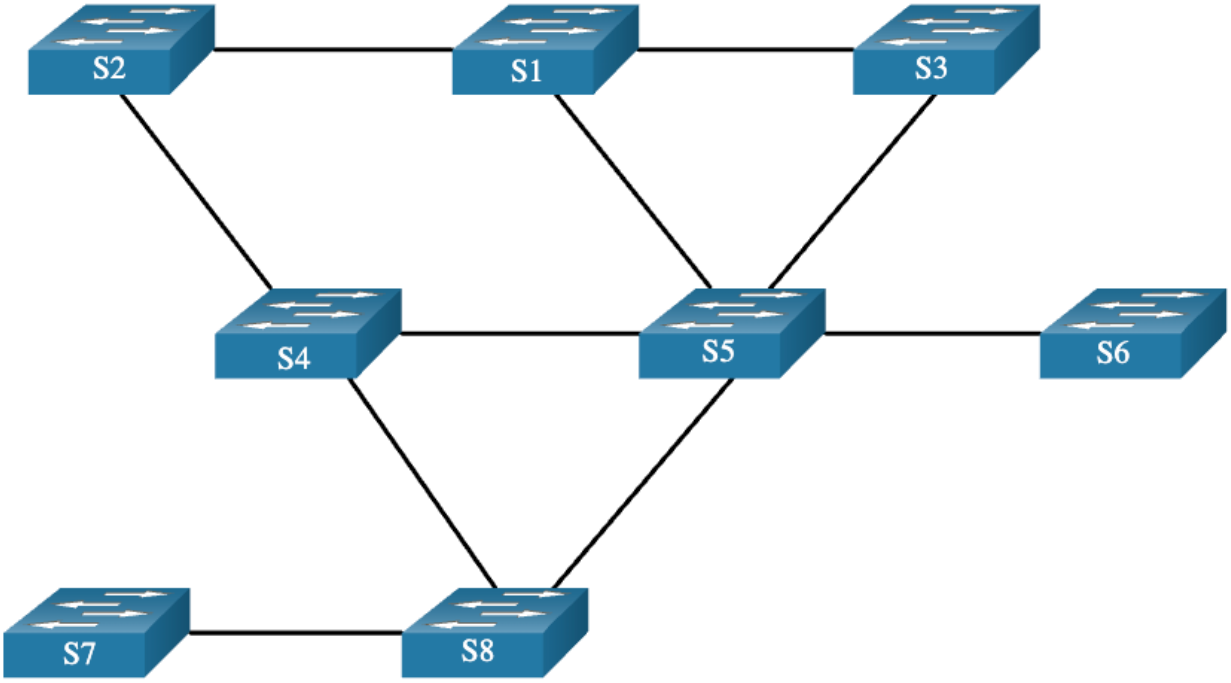
STP is based on an algorithm invented by Radia Perlman while working for Digital Equipment Corporation, and published in the 1985 paper "An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN.” Her spanning tree algorithm (STA) creates a loop-free topology by selecting a single root bridge where all other switches determine a single least-cost path.

Without the loop prevention protocol, loops would occur rendering a redundant switch network inoperable.

STP prevents loops from occurring by configuring a loop-free path through the network using strategically placed "blocking-state" ports. The switches running STP are able to compensate for failures by dynamically unblocking the previously blocked ports and permitting traffic to traverse the alternate paths.

STA Scenario Topology

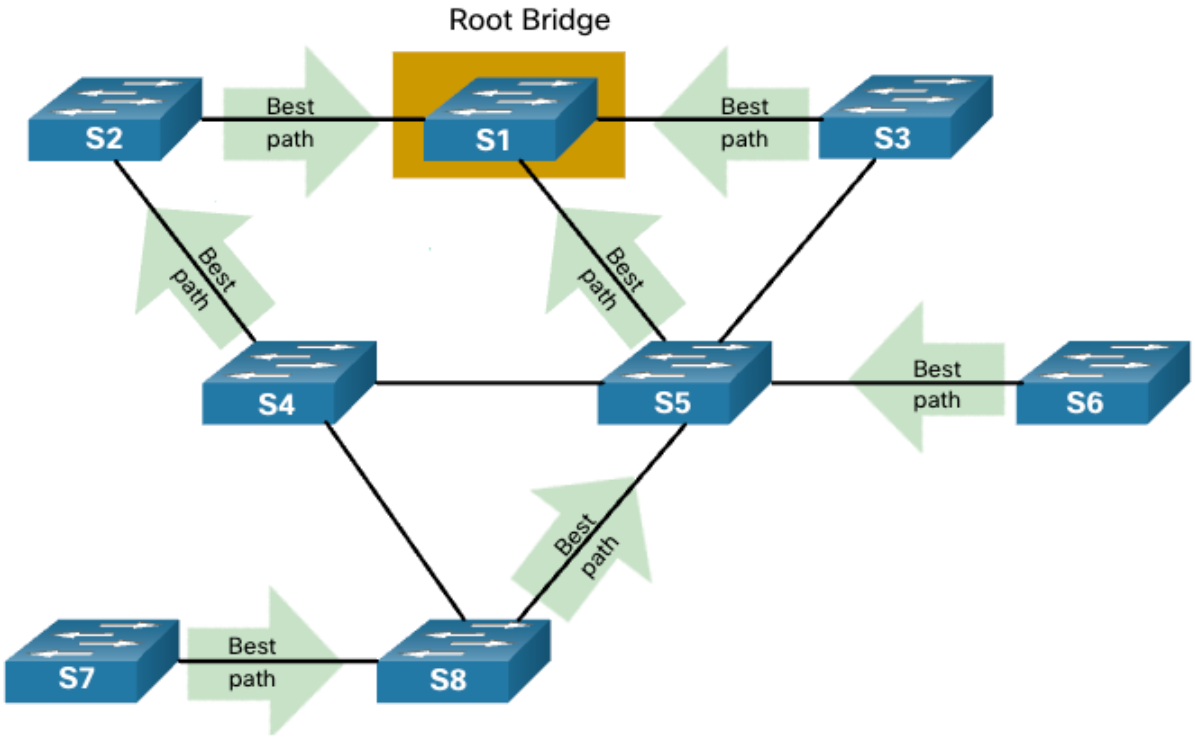
This STA scenario uses an Ethernet LAN with redundant connections between multiple switches.



Select the Root Bridge

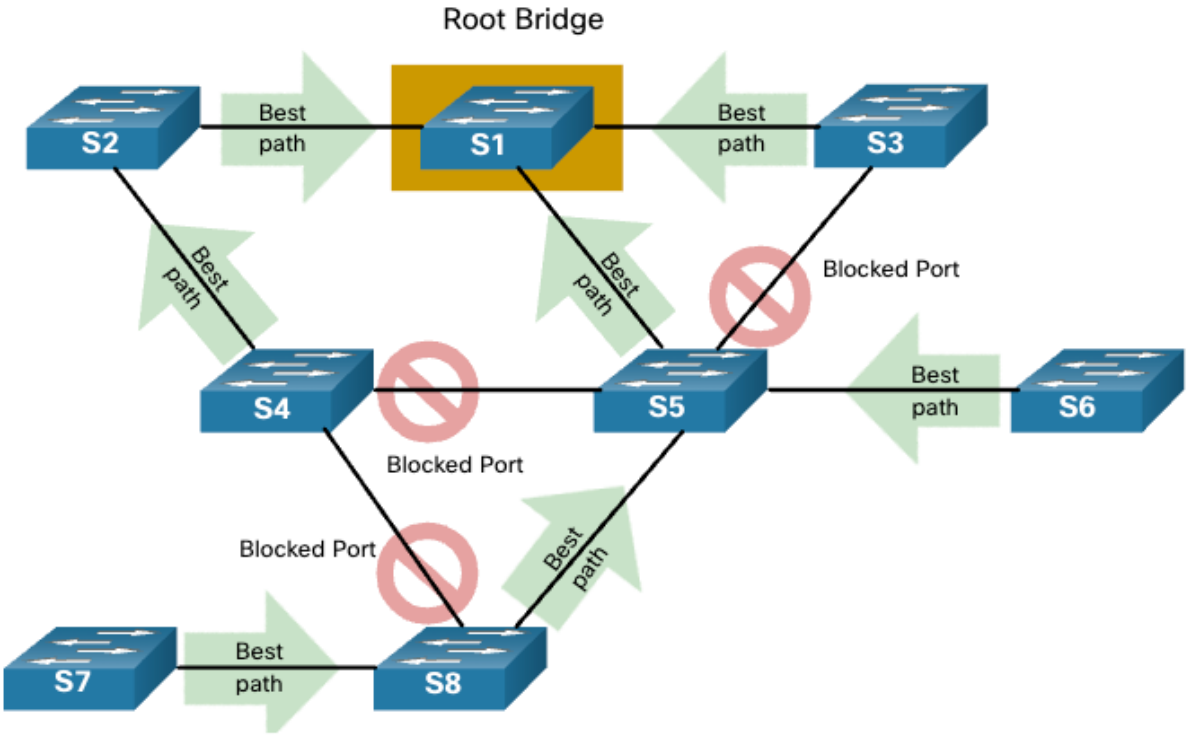
The spanning tree algorithm begins by selecting a single root bridge. The figure shows that switch S1 has been selected as the root bridge. In this topology, all links are equal cost (same bandwidth). Each switch will determine a single, least cost path from itself to the root bridge.

Note: The STA and STP refers to switches as bridges. This is because in the early days of Ethernet, switches were referred to as bridges.



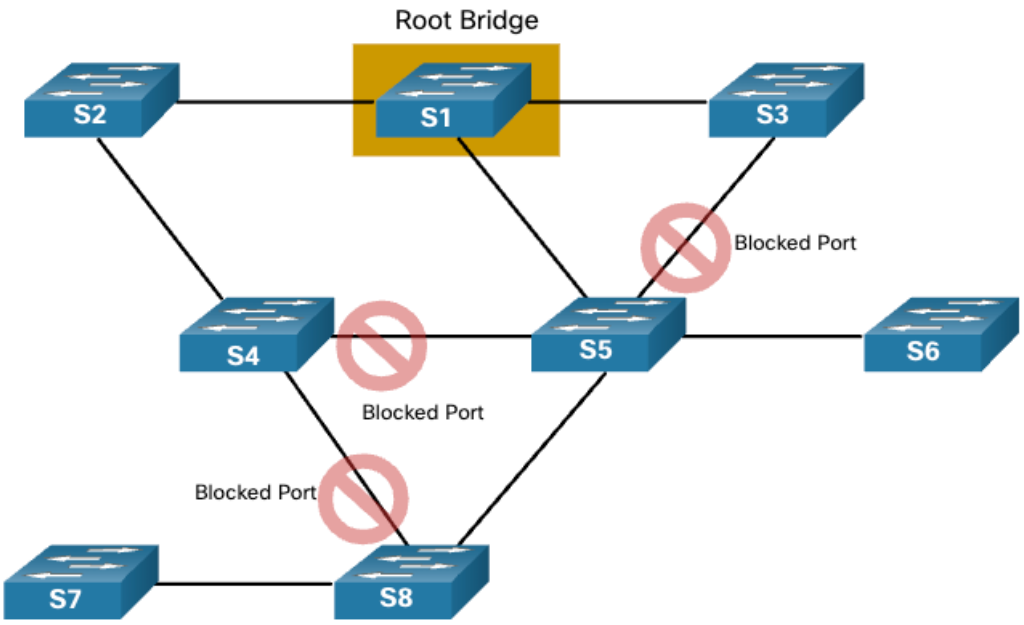
Block Redundant Paths

STP ensures that there is only one logical path between all destinations on the network by intentionally blocking redundant paths that could cause a loop, as shown in the figure. When a port is blocked, user data is prevented from entering or leaving that port. Blocking the redundant paths is critical to preventing loops on the network.



Loop-Free Topology

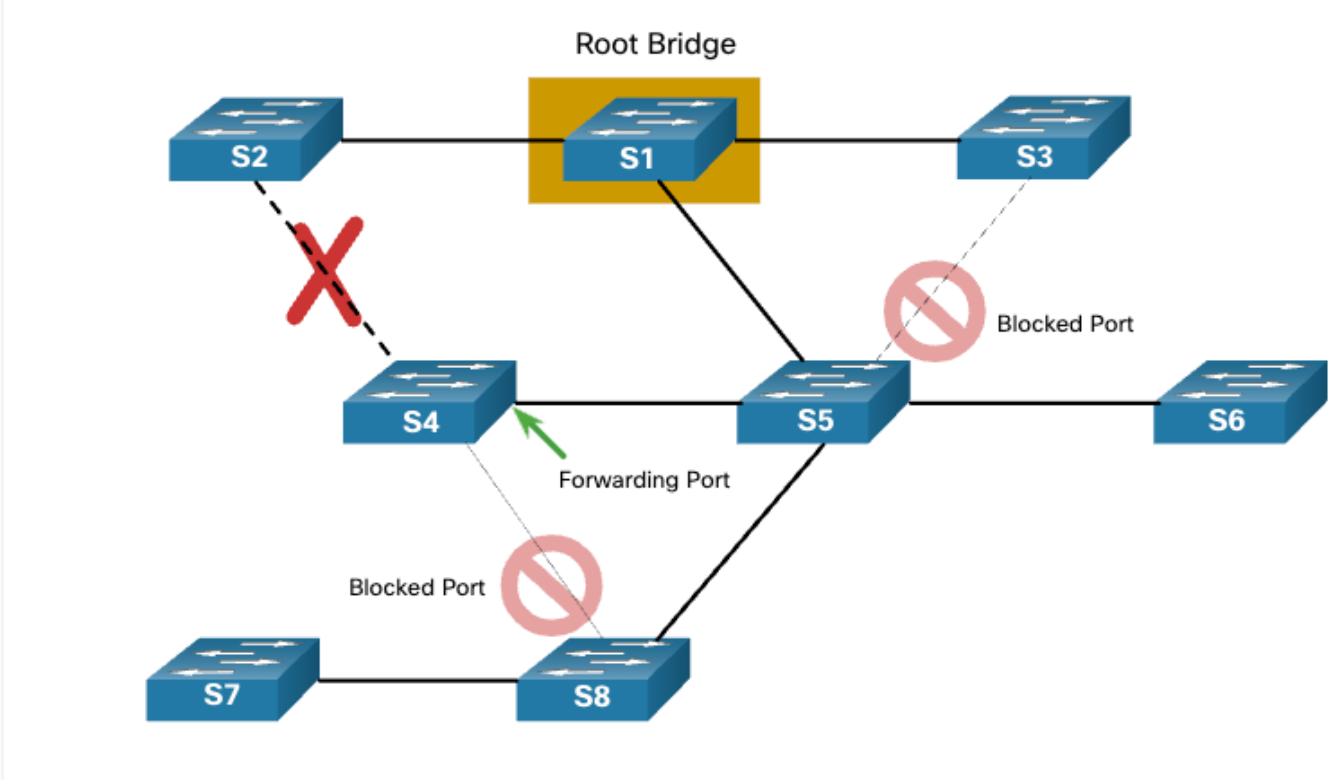
A blocked port has the effect of making that link a non-forwarding link between the two switches, as shown in the figure. Notice that this creates a topology where each switch has only a single path to the root bridge, similar to branches on a tree that connect to the root of the tree.



Link Failure Causes Recalculation

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. STP recalculations can also occur any time a new switch or new inter-switch link is added to the network.

The figure shows a link failure between switches S2 and S4 causing STP to recalculate. Notice that the previously redundant link between S4 and S5 is now forwarding to compensate for this failure. There is still only one path between every switch and the root bridge.

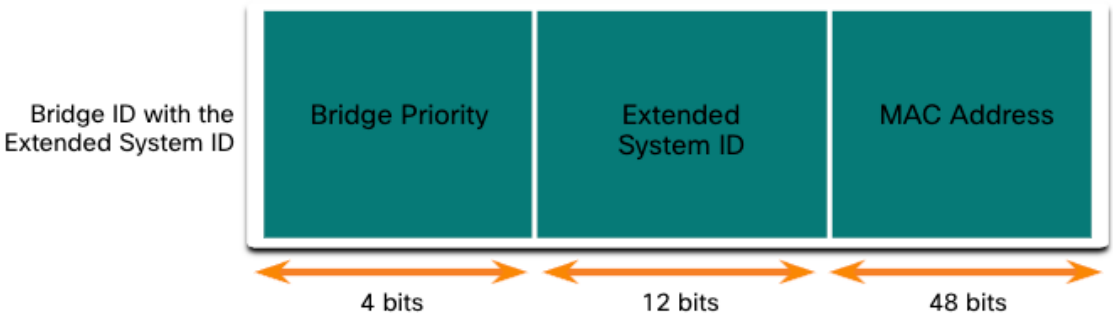


Steps to a Loop-Free Topology

Now you know how loops are created and the basics of using spanning tree protocol to prevent them. This topic will take you, step by step, through the operation of STP. Using the STA, STP builds a loop-free topology in a four-step process:

1. Elect the root bridge.
2. Elect the root ports.
3. Elect designated ports.
4. Elect alternate (blocked) ports.

During STA and STP functions, switches use Bridge Protocol Data Units (BPDUs) to share information about themselves and their connections. BPDUs are used to elect the root bridge, root ports, designated ports, and alternate ports. Each BPDU contains a bridge ID (BID) that identifies which switch sent the BPDU. The BID is involved in making many of the STA decisions including root bridge and port roles. As shown in the figure, the BID contains a priority value, an extended system ID, and the MAC address of the switch. The lowest BID value is determined by the combination of these three fields.



The BID includes the Bridge Priority, the Extended System ID, and the MAC Address of the switch.

Bridge Priority

The default priority value for all Cisco switches is the decimal value 32768. The range is 0 to 61440 in increments of 4096. A lower bridge priority is preferable. A bridge priority of 0 takes precedence over all other bridge priorities.

Extended System ID

The extended system ID value is a decimal value added to the bridge priority value in the BID to identify the VLAN for this BPDU.

Early implementations of IEEE 802.1D were designed for networks that did not use VLANs. There was a single common spanning tree across all switches. For this reason, in older switches, the extended system ID was not included in the BPDUs. As VLANs became common for network infrastructure segmentation, 802.1D was enhanced to include support for VLANs, which required that the 12-bit VLAN ID be included in the BPDU frame. VLAN information is included in the BPDU frame through the use of the extended system ID.

The extended system ID allows later implementations of STP to have different root bridges for different sets of VLANs. This can allow for redundant, non-forwarding links in a STP topology for one set of VLANs to be used by a different set of VLANs using a different root bridge.

MAC address

When two switches are configured with the same priority and have the same extended system ID, the switch having the MAC address with the lowest value, expressed in hexadecimal, will have the lower BID.

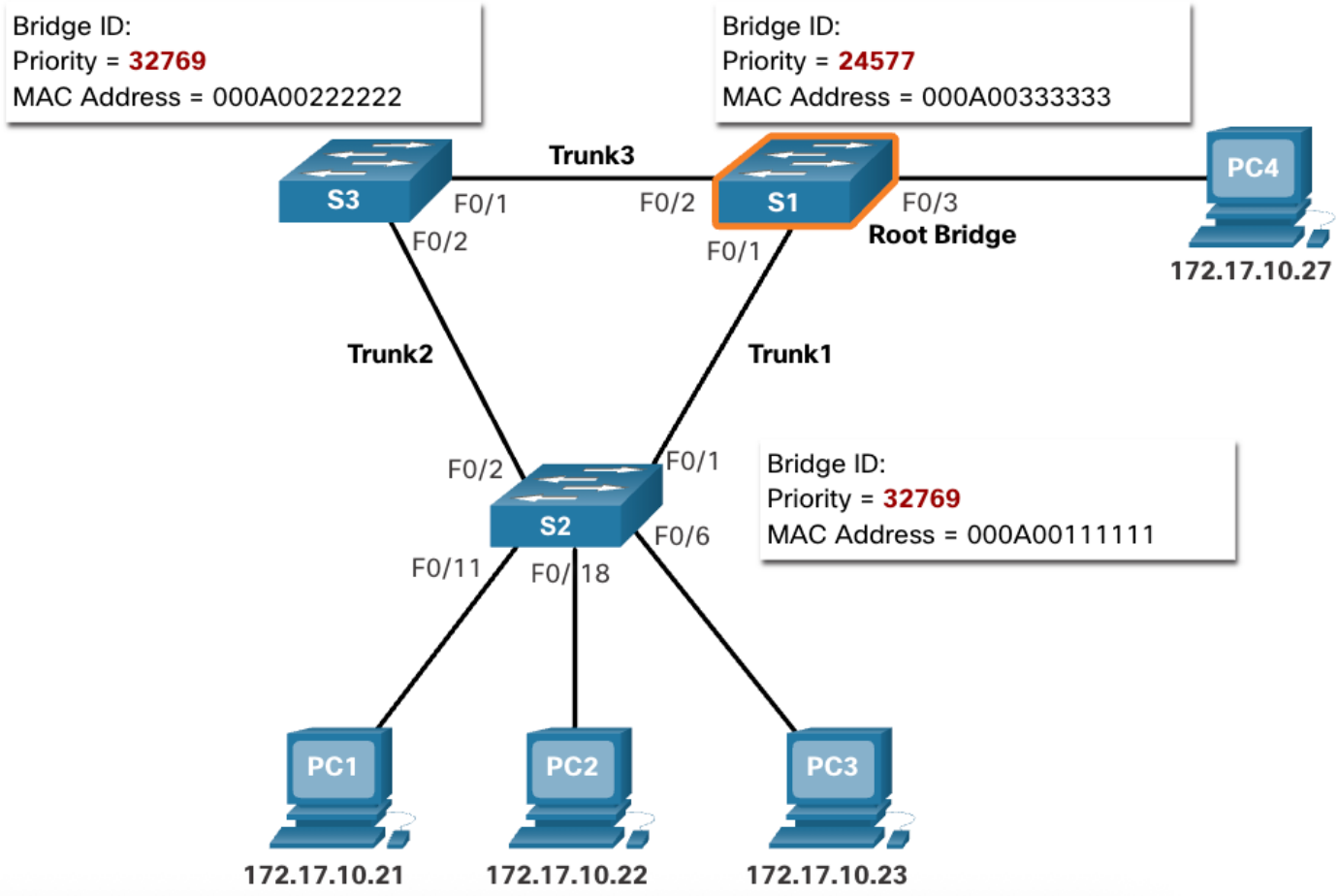
Elect the Root Bridge

The STA designates a single switch as the root bridge and uses it as the reference point for all path calculations. Switches exchange BPDUs to build the loop-free topology beginning with selecting the root bridge.

An election process determines which switch becomes the root bridge. All switches in the broadcast domain participate in the election process. After a switch boots, it begins to send out BPDU frames every two seconds. These BPDU frames contain the BID of the sending switch and the BID of the root bridge, known as the Root ID.

The switch with the lowest BID will become the root bridge. At first, all switches declare themselves as the root bridge with their own BID set as the Root ID. Eventually, the switches learn through the exchange of BPDUs which switch has the lowest BID and will agree on one root bridge.

In the figure, S1 is elected the root bridge because it has the lowest BID.

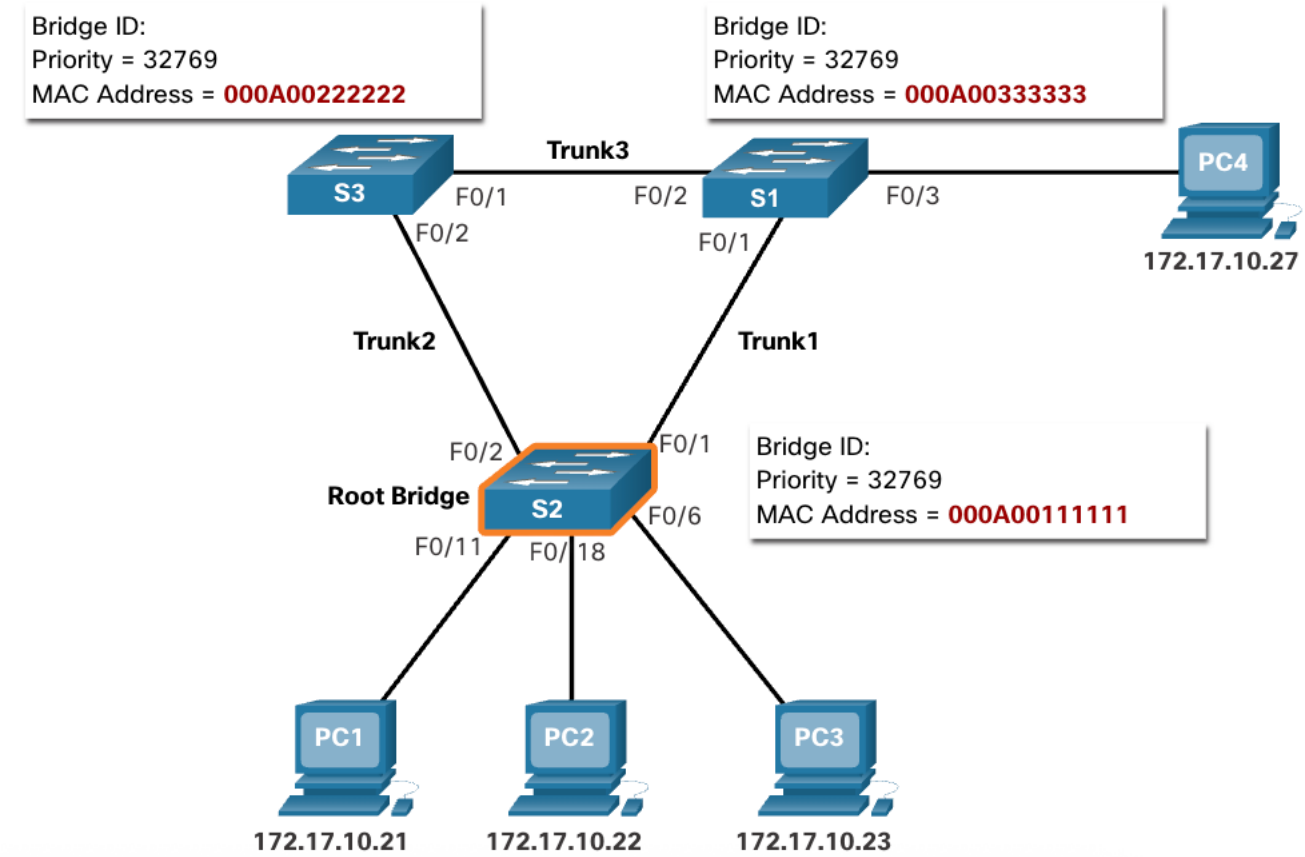


Impact of Default BIDs

Because the default priority is 32768, it is possible for two or more switches to have the same priority. In this scenario, where the priorities are the same, the switch with the lowest MAC address will become the root bridge. To ensure that the root bridge decision best meets network requirements, it is recommended that the administrator configure the desired root bridge switch with a lower priority.

In the figure, all switches are configured with the same priority of 32769. Here the MAC address becomes the deciding factor as to which switch becomes the root bridge. The switch with the lowest hexadecimal MAC address value is the preferred root bridge. In this example, S2 has the lowest value for its MAC address and is elected as the root bridge for that spanning tree instance.

Note: In the example, the priority of all the switches is 32769. The value is based on the 32768 default bridge priority and the extended system ID (VLAN 1 assignment) associated with each switch (32768+1).



Determine the Root Path Cost

When the root bridge has been elected for a given 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, known as the internal root path cost, is determined by the sum of all the individual port costs along the path from the switch to the root bridge.

Note: The BPDU includes the root path cost. This is the cost of the path from the sending switch to the root bridge.

When a switch receives the BPDU, it adds the ingress port cost of the segment to determine its internal root path cost.

The default port costs are defined by the speed at which the port operates. The table shows the default port costs suggested by IEEE. Cisco switches by default use the values as defined by the IEEE 802.1D standard, also known as the short path cost, for both STP and RSTP. However, the IEEE standard suggests using the values defined in the IEEE-802.1w, also known as long path cost, when using 10 Gbps links and faster.

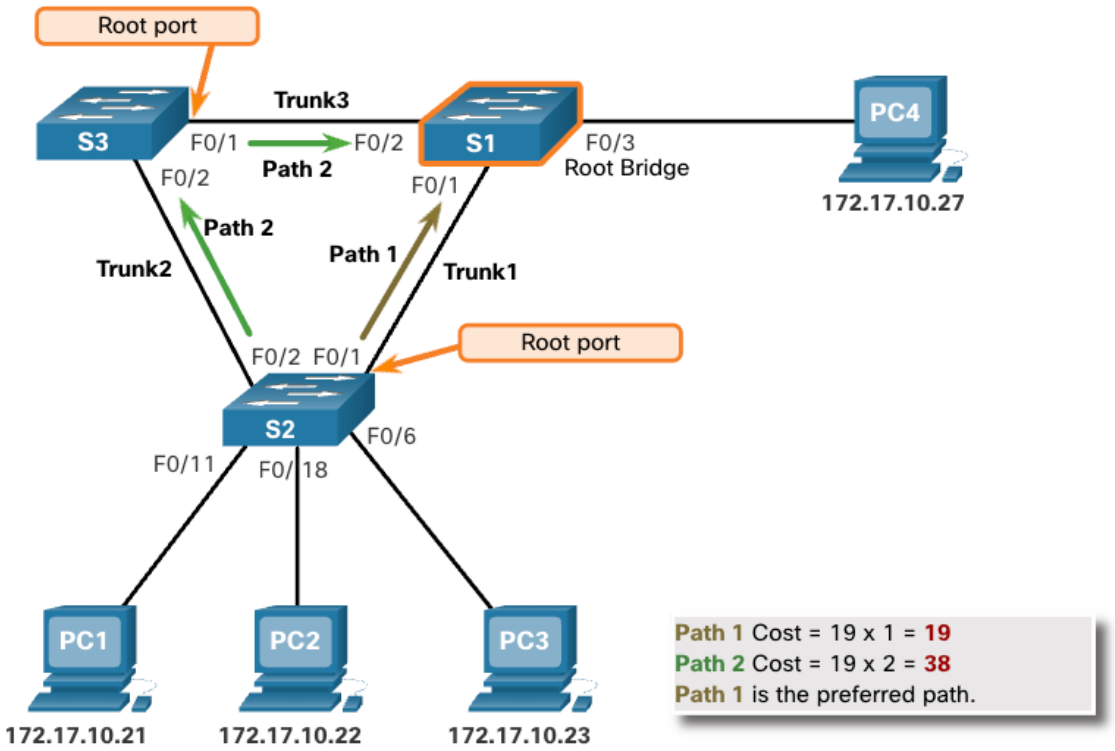
Link Speed	STP Cost: IEEE 802.1D-1998	RSTP Cost: IEEE 802.1w-2004
10 Gbps	2	2,000
1 Gbps	4	20,000
100 Mbps	19	200,000
10 Mbps	100	2,000,000

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

Elect the Root Ports

After the root bridge has been determined, the STA algorithm is used to select the root port. Every non-root switch will select one root port. The root port is the port closest to the root bridge in terms of overall cost (best path) to the root bridge. This overall cost is known as the internal root path cost.

The internal root path cost is equal to the sum of all the port costs along the path to the root bridge, as shown in the figure. Paths with the lowest cost become preferred, and all other redundant paths are blocked. In the example, the internal root path cost from S2 to the root bridge S1 over path 1 is 19 (based on the IEEE-specified individual port cost) while the internal root 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 and F0/1 becomes the root port on S2.



Elect Designated Ports

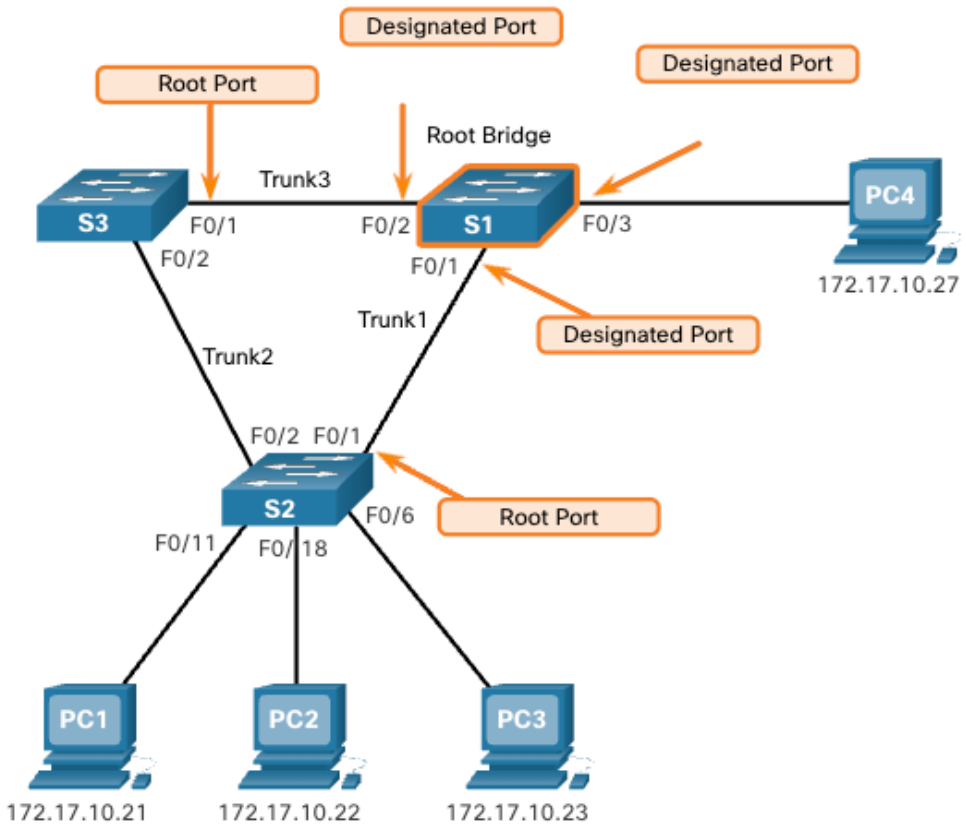
The loop prevention part of spanning tree becomes evident during these next two steps. After each switch selects a root port, the switches will then select designated ports.

Every segment between two switches will have one designated port. The designated port on the segment (with two switches) that has the LOWEST internal root path cost to the root bridge. In other words, the designated port has the best path to receive traffic leading to the root bridge.

What is not a root port or a designated port becomes an alternate or blocked port. The end result is a single path from every switch to the root bridge.

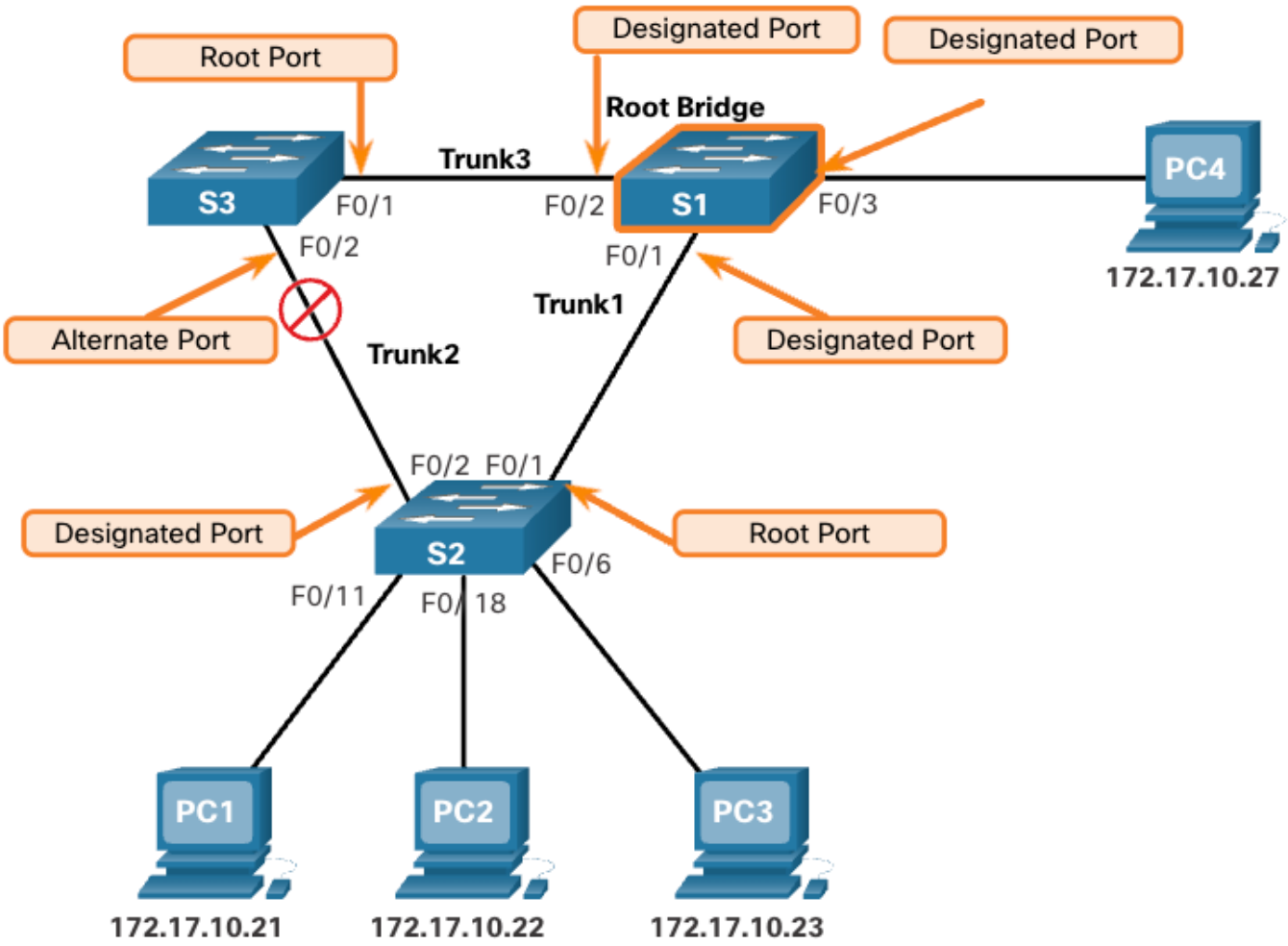
Designated Ports on Root Bridge

All ports on the root bridge are designated ports, as shown in the figure. This is because the root bridge has the lowest cost to itself.



All the ports on the root bridge are designated ports.

Elect Alternate (Blocked) Ports



The Fa0/2 interface of S3 is not a root port or a designated port, so it becomes an alternate or blocked port.

Elect a Root Port from Multiple Equal-Cost Paths

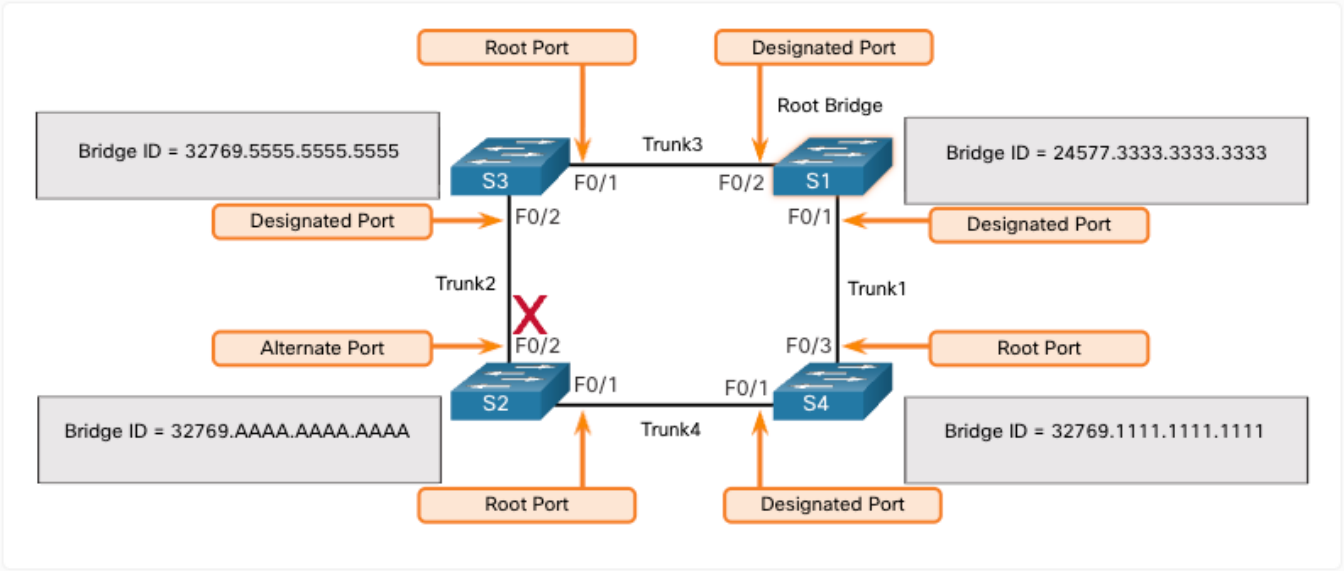
Root port and designated ports are based on the lowest path cost to the root bridge. But what happens if the switch has multiple equal-cost paths to the root bridge? How does a switch designate a root port?

When a switch has multiple equal-cost paths to the root bridge, the switch will determine a port using the following criteria:

1. Lowest sender BID
2. Lowest sender port priority
3. Lowest sender port ID

1. Lowest Sender BID

The figure shows a topology with four switches, including switch S1 as the root bridge. Examining the port roles, port F0/1 on switch S3 and port F0/3 on switch S4 have been selected as root ports because they have the lowest cost path (root path cost) to the root bridge for their respective switches. S2 has two ports, F0/1 and F0/2 with equal cost paths to the root bridge. In this case the bridge IDs of the neighboring switches, S3 and S4, will be used to break the tie. This is known as the sender's BID. S3 has a BID of 32769.5555.5555.5555 and S4 has a BID of 32769.1111.1111.1111. Because S4 has a lower BID, the F0/1 port of S2, which is the port connected to S4, will be the root port.



STP Timers and Port States

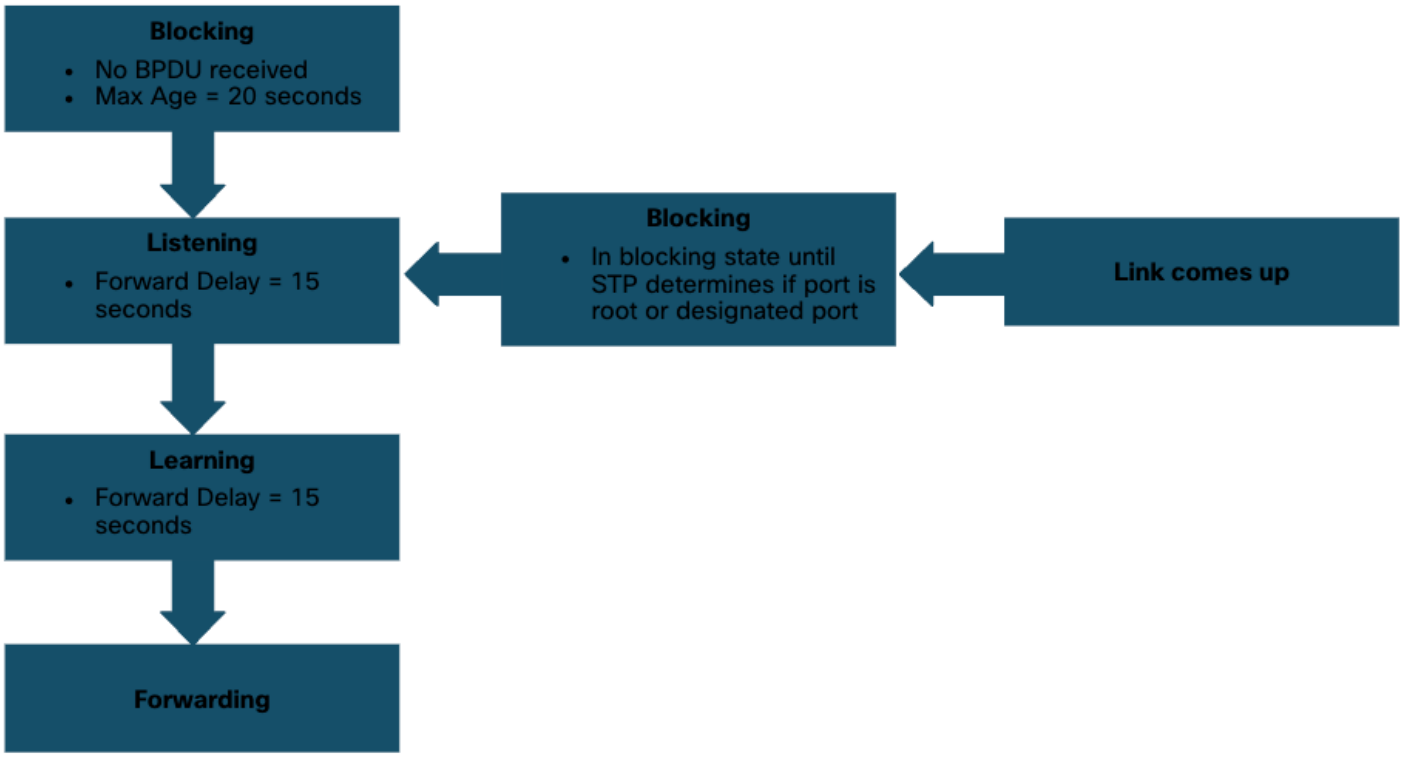
STP convergence requires three timers, as follows:

- **Hello Timer** -The hello time is the interval between BPDUs. The default is 2 seconds but can be modified to between 1 and 10 seconds.
- **Forward Delay Timer** -The forward delay is the time that is spent in the listening and learning state. The default is 15 seconds but can be modified to between 4 and 30 seconds.
- **Max Age Timer** -The max age is the maximum length of time that a switch waits before attempting to change the STP topology. The default is 20 seconds but be modified to between 6 and 40 seconds.

Note: The default times can be changed on the root bridge, which dictates the value of these timers for the STP domain.

STP facilitates 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. If a switch port transitions directly from the blocking state to the forwarding state without information about the full topology during the transition, the port can temporarily create a data loop. For this reason, STP has five ports states, four of which are operational port states as shown in the figure. The disabled state is considered non-operational.

Note: To avoid problems with STP, IEEE recommends a maximum diameter of seven switches when using the default STP timers.



The details of each port state are shown in the table.

Port State	Description
Blocking	The port is an alternate port and does not participate in frame forwarding. The port receives BPDU frames to determine the location and root ID of the root bridge. BPDU frames also determine which port roles each switch port should assume in the final active STP topology. With a Max Age timer of 20 seconds, a switch port that has not received an expected BPDU from a neighbor switch will go into the blocking state.
Listening	After the blocking state, a port will move to the listening state. The port receives BPDUs to determine the path to the root. The switch port also transmits its own BPDU frames and informs adjacent switches that the switch port is preparing to participate in the active topology.
Learning	A switch port transitions to the learning state after the listening state. During the learning state, the switch port receives and processes BPDUs and prepares to participate in frame forwarding. It also begins to populate the MAC address table. However, in the learning state, user frames are not forwarded to the destination.
Forwarding	In the forwarding state, a switch port is considered part of the active topology. The switch port forwards user traffic and sends and receives BPDU frames.
Disabled	A switch port in the disabled state does not participate in spanning tree and does not forward frames. The disabled state is set when the switch port is administratively disabled.

Operational Details of Each Port State

Port State	BPDU	MAC Address Table	Forwarding Data Frames
Blocking	Receive only	No update	No
Listening	Receive and send	No update	No
Learning	Receive and send	Updating table	No
Forwarding	Receive and send	Updating table	Yes
Disabled	None sent or received	No update	No

Per-VLAN Spanning Tree

Up until now, we have discussed STP in an environment where there is only one VLAN. However, STP can be configured to operate in an environment with multiple VLANs.

In Per-VLAN Spanning Tree (PVST for Cisco and VSTP for Juniper) versions of STP, there is a root bridge elected for each spanning tree instance. This makes it possible to have different root bridges for different sets of VLANs. STP operates a separate instance of STP for each individual VLAN. If all ports on all switches are members of VLAN 1, then there is only one spanning tree instance.

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.

The latest standard for spanning tree is contained in IEEE-802-1D-2004, the IEEE standard for Local and metropolitan area networks:Media Access Control (MAC) Bridges. This version of the standard states that switches and bridges that comply with the standard will use Rapid Spanning Tree Protocol (RSTP) instead of the older STP protocol specified in the original 802.1d standard. In this curriculum, when the original Spanning Tree Protocol is the context of a discussion, the phrase “original 802.1D spanning tree” is used to avoid confusion. Because the two protocols share much of the same terminology and methods for the loop-free path, the primary focus will be on the current standard and the Cisco and Juniper proprietary implementations of STP and RSTP.

RSTP Concepts

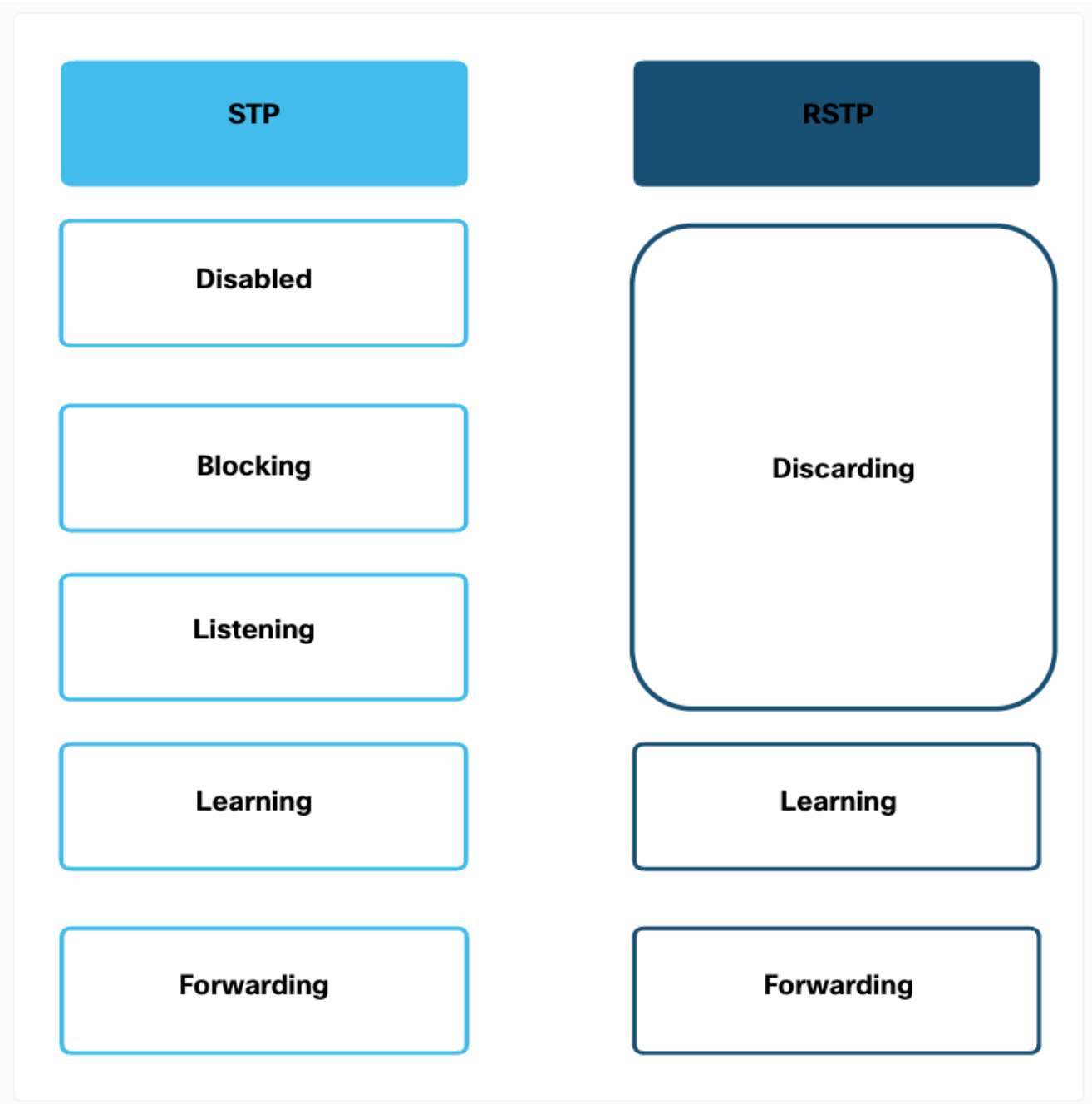
RSTP (IEEE 802.1w) supersedes the original 802.1D while retaining backward compatibility. The 802.1w STP terminology remains primarily the same as the original IEEE 802.1D STP terminology. Most parameters have been left unchanged. Users that are familiar with the original STP standard can easily configure RSTP. The same spanning tree algorithm is used for both STP and RSTP to determine port roles and topology.

RSTP increases the speed of the recalculation of the spanning tree when the Layer 2 network topology changes. RSTP can achieve much faster convergence in a properly configured network, sometimes in as little as a few hundred milliseconds. If a port is configured to be an alternate port it can immediately change to a forwarding state without waiting for the network to converge.

Note: Rapid PVST+ is the Cisco and VSTP is Juniper implementation of RSTP on a per-VLAN basis . With Rapid PVST+ and VSTP an independent instance of RSTP runs for each VLAN.

STP and RSTP Port States

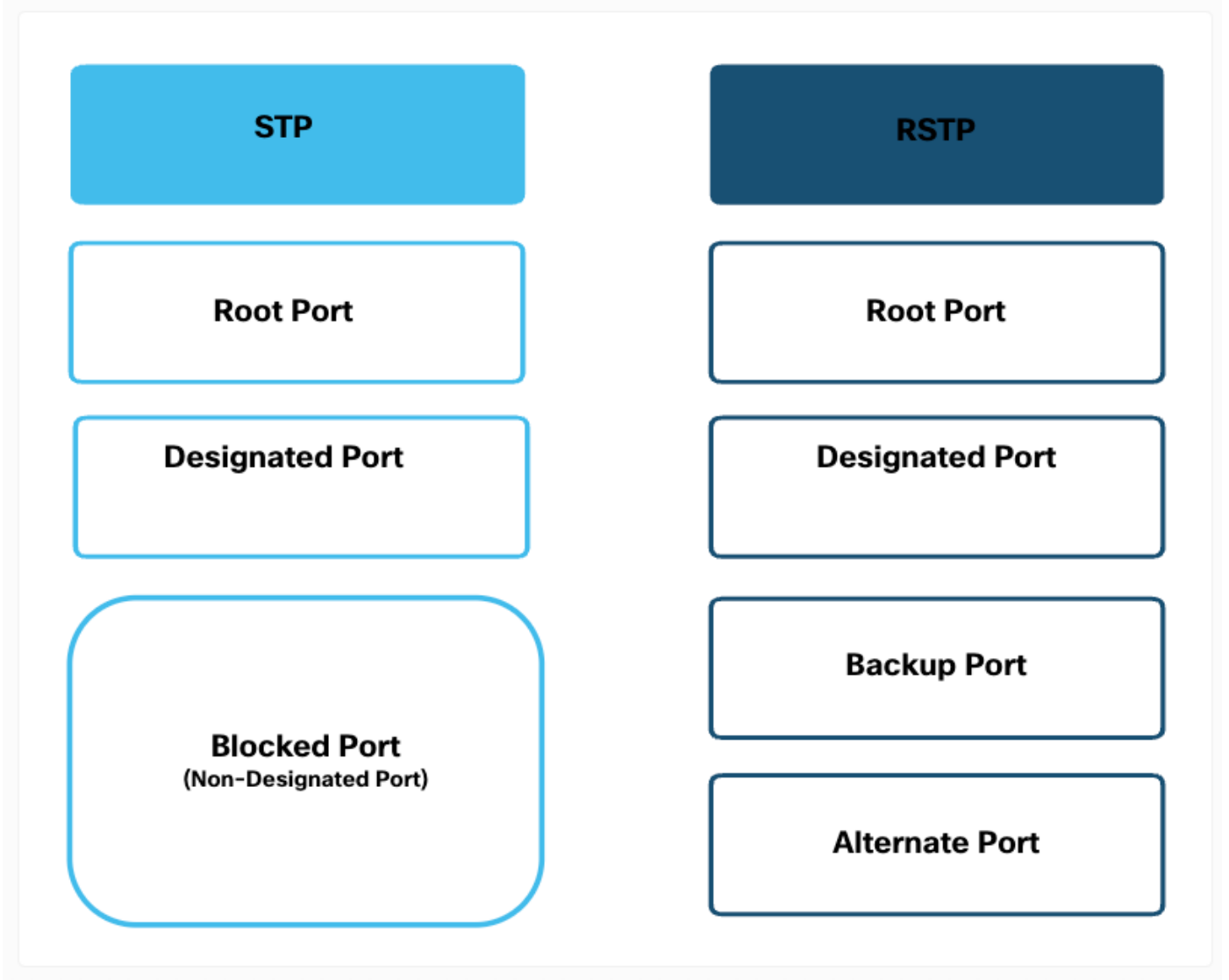
As shown in the figure, there are only three port states in RSTP that correspond to the three possible operational states in STP. The 802.1D disabled, blocking, and listening states are merged into a unique 802.1w discarding state.



STP and RSTP Port Roles

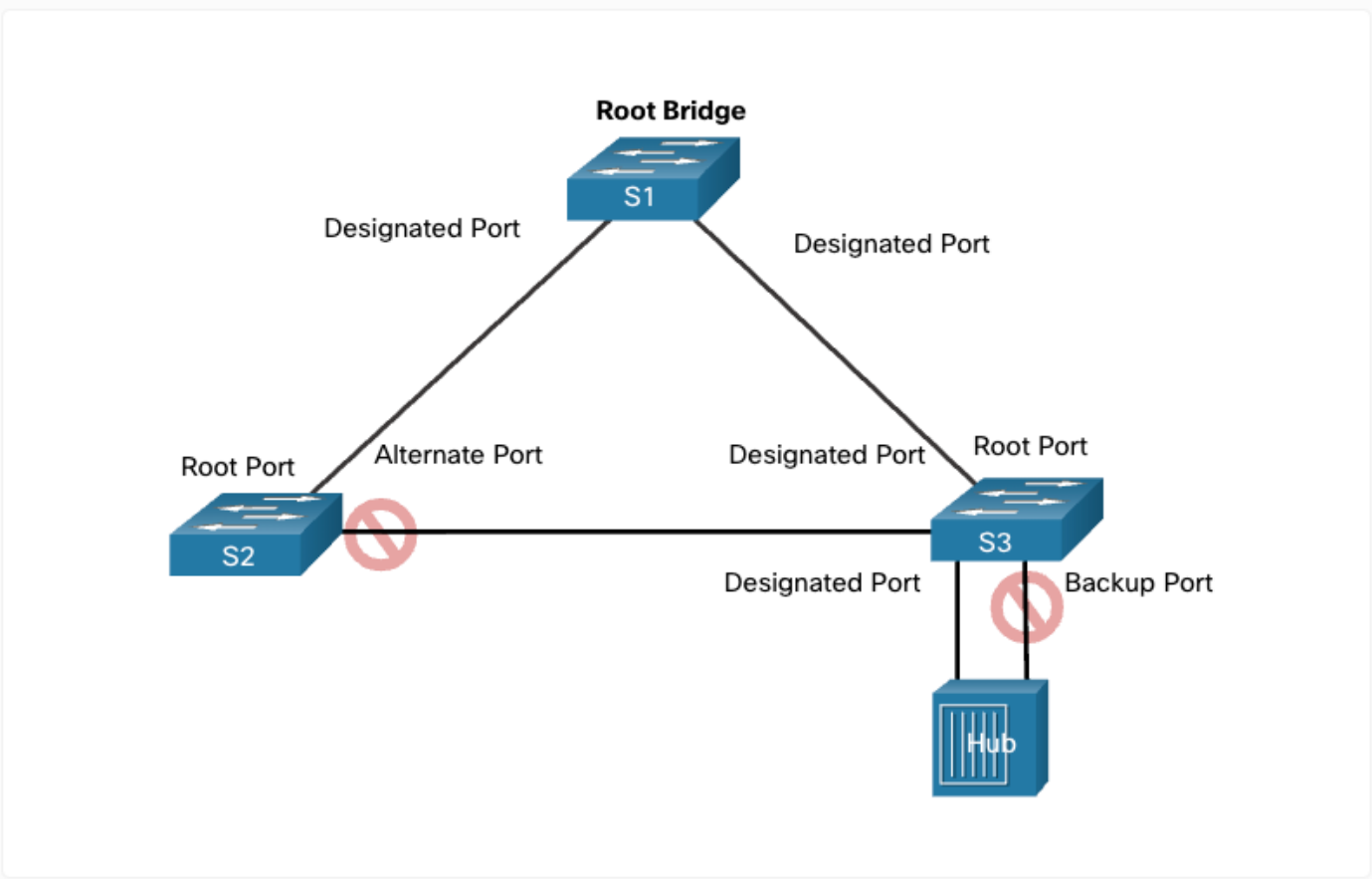
As shown in the figure, root ports and designated ports are the same for both STP and RSTP. However, there are two RSTP port roles that correspond to the blocking state of STP. In STP, a

blocked port is defined as not being the designated or root port. RSTP has two port roles for this purpose.



RSTP Alternate and Backup Ports

As shown in the figure, the alternate port has an alternate path to the root bridge. The backup port is a backup to a shared medium, such as a hub. A backup port is less common because hubs are now considered legacy devices.



PortFast, Edge port, BPDU Guard and BPDU-block

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 (for Cisco) and Edge port (for Juniper), 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. You can use PortFast or Edge port on access ports to allow devices connected to these ports, such as DHCP clients, to access the network immediately, rather than waiting for IEEE 802.1D STP to converge on each VLAN. Because the purpose of PortFast/Edge 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/Edge on a port connecting to another switch, you risk creating a spanning tree loop. PortFast/Edge is only for use on switch ports that connect to end devices.

In a valid PortFast/Edge 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. The BPDU guard feature provides a secure response to invalid configurations because an administrator must manually put the interface back into service.

Cisco Configuration

Configure rapid PVST+, and bridge priority for vlan 10

Switch(config)# spanning-tree mode rapid-pvst
Switch(config)# spanning-tree vlan 10 priority 4096

Configure portfast and BPDU guard on an interface

Switch(config-if)# panning-tree portfast
Switch (config-if)#spanning-tree bpduguard enable

Juniper configuration

Configure VSTP

set protocols vstp interface all
set protocols vstp vlan all bridge-priority 4k

configure edge port and BPDU block

set protocols vstp interface ge-0/0/1 edge
set protocols vstp bpdu-block-on-edge interface ge-0/0/1

Chapter 5
Etherchannel/ Aggregated Ethernet Interfaces

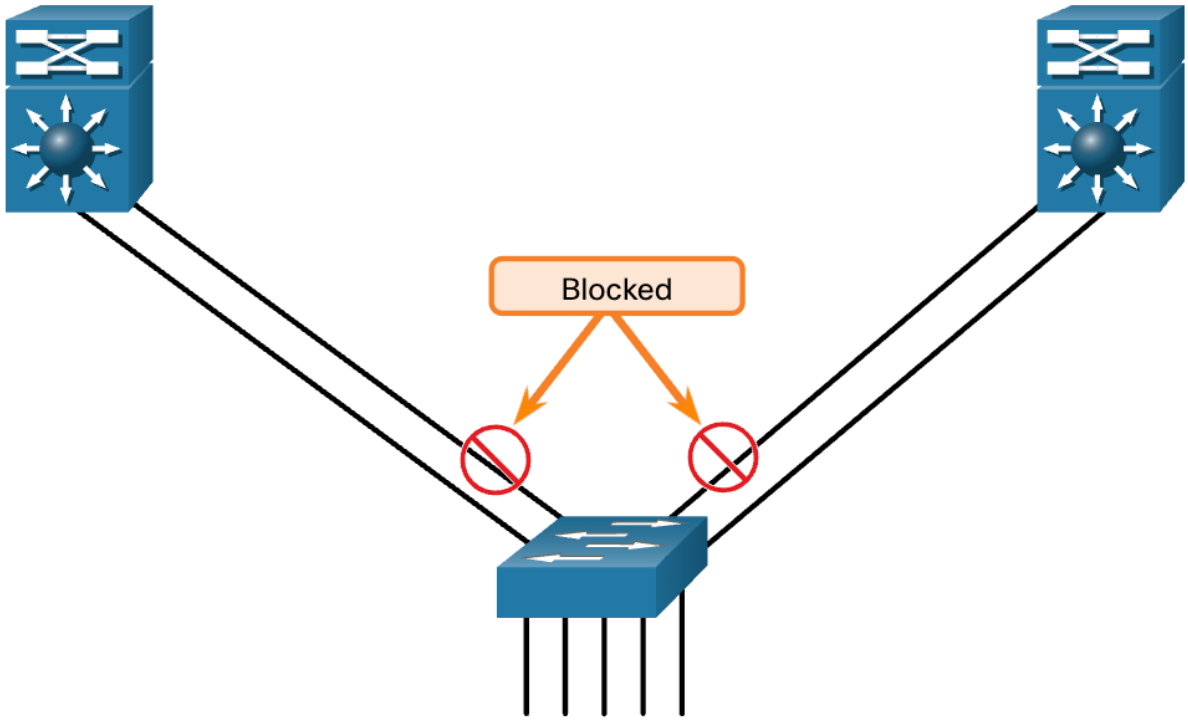
Link Aggregation

There are scenarios in which more bandwidth or redundancy between devices is needed than what can be provided by a single link. Multiple links could be connected between devices to increase bandwidth. However, Spanning Tree Protocol (STP), will block redundant links to prevent switching loops, as shown in the figure.

A link aggregation technology is needed that allows redundant links between devices that will not be blocked by STP. That technology is known as EtherChannel for Cisco and Aggregated Ethernet Interfaces for Juniper.

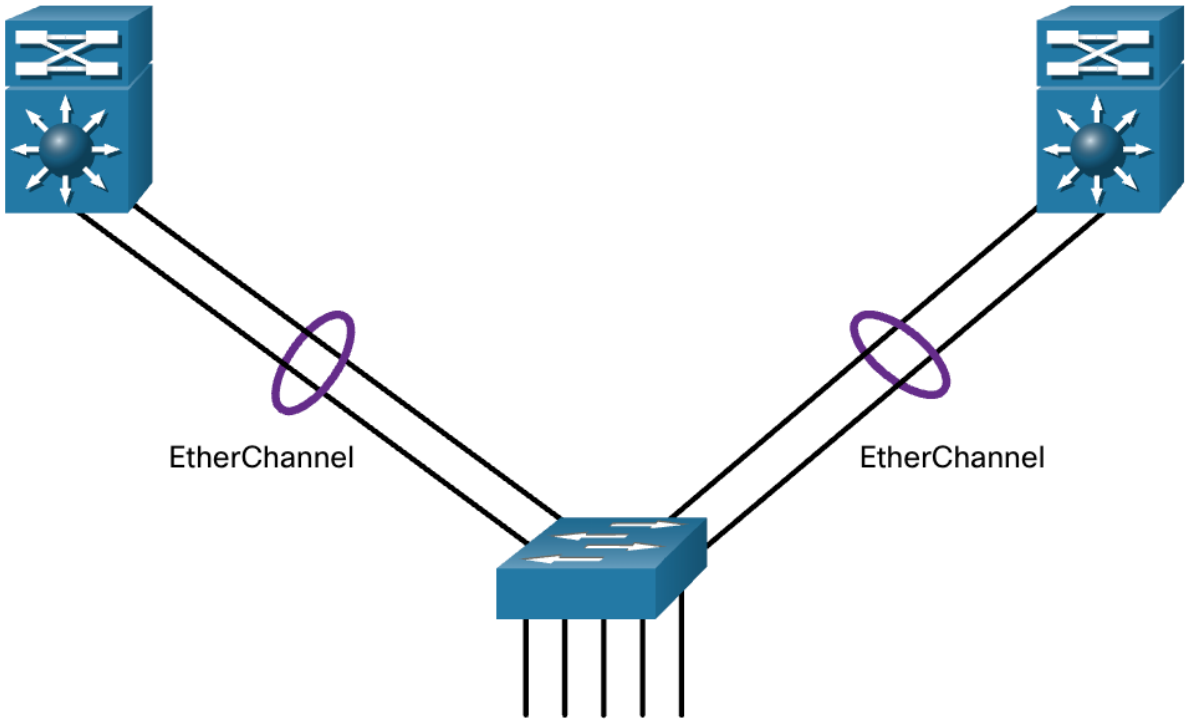
EtherChannel is a link aggregation technology that groups multiple physical Ethernet links together into one single logical link. It is used to provide fault-tolerance, load sharing, increased bandwidth, and redundancy between switches, routers, and servers.

EtherChannel technology makes it possible to combine the number of physical links between the switches to increase the overall speed of switch-to-switch communication.



EtherChannel

EtherChannel technology was originally developed by Cisco as a LAN switch-to-switch technique of grouping several Fast Ethernet or Gigabit Ethernet ports into one logical channel. When an EtherChannel is configured, the resulting virtual interface is called a port channel. The physical interfaces are bundled together into a port channel interface, as shown in the figure.



Advantages of EtherChannel

EtherChannel technology has many advantages, including the following:

- Most configuration tasks can be done on the EtherChannel interface instead of on each individual port, ensuring configuration consistency throughout the links.
- EtherChannel relies on existing switch ports. There is no need to upgrade the link to a faster and more expensive connection to have more bandwidth.
- Load balancing takes place between links that are part of the same EtherChannel. Depending on the hardware platform, one or more load-balancing methods can be implemented. These methods include source MAC and destination MAC load balancing, or source IP and destination IP load balancing, across the physical links.
- EtherChannel creates an aggregation that is seen as one logical link. When several EtherChannel bundles exist between two switches, STP may block one of the bundles to prevent switching loops. When STP blocks one of the redundant links, it blocks the entire EtherChannel. This blocks all the ports belonging to that EtherChannel link. Where there is only one EtherChannel link, all physical links in the EtherChannel are active because STP sees only one (logical) link.
- EtherChannel provides redundancy because the overall link is seen as one logical connection. Additionally, the loss of one physical link within the channel does not create a change in the topology. Therefore, a spanning tree recalculation is not required. Assuming at least one physical link is present; the EtherChannel remains functional, even if its overall throughput decreases because of a lost link within the EtherChannel.

Implementation Restrictions

EtherChannel implementation has certain implementation restrictions, including the following:

- Interface types cannot be mixed. For example, Fast Ethernet and Gigabit Ethernet cannot be mixed within a single EtherChannel.
- Currently for Cisco equipment each EtherChannel can consist of up to eight compatibly-configured Ethernet ports. EtherChannel provides full-duplex bandwidth up to 800 Mbps (Fast EtherChannel) or 8 Gbps (Gigabit EtherChannel) between one switch and another switch or host. For Juniper Most EX and QFX switches support up to 128 physical interfaces per LAG (Link aggregeation)
- The individual EtherChannel group member port configuration must be consistent on both devices. If the physical ports of one side are configured as trunks, the physical ports of the other side must also be configured as trunks within the same native VLAN. Additionally, all ports in each EtherChannel link must be configured as Layer 2 ports.
- Each EtherChannel has a logical port channel interface, as shown in the figure. A configuration applied to the port channel interface affects all physical interfaces that are assigned to that interface.

LACP Operation

LACP is part of an IEEE specification (802.3ad) that allows several physical ports to be bundled to form a single logical channel. LACP allows a switch to negotiate an automatic bundle by sending LACP packets to the other switch. Because LACP is an IEEE standard, it can be used to facilitate EtherChannels in multivendor environments including Cisco and Juniper.

LACP helps create the EtherChannel link by detecting the configuration of each side and making sure that they are compatible so that the EtherChannel link can be enabled when needed. The modes for LACP are as follows:

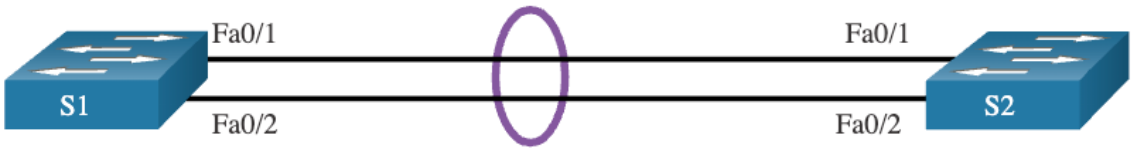
- **LACP active** - This LACP mode places a port in an active negotiating state. In this state, the port initiates negotiations with other ports by sending LACP packets.
- **LACP passive** - This LACP mode places a port in a passive negotiating state. In this state, the port responds to the LACP packets that it receives but does not initiate LACP packet negotiation.

Configuration Guidelines

- **EtherChannel support** - All Ethernet interfaces must support EtherChannel with no requirement that interfaces be physically contiguous.
- **Speed and duplex** - Configure all interfaces in an EtherChannel to operate at the same speed and in the same duplex mode.
- **VLAN match** - All interfaces in the EtherChannel bundle must be assigned to the same VLAN or be configured as a trunk (shown in the figure).
- **Range of VLANs** - An EtherChannel supports the same allowed range of VLANs on all the interfaces in a trunking EtherChannel. If the allowed range of VLANs is not the same, the interfaces do not form an EtherChannel.

LACP Configuration Example for Cisco

EtherChannel is disabled by default and must be configured. The topology in the figure will be used to demonstrate an EtherChannel configuration example using LACP.



Configuring EtherChannel with LACP requires the following three steps:

Step 1. Specify the interfaces that compose the EtherChannel group using the **interface range** *interface* global configuration mode command. The **range** keyword allows you to select several interfaces and configure them all together.

Step 2. Create the port channel interface with the **channel-group** *identifier* **mode active** command in interface range configuration mode. The identifier specifies a channel group number. The **mode active** keywords identify this as an LACP EtherChannel configuration.

Step 3. To change Layer 2 settings on the port channel interface, enter port channel interface configuration mode using the **interface port-channel** command, followed by the interface identifier. In the example, S1 is configured with an LACP EtherChannel. The port channel is configured as a trunk interface with the allowed VLANs specified.

```
S1(config)# interface range FastEthernet 0/1 - 2
S1(config-if-range)# channel-group 1 mode active
```

```
Creating a port-channel interface Port-channel 1
S1(config-if-range)# exit
S1(config)# interface port-channel 1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk allowed vlan 1,2,20
```

When several port channel interfaces are configured on the same device, use the **show etherchannel summary** command to display one line of information per port channel. In the output, the switch has one EtherChannel configured; group 1 uses LACP.

The interface **bundle consists of the FastEthernet0/1 and FastEthernet0/2 interfaces. The group is a Layer** 2 EtherChannel and it is in use, as indicated by the letters SU next to the port channel number.

```
S1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator
        M - not in use, minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
        A - formed by Auto LAG
Number of channel-groups in use: 1
Number of aggregators:           1
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         LACP      Fa0/1(P)    Fa0/2(P)
```

LACP Configuration Example for Juniper

Define Number of Aggregated Interfaces: Specify how many LAGs can be created

set chassis aggregated-devices ethernet device-count <number>

Configure Physical Interfaces: Assign interfaces to the ae interface and set them to act as 802.3ad links.

et interfaces <interface-name> ether-options 802.3ad ae<id>

Example: set interfaces ge-0/0/0 ether-options 802.3ad ae1
Example: set interfaces ge-0/0/1 ether-options 802.3ad ae1

Configure the ae **Interface with LACP:** Enable LACP in active mode.

set interfaces ae<id> aggregated-ether-options lacp active
set interfaces ae<id> aggregated-ether-options lacp periodic fast

Configure Logical Unit (Optional but typical): Set up the ae interface for Layer 2/3 traffic.

set interfaces ae<id> unit 0 family ethernet-switching interface-mode trunk

Chapter 6
DHCPv4

The Dynamic Host Configuration Protocol (DHCP) dynamically assigns IP addresses to devices. DHCPv4 is for an IPv4 network. This means that you, the network administrator, do not have to spend your day configuring IP addresses for every device on your network. In a small home or office, that would not be very difficult, but any large network might have hundreds, or even thousands of devices.

In this module, you will learn how to configure a Cisco IOS and Juniper router to be a DHCPv4 server. Then you will learn how to configure a router as a client. DHCPv4 configuration skills will significantly reduce your workload, and who doesn't want that?

DHCPv4 Server and Client

Dynamic Host Configuration Protocol v4 (DHCPv4) assigns IPv4 addresses and other network configuration information dynamically. Because desktop clients typically make up the bulk of network nodes, DHCPv4 is an extremely useful and timesaving tool for network administrators.

A dedicated DHCPv4 server is scalable and relatively easy to manage. However, in a small branch or SOHO location, a router can be configured to provide DHCPv4 services without the need for a dedicated server.

The DHCPv4 server dynamically assigns, or leases, an IPv4 address from a pool of addresses for a limited period of time chosen by the server, or until the client no longer needs the address.

Clients lease the information from the server for an administratively defined period. Administrators configure DHCPv4 servers to set the leases to time out at different intervals. The lease is typically anywhere from 24 hours to a week or more. When the lease expires, the client must ask for another address, although the client is typically reassigned the same address.

DHCPv4 Operation

DHCPv4 works in a client/server mode. When a client communicates with a DHCPv4 server, the server assigns or leases an IPv4 address to that client. The client connects to the network with that leased IPv4 address until the lease expires. The client must contact the DHCP server periodically to extend the lease. This lease mechanism ensures that clients that move or power off do not keep addresses that they no longer need. When a lease expires, the DHCP server returns the address to the pool where it can be reallocated as necessary.

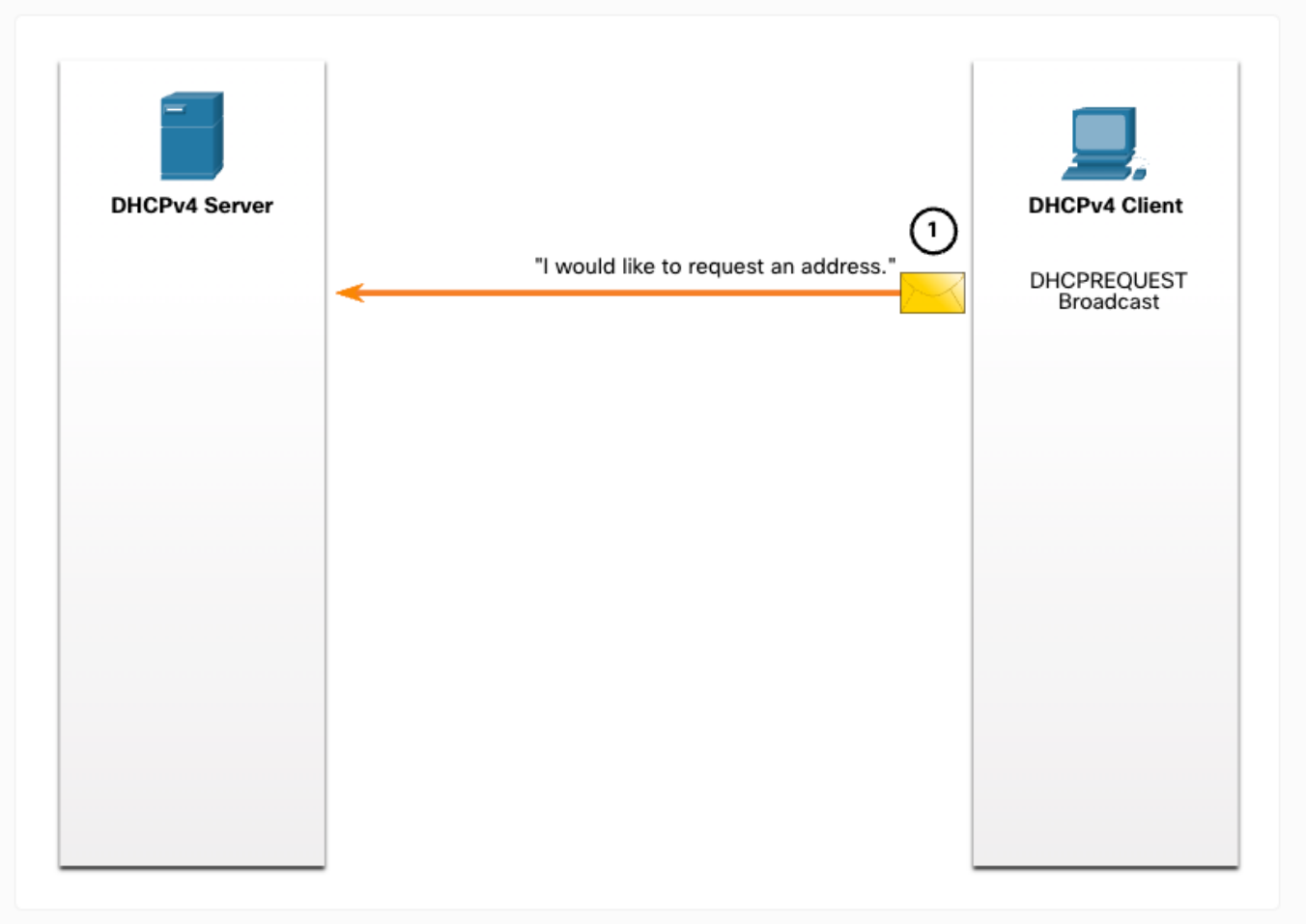
Steps to Obtain a Lease

When the client boots (or otherwise wants to join a network), it begins a four-step process to obtain a lease:

1. DHCP Discover (DHCPDISCOVER)
2. DHCP Offer (DHCPOFFER)
3. DHCP Request (DHCPREQUEST)
4. DHCP Acknowledgment (DHCPACK)

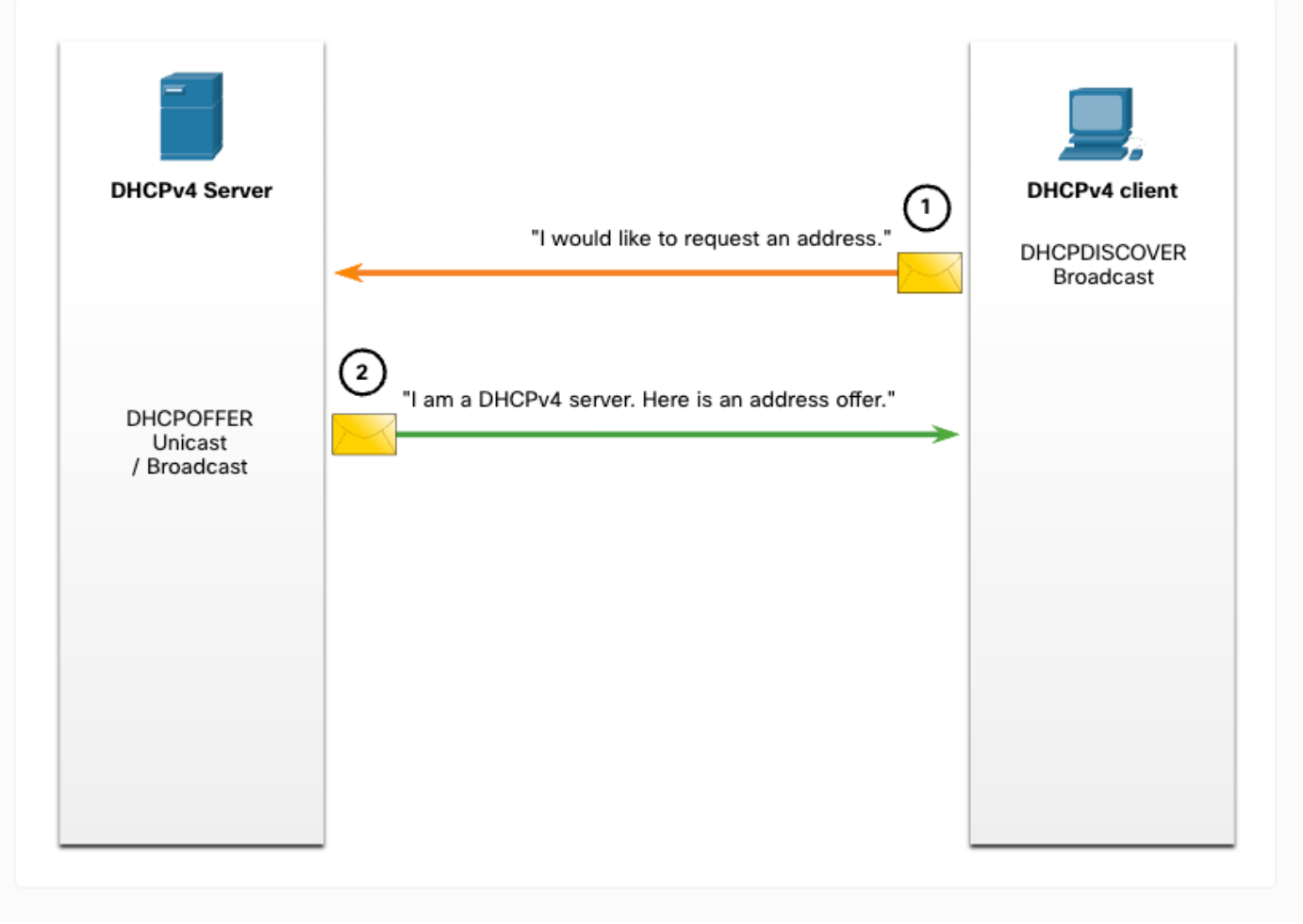
Step 1. DHCP Discover (DHCPDISCOVER)

The client starts the process using a broadcast DHCPDISCOVER message with its own MAC address to discover available DHCPv4 servers. Because the client has no valid IPv4 information at bootup, it uses Layer 2 and Layer 3 broadcast addresses to communicate with the server. The purpose of the DHCPDISCOVER message is to find DHCPv4 servers on the network.



Step 2. DHCP Offer (DHCPOFFER)

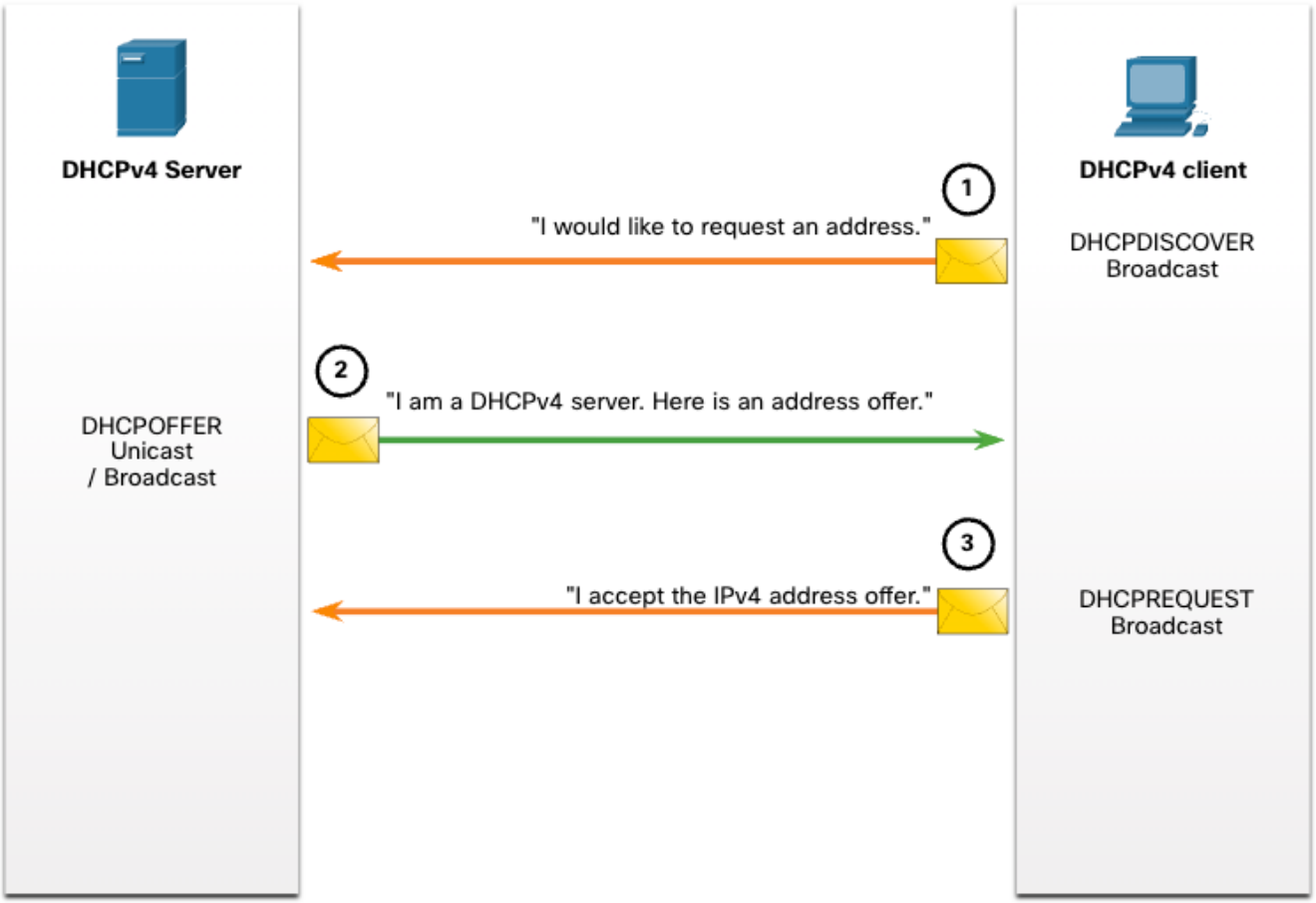
When the DHCPv4 server receives a DHCPDISCOVER message, it reserves an available IPv4 address to lease to the client. The server also creates an ARP entry consisting of the MAC address of the requesting client and the leased IPv4 address of the client. The DHCPv4 server sends the binding DHCPOFFER message to the requesting client.



Step 3. DHCP Request (DHCPREQUEST)

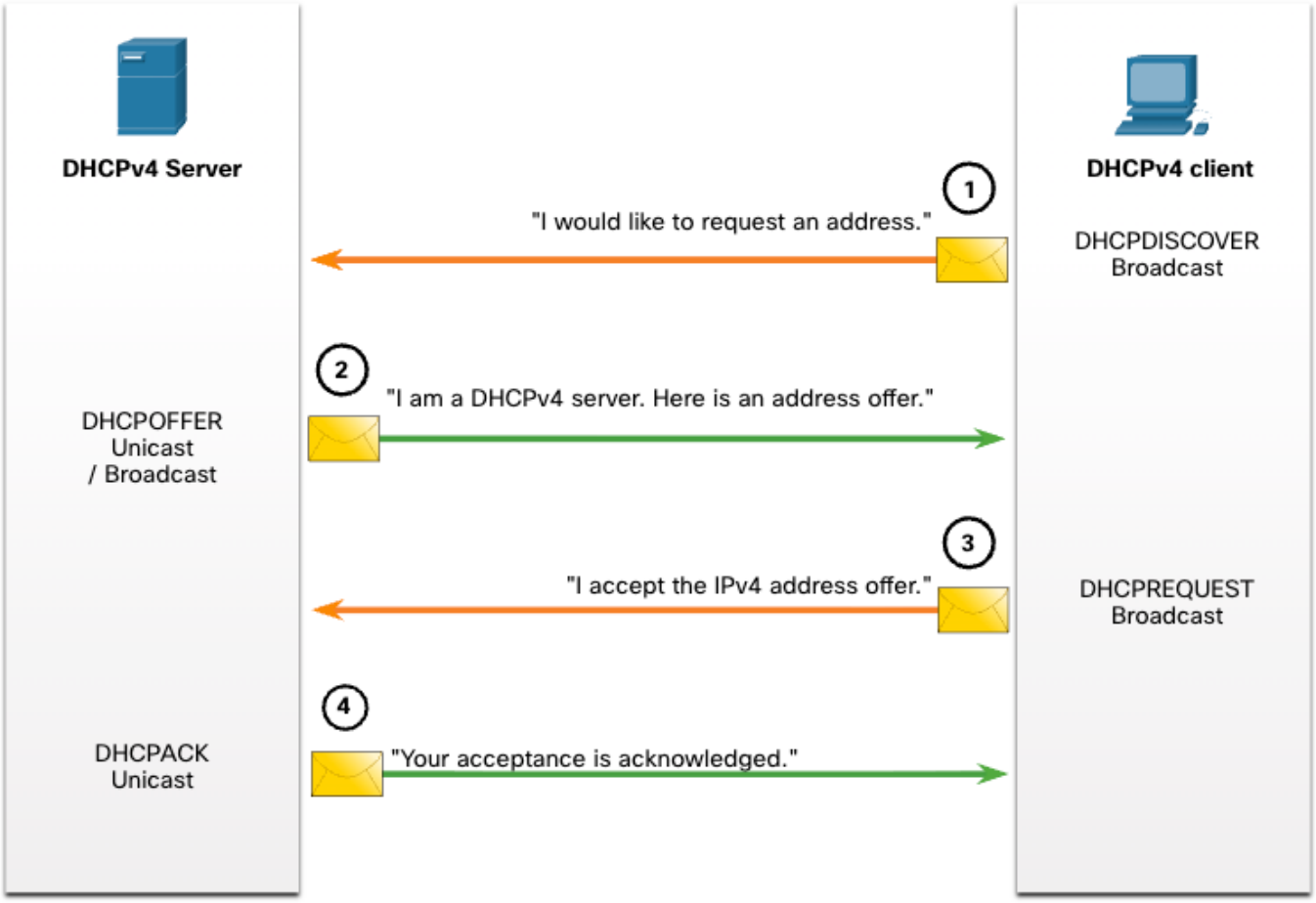
When the client receives the DHCPOFFER from the server, it sends back a DHCPREQUEST message. This message is used for both lease origination and lease renewal. When used for lease origination, the DHCPREQUEST serves as a binding acceptance notice to the selected server for the parameters it has offered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

Many enterprise networks use multiple DHCPv4 servers. The DHCPREQUEST message is sent in the form of a broadcast to inform this DHCPv4 server and any other DHCPv4 servers about the accepted offer.



Step 4. DHCP Acknowledgment (DHCPACK)

On receiving the DHCPREQUEST message, the server may verify the lease information with an ICMP ping to that address to ensure it is not being used already, it will create a new ARP entry for the client lease, and reply with a DHCPACK message. The DHCPACK message is a duplicate of the DHCPOFFER, except for a change in the message type field. When the client receives the DHCPACK message, it logs the configuration information and may perform an ARP lookup for the assigned address. If there is no reply to the ARP, the client knows that the IPv4 address is valid and starts using it as its own.



Steps to Renew a Lease

Prior to lease expiration, the client begins a two-step process to renew the lease with the DHCPv4 server, as shown in the figure:

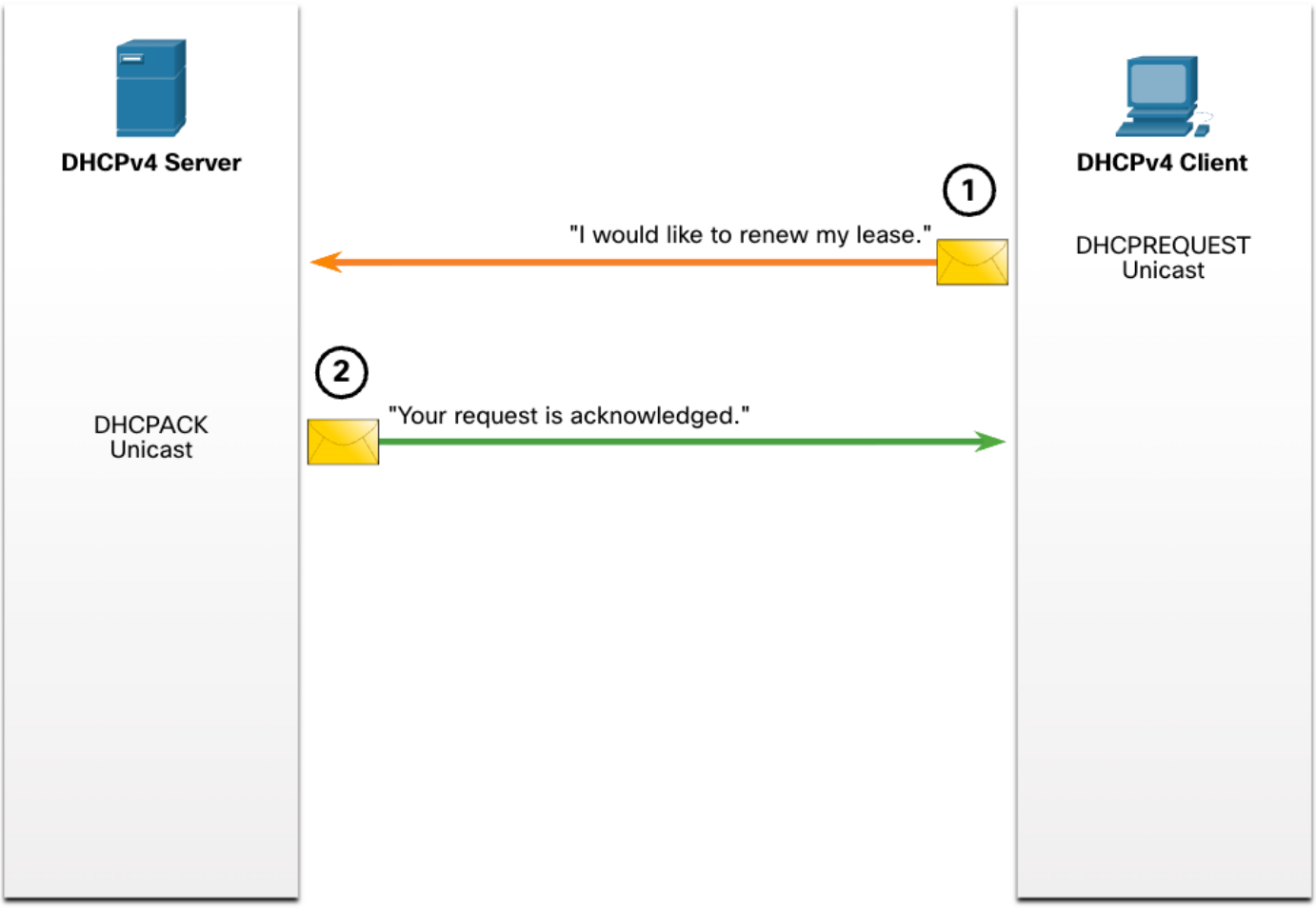
1. DHCP Request (DHCPREQUEST)

Before the lease expires, the client sends a DHCPREQUEST message directly to the DHCPv4 server that originally offered the IPv4 address. If a DHCPACK is not received within a specified amount of time, the client broadcasts another DHCPREQUEST so that one of the other DHCPv4 servers can extend the lease.

2. DHCP Acknowledgment (DHCPACK)

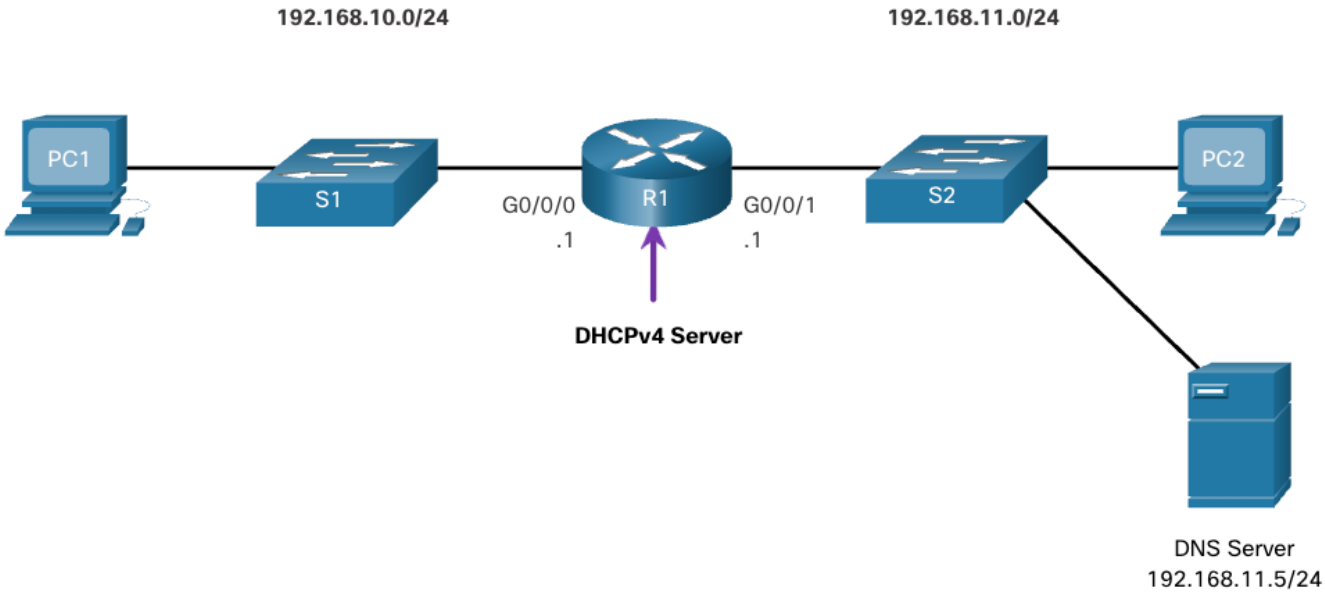
On receiving the DHCPREQUEST message, the server verifies the lease information by returning a DHCPACK.

Note: These messages (primarily the DHCPOFFER and DHCPACK) can be sent as unicast or broadcast according to IETF RFC 2131.



Configure DHCP server for Cisco IOS

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.



Steps to Configure a Cisco IOS DHCPv4 Server

Step 1. Exclude IPv4 addresses.
Step 2. Define a DHCPv4 pool name.
Step 3. Configure the DHCPv4 pool.

Step 1. Exclude IPv4 Addresses

The router functioning as the DHCPv4 server assigns all IPv4 addresses in a DHCPv4 address pool unless it is 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. The command syntax to exclude IPv4 addresses is the following:

```
Router(config)# ip dhcp excluded-address low-address [high-address]
```

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 be those addresses that are assigned to routers, servers, printers, and other devices that have been, or will be, manually configured. You can also enter the command multiple times.

Step 2. Define a DHCPv4 Pool Name

Configuring a DHCPv4 server involves defining a pool of addresses to assign.

As shown in the example, 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 the prompt Router(dhcp-config)#.

The command syntax to define the pool is the following:

```
Router(config)# ip dhcp pool pool-name
Router(dhcp-config)#
```

Step 3. Configure the DHCPv4 Pool

The table lists the tasks to complete the DHCPv4 pool configuration.

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** 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. The **netbios-name-server** command is used to define the NetBIOS WINS server.

Configuration Example

```
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#
```

Juniper Router configuration:

Define Address Pool: Specify the subnet, range of IP addresses, and default gateway

set access address-assignment pool POOL_NAME family inet network 192.168.10.0/24
set access address-assignment pool POOL_NAME family inet range RANGE_NAME low 192.168.10.100
set access address-assignment pool POOL_NAME family inet range RANGE_NAME high 192.168.10.200
set access address-assignment pool POOL_NAME family inet dhcp-attributes router 192.168.10.1
set access address-assignment pool POOL_NAME family inet dhcp-attributes dns-server 8.8.8.8

Configure DHCP Local Server: Associate the pool with an interface

set system services dhcp-local-server group GROUP_NAME interface g0/0/0

Configure Layer 3 Interface (IRB): Ensure the interface has an IP address

set interfaces irb unit 0 family inet address 192.168.10.1/24

Cisco Router as a DHCPv4 Client

There are scenarios where you might have access to a DHCP server through your ISP. In these instances, you can configure a Cisco IOS router as a DHCPv4 client. This topic walks you through that process.

Sometimes, Cisco routers in a small office or home office (SOHO) and branch sites have to be configured as DHCPv4 clients in a similar manner to client computers. The method used depends on the ISP. However, in its simplest configuration, the Ethernet interface is used to connect to a cable or DSL modem.

To configure an Ethernet interface as a DHCP client, use the **ip address dhcp** interface configuration mode command.

In the figure, assume that an ISP has been configured to provide select customers with IP addresses from the 209.165.201.0/27 network range after the G0/0/1 interface is configured with the **ip address dhcp** command.

To configure an Ethernet interface as a DHCP client, use the **ip address dhcp** interface configuration mode command, as shown in the example. This configuration assumes that the ISP has been configured to provide select customers with IPv4 addressing information.

```
SOHO(config)# interface G0/0/1
SOHO(config-if)# ip address dhcp
SOHO(config-if)# no shutdown
```

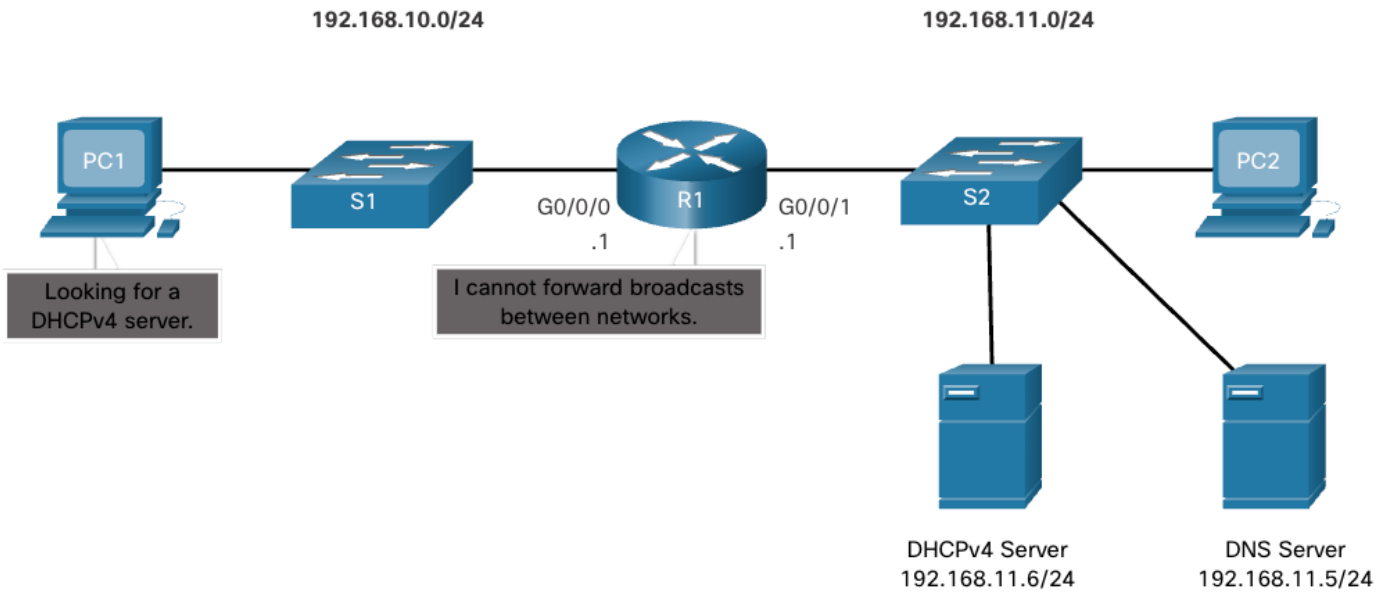
For Junos you can use command:

set interfaces ge-0/0/0 unit 0 family inet dhcp

DHCPv4 Relay

In a complex hierarchical network, enterprise servers are usually located centrally. These servers may provide DHCP, DNS, TFTP, and FTP services for the network. Network clients are not typically on the same subnet as those servers. In order to locate the servers and receive services, clients often use broadcast messages.

In the figure, PC1 is attempting to acquire an IPv4 address from a DHCPv4 server using a broadcast message. In this scenario, R1 is not configured as a DHCPv4 server and does not forward the broadcast. Because the DHCPv4 server is located on a different network, PC1 cannot receive an IP address using DHCP. R1 must be configured to relay DHCPv4 messages to the DHCPv4 server.



ip helper-address

A better solution is to configure R1 with the **ip helper-address** *address* interface configuration command. This will cause R1 to relay DHCPv4 broadcasts to the DHCPv4 server. As shown in the example, the interface on R1 receiving the broadcast from PC1 is configured to relay DHCPv4 address to the DHCPv4 server at 192.168.11.6.

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

For Junos:

```
set forwarding-options dhcp-relay server-group my-dhcp-servers 192.168.11.6
set forwarding-options dhcp-relay active-server-group my-dhcp-servers
set forwarding-options dhcp-relay group DHCP interface g0/0/0
```

To show DHCP bindings, use command:

For Cisco:

R1#Show dhcp bindings

For Junos:

show dhcp server binding

Chapter 7
Access Control Lists

Routers make routing decisions based on information in the packet header. Traffic entering a router interface is routed solely based on information within the routing table. The router compares the destination IP address with routes in the routing table to find the best match and then forwards the packet based on the best match route. That same process can be used to filter traffic using an access control list (ACL).

For Cisco

An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header. By default, a router does not have any ACLs configured. 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 uses a sequential list of permit or deny statements, known as access control entries (ACEs).

Note: ACEs are also commonly called ACL statements.

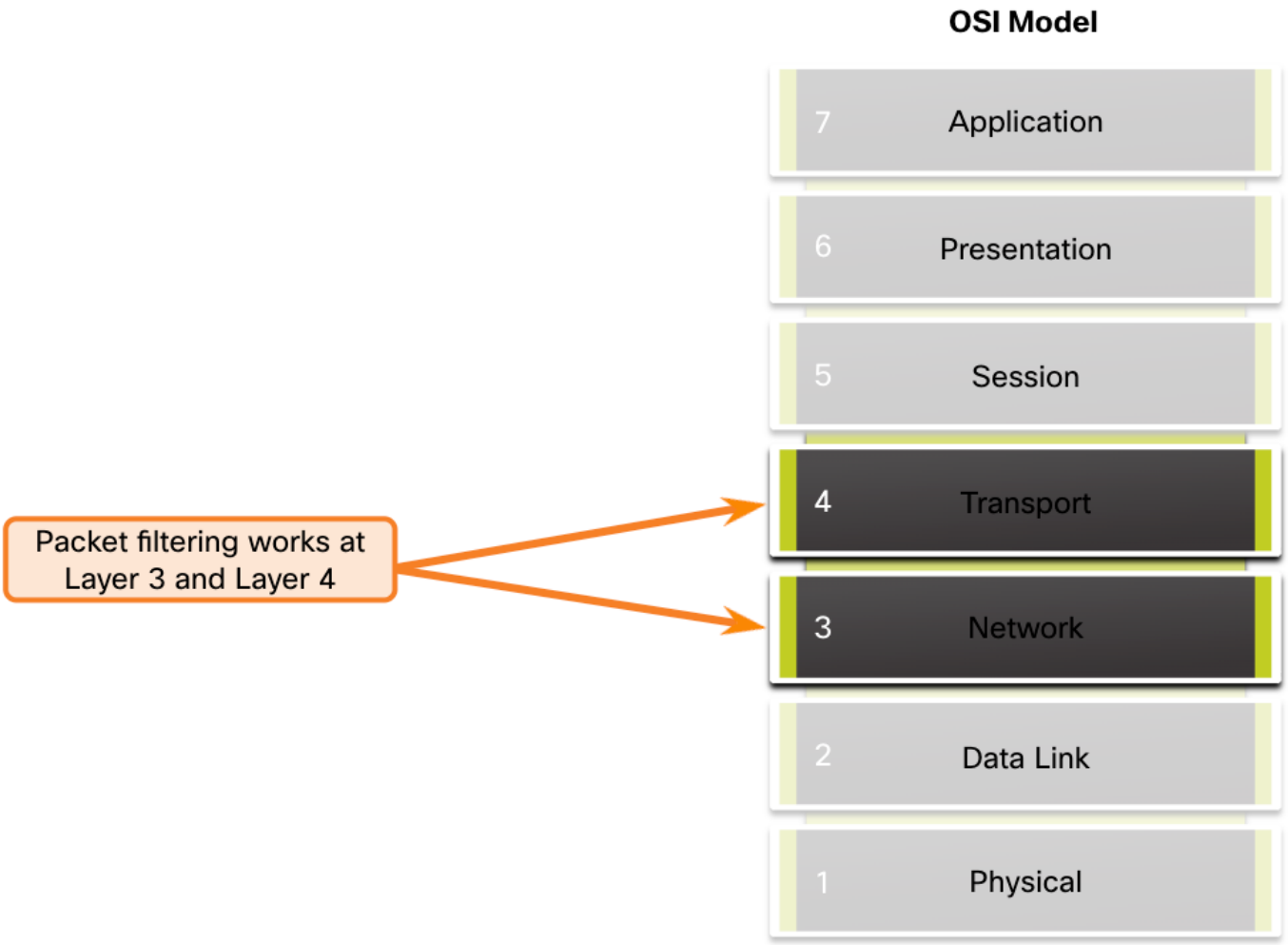
When network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each ACE, in sequential order, to determine if the packet matches one of the ACEs. This process is called packet filtering.

Several tasks performed by routers require the use of ACLs to identify traffic. The table lists some of these tasks with examples.

Task	Example
Limit network traffic to increase network performance	• A corporate policy prohibits video traffic on the network to reduce the network load. • A policy can be enforced using ACLs to block video traffic.
Provide traffic flow control	• A corporate policy requires that routing protocol traffic be limited to certain links only. • A policy can be implemented using ACLs to restrict the delivery of routing updates to only those that come from a known source.
Provide a basic level of security for network access	• Corporate policy demands that access to the Human Resources network be restricted to authorized users only. • A policy can be enforced using ACLs to limit access to specified networks.
Filter traffic based on traffic type	• Corporate policy requires that email traffic be permitted into a network, but that Telnet access be denied. • A policy can be implemented using ACLs to filter traffic by type.
Screen hosts to permit or deny access to network services	• Corporate policy requires that access to some file types (e.g., FTP or HTTP) be limited to user groups. • A policy can be implemented using ACLs to filter user access to services.
Provide priority to certain classes of network traffic	• Corporate traffic specifies that voice traffic be forwarded as fast as possible to avoid any interruption. • A policy can be implemented using ACLs and QoS services to identify voice traffic and process it immediately.

Packet Filtering

Packet filtering controls access to a network by analyzing the incoming and/or outgoing packets and forwarding them or discarding them based on given criteria. Packet filtering can occur at Layer 3 or Layer 4, as shown in the figure.



Cisco routers support two types of ACLs:

- **Standard ACLs** - ACLs only filter at Layer 3 using the source IPv4 address only.
- **Extended ACLs** - ACLs filter at Layer 3 using the source and / or destination IPv4 address. They can also filter at Layer 4 using TCP, UDP ports, and optional protocol type information for finer control.

ACL Operation

ACLs define the set of rules that give added control for packets that enter inbound interfaces, packets that relay through the router, and packets that exit outbound interfaces of the router.

ACLs can be configured to apply to inbound traffic and outbound traffic, as shown in the figure.

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

An inbound ACL filters packets before they are routed to the outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the packet is discarded. If the packet is permitted by the ACL, 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 packets that need to be examined.

An outbound ACL filters packets after being routed, regardless of the inbound interface. 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.

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

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

The last ACE statement of an ACL is always an implicit deny that blocks all traffic. By default, this statement is automatically implied at the end of an ACL even though it is hidden and not displayed in the configuration.

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

Wildcard Mask Overview

In the previous topic, you learned about the purpose of ACL. This topic explains how ACL uses wildcard masks. An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match. Wildcard masks are also used by the Open Shortest Path First (OSPF) routing protocol.

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

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 table lists some examples of wildcard masks and what they would identify.

Wildcard Mask	Last Octet (in Binary)	Meaning (0 - match, 1 - ignore)
`0.0.0.0`	`00000000`	Match all octets.
`0.0.0.63`	`00111111`	• Match the first three octets • Match the two left most bits of the last octet • Ignore the last 6 bits
`0.0.0.15`	`00001111`	• Match the first three octets • Match the four left most bits of the last octet • Ignore the last 4 bits of the last octet
`0.0.0.252`	`11111100`	• Match the first three octets • Ignore the six left most bits of the last octet • Match the last two bits
`0.0.0.255`	`11111111`	• Match the first three octet • Ignore the last octet

Wildcard Mask Keywords

Working with decimal representations of binary wildcard mask bits can be tedious. To simplify this task, the Cisco IOS provides two keywords to identify the most common uses of wildcard masking. Keywords reduce ACL keystrokes and make it easier to read the ACE.

The two keywords are:

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

For example, in the command output, two ACLs are configured. The ACL 10 ACE permits only the 192.168.10.10 host and the ACL 11 ACE permits all hosts.

```
R1(config)# access-list 10 permit 192.168.10.10 0.0.0.0
R1(config)# access-list 11 permit 0.0.0.0 255.255.255.255
R1(config)#
```

Alternatively, the keywords **host** and **any** could have been used to replace the highlighted output.

The following commands accomplishes the same task as the previous commands.

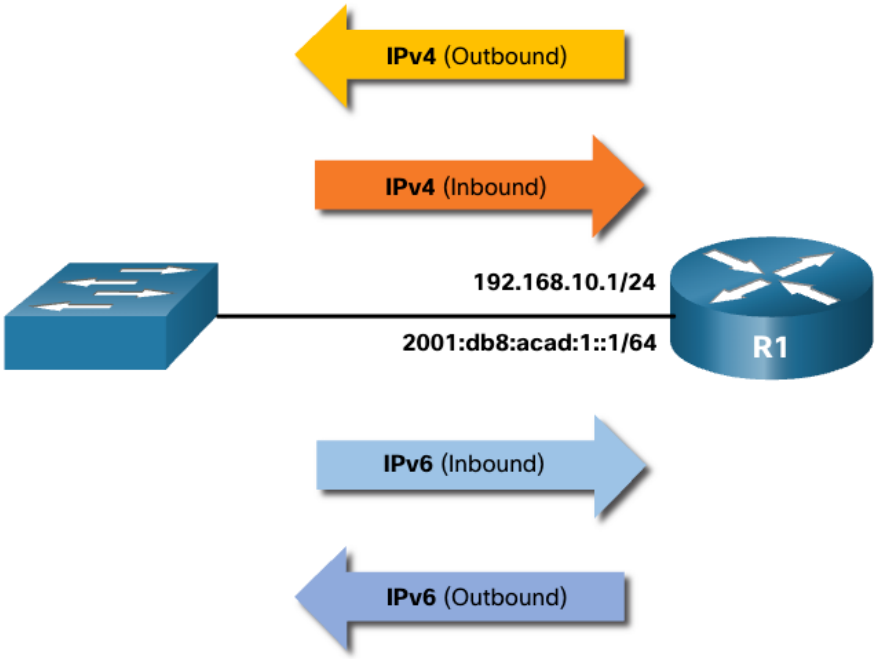
```
R1(config)# access-list 10 permit host 192.168.10.10
R1(config)# access-list 11 permit any
R1(config)#
```

Limited Number of ACLs per Interface

In a previous topic, you learned about how wildcard masks are used in ACLs. This topic will focus on the guidelines for ACL creation. There is a limit on the number of ACLs that can be applied on a router interface. For example, a dual-stacked (i.e., IPv4 and IPv6) router interface can have up to four ACLs applied, as shown in the figure.

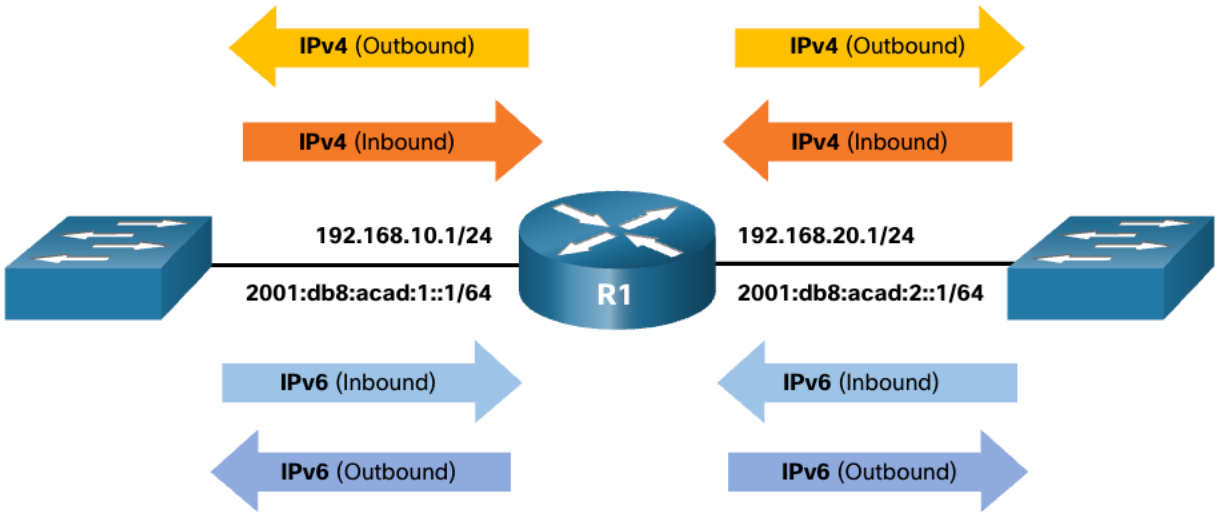
Specifically, a router interface can have:

- one outbound IPv4 ACL
- one inbound IPv4 ACL
- one inbound IPv6 ACL
- one outbound IPv6 ACL



Assume R1 has two dual-stacked interfaces that require inbound and outbound IPv4 and IPv6 ACLs applied. As shown in the figure, R1 could have up to 8 ACLs configured and applied to interfaces. 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.

Note: ACLs do not have to be configured in both directions. The number of ACLs and their direction applied to the interface will depend on the security policy of the organization.



ACL Best Practices

Using ACLs requires attention to detail and great care. Mistakes can be costly in terms of downtime, troubleshooting efforts, and poor network service. Basic planning is required before configuring an ACL.

The table presents guidelines that form the basis of an ACL best practices list.

Guideline	Benefit
Base ACLs on the organizational security policies.	This will ensure you implement organizational security guidelines.
Write out what you want the ACL to do.	This will help you avoid inadvertently creating potential access problems.
Use a text editor to create, edit, and save all of your ACLs.	This will help you create a library of reusable ACLs.
Document the ACLs using the **remark** command.	This will help you (and others) understand the purpose of an ACE.
Test the ACLs on a development network before implementing them on a production network.	This will help you avoid costly errors.

Standard and Extended ACLs

The previous topics covered the purpose of ACL and the guidelines for ACL creation. This topic will cover standard and extended ACLs, named and numbered ACLs, and the examples of placement of these ACLs.

There are two types of IPv4 ACLs:

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

For example, refer to the following standard ALC command.

```
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
R1(config)#
```

ACL 10 permits hosts on the source network 192.168.10.0/24. Because of the implied "deny any" at the end, all traffic except for traffic coming from the 192.168.10.0/24 network is blocked with this ACL.

In the next example, an extended ACL 100 permits traffic originating from any host on the 192.168.10.0/24 network to any IPv4 network if the destination host port is 80 (HTTP).

```
R1(config)# access-list 100 permit tcp 192.168.10.0 0.0.0.255 any eq www
R1(config)#
```

Notice how the standard ACL 10 is only capable of filtering by source address while the extended ACL 100 is filtering on the source, and destination Layer 3, and Layer 4 protocol (i.e., TCP) information.

Numbered and Named ACLs

Numbered ACLs

ACLs number 1 to 99, or 1300 to 1999 are standard ACLs while ACLs number 100 to 199, or 2000 to 2699 are extended ACLs, as shown in the output.

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

Named ACLs

Named ACLs is the preferred method to use when configuring ACLs. Specifically, standard and extended ACLs can be named to provide information about the purpose of the ACL. For example, naming an extended ACL FTP-FILTER is far better than having a numbered ACL 100.

The **ip access-list** global configuration command is used to create a named ACL, as shown in the following example.

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

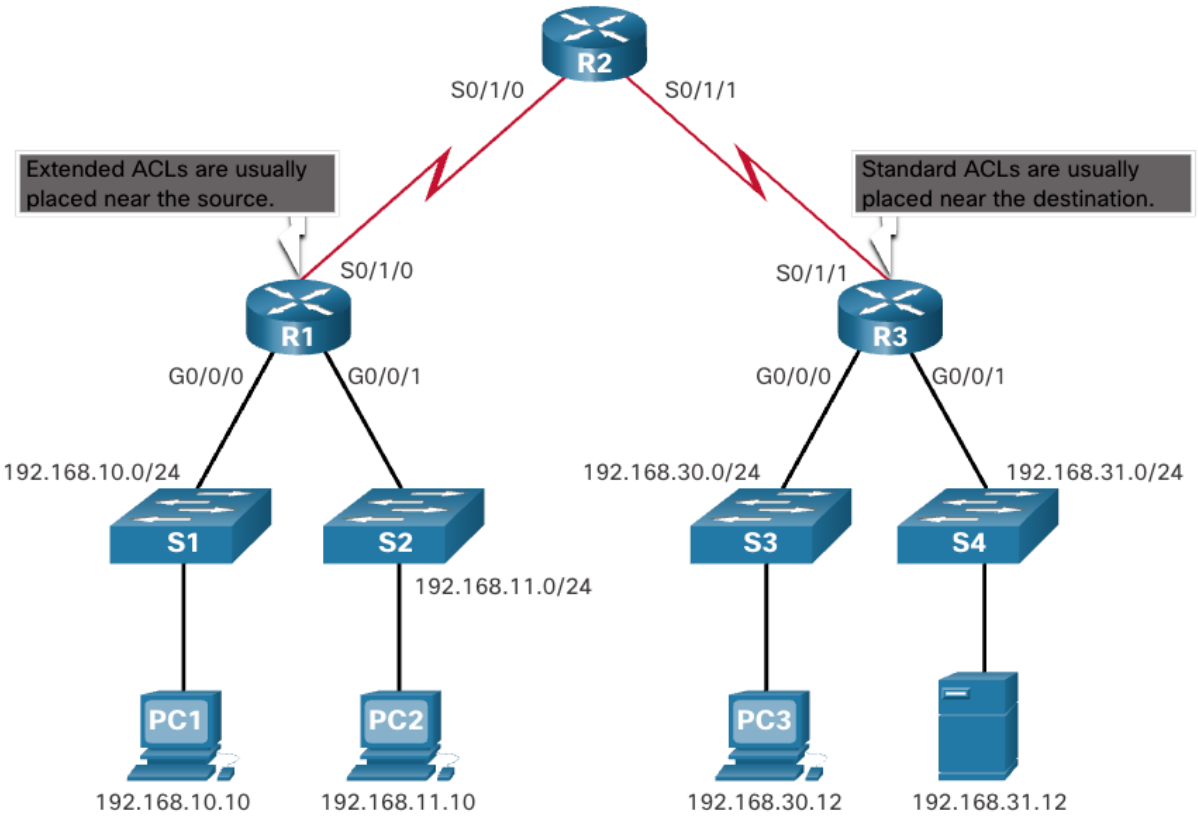
The following summarizes the rules to follow for named ACLs.

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

Where to Place ACLs

Every ACL should be placed where it has the greatest impact on efficiency.

The figure illustrates where standard and extended ACLs should be located in an enterprise network. Assume the objective to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.



Extended ACLs should be located 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 should be located as close to the destination as possible. If a standard ACL was placed at the source of the traffic, the "permit" or "deny" will occur based on the given source address no matter where the traffic is destined.

Example of ACLs

Standard numbered ACL

```
Router(config)# access-list 10 deny 192.168.10.0 0.0.0.255
Router(config)# access-list 10 permit any
```

Standard named ACL

```
Router(config)# ip access-list standard OFFICE-LAN
Router(config-std-nacl)# permit 192.168.1.0 0.0.0.255
Router(config-std-nacl)# deny any
Router(config-std-nacl)# exit
```

Apply a Standard IPv4 ACL

After a standard IPv4 ACL is configured, it must be linked to an interface or feature. The following command can be used to bind a numbered or named standard IPv4 ACL to an interface:

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

Example:

```
Router(config-if)#ip access-group 10 in
```

To remove an ACL from an interface, first enter the **no ip access-group** interface configuration command. However, the ACL will still be configured on the router. To remove the ACL from the router, use the **no access-list** global configuration command.

Two Methods to Modify an ACL

After an ACL is configured, it may need to be modified. ACLs with multiple ACEs can be complex to configure. Sometimes the configured ACE does not yield the expected behaviors. For these reasons, ACLs may initially require a bit of trial and error to achieve the desired filtering result.

This section will discuss two methods to use when modifying an ACL:

- Use a Text Editor
- Use Sequence Numbers

Text Editor Method

ACLs with multiple ACEs should be created in a text editor. This allows you to plan the required ACEs, create the ACL, and then paste it into the router interface. It also simplifies the tasks to edit and fix an ACL.

For example, assume ACL 1 was entered incorrectly using **19** instead of **192** for the first octet, as shown in the running configuration.

```
R1# show run | section access-list
access-list 1 deny 19.168.10.10
access-list 1 permit 192.168.10.0 0.0.0.255
R1#
```

In the example, the first ACE should have been to deny the host at 192.168.10.10. However, the ACE was incorrectly entered.

To correct the error:

- Copy the ACL from the running configuration and paste it into the text editor.
- Make the necessary edits changes.
- Remove the previously configured ACL on the router otherwise, pasting the edited ACL commands will only append (i.e., add) to the existing ACL ACEs on the router.
- Copy and paste the edited ACL back to the router.

Assume that ACL 1 has now been corrected. Therefore, the incorrect ACL must be deleted, and the corrected ACL 1 statements must be pasted in global configuration mode, as shown in the output.

```
R1(config)# no access-list 1
R1(config)#
R1(config)# access-list 1 deny 192.168.10.10
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#
```

Sequence Numbers Method

An ACL ACE can also be deleted or added using the ACL sequence numbers. Sequence numbers are automatically assigned when an ACE is entered. These numbers are listed in the **show access-lists** command. The **show running-config** command does not display sequence numbers.

In the previous example, the incorrect ACE for ACL 1 is using sequence number 10, as shown in the example.

```
R1# show access-lists
Standard IP access list 1
    10 deny 19.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Use the **ip access-list standard** command to edit an ACL. Statements cannot be overwritten using the same sequence number as an existing statement. Therefore, the current statement must be deleted first with the **no 10** command. Then the correct ACE can be added using sequence number 10 is configured. Verify the changes using the **show access-lists** command, as shown in the example.

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Extended ACLs

In the previous topics, you learned about how to configure and modify standard ACLs, and how to secure VTY ports with a standard IPv4 ACL. Standard ACLs only filter on source address. When more precise traffic-filtering control is required, extended IPv4 ACLs can be created.

Extended ACLs are used more often than standard ACLs because they provide a greater degree of control. They can filter on source address, destination address, protocol (i.e., IP, TCP, UDP, ICMP), and port number. This provides a greater range of criteria on which to base the ACL. For example, one extended ACL can allow email traffic from a network to a specific destination while denying file transfers and web browsing.

Like standard ACLs, extended ACLs can be created as:

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

Numbered Extended IPv4 ACL Syntax

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.

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

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

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

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

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

Apply a Numbered Extended IPv4 ACL

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

Named Extended IPv4 ACL Syntax

Naming an ACL makes it easier to understand its function. To create a named extended ACL, use the following global configuration command:

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

Junos

In Junos ACLs are called Firewall Filter. Also there are no standard or extended ACLs in Junos.

Here is the Cisco Standard ACL equivalent:

set firewall family inet filter BLOCK-SOURCE term 1 from source-address 192.168.10.0/24
set firewall family inet filter BLOCK-SOURCE term 1 then discard
set firewall family inet filter BLOCK-SOURCE term 2 then accept

Cisco Extended ACL Equivalent

```
set firewall family inet filter BLOCK-HTTP term 1 from source-address 192.168.10.0/24
set firewall family inet filter BLOCK-HTTP term 1 from protocol tcp
set firewall family inet filter BLOCK-HTTP term 1 from destination-port 80
set firewall family inet filter BLOCK-HTTP term 1 then discard
set firewall family inet filter BLOCK-HTTP term 2 then accept
```

To apply Firewall filter:

```
set interfaces ge-0/0/0 unit 0 family inet filter input BLOCK-HTTP
```

Chapter 8
Routing concepts, Static route

Two Functions of Router

Before a router forwards a packet anywhere, it has to determine the best path for the packet to take. This topic explains how routers make this determination.

Ethernet switches are used to connect end devices and other intermediary devices, such as other Ethernet switches, to the same network. A router connects multiple networks, which means that it has multiple interfaces that each belong to a different IP network.

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

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

Best Path Equals Longest Match

What is meant by the router must determine the best path in the routing table? The best path in the routing table is also known as the longest match. The longest match is a process the router uses to find a match between the destination IP address of the packet and a routing entry in the routing table.

The routing table contains route entries consisting of a prefix (network address) and prefix length. For there to be a match between the destination IP address of a packet and a route in the routing

table, a minimum number of far-left bits must match between the IP address of the packet and the route in the routing table. The prefix length of the route in the routing table is used to determine the minimum number of far-left bits that must match. Remember that an IP packet only contains the destination IP address and not the prefix length.

The longest match is the route in the routing table that has the greatest number of far-left matching bits with the destination IP address of the packet. The route with the greatest number of equivalent far-left bits, or the longest match, is always the preferred route.

Note: The term prefix length will be used to refer to the network portion of both IPv4 and IPv6 addresses.

IPv4 Address Longest Match Example

In the table, an IPv4 packet has the destination IPv4 address 172.16.0.10. The router has three route entries in its IPv4 routing table that match this packet: 172.16.0.0/12, 172.16.0.0/18, and 172.16.0.0/26. Of the three routes, 172.16.0.0/26 has the longest match and would be chosen to forward the packet. Remember, for any of these routes to be considered a match there must be at least the number of matching bits indicated by the subnet mask of the route.

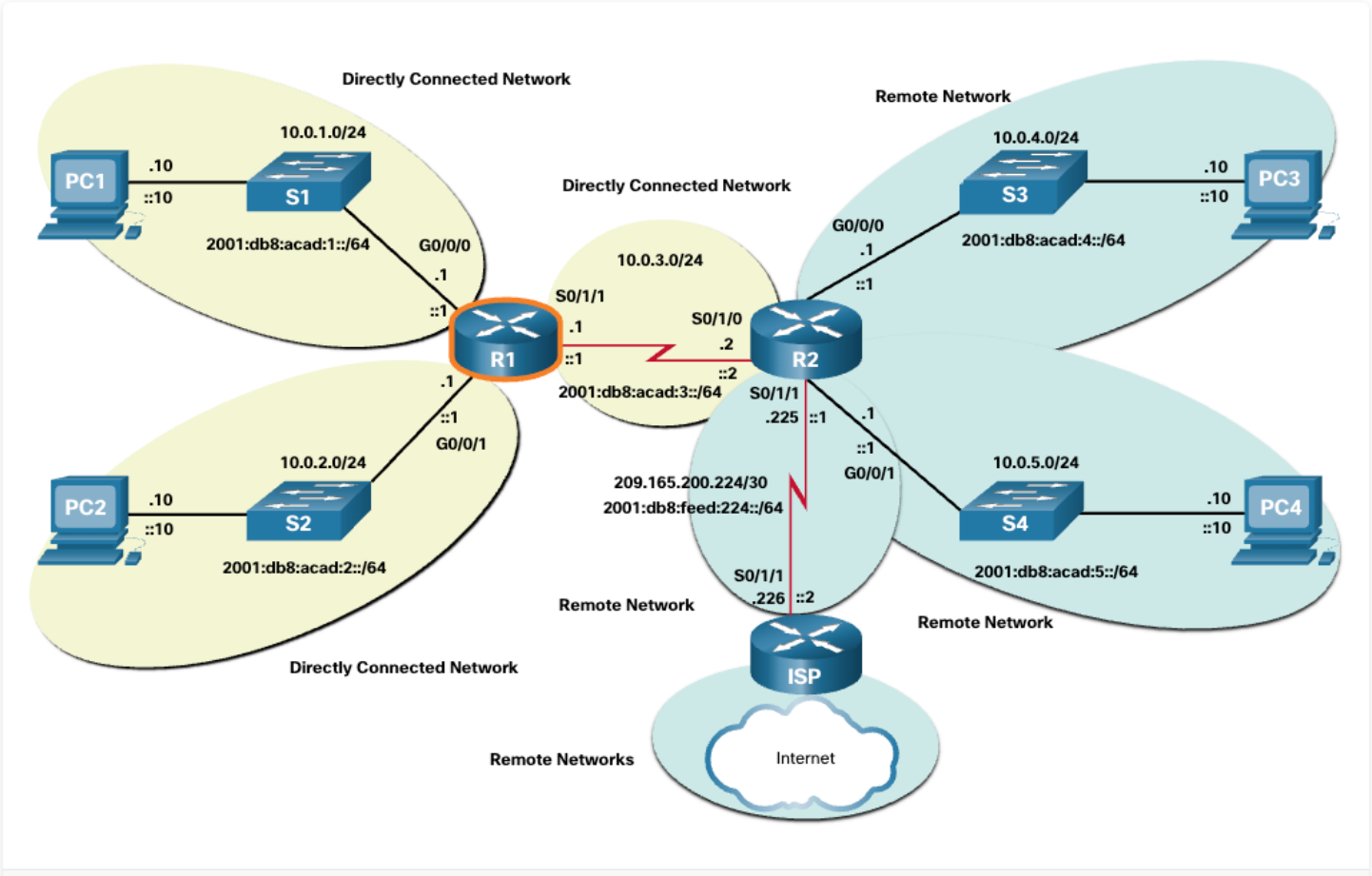
Destination IPv4 Address	Address in Binary
172.16.0.10	**10101100.00010000.00000000.00**001010

Route Entry	Prefix/Prefix Length	Address in Binary
1	172.16.0.0/**12**	**10101100.0001**0000.00000000.00001010
2	172.16.0.0/**18**	**10101100.00010000.00**000000.00001010
3	172.16.0.0/**26**	**10101100.00010000.00000000.00**001010

Build the Routing Table

A routing table consists of prefixes and their prefix lengths. But how does the router learn about these networks? How does R1 in the figure populate its routing table?

Networks from the Perspective of R1



Directly Connected Networks

Directly connected networks are networks that are configured on the active interfaces of a router. A directly connected network is added to the routing table when an interface is configured with an IP address and subnet mask (prefix length) and is active

Remote Networks

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

- **Static routes** - Added to the routing table when a route is manually configured.
- **Dynamic routing protocols** - Added to the routing table when routing protocols dynamically learn about the remote network. Dynamic routing protocols include Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), as well as several others.

Default Route

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

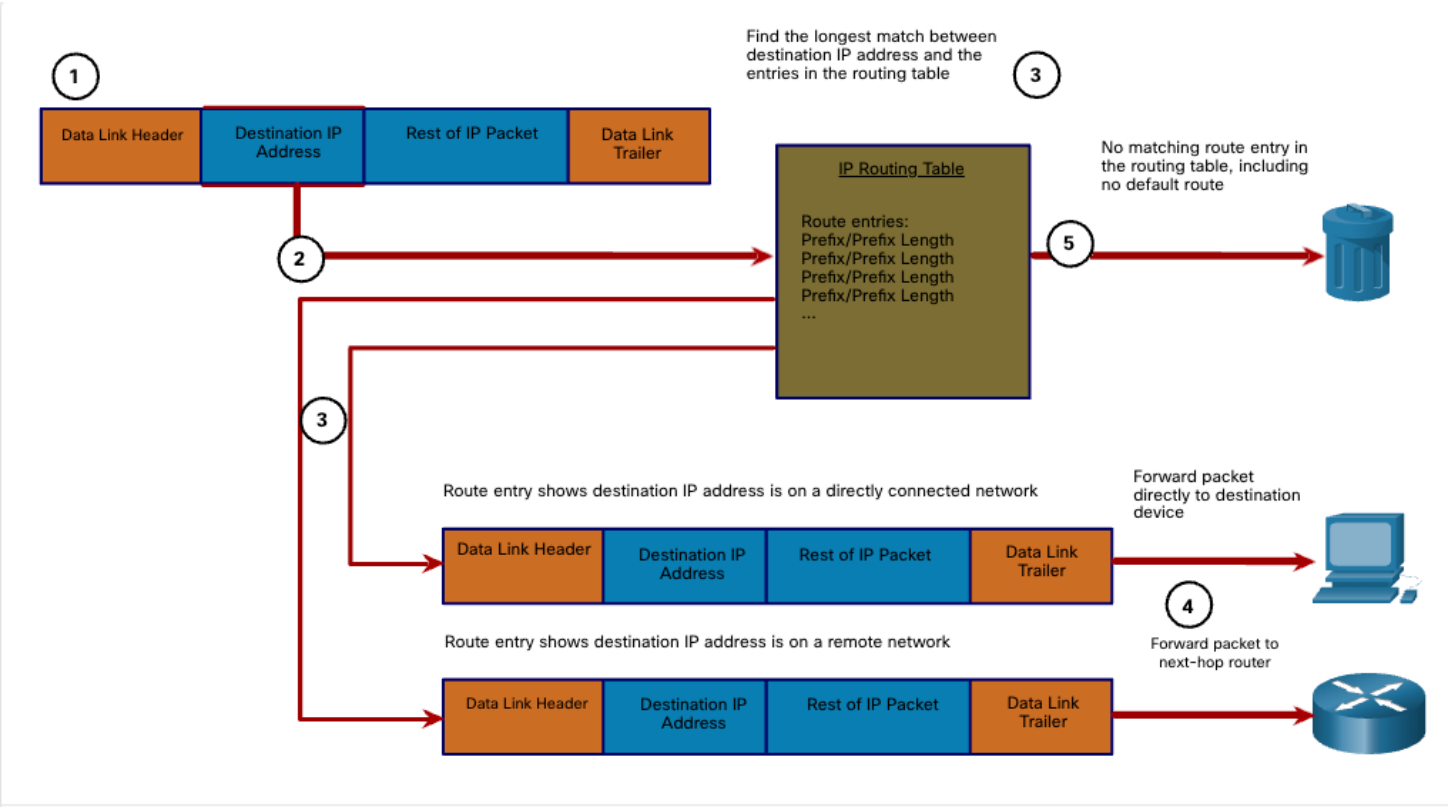
A default route over IPv4 has a route entry of 0.0.0.0/0 and a default route over IPv6 has a route entry of ::/0. The /0 prefix length indicates that zero bits or no bits need to match the destination IP address for this route entry to be used. If there are no routes with a longer match, more than 0

bits, then the default route is used to forward the packet. The default route is sometimes referred to as a gateway of last resort.

Packet Forwarding Decision Process

Now that the router has determined the best path for a packet based on the longest match, it must determine how to encapsulate the packet and forward it out the correct egress interface.

The figure demonstrates how a router first determines the best path, and then forwards the packet.



The following steps describe the packet forwarding process shown in the figure:

1. *The data link frame with an encapsulated IP packet arrives on the ingress interface.*
2. *The router examines the destination IP address in the packet header and consults its IP routing table.*
3. *The router finds the longest matching prefix in the routing table.*
4. *The router encapsulates the packet in a data link frame and forwards it out the egress interface. The destination could be a device connected to the network or a next-hop router.*
5. *However, if there is no matching route entry the packet is dropped.*

Forwards the Packet to a Device on a Directly Connected Network

If the route entry indicates that the egress interface is a directly connected network, this means that the destination IP address of the packet belongs to a device on the directly connected network. Therefore, the packet can be forwarded directly to the destination device. The destination device is typically an end device on an Ethernet LAN, which means the packet must be encapsulated in an Ethernet frame.

To encapsulate the packet in the Ethernet frame, the router needs to determine the destination MAC address associated with the destination IP address of the packet. The process varies based on whether the packet is an IPv4 or IPv6 packet:

IPv4 packet - The router checks its ARP table for the destination IPv4 address and an associated Ethernet MAC address. If there is no match, the router sends an ARP Request. The destination device will return an ARP Reply with its MAC address. The router can now forward the IPv4 packet in an Ethernet frame with the proper destination MAC address.

Forwards the Packet to a Next-Hop Router

If the route entry indicates that the destination IP address is on a remote network, this means the destination IP address of the packet belongs to a device on network that is not directly connected. Therefore, the packet must be forwarded to another router, specifically a next-hop router. The next-hop address is indicated in the route entry.

If the forwarding router and the next-hop router are on an Ethernet network, a similar process (ARP and ICMPv6 Neighbor Discovery) will occur for determining the destination MAC address of the packet as described previously. The difference is that the router will search for the IP address of the next-hop router in its ARP table or neighbor cache, instead of the destination IP address of the packet.

Note: This process will vary for other types of Layer 2 networks.

Drops the Packet - No Match in Routing Table

If there is no match between the destination IP address and a prefix in the routing table, and if there is no default route, the packet will be dropped.

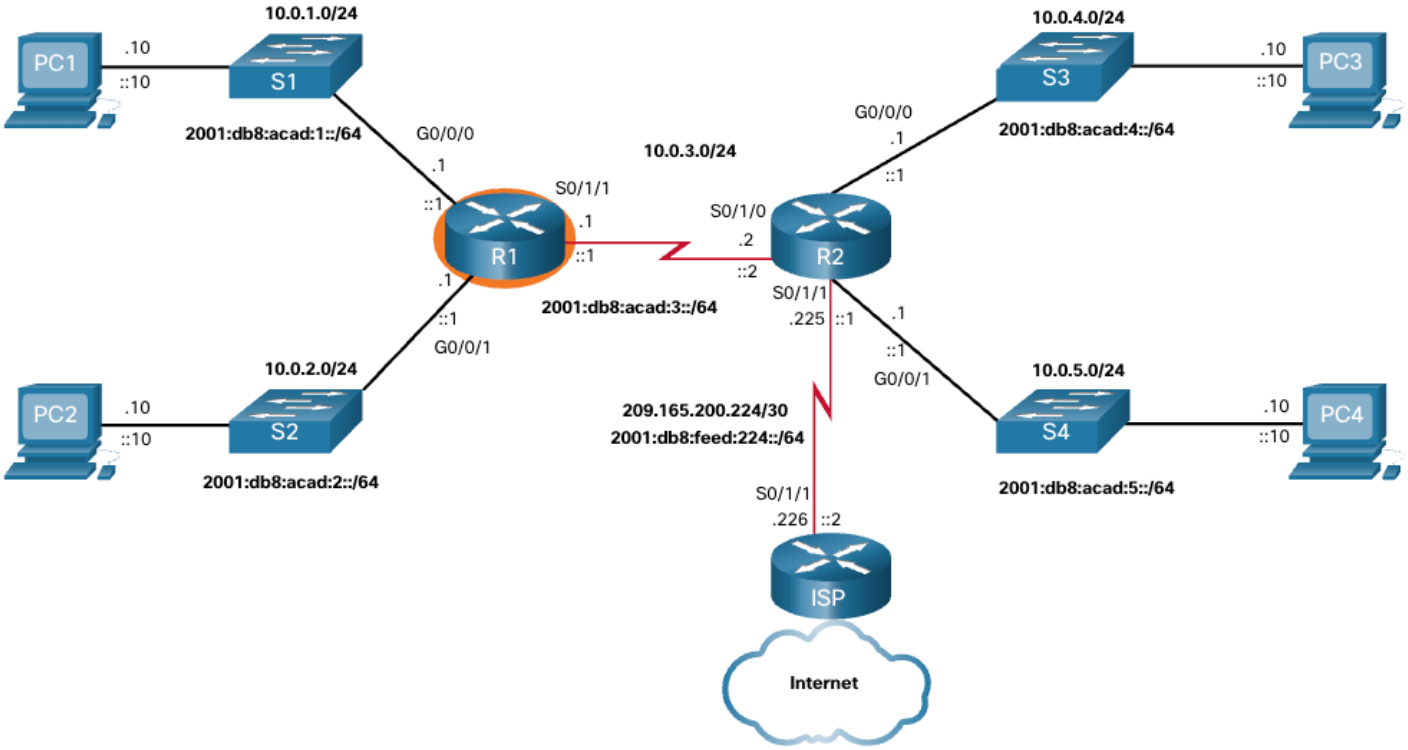
Route Sources

How does a router know where it can send packets? It creates a routing table that is based on the network in which it is located.

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

- Directly connected networks
- Static routes
- Dynamic routing protocols

In the figure, R1 and R2 are using the dynamic routing protocol OSPF to share routing information. In addition, R2 is configured with a default static route to ISP.



Routing Table Principles

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

Routing Table Principle	Example
Every router makes its decision alone, based on the information it has in its own routing table.	• R1 can only forward packets using its own routing table. • R1 does not know what routes are in the routing tables of other routers (e.g., R2).
The information in a routing table of one router does not necessarily match the routing table of another router.	Just because R1 has route in its routing table to a network in the internet via R2, that does not mean that R2 knows about that same network.
Routing information about a path does not provide return routing information.	R1 receives a packet with the destination IP address of PC1 and the source IP address of PC3. Just because R1 knows to forward the packet out its G0/0/0 interface, doesn't necessarily mean that it knows how to forward packets originating from PC1 back to the remote network of PC3.

Directly Connected Networks

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

Static Routes

After directly connected interfaces are configured and added to the routing table, static or dynamic routing can be implemented for accessing remote networks.

Static routes are manually configured. They define an explicit path between two networking devices. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured if the network topology changes. The benefits of using static routes include improved security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols, and no CPU cycles are used to calculate and communicate routes. The main disadvantage to using static routes is the lack of automatic reconfiguration if the network topology changes.

Static routing has three primary uses:

- It provides ease of routing table maintenance in smaller networks that are not expected to grow significantly.
- It uses a single default route to represent a path to any network that does not have a more specific match with another route in the routing table. Default routes are used to send traffic to any destination beyond the next upstream router.
- It routes to and from stub networks. A stub network is a network accessed by a single route, and the router has only one neighbor.

Dynamic Routing Protocols

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

Important advantages of dynamic routing protocols are the ability to select a best path, and the ability to automatically discover a new best path when there is a change in the topology.

Network discovery is the ability of a routing protocol to share information about the networks that it knows about with other routers that are also using the same routing protocol. Instead of depending on manually configured static routes to remote networks on every router, a dynamic routing protocol allows the routers to automatically learn about these networks from other routers. These networks, and the best path to each, are added to the routing table of the router, and identified as a network learned by a specific dynamic routing protocol.

Default Route

A default route is similar to a default gateway on a host. The default route specifies a next-hop router to use when the routing table does not contain a specific route that matches the destination IP address.

A default route can be either a static route or learned automatically from a dynamic routing protocol. A default route has an IPv4 route entry of 0.0.0.0/0 or an IPv6 route entry of ::/0. This means that zero or no bits need to match between the destination IP address and the default route.

Most enterprise routers have a default route in their routing table. This is to reduce the number of routes in a routing table.

A router, such as a home or small office router that only has one LAN, may reach all its remote networks through a default route. This is useful when the router has only directly connected networks and one exit point to a service provider router.

Static or Dynamic?

The previous topic discussed the ways that a router creates its routing table. So, you now know that routing, like IP addressing, can be either static or dynamic. Should you use static or dynamic routing? The answer is both! Static and dynamic routing are not mutually exclusive. Rather, most networks use a combination of dynamic routing protocols and static routes.

Static Routes

Static routes are commonly used in the following scenarios:

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

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

Dynamic Routing Protocols

Dynamic routing protocols help the network administrator manage the time-consuming and exacting process of configuring and maintaining static routes. Dynamic routing protocols are implemented in any type of network consisting of more than just a few routers. Dynamic routing protocols are scalable and automatically determine better routes if there is a change in the topology.

Dynamic routing protocols are commonly used in the following scenarios:

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

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

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

Dynamic Routing Protocol Concepts

A routing protocol is a set of processes, algorithms, and messages that are used to exchange routing information and populate the routing table with the choice of best paths. The purpose of dynamic routing protocols includes the following:

- Discovery of remote networks
- Maintaining up-to-date routing information
- Choosing the best path to destination networks
- Ability to find a new best path if the current path is no longer available

The main components of dynamic routing protocols include the following:

- **Data structures** - Routing protocols typically use tables or databases for their operations. This information is kept in RAM.
- **Routing protocol messages** - Routing protocols use various types of messages to discover neighboring routers, exchange routing information, and other tasks to learn and maintain accurate information about the network.
- **Algorithm** - An algorithm is a finite list of steps used to accomplish a task. Routing protocols use algorithms for facilitating routing information and for the best path determination.

Routing protocols allow routers to dynamically share information about remote networks and automatically offer this information to their own routing tables.

Routing protocols determine the best path, or route, to each network. That route is then offered to the routing table. The route will be installed in the routing table if there is not another routing source with a lower AD. A primary benefit of dynamic routing protocols is that routers exchange routing information when there is a topology change. This exchange allows routers to automatically learn about new networks and to find alternate paths when there is a link failure to a current network.

Best Path

Before a path to a remote network is offered to the routing table, the dynamic routing protocol must determine the best path to that network. Determining the best path may involve the evaluation of multiple paths to the same destination network and selecting the optimum or shortest path to reach that network. Whenever multiple paths to the same network exist, each path uses a different exit interface on the router to reach that network.

The best path is selected by a routing protocol based on the value or metric it uses to determine the distance to reach a network. A metric is the quantitative value used to measure the distance to a given network. The best path to a network is the path with the lowest metric.

Dynamic routing protocols typically use their own rules and metrics to build and update routing tables. The routing algorithm generates a value, or a metric, for each path through the network. Metrics can be based on either a single characteristic or several characteristics of a path. Some routing protocols can base route selection on multiple metrics, combining them into a single metric.

The following table lists common dynamic protocols and their metrics.

Routing Protocol	Metric
Routing Information Protocol (RIP)	• The metric is “hop count”. • Each router along a path adds a hop to the hop count. • A maximum of 15 hops allowed.
Open Shortest Path First (OSPF)	• The metric is “cost” which is the based on the cumulative bandwidth from source to destination. • Faster links are assigned lower costs compared to slower (higher cost) links.
Enhanced Interior Gateway Routing Protocol (EIGRP)	• It calculates a metric based on the slowest bandwidth and delay values. • It could also include load and reliability into the metric calculation.

Load Balancing

What happens if a routing table has two or more paths with identical metrics to the same destination network?

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

If configured correctly, load balancing can increase the effectiveness and performance of the network.

Equal cost load balancing is implemented automatically by dynamic routing protocols. It is enabled with static routes when there are multiple static routes to the same destination network using different next-hop routers.

Note: Only EIGRP supports unequal cost load balancing.

Types of Static Routes

Static routes are commonly implemented on a network. This is true even when there is a dynamic routing protocol configured. For instance, an organization could configure a default static route to the service provider and advertise this route to other corporate routers using the dynamic routing protocol.

Static routes can be configured for IPv4 and IPv6. Both protocols support the following types of static routes:

- Standard static route
- Default static route
- Floating static route
- Summary static route

Next-Hop Options

When configuring a static route, the next hop can be identified by an IP address, exit interface, or both. How the destination is specified creates one of the three following types of static route:

- **Next-hop route** - Only the next-hop IP address is specified
- **Directly connected static route** - Only the router exit interface is specified
- **Fully specified static route** - The next-hop IP address and exit interface are specified

IPv4 Static Route Command (Next hop IP address)

Cisco

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

Juniper

```
set routing-options static route 172.16.1.0/24 next-hop 172.16.2.2
```

IPv4 Static Route Command (Exit Interface)

Cisco

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

Juniper

```
set routing-options static route 172.16.1.0/24 next-hop s0/1/0
```

IPv4 Fully Specified Static Route

Cisco

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

Juniper

```
set routing-options static route 172.16.1.0/24 next-hop GigabitEthernet
0/0/1 next-hop 172.16.2.2
```

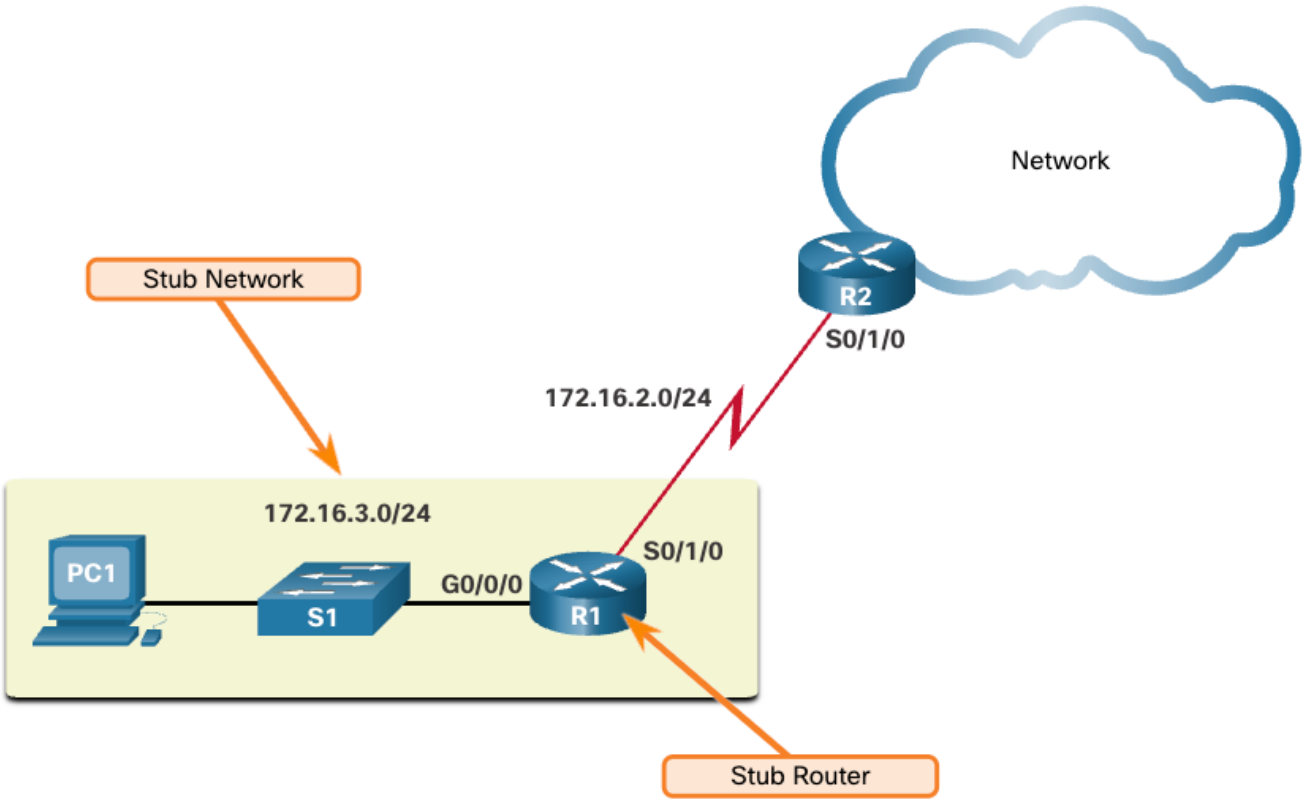
Default Static Route

This topic shows you how to configure a default route for IPv4 and IPv6. It also explains the situations in which a default route is a good choice. A default route is a static route that matches all packets. Instead of routers storing routes for all of the networks in the internet, they can store a single default route to represent any network that is not in the routing table.

Routers commonly use default routes that are either configured locally or learned from another router, using a dynamic routing protocol. A default route does not require any far-left bits to match between the default route and the destination IP address. A default route is used when no other routes in the routing table match the destination IP address of the packet. In other words, if a more specific match does not exist, then the default route is used as the Gateway of Last Resort.

Default static routes are commonly used when connecting an edge router to a service provider network, or a stub router (a router with only one upstream neighbor router).

The figure shows a typical default static route scenario.



Cisco

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

Juniper

```
set routing-options static route 0.0.0.0/0 next-hop 172.16.2.2
```

Floating Static Routes

Another type of static route is a floating static route. Floating static routes are static routes that are used to provide a backup path to a primary static or dynamic route, in the event of a link failure. The floating static route is only used when the primary route is not available.

To accomplish this, the floating static route is configured with a higher administrative distance than the primary route. The administrative distance represents the trustworthiness of a route. If multiple paths to the destination exist, the router will choose the path with the lowest administrative distance.

The administrative distance of a static route can be increased to make the route less desirable than that of another static route or a route learned through a dynamic routing protocol. In this way, the static route “floats” and is not used when the route with the better administrative distance is active. However, if the preferred route is lost, the floating static route can take over, and traffic can be sent through this alternate route.

Cisco

```
R1(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.2 50
```

Juniper

```
set routing-options static route 0.0.0.0/0 next-hop 10.10.10.2
preference 50
```

Chapter 9
OSPF

Introduction to OSPF

This topic is a brief overview of Open Shortest Path First (OSPF), which includes single-area and multiarea. OSPFv2 is used for IPv4 networks. OSPFv3 is used for IPv6 networks. The primary focus of this entire module is single-area OSPFv2.

OSPF is a link-state routing protocol that was developed as an alternative for the distance vector Routing Information Protocol (RIP). RIP was an acceptable routing protocol in the early days of networking and the internet. However, the RIP reliance on hop count as the only metric for determining best route quickly became problematic. Using hop count does not scale well in larger networks with multiple paths of varying speeds. OSPF has significant advantages over RIP in that it offers faster convergence and scales to much larger network implementations.

OSPF is a link-state routing protocol that uses the concept of areas. A network administrator can divide the routing domain into distinct areas that help control routing update traffic. A link is an interface on a router. A link is also a network segment that connects two routers, or a stub network such as an Ethernet LAN that is connected to a single router. Information about the state of a link is known as a link-state. All link-state information includes the network prefix, prefix length, and cost.

This module covers basic, single-area OSPF implementations and configurations.

Components of OSPF

All routing protocols share similar components. They all use routing protocol messages to exchange route information. The messages help build data structures, which are then processed using a routing algorithm.

Routing Protocol Messages

Routers running OSPF exchange messages to convey routing information using five types of packets. These packets, as shown in the figure, are as follows:

- Hello packet
- Database description packet
- Link-state request packet
- Link-state update packet
- Link-state acknowledgment packet

These packets are used to discover neighboring routers and also to exchange routing information to maintain accurate information about the network.

Data Structures

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

- **Adjacency database** - This creates the neighbor table.
- **Link-state database (LSDB)** - This creates the topology table.
- **Forwarding database** - This creates the routing table.

These tables contain a list of neighboring routers to exchange routing information. The tables are kept and maintained in RAM. In the following table, take a particular note of the command used to display each table.

Database	Table	Description
Adjacency Database	Neighbor Table	• List of all neighbor routers to which a router has established bidirectional communication. • This table is unique for each router. • Can be viewed using the **show ip ospf neighbor** command.
Link-state Database	Topology Table	• Lists information about all other routers in the network. • This database represents the network topology. • All routers within an area have identical LSDB. • Can be viewed using the **show ip ospf database** command.
Forwarding Database	Routing Table	• List of routes generated when an algorithm is run on the link-state database. • The routing table of each router is unique and contains information on how and where to send packets to other routers. • Can be viewed using the **show ip route** command.

Link-State Operation

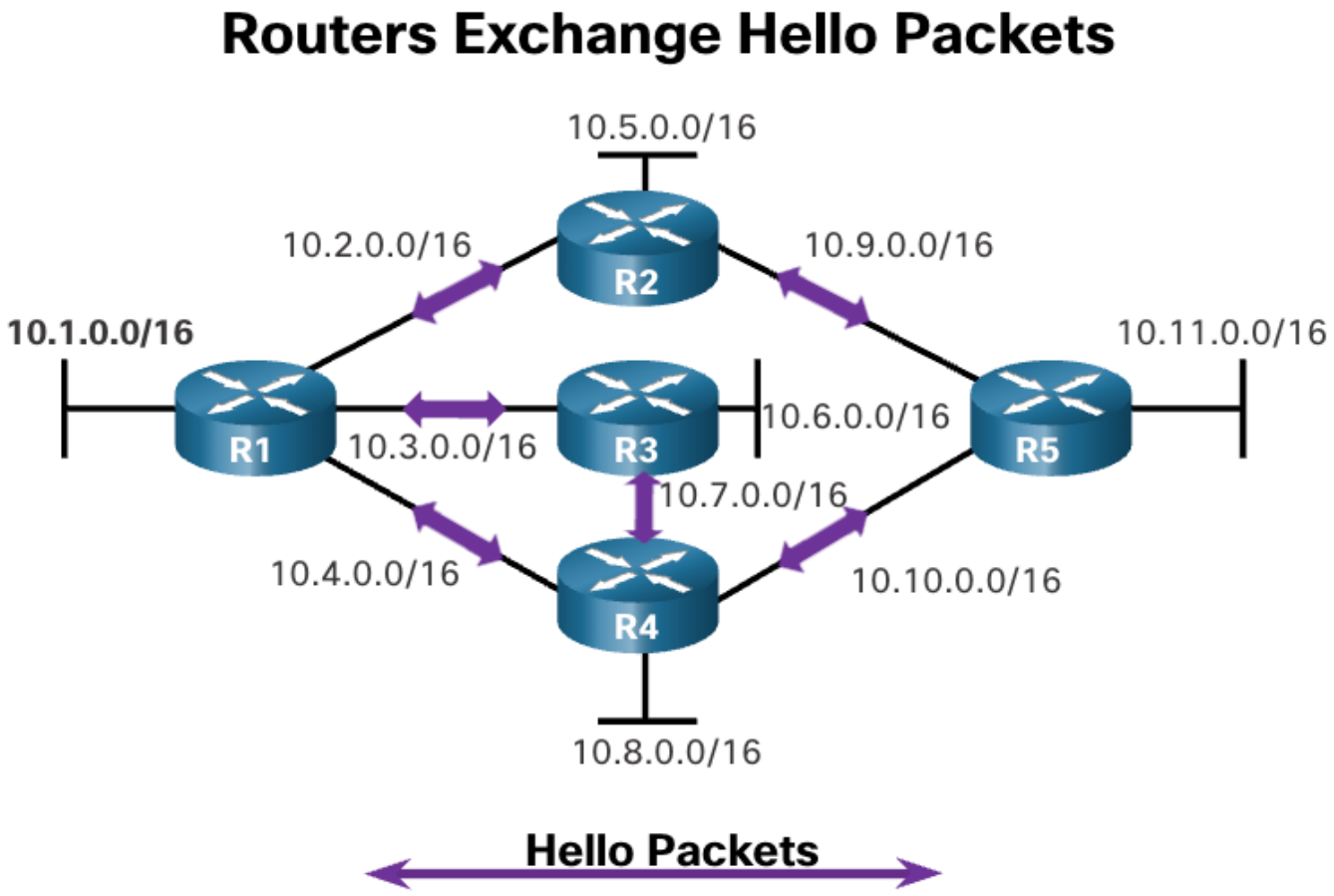
To maintain routing information, OSPF routers complete a generic link-state routing process to reach a state of convergence. The figure shows a five router topology. Each link between routers is labeled with a cost value. In OSPF, cost is used to determine the best path to the destination. The following are the link-state routing steps that are completed by a router:

1. Establish Neighbor Adjacencies
2. Exchange Link-State Advertisements
3. Build the Link State Database

4. Execute the SPF Algorithm
5. Choose the Best Route

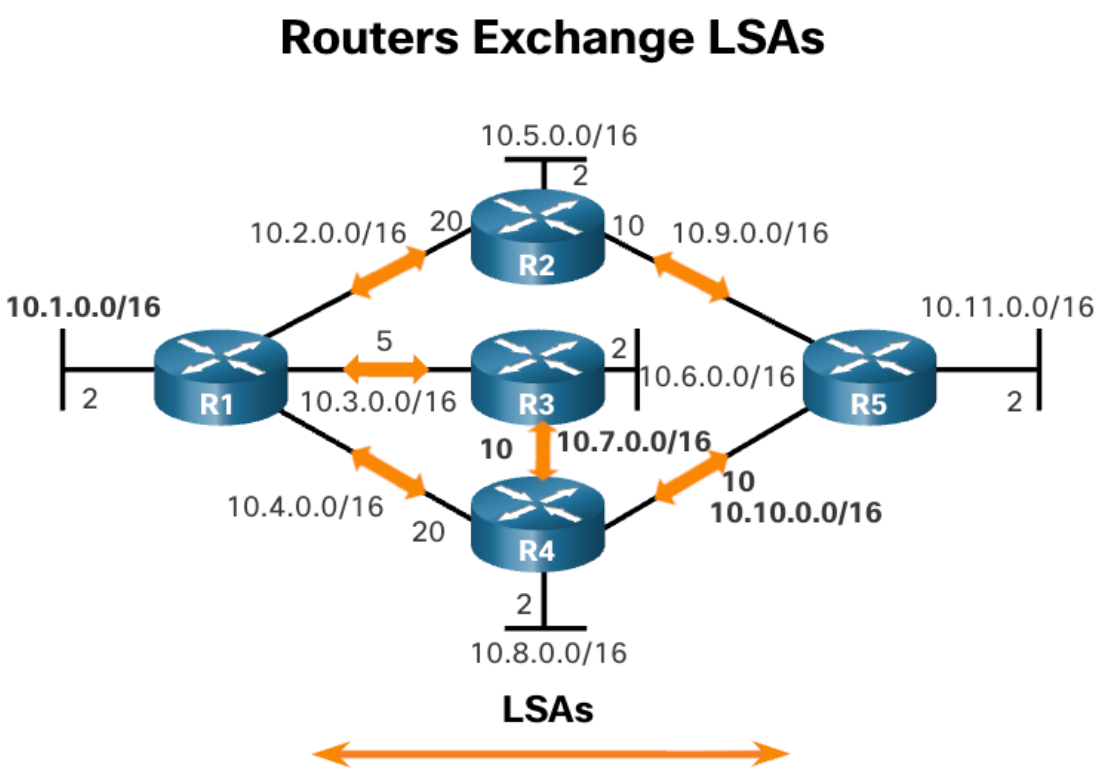
1. Establish Neighbor Adjacencies

OSPF-enabled routers must recognize each other on the network before they can share information. An OSPF-enabled router sends Hello packets out all OSPF-enabled interfaces to determine if neighbors are present on those links. If a neighbor is present, the OSPF-enabled router attempts to establish a neighbor adjacency with that neighbor.



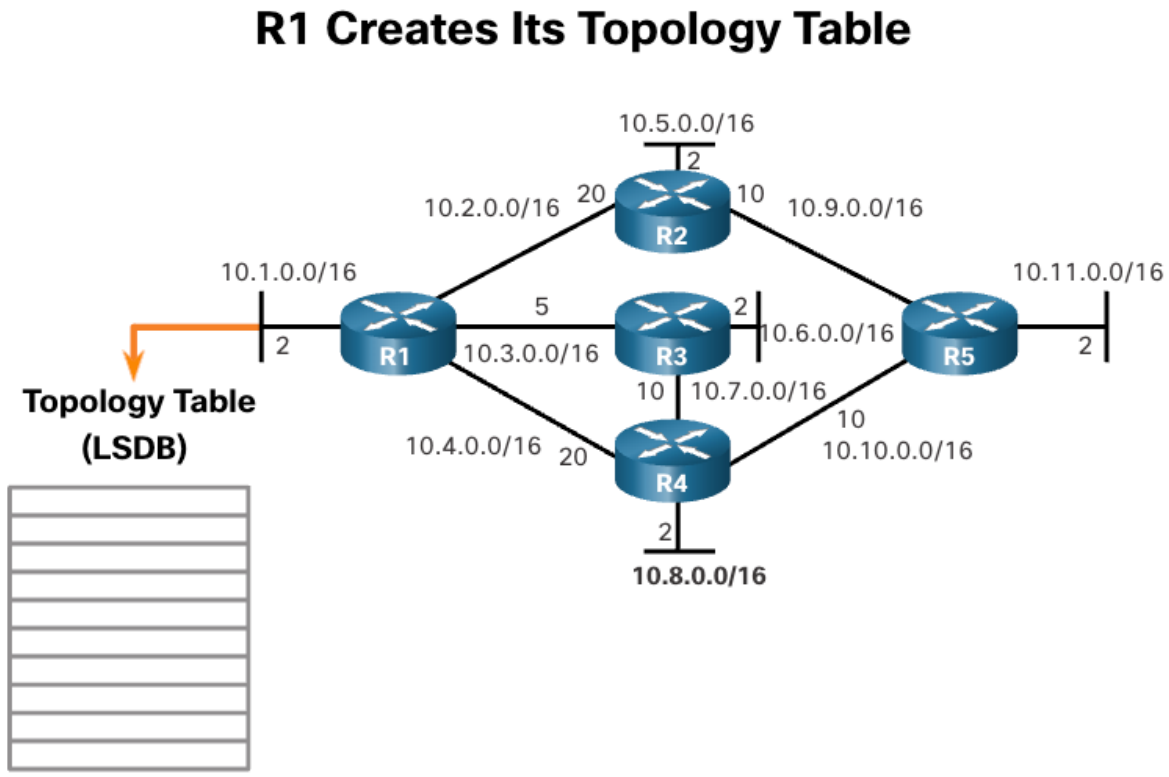
2. Exchange Link-State Advertisements

After adjacencies are established, routers then exchange link-state advertisements (LSAs). LSAs contain the state and cost of each directly connected link. Routers flood their LSAs to adjacent neighbors. Adjacent neighbors receiving the LSA immediately flood the LSA to other directly connected neighbors, until all routers in the area have all LSAs.



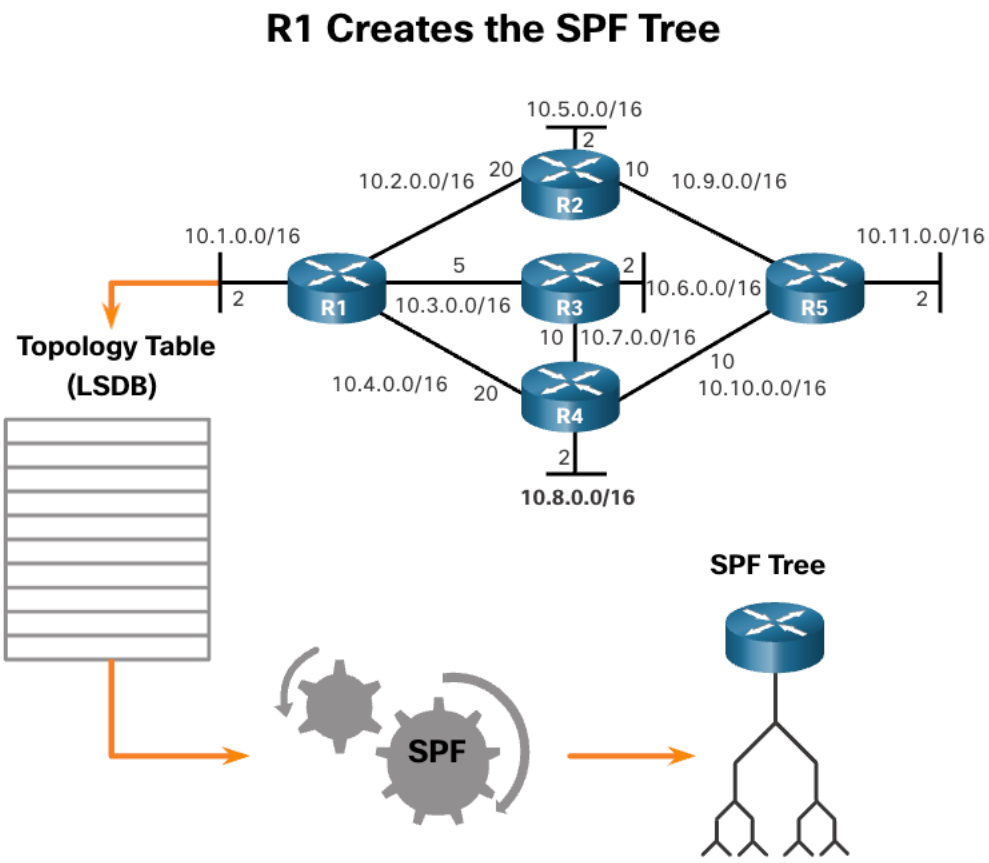
3. Build the Link State Database

After LSAs are received, OSPF-enabled routers build the topology table (LSDB) based on the received LSAs. This database eventually holds all the information about the topology of the area.



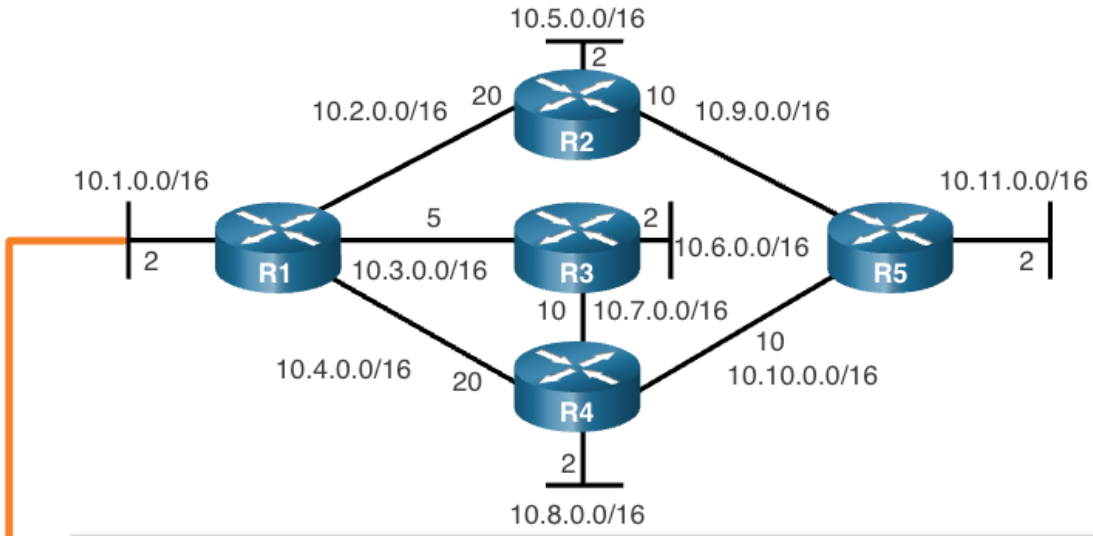
4. Execute the SPF Algorithm

Routers then execute the SPF algorithm. The gears in the figure for this step are used to indicate the execution of the SPF algorithm. The SPF algorithm creates the SPF tree.



5. Choose the Best Route

After the SPF tree is built, the best paths to each network are offered to the IP routing table. The route will be inserted into the routing table unless there is a route source to the same network with a lower administrative distance, such as a static route. Routing decisions are made based on the entries in the routing table.



Destination	Shortest Path	Cost
10.5.0.0/16	R1→R2	22
10.6.0.0/16	R1→R3	7
10.7.0.0/16	R1→R3	15
10.8.0.0/16	R1→R3→R4	17
10.9.0.0/16	R1→R2	30
10.10.0.0/16	R1→R3→R4	25
10.11.0.0/16	R1→R3→R4→R5	27

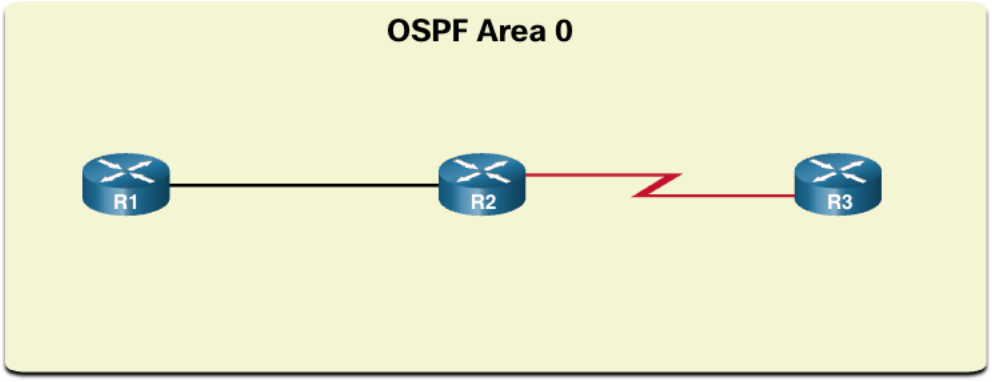
Single-Area and Multiarea OSPF

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

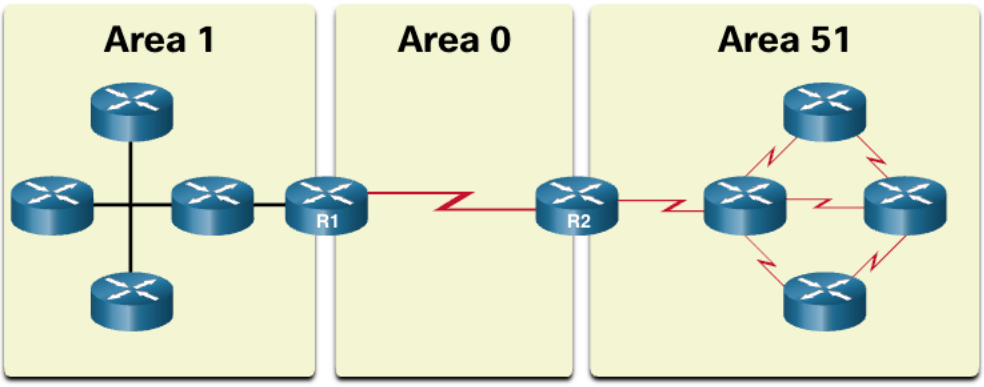
- **Single-Area OSPF** - All routers are in one area. Best practice is to use area 0.
- **Multiarea OSPF** - OSPF is implemented using multiple areas, in a hierarchical fashion. All areas must connect to the backbone area (area 0). Routers interconnecting the areas are referred to as Area Border Routers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

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

Single-Area OSPF



Multiarea OSPF



Multiarea OSPF

With multiarea OSPF, one large routing domain can be divided into smaller areas, to support hierarchical routing. Routing still occurs between the areas (interarea routing), while many of the processor intensive routing operations, such as recalculating the database, are kept within an area.

For instance, any time a router receives new information about a topology change within the area (including the addition, deletion, or modification of a link) the router must rerun the SPF algorithm, create a new SPF tree, and update the routing table. The SPF algorithm is CPU-intensive and the time it takes for calculation depends on the size of the area.

Note: Routers in other areas receive updates regarding topology changes, but these routers only update the routing table, not rerun the SPF algorithm.

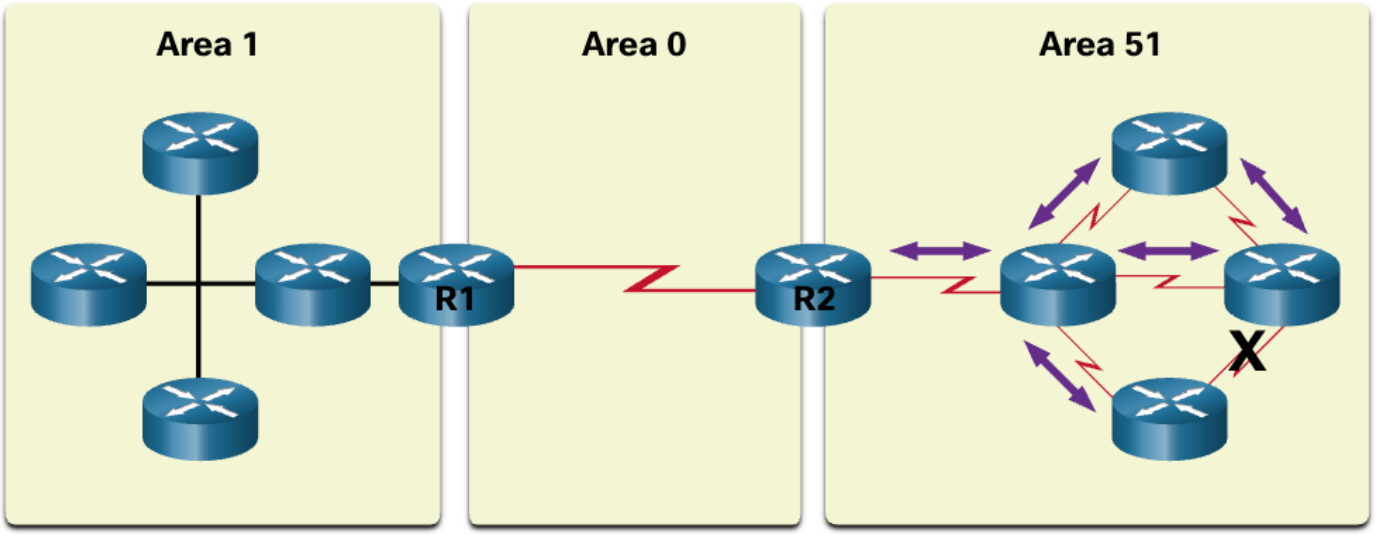
Too many routers in one area would make the LSDBs very large and increase the load on the CPU. Therefore, arranging routers into areas effectively partitions a potentially large database into smaller and more manageable databases.

The hierarchical-topology design options with multiarea OSPF can offer the following advantages.

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

For example, in the figure R2 is an ABR for area 51. A topology change in area 51 would cause all area 51 routers to rerun the SPF algorithm, create a new SPF tree and update their IP routing tables. The ABR, R2, would send an LSA to routers in the area 0, which would eventually be flooded to all routers in the OSPF routing domain. This type of LSA does not cause routers in other areas to rerun the SPF algorithm. They only have to update their LSDB and routing table.

Link Change Impacts Local Area Only



- *Link failure affects the local area only (area 51).*
- *The ABR (R2) isolates the flooding of a specific LSA to area 51.*
- *Routers in areas 0 and 1 do not need to run the SPF algorithm.*

Types of OSPF Packets

Link-state packets are the tools used by OSPF to help determine the fastest available route for a packet. OSPF uses the following link-state packets (LSPs) to establish and maintain neighbor adjacencies and exchange routing updates. Each packet serves a specific purpose in the OSPF routing process, as follows:

- **Type 1: Hello packet** - This is used to establish and maintain adjacency with other OSPF routers.
- **Type 2: Database Description (DBD) packet** - This contains an abbreviated list of the LSDB of the sending router and is used by receiving routers to check against the local LSDB. The LSDB must be identical on all link-state routers within an area to construct an accurate SPF tree.
- **Type 3: Link-State Request (LSR) packet** - Receiving routers can then request more information about any entry in the DBD by sending an LSR.
- **Type 4: Link-State Update (LSU) packet** - This is used to reply to LSRs and to announce new information. LSUs contain several different types of LSAs.
- **Type 5: Link-State Acknowledgment (LSAck) packet** - When an LSU is received, the router sends an LSAck to confirm receipt of the LSU. The LSAck data field is empty.

The table summarizes the five different types of LSPs used by OSPFv2. OSPFv3 has similar packet types.

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

Link-State Updates

Routers initially exchange Type 2 DBD packets, which is an abbreviated list of the LSDB of the sending router. It is used by receiving routers to check against the local LSDB.

A Type 3 LSR packet is used by the receiving routers to request more information about an entry in the DBD.

The Type 4 LSU packet is used to reply to an LSR packet.

A Type 5 packet is used to acknowledge the receipt of a Type 4 LSU.

LSUs are also used to forward OSPF routing updates, such as link changes. Specifically, an LSU packet can contain 11 different types of OSPFv2 LSAs, with some of the more common ones shown in the figure. OSPFv3 renamed several of these LSAs and also contains two additional LSAs.

Note: The difference between the LSU and LSA terms can sometimes be confusing because these terms are often used interchangeably. However, an LSU contains one or more LSAs.

LSUs Contain LSAs

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

LSAs	
LSA Type	**Description**
1	Router LSAs
2	Network LSAs
3 or 4	Summary LSAs
5	Autonomous System External LSAs
6	Multicast OSPF LSAs
7	Defined for Not-So-Stubby Areas
8	External Attributes LSA for Border Gateway Patrol (BGPs)

- *An LSU contains one or more LSAs.*
- *LSAs contain route information for destination networks.*

Hello Packet

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

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

OSPF Operational States

Now that you know about the OSPF link-state packets, this topic explains how they work with OSPF-enabled routers. When an OSPF router is initially connected to a network, it attempts to:

- Create adjacencies with neighbors
- Exchange routing information
- Calculate the best routes
- Reach convergence

The table details the states OSPF progresses through while attempting to reach convergence:

State	Description
Down State	• No Hello packets received = Down. • Router sends Hello packets. • Transition to Init state.
Init State	• Hello packets are received from the neighbor. • They contain the Router ID of the sending router. • Transition to Two-Way state.
Two-Way State	• In this state, communication between the two routers is bidirectional. • On multiaccess links, the routers elect a DR and a BDR. • Transition to ExStart state.
ExStart State	On point-to-point networks, the two routers decide which router will initiate the DBD packet exchange and decide upon the initial DBD packet sequence number.
Exchange State	• Routers exchange DBD packets. • If additional router information is required then transition to Loading; otherwise, transition to the Full state.
Loading State	• LSRs and LSUs are used to gain additional route information. • Routes are processed using the SPF algorithm. • Transition to the Full state.
Full State	The link-state database of the router is fully synchronized.

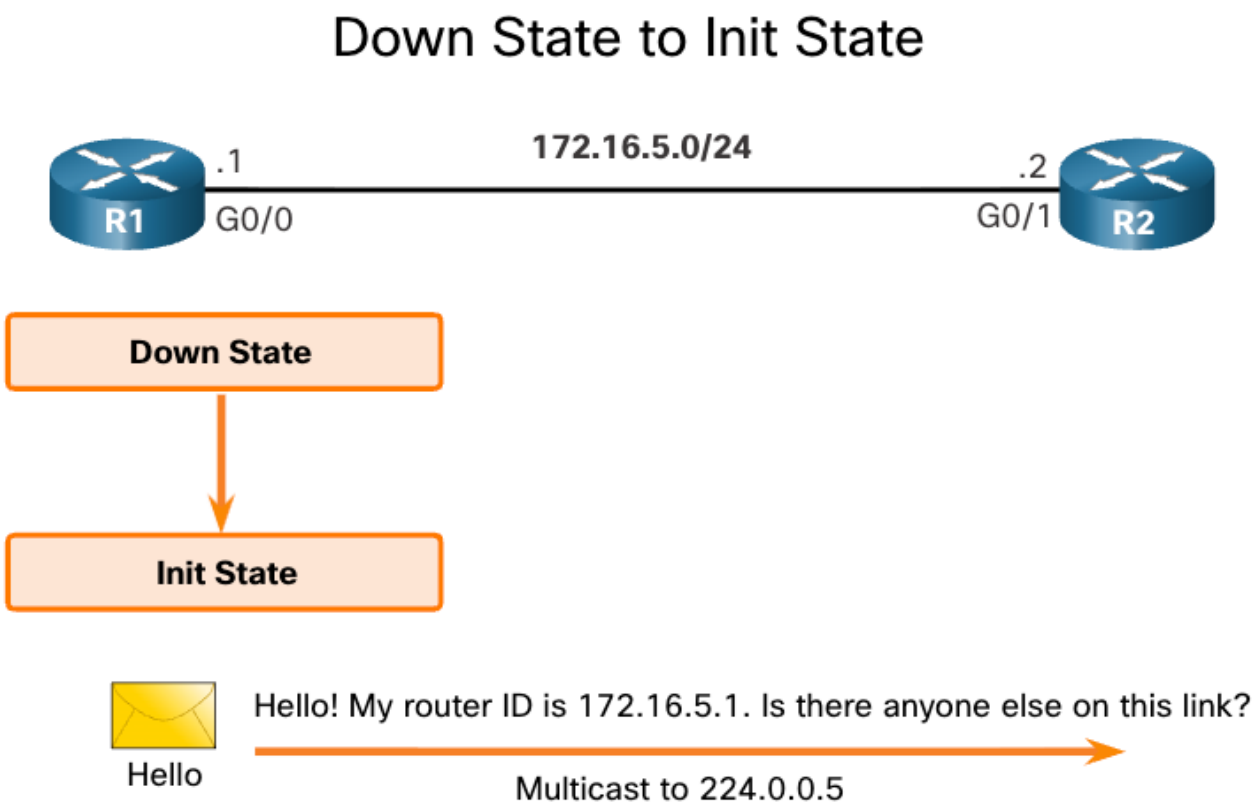
Establish Neighbor Adjacencies

When OSPF is enabled on an interface, the router must determine if there is another OSPF neighbor on the link. To accomplish this, the router sends a Hello packet that contains its router ID out all OSPF-enabled interfaces. The Hello packet is sent to the reserved All OSPF Routers IPv4 multicast address 224.0.0.5. Only OSPFv2 routers will process these packets. The OSPF router ID is used by the OSPF process to uniquely identify each router in the OSPF area. A router ID is a 32-bit number formatted like an IPv4 address and assigned to uniquely identify a router among OSPF peers.

When a neighboring OSPF-enabled router receives a Hello packet with a router ID that is not within its neighbor list, the receiving router attempts to establish an adjacency with the initiating router.

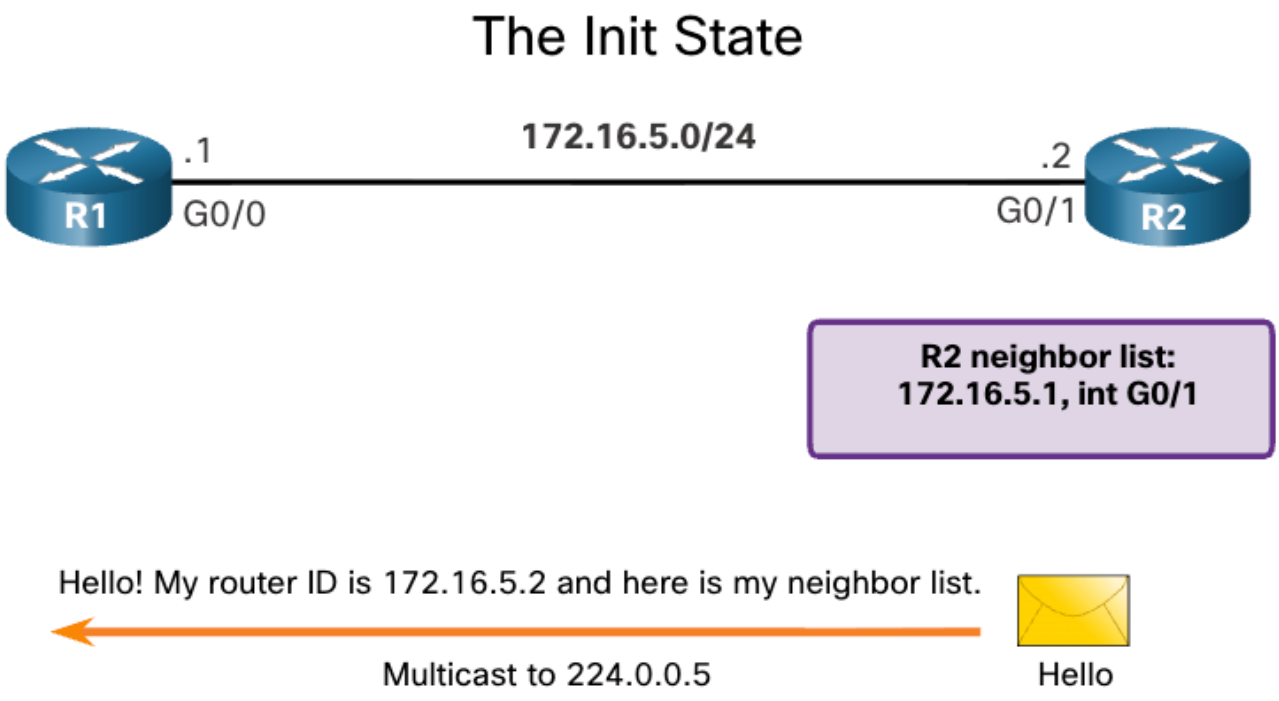
Down State to Init State

When OSPFv2 is enabled, the enabled Gigabit Ethernet 0/0 interface transitions from the Down state to the Init state. R1 starts sending Hello packets out all OSPF-enabled interfaces to discover OSPF neighbors to develop adjacencies with.



The Init State

R2 receives the Hello packet from R1 and adds the R1 router ID to its neighbor list. R2 then sends a Hello packet to R1. The packet contains the R2 Router ID and the R1 Router ID in its list of neighbors on the same interface.

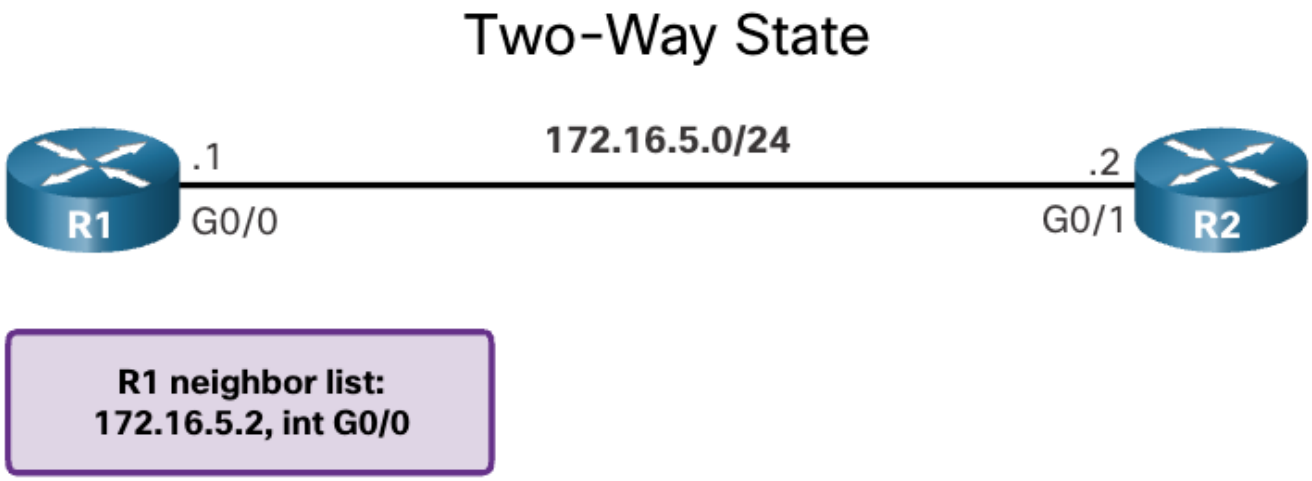


Two-Way State

R1 receives the Hello and adds the R2 Router ID to its list of OSPF neighbors. It also notices its own Router ID in the list of neighbors of the Hello packet. When a router receives a Hello packet with its Router ID listed in the list of neighbors, the router transitions from the Init state to the Two-Way state.

The action performed in Two-Way state depends on the type of interconnection between the adjacent routers, as follows:

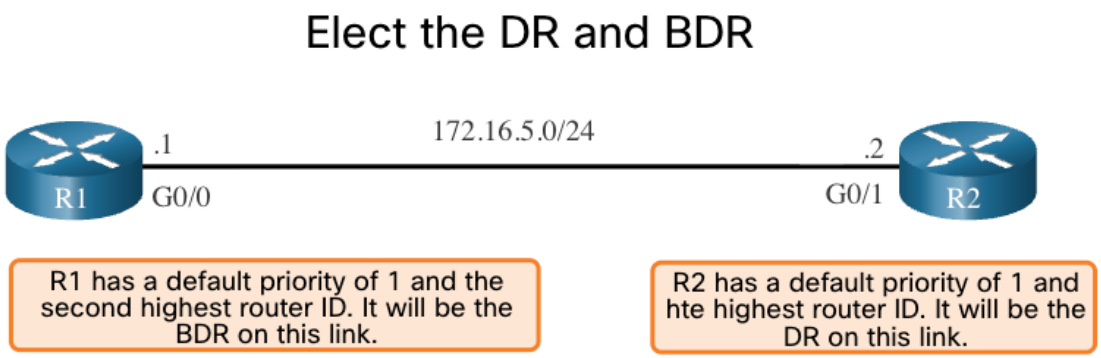
- If the two adjacent neighbors are interconnected over a point-to-point link, then they immediately transition from the Two-Way state to the ExStart state.
- If the routers are interconnected over a common Ethernet network, then a designated router DR and a BDR must be elected.



Elect the DR and BDR

Because R1 and R2 are interconnected over an Ethernet network, a DR and BDR election takes place. As shown in the figure, R2 becomes the DR and R1 is the BDR. This process only occurs on multiaccess networks such as Ethernet LANs.

Hello packets are continually exchanged to maintain router information.



Synchronizing OSPF Databases

After the Two-Way state, routers transition to database synchronization states. While the Hello packet was used to establish neighbor adjacencies, the other four types of OSPF packets are used during the process of exchanging and synchronizing LSDBs. This is a three step process, as follows:

1. Decide first router
2. Exchange DBDs
3. Send an LSR

1. Decide First Router

In the ExStart state, the two routers decide which router will send the DBD packets first. The router with the higher router ID will be the first router to send DBD packets during the Exchange state. In the figure, R2 has the higher router ID and sends its DBD packets first.

2. Exchange DBDs

In the Exchange state, the two routers exchange one or more DBD packets. A DBD packet includes information about the LSA entry header that appears in the LSDB of the router. The entries can be about a link or about a network. Each LSA entry header includes information about the link-state type, the address of the advertising router, the cost of the link, and the sequence number. The router uses the sequence number to determine the newness of the received link-state information.

In the figure, R2 sends a DBD packet to R1. When R1 receives the DBD, it performs the following actions:

1. It acknowledges the receipt of the DBD using the LSAck packet.
2. R1 then sends DBD packets to R2.
3. R2 acknowledges R1.

3. Send an LSR

R1 compares the information received with the information it has in its own LSDB. If the DBD packet has a more current link-state entry, the router transitions to the Loading state.

For example, in the figure, R1 sends an LSR regarding network 172.16.6.0 to R2. R2 responds with the complete information about 172.16.6.0 in an LSU packet. Again, when R1 receives an LSU, it sends an LSAck. R1 then adds the new link-state entries into its LSDB.

After all LSRs have been satisfied for a given router, the adjacent routers are considered synchronized and in a full state. Updates (LSUs) are sent only to neighbors in the following conditions:

- When a change is perceived (incremental updates)
- Every 30 minutes

The Need for a DR

Why is a DR and BDR election necessary?

Multiaccess networks can create two challenges for OSPF regarding the flooding of LSAs, as follows:

- **Creation of multiple adjacencies** - Ethernet networks could potentially interconnect many OSPF routers over a common link. Creating adjacencies with every router is unnecessary and undesirable. It would lead to an excessive number of LSAs exchanged between routers on the same network.
- **Extensive flooding of LSAs** - Link-state routers flood their LSAs any time OSPF is initialized, or when there is a change in the topology. This flooding can become excessive.

To understand the problem with multiple adjacencies, we must study a formula:

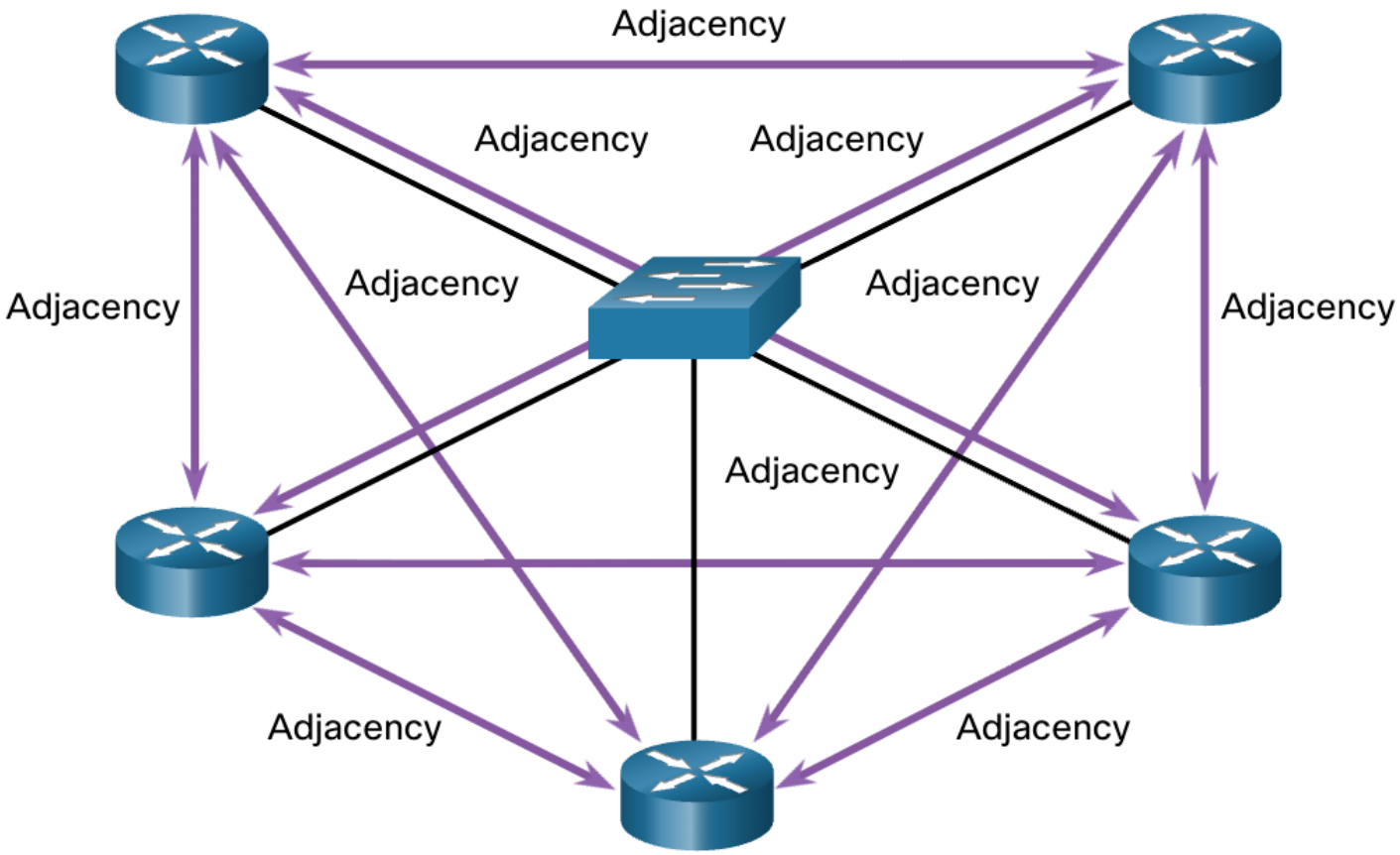
For any number of routers (designated as *n*) on a multiaccess network, there are *n (n - 1) / 2* adjacencies.

For example, the figure shows a simple topology of five routers, all of which are attached to the same multiaccess Ethernet network. Without some type of mechanism to reduce the number of adjacencies, collectively these routers would form 10 adjacencies:

5 (5 - 1) / 2 = 10

This may not seem like much, but as routers are added to the network, the number of adjacencies increases dramatically. For example, a multiaccess network with 20 routers would create 190 adjacencies.

Creating Adjacencies With Every Neighbor



- *Number of Adjacencies = n (n - 1) / 2*
- *n = number of routers*
- *Example: 5 (5 - 1) / 2 = 10 adjacencies*

LSA Flooding With a DR

A dramatic increase in the number of routers also dramatically increases the number of LSAs exchanged between the routers. This flooding of LSAs significantly impacts the operation of OSPF.

Flooding LSAs

To understand the problem of extensive flooding of LSAs, play the animation in the figure. In the animation, R2 sends out an LSA. This event triggers every other router to also send out an LSA. Not shown in the animation are the required acknowledgments sent for every LSA received. If every router in a multiaccess network had to flood and acknowledge all received LSAs to all other routers on that same multiaccess network, the network traffic would become quite chaotic.

LSAs and DR

The solution to managing the number of adjacencies and the flooding of LSAs on a multiaccess network is the DR. On multiaccess networks, OSPF elects a DR to be the collection and distribution point for LSAs sent and received. A BDR is also elected in case the DR fails. All other routers become DROTHERs. A DROTHER is a router that is neither the DR nor the BDR.

Note: The DR is only used for the dissemination of LSAs. The router will still use the best next-hop router indicated in the routing table for the forwarding of all other packets.

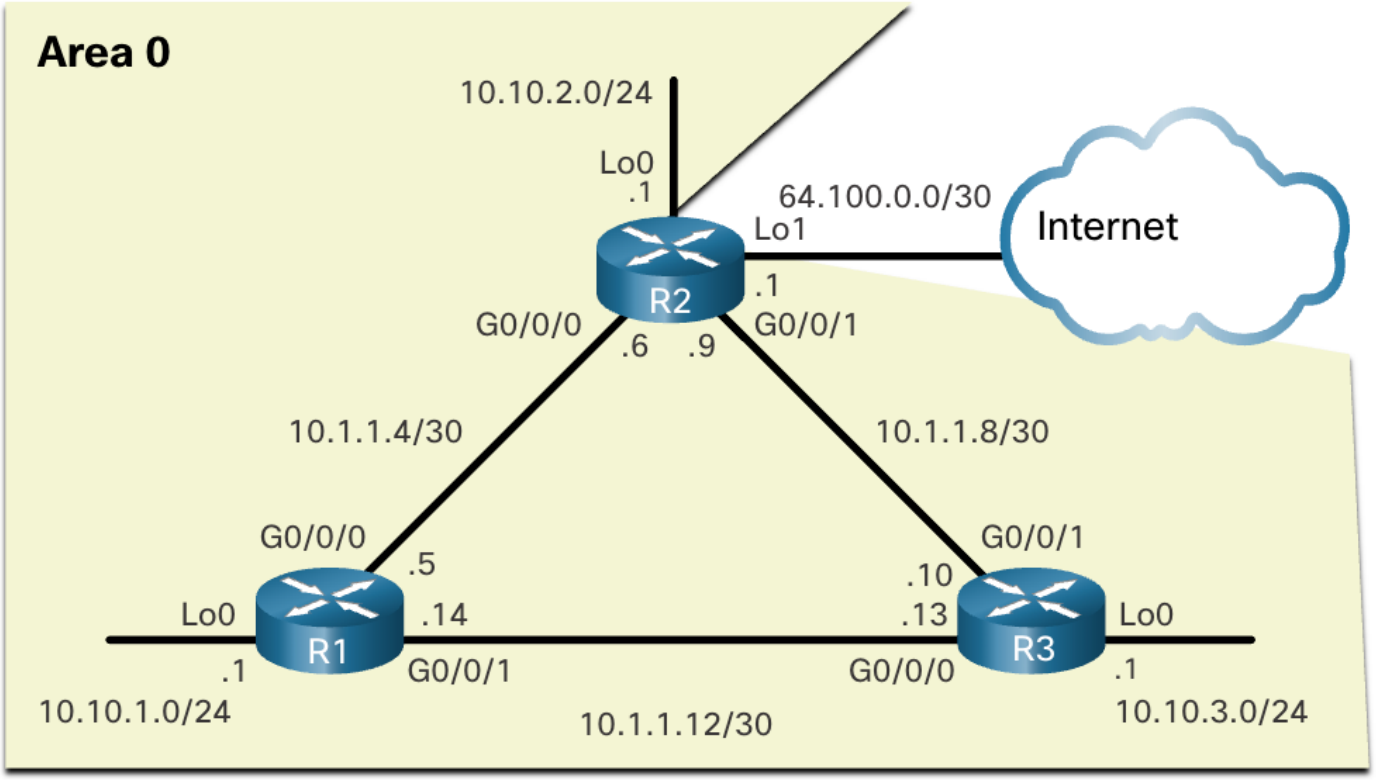
OSPF Configuration

OSPF Reference Topology

To get you started, this topic discusses the foundation on which OSPF bases its entire process, the OSPF router ID.

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

Note: In this topology the loopback interface is used to simulate the WAN link to the Internet and a LAN connected to each router. This is done to allow this topology to be duplicated for demonstration purposes on routers that only have two Gigabit Ethernet interfaces.



To configure Ospf for Cisco you should enable OSPF process from global configuration mode

```
R1(config)# router ospf 10
```

For Juniper:

In **Juniper Junos OS**, you **don't "enable the OSPF process" with a single command** like in **Cisco IOS**

```
R1(config)# router ospf 10
```

OSPF becomes active when you **configure interfaces inside an OSPF area**

set protocols ospf area 0.0.0.0 interface ge-0/0/0

Router IDs

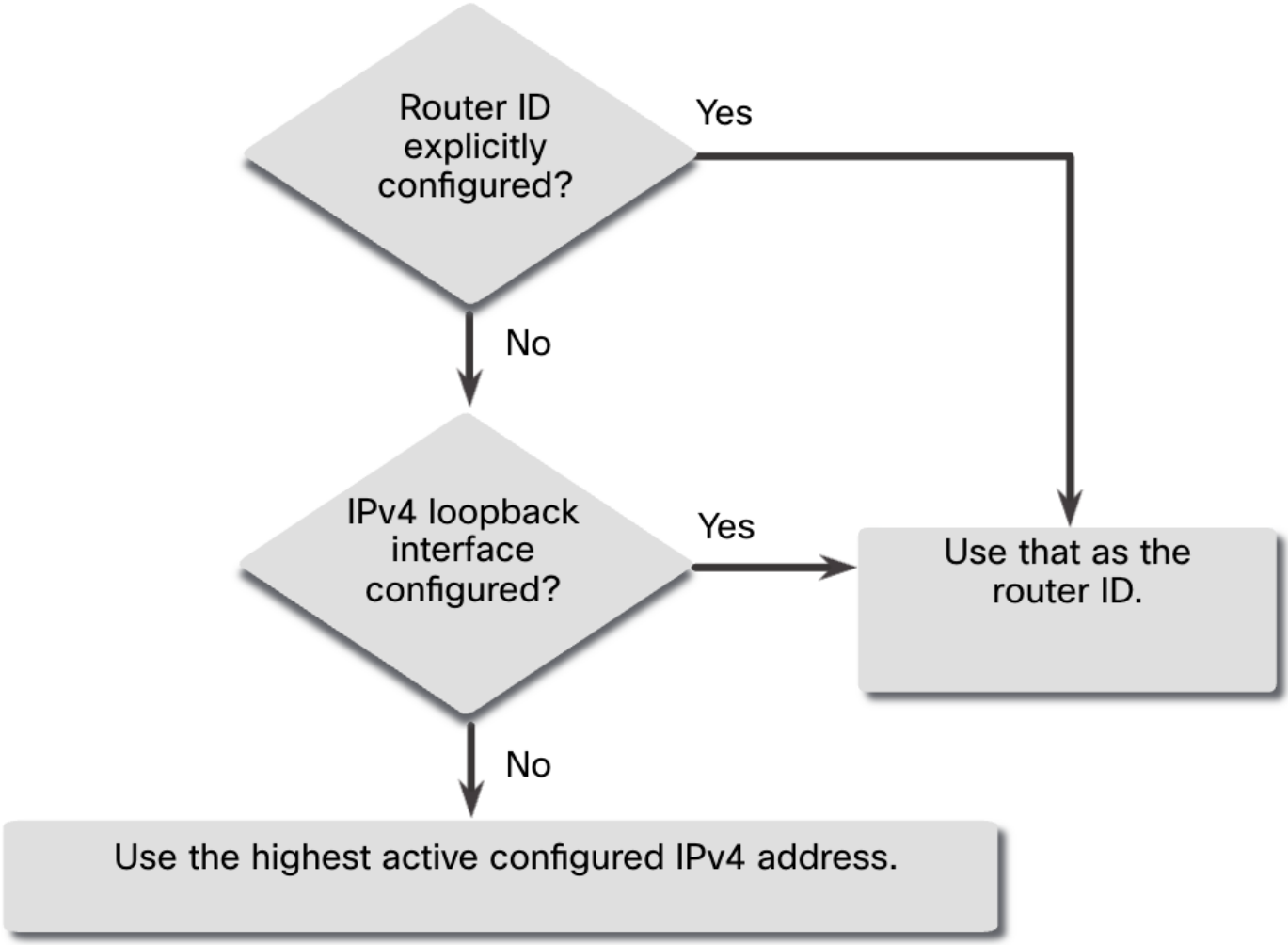
An OSPF router ID is a 32-bit value, represented as an IPv4 address. The router ID is used to uniquely identify an OSPF router. All OSPF packets include the router ID of the originating router. Every router requires a router ID to participate in an OSPF domain. The router ID can be defined by an administrator or automatically assigned by the router. The router ID is used by an OSPF-enabled router to do the following:

- **Participate in the synchronization of OSPF databases** - During the Exchange State, the router with the highest router ID will send their database descriptor (DBD) packets first.
- **Participate in the election of the designated router (DR)** - In a multiaccess LAN environment, the router with the highest router ID is elected the DR. The routing device with the second highest router ID is elected the backup designated router (BDR).

Router ID Order of Precedence

But how does the router determine the router ID? As illustrated in the figure, routers derive the router ID based on one of three criteria, in the following preferential order:

1. The router ID is explicitly. The *rid* value is any 32-bit value expressed as an IPv4 address. This is the recommended method to assign a router ID.
2. If the router ID is not explicitly configured, the router chooses the highest IPv4 address of any of configured loopback interfaces. This is the next best alternative to assigning a router ID.
3. If no loopback interfaces are configured, then the router chooses the highest active IPv4 address of any of its physical interfaces. This is the least recommended method because it makes it more difficult for administrators to distinguish between specific routers.



Explicitly Configure a Router ID

For Cisco:

```
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# end
*May 23 19:33:42.689: %SYS-5-CONFIG_I: Configured from console by
console
R1# show ip protocols | include Router ID
Router ID 1.1.1.1
R1#
```

For Juniper:

set routing-options router-id 1.1.1.1

Modify a Router ID

For Cisco:

```
R1# show ip protocols | include Router ID
Router ID 10.10.1.1
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
% OSPF: Reload or use "clear ip ospf process" command, for this to take
effect
R1(config-router)# end
R1# clear ip ospf process
Reset ALL OSPF processes? [no]: y
*Jun  6 01:09:46.975: %OSPF-5-ADJCHG: Process 10, Nbr 3.3.3.3 on
GigabitEthernet0/0/1 from FULL to DOWN, Neighbor Down: Interface down or
detached
*Jun  6 01:09:46.975: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2 on
GigabitEthernet0/0/0 from FULL to DOWN, Neighbor Down: Interface down or
detached
*Jun  6 01:09:46.981: %OSPF-5-ADJCHG: Process 10, Nbr 3.3.3.3 on
GigabitEthernet0/0/1 from LOADING to FULL, Loading Done
*Jun  6 01:09:46.981: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2 on
GigabitEthernet0/0/0 from LOADING to FULL, Loading Done
R1# show ip protocols | include Router ID
Router ID 1.1.1.1
R1#
```

For Juniper:

user@host> restart routing

Example of configuring OSPF

For Cisco:

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

For Juniper:

set protocols ospf area 0.0.0.0 interface ge-0/0/5
set protocols ospf area 0.0.0.0 interface ge-0/0/6
set protocols ospf area 0.0.0.0 interface ge-0/0/7

Passive Interface

By default, OSPF messages are forwarded out all OSPF-enabled interfaces. However, these messages really only need to be sent out interfaces that are connecting to other OSPF-enabled routers.

Refer to the topology in the figure. OSPFv2 messages are forwarded out the three loopback interfaces even though no OSPFv2 neighbor exists on these simulated LANs. In a production network, these loopbacks would be physical interfaces to networks with users and traffic. Sending out unneeded messages on a LAN affects the network in three ways, as follows:

- **Inefficient Use of Bandwidth** - Available bandwidth is consumed transporting unnecessary messages.
- **Inefficient Use of Resources** - All devices on the LAN must process and eventually discard the message.
- **Increased Security Risk** - Without additional OSPF security configurations, OSPF messages can be intercepted with packet sniffing software. Routing updates can be modified and sent back to the router, corrupting the routing table with false metrics that misdirect traffic.

Configure Passive interface

For Cisco:

```
R1(config)# router ospf 10
R1(config-router)# passive-interface loopback 0
```

For Juniper:

set protocols ospf area 0.0.0.0 interface ge-0/0/5 passive

OSPF Cost Metric

Routing protocol uses a metric to determine the best path of a packet across a network. A metric gives indication of the overhead that is required to send packets across a certain interface. OSPF uses cost as a metric. A lower cost indicates a better path than a higher cost.

The cost of an interface is inversely proportional to the bandwidth of the interface. Therefore, a higher bandwidth indicates a lower cost. The formula used to calculate the OSPF cost is:

Cost = reference bandwidth / interface bandwidth

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

Cost = 100,000,000 bps / interface bandwidth in bps

Refer to the table for a breakdown of the cost calculation. Because the OSPF cost value must be an integer, FastEthernet, Gigabit Ethernet, and 10 Gigabit Ethernet (10 GigE) interfaces share the same cost. To correct this situation, you can Adjust the reference bandwidth

For Cisco:

```
R1(config-router)# auto-cost reference-bandwidth 10000
```

For Juniper:

set protocols ospf reference-bandwidth 10g

Propagate a Default Static Route in OSPFv2

Your network users will need to send packets out of your network to non-OSPF networks, such as the internet. This is where you will need to have a default static route that they can use. In the topology in the figure, R2 is connected to the internet and should propagate a default route to R1 and R3. The router connected to the internet is sometimes called the edge router or the gateway router. However, in OSPF terminology, the router located between an OSPF routing domain and a non-OSPF network is called the autonomous system boundary router (ASBR).

To propagate defaulf route, first you should configure default static route both in Cisco and Juniper.

Here is the example of both Cisco and Juniper how to propagate default route:

For Cisco:

```
R2(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.1
R2(config)# router ospf 10
R2(config-router)# default-information originate
```

For Juniper:

set routing-options static route 0.0.0.0/0 next-hop 192.168.1.1
set policy-options policy-statement EXPORT-DEFAULT term 1 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement EXPORT-DEFAULT term 1 then accept
set protocols ospf export EXPORT-DEFAULT

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

Chapter 10
NAT

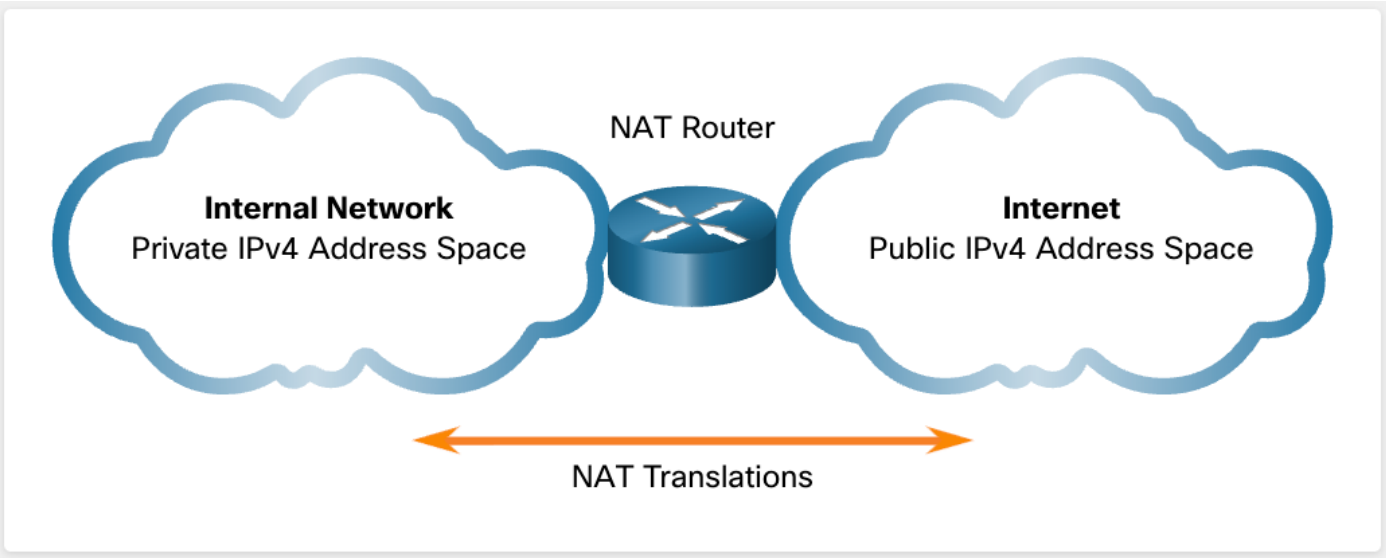
IPv4 Private Address Space

As you know, there are not enough public IPv4 addresses to assign a unique address to each device connected to the internet. Networks are commonly implemented using private IPv4 addresses, as defined in RFC 1918. The range of addresses included in RFC 1918 are included in the following table. It is very likely that the computer that you use to view this course is assigned a private address.

Class	RFC 1918 Internal Address Range	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, as shown in the figure. 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 been the primary 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 limitations and disadvantages, which will be explored later in this

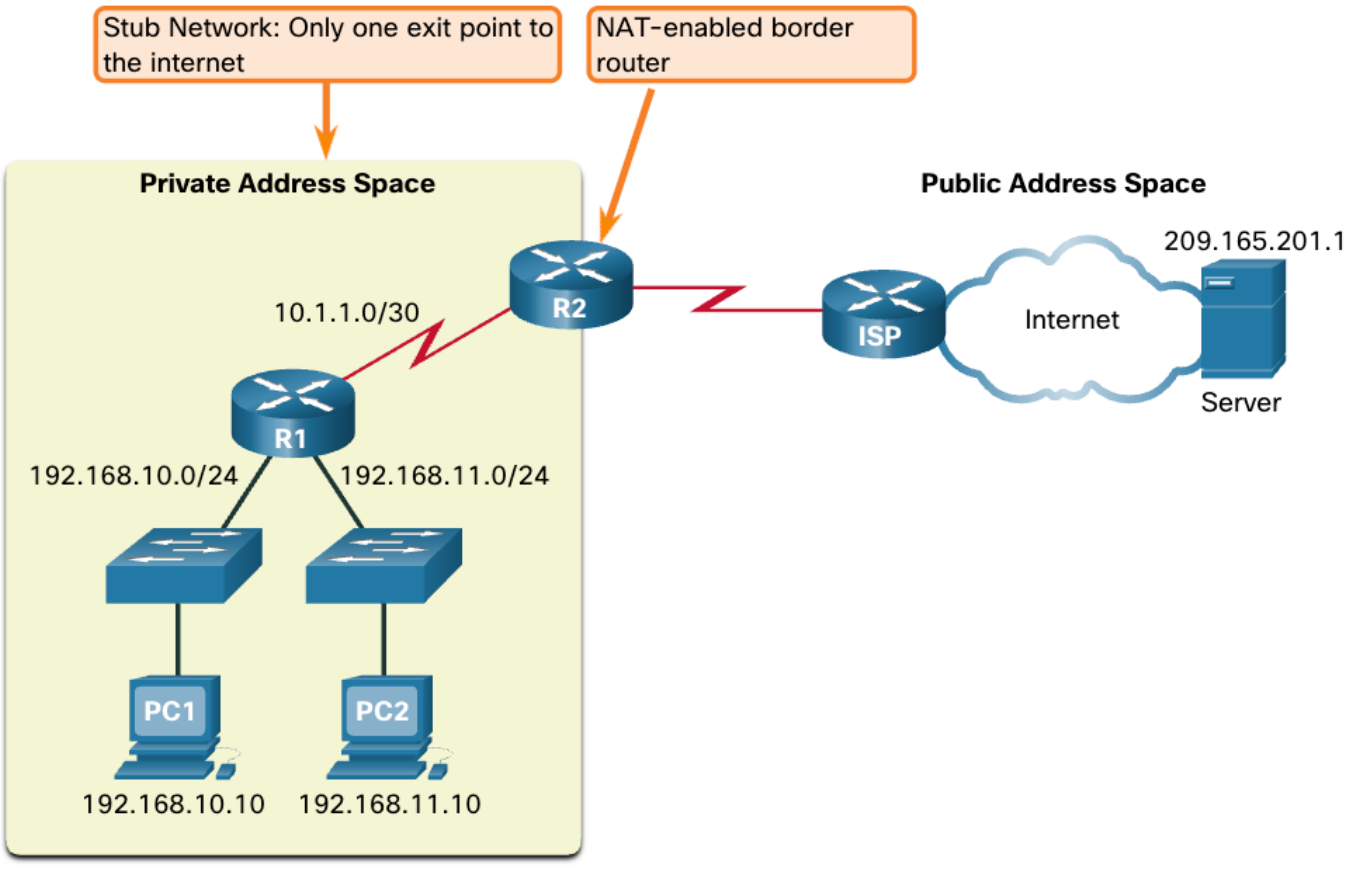
module. The solution to the exhaustion of IPv4 address space and the limitations of NAT is the eventual transition to IPv6.

What is NAT

NAT has many uses, but its primary use is to conserve public IPv4 addresses. It does this by allowing networks to use private IPv4 addresses internally and providing translation to a public address only when needed. NAT has a perceived 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 one or more networks with a single connection to its neighboring network, one way in and one way out of the network. In the example in the figure, R2 is a border router. As seen from the ISP, R2 forms a stub network.



When a device inside the stub network wants to communicate with a device outside of its network, the packet is forwarded to the border router. The border router performs the NAT process, translating the internal private address of the device to a public, outside, routable address.

Note: The connection to the ISP may use a private address or a public address that is shared among customers. For the purposes of this module, a public address is shown.

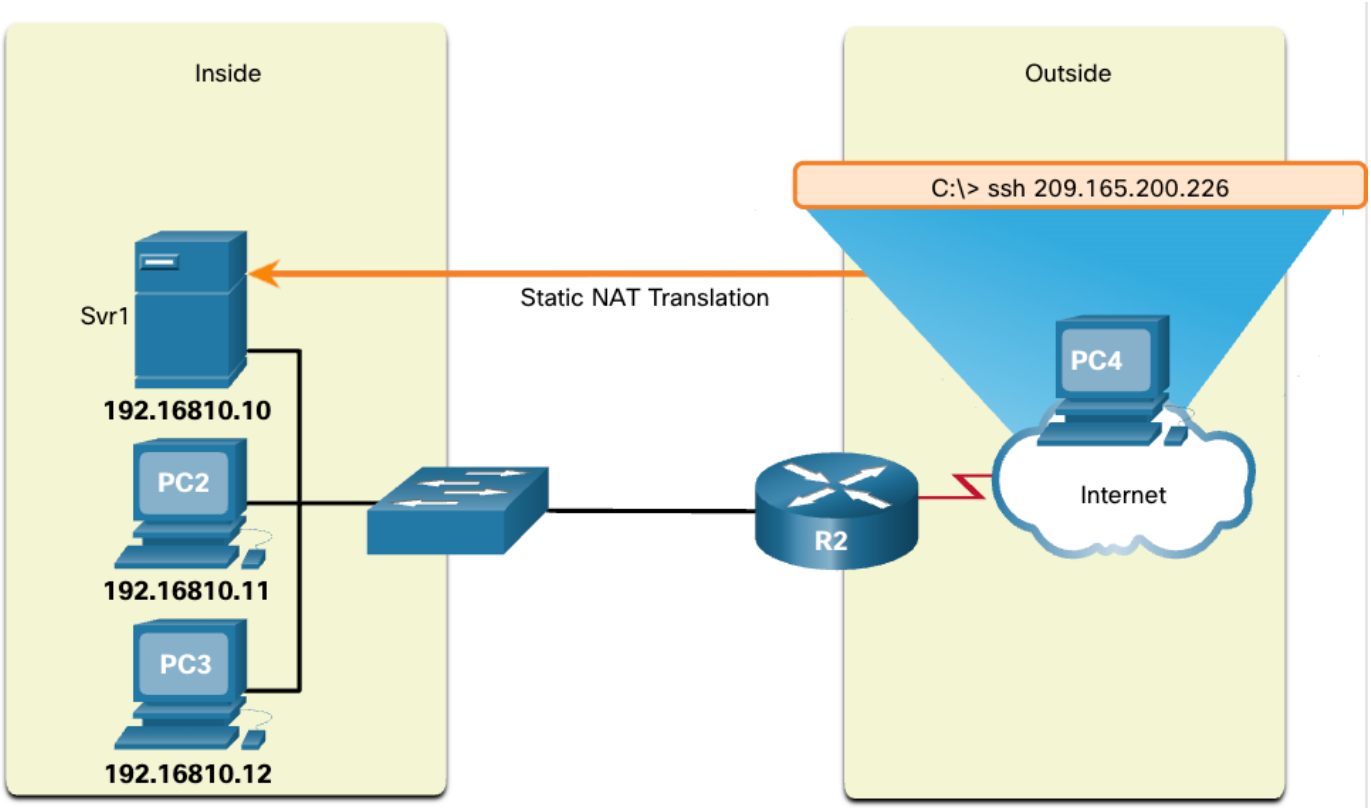
Types of NAT

Static NAT

Now that you have learned about NAT and how it works, this topic will discuss the many versions of NAT that are available to you.

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 appear to have public IPv4 addresses.



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

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 use SSH to gain access to the inside

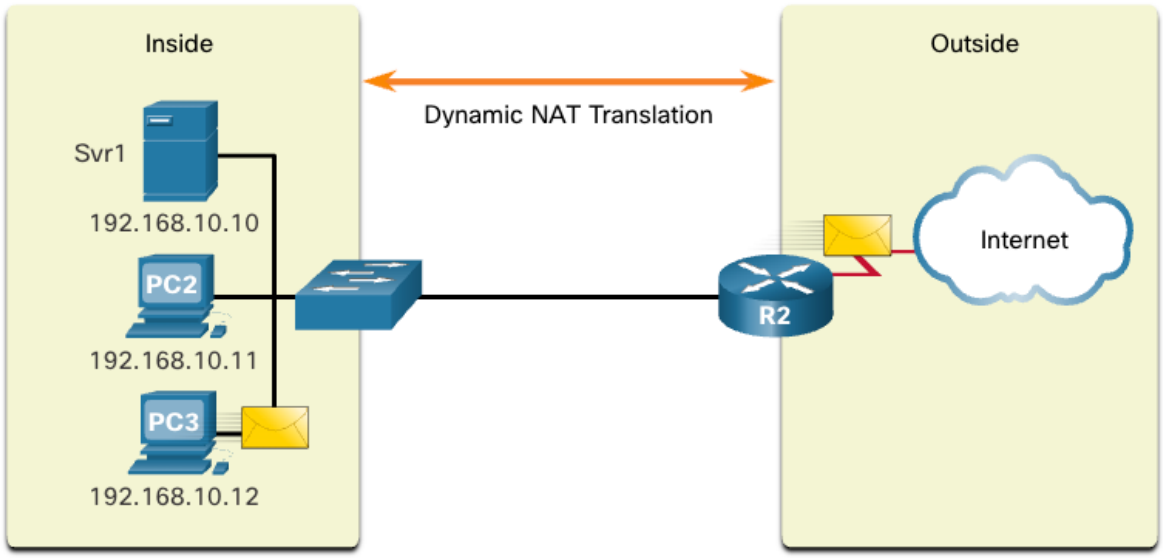
global address of Svr1 (209.165.200.226). R2 translates this inside global address to the inside local address 192.168.10.10 and connects the session to Svr1.

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

Dynamic NAT

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

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.



IPv4 NAT Pool	
Inside Local Address	**Inside Global Address Pool - Addresses reachable via R2**
192.168.10.12	209.165.200.226
Available	209.165.200.227
Available	209.165.200.228
Available	209.165.200.229
Available	209.165.200.230

Port Address Translation

Port Address Translation (PAT), also known as NAT overload, maps multiple private IPv4 addresses to a single public IPv4 address or a few addresses. 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 for both the home and the enterprise.

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, or a specially assigned query ID for ICMP, 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.

Next Available Port

PAT attempts to preserve the original source port. However, if the original source port is already used, PAT assigns the first available port number starting from the beginning of the appropriate port group 0-511, 512-1,023, or 1,024-65,535. When there are no more ports available and there is more than one external address in the address pool, PAT moves to the next address to try to allocate the original source port. This process continues until there are no more available ports or external IPv4 addresses.

Packets without a Layer 4 Segment

What about IPv4 packets carrying data other than a TCP or UDP segment? These packets do not contain a Layer 4 port number. PAT translates most common protocols carried by IPv4 that do not use TCP or UDP as a transport layer protocol. The most common of these is ICMPv4. Each of these types of protocols is handled differently by PAT. For example, ICMPv4 query messages, echo requests, and echo replies include a Query ID. ICMPv4 uses the Query ID to identify an echo request with its corresponding echo reply. The Query ID is incremented with each echo request sent. PAT uses the Query ID instead of a Layer 4 port number.

Note: Other ICMPv4 messages do not use the Query ID. These messages and other protocols that do not use TCP or UDP port numbers vary and are beyond the scope of this curriculum.

Advantages of NAT

NAT solves our problem of not having enough IPv4 addresses, but it can also create other problems. This topic addresses the advantages and disadvantage of NAT.

NAT provides many benefits, including the following:

- NAT conserves the legally registered addressing scheme by allowing the privatization of intranets. NAT conserves addresses through application port-level multiplexing. With NAT overload (PAT), 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.
- Using RFC 1918 IPv4 addresses, NAT hides the IPv4 addresses of users and other devices. Some people consider this a security feature; however, most experts agree that NAT does not provide security. A stateful firewall is what provides security on the edge of the network.

Disadvantages of NAT

NAT does have 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 forwarding delays because the translation of each IPv4 address within the packet headers takes time. The first packet is always process-switched going 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.

The forwarding delays caused by the NAT process becomes more of an issue as the pools of public IPv4 addresses for ISPs become depleted. Many ISPs are having to assign customers a private IPv4 address instead of a public IPv4 address. This means the customer's router translates the packet from its private IPv4 address to the private IPv4 address of the ISP. Before forwarding the packet to another provider, the ISP will then perform NAT again, translating its private IPv4 addresses to one of its limited number of public IPv4 addresses. This process of two layers of NAT translation is known as Carrier Grade NAT (CGN).

Another disadvantage of using NAT is that end-to-end addressing is lost. This is known as the end-to-end principle. 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.

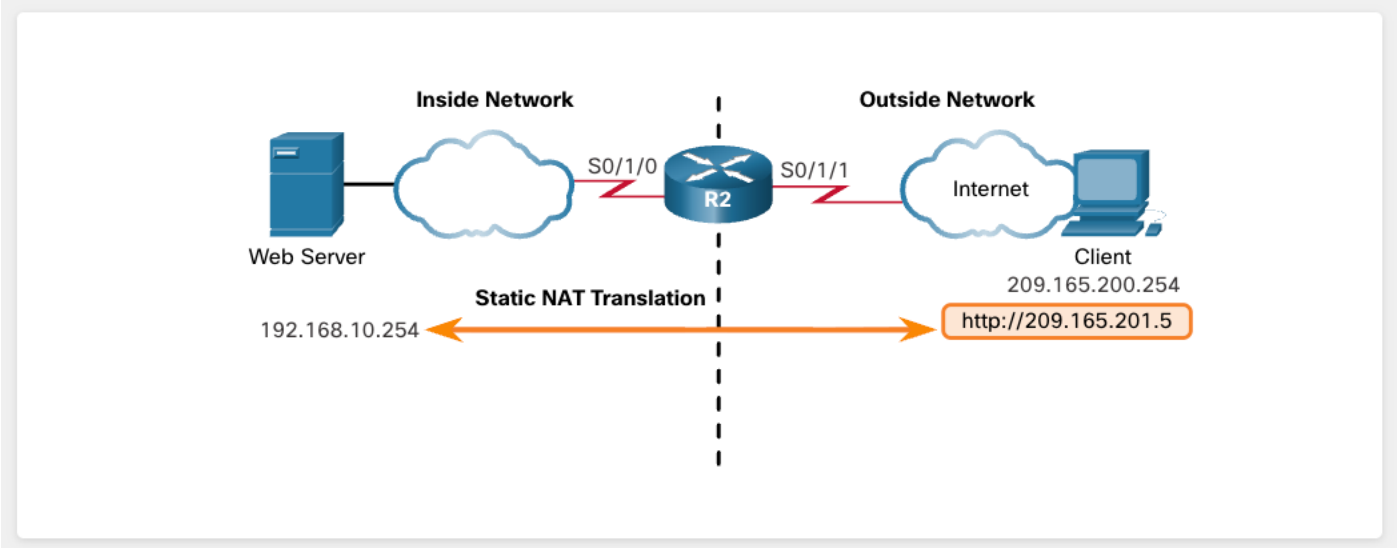
Using NAT also complicates the use of tunneling protocols, such as IPsec, because NAT modifies values in the headers, causing integrity checks to fail.

Services that require the initiation of TCP connections from the outside network, or stateless protocols, such as those using UDP, can be disrupted. Unless the NAT router has been configured to support such protocols, incoming packets cannot reach their destination. Some protocols can accommodate one instance of NAT between participating hosts (passive mode FTP, for example), but fail when both systems are separated from the internet by NAT.

Static NAT

In this topic, you will learn how to configure and verify static NAT. It includes a Packet Tracer activity to test your skills and knowledge. Static NAT is a one-to-one mapping between an inside address and an outside address. Static NAT allows external devices to initiate connections to internal devices using the statically assigned public address. For instance, an internal web server may be mapped to a specific inside global address so that it is accessible from outside networks.

The figure 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.



Configure Static NAT for Cisco

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 are configured as a static NAT translation.

```
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5
```

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 R2 Serial 0/1/0 interface is an inside interface and Serial 0/1/1 is an outside interface.

```
R2(config)# interface serial 0/1/0
R2(config-if)# ip address 192.168.1.2 255.255.255.252
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface serial 0/1/1
R2(config-if)# ip address 209.165.200.1 255.255.255.252
R2(config-if)# ip nat outside
```

With this configuration in place, packets arriving on the inside interface of R2 (Serial 0/1/0) from the configured inside local IPv4 address (192.168.10.254) are translated and then forwarded towards the outside network. Packets arriving on the outside interface of R2 (Serial 0/1/1), that are addressed to the configured inside global IPv4 address (209.165.201.5), are translated to the inside local address (192.168.10.254) and then forwarded to the inside network.

Configure Static NAT for Juniper

Define the Rule Set
Create a rule set and specify the originating traffic zone (usually the `untrust` or `outside` zone).

set security nat static rule-set RS1 from zone outside

Create the NAT Rule
Define a rule within the set that matches the public "destination" IP and translates it to the private "prefix" IP.

set security nat static rule-set RS1 rule R1 match destination-address 209.165.201.5/32
set security nat static rule-set RS1 rule R1 then static-nat prefix 192.168.10.254/32

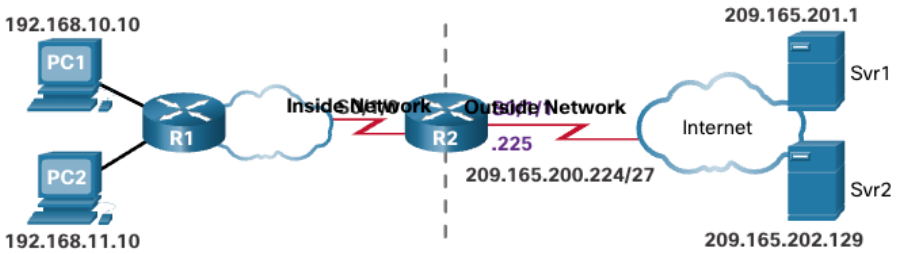
Configure Proxy ARP
If the public IP (e.g., 209.165.201.5) is in the same subnet as your external interface but is not assigned to the interface itself, you must enable proxy ARP so the device responds to ARP requests for that address.

set security nat proxy-arp interface ge-0/0/0.0 address 209.165.201.5/32

PAT

In this topic, you will learn how to configure and verify PAT. It includes a Packet Tracer activity to test your skills and knowledge. There are two ways to configure PAT, depending on how the ISP allocates public IPv4 addresses. In the first instance, the ISP allocates a single public IPv4 address that is required for the organization to connect to the ISP and in the other, it allocates more than one public IPv4 address to the organization.

Both methods will be demonstrated using the scenario shown in the figure.



NAT Table			
Inside Local Address	**Inside Global Address**	**Outside Global Address**	**Outside Local Address**
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.11.10:1444	209.165.200.225:1445	209.165.202.129:80	209.165.202.129:80

Configure PAT for Cisco

To configure PAT to use a single IPv4 address, simply add the keyword **overload** to the **ip nat inside source** command. The rest of the configuration is the similar to static and dynamic NAT configuration except that with PAT, multiple hosts can use the same public IPv4 address to access the internet.

In the example, all hosts from network 192.168.0.0/16 (matching ACL 1) that send traffic through router R2 to the internet will be translated to IPv4 address 209.165.200.225 (IPv4 address of interface S0/1/1). The traffic flows will be identified by port numbers in the NAT table because the **overload** keyword is configured.

```
R2(config)# ip nat inside source list 1 interface serial 0/1/1 overload
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# interface serial0/1/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface Serial0/1/1
R2(config-if)# ip nat outside
```

Configure PAT (Source NAT) for Juniper

```
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 192.168.0.0/24
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat interface
set security policies from-zone trust to-zone untrust policy internet-access match source-address any
set security policies from-zone trust to-zone untrust policy internet-access match destination-address
any
set security policies from-zone trust to-zone untrust policy internet-access match application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```