Network Protocols Laboratory

Lab manual 2018

Network protocols laboratory

Contents

Introduction Part

Lab 0: IP addressing (IPv6)

The Need for IPv6

IPv6 is designed to be the successor to IPv4. IPv6 has a larger 128-bit address space, providing for 340 undecillion addresses. (That is the number 340, followed by 36 zeroes.) However, IPv6 is more than just larger addresses. When the IETF began its development of a successor to IPv4, it used this opportunity to fix the limitations of IPv4 and include additional enhancements. One example is Internet Control Message Protocol version 6 (ICMPv6), which includes address resolution and address auto-configuration not found in ICMP for IPv4 (ICMPv4). ICMPv4 and ICMPv6 will be discussed later in this chapter.

Need for IPv6

The depletion of IPv4 address space has been the motivating factor for moving to IPv6. As Africa, Asia and other areas of the world become more connected to the Internet, there are not enough IPv4 addresses to accommodate this growth. As shown in the figure, four out of the five RIRs have run out of IPv4 addresses.

IPv4 has a theoretical maximum of 4.3 billion addresses. Private addresses in combination with Network Address Translation (NAT) have been instrumental in slowing the depletion of IPv4 address space. However, NAT breaks many applications and has limitations that severely impede peer-to-peer communications.

Internet of Everything

The Internet of today is significantly different than the Internet of past decades. The Internet of today is more than email, web pages, and file transfer between computers. The evolving Internet is becoming an Internet of things. No longer will the only devices accessing the Internet be computers, tablets, and smartphones. The sensor-equipped, Internet-ready devices of tomorrow will include everything from automobiles and biomedical devices, to household appliances and natural ecosystems.

With an increasing Internet population, a limited IPv4 address space, issues with NAT and an Internet of Everything, the time has come to begin the transition to IPv6.

IPv4 and IPv6 Coexistence

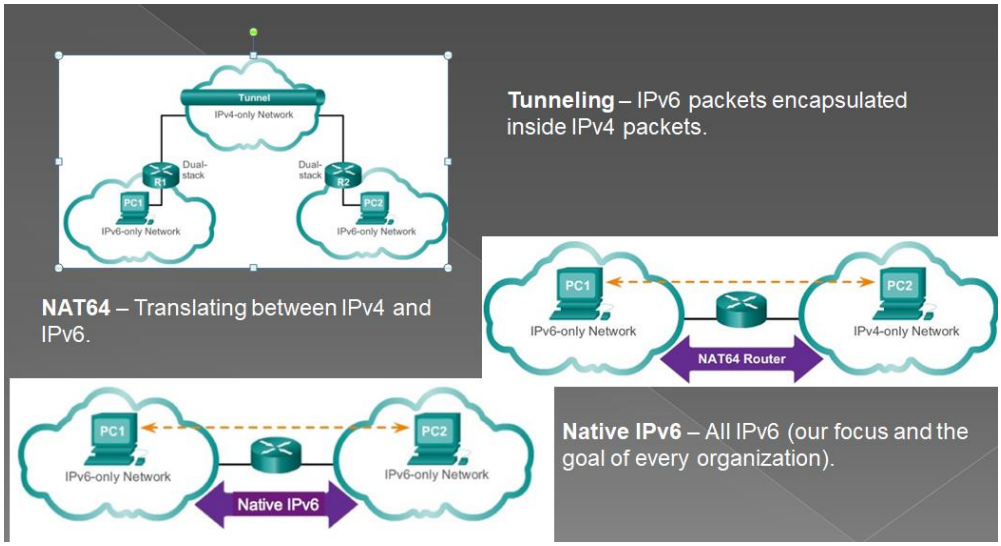
There is not a single date to move to IPv6. For the foreseeable future, both IPv4 and IPv6 will coexist. The transition is expected to take years. The IETF has created various protocols and tools to help

network administrators migrate their networks to IPv6. The migration techniques can be divided into three categories:

- **Dual Stack** – As shown in Figure 1, dual stack allows IPv4 and IPv6 to coexist on the same network segment. Dual stack devices run both IPv4 and IPv6 protocol stacks simultaneously.
- **Tunneling** – As shown in Figure 2, tunneling is a method of transporting an IPv6 packet over an IPv4 network. The IPv6 packet is encapsulated inside an IPv4 packet, similar to other types of data.
- **Translation** – As shown in Figure 3, Network Address Translation 64 (NAT64) allows IPv6-enabled devices to communicate with IPv4-enabled devices using a translation technique similar to NAT for IPv4. An IPv6 packet is translated to an IPv4 packet and vice versa.

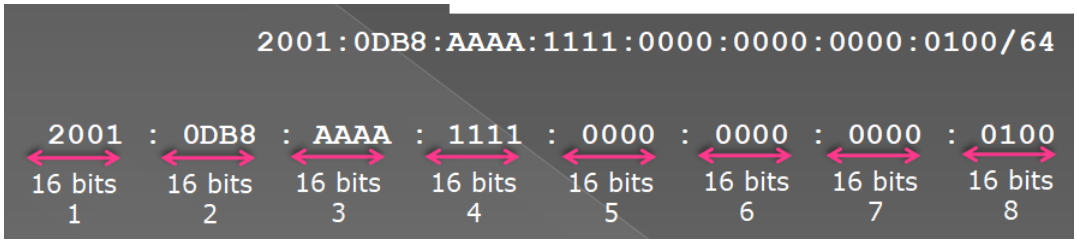
Note: Tunneling and translation are only used where needed. The goal should be native IPv6 communications from source to destination.

.



- Routers do not fragment IPv6 packets unless it is the source of the packet.
- Use of a Link-Local Address.
- ICMPv6 is more robust than ICMPv4.
- SLAAC (Stateless Address Autoconfiguration) for dynamic addressing.

IPv6 Address Notation



IPv6 addresses are 128-bit addresses represented in:

- Eight 16-bit segments or “hextets” (not a formal term)
- Hexadecimal (non-case sensitive) between 0000 and FFFF
- Separated by colons

- Reading and subnetting IPv6 is easier than IPv4!
- 340 undecillion addesses or …

Rule 1 – Omit Leading 0s

The first rule to help reduce the notation of IPv6 addresses is to omit any leading 0s (zeros) in any 16-bit section or hextet. For example:

- 01AB can be represented as 1AB
- 09F0 can be represented as 9F0
- 0A00 can be represented as A00
- 00AB can be represented as AB

Preferred	2001:0DB8:0000:1111:0000:0000:0000:0200
No leading 0s	2001: DB8: 0:1111: 0: 0: 0: 200

Rule 2 – Omit All 0 Segments

- The second rule can reduce this address even further:
- Any single, contiguous string of one or more 16-bit segments consisting of all zeroes can be represented with a double colon.
- Only a single contiguous string of all-zero segments can be represented with a double colon.
- Both of these are correct…

Preferred	2001:0DB8:0000:1111:0000:0000:0000:0200
No leading 0s	2001: DB8: 0:1111: 0: 0: 0: 200
Compressed	2001:DB8:0:1111::200

Preferred	FF02:0000:0000:0000:0000:0000:0000:0001
No leading 0s	FF02: 0: 0: 0: 0: 0: 0: 1
Compressed	FF02::1

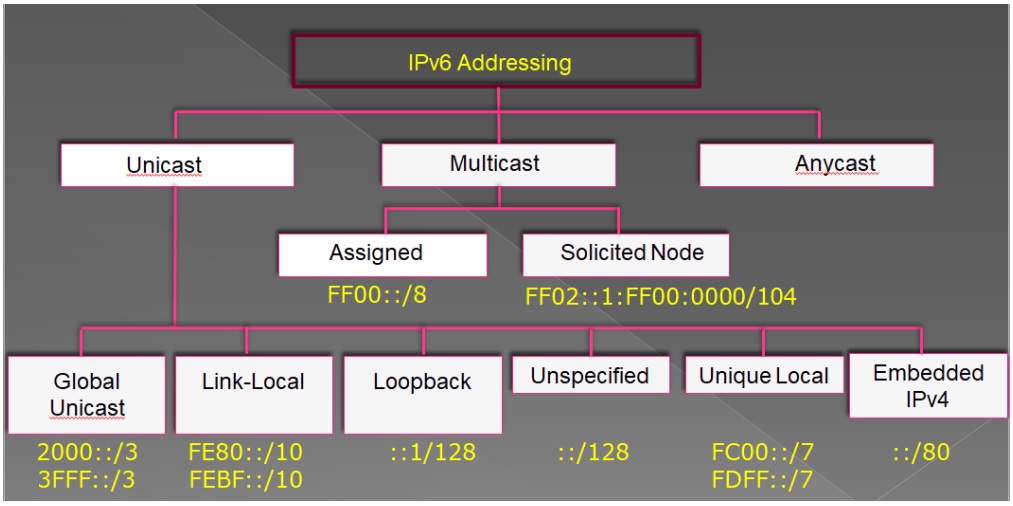
::1, ::

IPv6 Address Types

There are three types of IPv6 addresses:

- **Unicast** - An IPv6 unicast address uniquely identifies an interface on an IPv6-enabled device. As shown in the figure, a source IPv6 address must be a unicast address.
- **Multicast** - An IPv6 multicast address is used to send a single IPv6 packet to multiple destinations.
- **Anycast** - An IPv6 anycast address is any IPv6 unicast address that can be assigned to multiple devices. A packet sent to an anycast address is routed to the nearest device having that address. Anycast addresses are beyond the scope of this course.

Unlike IPv4, IPv6 does not have a broadcast address. However, there is an IPv6 all-nodes multicast address that essentially gives the same result.



IPv6 Unicast Addresses

An IPv6 unicast address uniquely identifies an interface on an IPv6-enabled device. A packet sent to a unicast address is received by the interface that is assigned that address. Similar to IPv4, a source IPv6 address must be a unicast address. The destination IPv6 address can be either a unicast or a multicast address.

The most common types of IPv6 unicast addresses are global unicast addresses (GUA) and link-local unicast addresses.

Global unicast

A global unicast address is similar to a public IPv4 address. These are globally unique, Internet routable addresses. Global unicast addresses can be configured statically or assigned dynamically.

Link-local

Link-local addresses are used to communicate with other devices on the same local link. With IPv6, the term link refers to a subnet. Link-local addresses are confined to a single link. Their uniqueness must only be confirmed on that link because they are not routable beyond the link. In other words, routers will not forward packets with a link-local source or destination address.

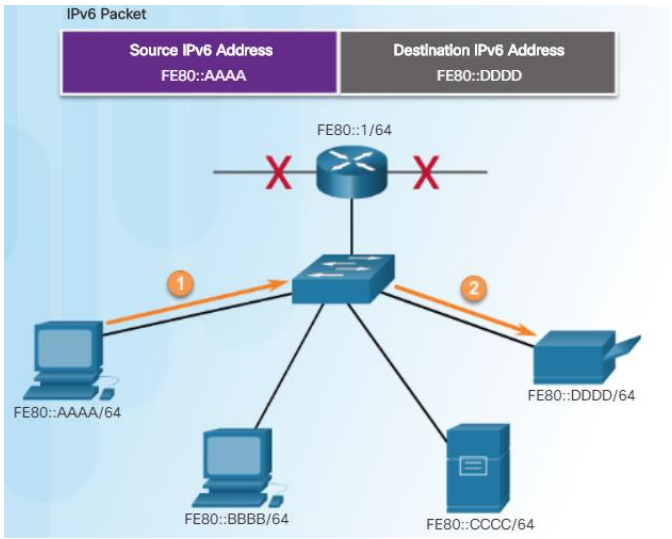
Unique local

Another type of unicast address is the unique local unicast address. IPv6 unique local addresses have some similarity to RFC 1918 private addresses for IPv4, but there are significant differences. Unique local addresses are used for local addressing within a site or between a limited number of sites. These addresses should not be routable in the global IPv6 and should not be translated to a global IPv6 address. Unique local addresses are in the range of FC00::/7 to FDFF::/7.

With IPv4, private addresses are combined with NAT/PAT to provide a many-to-one translation of private-to-public addresses. This is done because of the limited availability of IPv4 address space. Many sites also use the private nature of RFC 1918 addresses to help secure or hide their network from potential security risks. However, this was never the intended use of these technologies, and the IETF has always recommended that sites take the proper security precautions on their Internet-facing router. Unique local addresses can be used for devices that will never need or have access from another network.

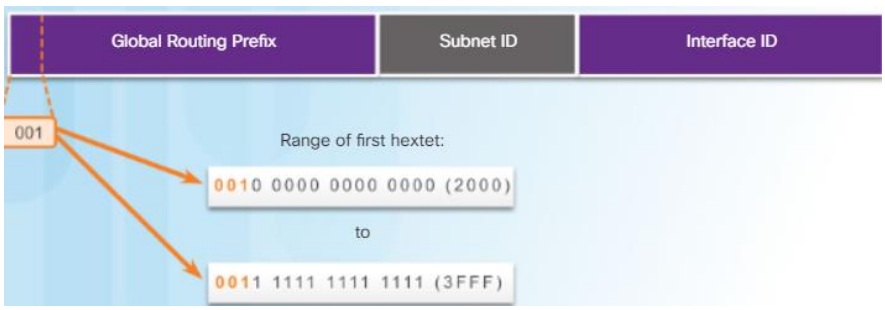
IPv6 Link-Local Unicast Addresses

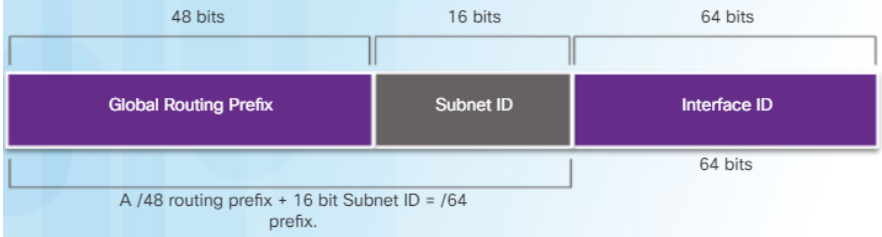
An IPv6 link-local address enables a device to communicate with other IPv6-enabled devices on the same link and only on that link (subnet). Packets with a source or destination link-local address cannot be routed beyond the link from which the packet originated.
The global unicast address is not a requirement. However, every IPv6-enabled network interface is required to have a link-local address.
If a link-local address is not configured manually on an interface, the device will automatically create its own without communicating with a DHCP server. IPv6-enabled hosts create an IPv6 link-local address even if the device has not been assigned a global unicast IPv6 address. This allows IPv6-enabled devices to communicate with other IPv6-enabled devices on the same subnet. This includes communication with the default gateway (router).



Structure of an IPv6 Global Unicast Address

- Global unicast addresses are unique routable and equivalent to IPv4 public addresses
    - Except under very specific circumstances, all end users will have a global unicast address
    - Terminology:
        - **Prefix** equivalent to *network address*
        - **Prefix length** equivalent to *subnet mask in IPv4*
        - **Interface ID** equivalent to *host portion*

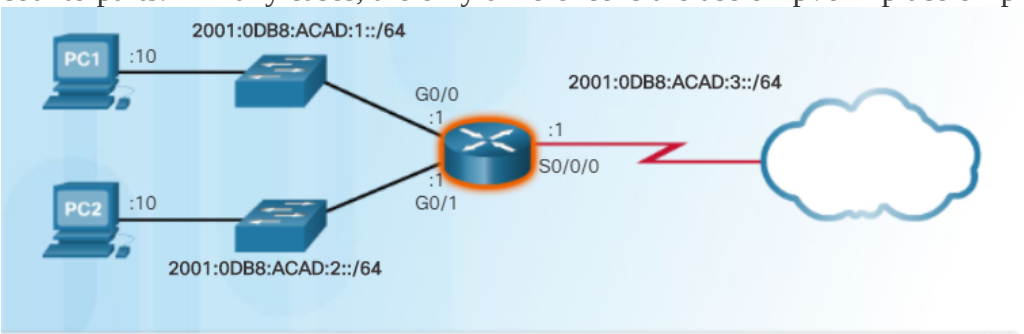




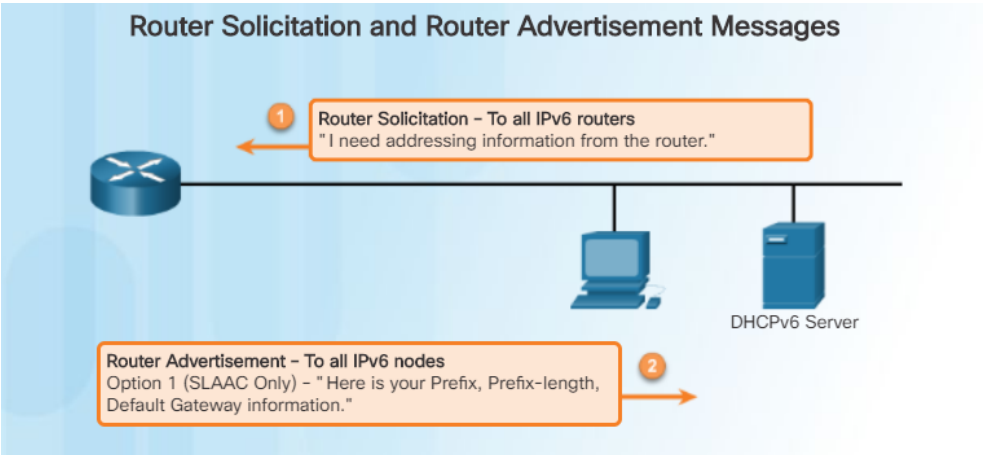
Static Configuration of a Global Unicast Address

Router Configuration

Most IPv6 configuration and verification commands in the Cisco IOS are similar to their IPv4 counterparts. In many cases, the only difference is the use of ipv6 in place of ip within the commands.




```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ipv6 address 2001:db8:acad:1::1/64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ipv6 address 2001:db8:acad:2::1/64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ipv6 address 2001:db8:acad:3::1/64
R1(config-if)#clock rate 56000
R1(config-if)#no shutdown
```



Router Advertisement Options

Option 1 (SLAAC Only) – "I'm everything you need (Prefix, Prefix-length, Default Gateway)"
Option 2 (SLAAC and DHCPv6) – "Here is my information but you need to get other information such as DNS addresses from a DHCPv6 server."
Option 3 (DHCPv6 Only) – "I can't help you. Ask a DHCPv6 server for all your information."

Dynamic Configuration - SLAAC

Stateless Address Autoconfiguration (SLAAC) is a method that allows a device to obtain its prefix, prefix length, default gateway address, and other information from an *IPv6 router*without the use of a DHCPv6 server. Using SLAAC, devices rely on the local router's ICMPv6 Router Advertisement (RA) messages to obtain the necessary information.

IPv6 routers periodically send out ICMPv6 RA messages, every 200 seconds, to all IPv6-enabled devices on the network. An RA message will also be sent in response to a host sending an ICMPv6 Router Solicitation (RS) message.

IPv6 routing is not enabled by default. To enable a router as an IPv6 router, the **ipv6 unicast-routing** global configuration command must be used.

Note: IPv6 addresses can be configured on a router without it being an IPv6 router.

The ICMPv6 RA message is a suggestion to a device on how to obtain an IPv6 global unicast address. The ultimate decision is up to the device's operating system. The ICMPv6 RA message includes:

- **Network prefix and prefix length** – Tells the device which network it belongs to.
- **Default gateway address** – This is an IPv6 link-local address, the source IPv6 address of the RA message.
- **DNS addresses and domain name** – Addresses of DNS servers and a domain name.

As shown in Figure 1, there are three options for RA messages:

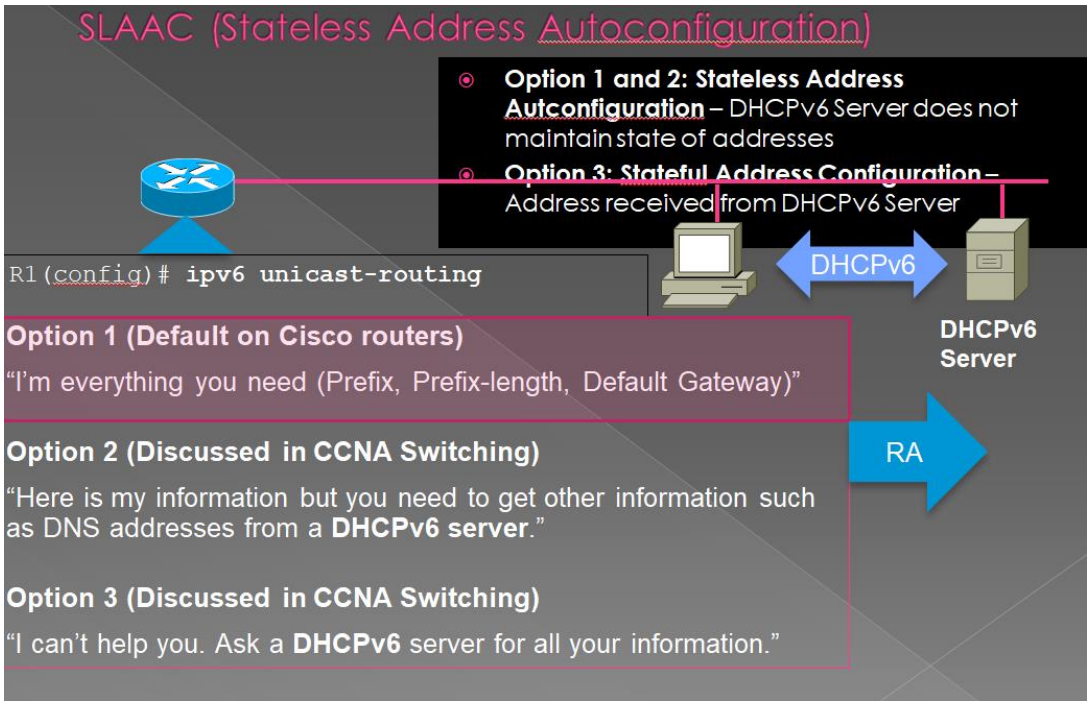
- Option 1: SLAAC
- Option 2: SLAAC with a stateless DHCPv6 server
- Option 3: Stateful DHCPv6 (no SLAAC)

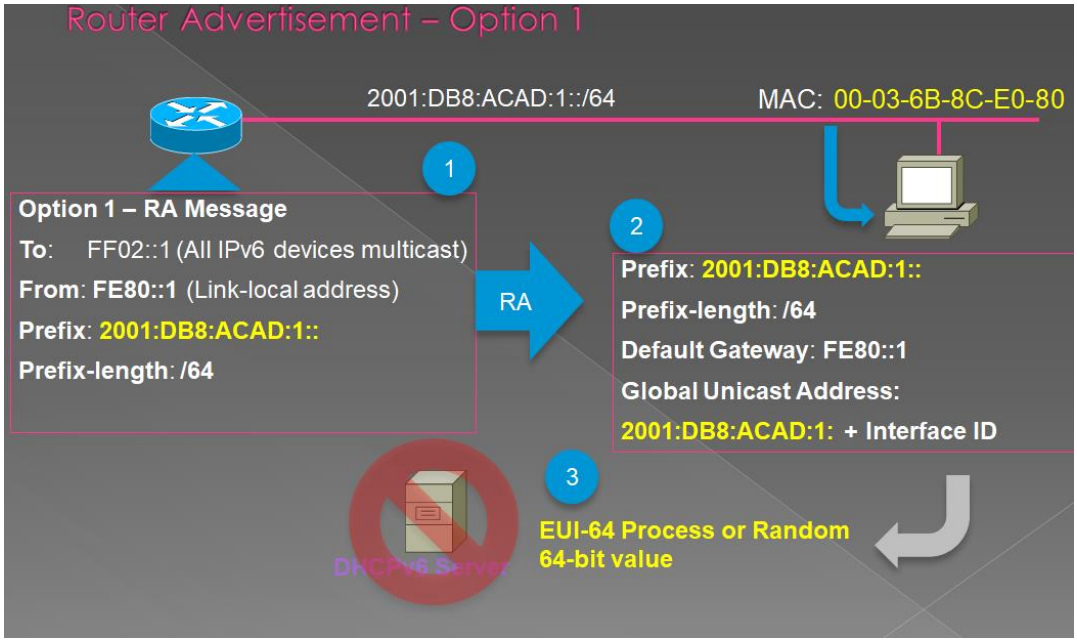
RA Option 1: SLAAC

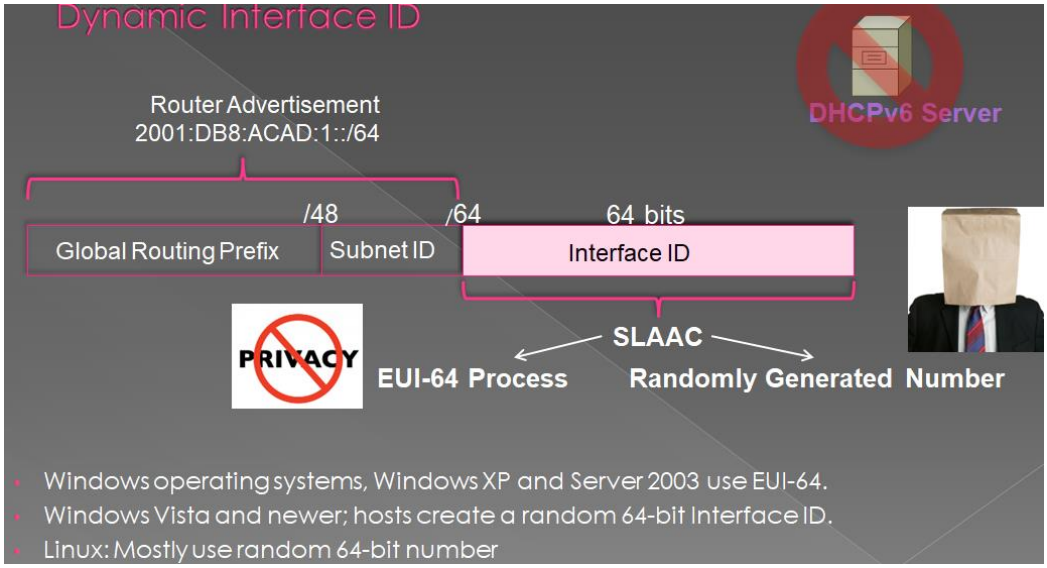
By default, the RA message suggests that the receiving device use the information in the RA message to create its own IPv6 global unicast address and for all other information. The services of a DHCPv6 server are not required.

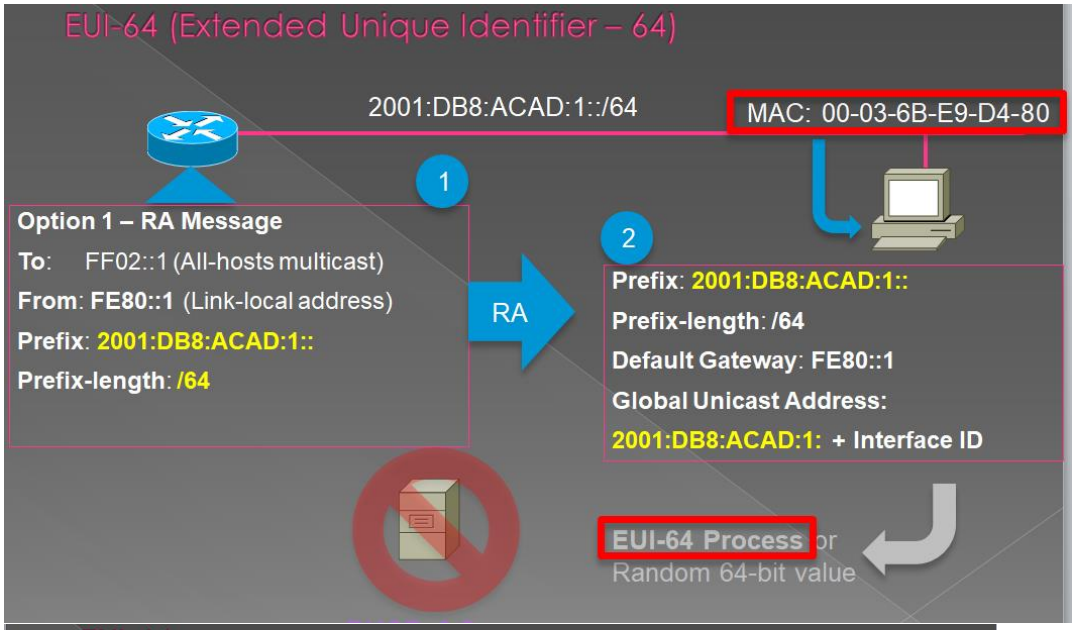
SLAAC is stateless, which means there is no central server (for example, a stateful DHCPv6 server) allocating global unicast addresses and keeping a list of devices and their addresses. With SLAAC, the client device uses the information in the RA message to create its own global unicast address. As shown in Figure 2, the two parts of the address are created as follows:

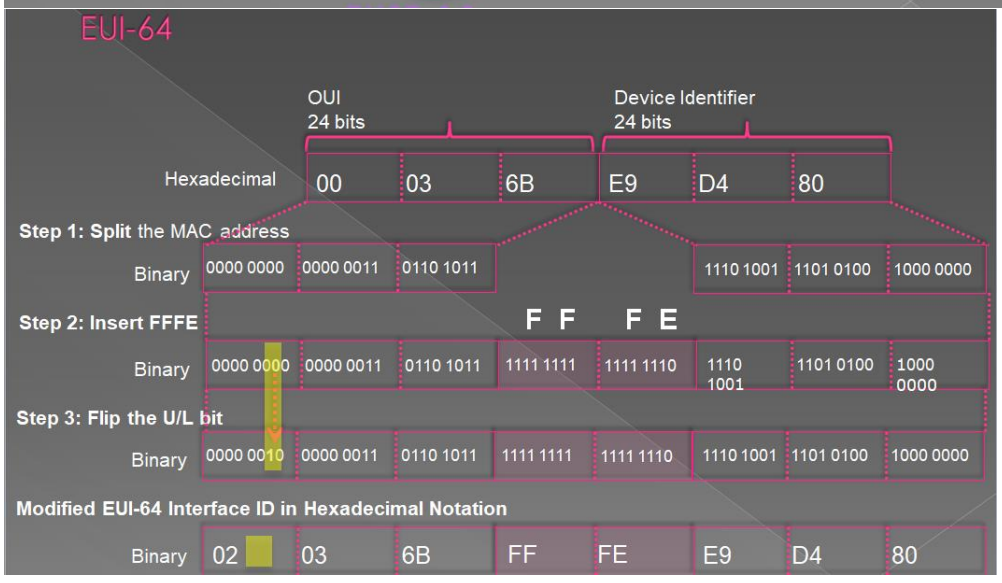
- **Prefix** – Received in the RA message
- **Interface ID** – Uses the EUI-64 process or by generating a random 64-bit number


SLAAC (Stateless Address Autoconfiguration)
Option 1 and 2: Stateless Address Autconfiguration – DHCPv6 Server does not maintain state of addresses
Option 3: Stateful Address Configuration – Address received from DHCPv6 Server
R1(config)# ipv6 unicast-routing
DHCPv6
DHCPv6 Server
Option 1 (Default on Cisco routers)
"I'm everything you need (Prefix, Prefix-length, Default Gateway)"
Option 2 (Discussed in CCNA Switching)
"Here is my information but you need to get other information such as DNS addresses from a DHCPv6 server."
Option 3 (Discussed in CCNA Switching)
"I can't help you. Ask a DHCPv6 server for all your information."
RA


Router Advertisement – Option 1
2001:DB8:ACAD:1::/64
MAC: 00-03-6B-8C-E0-80
1
Option 1 – RA Message
To: FF02::1 (All IPv6 devices multicast)
From: FE80::1 (Link-local address)
Prefix: 2001:DB8:ACAD:1::
Prefix-length: /64
RA
2
Prefix: 2001:DB8:ACAD:1::
Prefix-length: /64
Default Gateway: FE80::1
Global Unicast Address:
2001:DB8:ACAD:1: + Interface ID
3
EUI-64 Process or Random 64-bit value
DHCPv6 Server

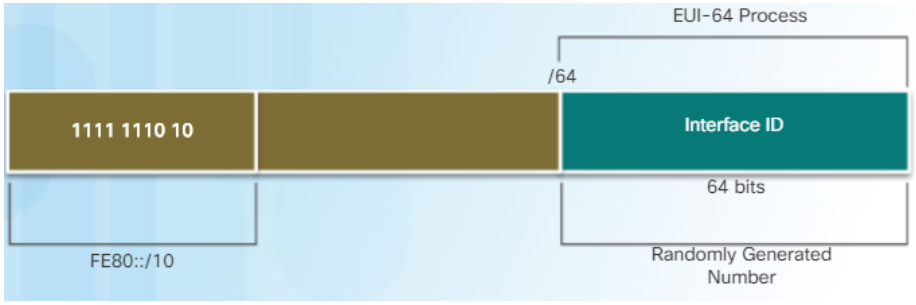

Dynamic Interface ID
Router Advertisement
2001:DB8:ACAD:1::/64
DHCPv6 Server
/48
/64
64 bits
Global Routing Prefix
Subnet ID
Interface ID
PRIVACY
SLAAC
EUI-64 Process
Randomly Generated Number
Windows operating systems, Windows XP and Server 2003 use EUI-64.
Windows Vista and newer; hosts create a random 64-bit Interface ID.
Linux: Mostly use random 64-bit number

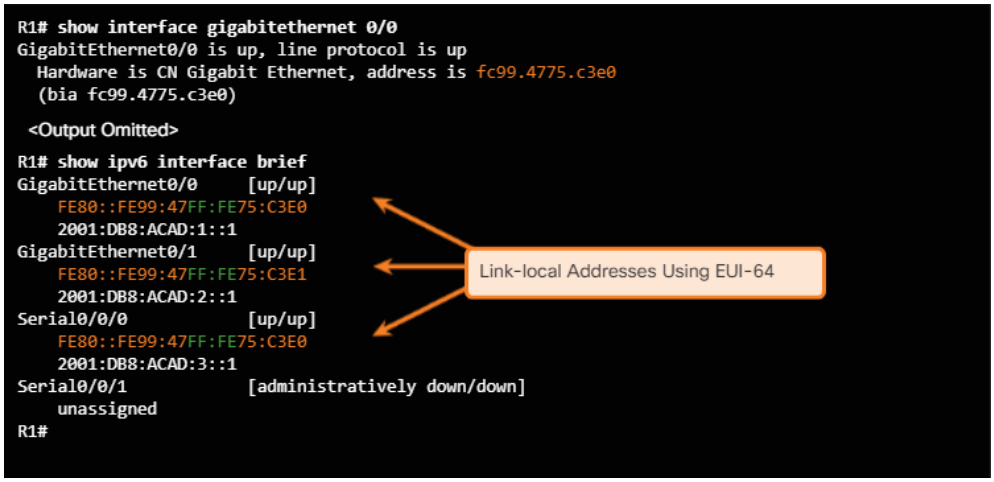

EUI-64 (Extended Unique Identifier – 64)
2001:DB8:ACAD:1::/64
MAC: 00-03-6B-E9-D4-80
1
Option 1 – RA Message
To: FF02::1 (All-hosts multicast)
From: FE80::1 (Link-local address)
Prefix: 2001:DB8:ACAD:1::
Prefix-length: /64
RA
2
Prefix: 2001:DB8:ACAD:1::
Prefix-length: /64
Default Gateway: FE80::1
Global Unicast Address:
2001:DB8:ACAD:1: + Interface ID
EUI-64 Process or
Random 64-bit value


EUI-64
OUI
24 bits
Device Identifier
24 bits
Hexadecimal
00 03 6B E9 D4 80
Step 1: Split the MAC address
Binary
0000 0000 0000 0011 0110 1011 1110 1001 1101 0100 1000 0000
Step 2: Insert FFFE
F F F E
Binary
0000 0000 0000 0011 0110 1011 1111 1111 1111 1110 1110 1001 1101 0100 1000 0000
Step 3: Flip the U/L bit
Binary
0000 0010 0000 0011 0110 1011 1111 1111 1111 1110 1110 1001 1101 0100 1000 0000
Modified EUI-64 Interface ID in Hexadecimal Notation
Binary
02 03 6B FF FE E9 D4 80

Link-local Unicast

- Used to communicate with other devices on the link.
- Are NOT routable off the link (network).
- Only have to be unique on the link.
- Are not included in the IPv6 routing table.
- An IPv6 device must have at least a link-local address.
- Used by:
    - Hosts to communicate to the IPv6 network before it has a global unicast address.
    - Router's link-local address is used by hosts as the default gateway address.
    - Adjacent routers to exchange routing updates





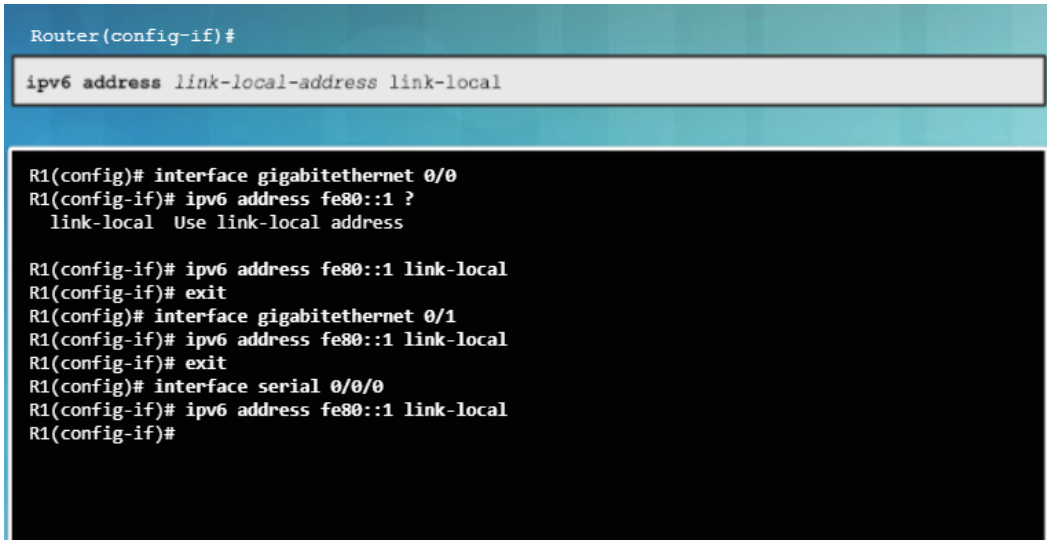
Static Link-Local Addresses

Configuring the link-local address manually provides the ability to create an address that is recognizable and easier to remember. Typically, it is only necessary to create recognizable link-local addresses on routers. This is beneficial because router link-local addresses are used as default gateway addresses and in routing advertisement messages.

Link-local addresses can be configured manually using the same interface command used to create IPv6 global unicast addresses but with the additional **link-local** parameter. When an address begins with this hextet within the range of FE80 to FEBF, the link-local parameter must follow the address.

The figure shows the configuration of a link-local address using the **ipv6 address** interface command. The link-local address FE80::1 is used to make it easily recognizable as belonging to router R1. The

same IPv6 link-local address is configured on all of R1's interfaces. FE80::1 can be configured on each link because it only has to be unique on that link.



Verifying IPv6 Address Configuration

- The **show interface** command displays the MAC address of the Ethernet interfaces. EUI-64 uses this MAC address to generate the Interface ID for the link-local address.
- Additionally, the **show ipv6 interface brief** command displays abbreviated output for each of the interfaces. The **[up/up]** output on the same line as the interface indicates the Layer 1/Layer 2 interface state. This is the same as the **Status** and **Protocol** columns in the equivalent IPv4 command.
- Notice that each interface has two IPv6 addresses. The second address for each interface is the global unicast address that was configured. The first address, the one that begins with FE80, is the link-local unicast address for the interface. Recall that the link-local address is automatically added to the interface when a global unicast address is assigned.
- Also, notice that R1's Serial 0/0/0 link-local address is the same as its GigabitEthernet 0/0 interface. Serial interfaces do not have Ethernet MAC addresses, so Cisco IOS uses the MAC address of the first available Ethernet interface. This is possible because link-local interfaces only have to be unique on that link.
- The link-local address of the router interface is typically the default gateway address for devices on that link or network.
- **show ipv6 route**command can be used to verify that IPv6 networks and specific IPv6 interface addresses have been installed in the IPv6 routing table. The **show ipv6 route** command will only display IPv6 networks, not IPv4 networks.
- The **ping** command for IPv6 is identical to the command used with IPv4, except that an IPv6 address is used.

Assigned IPv6 Multicast Addresses

IPv6 multicast addresses are similar to IPv4 multicast addresses. Recall that a multicast address is used to send a single packet to one or more destinations (multicast group). IPv6 multicast addresses have the prefix FF00::/8.
Note: Multicast addresses can only be destination addresses and not source addresses.
There are two types of IPv6 multicast addresses:

- Assigned multicast
- Solicited node multicast

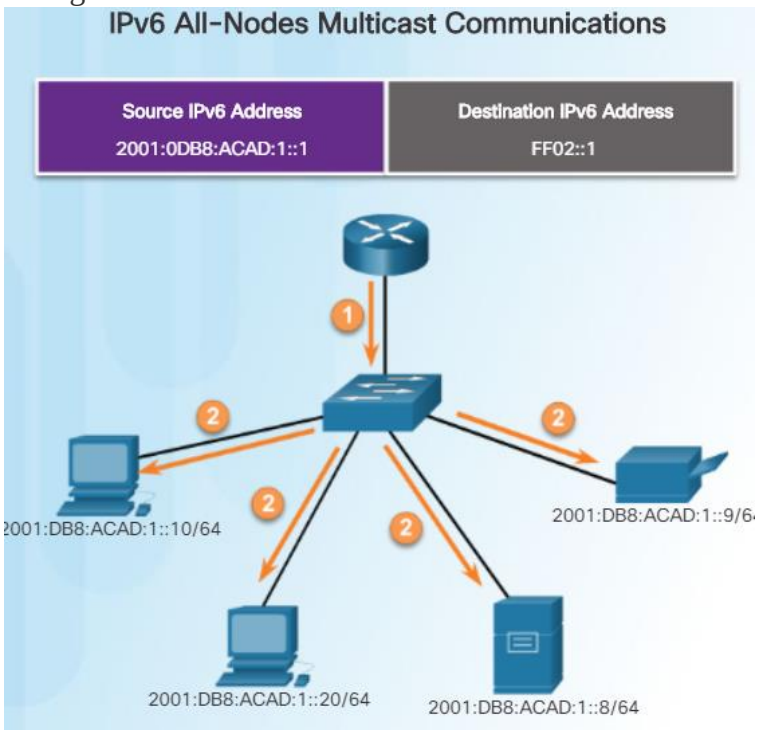
Assigned Multicast

Assigned multicast addresses are reserved multicast addresses for predefined groups of devices. An assigned multicast address is a single address used to reach a group of devices running a common protocol or service. Assigned multicast addresses are used in context with specific protocols such as DHCPv6.

Two common IPv6 assigned multicast groups include:

- **FF02::1 All-nodes multicast group** – This is a multicast group that all IPv6-enabled devices join. A packet sent to this group is received and processed by all IPv6 interfaces on the link or network. This has the same effect as a broadcast address in IPv4. The figure shows an example of communication using the all-nodes multicast address. An IPv6 router sends Internet Control Message Protocol version 6 (ICMPv6) RA messages to the all-node multicast group. The RA message informs all IPv6-enabled devices on the network about addressing information, such as the prefix, prefix length, and default gateway.
- **FF02::2 All-routers multicast group** – This is a multicast group that all IPv6 routers join. A router becomes a member of this group when it is enabled as an IPv6 router with the **ipv6 unicast-routing** global configuration command. A packet sent to this group is received and processed by all IPv6 routers on the link or network.

IPv6-enabled devices send ICMPv6 Router Solicitation (RS) messages to the all-routers multicast address. The RS message requests an RA message from the IPv6 router to assist the device in its address configuration.



Part One: Routing protocols

Lab 1: Routing Information Protocol: RIPv2

Lab 2: Enhanced Interior Gateway Routing Protocol (EIGRP) part 1

Lab 3: Enhanced Interior Gateway Routing Protocol (EIGRP) part 2

Lab 4: Single-Area OSPF OSPFv2 and OSPFv3

Lab 5: Multiarea OSPF

Lab 1: Routing Information Protocol: RIPv2

Classifying Routing Protocols

Dynamic routing protocols are used to facilitate the exchange of routing information between routers. 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 routing protocol's choice of best paths. The purpose of dynamic routing protocols includes:

- 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

Routing protocols can be classified into different groups according to their characteristics. Specifically, routing protocols can be classified by their:

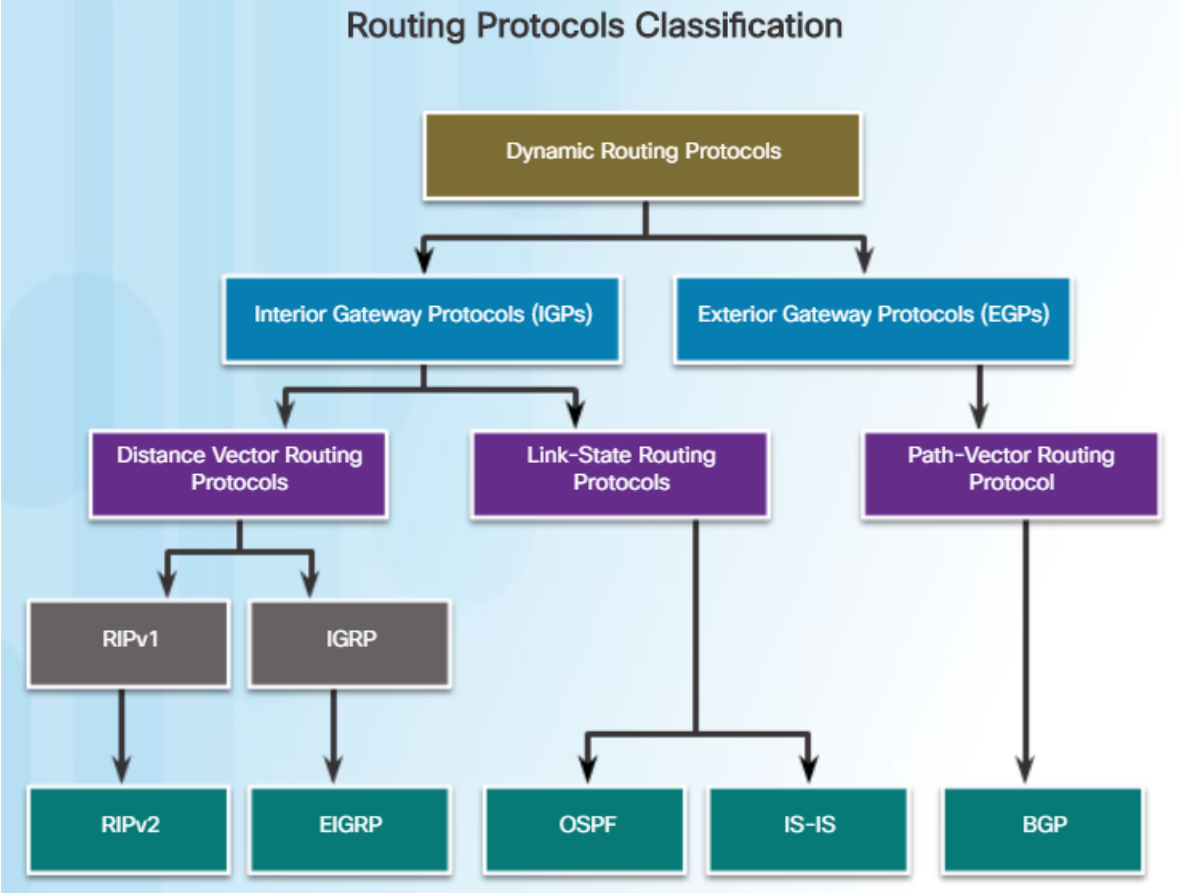
- **Purpose** - Interior Gateway Protocol (IGP) or Exterior Gateway Protocol (EGP)
- **Operation** - Distance vector protocol, link-state protocol, or path-vector protocol
- **Behavior** - Classful (legacy) or classless protocol

For example, IPv4 routing protocols are classified as follows:

- **RIPv1 (legacy)** - IGP, distance vector, classful protocol
- **IGRP (legacy)** - IGP, distance vector, classful protocol developed by Cisco (deprecated from 12.2 IOS and later)
- **RIPv2** - IGP, distance vector, classless protocol
- **EIGRP** - IGP, distance vector, classless protocol developed by Cisco
- **OSPF** - IGP, link-state, classless protocol
- **IS-IS** - IGP, link-state, classless protocol
- **BGP** - EGP, path-vector, classless protocol

The classful routing protocols, RIPv1 and IGRP, are legacy protocols and are only used in older networks. These routing protocols have evolved into the classless routing protocols, RIPv2 and EIGRP, respectively. Link-state routing protocols are classless by nature.

The figure displays a hierarchical view of dynamic routing protocol classification



Routing protocols are used to facilitate the exchange of routing information between routers. 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 routing protocol's choice of best paths. The purpose of dynamic routing protocols includes:

- 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:

- Data structures - Routing protocols typically use tables or databases for its 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 best path determination.

Routing protocols allow routers to dynamically share information about remote networks and automatically offer this information to their own routing tables. Click Play in the figure to see an animation of this process.

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 is there is not another routing source with a lower administrative distance. For example, a static route with an administrative distance of 1 will have precedence over the same network learned by a dynamic routing protocol. 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 also to find alternate paths when there is a link failure to a current network.

Routing Information Protocol

The Routing Information Protocol (RIP) was a first generation routing protocol for IPv4 originally specified in RFC 1058. It is easy to configure, making it a good choice for small networks.

RIPv1 has the following key characteristics:

- Routing updates are broadcasted (255.255.255.255) every 30 seconds.

- The hop count is used as the metric for path selection.

- A hop count greater than 15 hops is deemed infinite (too far). That 15th hop router would not propagate the routing update to the next router.

In 1993, RIPv1 was updated to a classless routing protocol known as RIP version 2 (RIPv2). RIPv2 included the following improvements:

- **Classless routing protocol** - It supports VLSM and CIDR, because it includes the subnet mask in the routing updates.

- **Increased efficiency** - It forwards updates to multicast address 224.0.0.9, instead of the broadcast address 255.255.255.255.

- **Reduced routing entries** - It supports manual route summarization on any interface.

- **Secure** - It supports an authentication mechanism to secure routing table updates between neighbors.

The table in the figure summarizes the differences between RIPv1 and RIPv2.

RIP updates are encapsulated into a UDP segment, with both source and destination port numbers set to UDP port 520.

In 1997, the IPv6 enabled version of RIP was released. RIPng is based on RIPv2. It still has a 15 hop limitation and the administrative distance is 120.

RIPv1 versus RIPv2

Characteristics and Features	RIPv1	RIPv2
Metric	Both use hop count as a simple metric. The maximum number of hops is 15.	
Updates Forwarded to Address	255.255.255.255	224.0.0.9
Supports VLSM	✗	✓
Supports CIDR	✗	✓
Supports Summarization	✗	✓
Supports Authentication	✗	✓

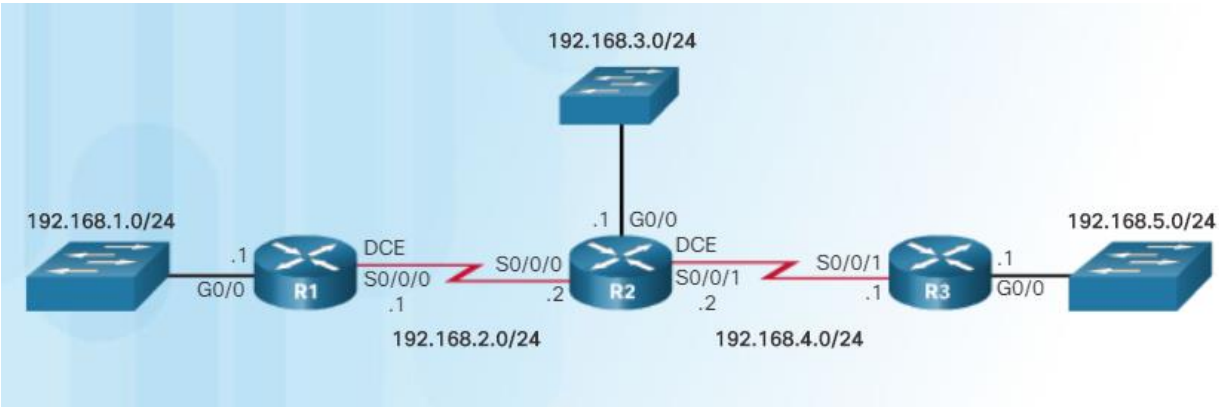
Router RIP Configuration Mode

Although RIP is rarely used in modern networks, it is useful as a foundation for understanding basic network routing. This section provides a brief overview of how to configure basic RIP settings and how to verify RIPv2.

Refer to the reference topology in Figure 1 and the addressing table in Figure 2. In this scenario, all routers have been configured with basic management features and all interfaces identified in the reference topology are configured and enabled. There are no static routes configured and no routing protocols enabled; therefore, remote network access is currently impossible. RIPv1 is used as the dynamic routing protocol. To enable RIP, use the router rip command, as shown in Figure 3. This command does not directly start the RIP process. Instead, it provides access to the router configuration mode where the RIP routing settings are configured. When enabling RIP, the default version is RIPv1.

To disable and eliminate RIP, use the no router rip global configuration command. This command stops the RIP process and erases all existing RIP configurations.

Figure 4 displays the various RIP commands that can be configured. The highlighted keywords are covered in this section.



Addressing Table

Device	Interface	IP Address	Subnet Mask
R1	G0/0	192.168.1.1	255.255.255.0
	S0/0/0	192.168.2.1	255.255.255.0
R2	G0/0	192.168.3.1	255.255.255.0
	S0/0/0	192.168.2.2	255.255.255.0
	S0/0/1	192.168.4.2	255.255.255.0
R3	G0/0	192.168.5.1	255.255.255.0
	S0/0/1	192.168.4.1	255.255.255.0

```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# router rip
R1(config-router)#
```

RIP Configuration Options

```
R1(config-router)# ?

Router configuration commands:
  address-family         Enter Address Family command mode
  auto-summary           Enable automatic network number summarization
  default                Set a command to its defaults
  default-information    Control distribution of default information
  default-metric         Set metric of redistributed routes
  distance               Define an administrative distance
  distribute-list        Filter networks in routing updates
  exit                   Exit from routing protocol configuration mode
  flash-update-threshold Specify flash update threshold insecond
  help                   Description of the interactive help system
  input-queue            Specify input queue depth
  maximum-paths          Forward packets over multiple paths
  neighbor               Specify a neighbor router
  network                Enable routing on an IP network
  no                     Negate a command or set its defaults
  offset-list            Add or subtract offset from RIP metrics
  output-delay           Interpacket delay for RIP updates
  passive-interface      Suppress routing updates on an interface
  redistribute           Redistribute information from another routing protocol
  timers                 Adjust routing timers
  traffic-share          How to compute traffic share over alternate paths
  validate-update-source Perform sanity checks against source address of routing updates
  version                Set routing protocol version

R1(config-router)#
```

Advertise Networks

By entering the RIP router configuration mode, the router is instructed to run RIPv1. But the router still needs to know which local interfaces it should use for communication with other routers, as well as which locally connected networks it should advertise to those routers.

To enable RIP routing for a network, use thenetwork network-address router configuration mode command. Enter the classful network address for each directly connected network. This command:

- Enables RIP on all interfaces that belong to a specific network. Associated interfaces now both send and receive RIP updates.
- Advertises the specified network in RIP routing updates sent to other routers every 30 seconds.

Note: RIPv1 is a classful routing protocol for IPv4. Therefore, if a subnet address is entered, the IOS automatically converts it to the classful network address. For example, entering thenetwork 192.168.1.32 command would automatically be converted to network 192.168.1.0 in the running configuration file. The IOS does not give an error message, but instead corrects the input and enters the classful network address.

In Figure 1, the network command is used to advertise the R1 directly connected networks.

```
R1(config)# router rip
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.2.0
R1(config-router)#
```

Configure RIP on R2 to advertise the appropriate networks based on the topology. Start with the lowest IP network.

R2(config)# router rip

R2(config-router)# network 192.168.2.0

R2(config-router)# network 192.168.3.0

R2(config-router)# network 192.168.4.0

You are now logged into R3. Advertise the RIP networks for R3. Start with the lowest IP network.

R3(config)# router rip

R3(config-router)# network 192.168.4.0

R3(config-router)# network 192.168.5.0

Verify RIP Routing

The show ip protocols command displays the IPv4 routing protocol settings currently configured on the router. This output displayed in Figure 1 confirms most RIP parameters including:

1. RIP routing is configured and running on router R1.

2. The values of various timers; for example, the next routing update, is sent by R1 in 16 seconds.

3. The version of RIP configured is currently RIPv1.

4. R1 is currently summarizing at the classful network boundary.

5. The classful networks are advertised by R1. These are the networks that R1 includes in its RIP updates.

6. The RIP neighbors are listed, including their next-hop IP address, the associated AD that R2 uses for updates sent by this neighbor, and when the last update was received from this neighbor.

Note: This command is also very useful when verifying the operations of other routing protocols (i.e., EIGRP and OSPF).

The show ip route command displays the RIP routes installed in the routing table.

Enable and Verify RIPv2

By default, when a RIP process is configured on a Cisco router, it is running RIPv1, as shown in Figure 1. However, even though the router only sends RIPv1 messages, it can interpret both RIPv1 and RIPv2 messages. A RIPv1 router ignores the RIPv2 fields in the route entry.

Use the version 2 router configuration mode command to enable RIPv2, as shown in Figure 2. Notice how the show ip protocols command verifies that R2 is now configured to send and receive version 2 messages only. The RIP process now includes the subnet mask in all updates, making RIPv2 a classless routing protocol.

Note: Configuring version 1 enables RIPv1 only, while configuring no version returns the router to the default setting of sending version 1 updates but listening for version 1 and version 2 updates.

Disable Auto Summarization

As shown in Figure 1, RIPv2 automatically summarizes networks at major network boundaries by default, just like RIPv1.

To modify the default RIPv2 behavior of automatic summarization, use the no auto-summary router configuration mode command . This command has no effect when using RIPv1. When automatic summarization has been disabled, RIPv2 no longer summarizes networks to their classful address at boundary routers. RIPv2 now includes all subnets and their appropriate masks in its routing updates. The show ip protocols now states that “automatic network summarization is not in effect”.

Note: RIPv2 must be enabled before automatic summarization is disabled.

Configure Passive Interfaces

By default, RIP updates are forwarded out all RIP-enabled interfaces. However, RIP updates really only need to be sent out interfaces that are connected to other RIPenabled routers.

For instance, refer to the topology in Figure 1. RIP sends updates out of its G0/0 interface even though no RIP device exists on that LAN. R1 has no way of knowing this and, as a result, sends an update every 30 seconds. Sending out unneeded updates on a LAN impacts the network in three ways:

- **Wasted Bandwidth** - Bandwidth is used to transport unnecessary updates. Because RIP updates are either broadcasted or multicasted, switches also forward the updates out all ports.

- **Wasted Resources** - All devices on the LAN must process the update up to the transport layers, at which point the devices will discard the update.

- **Security Risk** - Advertising updates on a broadcast network is a security risk. RIP updates 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.

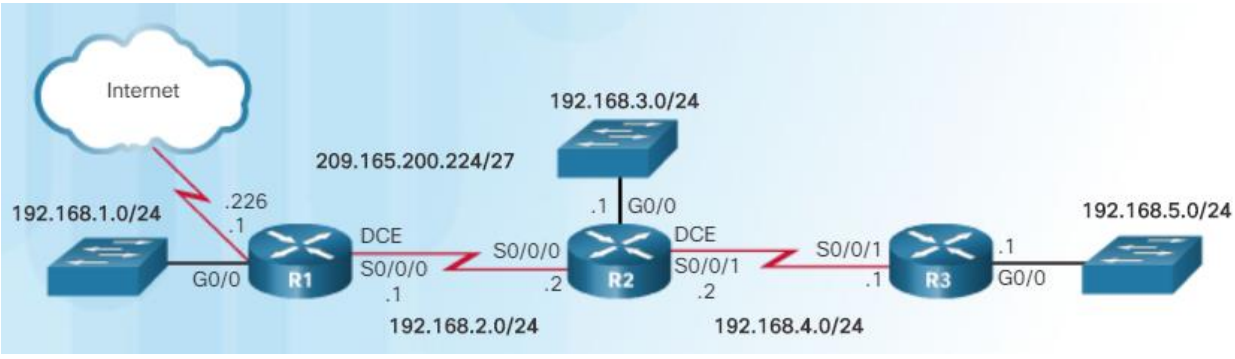
Use the **passive-interface** router configuration command to prevent the transmission of routing updates through a router interface, but still allow that network to be advertised to other routers. The command stops routing updates out the specified interface. However, the network that the specified interface belongs to is still advertised in routing updates that are sent out other interfaces.

Note: All routing protocols support the **passive-interface** command.

As an alternative, all interfaces can be made passive using the **passive-interface default** command. Interfaces that should not be passive can be re-enabled using the **no passive-interface** command.

Propagate a Default Route

Refer to Figure below. In this scenario, R1 is the edge router, single-homed to a service provider. Therefore, all that is required for R1 to reach the Internet is a default static route going out of the Serial 0/0/1 interface.



Similar default static routes could be configured on R2 and R3, but it is much more scalable to enter it one time on the edge router R1 and then have R1 propagate it to all other routers using RIP. To provide Internet connectivity to all other networks in the RIP routing domain, the default static route needs to be advertised to all other routers that use the dynamic routing protocol.

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

- A default static route using the **ip route 0.0.0.0 0.0.0.0** command.

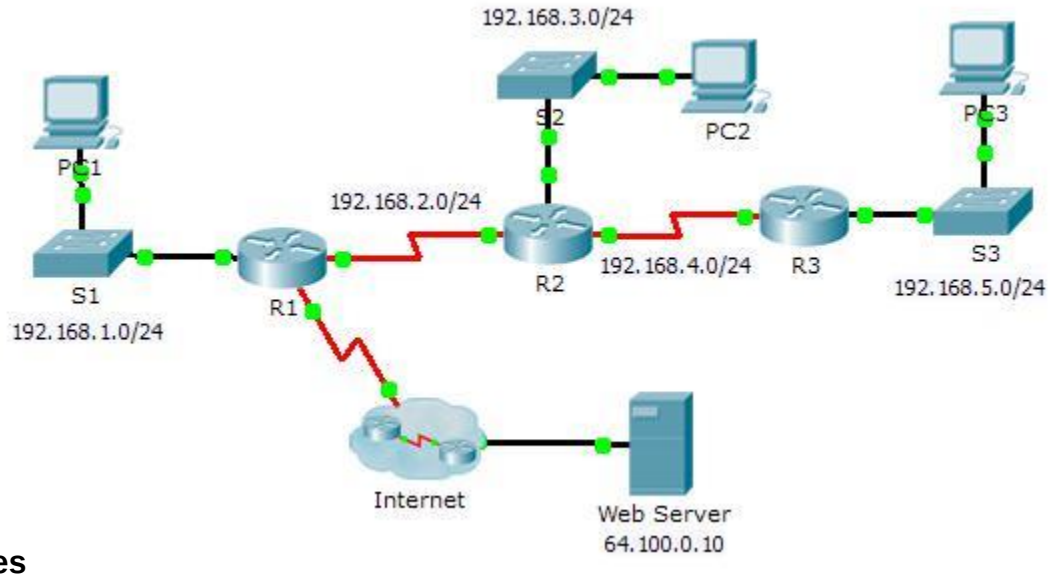
- The **default-information originate** router configuration command. This instructs R1 to originate default information, by propagating the static default route in RIP updates.

The example in Figure configures a fully-specified default static route to the service provider and then the route is propagated by RIP. Notice that R1 now has a Gateway of Last Resort and default route installed in its routing table.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 S0/0/1 209.165.200.226
R1(config)# router rip
R1(config-router)# default-information originate
R1(config-router)# ^Z
R1#
*Mar 10 23:33:51.801: %SYS-5-CONFIG_I: Configured from console by console
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.226 to network 0.0.0.0

S*  0.0.0.0/0 [1/0] via 209.165.200.226, Serial0/0/1
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
    192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, Serial0/0/0
L       192.168.2.1/32 is directly connected, Serial0/0/0
R   192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0
R   192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0
R   192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:08, Serial0/0/0
    209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C       209.165.200.0/24 is directly connected, Serial0/0/1
L       209.165.200.225/27 is directly connected, Serial0/0/1
```

Packet Tracer – Configuring RIPv2

Topology



Objectives

Part 1: Configure RIPv2

Part 2: Verify Configurations

Background

Although RIP is rarely used in modern networks, it is useful as a foundation for understanding basic network routing. In this activity, you will configure a default route, RIP version 2, with appropriate network statements and passive interfaces, and verify full connectivity.

Part 1: Configure RIPv2

Step 1: Configure RIPv2 on R1.

a. Use the appropriate command to create a default route on **R1** for all Internet traffic to exit the network through S0/0/1.

b. Enter RIP protocol configuration mode.

c. Use version 2 of the RIP protocol and disable the summarization of networks.

d. Configure RIP for the networks that connect to **R1**.

e. Configure the LAN port that contains no routers so that it does not send out any routing information.

f. Advertise the default route configured in step 1a with other RIP routers.

g. Save the configuration.

Step 2: Configure RIPv2 on R2.

a. Enter RIP protocol configuration mode.
b. Use version 2 of the RIP protocol and disable the summarization of networks.
c. Configure RIP for the networks directly connected to **R2**.
d. Configure the interface that contains no routers so that it does not send out routing information.
e. Save the configuration.

Step 3: Configure RIPv2 on R3

Repeat Step 2 on **R3**.

Part 2: Verify Configurations

Step 1: View routing tables of R1, R2, and R3.

a. Use the appropriate command to show the routing table of **R1**. RIP (R) now appears with connected (C) and local (L) routes in the routing table. All networks have an entry. You also see a default route listed.

b. View the routing tables for **R2** and **R3**. Notice that each router has a full listing of all the 192.168.x.0 networks and a default route.

Step 2: Verify full connectivity to all destinations.

Every device should now be able to ping every other device inside the network. In addition, all devices should be able to ping the **Web Server**.

Lab 2: Enhanced Interior Gateway Routing Protocol (EIGRP) part 1

Distance Vector Algorithm

At the core of the distance vector protocol is the routing algorithm. The algorithm is used to calculate the best paths and then send that information to the neighbors.

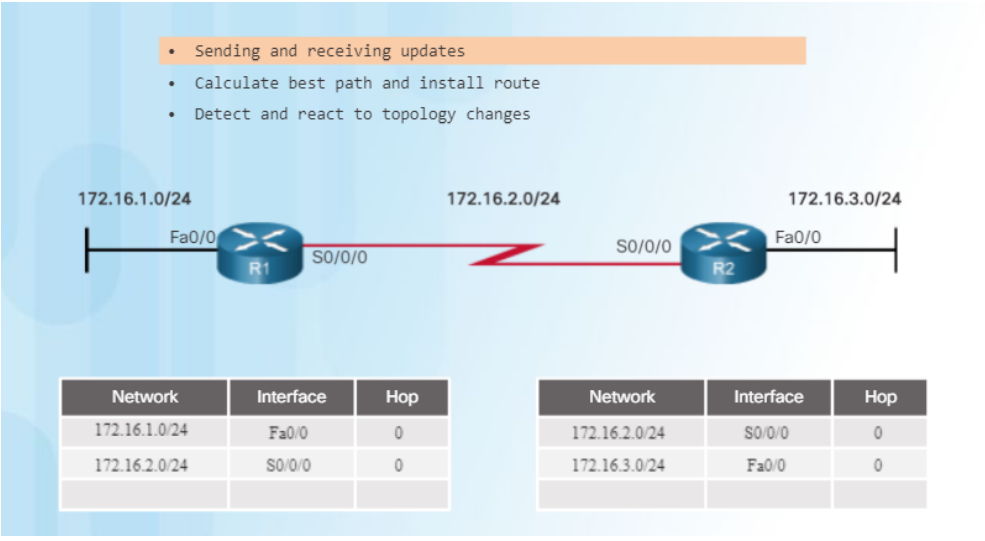
The algorithm used for the routing protocols defines the following processes:

- Mechanism for sending and receiving routing information
- Mechanism for calculating the best paths and installing routes in the routing table
- Mechanism for detecting and reacting to topology changes

In the figure animation, R1 and R2 are configured with the RIP routing protocol. The algorithm sends and receives updates. Both R1 and R2 then glean new information from the update. In this case, each router learns about a new network. The algorithm on each router makes its calculations independently and updates the routing table with the new information. When the LAN on R2 goes down, the algorithm constructs a triggered update and sends it to R1. R1 then removes the network from the routing table.

Different routing protocols use different algorithms to install routes in the routing table, send updates to neighbors, and make path determination decisions. For example:

- RIP uses the Bellman-Ford algorithm as its routing algorithm. It is based on two algorithms developed in 1958 and 1956 by Richard Bellman and Lester Ford, Jr.
- IGRP and EIGRP use the Diffusing Update Algorithm (DUAL) routing algorithm developed by Dr. J.J. Garcia-Luna-Aceves at SRI International.



Enhanced Interior-Gateway Routing Protocol

The Interior Gateway Routing Protocol (IGRP) was the first proprietary IPv4 routing protocol developed by Cisco in 1984. It used the following design characteristics:

- Bandwidth, delay, load, and reliability are used to create a composite metric.
- Routing updates are broadcast every 90 seconds, by default.
- Maximum limit of 255 hops

In 1992, IGRP was replaced by Enhanced IGRP (EIGRP). Like RIPv2, EIGRP also introduced support for VLSM and CIDR. EIGRP increases efficiency, reduces routing updates, and supports secure message exchange.

The table in the figure summarizes the differences between IGRP and EIGRP.

EIGRP also introduced:

- **Bounded triggered updates** - It does not send periodic updates. Only routing table changes are propagated, whenever a change occurs. This reduces the amount of load the routing protocol places on the network. Bounded triggered updates means that EIGRP only sends to the neighbors that need it. It uses less bandwidth, especially in large networks with many routes.
- **Hello keepalive mechanism** - A small Hello message is periodically exchanged to maintain adjacencies with neighboring routers. This requires a very low usage of network resources during normal operation, as compared to periodic updates.
- **Maintains a topology table** - Maintains all the routes received from neighbors (not only the best paths) in a topology table. DUAL can insert backup routes into the EIGRP topology table.
- **Rapid convergence** - In most cases, it is the fastest IGP to converge because it maintains alternate routes, enabling almost instantaneous convergence. If a primary route fails, the router can use the already identified alternate route. The switchover to the alternate route is immediate and does not involve interaction with other routers.
- **Multiple network layer protocol support** - EIGRP uses Protocol Dependent Modules (PDM), which means that it is the only protocol to include support for protocols other than IPv4 and IPv6, such as legacy IPX and AppleTalk.

Features of EIGRP

EIGRP was initially released in 1992 as a proprietary protocol available only on Cisco devices. However, in 2013, Cisco released a basic functionality of EIGRP as an open standard to the IETF, as an informational RFC. This means that other networking vendors can now implement EIGRP on their equipment to interoperate with both Cisco and non-Cisco routers running EIGRP. However, advanced features of EIGRP, such as EIGRP stub, needed for the Dynamic Multipoint Virtual Private Network (DMVPN) deployment, will not be released to the IETF. As an informational RFC, Cisco will continue to maintain control of EIGRP.

EIGRP includes features of both link-state and distance vector routing protocols. However, EIGRP is still based on the key distance vector routing protocol principle, in which information about the rest of the network is learned from directly connected neighbors.

EIGRP is an advanced distance vector routing protocol that includes features not found in other distance vector routing protocols like RIP and IGRP.

In Cisco IOS Release 15.0(1)M, Cisco introduced a new EIGRP configuration option called **named EIGRP**. Named EIGRP enables the configuration of EIGRP for both IPv4 and IPv6 under a single configuration mode. This helps eliminate configuration complexity that occurs when configuring EIGRP for both IPv4 and IPv6. Named EIGRP is beyond the scope of this course.

Features of EIGRP include:

- **Diffusing Update Algorithm -** As the computational engine that drives EIGRP, the Diffusing Update Algorithm (DUAL) resides at the center of the routing protocol. DUAL guarantees loop-free and backup paths throughout the routing domain. Using DUAL, EIGRP stores all available backup routes for destinations so that it can quickly adapt to alternate routes when necessary.

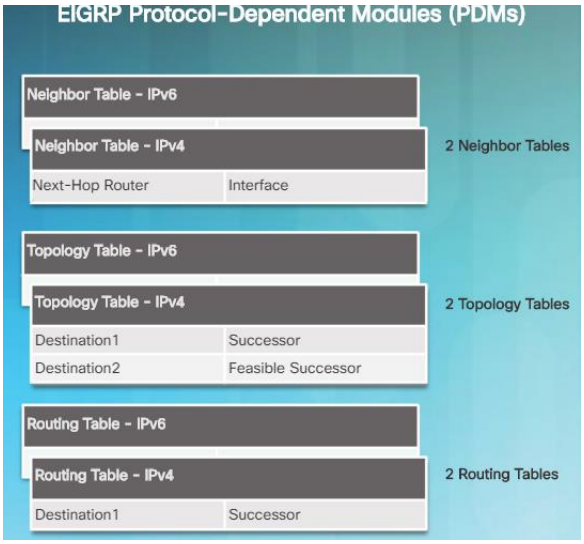
- **Establishing Neighbor Adjacencies -** EIGRP establishes relationships with directly connected routers that are also enabled for EIGRP. Neighbor adjacencies are used to track the status of these neighbors.

- **Reliable Transport Protocol -** The Reliable Transport Protocol (RTP) is unique to EIGRP and provides delivery of EIGRP packets to neighbors. RTP and the tracking of neighbor adjacencies set the stage for DUAL.

- **Partial and Bounded Updates -** EIGRP uses the terms partial and bounded when referring to its updates. Unlike RIP, EIGRP does not send periodic updates and route entries do not age out. The term partial means that the update only includes information about the route changes, such as a new link or a link becoming unavailable. The term bounded refers to the propagation of partial updates that are sent only to those routers that the changes affect. This minimizes the bandwidth that is required to send EIGRP updates.

- **Equal and Unequal Cost Load Balancing -** EIGRP supports equal cost load balancing and unequal cost load balancing, which allows administrators to better distribute traffic flow in their networks.

Note: The term “hybrid routing” protocol may be used in some older documentation to define EIGRP. However, this term is misleading because EIGRP is not a hybrid between distance vector and link-state routing protocols. EIGRP is solely a distance vector routing protocol; therefore, Cisco no longer uses this term to refer to it.



Reliable Transport Protocol

EIGRP was designed as a network layer independent routing protocol. Because of this design, EIGRP cannot use the services of UDP or TCP. Instead, EIGRP uses the Reliable Transport Protocol (RTP) for the delivery and reception of EIGRP packets. This allows EIGRP to be flexible and can be used for protocols other than those from the TCP/IP protocol suite, such as the now obsolete IPX and AppleTalk protocols.

The figure conceptually shows how RTP operates.

Although “reliable” is part of its name, RTP includes both reliable delivery and unreliable delivery of EIGRP packets, similar to TCP and UDP, respectively. Reliable RTP requires an acknowledgment to be returned by the receiver to the sender. An unreliable RTP packet does not require an acknowledgment. For example, an EIGRP update packet is sent reliably over RTP and requires an acknowledgment. An EIGRP Hello packet is also sent over RTP, but unreliably. This means that EIGRP Hello packets do not require an acknowledgment.

RTP can send EIGRP packets as unicast or multicast.

- Multicast EIGRP packets for IPv4 use the reserved IPv4 multicast address 224.0.0.10.
- Multicast EIGRP packets for IPv6 are sent to the reserved IPv6 multicast address FF02::A.

Authentication

Like other routing protocols, EIGRP can be configured for authentication. RIPv2, EIGRP, OSPF, IS-IS, and BGP can each be configured to authenticate their routing information.

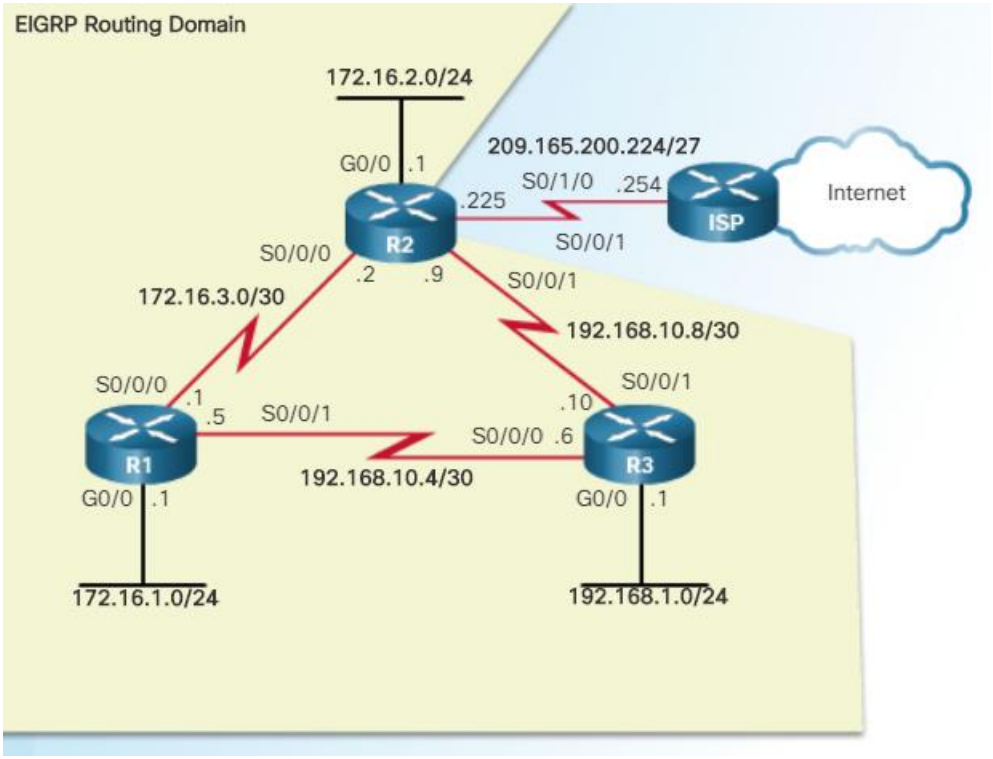
It is a good practice to authenticate transmitted routing information. Doing so ensures that routers only accept routing information from other routers that have been configured with the same password or authentication information.

Note: Authentication does not encrypt the EIGRP routing updates.

EIGRP Network Topology

Figure displays the topology EIGRP for IPv4.

The routers in the topology have a starting configuration that includes addresses on the interfaces. There is currently no static routing or dynamic routing configured on any of the routers.




```
R1(config)# router eigrp 1
R1(config-router)#
```

EIGRP Router ID

The EIGRP router ID is used to uniquely identify each router in the EIGRP routing domain.

The router ID is used in both EIGRP and OSPF routing protocols. However, the role of the router ID is more significant in OSPF. In EIGRP IPv4 implementations, the use of the router ID is not that apparent. EIGRP for IPv4 uses the 32-bit router ID to identify the originating router for redistribution of external routes. The need for a router ID becomes more evident in the discussion of EIGRP for IPv6. While the router ID is necessary for redistribution, the details of EIGRP redistribution are beyond the scope of this curriculum. For purposes of this curriculum, it is only necessary to understand what the router ID is and how it is determined.

To determine its router ID, a Cisco IOS router will use the following three criteria in order:

1. Use the address configured with the **eigrp router-id** *ipv4-address* router configuration mode command.

2. If the router ID is not configured, choose the highest IPv4 address of any of its loopback interfaces.

3. If no loopback interfaces are configured, choose the highest active IPv4 address of any of its physical interfaces.

If the network administrator does not explicitly configure a router ID using the **eigrp router-id** command, EIGRP generates its own router ID using either a loopback or physical IPv4 address. A loopback address is a virtual interface and is automatically in the up state when configured. The interface does not need to be enabled for EIGRP, meaning that it does not need to be included in one of the EIGRP **network** commands. However, the interface must be in the up/up state.

Using the criteria described above, the figure shows the default EIGRP router IDs that are determined by the routers' highest active IPv4 address.

Note: The **eigrp router-id** command is used to configure the router ID for EIGRP. Some versions of IOS will accept the command**router-id**, without first specifying **eigrp**. The running-config, however, will display **eigrp router-id** regardless of which command is used.

Configuring the EIGRP Router ID

The **eigrp router-id** *ipv4-address* router configuration command is the preferred method used to configure the EIGRP router ID. This method takes precedence over any configured loopback or physical interface IPv4 addresses.

Note: The IPv4 address used to indicate the router ID is actually any 32-bit number displayed in dotted-decimal notation.

The *ipv4-address* router ID can be configured with any IPv4 address except 0.0.0.0 and 255.255.255.255. The router ID should be a unique 32-bit number in the EIGRP routing domain; otherwise, routing inconsistencies can occur.

Figure shows the configuration of the EIGRP router ID for routers R1

```
R1(config)# router eigrp 1
R1(config-router)# eigrp router-id 1.1.1.1
R1(config-router)#
```

If a router ID is not explicitly configured, then the router would use its highest IPv4 address configured on a loopback interface. The advantage of using a loopback interface is that unlike physical interfaces, loopbacks cannot fail. There are no actual cables or adjacent devices on which the loopback interface depends for being in the up state. Therefore, using a loopback address for the router ID can provide a more consistent router ID than using an interface address.

If the **eigrp router-id** command is not used and loopback interfaces are configured, EIGRP chooses the highest IPv4 address of any of its loopback interfaces. The following commands are used to enable and configure a loopback interface:

```
Router(config)# interface loopback number

Router(config-if)# ip address ipv4-address subnet-mask
```

Verifying the EIGRP Process

Figure 2 shows the **show ip protocols** output for R1, including its router ID. The **show ip protocols** command displays the parameters and current state of any active routing protocol processes, including both EIGRP and OSPF. The**show ip protocols** command displays different types of output specific to each routing protocol.

The network Command

EIGRP router configuration mode allows for the configuration of the EIGRP routing protocol.. To enable EIGRP routing on an interface, use the **network** *ipv4-network-address* router configuration mode command. The *ipv4-network-address* is the classful network address for each directly connected network.

The **network** command has the same function as in all IGP routing protocols. The **network** command in EIGRP:

- Enables any interface on this router that matches the network address in the **network** router configuration mode command to send and receive EIGRP updates.
- The network of the interfaces is included in EIGRP routing updates.

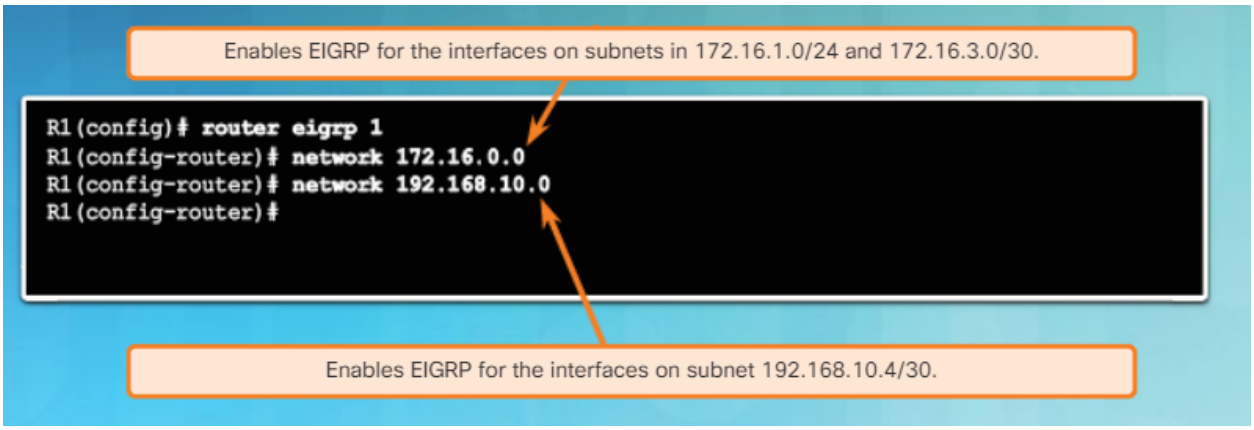


Figure shows the **network** commands required to configure EIGRP on R1. In the figure, a single classful **network** statement,**network 172.16.0.0**, is used on R1 to include both interfaces in subnets 172.16.1.0/24 and 172.16.3.0/30. Notice that only the classful network address is used.

```
R2(config)# router eigrp 1
R2(config-router)# network 172.16.0.0
R2(config-router)#
*Feb 28 17:51:42.543: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1:
Neighbor 172.16.3.1 (Serial0/0/0) is up: new adjacency
R2(config-router)#
```

Figure 2 shows the **network** command used to enable EIGRP on R2's interfaces for subnets 172.16.1.0/24 and 172.16.2.0/24. When EIGRP is configured on R2's S0/0/0 interface, DUAL sends a notification message to the console stating that a neighbor adjacency with another EIGRP router on that interface has been established. This new adjacency happens automatically because both R1 and R2 use the same autonomous system number (i.e., **1**), and both routers now send updates on their interfaces in the 172.16.0.0 network.

DUAL automatically generates the notification message because the **eigrp log-neighbor-changes** router configuration mode command is enabled by default. Specifically, the command helps verify neighbor adjacencies during configuration of EIGRP and displays any changes in EIGRP neighbor adjacencies, such as when an EIGRP adjacency has been added or removed.

The network Command and Wildcard Mask

By default, when using the **network** command and an IPv4 network address, such as 172.16.0.0, all interfaces on the router that belong to that classful network address are enabled for EIGRP. However, there may be times when the network administrator does not want to include all interfaces within a network when enabling EIGRP. For example, in Figure 1, assume that an administrator wants to enable EIGRP on R2, but only for the subnet 192.168.10.8 255.255.255.252, on the S0/0/1 interface.

To configure EIGRP to advertise specific subnets only, use the *wildcard-mask* option with the network command:

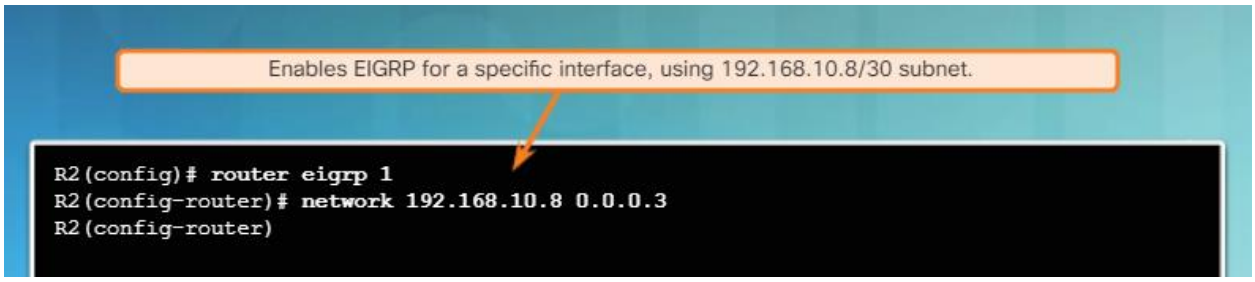
```
Router(config-router)# network network-address [wildcard-mask]
```

A wildcard mask is similar to the inverse of a subnet mask. In a subnet mask, binary 1s are significant while binary 0s are not. In a wildcard mask, binary 0s are significant, while binary 1s are not. For example, the inverse of subnet mask 255.255.255.252 is 0.0.0.3.

Calculating a wildcard mask may seem daunting at first but it's actually pretty easy to do. To calculate the inverse of the subnet mask, subtract the subnet mask from 255.255.255.255 as follows:

```
  255.255.255.255
- 255.255.255.252
  ---------------
    0.   0.   0.   3    Wildcard mask
```

Figure continues the EIGRP network configuration of R2. The **network 192.168.10.8 0.0.0.3** command specifically enables EIGRP on the S0/0/1 interface, a member of the 192.168.10.8 255.255.255.252 subnet.



Configuring a wildcard mask is the official command syntax of the EIGRP **network**command. However, the Cisco IOS versions also accepts a subnet mask to be used instead. For example, Figure configures the same S0/0/1 interface on R2, but this time using a subnet mask in the **network** command. Notice in the output of the **show running-config** command, the IOS converted the subnet mask command to its wildcard mask.

```
R2(config)# router eigrp 1
R2(config-router)# network 192.168.10.8 255.255.255.252
R2(config-router)# end
R2# show running-config | section eigrp 1
router eigrp 1
 network 172.16.0.0
 network 192.168.10.8 0.0.0.3
 eigrp router-id 2.2.2.2
R2#
```

Passive Interface

As soon as a new interface is enabled within the EIGRP network, EIGRP attempts to form a neighbor adjacency with any neighboring routers to send and receive EIGRP updates.

At times it may be necessary, or advantageous, to include a directly connected network in the EIGRP routing update, but not allow any neighbor adjacencies off of that interface to form. The **passive-interface** command can be used to prevent the neighbor adjacencies. There are two primary reasons for enabling the**passive-interface** command:

- To suppress unnecessary update traffic, such as when an interface is a LAN interface, with no other routers connected
- To increase security controls, such as preventing unknown rogue routing devices from receiving EIGRP updates

R1, R2, and R3 do not have neighbors on their GigabitEthernet 0/0 interfaces.

The **passive-interface** router configuration mode command disables the transmission and receipt of EIGRP Hello packets on these interfaces.

```
Router(config)# router eigrp as-number

Router(config-router)# passive-interface interface-type interface-number
```

Figure 2 shows the **passive-interface** command configured to suppress Hello packets on the LANs for R1 and R3.

Without a neighbor adjacency, EIGRP cannot exchange routes with a neighbor. Therefore, the **passive-interface** command prevents the exchange of routes on the interface. Although EIGRP does not send or receive routing updates on an interface configured with the **passive-interface** command, it still includes the address of the interface in routing updates sent out of other non-passive interfaces.

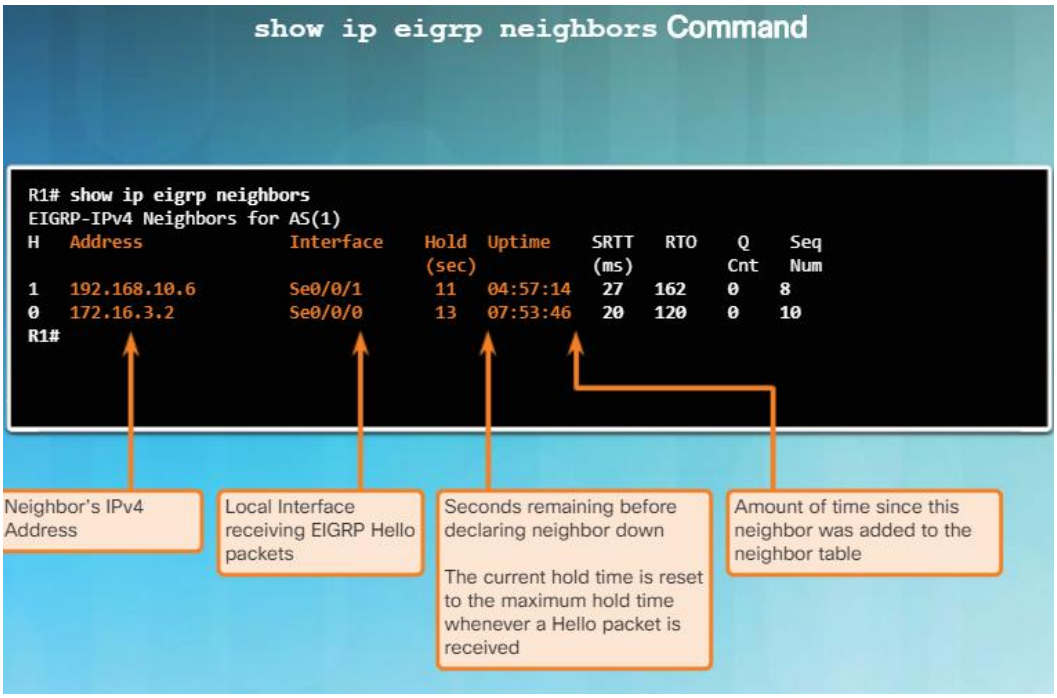
Note: To configure all interfaces as passive, use the **passive-interface default** command. To disable an interface as passive, use the **no passive-interface** *interface-type interface-number* command.

An example of using the passive interface to increase security controls is when a network must connect to a third-party organization, for which the local administrator has no control, such as when connecting to an ISP network. In this case, the local network administrator would need to advertise the interface link through their own network, but would not want the third-party organization to receive or send routing updates to the local routing device, as this is a security risk.

Verifying the Passive Interface

To verify whether any interface on a router is configured as passive, use the **show ip protocols** privileged EXEC mode command, as shown in Figure 3. Notice that although R3's GigabitEthernet 0/0 interface is a passive interface, EIGRP still includes the interface's network address of 192.168.1.0 in its routing updates.

Verifying EIGRP: Examining Neighbors



Verifying EIGRP: show ip protocols Command

The **show ip protocols** command is useful to identify the parameters and other information about the current state of any active IPv4 routing protocol processes configured on the router. The **show ip protocols** command displays different types of output specific to each routing protocol.
The output in Figure indicates several EIGRP parameters, including:
1. EIGRP is an active dynamic routing protocol on R1 configured with the autonomous system number 1.
2. The EIGRP router ID of R1 is 1.1.1.1.
3. The EIGRP administrative distances on R1 are internal AD of 90 and external of 170 (default values).
4. By default, EIGRP does not automatically summarize networks. Subnets are included in the routing updates.
5. The EIGRP neighbor adjacencies R1 has with other routers used to receive EIGRP routing updates.

Note: Prior to IOS 15, EIGRP automatic summarization was enabled by default.
The output from the **show ip protocols** command is useful in debugging routing operations. Information in the Routing Information Sources field can help identify a router suspected of delivering bad routing information. The field lists all the EIGRP routing sources the Cisco IOS software uses to build its IPv4 routing table. For each source, note the following:

- IPv4 address
- Administrative distance
- Time the last update was received from this source

Default Administrative Distances

Route Source	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal BGP	200

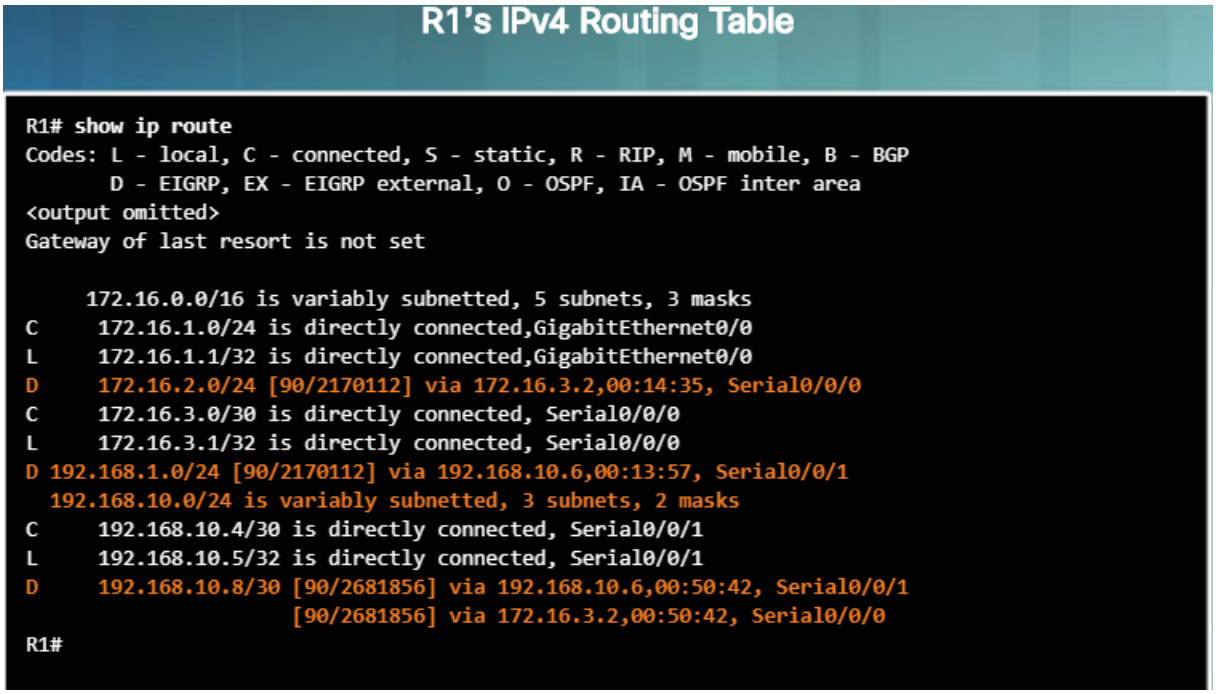
As shown in Figure , EIGRP has a default AD of 90 for internal routes and 170 for routes imported from an external source, such as default routes. When compared to other IGPs, EIGRP is the most preferred by the Cisco IOS, because it has the lowest administrative distance. EIGRP has a third AD value of 5, for summary routes.

Verifying EIGRP: Examine the IPv4 routing table

Another way to verify that EIGRP and other functions of the router are configured properly is to examine the IPv4 routing tables with theshow ip route command. As with any dynamic routing protocol, the network administrator must verify the information in the routing table to ensure that it is populated as expected, based on configurations entered. For this reason, it is important to have a good understanding of the routing protocol configuration commands, as well as the routing protocol operations and the processes used by the routing protocol to build the IP routing table.

Notice that the outputs used throughout this course are from Cisco IOS 15. Prior to IOS 15, EIGRP automatic summarization was enabled by default. The state of automatic summarization can make a difference in the information displayed in the IPv4 routing table. If a previous version of the IOS is used, automatic summarization can be disabled using the no auto-summary router configuration mode command:

```
Router(config-router)# no auto-summary
```



In Figure, the IPv4 routing table is examined using the **show ip route** command. EIGRP routes are denoted in the routing table with a **D**. The letter D was used to represent EIGRP because the protocol is based upon the DUAL algorithm.

The **show ip route** command verifies that routes received by EIGRP neighbors are installed in the IPv4 routing table. The **show ip route** command displays the entire routing table, including remote networks learned dynamically, directly connected and static routes. For this reason, it is normally the first command used to check for convergence. After routing is correctly configured on all routers, the **show ip route** command reflects that each router has a full routing table, with a route to each network in the topology.

Notice that R1 has installed routes to three IPv4 remote networks in its IPv4 routing table:

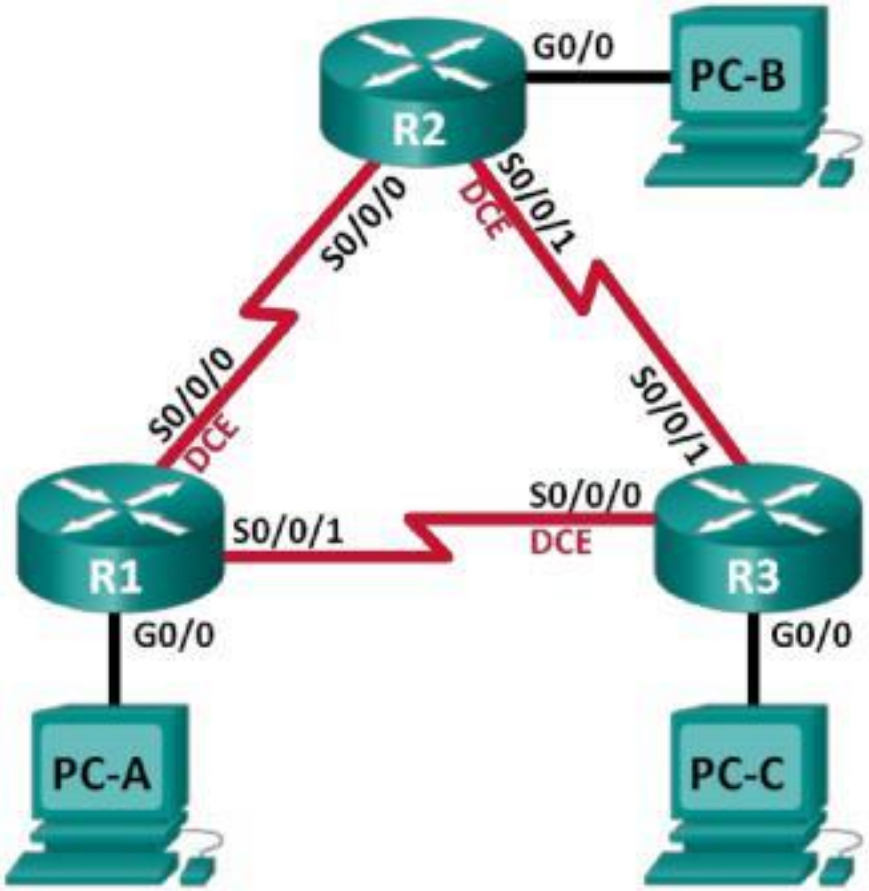
- 172.16.2.0/24 network, received from router R2 on the Serial0/0/0 interface
- 192.168.1.0/24 network, received from router R2 on the Serial0/0/1 interface

- 192.168.10.8/30 network, received from both R2 on the Serial0/0/0 interface, and from R3 on the Serial0/0/1 interface

R1 has two paths to the 192.168.10.8/30 network, because its cost or metric to reach that network is the same or equal using both routers. These are known as equal cost routes. R1 uses both paths to reach this network, which is known as load balancing. The EIGRP metric is discussed later in this chapter.

Lab – Configuring Basic EIGRP for IPv4

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A
	S0/0/1	10.3.3.1	255.255.255.252	N/A
R2	G0/0	192.168.2.1	255.255.255.0	N/A
	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A
R3	G0/0	192.168.3.1	255.255.255.0	N/A
	S0/0/0 (DCE)	10.3.3.2	255.255.255.252	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

Objectives

Part 1: Build the Network and Verify Connectivity

Part 2: Configure EIGRP Routing

Part 3: Verify EIGRP Routing

Part 4: Configure Bandwidth and Passive Interfaces

Background / Scenario

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

In this lab, you will configure EIGRP for the topology and networks shown above. You will modify bandwidth and configure passive interfaces to allow EIGRP to function more efficiently.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Note: Make sure that the routers have been erased and have no startup configurations. If you are unsure, contact your instructor.

Required Resources

3 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable) 3

PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)

Console cables to configure the Cisco IOS devices via the console

ports Ethernet and serial cables as shown in the topology

Part 1: Build the Network and Verify Connectivity

In Part 1, you will set up the network topology and configure basic settings, such as the interface IP addresses, device access, and passwords.

Step 1: Cable the network as shown in the topology.

Step 2: Configure PC hosts.

Step 3: Initialize and reload the routers as necessary.

Step 4: Configure basic settings for each router.

a. Disable DNS lookup.

b. Configure IP addresses for the routers, as listed in the Addressing Table.

c. Configure device name as shown in the topology.

d. Assign **cisco** as the console and vty passwords.

e. Assign **class** as the privileged EXEC password.

f. Configure **logging synchronous** to prevent console and vty messages from interrupting command entry.

g. Configure a message of the day.

h. Copy the running configuration to the startup configuration.

Step 5: Verify connectivity.

The routers should be able to ping one another, and each PC should be able to ping its default gateway. The PCs will not be able to ping other PCs until EIGRP routing is configured. Verify and troubleshoot if necessary.

Part 2: Configure EIGRP Routing

Step 1: Enable EIGRP routing on R1. Use AS number 10.

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

Step 2: Advertise the directly connected networks on R1 using the wildcard mask.

```
R1(config-router)# network 10.1.1.0 0.0.0.3
R1(config-router)# network 192.168.1.0 0.0.0.255
R1(config-router)# network 10.3.3.0 0.0.0.3
```

Why is it a good practice to use wildcard masks when advertising networks? Could the mask have been omitted from any of the network statements above? If so, which one(s)?

Step 3: Enable EIGRP routing and advertise the directly connected networks on R2 and R3.

You will see neighbor adjacency messages as interfaces are added to the EIGRP routing process. The messages on R2 are displayed as an example.

```
*Apr 14 15:24:59.543: %DUAL-5-NBRCHANGE: EIGRP-IPv4 10: Neighbor 10.1.1.1
(Serial0/0/0) is up: new adjacency
```

Step 4: Verify end-to-end connectivity.

All devices should be able to ping each other if EIGRP is configured correctly.

Note: Depending on the operating system, it may be necessary to disable the firewall for the pings to the host PCs to be successful.

Part 3: Verify EIGRP Routing

Step 1: Examine the EIGRP neighbor table.

On R1, issue the **show ip eigrp neighbors** command to verify that the adjacency has been established with its neighboring routers.

```
R1# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(10)
H   Address                 Interface              HoldUptime    SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
1   10.3.3.2                Se0/0/1                  1300:24:58     8   100  0  17
0   10.1.1.2                Se0/0/0                  1300:29:23     7   100  0  23
```

Step 2: Examine the IP EIGRP routing table.

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

override Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D        10.2.2.0/30 [90/2681856] via 10.3.3.2, 00:29:01, Serial0/0/1
                     [90/2681856] via 10.1.1.2, 00:29:01, Serial0/0/0
D     192.168.2.0/24 [90/2172416] via 10.1.1.2, 00:29:01, Serial0/0/0
D192.168.3.0/24 [90/2172416] via 10.3.3.2, 00:27:56, Serial0/0/1
```

Why does R1 have two paths to the 10.2.2.0/30 network?

Step 3: Examine the EIGRP topology table.

```
R1# show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(10)/ID(192.168.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 192.168.3.0/24, 1 successors, FD is 2172416
        via 10.3.3.2 (2172416/28160), Serial0/0/1
P 192.168.2.0/24, 1 successors, FD is 2172416
        via 10.1.1.2 (2172416/28160), Serial0/0/0
P 10.2.2.0/30, 2 successors, FD is 2681856
        via 10.1.1.2 (2681856/2169856), Serial0/0/0
        via 10.3.3.2 (2681856/2169856), Serial0/0/1
P 10.3.3.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/0/1
P 192.168.1.0/24, 1 successors, FD is 2816
        via Connected, GigabitEthernet0/0
P 10.1.1.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/0/0
```

Why are there no feasible successors listed in the R1 topology table?

Step 4: Verify the EIGRP routing parameters and networks advertised.

Issue the **show ip protocols** command to verify the EIGRP routing parameters used.

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 10"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(10)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 192.168.1.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.1.1.0/30
```

```
    10.3.3.0/30
    192.168.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.3.3.2              90      02:38:34
    10.1.1.2              90      02:38:34
  Distance: internal 90 external 170
```

Based on the output of issuing the **show ip protocols** command, answer the following questions.

What AS number is used?

What networks are advertised?

What is the administrative distance for EIGRP?

How many equal cost paths does EIGRP use by default?

Part 4: Configure Bandwidth and Passive Interfaces

EIGRP uses a default bandwidth based on the type of interface in the router. In Part 4, you will modify the bandwidth so that the link between R1 and R3 has a lower bandwidth than the link between R1/R2 and R2/R3. In addition, you will set passive interfaces on each router.

Step 1: Observe the current routing settings.

a. Issue the **show interface s0/0/0** command on R1.

```
R1# show interface s0/0/0
Serial0/0/0 is up, line protocol is
  up Hardware is WIC MBRD Serial
  Internet address is 10.1.1.1/30
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not
  set Keepalive set (10 sec)
  Last input 00:00:01, output 00:00:02, output hang never
  Last clearing of "show interface" counters 03:43:45
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4050 packets input, 270294 bytes, 0 no buffer
     Received 1554 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     1 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 1
     abort 4044 packets output, 271278 bytes, 0 underruns
     0 output errors, 0 collisions, 5 interface resets
     4 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     12 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

What is the default bandwidth for this serial interface?

b. How many routes are listed in the routing table to reach the 10.2.2.0/30 network?

Step 2: Modify the bandwidth on the routers.

a. Modify the bandwidth on R1 for the serial interfaces.

```
R1(config)# interface s0/0/0
R1(config-if)# bandwidth 2000
R1(config-if)# interface s0/0/1
R1(config-if)# bandwidth 64
```

Issue **show ip route** command on R1. Is there a difference in the routing table? If so, what is it?

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

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.1.1.0/30 is directly connected, Serial0/0/0
L        10.1.1.1/32 is directly connected, Serial0/0/0
D        10.2.2.0/30 [90/2681856] via 10.1.1.2, 00:03:09, Serial0/0/0
C        10.3.3.0/30 is directly connected, Serial0/0/1
L        10.3.3.1/32 is directly connected, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
D     192.168.2.0/24 [90/1794560] via 10.1.1.2, 00:03:09, Serial0/0/0
D     192.168.3.0/24 [90/2684416] via 10.1.1.2, 00:03:08, Serial0/0/0
```

b. Modify the bandwidth on the R2 and R3 serial interfaces.

```
R2(config)# interface s0/0/0
R2(config-if)# bandwidth 2000
R2(config-if)# interface s0/0/1
R2(config-if)# bandwidth 2000

R3(config)# interface s0/0/0
R3(config-if)# bandwidth 64
R3(config-if)# interface s0/0/1
R3(config-if)# bandwidth 2000
```

Step 3: Verify the bandwidth modifications.

a. Verify bandwidth modifications. Issue a **show interface serial 0/0/x** command, with x being the appropriate serial interface on all three routers to verify that bandwidth is set correctly. R1 is shown as an example.

```
R1# show interface s0/0/0
Serial0/0/0 is up, line protocol is
  up Hardware is WIC MBRD Serial
  Internet address is 10.1.1.1/30
  MTU 1500 bytes, BW 2000 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not
  set Keepalive set (10 sec)
  Last input 00:00:01, output 00:00:02, output hang never
  Last clearing of "show interface" counters 04:06:06
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4767 packets input, 317155 bytes, 0 no buffer
     Received 1713 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     1 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 1
     abort 4825 packets output, 316451 bytes, 0 underruns
     0 output errors, 0 collisions, 5 interface resets
     4 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     12 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Based on your bandwidth configuration, try and determine what the R2 and R3 routing tables will look like before you issue a **show ip route** command. Are their routing tables the same or different?

Step 4: Configure G0/0 interface as passive on R1, R2, and R3.

A passive interface does not allow outgoing and incoming routing updates over the configured interface. The **passive-interface** *interface* command causes the router to stop sending and receiving Hello packets over an interface; however, the network associated with the interface is still advertised to other routers through the non-passive interfaces. Router interfaces connected to LANs are typically configured as passive.

```
R1(config)# router eigrp 10
R1(config-router)# passive-interface g0/0

R2(config)# router eigrp 10
R2(config-router)# passive-interface g0/0
```

```
3(config)# router eigrp 10
R3(config-router)# passive-interface g0/0
```

Step 5: Verify the passive interface configuration.

Issue a **show ip protocols** command on R1, R2, and R3 and verify that G0/0 has been configured as passive.

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 10"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(10)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 192.168.1.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.1.1.0/30
    10.3.3.0/30
    192.168.1.0
  Passive Interface(s):
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.3.3.2              90      00:48:09
    10.1.1.2              90      00:48:26
  Distance: internal 90  external 170
```

EIGRP Composite Metric

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

- **Bandwidth** - The slowest bandwidth among all of the outgoing interfaces, along the path from source to destination.
- **Delay** - The cumulative (sum) of all interface delay along the path (in tens of microseconds).

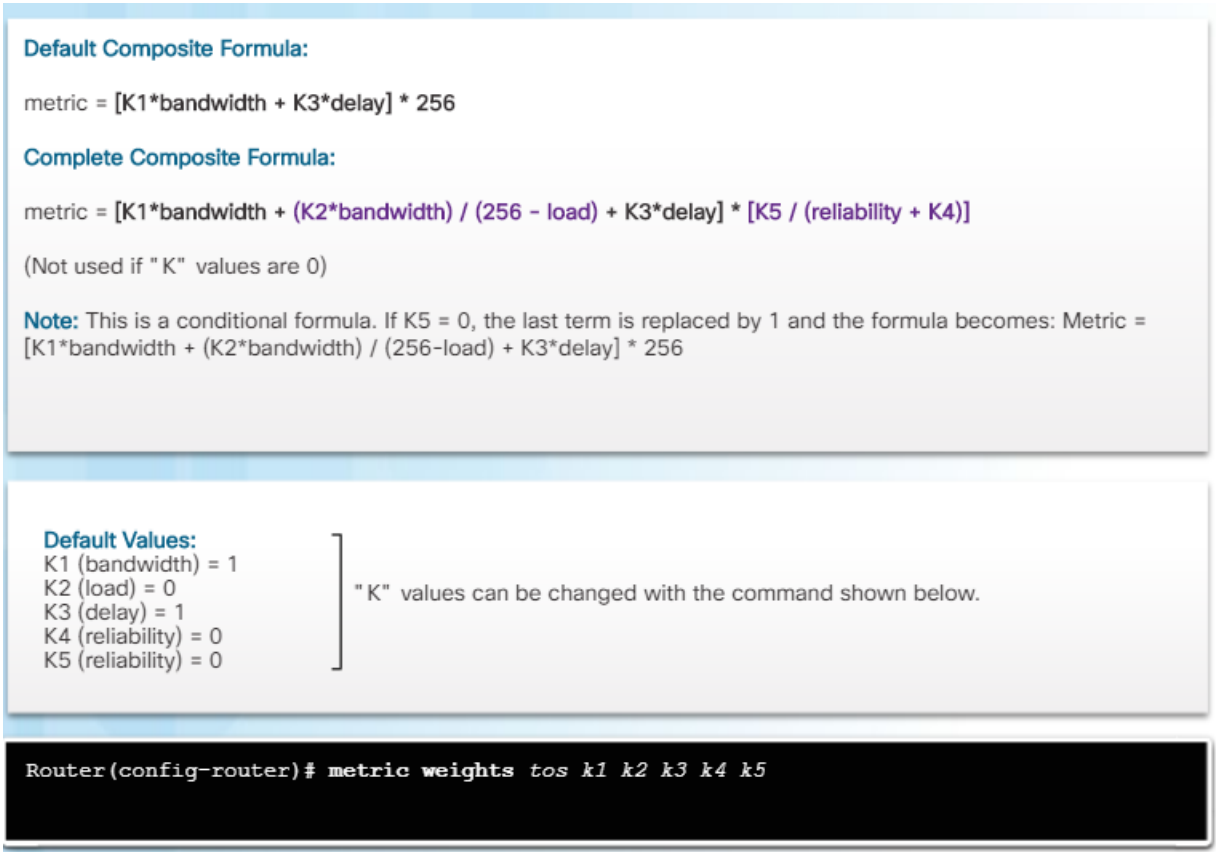
The following values can be used, but are not recommended, because they typically result in frequent recalculation of the topology table:

- **Reliability** - Represents the worst reliability between the source and destination, which is based on keepalives.
- **Load** - Represents the worst load on a link between the source and destination, which is computed based on the packet rate and the configured bandwidth of the interface.

Note: Although the MTU is included in the routing table updates, it is not a routing metric used by EIGRP.

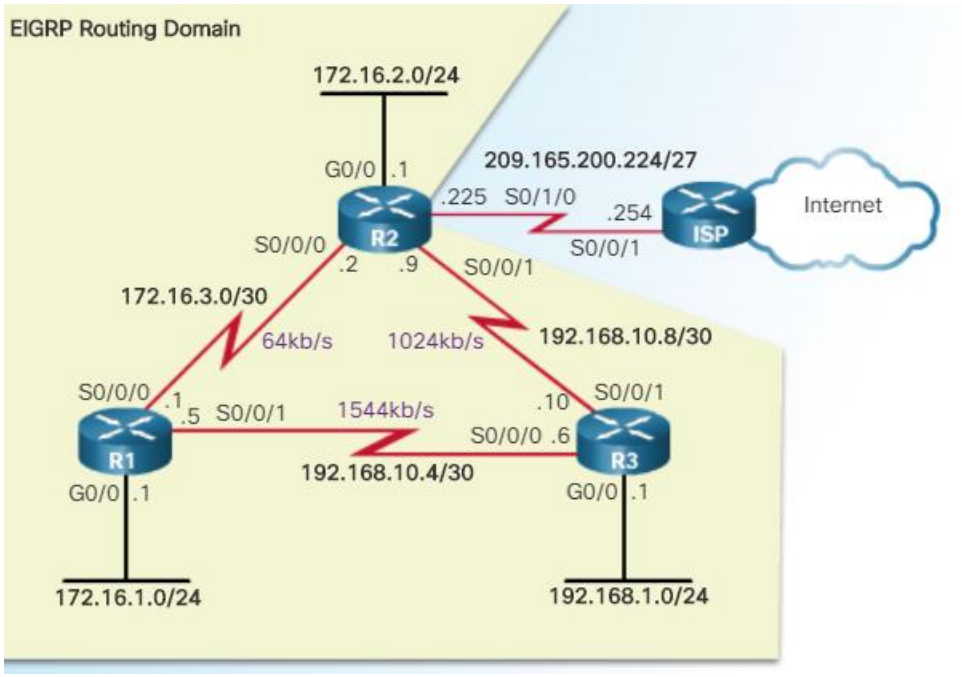
Verifying the *k* Values

The show ip protocols command is used to verify the k values. The command output for R1 is shown in Figure 2. Notice that the k values on R1 are set to the default.



Bandwidth Metric

The bandwidth metric is a static value used by some routing protocols, such as EIGRP and OSPF, to calculate their routing metric. The bandwidth is displayed in kilobits per second (kb/s).



Configuring the Bandwidth Parameter

Because both EIGRP and OSPF use bandwidth in default metric calculations, a correct value for bandwidth is very important to the accuracy of routing information.

Use the following interface configuration mode command to modify the bandwidth metric:

`Router(config-if)# bandwidth`*`kilobits-bandwidth-value`*

Verifying the Bandwidth Parameter

Use the **show interfaces** command to verify the new bandwidth parameters,

Modifying the bandwidth value does not change the actual bandwidth of the link.
The **bandwidth** command only modifies the bandwidth metric used by routing protocols, such as EIGRP and OSPF.

```
R1(config)# interface s 0/0/0
R1(config-if)# bandwidth 64
```

```
R2(config)# interface s 0/0/0
R2(config-if)# bandwidth 64
R2(config-if)# exit
R2(config)# interface s 0/0/1
R2(config-if)# bandwidth 1024
```

```
R3(config)# interface s 0/0/1
R3(config-if)# bandwidth 1024
```

Delay Metric

Delay is the measure of the time it takes for a packet to traverse a route. The delay (DLY) metric is a static value based on the type of link to which the interface is connected and is expressed in microseconds. Delay is not measured dynamically. In other words, the router does not actually track how long packets take to reach the destination. The delay value, much like the bandwidth value, is a default value that can be changed by the network administrator.

Interface Delay Values

Media	Delay
Ethernet	1,000
Fast Ethernet	100
Gigabit Ethernet	10
16M Token Ring	630
FDDI	100
T1 (Serial Default)	20,000
DS0 (64 kb/s)	20,000
1024 kb/s	20,000
56 kb/s	20,000

EIGRP for IPv6

Similar to its IPv4 counterpart, EIGRP for IPv6 exchanges routing information to populate the IPv6 routing table with remote prefixes. EIGRP for IPv6 was made available in Cisco IOS, Release 12.4(6)T.

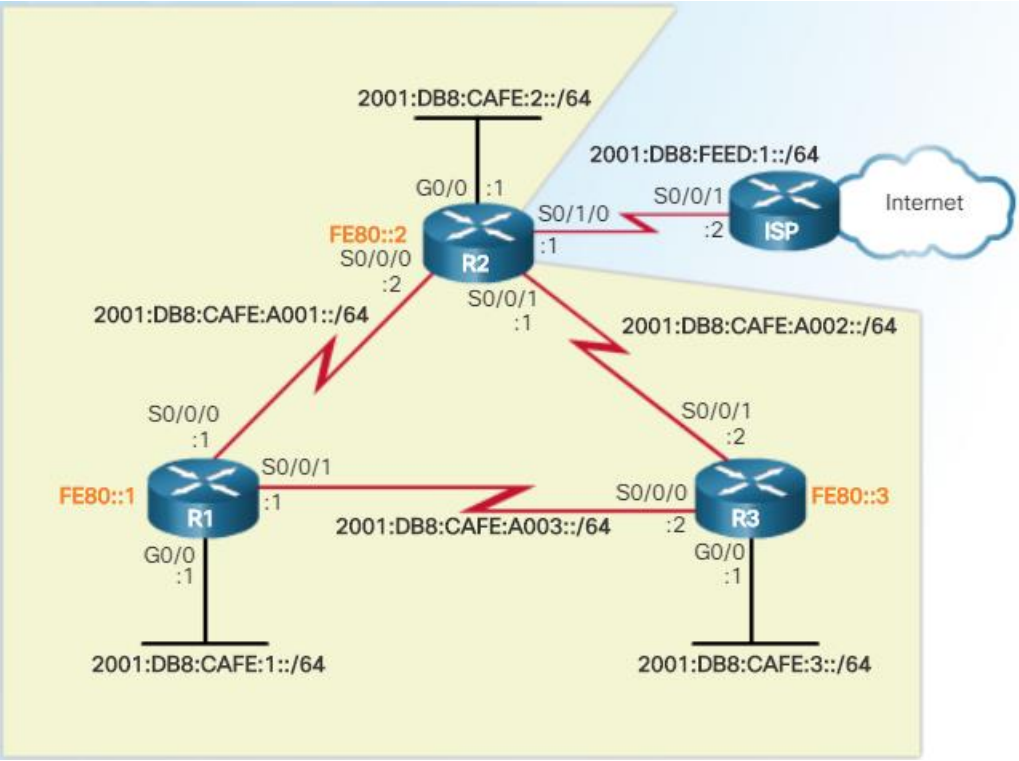
Note: In IPv6, the network address is referred to as the prefix and the subnet mask is called the prefix length.

EIGRP for IPv4 runs over the IPv4 network layer, communicating with other EIGRP IPv4 peers, and advertising only IPv4 routes. EIGRP for IPv6 has the same functionality as EIGRP for IPv4, but uses IPv6 as the network layer transport, communicating with EIGRP for IPv6 peers and advertising IPv6 routes.

EIGRP for IPv6 also uses DUAL as the computation engine to guarantee loop-free paths and backup paths throughout the routing domain.

As with all IPv6 routing protocols, EIGRP for IPv6 has separate processes from its IPv4 counterpart. The processes and operations are basically the same as in the IPv4 routing protocol; however, they run independently. EIGRP for IPv4 and EIGRP for IPv6 each have separate EIGRP neighbor tables, EIGRP topology tables, and IP routing tables, as shown in the figure. EIGRP for IPv6 is a separate protocol-dependent module (PDM).

The EIGRP for IPv6 configuration and verification commands are very similar to those used in EIGRP for IPv4. These commands are described later in this section.



Configuring IPv6 Link-local Addresses

Link-local addresses are automatically created when an IPv6 global unicast address is assigned to the interface. Global unicast addresses are not required on an interface; however, IPv6 link-local addresses are, as shown in Figure 1.

Unless configured manually, Cisco routers create the link-local address using FE80::/10 prefix and the EUI-64 process. EUI-64 involves using the 48-bit Ethernet MAC address, inserting FFFE in the middle

and flipping the seventh bit. For serial interfaces, Cisco uses the MAC address of an Ethernet interface. A router with several serial interfaces can assign the same link-local address to each IPv6 interface, because link-local addresses only need to be local on the link.

Link-local addresses can be configured manually using the same interface configuration mode command used to create IPv6 global unicast addresses, but with different parameters:

```
Router(config-if)# ipv6 address link-local-address link-local
```

A link-local address has a prefix within the range FE80 to FEBF. When an address begins with this hextet (16-bit segment), the link-local keyword must follow the address.

Figure shows the configuration of a link-local address using the ipv6 address interface configuration mode command. The link-local address FE80::1 is used to make it easily recognizable as belonging to router R1. The same IPv6 link-local address is configured on all of R1's interfaces. FE80::1 can be configured on each link because it only has to be unique on that link.

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

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

Configuring the EIGRP for IPv6 Routing Process

The ipv6 unicast-routing global configuration mode command enables IPv6 routing on the router. This command is required before any IPv6 routing protocol can be configured. This command is not required to configure IPv6 addresses on the interfaces, but is necessary for the router to be enabled as an IPv6 router.

EIGRP for IPv6

The following global configuration mode command is used to enter router configuration mode for EIGRP for IPv6:

```
Router(config)# ipv6 router eigrp autonomous-system
```

Similar to EIGRP for IPv4, the *autonomous-system* value must be the same on all routers in the routing domain. In Figure 1, the EIGRP for IPv6 routing process could not be configured until IPv6 routing was enabled with the **ipv6 unicast-routing** global configuration mode command.

Configuring the Router ID

As shown in Figure 2, the eigrp router-id command is used to configure the router ID. EIGRP for IPv6 uses a 32 bit value for the router ID. To obtain that value, EIGRP for IPv6 uses the same process as EIGRP for IPv4. The eigrp router-id command takes precedence over any loopback or physical interface IPv4 addresses. If an EIGRP for IPv6 router does not have any active interfaces with an IPv4 address, then theeigrp router-id command is required.

The router ID should be a unique 32-bit number in the EIGRP for IP routing domain; otherwise, routing inconsistencies can occur.

```
R1(config)# ipv6 router eigrp 2
R1(config-rtr)# eigrp router-id 1.0.0.0
R1(config-rtr)# no shutdown
R1(config-rtr)#
```

The ipv6 eigrp Interface Command

EIGRP for IPv6 uses a different method to enable an interface for EIGRP. Instead of using the network router configuration mode command to specify matching interface addresses, EIGRP for IPv6 is configured directly on the interface.

Use the following interface configuration mode command to enable EIGRP for IPv6 on an interface:

`Router(config-if)#` **`ipv6 eigrp`***`autonomous-system`*

The autonomous-system value must be the same as the autonomous system number used to enable the EIGRP routing process. Similar to the network command used in EIGRP for IPv4, the ipv6 eigrp interface command:

- Enables the interface to form adjacencies and send or receive EIGRP for IPv6 updates.
- Includes the prefix (network) of this interface in EIGRP for IPv6 routing updates.

Figure shows the configuration to enable EIGRP for IPv6 on routers R1 and R2 interfaces. Notice the message following the serial 0/0/0 interface in R2:

```
R1(config)# interface g0/0
R1(config-if)# ipv6 eigrp 2
R1(config-if)# exit
R1(config)# interface s 0/0/0
R1(config-if)# ipv6 eigrp 2
R1(config-if)# exit
R1(config)# interface s 0/0/1
R1(config-if)# ipv6 eigrp 2
R1(config-if)#
```

```
R2(config)# interface g 0/0
R2(config-if)# ipv6 eigrp 2
R2(config-if)# exit
R2(config)# interface s 0/0/0
R2(config-if)# ipv6 eigrp 2
R2(config-if)# exit
%DUAL-5-NBRCHANGE: EIGRP-IPv6 2: Neighbor FE80::1
 (Serial0/0/0) is up: new adjacency
R2(config)# interface s 0/0/1
R2(config-if)# ipv6 eigrp 2
R2(config-if)#
```

```
%DUAL-5-NBRCHANGE: EIGRP-IPv6 2: Neighbor FE80::1 (Serial0/0/0) is up: new adjacency
```

This message indicates that R2 has now formed an EIGRP-IPv6 adjacency with the neighbor at link-local address FE80::1. Because static link-local addresses were configured on all three routers, it is easy to determine that this adjacency is with router R1 (FE80::1).

Passive Interface with EIGRP for IPv6

The same **passive-interface** command used for IPv4 is used to configure an interface as passive with EIGRP for IPv6. As shown in Figure 3, the **show ipv6 protocols** command is used to verify the configuration.

```
R1(config)# ipv6 router eigrp 2
R1(config-rtr)# passive-interface gigabitethernet 0/0
R1(config-rtr)# end

R1# show ipv6 protocols

IPv6 Routing Protocol is "eigrp 2"
EIGRP-IPv6 Protocol for AS(2)


<output omitted>
  Interfaces:
    Serial0/0/0
    Serial0/0/1
    GigabitEthernet0/0 (passive)
 Redistribution:
```

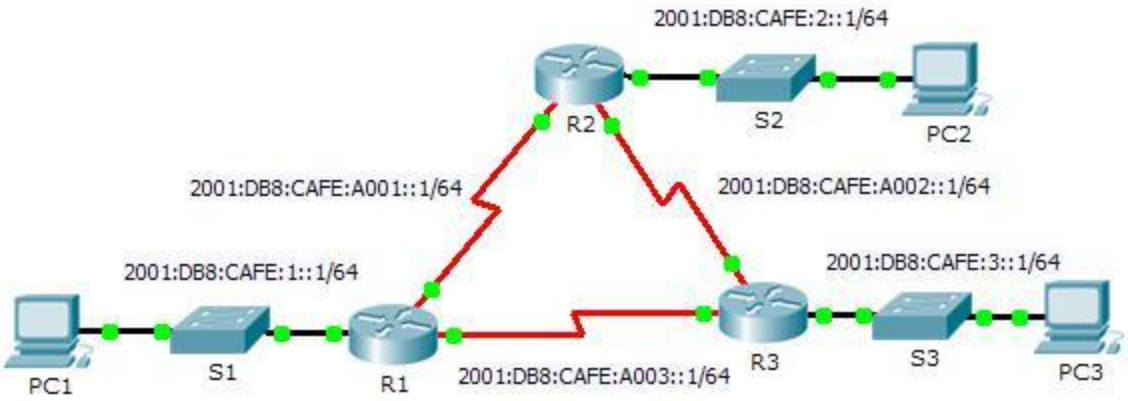
IPv6 Neighbor Table

Similar to EIGRP for IPv4, before any EIGRP for IPv6 updates can be sent or received, routers must establish adjacencies with their neighbors.

Use the **show ipv6 eigrp neighbors** command to view the neighbor table and verify that EIGRP for IPv6 has established an adjacency with its neighbors.

Packet Tracer – Configuring Basic EIGRP with IPv6

Topology



Addressing Table

Device	Interface	IPv6 Address	Default Gateway
R1	G0/0	2001:DB8:CAFE:1::1/64	N/A
	S0/0/0	2001:DB8:CAFE:A001::1/64	N/A
	S0/0/1	2001:DB8:CAFE:A003::1/64	N/A
	Link-local	FE80::1	N/A
R2	G0/0	2001:DB8:CAFE:2::1/64	N/A
	S0/0/0	2001:DB8:CAFE:A001::2/64	N/A
	S0/0/1	2001:DB8:CAFE:A002::1/64	N/A
	Link-local	FE80::2	N/A
R3	G0/0	2001:DB8:CAFE:3::1/64	N/A
	S0/0/0	2001:DB8:CAFE:A003::2/64	N/A
	S0/0/1	2001:DB8:CAFE:A002::2/64	N/A
	Link-local	FE80::3	N/A
PC1	NIC	2001:DB8:CAFE:1::3/64	Fe80::1
PC2	NIC	2001:DB8:CAFE:2::3/64	Fe80::2
PC3	NIC	2001:DB8:CAFE:3::3/64	Fe80::3

Objectives

Part 1: Configure EIGRP for IPv6 Routing

Part 2: Verify IPv6 EIGRP for IPv6 Routing

Scenario

In this activity, you will configure the network with EIGRP routing for IPv6. You will also assign router IDs, configure passive interfaces, verify the network is fully converged, and display routing information using **show** commands.

EIGRP for IPv6 has the same overall operation and features as EIGRP for IPv4. There are a few major differences between them:

- EIGRP for IPv6 is configured directly on the router interfaces.
- With EIGRP for IPv6, a router-id is required on each router or the routing process will not start.
- The EIGRP for IPv6 routing process uses a “shutdown” feature.

Part 1: Configure EIGRP for IPv6 Routing

Step 1: Enable IPv6 routing on each router.

Step 2: Enable EIGRP for IPv6 routing on each router.

The IPv6 routing process is shutdown by default. Issue a command that will enable EIGRP for IPv6 routing in R1, R2 and R3.

Enable the EIGRP process on all routers and use **1** as the Autonomous System number.

Step 3: Assign a router ID to each router.

The router IDs are as follows:

η. R1: 1.1.1.1

ι. R2: 2.2.2.2

φ. R3: 3.3.3.3

Step 4: Using AS 1, configure EIGRP for IPv6 on each interface.

Part 2: Verify EIGRP for IPv6 Routing

Step 1: Examine neighbor adjacencies.

Use the command **show ipv6 eigrp neighbors** to verify that the adjacency has been established with its neighboring routers. The link-local addresses of the neighboring routers are displayed in the adjacency table.

Step 2: Examine the IPv6 EIGRP routing table.

Use the **show ipv6 route** command to display the IPv6 routing table on all routers. EIGRP for IPv6 routes are denoted in the routing table with a **D**.

Step 3: Verify the parameters and current state of the active IPv6 routing protocol processes.

Use the command **show ipv6 protocols** to verify the configured parameter.

Step 4: Verify end-to-end connectivity.

PC1, PC2, and PC3 should now be able to ping each other. If not, troubleshoot your EIGRP configurations.

Lab 3: Enhanced Interior Gateway Routing Protocol (EIGRP) part 2

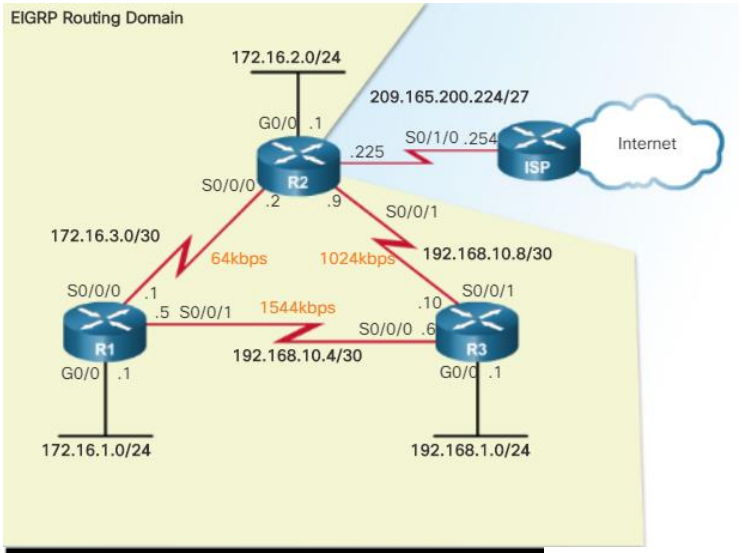
This lab. teaches you how to maintain your EIGRP networks and to influence them to do what you want them to do. EIGRP concepts from this chapter include:

- Auto-summarization

- Load balancing
- Default routes
- Hold-down timers
- Authentication

Network Topology

Before fine tuning EIGRP features, start with a basic implementation of EIGRP




```
R1# show running-config

<output omitted>
version 15.2
!
interface GigabitEthernet0/0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.3.1 255.255.255.252
 clock rate 64000
!
interface Serial0/0/1
 ip address 192.168.10.5 255.255.255.252
!
router eigrp 1
 network 172.16.0.0
 network 192.168.10.0
 eigrp router-id 1.1.1.1
```

EIGRP Automatic Summarization

One of the most common tuning methods of EIGRP is enabling and disabling automatic route summarization. Route summarization allows a router to group networks together and advertises them as one large group using a single, summarized route. The ability to summarize routes is necessary due to the rapid growth of networks.

A border router is a router that sits at the edge of a network. This router must be able to advertise all of the known networks within its routing table to a directly-connected network router or ISP router. This convergence can potentially result in very large route tables. Imagine if a single router had 10 different networks and had to advertise all 10 route entries to a connecting router. What if that connecting router also had 10 networks, and had to advertise all 20 routes to an ISP router? If every enterprise router followed this pattern, the routing table of the ISP router would be huge.

To limit the number of routing advertisements and the size of routing tables, EIGRP provides route summarization features. Summarization decreases the number of entries in routing updates and lowers the number of entries in local routing tables. It also reduces bandwidth utilization for routing updates and results in faster routing table lookups.

```
R1(config)# router eigrp 1
R1(config-router)# auto-summary
R1(config-router)#
```

Verifying Auto-Summary: Topology Table

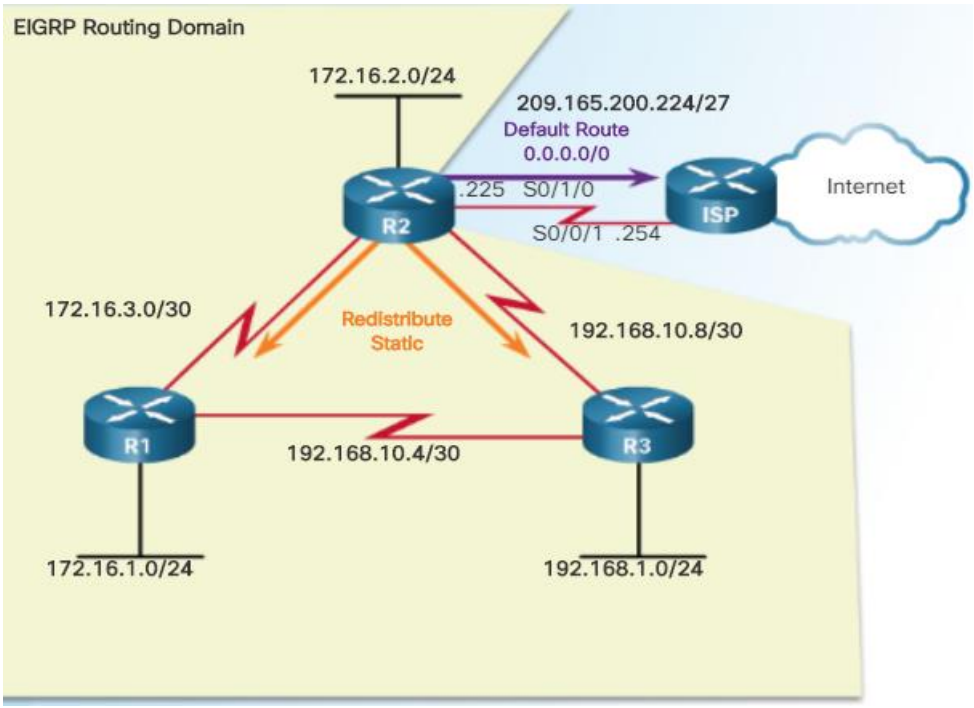
```
R3# show ip eigrp topology all-links

P 172.16.0.0/16, 1 successors, FD is 2170112, serno 9
        via 192.168.10.5 (2170112/2816), Serial0/0/0
        via 192.168.10.9 (3012096/2816), Serial0/0/1

<output omitted>
```

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0

<output omitted>

Automatic Summarization: enabled
    192.168.10.0/24 for Gi0/0, Se0/0/0
      Summarizing 2 components with metric 2169856
    172.16.0.0/16 for Se0/0/1
      Summarizing 3 components with metric 2816
```

Propagating a Default Static Route

Using a static route to 0.0.0.0/0 as a default route is not routing protocol-dependent. The "quad zero" default static route can be used with any currently supported routing protocols. The default static route is usually configured on the router that has a connection to a network outside the EIGRP routing domain; for example, to an ISP.

In Figure, R2 is the gateway router connecting the EIGRP routing domain to the Internet. When the default static route is configured, it is necessary to propagate that route throughout the EIGRP domain, as shown in Figure .



One method of propagating a default static route within the EIGRP routing domain is by using the **redistribute static** command. The**redistribute static** command tells EIGRP to include static routes in its EIGRP updates to other routers. Figure shows the configuration of the default static route and the **redistribute static** command on router R2.

```
R2(config)# ip route 0.0.0.0 0.0.0.0 serial 0/1/0
R2(config)# router eigrp 1
R2(config-router)# redistribute static
```

Figure verifies that the default route has been received by router R2 and installed in its IPv4 routing table.

```
R2# show ip route | include 0.0.0.0
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
S* 0.0.0.0/0 is directly connected, Serial0/1/0
R2#
```

In Figure , the **show ip protocols** command verifies that R2 is redistributing static routes within the EIGRP routing domain.

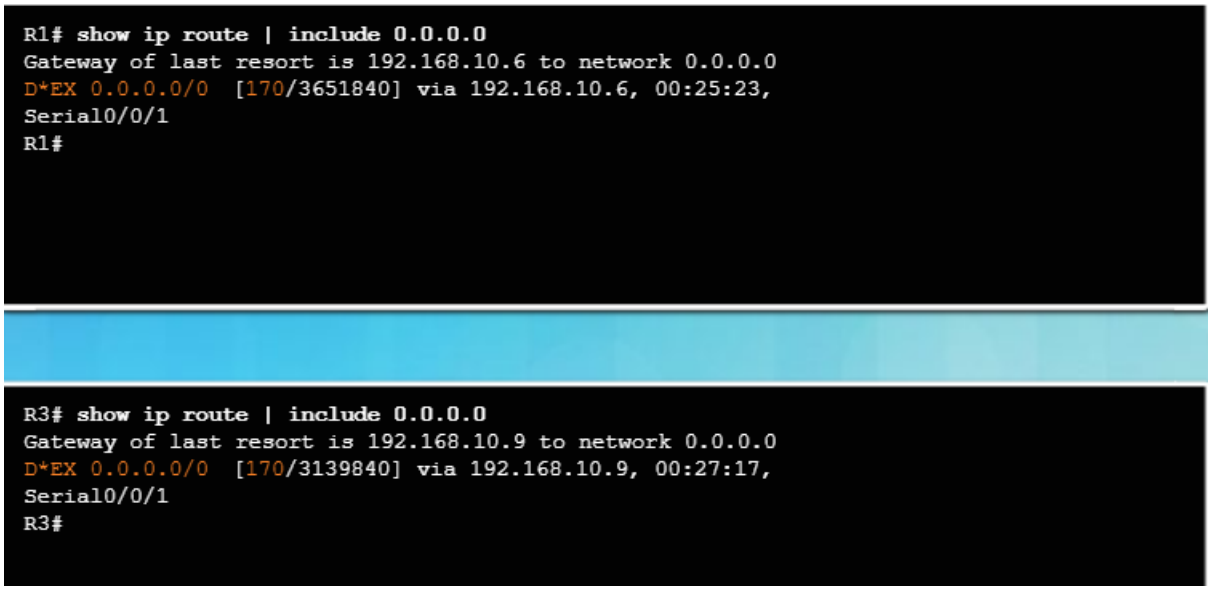
```
R2# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  Redistributing: static
  EIGRP-IPv4 Protocol for AS(1)
  <output omitted>
```

Verifying the Propagated Default Route

The figure displays a portion of the IPv4 routing tables for R1 and R3.
In the routing tables for R1 and R3, notice the routing source and administrative distance for the new default route learned using EIGRP. The entry for the EIGRP learned default route is identified by the following:

- **D** - This route was learned from an EIGRP routing update.
- ***** - The route is a candidate for a default route.
- **EX** - The route is an external EIGRP route, in this case a static route outside of the EIGRP routing domain.
- **170** - This is the administrative distance of an external EIGRP route.

Notice that R1 selects R3 as the successor to the default route because it has a lower feasible distance. Default routes provide a default path to outside the routing domain and, like summary routes, minimize the number of entries in the routing table.



EIGRP for IPv6: Default Route

Recall that EIGRP maintains separate tables for IPv4 and IPv6; therefore, an IPv6 default route must be propagated separately, as shown in Figure 1. Similar to EIGRP for IPv4, a default static route is configured on the gateway router (R2), as shown in Figure 2:

```
R2(config)# ipv6 route ::/0 serial 0/1/0
```

EIGRP Bandwidth Utilization

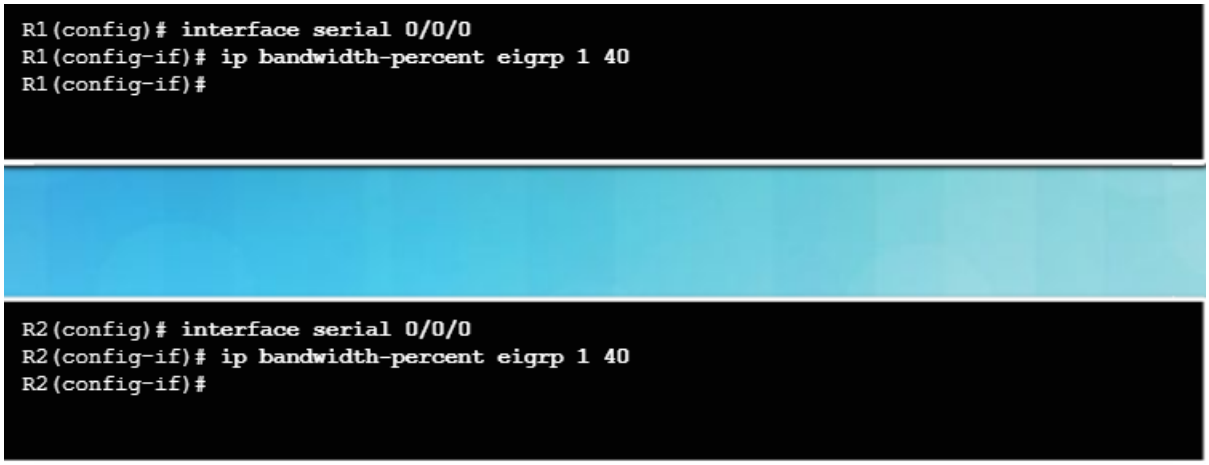
EIGRP Bandwidth for IPv4

By default, EIGRP uses only up to 50 percent of an interface's bandwidth for EIGRP information. This prevents the EIGRP process from over-utilizing a link and not allowing enough bandwidth for the routing of normal traffic.

Use the **ip bandwidth-percent eigrp** command to configure the percentage of bandwidth that can be used by EIGRP on an interface.

```
Router(config-if)# ip bandwidth-percent eigrp as-number percent
```

In Figure 1, R1 and R2 share a very slow 64 kb/s link. The configuration to limit how much bandwidth EIGRP uses is shown in Figure 2. The **ip bandwidth-percent eigrp** command uses the amount of configured bandwidth (or the default bandwidth) when calculating the percent that EIGRP can use. In this example, EIGRP is limited to no more than 50 percent of the link's bandwidth. Therefore, EIGRP never uses more the 32 kb/s of the link's bandwidth for EIGRP packet traffic.



Hello and Hold Timers

Hello Intervals and Hold Times with EIGRP for IPv4

EIGRP uses a lightweight Hello protocol to establish and monitor the connection status of its neighbor. The hold time tells the router the maximum time that the router should wait to receive the next Hello before declaring that neighbor as unreachable.

Hello intervals and hold times are configurable on a per-interface basis and do not have to match with other EIGRP routers to establish or maintain adjacencies. The command to configure a different Hello interval is:

```
Router(config-if)# ip hello-interval eigrp as-number seconds
```

If the Hello interval is changed, ensure that the hold time value is equal to, or greater than, the Hello interval. Otherwise, neighbor adjacency goes down after the hold time expires and before the next Hello interval. Use the following command to configure a different hold time:

```
Router(config-if)# ip hold-time eigrp as-number seconds
```

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

Figure 1 shows the configuration of R1 to use a 50-second Hello interval and 150-second hold time. The **no** form can be used on both of these commands to restore the default values.

The Hello interval time and hold time do not need to match for two routers to form an EIGRP adjacency.

```
R1(config)# interface s0/0/0
R1(config-if)# ip hello-interval eigrp 1 50
R1(config-if)# ip hold-time eigrp 1 150
```

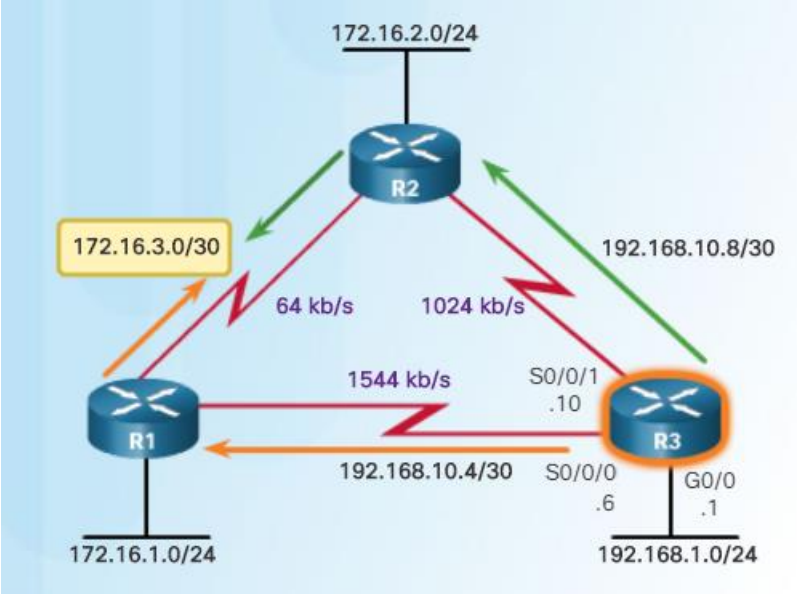
Default Hello Intervals and Hold Times for EIGRP

Bandwidth	Example Link	Default Hello Interval	Default Hold Time
1.544 Mbps	Multipoint Frame Relay	60 seconds	180 seconds
Greater than 1.544 Mbps	T1, Ethernet	5 seconds	15 seconds

Load Balancing IPv4

Equal-cost load balancing is the ability of a router to distribute outbound traffic using all interfaces that have the same metric from the destination address. Load balancing uses network segments and bandwidth more efficiently. For IP, Cisco IOS Software applies load balancing using up to four equal-cost paths by default.

Figure 1 shows the EIGRP for IPv4 network topology. In this topology, R3 has two EIGRP equal-cost routes for the network between R1 and R2, 172.16.3.0/30. One route is via R1 at 192.168.10.4/30 and the other route is via R2 at 192.168.10.8/30.



The show ip protocols command can be used to verify the number of equal-cost paths currently configured on the router. The output in Figure 2 shows that R3 is using the default of four equal-cost paths.

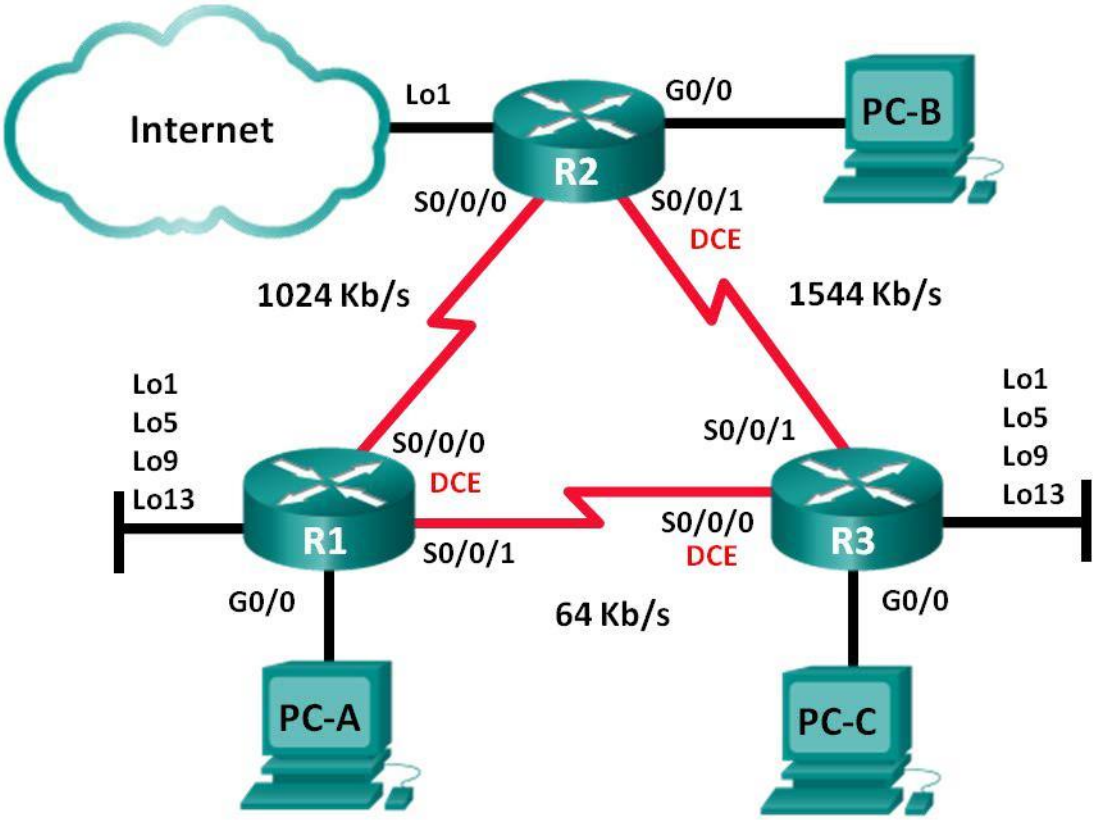
The routing table maintains both routes. Figure 3 shows that R3 has two EIGRP equal-cost routes for the 172.16.3.0/30 network. One route is via R1 at 192.168.10.5 and the other route is via R2 at 192.168.10.9. Looking at the topology in Figure 1, it may seem as if the path via R1 is the better route because there is a 1544 kb/s link between R3 and R1, whereas the link to R2 is only a 1024 kb/s link. However, EIGRP only uses the slowest bandwidth in its composite metric which is the 64 kb/s link between R1 and R2. Both paths have the same 64 kb/s link as the slowest bandwidth, this results in both paths being equal.

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

The value argument refers to the number of paths that should be maintained for load balancing. If the value is set to 1, load balancing is disabled.

Lab – Configuring Advanced EIGRP for IPv4 Features

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0 (DCE)	192.168.12.1	255.255.255.252	N/A
	S0/0/1	192.168.13.1	255.255.255.252	N/A
	Lo1	192.168.11.1	255.255.255.252	N/A
	Lo5	192.168.11.5	255.255.255.252	N/A
	Lo9	192.168.11.9	255.255.255.252	N/A
	Lo13	192.168.11.13	255.255.255.252	N/A
R2	G0/0	192.168.2.1	255.255.255.0	N/A
	S0/0/0	192.168.12.2	255.255.255.252	N/A
	S0/0/1 (DCE)	192.168.23.1	255.255.255.252	N/A
	Lo1	192.168.22.1	255.255.255.252	N/A
R3	G0/0	192.168.3.1	255.255.255.0	N/A
	S0/0/0 (DCE)	192.168.13.2	255.255.255.252	N/A
	S0/0/1	192.168.23.2	255.255.255.252	N/A
	Lo1	192.168.33.1	255.255.255.252	N/A
	Lo5	192.168.33.5	255.255.255.252	N/A
	Lo9	192.168.33.9	255.255.255.252	N/A
	Lo13	192.168.33.13	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure EIGRP and Verify Connectivity

Part 3: Configure EIGRP for Automatic Summarization

Part 4: Configure and Propagate a Default Static Route

Part 5: Fine-Tune EIGRP

- Configure bandwidth utilization for EIGRP.
- Configure the hello interval and hold timer for EIGRP.

Background / Scenario

EIGRP has advanced features to allow changes related to summarization, default route propagation, bandwidth utilization, and metrics.

In this lab, you will configure automatic summarization for EIGRP, configure EIGRP route propagation, and fine-tune EIGRP metrics.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at this end of the lab for the correct interface identifiers.

Note: Ensure that the routers have been erased and have no startup configurations. If you are unsure, contact your instructor.

Required Resources

- 3 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 3 PCs (Windows with a terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings on the PC hosts and routers.

Step 1: Cable the network as shown in the topology.

Step 2: Configure PC hosts.

Step 3: Initialize and reload the routers as necessary.

Step 4: Configure basic settings for each router.

a. Disable DNS lookup.

b. Configure device name as shown in the topology.

c. Assign **cisco** as the console and vty passwords.

d. Assign **class** as the privileged EXEC password.

e. Configure **logging synchronous** to prevent console messages from interrupting command entry.

f. Configure the IP address listed in the Addressing Table for all interfaces.

Note: Do **NOT** configure the loopback interfaces at this time.

g. Copy the running configuration to the startup configuration.

Part 2: Configure EIGRP and Verify Connectivity

In Part 2, you will configure basic EIGRP for the topology and set bandwidths for the serial interfaces.

Note: This lab provides minimal assistance with the actual commands necessary to configure EIGRP. However, the required commands are provided in Appendix A. Test your knowledge by trying to configure the devices without referring to the appendix.

Step 1: Configure EIGRP.

a. On R1, configure EIGRP routing with an autonomous system (AS) ID of 1 for all directly connected networks. Write the commands used in the space below.

b. For the LAN interface on R1, disable the transmission of EIGRP hello packets. Write the command used in the space below.

c. On R1, configure the bandwidth for S0/0/0 to 1024 Kb/s and the bandwidth for S0/0/1 to 64 Kb/s. Write the commands used in the space below. **Note**: The **bandwidth** command only affects the EIGRP metric calculation, not the actual bandwidth of the serial link.

d. On R2, configure EIGRP routing with an AS ID of 1 for all networks, disable the transmission of EIGRP hello packets for the LAN interface, and configure the bandwidth for S0/0/0 to 1024 Kb/s.

e. On R3, configure EIGRP routing with an AS ID of 1 for all networks, disable the transmission of EIGRP hello packets for the LAN interface, and configure the bandwidth for S0/0/0 to 64 Kb/s.

Step 2: Test connectivity.

All PCs should be able to ping one another. Verify and troubleshoot if necessary.

Note: It may be necessary to disable the PC firewall to ping between PCs.

Part 3: Configure EIGRP for Automatic Summarization

In Part 3, you will add loopback interfaces and enable EIGRP automatic summarization on R1 and R3. You will also observe the effects on the routing table of R2.

Step 1: Configure EIGRP for automatic summarization.

a. Issue the **show ip protocols** command on R1. What is the default status of automatic summarization in EIGRP?

b. Configure the loopback addresses on R1.

c. Add the appropriate network statements to the EIGRP process on R1. Record the commands used in the space below.

d. On R2, issue the **show ip route eigrp** command. How are the loopback networks represented in the output?

e. On R1, issue the **auto-summary** command inside the EIGRP process.

```
R1(config)# router eigrp 1
R1(config-router)# auto-summary
R1(config-router)#
*Apr 14 01:14:55.463: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.13.2
(Serial0/0/1) is resync: summary configured
*Apr 14 01:14:55.463: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.12.2
(Serial0/0/0) is resync: summary configured
*Apr 14 01:14:55.463: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.13.2
(Serial0/0/1) is resync: summary up, remove components
R1(config-router)#67: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.12.2
(Serial0/0/0) is resync: summary up, remove components
*Apr 14 01:14:55.467: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.12.2
(Serial0/0/0) is resync: summary up, remove components
*Apr 14 01:14:55.467: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.13.2
(Serial0/0/1) is resync: summary up, remove components
```

How does the routing table on R2 change?

f. Repeat substeps b through e by adding loopback interfaces, adding EIGRP process networks and auto-summary on R3.

Part 4: Configure and Propagate a Default Static Route

In Part 4, you will configure a default static route on R2 and propagate the route to all other routers.

i. Configure the loopback address on R2.

j. Configure a default static route with an exit interface of Lo1.

```
R2(config)# ip route 0.0.0.0 0.0.0.0 Lo1
```

k. Use the **redistribute static** command within the EIGRP process to propagate the default static route to other participating routers.

```
R2(config)# router eigrp 1 R2(config-
router)# redistribute static
```

l. Use the **show ip protocols** command on R2 to verify the static route is being distributed.

```
R2# show ip protocols
*** IP Routing is NSF aware ***
<output omitted>
Routing Protocol is "eigrp 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
Redistributing: static
EIGRP-IPv4 Protocol for AS(1)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 192.168.23.1
Topology : 0 (base)
Active Timer: 3 min
```

```
Distance: internal 90 external 170
Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1

Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
192.168.2.0
192.168.12.0/30
192.168.23.0/30
Passive Interface(s):
GigabitEthernet0/0
Routing Information Sources:
Gateway                   Distance      Last Update
192.168.12.1                    90      00:13:20
192.168.23.2                    90      00:13:20
Distance: internal 90 external 170
```

+ On R1, issue the **show ip route eigrp | include 0.0.0.0** command to view statements specific to the default route. How is the static default route represented in the output? What is the administrative distance (AD) for the propagated route?

Part 5: Fine-Tune EIGRP

In Part 5, you will configure the percentage of bandwidth that can be used for EIGRP traffic on an interface and change the hello interval and hold timers for EIGRP interfaces.

Step 1: Configure bandwidth utilization for EIGRP.

E Configure the serial link between R1 and R2 to allow only 75 percent of the link bandwidth for EIGRP traffic.

```
R1(config)# interface s0/0/0 R1(config-if)# ip
bandwidth-percent eigrp 1 75 R2(config)# interface
s0/0/0 R2(config-if)# ip bandwidth-percent eigrp 1
75
```

F Configure the serial link between R1 and R3 to allow 40 percent of the links bandwidth for EIGRP traffic.

Step 2: Configure the hello interval and hold timer for EIGRP.

b. On R2, use the **show ip eigrp interfaces detail** command to view the hello interval and hold timer for EIGRP.

```
R2# show ip eigrp interfaces detail

       EIGRP-IPv4 Interfaces for AS(1)
                                  Xmit Queue   PeerQ        Mean   Pacing Time   Multicast    Pending
       Interface           Peers  Un/Reliable  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
       Se0/0/0                 1        0/0          0/0          1        0/15          50           0
         Hello-interval is 5, Hold-time is 15
```

```
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 29/1
Hello's sent/expedited: 390/2
Un/reliable mcasts: 0/0   Un/reliable ucasts: 35/39
Mcast exceptions: 0   CR  packets: 0  ACKs suppressed: 0
Retransmissions sent: 0   Out-of-sequence rcvd: 0
Topology-ids on interface - 0
Interface BW percentage   is 75
Authentication mode is not set
Se0/0/1                  10/0               0/0          1       0/16          50          0
  Hello-interval is 5, Hold-time is 15
  Split-horizon is enabled
  Next xmit serial <none>
  Packetized sent/expedited: 34/5
  Hello's sent/expedited: 382/2
  Un/reliable mcasts: 0/0 Un/reliable ucasts: 31/42
  Mcast exceptions: 0 CR  packets: 0  ACKs suppressed: 2
  Retransmissions sent: 0 Out-of-sequence rcvd: 0
  Topology-ids on interface - 0
  Authentication mode is not set
```

What is the default value for hello time?

What is the default value for hold time?

b. Configure S0/0/0 and S0/0/1 interfaces on R1 to use a hello interval of 60 seconds and a hold time of 180 seconds in that specific order.

```
R1(config)# interface s0/0/0 R1(config-if)# ip
hello-interval eigrp 1 60 R1(config-if)# ip
hold-time eigrp 1 180 R1(config)# interface
s0/0/1 R1(config-if)# ip hello-interval eigrp 1
60 R1(config-if)# ip hold-time eigrp 1 180
```

c. Configure the serial interfaces on R2 and R3 to use a hello interval of 60 seconds and a hold time of 180 seconds.

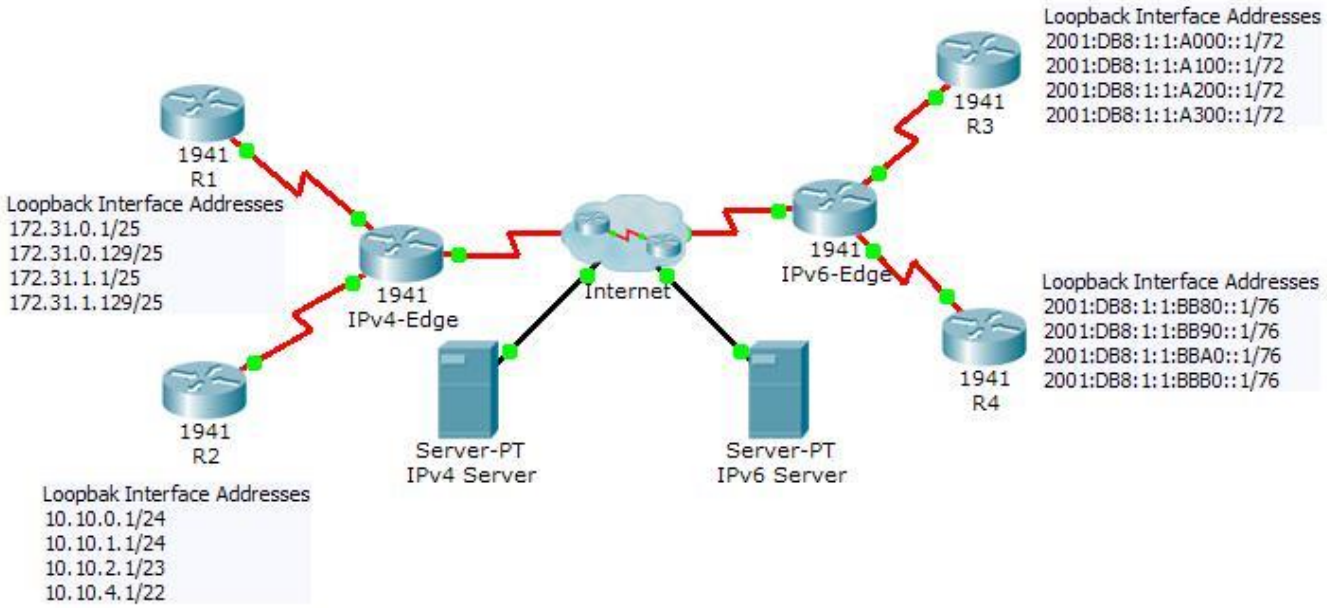
Reflection

b What are the benefits of summarizing routes?

c When setting EIGRP timers, why is it important to make the hold time value equal to or greater than the hello interval?

Packet Tracer - Skills Integration Challenge

Topology



Addressing Table

Device	Interface	IPv4 Address	Subnet Mask
		IPv6 Address/Prefix	
IPv4-Edge	S0/0/0	172.31.6.1	255.255.255.252
	S0/0/1	10.10.8.1	255.255.255.252
	S0/1/0	209.165.200.226	255.255.255.224
R1	S0/0/0	172.31.6.2	255.255.255.252
	Lo8	172.31.0.1	255.255.255.128
	Lo9	172.31.0.129	255.255.255.128
	Lo10	172.31.1.1	255.255.255.128
	Lo11	172.31.1.129	255.255.255.128
R2	S0/0/1	10.10.8.2	255.255.255.252
	Lo1	10.10.0.1	255.255.255.0
	Lo2	10.10.1.1	255.255.255.0
	Lo3	10.10.2.1	255.255.254.0
	Lo4	10.10.4.1	255.255.252.0
IPv6-Edge	S0/0/0	2001:DB8:A001:6::1/64	
	S0/0/1	2001:DB8:A001:7::1/64	
	S0/1/0	2001:DB8:CAFE:1::2/64	
R3	S0/0/0	2001:DB8:A001:7::2/64	
R4	S0/0/1	2001:DB8:A001:6::2/64	

Scenario

In this activity, you must implement EIGRP for IPv4 and IPv6 on two separate networks. Your task includes enabling EIGRP, assigning router IDs, changing the hello timers, and limiting EIGRP advertisements.

Requirements

EIGRP for IPv4

k. Implement EIGRP on IPv4-enabled routers using Autonomous System 1.
   - Use a single classful network address to advertise the loopback interfaces.
   - Use the wildcard mask to advertise the /30 networks between **R1**, **R2** and **IPv4-Edge**.
   - Use the **default** passive interface method and only allow EIGRP updates out the active EIGRP serial interfaces.

l. Configure a directly attached default route on **IPv4-Edge** and propagate it in EIGRP updates.

m. Configure the serial interfaces between **R1**, **R2** and **IPv4-Edge** to send hellos every 10 seconds.

f. **R1** and **R2** should have a default route in the routing table (D*EX).

g. Verify **R1** and **R2** can ping the **IPv4 Server**. **IPv4 Server** should also be able to ping every loopback address on **R1** and **R2**.

EIGRP for IPv6

c. Implement EIGRP for IPv6 on the IPv6-enabled routers using Autonomous System 1.

- Assign **IPv6-Edge** with the router ID of 1.1.1.1
- Assign **R3** with the router ID of 3.3.3.3
- Assign **R4** with the router ID of 4.4.4.4

d. Configure a directly attached default route on **IPv6-Edge** and propagate it in EIGRP updates.

e. **R3** and **R4** should show a default external route in the routing table.

f. Verify **R3** and **R4** can ping the **IPv6 Server**. **IPv6 Server** should also be able to ping every loopback address on **R3** and **R4**.

Suggested Scoring Rubric

Note: Packet Tracer does not currently grade EIGRP for IPv6 summary routes. Therefore, part of your grade depends on routing table verification by your instructor.

Scored Work	Possible Points	Earned Points
IPv6-Edge Routing Table	10	
Packet Tracer Score	90	
Total Score	**100**	

Lab 4: Single-Area OSPF OSPFv2 and OSPFv3

Shortest Path First Protocols

Link-state routing protocols are also known as shortest path first protocols and are built around Edsger Dijkstra's shortest path first (SPF) algorithm. The SPF algorithm is discussed in more detail in a later section.

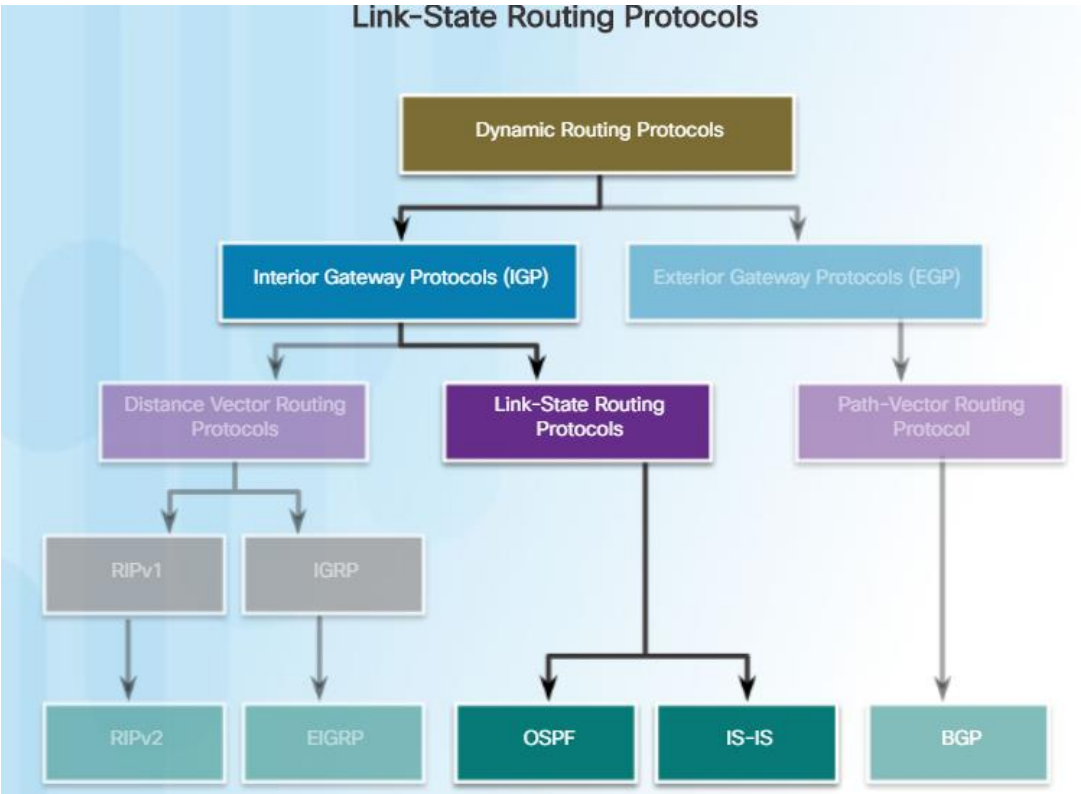
The IPv4 link-state routing protocols are shown in the figure:

- Open Shortest Path First (OSPF)
- Intermediate System-to-Intermediate System (IS-IS)

Link-state routing protocols have the reputation of being much more complex than their distance vector counterparts. However, the basic functionality and configuration of link-state routing protocols is equally straight-forward.

Just like RIP and EIGRP, basic OSPF operations can be configured using the:

- **router ospf** *process-id* global configuration command
- **network** command to advertise networks

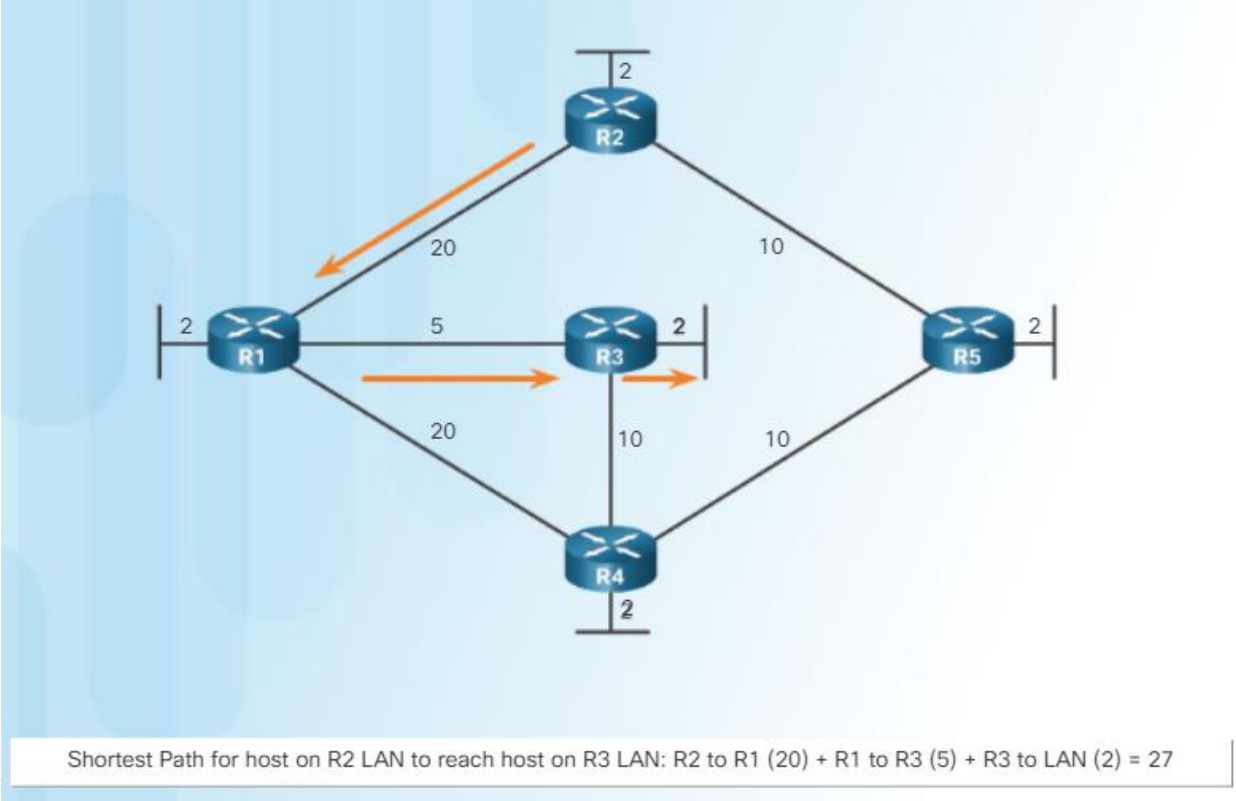


Dijkstra's Algorithm

All link-state routing protocols apply Dijkstra's algorithm to calculate the best path route. The algorithm is commonly referred to as the shortest path first (SPF) algorithm. This algorithm uses accumulated costs along each path, from source to destination, to determine the total cost of a route.

In the figure, each path is labeled with an arbitrary value for cost. The cost of the shortest path for R2 to send packets to the LAN attached to R3 is 27. Each router determines its own cost to each destination in the topology. In other words, each router calculates the SPF algorithm and determines the cost from its own perspective.

Note: The focus of this section is on cost, which is determined by the SPF tree. For this reason, the graphics throughout this section show the connections of the SPF tree, not the topology. All links are represented with a solid black line.



Shortest Path for host on R2 LAN to reach host on R3 LAN: R2 to R1 (20) + R1 to R3 (5) + R3 to LAN (2) = 27

Link-State Routing Process

So exactly how does a link-state routing protocol work? With link-state routing protocols, a link is an interface on a router. Information about the state of those links is known as link-states.

All routers in an OSPF area will complete the following generic link-state routing process to reach a state of convergence:

1. Each router learns about its own links and its own directly connected networks. This is done by detecting that an interface is in the up state.

2. Each router is responsible for meeting its neighbors on directly connected networks. Link state routers do this by exchanging Hello packets with other link-state routers on directly connected networks.

3. Each router builds a Link-State Packet (LSP) containing the state of each directly connected link. This is done by recording all the pertinent information about each neighbor, including neighbor ID, link type, and bandwidth.

4. Each router floods the LSP to all neighbors. Those neighbors store all LSPs received in a database. They then flood the LSPs to their neighbors until all routers in the area have received the LSPs. Each router stores a copy of each LSP received from its neighbors in a local database.

5. Each router uses the database to construct a complete map of the topology and computes the best path to each destination network. Like having a road map, the router now has a complete map of all destinations in the topology and the routes to reach them. The SPF algorithm is used to construct the map of the topology and to determine the best path to each network.

Note: This process is the same for both OSPF for IPv4 and OSPF for IPv6. The examples in this section will refer to OSPF for IPv4.

Say Hello

The second step in the link-state routing process is that each router is responsible for meeting its neighbors on directly connected networks.

Routers with link-state routing protocols use a Hello protocol to discover any neighbors on its links. A neighbor is any other router that is enabled with the same link-state routing protocol.

Click Play in the figure to view an animation on the link-state neighbor discovery process with Hello packets.

In the animation, R1 sends Hello packets out of its links (interfaces) to discover any neighbors. R2, R3, and R4 reply to the Hello packet with their own Hello packets because these routers are configured with the same link-state routing protocol. There are no neighbors out the FastEthernet 0/0 interface. Because R1 does not receive a Hello on this interface, it does not continue with the link-state routing process steps for the FastEthernet 0/0 link.

When two link-state routers learn that they are neighbors, they form an adjacency. These small Hello packets continue to be exchanged between two adjacent neighbors and serve as a keepalive function to monitor the state of the neighbor. If a router stops receiving Hello packets from a neighbor, that neighbor is considered unreachable and the adjacency is broken.

Building the Link-State Packet

The third step in the link-state routing process is that each router builds a link-state packet (LSP) containing the state of each directly connected link.

After a router has established its adjacencies, it can build its LSP that contains the link-state information about its links. A simplified version of the LSP from R1 displayed in the figure would contain the following:

1. R1; Ethernet network 10.1.0.0/16; Cost 2

2. R1 -> R2; Serial point-to-point network; 10.2.0.0/16; Cost 20

3. R1 -> R3; Serial point-to-point network; 10.3.0.0/16; Cost 5

4. R1 -> R4; Serial point-to-point network; 10.4.0.0/16; Cost 20

Flooding the LSP

The fourth step in the link-state routing process is that each router floods the LSP to all neighbors, who then store all LSPs received in a database.

Each router floods its link-state information to all other link-state routers in the routing area. Whenever a router receives an LSP from a neighboring router, it immediately sends that LSP out all other interfaces except the interface that received the LSP. This process creates a flooding effect of LSPs from all routers throughout the routing area.

Notice how the LSPs are flooded almost immediately after being received without any intermediate calculations. Link-state routing protocols calculate the SPF algorithm after the flooding is complete. As a result, link-state routing protocols reach convergence very quickly.

Remember that LSPs do not need to be sent periodically. An LSP only needs to be sent:

- During initial startup of the routing protocol process on that router (e.g., router restart)
- Whenever there is a change in the topology (e.g., a link going down or coming up, a neighbor adjacency being established or broken)

In addition to the link-state information, other information is included in the LSP, such as sequence numbers and aging information, to help manage the flooding process. This information is used by each router to determine if it has already received the LSP from another router or if the LSP has newer information than what is already contained in the link-state database. This process allows a router to keep only the most current information in its link-state database.

Building the Link-State Database

The final step in the link-state routing process is that each router uses the database to construct a complete map of the topology and computes the best path to each destination network.

Eventually, all routers receive an LSP from every other link-state router in the routing area. These LSPs are stored in the link-state database.

The example in the figure displays the link-state database content of R1.

As a result of the flooding process, R1 has learned the link-state information for each router in its routing area. Notice that R1 also includes its own link-state information in the link-state database.

With a complete link-state database, R1 can now use the database and the shortest path first (SPF) algorithm to calculate the preferred path or shortest path to each network resulting in the SPF tree.

Building the SPF Tree

Each router in the routing area uses the link-state database and SPF algorithm to construct the SPF tree.

For example, using the link-state information from all other routers, R1 can now begin to construct an SPF tree of the network. To begin, the SPF algorithm interprets each router's LSP to identify networks and associated costs.

R1 identifies its directly connected networks and costs.

R1 keeps adding any unknown network and associated costs to the SPF tree. Notice that R1 ignores any networks it has already identified.

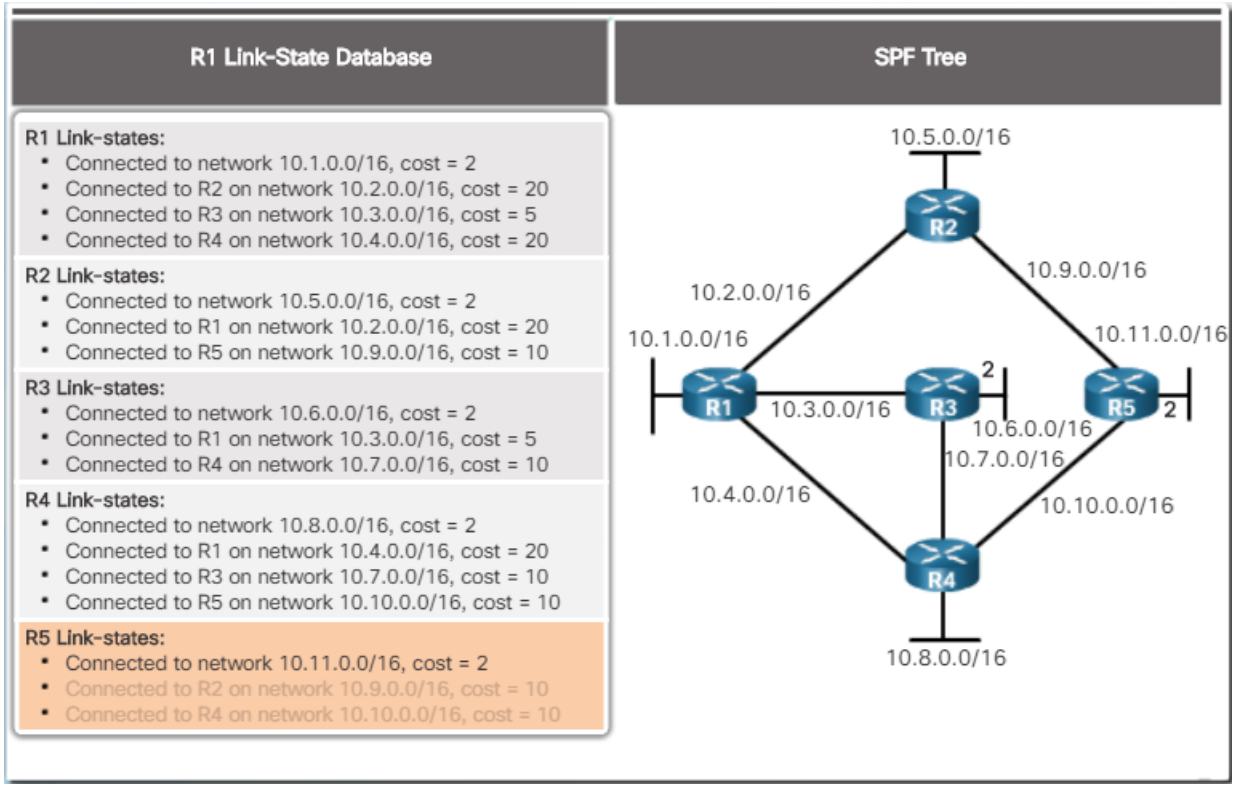
The SPF algorithm then calculates the shortest paths to reach each individual network resulting in the SPF tree as shown in Figure. R1 now has a complete topology view of the link-state area.

Each router constructs its own SPF tree independently from all other routers. To ensure proper routing, the link-state databases used to construct those trees must be identical on all routers.

Adding OSPF Routes to the Routing Table

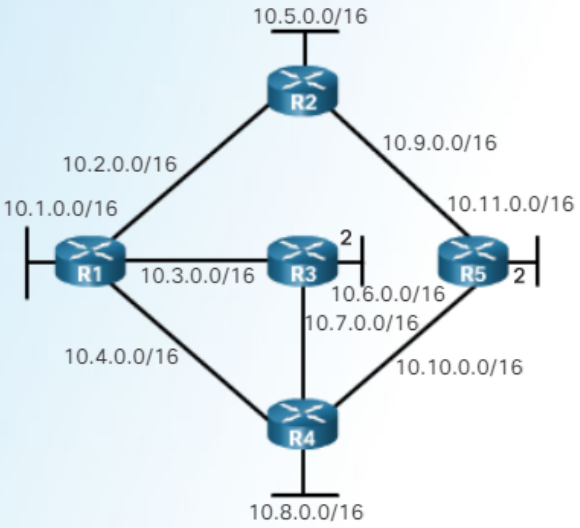
Using the shortest path information determined by the SPF algorithm, these paths can now be added to the routing table. The figure shows the routes that have now been added to R1's IPv4 routing table.

The routing table also includes all directly connected networks and routes from any other sources, such as static routes. Packets are now forwarded according to these entries in the routing table



Resulting SPF Tree of R1

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



Advantages of Link-State Routing Protocols

- Each router builds its own topological map of the network to determine the shortest path.
- Immediate flooding of LSPs achieves faster convergence.
- LSPs are sent only when there is a change in the topology and contain only the information regarding that change.
- Hierarchical design used when implementing multiple areas.

Disadvantages of Link-State Routing Protocols

- Maintaining a link-state database and SPF tree requires additional memory.
- Calculating the SPF algorithm also requires additional CPU processing.
- Bandwidth can be adversely affected by LSP flooding.

Features of OSPF

OSPF features, as shown in Figure 1, include:

- **Classless** - OSPFv2 is classless by design; therefore, it supports IPv4 VLSM and CIDR.
- **Efficient** - Routing changes trigger routing updates (no periodic updates). It uses the SPF algorithm to choose the best path.
- **Fast convergence** - It quickly propagates network changes.

- **Scalable** - It works well in small and large network sizes. Routers can be grouped into areas to support a hierarchical system.

- **Secure** - OSPFv2 supports Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) authentication. OSPFv3 uses Internet Protocol Security (IPsec) to add authentication for OSPFv3 packets. When authentication is enabled, OSPF routers only accept encrypted routing updates from peers with the same pre-shared password.

Administrative distance (AD) is the trustworthiness (or preference) of the route source. OSPF has a default administrative distance of 110.

OSPF Data Structures

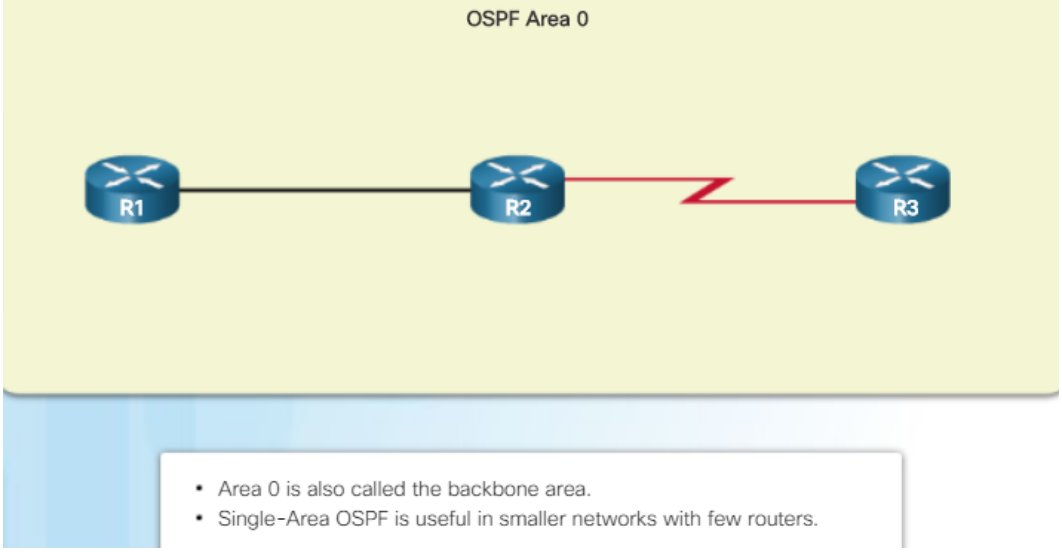
Database	Table	Description
Adjacency Database	Neighbor Table	• List of all neighbor routers to which a router has established bidirectional communication. • This table is unique for each router. • Can be viewed using the `show ip ospf neighbor` command.
Link-state Database (LSDB)	Topology Table	• Lists information about all other routers in the network. • The 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. • Each router's routing table is unique and contains information on how and where to send packets to other routers. • Can be viewed using the `show ip route` command.

Single-Area and Multiarea OSPF

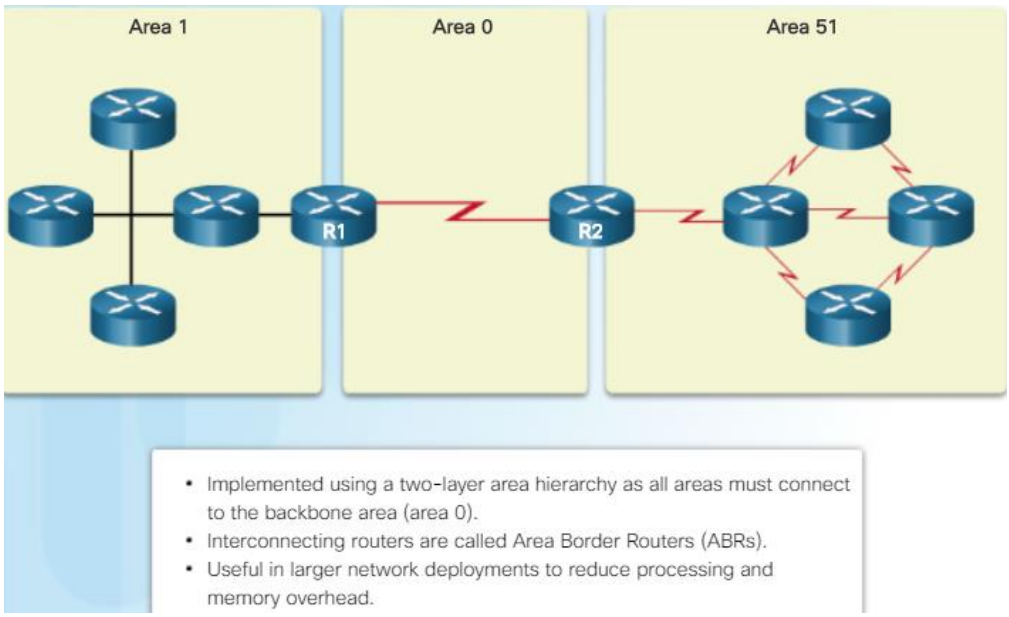
To make OSPF more efficient and scalable, OSPF supports hierarchical routing using areas. An OSPF area is a group of routers that share the same link-state information in their LSDBs.

OSPF can be implemented in one of two ways:

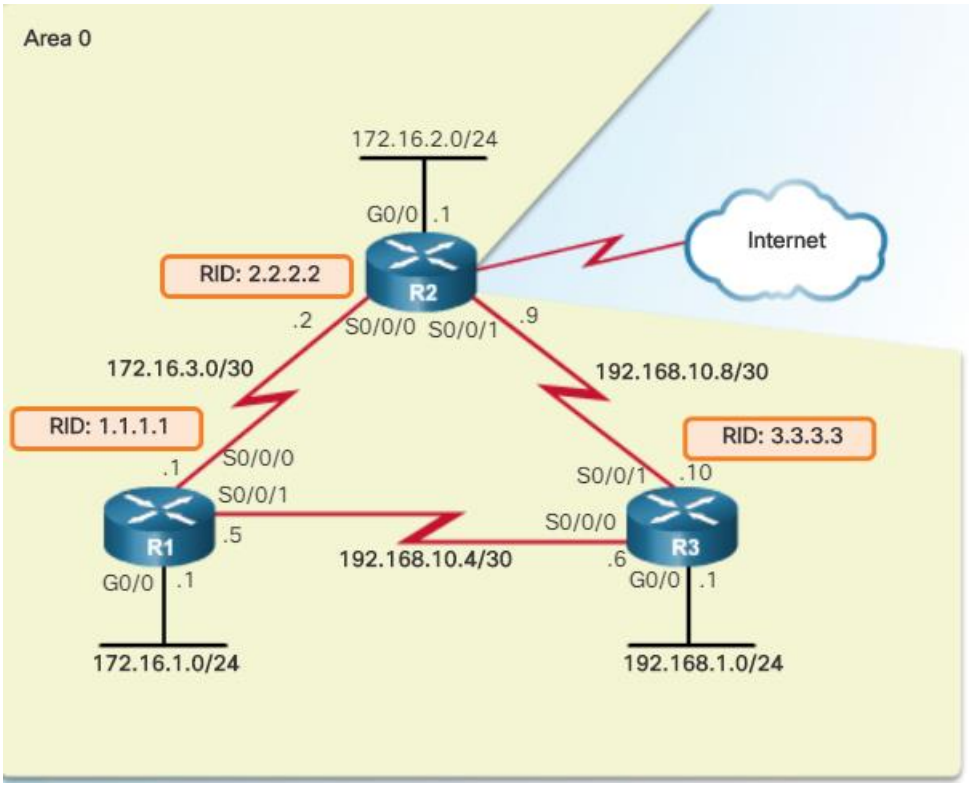
- **Single-Area OSPF** - In Figure, all routers are in one area called the backbone area (area 0).



- **Multiarea OSPF** - In Figure, OSPF is implemented using multiple areas, in a hierarchal fashion. All areas must connect to the backbone area (area 0). Routers interconnecting the areas are referred to as Area Border Routers (ABRs).



OSPF Network Topology



Router OSPF Configuration Mode

OSPFv2 is enabled using the **router ospf** *process-id* global configuration mode command. The *process-id* value represents a number between 1 and 65,535 and is selected by the network administrator. The *process-id* value is locally significant, which means that it does not have to be the same value on the other OSPF routers to establish adjacencies with those neighbor

Router IDs

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 the OSPF-enabled router to:

- **Uniquely identify the router** - The router ID is used by other routers to uniquely identify each router within the OSPF domain and all packets that originate from them.

- **Participate in the election of the DR** - In a multiaccess LAN environment, the election of the DR occurs during initial establishment of the OSPF network. When OSPF links become active, the routing device configured with the highest priority is elected the DR. Assuming there is no priority configured, or there is a tie, then the router with the highest router ID is elected the DR. The routing device with the second highest router ID is elected the BDR.

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

- The router ID is explicitly configured using the OSPF **router-id** *rid* router configuration mode command. The *rid* value is any 32-bit value expressed as an IPv4 address.
- If the router ID is not explicitly configured, the router chooses the highest IPv4 address of any of configured loopback interfaces.
- If no loopback interfaces are configured, then the router chooses the highest active IPv4 address of any of its physical interfaces.

Configuring an OSPF Router ID

Use the **router-id** *rid* router configuration mode command to manually assign a 32-bit value expressed as an IPv4 address to a router. An OSPF router identifies itself to other routers using this router ID.

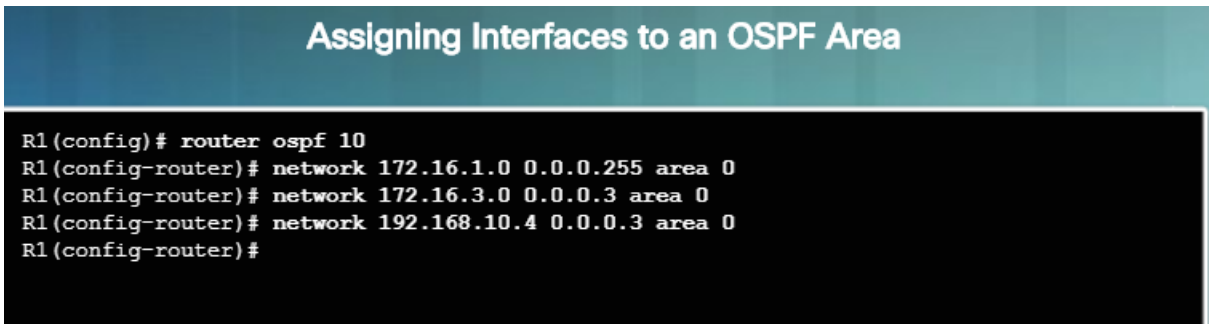
R1 is configured with a router ID of 1.1.1.1, R2 with 2.2.2.2, and R3 with 3.3.3.3.

```
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# end
```

Modifying a Router ID

Sometimes a router ID needs to be changed, for example, when a network administrator establishes a new router ID scheme for the network. However, after a router selects a router ID, an active OSPFv2 router does not allow the router ID to be changed until the router is reloaded or the OSPFv2 process cleared.

The OSPFv2 routing process is cleared using the **clear ip ospf process** privileged EXEC mode command.



Passive Interface

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

```
R1(config)# router ospf 10
R1(config-router)# passive-interface GigabitEthernet 0/0
R1(config-router)# end
```

OSPF Metric = Cost

Recall that a 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. More overhead and time delays equal a higher cost. Therefore, a 10-Mb/s Ethernet line has a higher cost than a 100-Mb/s Ethernet line.

The formula used to calculate the OSPF cost is:

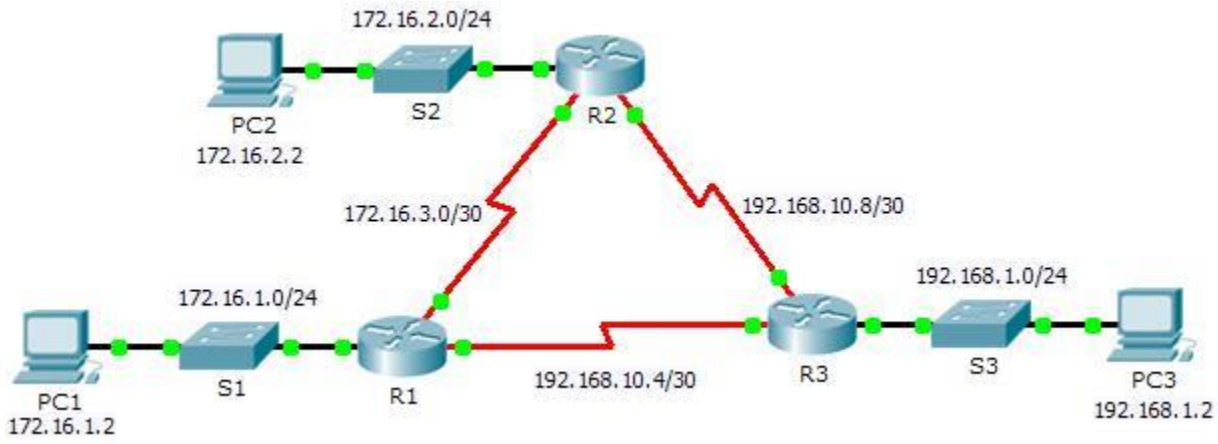
- **Cost** = *reference bandwidth* / *interface bandwidth*

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

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

Packet Tracer – Configuring OSPFv2 in a Single Area

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	172.16.1.1	255.255.255.0	N/A
	S0/0/0	172.16.3.1	255.255.255.252	N/A
	S0/0/1	192.168.10.5	255.255.255.252	N/A
R2	G0/0	172.16.2.1	255.255.255.0	N/A
	S0/0/0	172.16.3.2	255.255.255.252	N/A
	S0/0/1	192.168.10.9	255.255.255.252	N/A
R3	G0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	192.168.10.6	255.255.255.252	N/A
	S0/0/1	192.168.10.10	255.255.255.252	N/A
PC1	NIC	172.16.1.2	255.255.255.0	172.16.1.1
PC2	NIC	172.16.2.2	255.255.255.0	172.16.2.1
PC3	NIC	192.168.1.2	255.255.255.0	192.168.1.1

Objectives

Part 1: Configure OSPFv2 Routing

Part 2: Verify the Configurations

Background

In this activity, the IP addressing is already configured. You are responsible for configuring the three router topology with basic single area OSPFv2 and then verifying connectivity between end devices.

Part 1: Configure OSPFv2 Routing

Step 1: Configure OSPF on the R1, R2 and R3.

Use the following requirements to configure OSPF routing on all three routers:

- Process ID 10
- Router ID for each router: R1 = 1.1.1.1; R2 = 2.2.2.2; R3 = 3.3.3.3
- Network address for each interface
- LAN interface set to passive (do not use the **default** keyword)

Step 2: Verify OSPF routing is operational.

On each router, the routing table should now have a route to every network in the topology.

Part 2: Verify the Configurations

Each PC should be able to ping the other two PCs. If not, check your configurations.

Adjusting the Reference Bandwidth

OSPF uses a reference bandwidth of 100 Mb/s for any links that are equal to or faster than a fast Ethernet connection. Therefore, the cost assigned to a fast Ethernet interface with an interface bandwidth of 100 Mb/s would equal 1.
to adjust the costs for:

- **Gigabit Ethernet -auto-cost reference-bandwidth 1000**
- **10 Gigabit Ethernet -auto-cost reference-bandwidth 10000**

To return to the default reference bandwidth, use the **auto-cost reference-bandwidth 100** command.

Adjusting the Interface Bandwidth

To adjust the interface bandwidth use the **bandwidth** *kilobits* interface configuration command. Use the **no bandwidth** command to restore the default value.

The example in Figure adjusts the R1 Serial 0/0/1 interface bandwidth to 64 kb/s. A quick verification confirms that the interface bandwidth setting is now 64 kb/s.

```
R1(config)# int s0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# end
R1#
*Mar 27 10:10:07.735: %SYS-5-CONFIG_I: Configured from console by c
R1#
R1#  show interfaces serial 0/0/1 | include BW
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
R1#
R1#  show ip ospf interface serial 0/0/1 | include Cost:
  Process ID 10, Router ID 1.1.1.1, Network Type
  POINT_TO_POINT, Cost: 15625
R1#
```

Manually Setting the OSPF Cost

As an alternative to setting the default interface bandwidth, the cost can be manually configured on an interface using the **ip ospf cost** *value* interface configuration command.

```
R1(config)# int s0/0/1
R1(config-if)# no bandwidth 64
R1(config-if)# ip ospf cost 15625
R1(config-if)# end
R1#
R1# show interface serial 0/0/1 | include BW
    MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
R1#
R1# show ip ospf interface serial 0/0/1 | include Cost:
    Process ID 10, Router ID 1.1.1.1, Network Type POINT_TO_POINT,
    Cost: 15625
```

Verify OSPF Neighbors

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

```
R1# show ip ospf neighbor

Neighbor ID  Pri State  Dead Time  Address       Interface
3.3.3.3      0   FULL/- 00:00:37   192.168.10.6  Serial0/0/1
2.2.2.2      0   FULL/- 00:00:30   172.16.3.2    Serial0/0/0
R1#
```

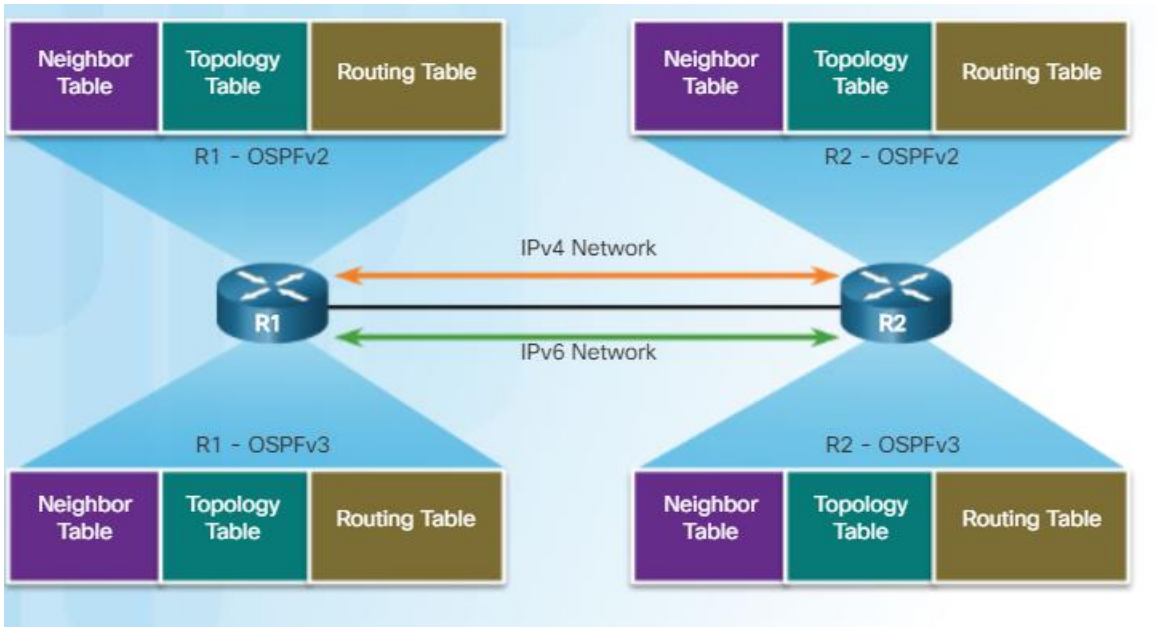
Verify OSPF Protocol Settings

The **show ip protocols** command is a quick way to verify vital OSPF configuration information. This includes the OSPFv2 process ID, the router ID, networks the router is advertising, the neighbors the router is receiving updates from, and the default administrative distance, which is 110 for OSPF.

OSPFv3

OSPFv3 is the OSPFv2 equivalent for exchanging IPv6 prefixes. Recall that in IPv6, the network address is referred to as the prefix and the subnet mask is called the prefix-length.

Similar to its IPv4 counterpart, OSPFv3 exchanges routing information to populate the IPv6 routing table with remote prefixes, as shown in the figure.



Similarities Between OSPFv2 to OSPFv3

OSPFv2 and OSPFv3	
Link-State	Yes
Routing Algorithm	SPF
Metric	Cost
Areas	Supports the same two-level hierarchy
Packet Types	Same Hello, DBD, LSR, LSU, and LSAck packets
Neighbor Discovery	Transitions through the same states using Hello packets
DR and BDR	Function and election process is the same
Router ID	32-bit router ID: determined by the same process in both protocols

Differences Between OSPFv2 and OSPFv3

Differeneces Between OSPFv2 and OSPFv3

	OSPFv2	OSPFv3
Advertises	IPv4 networks	IPv6 prefixes
Source Address	IPv4 source address	IPv6 link-local address
Destination Address	Choice of: • Neighbor IPv4 unicast address • 224.0.0.5 all-OSPF-routers multicast address • 224.0.0.6 DR/BDR multicast address	Choice of: • Neighbor IPv6 link-local address • FF02::5 all-OSPFv3-routers multicast address • FF02::6 DR/BDR multicast address
Advertise Networks	Configured using the `network` router configuration command	Configured using the `ipv6 ospf` *`process-id`* `area` *`area-id`* interface configuration command
IP Unicast Routing	IPv4 unicast routing is enabled by default.	IPv6 unicast forwarding is not enabled by default. The `ipv6 unicast-routing` global configuration command must be configured.
Authentication	Plain text and MD5	IPv6 authentication

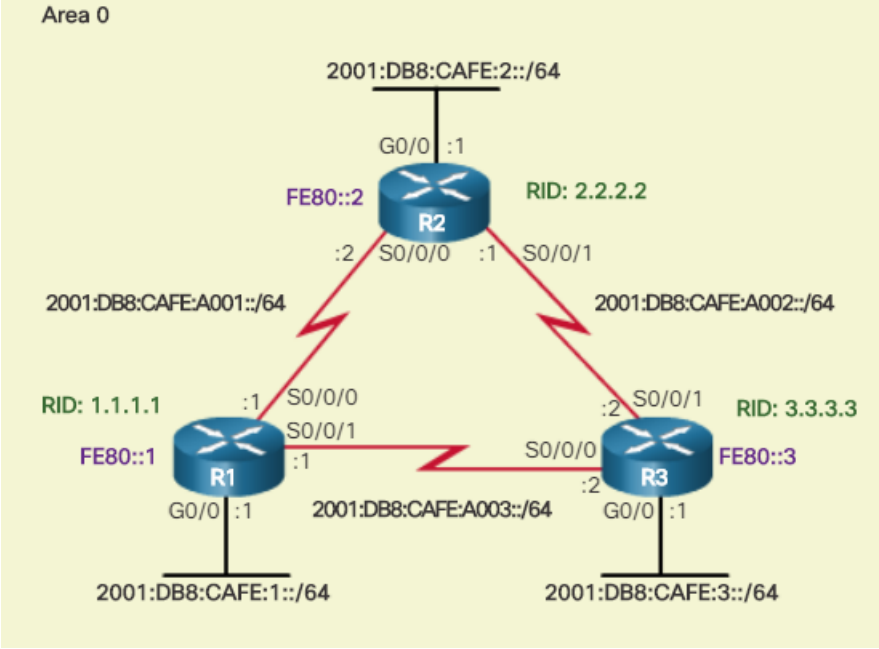
Link-Local Addresses

IPv6 link-local addresses are ideal for exchanging messages between neighbors on the same subnet or link.. An IPv6 link-local address enables a device to communicate with other IPv6-enabled devices on the same link and only on that link (subnet). Packets with a source or destination link-local address cannot be routed beyond the link from where the packet originated.

As shown in the figure, OSPFv3 messages are sent using:

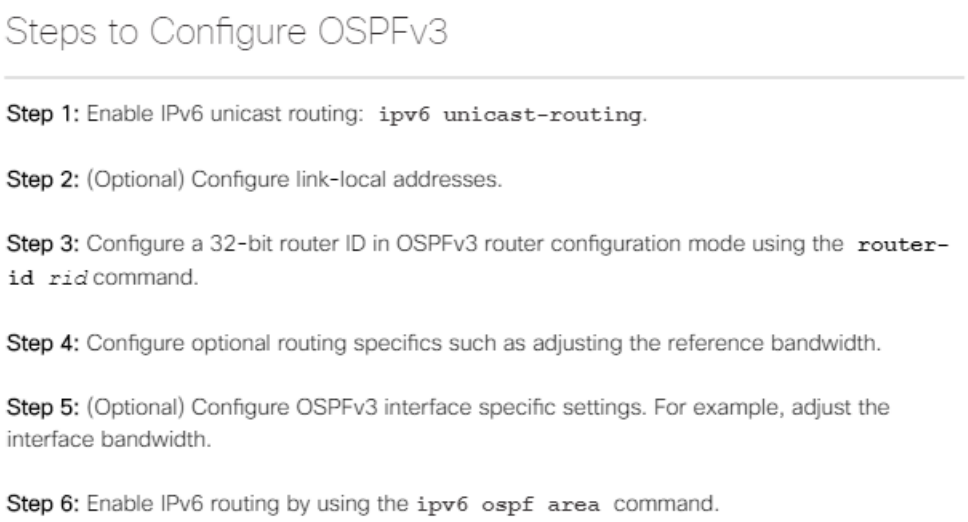
- **Source IPv6 address** - This is the IPv6 link-local address of the exit interface.
- **Destination IPv6 address** - OSPFv3 packets can be sent to a unicast address using the neighbor IPv6 link-local address. They can also be sent using a multicast address. The FF02::5 address is the all OSPF router address, while the FF02::6 is the DR/BDR multicast address.

OSPFv3 Network Topology



The FE80 address represents the link-local address assigned to each router.

```
R1(config)# ipv6 unicast-routing
R1(config)#
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# description R1 LAN
R1(config-if)# ipv6 address 2001:DB8:CAFE:1::1/64
R1(config-if)# no shut
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# description Link to R2
R1(config-if)# ipv6 address 2001:DB8:CAFE:A001::1/64
R1(config-if)# clock rate 128000
R1(config-if)# no shut
R1(config-if)#
R1(config-if)# interface Serial0/0/1
R1(config-if)# description Link to R3
R1(config-if)# ipv6 address 2001:DB8:CAFE:A003::1/64
R1(config-if)# no shut
R1(config-if)# end
R1#
```



Assigning Link-Local Addresses

Link-local addresses created using the EUI-64 format or in some cases, random interface IDs, make it difficult to recognize and remember those addresses. Because IPv6 routing protocols use IPv6 link-local addresses for unicast addressing and next-hop address information in the routing table, it is common practice to make it an easily recognizable address.

Configuring the link-local address manually provides the ability to create an address that is recognizable and easier to remember.

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# exit
R1(config)# interface Serial0/0/0
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# exit
R1(config)# interface Serial0/0/1
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)#
```

Configuring the OSPFv3 Router ID

Use the **ipv6 router ospf** *process-id* global configuration mode command to enter router configuration mode. The IPv6 router configuration mode prompt is different than the IPv4 router configuration mode prompt. Use the IPv6 router confirmation mode to configure global OSPFv3 parameters, such as a assigning a 32-bit OSPFv3 router ID and reference bandwidth.

```
R1(config)# ipv6 router ospf 10
R1(config-rtr)#
*Mar 29 11:21:53.739: %OSPFv3-4-NORTRID: Process OSPFv3-1-
IPv6 could not pick a router-id, please configure manually
R1(config-rtr)#
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)#
R1(config-rtr)# auto-cost reference-bandwidth 1000
% OSPFv3-1-IPv6:  Reference bandwidth is changed. Please
ensure reference bandwidth is consistent across all routers.
R1(config-rtr)#
R1(config-rtr)# end
R1#
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 10"
  Router ID 1.1.1.1
  Number of areas: 0 normal, 0 stub, 0 nssa
  Redistribution:
    None
```

Modifying an OSPFv3 Router ID

```
R1(config)# ipv6 router ospf 10
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)# end
R1#
```

```
R1# clear ipv6 ospf process
Reset selected OSPFv3 processes? [no]: y
R1#
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 10"
  Router ID 1.1.1.1
```

Enabling OSPFv3 on Interfaces

OSPFv3 uses a different method to enable an interface for OSPF. Instead of using the **network** router configuration mode command to specify matching interface addresses, OSPFv3 is configured directly on the interface.

To enable OSPFv3 on an interface, use the **ipv6 ospf** *process-id* **area** *area-id* interface configuration mode command.

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/1
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# end
R1#
R1# show ipv6 ospf interfaces brief
Interface  PID   Area     Intf ID  Cost   State  Nbrs F/C
Se0/0/1    10     0         7      15625  P2P    0/0
Se0/0/0    10     0         6      647    P2P    0/0
Gi0/0      10     0         3       1     WAIT   0/0
R1#
```

Verify OSPFv3 Neighbors

Use the **show ipv6 ospf neighbor** command to verify that the router has formed an adjacency with its neighboring routers. If the router ID of the neighboring router is not displayed, or if it does not show as being in a state of FULL, the two routers have not formed an OSPFv3 adjacency.

Verify OSPFv3 Protocol Settings

show ipv6 protocols command is a quick way to verify vital OSPFv3 configuration information, including the OSPFv3 process ID, the router ID, and the interfaces enabled for OSPFv3.

Verify OSPFv3 Interfaces

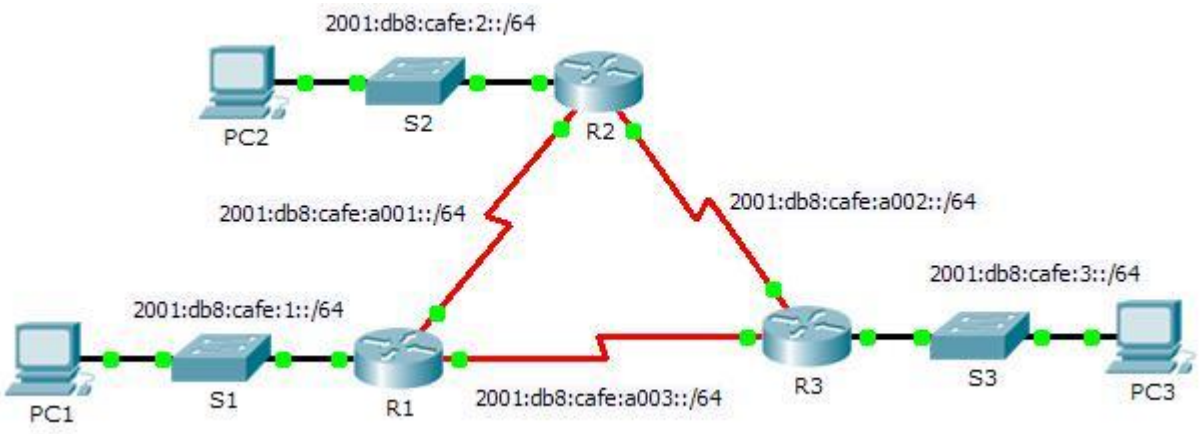
To retrieve and view a summary of OSPFv3-enabled interfaces on R1, use the **show ipv6 ospf interface brief** command

Verify the IPv6 Routing Table

show ipv6 route ospf command provides specifics about OSPFv3 routes in the routing table.

Packet Tracer – Configuring Basic OSPFv3 in a Single Area

Topology



Addressing Table

Device	Interface	IPv6 Address/Prefix	Default Gateway
R1	G0/0	2001:db8:cafe:1::1/64	N/A
	S0/0/0	2001:db8:cafe:a001::1/64	N/A
	S0/0/1	2001:db8:cafe:a003::1/64	N/A
R2	G0/0	2001:db8:cafe:2::1/64	N/A
	S0/0/0	2001:db8:cafe:a001::2/64	N/A
	S0/0/1	2001:db8:cafe:a002::1/64	N/A
R3	G0/0	2001:db8:cafe:3::1/64	N/A
	S0/0/0	2001:db8:cafe:a003::264	N/A
	S0/0/1	2001:db8:cafe:a002::2/64	N/A
PC1	NIC	2001:db8:cafe:1::10/64	fe80::1
PC2	NIC	2001:db8:cafe:2::10/64	fe80::2
PC3	NIC	2001:db8:cafe:3::10/64	fe80::3

Objectives

Part 1: Configure OSPFv3 Routing

Part 2: Verify Connectivity

Background

In this activity, the IPv6 addressing is already configured. You are responsible for configuring the three router topology with basic single area OSPFv3 and then verifying connectivity between end devices.

Part 1: Configure OSPFv3 Routing

Step 1: Configure OSPFv3 on R1, R2 and R3.

Use the following requirements to configure OSPF routing on all three routers:

- Enable IPv6 routing
- Process ID 10
- Router ID for each router: R1 = 1.1.1.1; R2 = 2.2.2.2; R3 = 3.3.3.3
- Enable OSPFv3 on each interface

Note: Packet Trace version 6.0.1 does not support the **auto-cost reference-bandwidth** command, so you will not be adjust bandwidth costs in this activity.

Step 2: Verify OSPF routing is operational.

Verify each router has established adjacency with the other two routers. Verify the routing table has a route to every network in the topology.

Part 2: Verify Connectivity

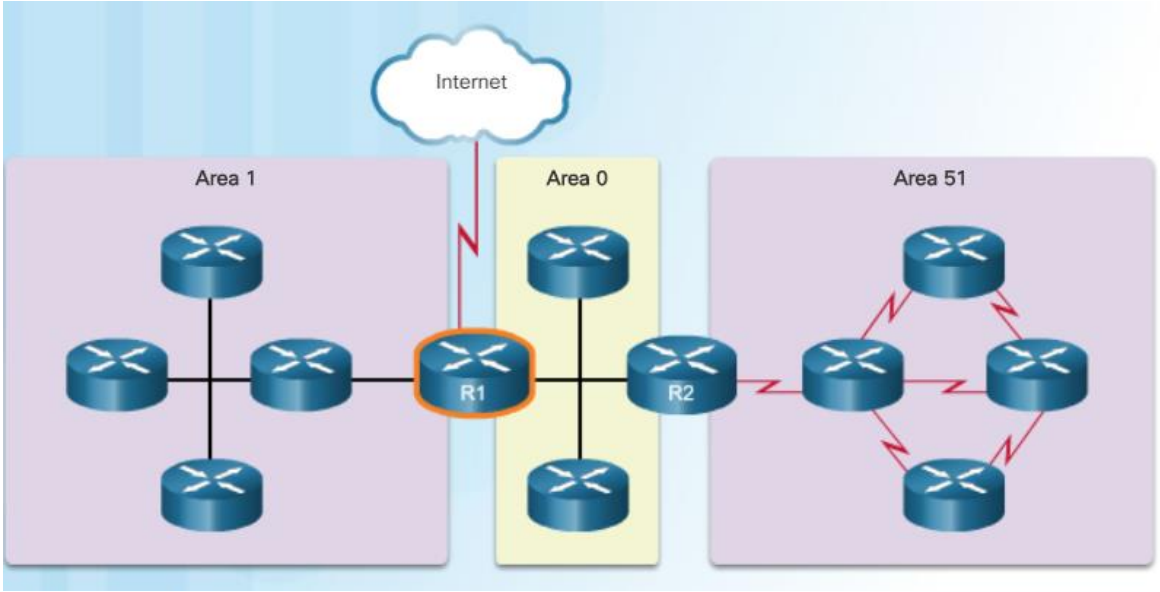
Each PC should be able to ping the other two PCs. If not, check your configurations.

Note: This activity is graded using only connectivity tests. The instructions window will not show your score. To see your score, click **Check Results** > **Assessment Items**. To see the results of a specific connectivity test, click **Check Results** > **Connectivity Tests**.

Lab 5: Multiarea OSPF

Multiarea OSPF is used to divide a large OSPF network. Too many routers in one area increase the load on the CPU and create a large link-state database. In this chapter, directions are provided to effectively partition a large single area into multiple areas. Area 0, used in a single-area OSPF, is known as the backbone area.

OSPF Two-Layer Area Hierarchy



Multiarea OSPF is implemented in a two-layer area hierarchy:

- Backbone (Transit) area
- Regular (Non-backbone) area

Note: A regular area can have a number of subtypes, including a standard area, stub area, totally stubby area, and not-so-stubby area (NSSA). Stub, totally stubby, and NSSAs are beyond the scope of this lab.

OSPF enforces this rigid two-layer area hierarchy. The underlying physical connectivity of the network must map to the two-layer area structure, with all non-backbone areas attaching directly to area 0. All traffic moving from one area to another area must traverse the backbone area. This traffic is referred to as interarea traffic.

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

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

the hierarchical-topology possibilities of multiarea OSPF have these advantages:

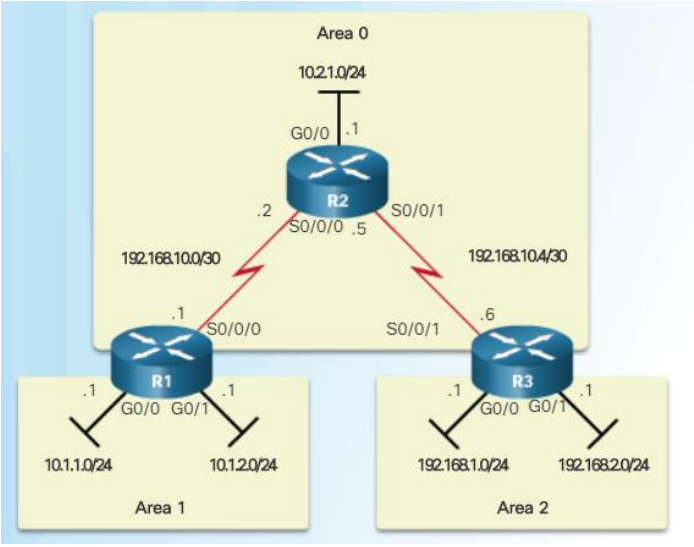
- Smaller routing tables
- Reduced link-state update
- Reduced frequency of SPF calculations

Types of OSPF Routers

There are four different types of OSPF routers:

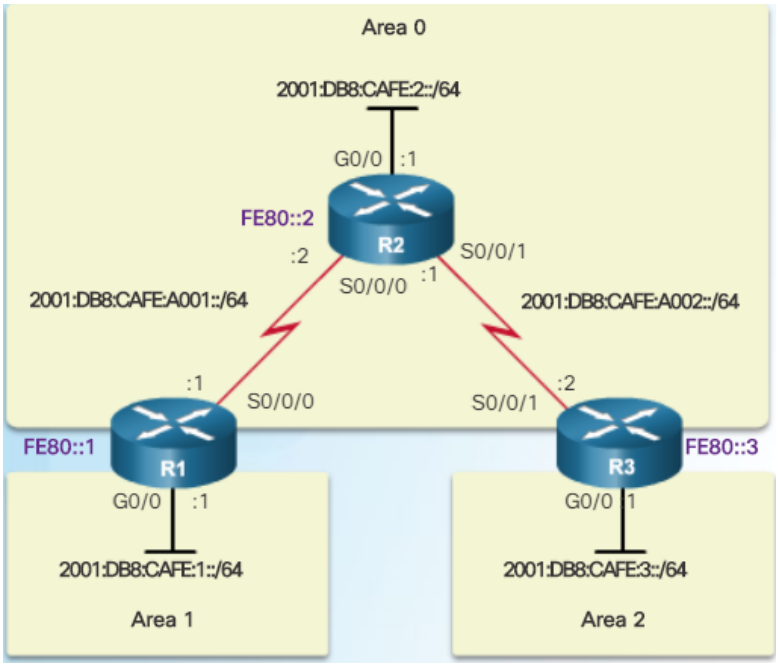
- **Internal router** – This is a router that has all of its interfaces in the same area.
- **Backbone router** – This is a router in the backbone area. The backbone area is set to area 0.
- **Area Border Router (ABR)** – This is a router that has interfaces attached to multiple areas. It must maintain separate LSDBs for each area it is connected to, and can route between areas. ABRs are exit points for the area, which means that routing information destined for another area can get there only via the ABR of the local area. ABRs can be configured to summarize the routing information from the LSDBs of their attached areas. ABRs distribute the routing information into the backbone.
- **Autonomous System Boundary Router (ASBR)** – This is a router that has at least one interface attached to an external internetwork. An external network is a network that is not part of this OSPF routing domain. For example, a network connection to an ISP. An ASBR can import external network information to the OSPF network, and vice versa, using a process called route redistribution. Redistribution in multiarea OSPF occurs when an ASBR connects different routing domains (e.g., EIGRP and OSPF) and configures them to exchange and advertise routing information between those routing domains. A static route, including a default route, can also be redistributed as an external route into the OSPF routing domain.

Configuring Multiarea OSPFv2




```
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.1.1.1 0.0.0.0 area 1
R1(config-router)# network 10.1.2.1 0.0.0.0 area 1
R1(config-router)# network 192.168.10.1 0.0.0.0 area 0
R1(config-router)# end
R1#
```

Configuring Multiarea OSPFv3




```
R1(config)# ipv6 router ospf 10
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)# exit
R1(config)#
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ipv6 ospf 10 area 1
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)# end
R1#
```

Verifying Multiarea OSPFv2 and OSPFv3

The same verification commands used to verify single-area OSPFv2 also can be used to verify the multiarea OSPF topology in the figure:

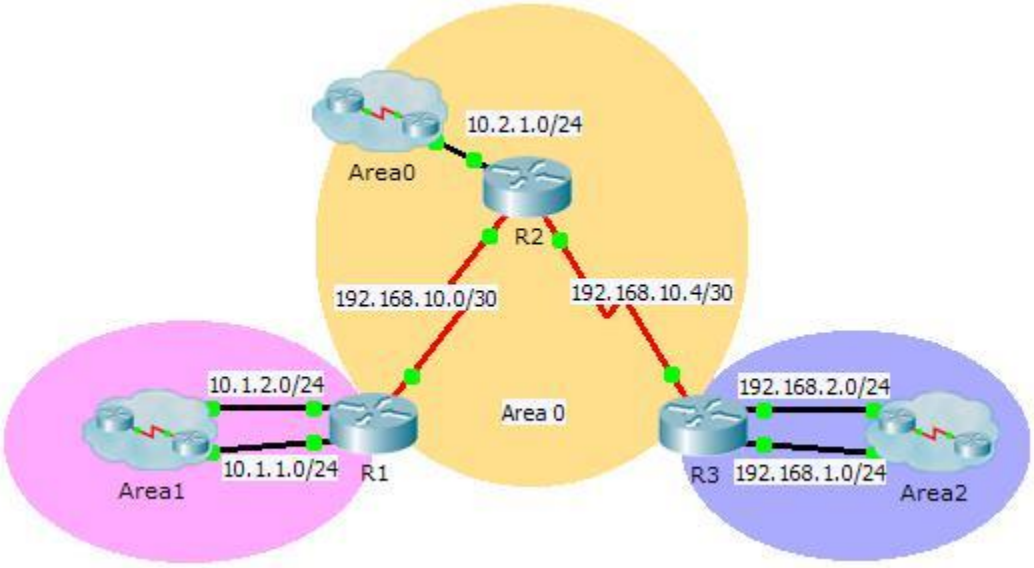
- **show ip ospf neighbor**
- **show ip ospf**
- **show ip ospf interface**

Commands that verify specific multiarea OSPFv2 information include:

- **show ip protocols**
- **show ip ospf interface brief**
- **show ip route ospf**
- **show ip ospf database**
  for IPv6 use the same command but change IP to IPv6

Packet Tracer – Configuring Multiarea OSPFv2

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	OSPFv2 Area
R1	G0/0	10.1.1.1	255.255.255.0	1
	G0/1	10.1.2.1	255.255.255.0	1
	S0/0/0	192.168.10.2	255.255.255.252	0
R2	G0/0	10.2.1.1	255.255.255.0	0
	S0/0/0	192.168.10.1	255.255.255.252	0
	S0/0/1	192.168.10.5	255.255.255.252	0
R3	G0/0	192.168.2.1	255.255.255.0	2
	G0/1	192.168.1.1	255.255.255.0	2
	S0/0/1	192.168.10.6	255.255.255.252	0

Objectives

Part 1: Configure Multiarea OSPFv2

Part 2: Verify and Examine Multiarea OSPFv2

Background

In this activity, you will configure multiarea OSPFv2. The network is already connected and interfaces are configured with IPv4 addressing. Your job is to enable multiarea OSPFv2, verify connectivity, and examine the operation of multiarea OSPFv2.

Part 1: Configure OSPFv2

Step 1: Configure OSPFv2 on R1.

Configure OSPFv2 on R1 with a process ID of 1 and a router ID of 1.1.1.1.

Step 2: Advertise each directly connected network in OSPFv2 on R1.

Configure each network in OSPFv2 assigning areas according to the **Addressing Table**.

```
R1(config-router)# network 10.1.1.0 0.0.0.255 area 1
R1(config-router)# network 10.1.2.0 0.0.0.255 area 1
R1(config-router)# network 192.168.10.0 0.0.0.3 area 0
```

Step 3: Configure OSPFv2 on R2 and R3.

Repeat the steps above for **R2** and **R3** using a router ID of 2.2.2.2 and 3.3.3.3, respectively.

Part 2: Verify and Examine Multiarea OSPFv2

Step 1: Verify connectivity to each of the OSPFv2 areas.

From R1, ping each of the following remote devices in area 0 and area 2: 192.168.1.2, 192.168.2.2, and 10.2.1.2.

Step 2: Use show commands to examine the current OSPFv2 operations.

Use the following commands to gather information about your OSPFv2 multiarea implementation.

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

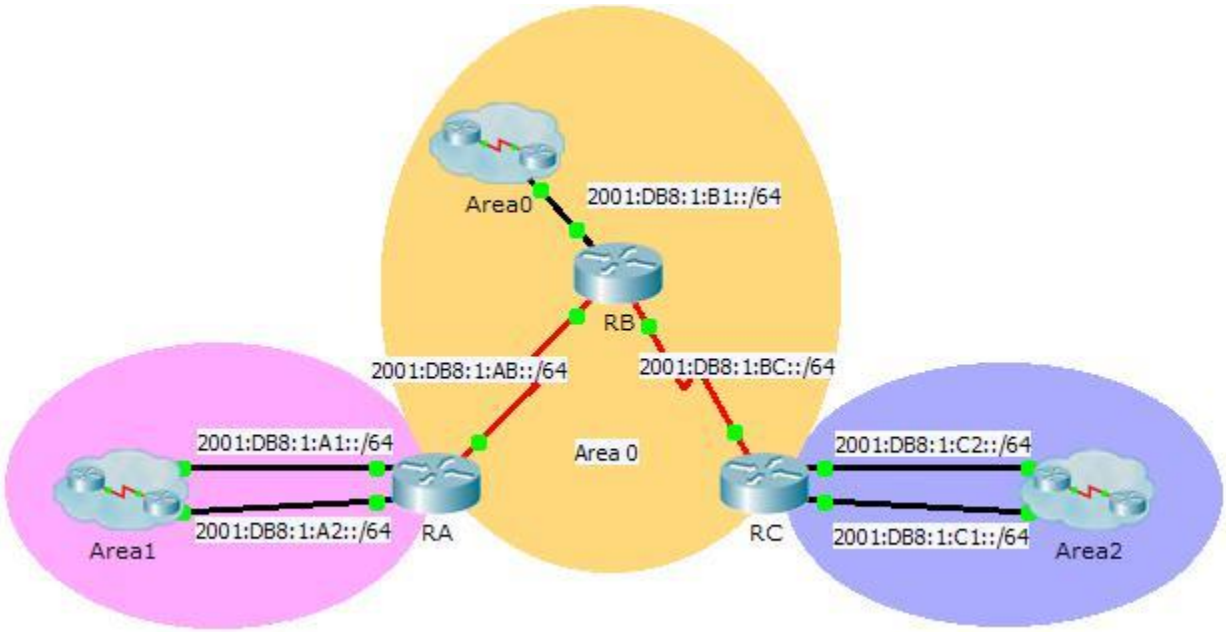
Reflection Questions

- Which router(s) are internal routers?
- Which router(s) are backbone routers?
- Which router(s) are area border routers?
- Which router(s) are autonomous system routers?
- Which routers are generating Type 1 LSAs?
- Which routers are generating Type 2 LSAs?
- Which routers are generating Type 3 LSAs?
- Which routers are generating Type 4 and 5 LSAs?
- How many inter area routes does each router have?

10. Why would there usually be an ASBR in this type of network?

Packet Tracer – Configuring Multiarea OSPFv3

Topology



Addressing Table

Device	Interface	IPv6 Address	OSPF Area
RA	G0/0	2001:DB8:1:A1::1/64	1
	G0/1	2001:DB8:1:A2::1/64	1
	S0/0/0	2001:DB8:1:AB::2/64	0
	Link-Local	FE80::A	N/A
RB	G0/0	2001:DB8:1:B1::1/64	0
	S0/0/0	2001:DB8:1:AB::1/64	0
	S0/0/1	2001:DB8:1:BC::1/64	0
	Link-Local	FE80::B	N/A
RC	G0/0	2001:DB8:1:C1::1/64	2
	G0/1	2001:DB8:1:C2::1/64	2
	S0/0/1	2001:DB8:1:BC::2/64	0
	Link-Local	FE80::C	N/A

Objectives

Part 1: Configure OSPFv3

Part 2: Verify Multiarea OSPFv3 Operations

Background

In this activity, you will configure multiarea OSPFv3. The network is already connected and interfaces are configured with IPv6 addressing. Your job is to enable multiarea OSPFv3, verify connectivity and examine the operation of multiarea OSPFv3.

Part 1: Configure OSPFv3

Step 1: Enable IPv6 routing and configure OSPFv3 on RA.

- Enable IPv6 routing.
- Configure OSPFv3 on RA with a process ID of 1 and a router ID of 1.1.1.1.

Step 2: Advertise each directly connected network in OSPFv3 on RA.

Configure each active IPv6 interface with OSPFv3 assigning each to the area listed in the **Addressing Table**.

Step 3: Configure OSPFv3 on RB and RC

Repeat the Steps 1 and 2 for **RB** and **RC**, changing the router ID to 2.2.2.2 and 3.3.3.3 respectively.

Part 2: Verify Multiarea OSPFv3 Operations

Step 1: Verify connectivity to each of the OSPFv3 areas.

From RA, ping each of the following remote devices in area 0 and area 2: 2001:DB8:1:B1::2, 2001:DB8:1:A1::2, 2001:DB8:1:A2::2, 2001:DB8:1:C1::2, and 2001:DB8:1:C2::2.

Step 2: Use show commands to examine the current OSPFv3 operations.

Use the following commands to gather information about your OSPFv3 multiarea implementation.

```
show ipv6 ospf
show ipv6 route
show ipv6 ospf database
show ipv6 ospf interface
show ipv6 ospf neighbor
```

Note: Packet Tracer output for **show ipv6 protocols** is currently not aligned with IOS 15 output. Refer to the real equipment labs for correct **show** command output.

OSPF Network Types

OSPF defines five network types:

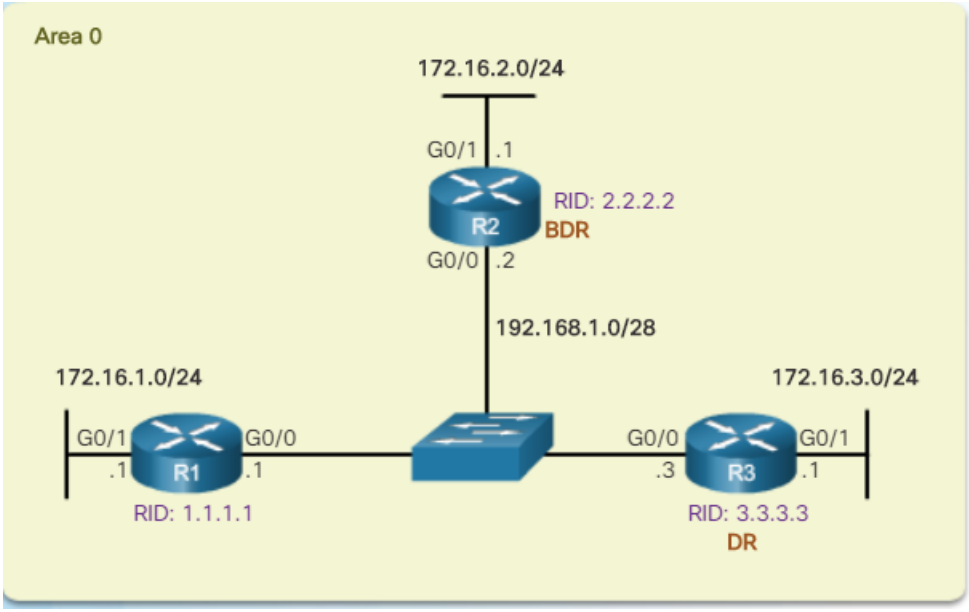
- **Point-to-point** - Two routers interconnected over a common link. No other routers are on the link.
- **Broadcast multiaccess** - Multiple routers interconnected over an Ethernet network
- **Nonbroadcast multiaccess (NBMA)** - Multiple routers interconnected in a network that does not allow broadcasts, such as Frame Relay.
- **Point-to-multipoint** - Multiple routers interconnected in a hub-and-spoke topology over an NBMA network. Often used to connect branch sites (spokes) to a central site (hub).
- **Virtual links** - Special OSPF network used to interconnect distant OSPF areas to the backbone area.

OSPF Designated Router

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. The BDR listens passively to this exchange and maintains a relationship with all the routers. If the DR stops producing Hello packets, the BDR promotes itself and assumes the role of DR.

All other non-DR or BDR routers become DROTHER (a router that is neither the DR nor the BDR).

Verifying DR/BDR Roles



In the multiaccess topology shown in Figure above, there are three routers interconnected over a common Ethernet multiaccess network, 192.168.1.0/28. Each router is configured with the indicated IPv4 address on the Gigabit Ethernet 0/0 interface.

Because the routers are connected over a common multiaccess broadcast network, OSPF has automatically elected a DR and BDR. In this example, R3 has been elected as the DR because its router ID is 3.3.3.3, which is the highest in this network. R2 is the BDR because it has the second highest router ID in the network.

To verify the roles of the OSPFv2 router, use the **show ip ospf interface** command. The output generated by R1 confirms that:

1. R1 is not the DR or BDR, but is a DROTHER with a default priority of 1. (1)

2. The DR is R3 with router ID 3.3.3.3 at IPv4 address 192.168.1.3, while the BDR is R2 with router ID 2.2.2.2 at IPv4 address 192.168.1.2. (2)

3. R1 has two adjacencies: one with the BDR and one with the DR. (3)

Note: For the equivalent OSPFv3 command, simply substitute **ip** with **ipv6**.

Verifying DR/BDR Adjacencies

To verify the OSPFv2 adjacencies, use the**show ip ospf neighbor** command

Default DR/BDR Election Process

How do the DR and BDR get elected? The OSPF DR and BDR election decision is based on the following criteria, in sequential order:
1. The routers in the network elect the router with the highest interface priority as the DR. The router with the second highest interface priority is elected as the BDR. The priority can be configured to be any number between 0 – 255. The higher the priority, the likelier the router will be selected as the DR. If the priority is set to 0, the router is not capable of becoming the DR. The default priority of multiaccess broadcast interfaces is 1. Therefore, unless otherwise configured, all routers have an equal priority value and must rely on another tie breaking method during the DR/BDR election.
2. If the interface priorities are equal, then the router with the highest router ID is elected the DR. The router with the second highest router ID is the BDR.
Recall that the router ID is determined in one of three ways:

- The router ID can be manually configured.
- If no router IDs are configured, the router ID is determined by the highest loopback IPv4 address.
- If no loopback interfaces are configured, the router ID is determined by the highest active IPv4 address.

Note: In an IPv6 network, if there are no IPv4 addresses configured on the router, then the router ID must be manually configured with the **router-id** *rid*command; otherwise, OSPFv3 does not start.

The OSPF Priority

The DR becomes the focal point for the collection and distribution of LSAs; therefore, this router must have sufficient CPU and memory capacity to handle the workload. It is possible to influence the DR/BDR election process through configurations.
To set the priority of an interface, use the following commands:

- **ip ospf priority** *value* - OSPFv2 interface command
- **ipv6 ospf priority** *value* - OSPFv3 interface command

The *value* can be:

- **0 -** Does not become a DR or BDR.
- **1** – **255 -** The higher the priority value, the more likely the router becomes the DR or BDR on the interface.

Changing the OSPF Priority

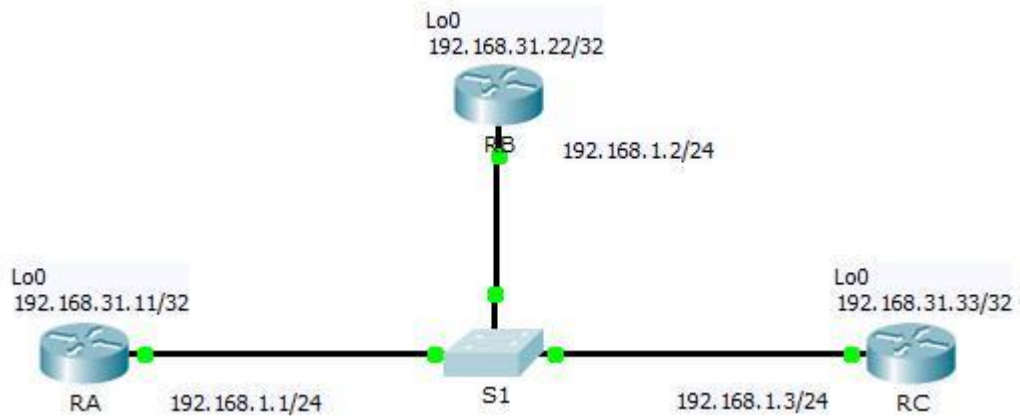
```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip ospf priority 255
R1(config-if)# end
R1#
```

```
R3(config)# interface GigabitEthernet 0/0
R3(config-if)# ip ospf priority 0
R3(config-if)# end
R3#
```

```
R1# clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R1#
*Apr 6 16:00:44.282: %OSPF-5-ADJCHG: Process 10, Nbr
```

Packet Tracer - Determining the DR and BDR

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask
RA	G0/0	192.168.1.1	255.255.255.0
	Lo0	192.168.31.11	255.255.255.255
RB	G0/0	192.168.1.2	255.255.255.0
	Lo0	192.168.31.22	255.255.255.255
RC	G0/0	192.168.1.3	255.255.255.0
	Lo0	192.168.31.33	255.255.255.255

Objectives

Part 1: Examine DR and BDR Changing Roles

Part 2: Modify OSPF Priority and Force Elections

Scenario

In this activity, you will examine DR and BDR roles and watch the roles change when there is a change in the network. You will then modify the priority to control the roles and force a new election. Finally, you will verify routers are filling the desired roles.

Part 1: Examine DR and BDR Changing Roles

Step 1: Wait until the amber link lights turn green.

When you first open the file in Packet Tracer, you may notice that the link lights for the switch are amber. These link lights will stay amber for 50 seconds while the switch makes sure that one of the routers is not another switch. Alternatively, you can click **Fast Forward Time** to bypass this process.

Step 2: Verify the current OSPF neighbor states.

a. Use the appropriate command on each router to examine the current DR and BDR.

b. Which router is the DR?

c. Which router is the BDR?

Step 3: Turn on IP OSPF adjacency debugging.

a. You can monitor the DR and BDR election process with a **debug** command. On **RA** and **RB**, enter the following command.

```
RA# debug ip ospf adj
RB# debug ip ospf adj
```

Step 4: Disable the Gigabit Ethernet 0/0 interface on RC.

a. Disable the link between **RC** and the switch to cause roles to change.

b. Wait about 30 seconds for the dead timers to expire on **RA** and **RB**. According to the debug output, which router was elected DR and which router was elected BDR?

Step 5: Restore the Gigabit Ethernet 0/0 interface on RC.

a. Re-enable the link between **RC** and the switch.

b. Wait for the new DR/BDR elections to occur. Did DR and BDR roles change? Why or why not?

Step 6: Disable the Gigabit Ethernet 0/0 interface on RB.

- Disable the link between **RB** and the switch to cause roles to change.
- Wait about 30 seconds for the holddown timers to expire on **RA** and **RC**. According to the debug output on **RA**, which router was elected DR and which router was elected BDR?

Step 7: Restore the Gigabit Ethernet 0/0 interface on RB.

- Re-enable the link between **RB** and the switch.
- Wait for the new DR/BDR elections to occur. Did DR and BDR roles change? Why or why not?

Step 8: Turn off Debugging.

Enter the command **undebug all** on **RA** and **RB** to disable debugging.

Part 2: Modify OSPF Priority and Force Elections

Step 1: Configure OSPF priorities on each router.

To change the DR and BDR, configure the Gigabit Ethernet 0/0 port of each router with the following OSPF interface priorities:

RA: 200

RB: 100

RC: 1 (This is the default priority)

Step 2: Force an election by reloading the switch.

Note: The command **clear ip ospf process** can also be used on the routers to reset the OSPF process.

Step 3: Verify DR and BDR elections were successful.

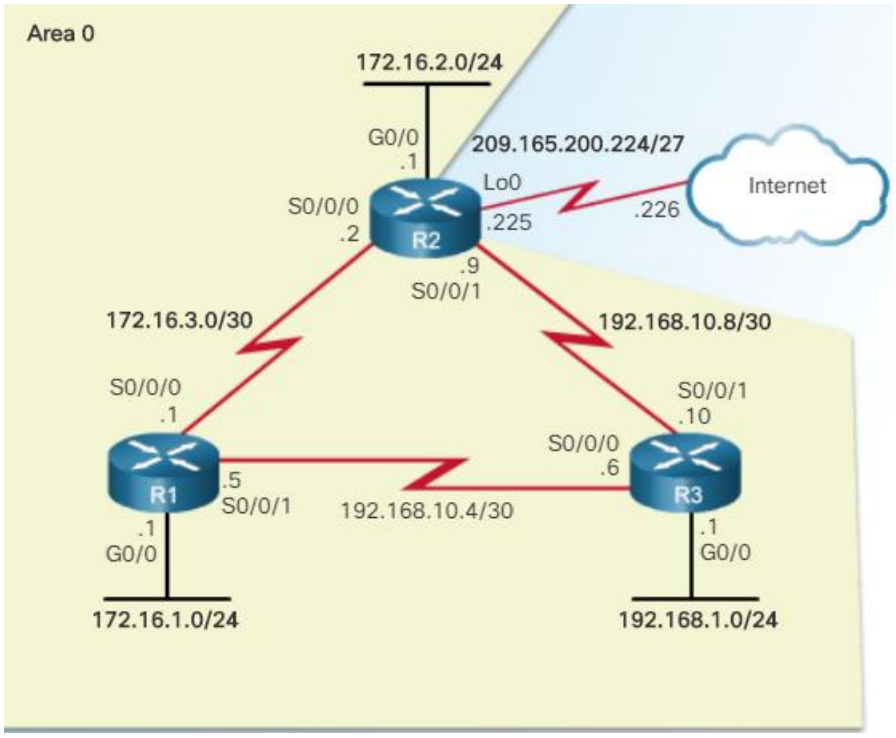
f. Wait long enough for OSPF to converge and for the DR/BDR election to occur. This should take a few minutes. You can click **Fast Forward Time** to speed up the process.

g. According to output from an appropriate command, which router is now DR and which router is now BDR?

Propagating a Default Static Route in OSPFv2

To propagate a default route, the edge router (R2) must be configured with:

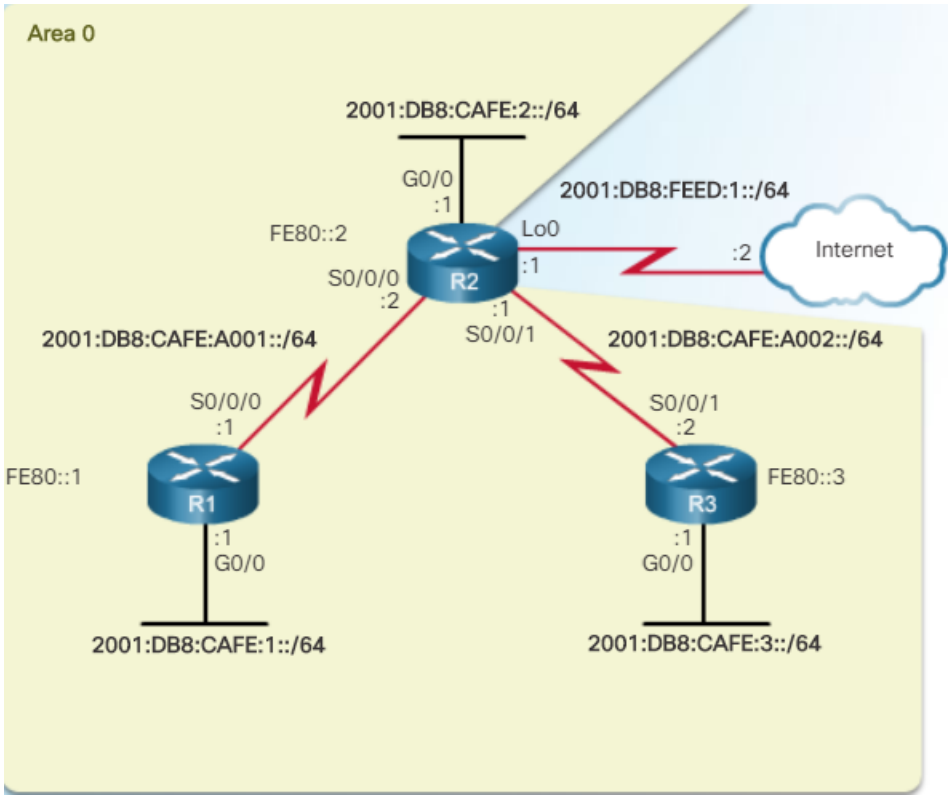
- A default static route using the **ip route 0.0.0.0 0.0.0.0** {*ip-address* | *exit-intf*} command.
- The **default-information originate** router configuration mode command. This instructs R2 to be the source of the default route information and propagate the default static route in OSPF updates




```
R2(config)# ip route 0.0.0.0 0.0.0.0 209.165.200.226
R2(config)#
R2(config)# router ospf 10
R2(config-router)# default-information originate
R2(config-router)# end
R2#
```

Propagating a Default Static Route in OSPFv3

- A default static route using the **ipv6 route ::/0** {*ipv6-address* | *exit-intf*} command.
- The **default-information originate** router configuration mode command. This instructs R2 to be the source of the default route information and propagate the default static route in OSPF updates.




```
R2(config)# ipv6 route ::/0 2001:DB8:FEED:1::2
R2(config)#
R2(config)# ipv6 router ospf 10
R2(config-rtr)# default-information originate
R2(config-rtr)# end
R2#
```

OSPF Hello and Dead Intervals

The OSPF Hello and Dead intervals are configurable on a per-interface basis. The OSPF intervals must match or a neighbor adjacency does not occur

Modifying OSPFv2 Intervals

It may be desirable to change the OSPF timers so that routers detect network failures in less time. Doing this increases traffic, but sometimes the need for quick convergence is more important than the extra traffic it creates.

Note: The default Hello and Dead intervals are based on best practices and should only be altered in rare situations.

OSPFv2 Hello and Dead intervals can be modified manually using the following interface configuration mode commands:

- **ip ospf hello-interval** *seconds*
- **ip ospf dead-interval** *seconds*

Use the **no ip ospf hello-interval** and **no ip ospf dead-interval** commands to reset the intervals to their default.

```
R1(config)# interface serial 0/0/0
R1(config-if)# ip ospf hello-interval 5
R1(config-if)# ip ospf dead-interval 20
R1(config-if)# end
R1#
```

Modifying OSPFv3 Intervals

Like OSPFv2, OSPFv3 intervals can also be adjusted.

OSPFv3 Hello and Dead intervals can be modified manually using the following interface configuration mode commands:

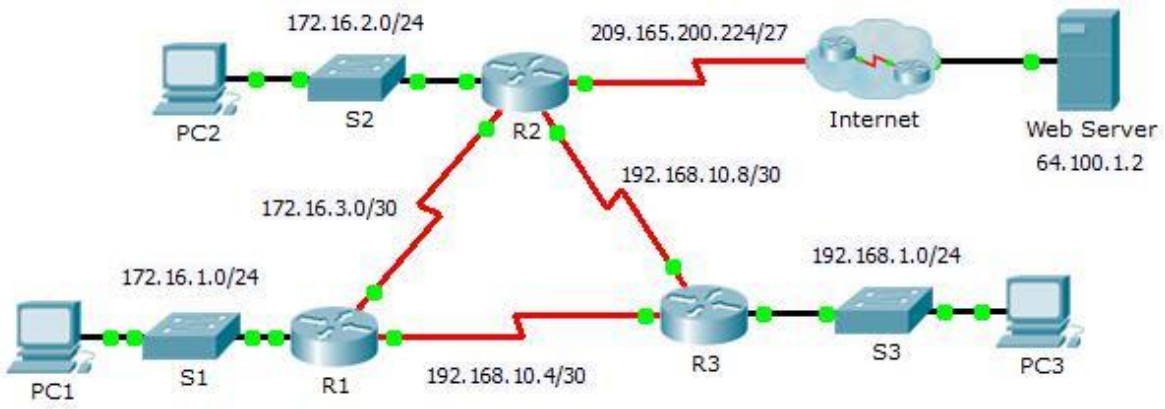
- **ipv6 ospf hello-interval** *seconds*
- **ipv6 ospf dead-interval** *seconds*

Note: Use the **no ipv6 ospf hello-interval** and **no ipv6 ospf dead-interval** commands to reset the intervals to their default.

```
R1(config)# interface serial 0/0/0
R1(config-if)# ipv6 ospf hello-interval 5
R1(config-if)# ipv6 ospf dead-interval 20
R1(config-if)# end
R1#
```

Packet Tracer - Propagating a Default Route in OSPFv2 And Configuring OSPF Advanced Features

Topology



Addressing Table

Device	Interface	IPv4 Address	Subnet Mask	Default Gateway
R1	G0/0	172.16.1.1	255.255.255.0	N/A
	S0/0/0	172.16.3.1	255.255.255.252	N/A
	S0/0/1	192.168.10.5	255.255.255.252	N/A
R2	G0/0	172.16.2.1	255.255.255.0	N/A
	S0/0/0	172.16.3.2	255.255.255.252	N/A
	S0/0/1	192.168.10.9	255.255.255.252	N/A
	S0/1/0	209.165.200.225	255.255.255.224	N/A
R3	G0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	192.168.10.6	255.255.255.252	N/A
	S0/0/1	192.168.10.10	255.255.255.252	N/A
PC1	NIC	172.16.1.2	255.255.255.0	172.16.1.1
PC2	NIC	172.16.2.2	255.255.255.0	172.16.2.1
PC3	NIC	192.168.1.2	255.255.255.0	192.168.1.1

Objectives

Part 1: Propagate a Default Route

Part 2: Verify Connectivity
Part 3: Modify OSPF Default Settings

Background

In this activity, you will configure an IPv4 default route to the Internet and propagate that default route to other OSPF routers. You will then verify the default route is in downstream routing tables and that hosts can now access a web server on the Internet.

Part 1: Propagate a Default Route

Step 1: Configure a default route on R2.

Configure **R2** with a directly attached default route to the Internet.

```
R2(config)# ip route 0.0.0.0 0.0.0.0 Serial0/1/0
```

Step 2: Propagate the route in OSPF.

Configure OSPF to propagate the default route in OSPF routing updates.

```
R2(config-router)# default-information originate
```

Step 3: Examine the routing tables on R1 and R3.

Examine the routing tables of **R1** and **R3** to verify that the route has been propagated.

```
R1> show ip route
<output omitted>
O*E2 0.0.0.0/0 [110/1] via 172.16.3.2, 00:00:08,
Serial0/0/0 !-------------------
R3> show ip route
<output omitted>
O*E2 0.0.0.0/0 [110/1] via 192.168.10.9, 00:08:15, Serial0/0/1
```

Part 2: Verify Connectivity

Verify that **PC1**, **PC2**, and **PC3** can ping the web server.

Part 3: Modify OSPF Default Settings

Step 1: Test connectivity between all end devices.

Before modifying the OSPF settings, verify that all PCs can ping the web server and each other.

Step 2: Adjust the hello and dead timers between R1 and R2.

- Enter the following commands on **R1**.

```
R1(config)# interface s0/0/0 R1(config-
if)# ip ospf hello-interval 15
R1(config-if)# ip ospf dead-interval 60
```

- After a short period of time, the OSPF connection with **R2** will fail. Both sides of the connection need to have the same timers in order for the adjacency to be maintained. Adjust the timers on **R2**.

Step 3: Adjust the bandwidth setting on R1.

a. Trace the path between **PC1** and the web server located at 64.100.1.2. Notice that the path from **PC1** to 64.100.1.2 is routed through **R2**. OSPF prefers the lower cost path.

b. On the **R1** Serial 0/0/0 interface, set the bandwidth to 64 Kb/s. This does not change the actual port speed, only the metric that the OSPF process on **R1** will use to calculate best routes.

```
R1(config-if)# bandwidth 64
```

c. Trace the path between **PC1** and the web server located at 64.100.1.2. Notice that the path from **PC1** to 64.100.1.2 is redirected through **R3**. OSPF prefers the lower cost path.

Part 4: Verify Connectivity

Verify all PCs can ping the web server and each other.

Lab6: VLANs and inter- VLAN routing

Network performance is an important factor in the productivity of an organization. One of the technologies used to improve network performance is the separation of large broadcast domains into smaller ones. By design, routers will block broadcast traffic at an interface. However, routers normally have a limited number of LAN interfaces. A router's primary role is to move information between networks, not to provide network access to end devices.

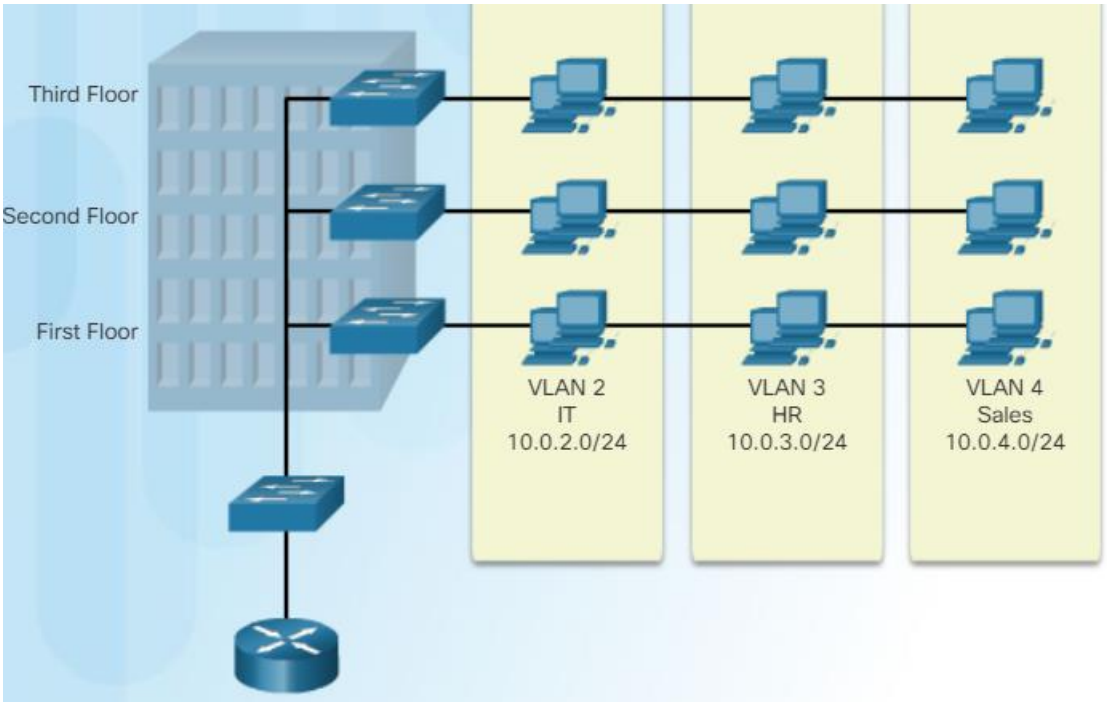
The role of providing access into a LAN is normally reserved for an access layer switch. A virtual local area network (VLAN) can be created on a Layer 2 switch to reduce the size of broadcast domains, similar to a Layer 3 device. VLANs are commonly incorporated into network design making it easier for a network to support the goals of an organization. While VLANs are primarily used within switched local area networks, modern implementations of VLANs allow them to span MANs and WANs.

Because VLANs segment the network, a Layer 3 process is required to allow traffic to move from one network segment to another.

This Layer 3 routing process can either be implemented using a router or a Layer 3 switch interface. The use of a Layer 3 device provides a method for controlling the flow of traffic between network segments, including network segments created by VLANs.

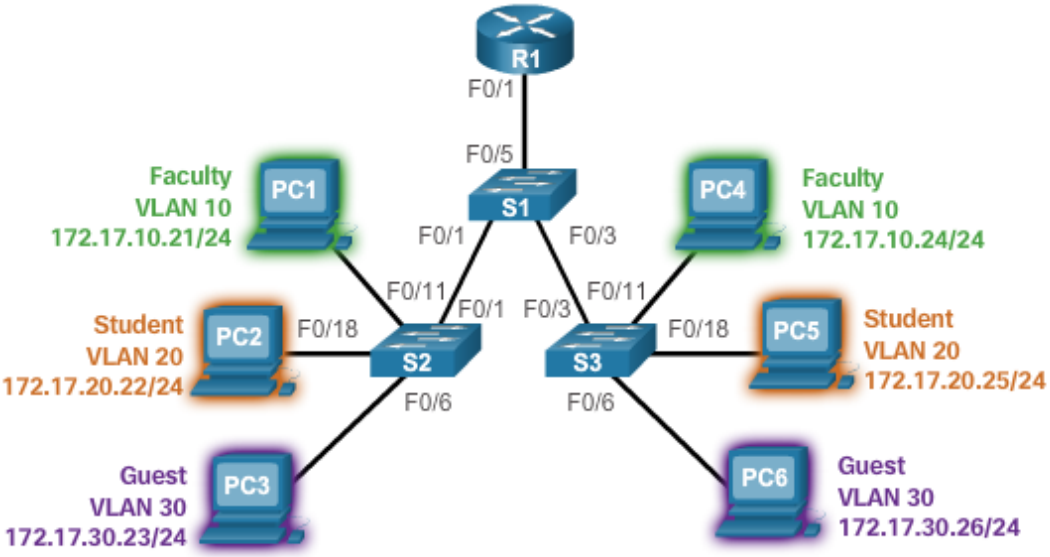
The first part of this chapter will cover how to configure, manage, and troubleshoot VLANs and VLAN trunks. The second part of this chapter focuses on implementing inter-VLAN routing using a router. Inter-VLAN routing on a Layer 3 switch is covered in a later course.

VLAN Definitions



- VLANs allow an administrator to segment networks based on factors such as **function, project team, or application**, without regard for the physical location of the user or device.
- VLANs enable the implementation of access and security policies **according to specific groupings of users.**

- A VLAN is a **logical partition** of a Layer 2 network.
- Multiple partitions can be created, allowing for multiple VLANs to co-exist.
- Each VLAN is a **broadcast domain**, usually with its own IP network.
- VLANs are mutually **isolated**, and packets can only pass between them via a **router**.
- The partitioning of the Layer 2 network takes place inside a Layer 2 device, usually via a switch.
- The hosts grouped within a VLAN are unaware of the VLAN's existence.



- Improved Security
- Reduced Cost
- Better Performance
- Smaller Broadcast Domains
- IT Efficiency
- Management Efficiency
- Simpler Project and Application Management

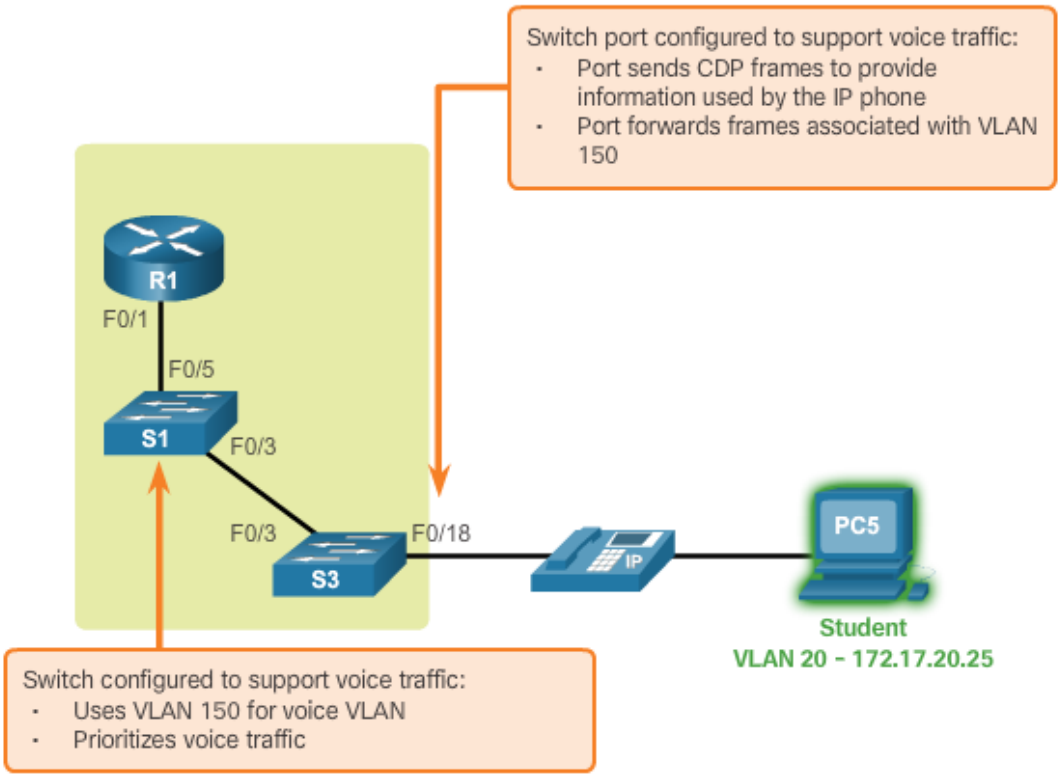
Types of VLANs

- Data VLAN – user generated traffic
- Default VLAN – all switch ports become part of this VLAN until switch is configured, **show vlan brief**
- Native VLAN – used for untagged traffic
- Management VLAN – used to access management capabilities

VLAN 1

```
Switch# show vlan brief

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

- All ports assigned to VLAN 1 by default.
- Native VLAN is VLAN 1 by default.
- Management VLAN is VLAN 1 by default.



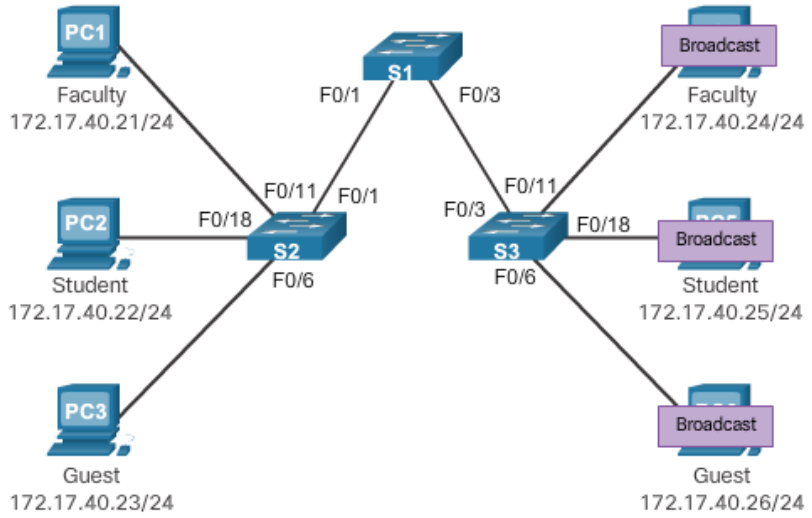
- A VLAN trunk is a point-to-point link that carries more than one VLAN.
- A VLAN trunk is usually established between switches so same-VLAN devices can communicate, even if physically connected to different switches.
- A VLAN trunk is not associated to any VLANs; neither is the trunk ports used to establish the trunk link.
- Cisco IOS supports IEEE802.1q, a popular VLAN trunk protocol.

Controlling Broadcast Domains with VLANs

- VLANs can be used to limit the reach of broadcast frames.
- A VLAN is a broadcast domain of its own.
- A broadcast frame sent by a device in a specific VLAN is forwarded within that VLAN only.
- VLANs help control the reach of broadcast frames and their impact in the network.
- Unicast and multicast frames are forwarded within the originating VLAN.

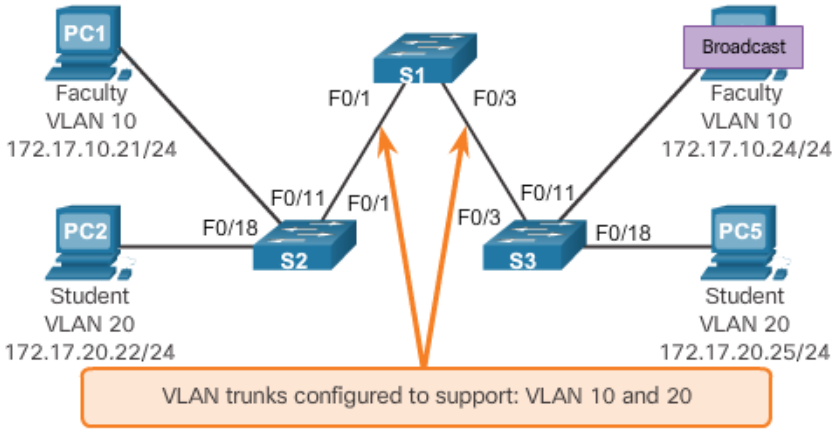
No VLAN Segmentation

PC1 sends out a local Layer 2 broadcast. The switches forward the broadcast frame out all available ports.



With VLAN Segmentation

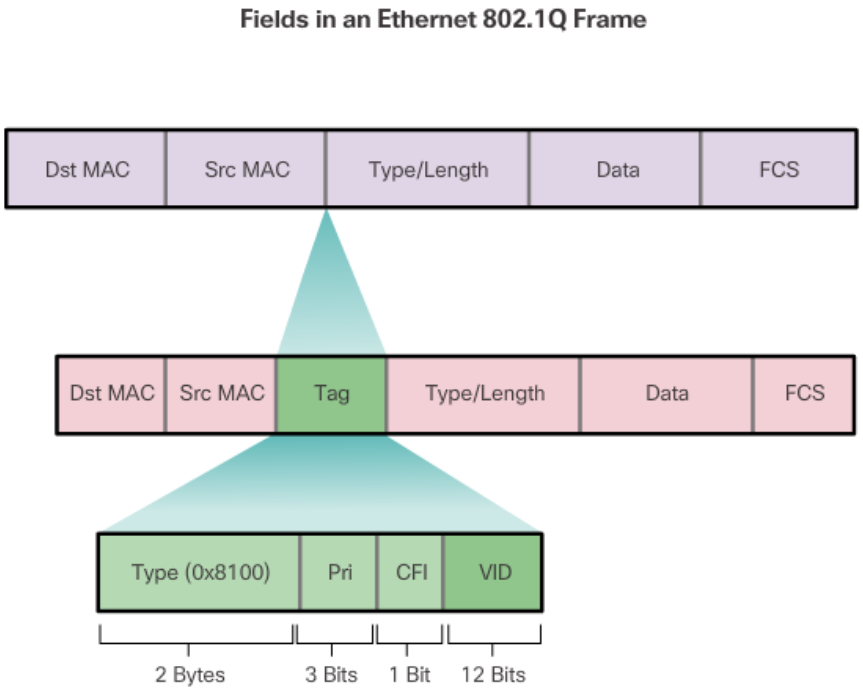
PC1 sends out a local Layer 2 broadcast. The switches forward the broadcast frame only out ports configured for VLAN10.



Tagging Ethernet Frames for VLAN Identification

- Frame tagging is the process of adding a VLAN identification header to the frame.
- It is used to properly transmit multiple VLAN frames through a trunk link.

- Switches tag frames to identify the VLAN to which they belong.
- Different tagging protocols exist; IEEE 802.1Q is a vey popular example.
- The protocol defines the structure of the tagging header added to the frame.
- Switches add VLAN tags to the frames before placing them into trunk links and remove the tags before forwarding frames through non-trunk ports.
- When properly tagged, the frames can transverse any number of switches via trunk links and still be forwarded within the correct VLAN at the destination.

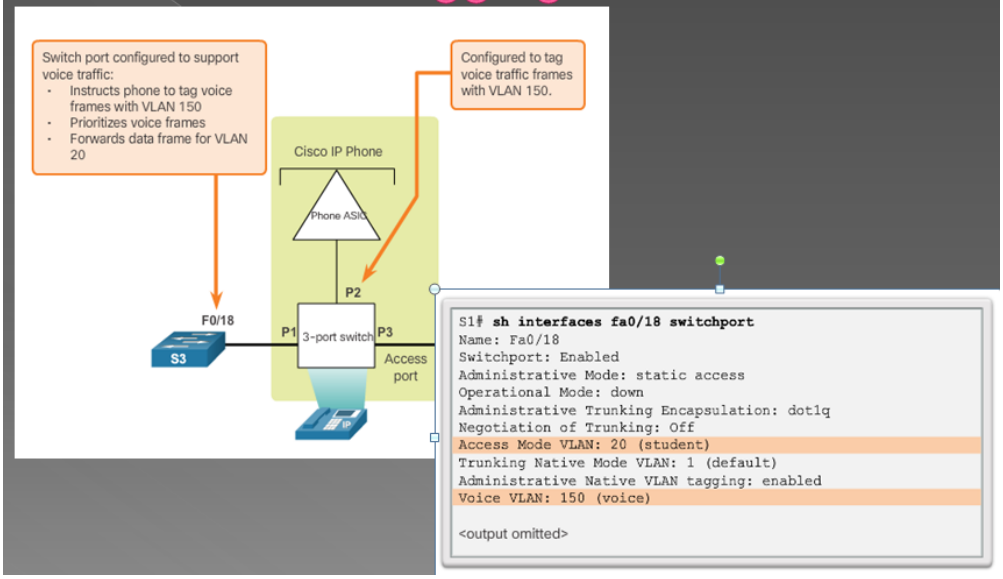


VLANs in a Multi-Switched Environment

Native VLANs and 802.1Q Tagging

- Control traffic sent on the native VLAN should not be tagged.
- Frames received untagged, remain untagged and are placed in the native VLAN when forwarded.
- If there are no ports associated to the native VLAN and no other trunk links, an untagged frame is dropped.
- When configuring a switch port on a Cisco switch, configure devices so that they do not send tagged frames on the native VLAN.
- In Cisco switches, the native VLAN is VLAN 1, by default.

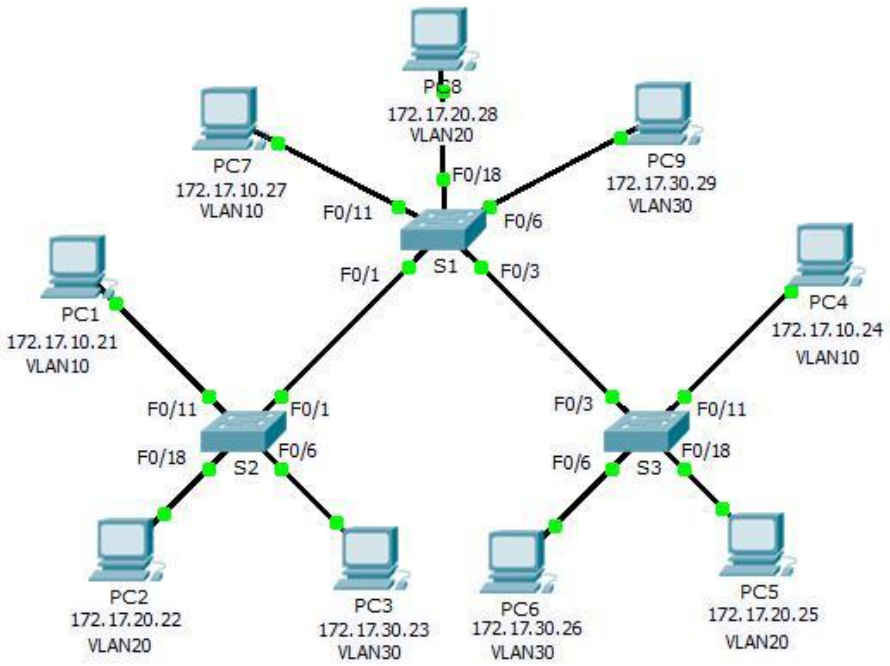
Voice VLAN Tagging




```
S1# sh interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 20 (student)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 150 (voice)

<output omitted>
```

Packet Tracer – Investigating a VLAN Implementation

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
S1	VLAN 99	172.17.99.31	255.255.255.0	N/A
S2	VLAN 99	172.17.99.32	255.255.255.0	N/A
S3	VLAN 99	172.17.99.33	255.255.255.0	N/A
PC1	NIC	172.17.10.21	255.255.255.0	172.17.10.1
PC2	NIC	172.17.20.22	255.255.255.0	172.17.20.1
PC3	NIC	172.17.30.23	255.255.255.0	172.17.30.1
PC4	NIC	172.17.10.24	255.255.255.0	172.17.10.1
PC5	NIC	172.17.20.25	255.255.255.0	172.17.20.1

PC6	NIC	172.17.30.26	255.255.255.0	172.17.30.1
PC7	NIC	172.17.10.27	255.255.255.0	172.17.10.1
PC8	NIC	172.17.20.28	255.255.255.0	172.17.20.1
PC9	NIC	172.17.30.29	255.255.255.0	172.17.30.1

Objectives

Part 1: Observe Broadcast Traffic in a VLAN Implementation

Part 2: Observe Broadcast Traffic without VLANs

Part 3: Complete Reflection Questions

Background

In this activity, you will observe how broadcast traffic is forwarded by the switches when VLANs are configured and when VLANs are not configured.

Part 1: Observe Broadcast Traffic in a VLAN Implementation

Step 1: Ping from PC1 to PC6.

g. Wait for all the link lights to turn to green. To accelerate this process, click **Fast Forward Time** located in the bottom yellow tool bar.

h. Click the **Simulation** tab and use the **Add Simple PDU** tool. Click on **PC1**, and then click on **PC6**.

i. Click the **Capture/Forward** button to step through the process. Observe the ARP requests as they traverse the network. When the Buffer Full window appears, click the **View Previous Events** button.

j. Were the pings successful? Why?

k. Look at the Simulation Panel, where did **S3** send the packet after receiving it?

In normal operation, when a switch receives a broadcast frame on one of its ports, it forwards the frame out all other ports. Notice that **S2** only sends the ARP request out Fa0/1 to **S1**. Also notice that **S3** only sends the ARP request out F0/11 to **PC4**. **PC1** and **PC4** both belong to VLAN 10. **PC6** belongs to VLAN 30. Because broadcast traffic is contained within the VLAN, **PC6** never receives the ARP request from **PC1**. Because **PC4** is not the destination, it discards the ARP request. The ping from **PC1** fails because **PC1** never receives an ARP reply.

Step 2: Ping from PC1 to PC4.

a. Click the **New** button under the Scenario 0 dropdown tab. Now click on the **Add Simple PDU** icon on the right side of Packet Tracer and ping from **PC1** to **PC4**.

b. Click the **Capture/Forward** button to step through the process. Observe the ARP requests as they traverse the network. When the Buffer Full window appears, click the **View Previous Events** button.

c. Were the pings successful? Why?

d. Examine the Simulation Panel. When the packet reached **S1**, why does it also forward the packet to **PC7**?

Part 2: Observe Broadcast Traffic without VLANs

Step 1: Clear the configurations on all three switches and delete the VLAN database.

a. Return to **Realtime** mode.

b. Delete the startup configuration on all 3 switches. What command is used to delete the startup configuration of the switches?

c. Where is the VLAN file stored in the switches?

d. Delete the VLAN file on all 3 switches. What command deletes the VLAN file stored in the switches?

Step 2: Reload the switches.

Use the r**eload** command in privileged EXEC mode to reset all the switches. Wait for the entire link to turn green. To accelerate this process, click **Fast Forward Time** located in the bottom yellow tool bar.

Step 3: Click Capture/Forward to send ARP requests and pings.

d. After the switches reload and the link lights return to green, the network is ready to forward your ARP and ping traffic.

e. Select **Scenario 0** from the drop down tab to return to Scenario 0.

f. From **Simulation** mode, click the **Capture/Forward** button to step through the process. Notice that the switches now forward the ARP requests out all ports, except the port on which the ARP request was received. This default action of switches is why VLANs can improve network performance. Broadcast traffic is contained within each VLAN. When the **Buffer Full** window appears, click the **View Previous Events** button.

Part 3: Complete Reflection Questions

b. If a PC in VLAN 10 sends a broadcast message, which devices receive it?

c. If a PC in VLAN 20 sends a broadcast message, which devices receive it?

d. If a PC in VLAN 30 sends a broadcast message, which devices receive it?

e. What happens to a frame sent from a PC in VLAN 10 to a PC in VLAN 30?

f. In terms of ports, what are the collision domains on the switch?

g. In terms of ports, what are the broadcast domains on the switch?

VLAN Ranges on Catalyst Switches

- Cisco Catalyst 2960 and 3560 Series switches support over 4,000 VLANs.
- VLANs are split into two categories:
  - Normal range VLANs
    - VLAN numbers from 1 to 1,005
    - Configurations stored in the vlan.dat (in the flash memory)
    - IDs 1002 through 1005 are reserved for Token Ring and Fiber Distributed Data Interface (FDDI) VLANs, automatically created and cannot be removed
  - Extended Range VLANs
    - VLAN numbers from 1,006 to 4,096
    - Configurations stored in the running configuration (NVRAM)
    - VLAN Trunking Protocol (VTP) does not learn extended VLANs

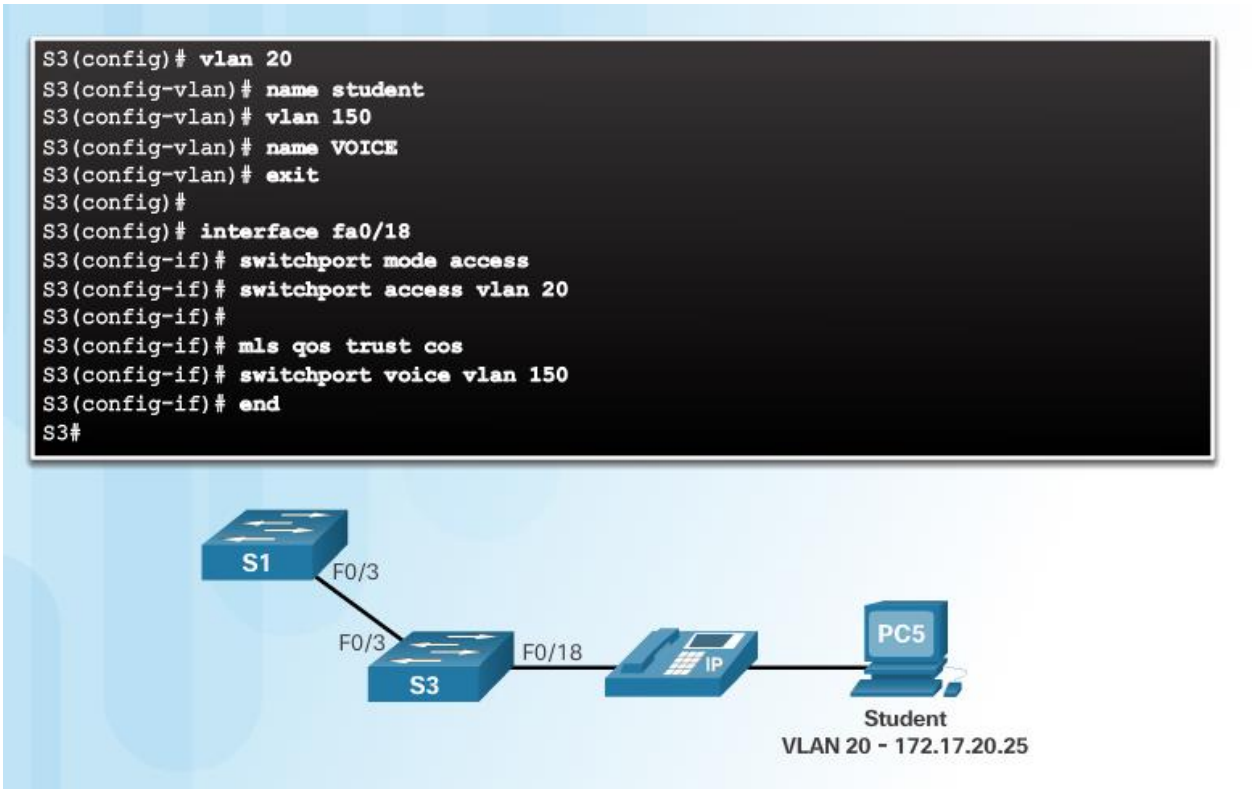
Create VLAN

Create a VLAN

Cisco Switch IOS Commands	
Enter global configuration mode.	`S1# configure terminal`
Create a VLAN with a valid id number.	`S1(config)# vlan vlan-id`
Specify a unique name to identify the VLAN.	`S1(config-vlan)# name vlan-name`
Return to the privileged EXEC mode.	`S1(config-vlan)# end`

Assign Ports to VLANs

Cisco Switch IOS Commands	
Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface interface_id`
Set the port to access mode.	`S1(config-if)# switchport mode access`
Assign the port to a VLAN.	`S1(config-if)# switchport access vlan vlan_id`
Return to the privileged EXEC mode.	`S1(config-if)# end`



Deleting VLANs

In the figure, the **no vlan** *vlan-id* global configuration mode command is used to remove VLAN 20 from the switch. Switch S1 had a minimal configuration with all ports in VLAN 1 and an unused VLAN 20 in the VLAN database. The **show vlan brief** command verifies that VLAN 20 is no longer present in the vlan.dat file after using the **no vlan 20** command.

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.

Alternatively, 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, the previously configured VLANs are no longer present. This effectively places the switch into its factory default condition with regard to VLAN configurations.

Note: For a Catalyst switch, the **erase startup-config** command must accompany the **delete vlan.dat** command prior to reload to restore the switch to its factory default condition.

```
S1# conf t
S1(config)# no vlan 20
S1(config)# end
S1#
S1# sh vlan brief

VLAN Name                 Status    Ports
---- ---------------      --------- -------------------------------
1    default              active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                    Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                    Fa0/9, Fa0/10, Fa0/12, Fa0/13
                                    Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                    Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                    Fa0/22, Fa0/23, Fa0/24, Gi0/1
                                    Gi0/2
1002 fddi-default         act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default        act/unsup
S1#
```

`show vlan` Command

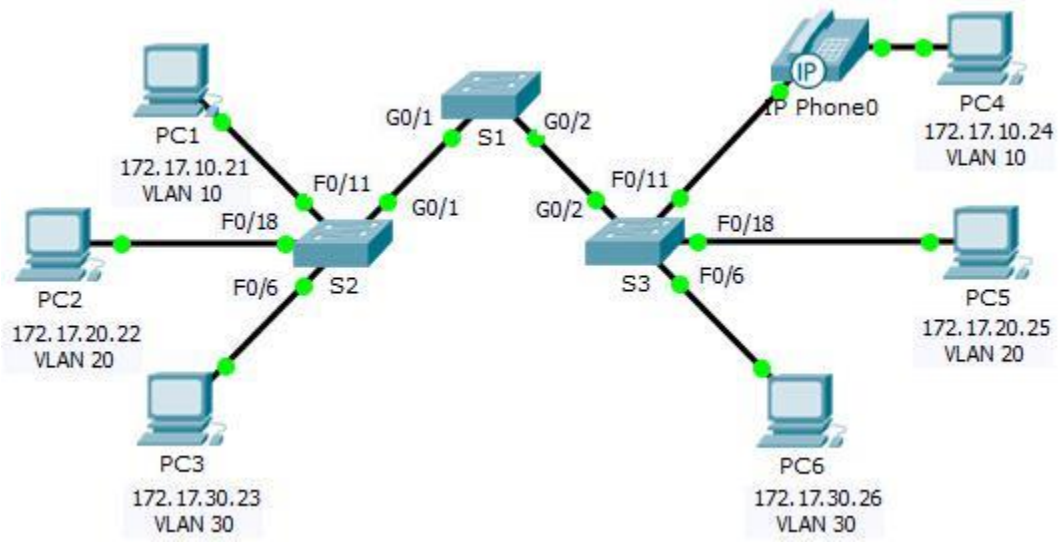
Cisco IOS CLI Command Syntax	
show vlan [**brief** | **id** *vlan-id* | **name** *vlan-name* | **summary**]	
Display one line for each VLAN with the VLAN name, status, and its ports.	**brief**
Display information about a single VLAN identified by VLAN ID number. For vlan-id, the range is 1 to 4094.	**id** *vlan-id*
Display information about a single VLAN identified by VLAN name. The VLAN name is an ASCII string from 1 to 32 characters.	**name** *vlan-name*
Display VLAN summary information.	**summary**

`show interfaces` Command

Cisco IOS CLI Command Syntax	
show interfaces [*interface-id* | **vlan** *vlan-id*] | **switchport**	
Valid interfaces include physical ports (including type, module, and port number) and port channels. The port-channel range is 1 to 6.	*interface-id*
VLAN identification. The range is 1 to 4094.	**vlan** *vlan-id*
Display the administrative and operational status of a switching port, including port blocking and port protection settings.	**switchport**

Packet Tracer – Configuring VLANs

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	VLAN
PC1	NIC	172.17.10.21	255.255.255.0	10
PC2	NIC	172.17.20.22	255.255.255.0	20
PC3	NIC	172.17.30.23	255.255.255.0	30
PC4	NIC	172.17.10.24	255.255.255.0	10
PC5	NIC	172.17.20.25	255.255.255.0	20
PC6	NIC	172.17.30.26	255.255.255.0	30

Objectives

Part 1: Verify the Default VLAN Configuration

Part 2: Configure VLANs

Part 3: Assign VLANs to Ports

Background

VLANs are helpful in the administration of logical groups, allowing members of a group to be easily moved, changed, or added. This activity focuses on creating and naming VLANs, and assigning access ports to specific VLANs.

Part 1: View the Default VLAN Configuration

Step 1: Display the current VLANs.

On S1, issue the command that displays all VLANs configured. By default, all interfaces are assigned to VLAN 1.

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

Notice that each PC can ping the other PC that shares the same network.

l. PC1 can ping PC4

m. PC2 can ping PC5

n. PC3 can ping PC6

Pings to PCs in other networks fail.

What benefit will configuring VLANs provide to the current configuration?

Part 2: Configure VLANs

Step 1: Create and name VLANs on S1.

Create the following VLANs. Names are case-sensitive:

e. VLAN 10: Faculty/Staff

f. VLAN 20: Students

g. VLAN 30: Guest(Default)

h. VLAN 99: Management&Native

i. VLAN 150: VOICE

Step 2: Verify the VLAN configuration.

Which command will only display the VLAN name, status, and associated ports on a switch?

Step 3: Create the VLANs on S2 and S3.

Using the same commands from Step 1, create and name the same VLANs on S2 and S3.

Step 4: Verify the VLAN configuration.

Part 3: Assign VLANs to Ports

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

Configure the interfaces as access ports and assign the VLANs as follows:

e. VLAN 10: FastEthernet 0/11

g. VLAN 20: FastEthernet 0/18

h. VLAN 30: FastEthernet 0/6

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

S3 uses the same VLAN access port assignments as S2. Configure the interfaces as access ports and assign the VLANs as follows:

h. VLAN 10: FastEthernet 0/11

i. VLAN 20: FastEthernet 0/18

j. VLAN 30: FastEthernet 0/6

Step 3: Assign the VOICE VLAN to FastEthernet 0/11 on S3.

As shown in the topology, the S3 FastEthernet 0/11 interface connects to a Cisco IP Phone and PC4. The IP phone contains an integrated three-port 10/100 switch. One port on the phone is labeled Switch and connects to F0/4. Another port on the phone is labeled PC and connects to PC4. The IP phone also has an internal port that connects to the IP phone functions.

The S3 F0/11 interface must be configured to support user traffic to PC4 using VLAN 10 and voice traffic to the IP phone using VLAN 150. The interface must also enable QoS and trust the Class of Service (CoS) values assigned by the IP phone.

Step 4: Verify loss of connectivity.

Previously, PCs that shared the same network could ping each other successfully.

Try pinging between PC1 and PC4. Although the access ports are assigned to the appropriate VLANs, were the pings successful? Why?

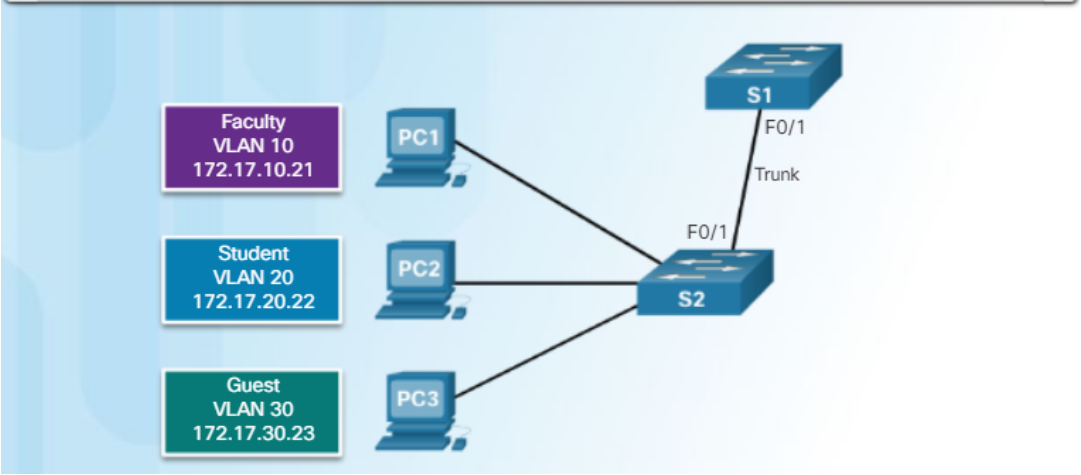
What could be done to resolve this issue?

Configuring IEEE 802.1Q Trunk Links

A VLAN trunk is an OSI Layer 2 link between two switches that carries traffic for all VLANs (unless the allowed VLAN list is restricted manually or dynamically). To enable trunk links, configure the ports on either end of the physical link with parallel sets of commands

Cisco Switch IOS Commands	
Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface interface_id`
Force the link to be a trunk link.	`S1(config-if)# switchport mode trunk`
Specify a native VLAN for untagged frames.	`S1(config-if)# switchport trunk native vlan vlan_id`
Specify the list of VLANs to be allowed on the trunk link.	`S1(config-if)# switchport trunk allowed vlan vlan-list`
Return to the privileged EXEC mode.	`S1(config-if)# end`

```
VLAN 10 - Faculty/Staff - 172.17.10.0/24
VLAN 20 - Students - 172.17.20.0/24
VLAN 30 - Guest - 172.17.30.0/24
VLAN 99 - Native - 172.17.99.0/24
```




```
S1(config)# interface FastEthernet0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# switchport trunk allowed vlan 10,20,30,99
S1(config-if)# end
```

Resetting the Trunk to Default State

Cisco Switch IOS Commands	
Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface interface_id`
Set trunk to allow all VLANs.	`S1(config-if)# no switchport trunk allowed vlan`
Reset native VLAN to default.	`S1(config-if)# no switchport trunk native vlan`
Return to the privileged EXEC mode.	`S1(config-if)# end`

```
S1(config)# interface f0/1
S1(config-if)# no switchport trunk allowed vlan
S1(config-if)# no switchport trunk native vlan
S1(config-if)# end
S1# show interfaces f0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled

<output omitted>
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001

<output omitted>
```

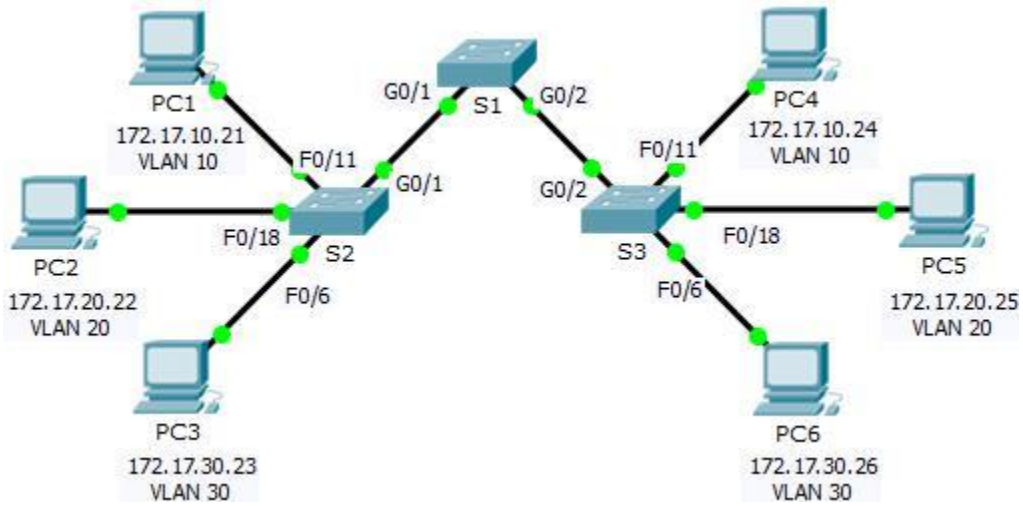
```
S1(config)# interface f0/1
S1(config-if)# switchport mode access
S1(config-if)# end
S1# show interfaces f0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled

<output omitted>
```

Verifying Trunk Configuration

```
S1(config)# interface f0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# end
S1# show interfaces f0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (VLAN0099)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
<output omitted>
```

Packet Tracer – Configuring Trunks

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Switch Port	VLAN
PC1	NIC	172.17.10.21	255.255.255.0	S2 F0/11	10
PC2	NIC	172.17.20.22	255.255.255.0	S2 F0/18	20
PC3	NIC	172.17.30.23	255.255.255.0	S2 F0/6	30
PC4	NIC	172.17.10.24	255.255.255.0	S3 F0/11	10
PC5	NIC	172.17.20.25	255.255.255.0	S3 F0/18	20
PC6	NIC	172.17.30.26	255.255.255.0	S3 F0/6	30

Objectives

Part 1: Verify VLANs

Part 2: Configure Trunks

Background

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

Part 1: Verify VLANs

Step 1: Display the current VLANs.

h. On **S1**, issue the command that will display all VLANs configured. There should be ten VLANs in total. Notice how all 24 access ports on the switch are assigned to VLAN 1.

i. On **S2** and **S3**, display and verify all the VLANs are configured and assigned to the correct switch ports according to the **Addressing Table**.

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

Although **PC1** and **PC4** are on the same network, they cannot ping one another. This is because the ports connecting the switches are assigned to VLAN 1 by default. In order to provide connectivity between the PCs on the same network and VLAN, trunks must be configured.

Part 2: Configure Trunks

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

o. Configure G0/1 and G0/2 interfaces on S1 for trunking.

p. Configure VLAN 99 as the native VLAN for G0/1 and G0/2 interfaces on **S1**.

The trunk port takes about a minute to become active due to Spanning Tree. Click **Fast Forward Time** to speed the process. After the ports become active, you will periodically receive the following syslog messages:

```
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on GigabitEthernet0/2
(99), with S3 GigabitEthernet0/2 (1).
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on GigabitEthernet0/1
(99), with S2 GigabitEthernet0/1 (1).
```

You configured VLAN 99 as the native VLAN on S1. However, S2 and S3 are using VLAN 1 as the default native VLAN as indicated by the syslog message.

Although you have a native VLAN mismatch, pings between PCs on the same VLAN are now successful. Why?

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

On **S2** and **S3**, issue the **show interface trunk** command to confirm that DTP has successfully negotiated trunking with S1 on S2 and S3. The output also displays information about the trunk interfaces on S2 and S3.

Which active VLANs are allowed to cross the trunk?

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

j. Configure VLAN 99 as the native VLAN for the appropriate interfaces on S2 and S3.

k. Issue **show interface trunk** command to verify the correct native VLAN configuration.

Step 4: Verify configurations on S2 and S3.

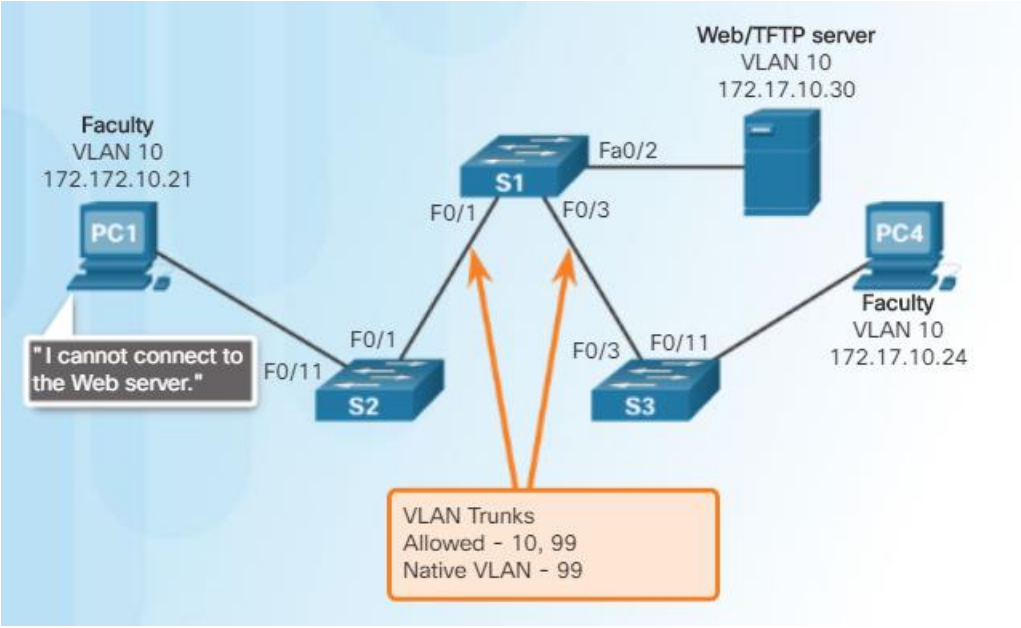
f. Issue the **show interface** *interface* **switchport** command to verify that the native VLAN is now 99.

g. Use the **show vlan** command to display information regarding configured VLANs. Why is port G0/1 on S2 no longer assigned to VLAN 1?

IP Addressing Issues with VLAN

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

In Figure 1, PC1 cannot connect to the Web/TFTP server shown.



Troubleshoot VLANs and Trunks

Common Problems with Trunks

Trunking issues are usually associated with incorrect configurations.

The most common type of trunk configuration errors are:

- Native VLAN mismatches
- Trunk mode mismatches
- Allowed VLANs on trunks

If a trunk problem is detected, the best practice guidelines recommend to troubleshoot in the order shown above.

Problem	Result	Example
Native VLAN Mismatches	Poses a security risk and creates unintended results.	For example, one port is defined as VLAN 99 and the other is defined as VLAN 100.
Trunk Mode Mismatches	Causes loss of network connectivity.	For example, both local and peer switchport modes are configured as dynamic auto.
Allowed VLANs on Trunks	Causes unexpected traffic or no traffic to be sent over the trunk.	The list of allowed VLANs does not support current VLAN trunking requirements.

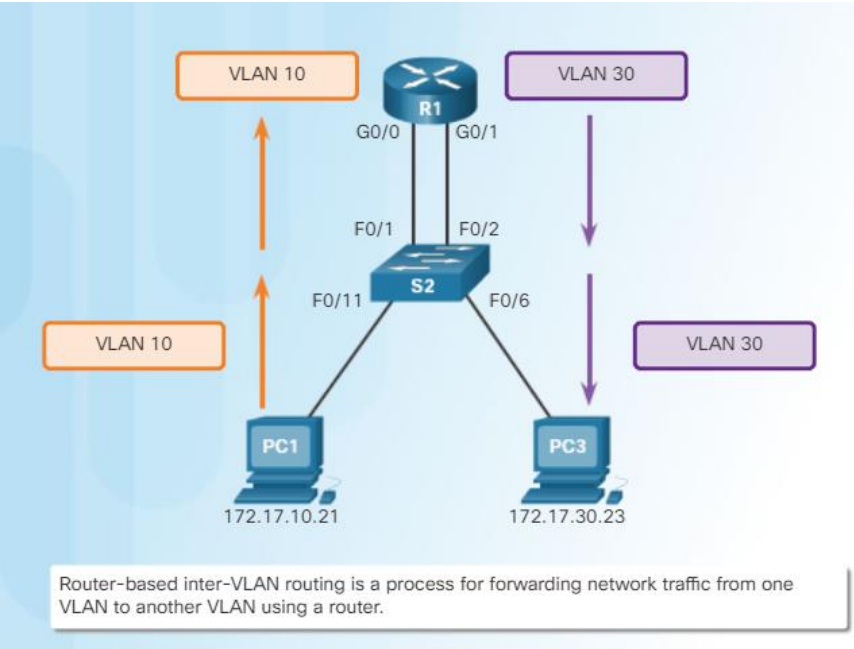
What is Inter-VLAN Routing?

VLANs are used to segment switched networks. Layer 2 switches, such as the Catalyst 2960 Series, can be configured with over 4,000 VLANs. A VLAN is a broadcast domain, so computers on separate VLANs are unable to communicate without the intervention of a routing device. Layer 2 switches have very limited IPv4 and IPv6 functionality and cannot perform the dynamic routing function of routers. While Layer 2 switches are gaining more IP functionality, such as the ability to perform static routing, this is insufficient to handle these large number of VLANs.

Any device that supports Layer 3 routing, such as a router or a multilayer switch, can be used to perform the necessary routing functionality. Regardless of the device used, the process of forwarding network traffic from one VLAN to another VLAN using routing is known as inter-VLAN routing.

There are three options for inter-VLAN routing:

- Legacy inter-VLAN routing
- Router-on-a-Stick
- Layer 3 switching using SVIs



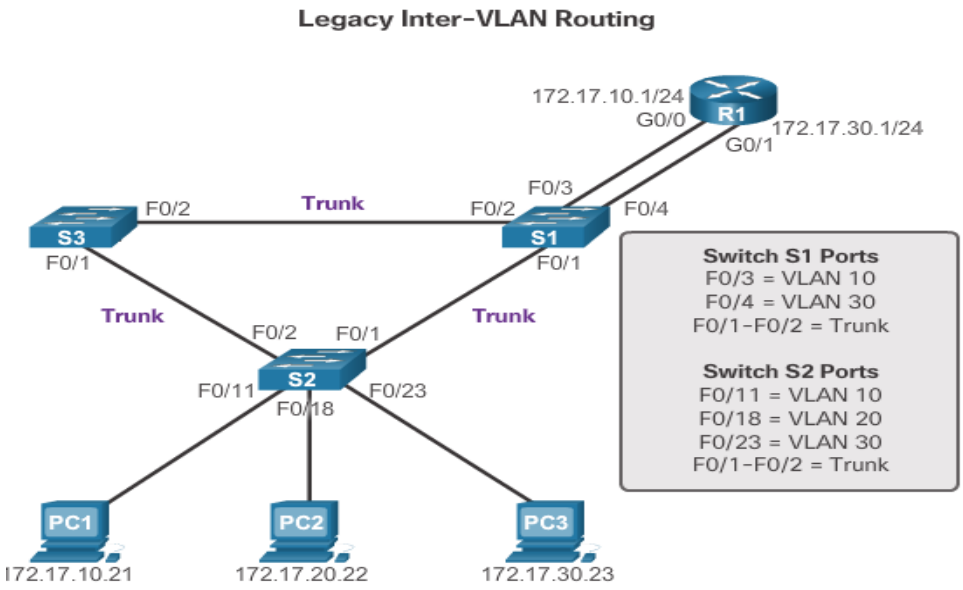
Router-based inter-VLAN routing is a process for forwarding network traffic from one VLAN to another VLAN using a router.

Legacy Inter-VLAN Routing

In the past:

- Actual routers were used to route between VLANs.
- Each VLAN was connected to a different physical router interface.
- Packets would arrive on the router through one interface, be routed and leave through another.
- Because the router interfaces were connected to VLANs and had IP addresses from that specific VLAN, routing between VLANs was achieved.
- Large networks with large number of VLANs required many router interfaces.

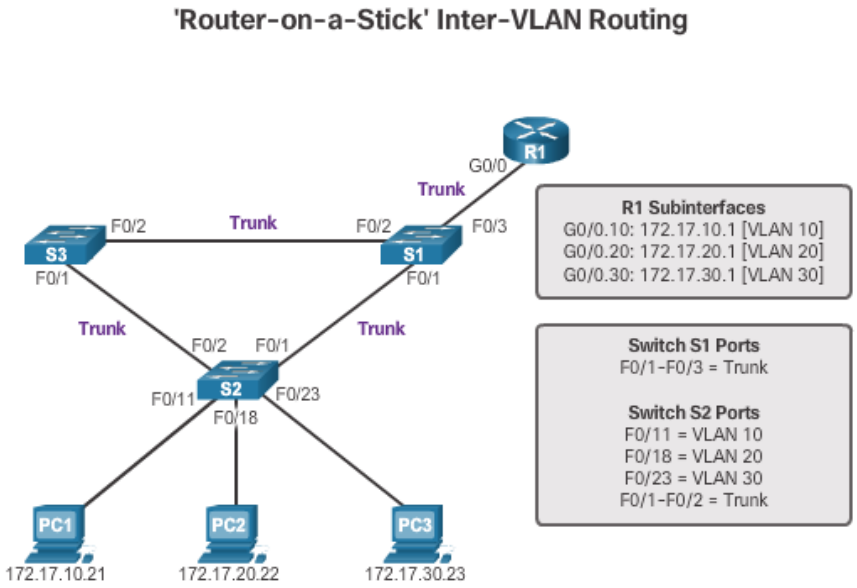
In this example, the router was configured with two separate physical interfaces to interact with the different VLANs and perform the routing.



Router-on-a-Stick Inter-VLAN Routing

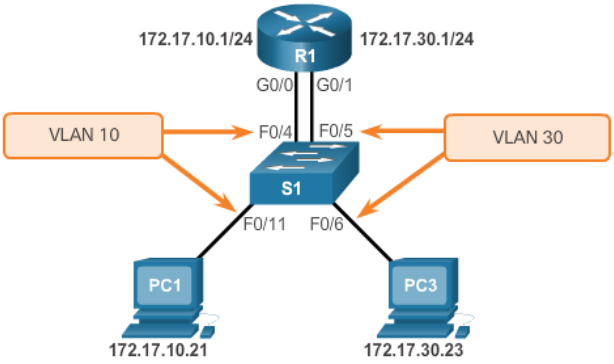
- The router-on-a-stick approach uses only one of the router's physical interface.
- One of the router's physical interfaces is configured as a 802.1Q trunk port so it can understand VLAN tags.
- Logical subinterfaces are created; one subinterface per VLAN.
- Each subinterface is configured with an IP address from the VLAN it represents.
- VLAN members (hosts) are configured to use the subinterface address as a default gateway.

Router interface configured to operate as a trunk link and is connected to a trunked switch port. The router performs inter-VLAN routing by accepting VLAN-tagged traffic on the trunk interface coming from the adjacent switch, and then, internally routing between the VLANs using subinterfaces. The router then forwards the routed traffic, VLAN-tagged for the destination VLAN, out the same physical interface as it used to receive the traffic.



Configure Legacy Inter-VLAN Routing: Switch Configuration

- Legacy inter-VLAN routing requires routers to have multiple physical interfaces.
- Each one of the router's physical interfaces is connected to a unique VLAN.
- Each interface is also configured with an IP address for the subnet associated with the particular VLAN.
- Network devices use the router as a gateway to access the devices connected to the other VLANs.

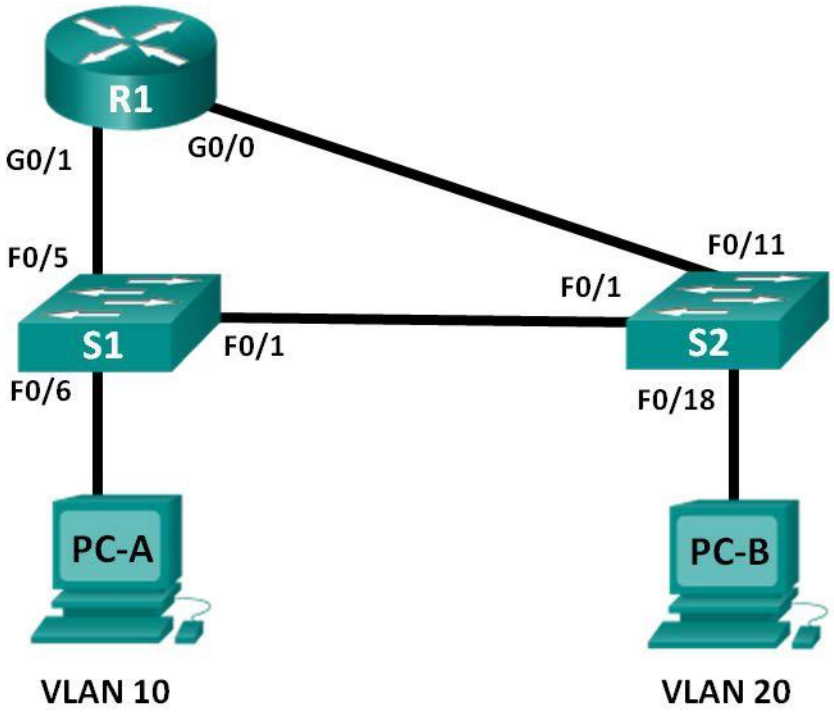



```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/11
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/4
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/6
S1(config-if)# switchport access vlan 30
S1(config-if)# interface f0/5
S1(config-if)# switchport access vlan 30
S1(config-if)# end
```

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

Lab – Configuring Per-Interface Inter-VLAN Routing

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.20.1	255.255.255.0	N/A
	G0/1	192.168.10.1	255.255.255.0	N/A

S1	VLAN 10	192.168.10.11	255.255.255.0	192.168.10.1
S2	VLAN 10	192.168.10.12	255.255.255.0	192.168.10.1
PC-A	NIC	192.168.10.3	255.255.255.0	192.168.10.1
PC-B	NIC	192.168.20.3	255.255.255.0	192.168.20.1

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure Switches with VLANs and Trunking

Part 3: Verify Trunking, VLANs, Routing, and Connectivity

Background / Scenario

Legacy inter-VLAN routing is seldom used in today's networks; however, it is helpful to configure and understand this type of routing before moving on to router-on-a-stick (trunk-based) inter-VLAN routing or

configuring Layer-3 switching. Also, you may encounter per-interface inter-VLAN routing in organizations with very small networks. One of the benefits of legacy inter-VLAN routing is ease of configuration.

In this lab, you will set up one router with two switches attached via the router Gigabit Ethernet interfaces. Two separate VLANs will be configured on the switches, and you will set up routing between the VLANs.

Note: This lab provides minimal assistance with the actual commands necessary to configure the router and switches. The required switch VLAN configuration commands are provided in Appendix A of this lab. Test your knowledge by trying to configure the devices without referring to the appendix.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS, Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS, Release 15.0(2) (lanbasek9 image) . Other routers, switches and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Note: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

Required Resources

i. 1 Router (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)

j. 2 Switches (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)

k. 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)

l. Console cables to configure the Cisco IOS devices via the console ports

m. Ethernet cables as shown in the topology

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and clear any configurations, if necessary.

Step 1: Cable the network as shown in the topology.

Step 2: Initialize and reload the router and switches.

Step 3: Configure basic settings for R1.

k. Console into R1 and enter global configuration mode.

l. Copy the following basic configuration and paste it to the running-configuration on R1.

```
no ip domain-
lookup hostname R1
service password-
encryption enable secret
class banner motd #
Unauthorized access is strictly prohibited.
# line con 0
password
cisco login
logging synchronous
line vty 0 4
password cisco
```

```
login
```

c. Configure addressing on G0/0 and G0/1 and enable both interfaces.

d. Copy the running configuration to the startup configuration.

Step 4: Configure basic settings on both switches.

c. Console into the switch and enter global configuration mode.

d. Copy the following basic configuration and paste it to running-configuration on the switch.

```
no ip domain-lookup
service password-
encryption enable secret
class banner motd #
Unauthorized access is strictly prohibited.
# Line con 0
password
cisco login
logging
synchronous line
vty 0 15 password
cisco login
exit
```

e. Configure the host name as shown in the topology.

f. Copy the running configuration to the startup configuration.

Step 5: Configure basic settings on PC-A and PC-B.

Configure PC-A and PC-B with IP addresses and a default gateway address according to the Addressing Table.

Part 2: Configure Switches with VLANs and Trunking

In Part 2, you will configure the switches with VLANs and trunking.

Step 1: Configure VLANs on S1.

- On S1, create VLAN 10. Assign **Student** as the VLAN name.
- Create VLAN 20. Assign **Faculty-Admin** as the VLAN name.
- Configure F0/1 as a trunk port.
- Assign ports F0/5 and F0/6 to VLAN 10 and configure both F0/5 and F0/6 as access ports.
- Assign an IP address to VLAN 10 and enable it. Refer to the Addressing Table.

• Configure the default gateway according to the Addressing Table.

Step 2: Configure VLANs on S2.

- On S2, create VLAN 10. Assign **Student** as the VLAN name.
- Create VLAN 20. Assign **Faculty-Admin** as the VLAN name.
- Configure F0/1 as a trunk port.

h. Assign ports F0/11 and F0/18 to VLAN 20 and configure both F0/11 and F0/18 as access ports.

i. Assign an IP address to VLAN 10 and enable it. Refer to the Addressing Table.

j. Configure the default gateway according to the Addressing Table.

Part 3: Verify Trunking, VLANs, Routing, and Connectivity

Step 1: Verify the R1 routing table.

h. On R1, issue the **show ip route** command. What routes are listed on R1?

i. On both S1 and S2, issue the **show interface trunk** command. Is the F0/1 port on both switches set to trunk?

j. Issue a **show vlan brief** command on both S1 and S2. Verify that VLANs 10 and 20 are active and that the proper ports on the switches are in the correct VLANs. Why is F0/1 not listed in any of the active VLANs?

k. Ping from PC-A in VLAN 10 to PC-B in VLAN 20. If Inter-VLAN routing is functioning correctly, the pings between the 192.168.10.0 network and the 192.168.20.0 should be successful.

Note: It may be necessary to disable the PC firewall to ping between PCs.

l. Verify connectivity between devices. You should be able to ping between all devices. Troubleshoot if you are not successful.

Reflection

What is an advantage of using legacy inter-VLAN routing?

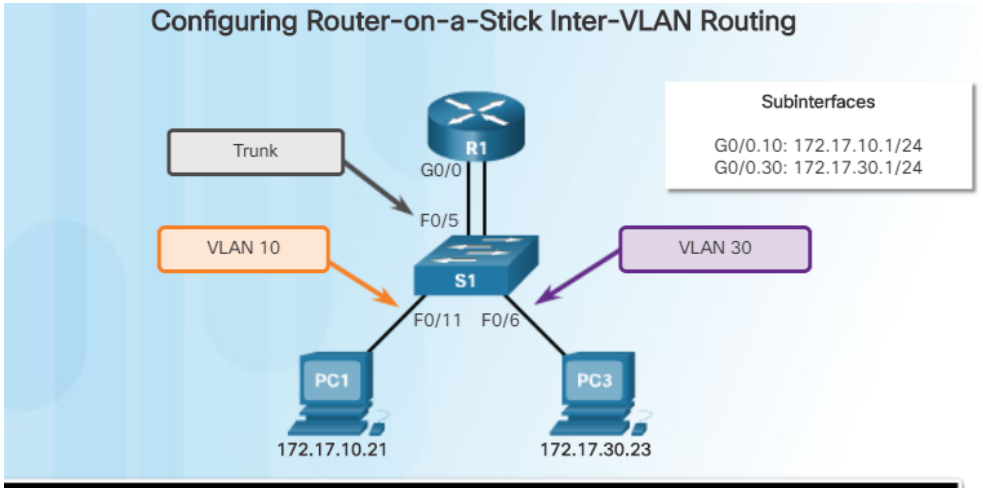
Configure Router-on-a-Stick: Switch Configuration

To enable inter-VLAN routing using router-on-a stick, start by enabling trunking on the switch port that is connected to the router.

In the figure, router R1 is connected to switch S1 on trunk port F0/5. VLANs 10 and 30 are added to switch S1.

Because switch port F0/5 is configured as a trunk port, the port does not need to be assigned to any VLAN. To configure switch port F0/5 as a trunk port, execute the**switchport mode trunk**command in interface configuration mode for port F0/5.

The router can now be configured to perform inter-VLAN routing.

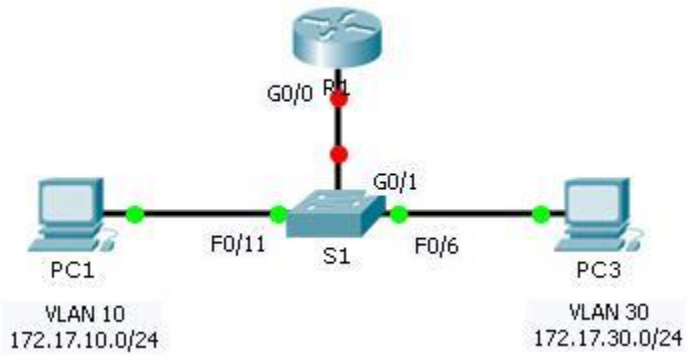



```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/5
S1(config-if)# switchport mode trunk
S1(config-if)# end
S1#
```

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

Packet Tracer – Configuring Router-on-a-Stick Inter-VLAN Routing

Topology



Addressing Table

Device	Interface	IPv4 Address	Subnet Mask	Default Gateway
R1	G0/0.10	172.17.10.1	255.255.255.0	N/A
	G0/0.30	172.17.30.1	255.255.255.0	N/A
PC1	NIC	172.17.10.10	255.255.255.0	172.17.10.1
PC2	NIC	172.17.30.10	255.255.255.0	172.17.30.1

Objectives

Part 1: Test Connectivity without Inter-VLAN Routing

Part 2: Add VLANs to a Switch

Part 3: Configure Subinterfaces

Part 4: Test Connectivity with Inter-VLAN Routing

Scenario

In this activity, you will check for connectivity prior to implementing inter-VLAN routing. You will then configure VLANs and inter-VLAN routing. Finally, you will enable trunking and verify connectivity between VLANs.

Part 1: Test Connectivity Without Inter-VLAN Routing

Step 1: Ping between PC1 and PC3.

Wait for switch convergence or click **Fast Forward Time** a few times. When the link lights are green for **PC1** and **PC3**, ping between **PC1** and **PC3**. Because the two PCs are on separate networks and **R1** is not configured, the ping fails.

Step 2: Switch to Simulation mode to monitor pings.

l. Switch to Simulation mode by clicking the **Simulation** tab or pressing **Shift+S**.

m. Click **Capture/Forward** to see the steps the ping takes between **PC1** and **PC3**. Notice how the ping never leaves **PC1**. What process failed and why?

Part 2: Add VLANs to a Switch

Step 1: Create VLANs on S1.

Return to **Realtime** mode and create VLAN 10 and VLAN 30 on **S1**.

Step 2: Assign VLANs to ports.

h. Configure interface F0/6 and F0/11 as access ports and assign VLANs.

Assign **PC1** to VLAN 10.

Assign **PC3** to VLAN 30.

i. Issue the **show vlan brief** command to verify VLAN configuration.

```
S1# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24, Gig0/1, Gig0/2
10   VLAN0010                         active    Fa0/11
30   VLAN0030                         active    Fa0/6
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

Step 3: Test connectivity between PC1 and PC3.

From **PC1**, ping **PC3**. The pings should still fail. Why were the pings unsuccessful?

Part 3: Configure Subinterfaces

Step 1: Configure subinterfaces on R1 using the 802.1Q encapsulation.

a. Create the subinterface G0/0.10.

Set the encapsulation type to 802.1Q and assign VLAN 10 to the subinterface.

Refer to the **Address Table** and assign the correct IP address to the subinterface.

b. Repeat for the G0/0.30 subinterface.

Step 2: Verify Configuration.

m. Use the **show ip interface brief** command to verify subinterface configuration. Both subinterfaces are down. Subinterfaces are virtual interfaces that are associated with a physical interface. Therefore, in order to enable subinterfaces, you must enable the physical interface that they are associated with.

n. Enable the G0/0 interface. Verify that the subinterfaces are now active.

Part 4: Test Connectivity with Inter-VLAN Routing

Step 1: Ping between PC1 and PC3.

From **PC1**, ping **PC3**. The pings should still fail.

Step 2: Enable trunking.

e. On **S1**, issue the **show vlan** command. What VLAN is G0/1 assigned to?

f. Because the router was configured with multiple subinterfaces assigned to different VLANs, the switch port connecting to the router must be configured as a trunk. Enable trunking on interface G0/1.

g. How can you determine that the interface is a trunk port using the **show vlan** command?

h. Issue the **show interface trunk** command to verify the interface is configured as a trunk.

Step 3: Switch to Simulation mode to monitor pings.

g. Switch to **Simulation** mode by clicking the **Simulation** tab or pressing **Shift+S**.

h. Click **Capture/Forward** to see the steps the ping takes between **PC1** and **PC3**.

i. You should see ARP requests and replies between **S1** and **R1**. Then ARP requests and replies between **R1** and **S3**. Then **PC1** can encapsulate an ICMP echo request with the proper data-link layer information and R1 will route the request to **PC3**.

Note: After the ARP process finishes, you may need to click Reset Simulation to see the ICMP process complete.

Lab 7: STP

Redundancy at OSI Layers 1 and 2

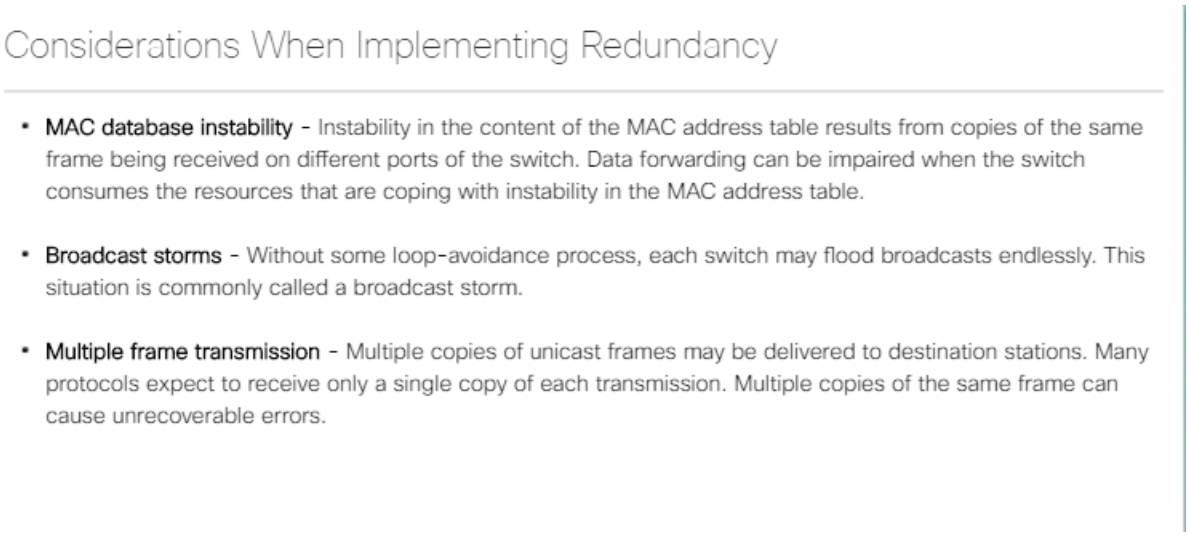
The three-tier hierarchical network design that uses core, distribution, and access layers with redundancy, attempts to eliminate a single point of failure on the network. Multiple cabled paths between switches provide physical redundancy in a switched network. This improves the reliability and availability of the network. Having alternate physical paths for data to traverse the network makes it possible for users to access network resources, despite path disruption.

For many organizations, the availability of the network is essential to supporting business needs; therefore, the network infrastructure design is a critical business element. Path redundancy provides the necessary availability of multiple network services by eliminating the possibility of a single point of failure.

Note: The OSI Layer 1 redundancy is illustrated using multiple links and devices, but more than just physical planning is required to complete the network setup. For the redundancy to work in a systematic way, the use of OSI Layer 2 protocols, such as STP, is also required.

Redundancy is an important part of the hierarchical design for 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. However, redundant paths in a switched Ethernet network may cause both physical and logical Layer 2 loops.

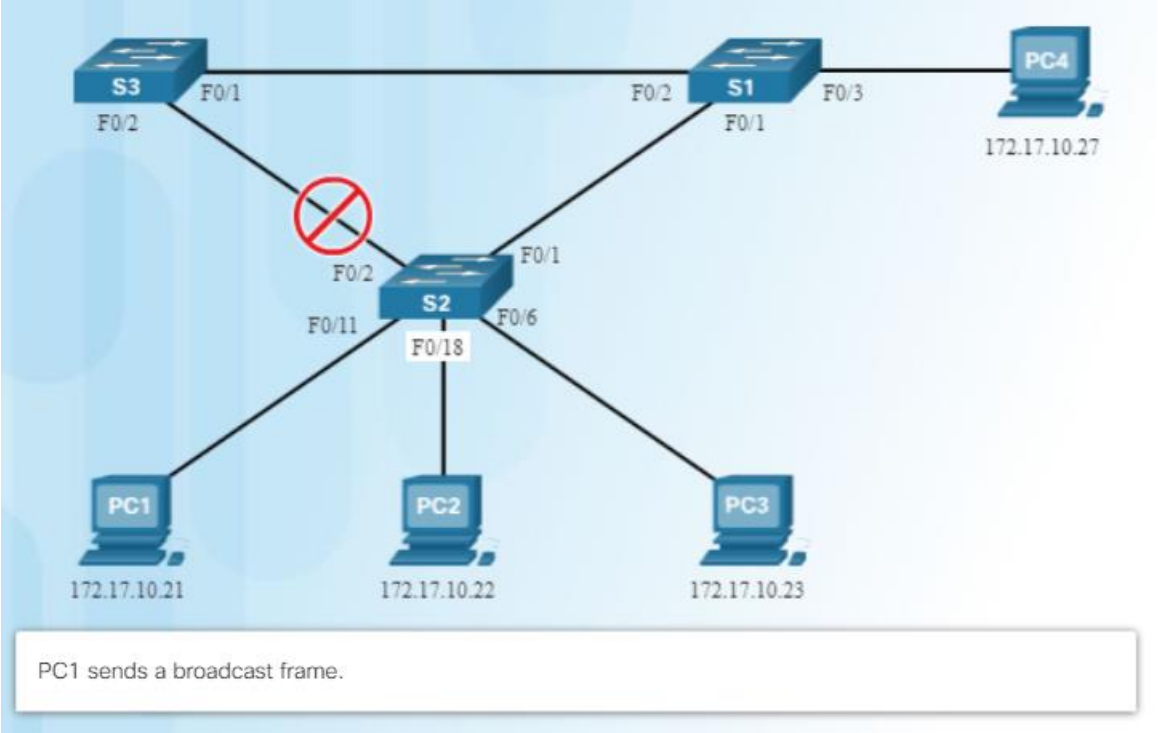
Logical Layer 2 loops may occur due to the natural operation of switches, specifically, the learning and forwarding process. When multiple paths exist between two devices on a network, and there is no spanning tree implementation on the switches, a Layer 2 loop occurs. A Layer 2 loop can result in the three primary issues listed in Figure.



Spanning Tree Algorithm: Introduction

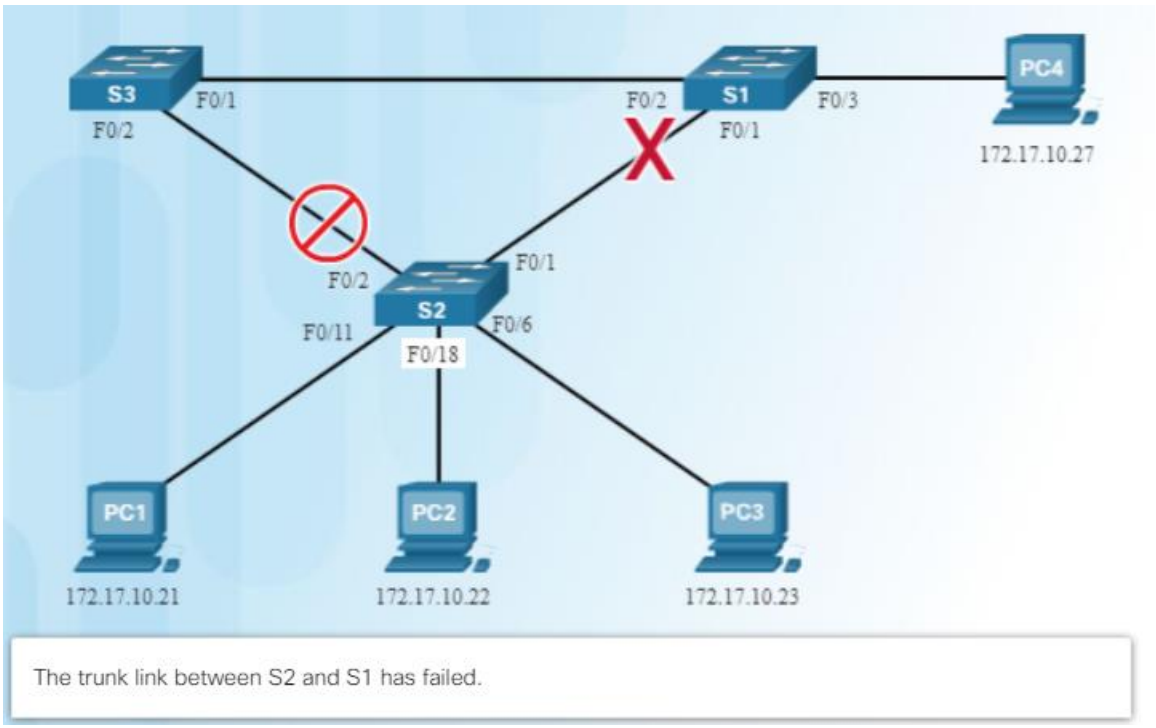
Redundancy increases the availability of the network topology by protecting the network from a single point of failure, such as a failed network cable or switch. When physical redundancy is introduced into a design, loops and duplicate frames occur. Loops and duplicate frames have severe consequences for a switched network. The Spanning Tree Protocol (STP) was developed to address these issues.

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. A port is considered blocked when user data is prevented from entering or leaving that port. This does not include bridge protocol data unit (BPDU) frames that are used by STP to prevent loops. Blocking the redundant paths is critical to preventing loops on the network. The physical paths still exist to provide redundancy, but these paths are disabled to prevent the loops from occurring. If the path is ever needed to compensate for a network cable or switch failure, STP recalculates the paths and unblocks the necessary ports to allow the redundant path to become active.



In the example, all switches have STP enabled:

- PC1 sends a broadcast out onto the network.
- S2 is configured with STP and has set the port for Trunk2 to a blocking state. The blocking state prevents ports from being used to forward user data, which prevents a loop from occurring. S2 forwards a broadcast frame out all switch ports, except the originating port from PC1 and the port for Trunk2.
- S1 receives the broadcast frame and forwards it out all of its switch ports, where it reaches PC4 and S3. S3 forwards the frame out the port for Trunk2 and S2 drops the frame. The Layer 2 loop is prevented.

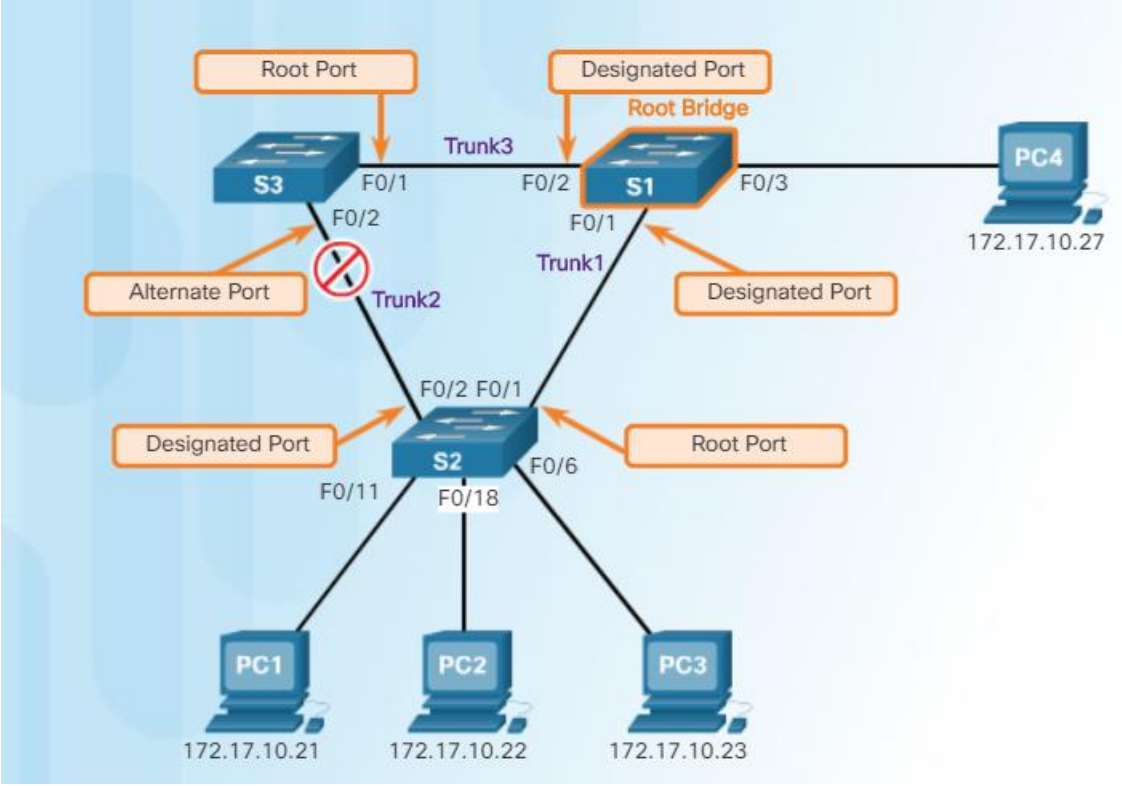


Click the **Play** in Figure to view STP recalculation when a failure occurs.

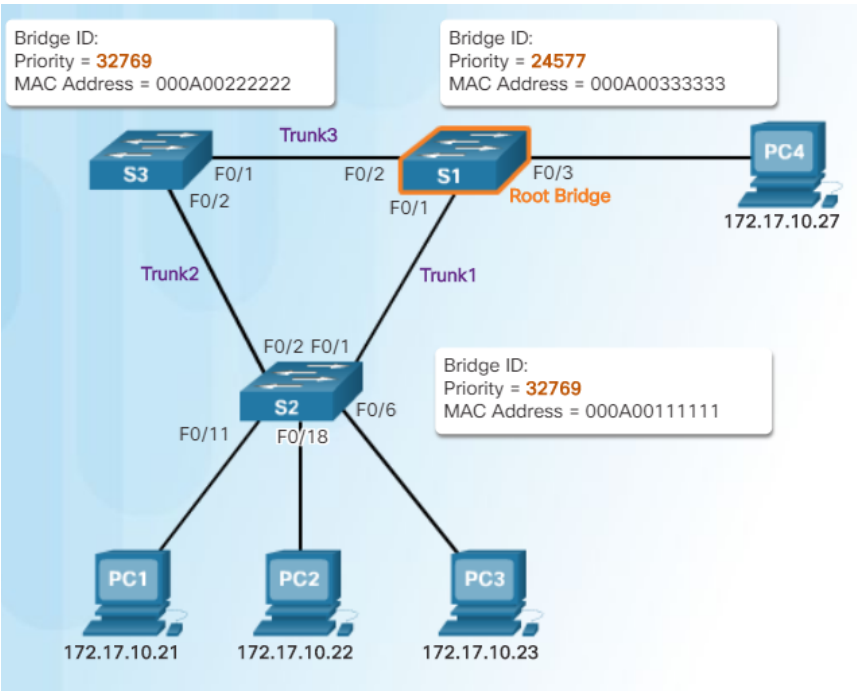
STP prevents loops from occurring by configuring a loop-free path through the network using strategically placed "blocking-state" ports. The switches running STP are able to compensate for failures by dynamically unblocking the previously blocked ports and permitting traffic to traverse the alternate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

Up to now, we have used the term Spanning Tree Protocol and the acronym STP. The usage of the Spanning Tree Protocol term and the STP acronym can be misleading. Many professionals generically use these to refer to 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 particular implementation or standard in context. The latest IEEE documentation on spanning tree (IEEE-802-1D-2004) says, "STP has now been superseded by the Rapid Spanning Tree Protocol (RSTP)."The IEEE uses "STP" to refer to the original implementation of spanning tree and "RSTP" to describe the version of spanning tree specified in IEEE-802.1D-2004. 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. Since 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 proprietary implementations of STP and RSTP.

Spanning Tree Algorithm: Port Roles



Spanning Tree Algorithm: Root Bridge



As shown in Figure above, every spanning tree instance (switched LAN or broadcast domain) has a switch designated as the root bridge. The root bridge serves as a reference point for all spanning tree calculations to determine which redundant paths to block.

An election process determines which switch becomes the root bridge.

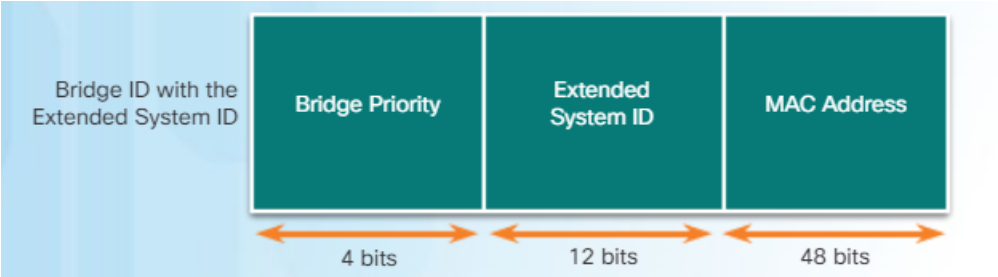


Figure shows the BID fields. The BID is made up of a priority value, an extended system ID, and the MAC address of the switch. The bridge priority value is automatically assigned, but can be modified. The extended system ID is used to specify a VLAN ID or a multiple spanning tree protocol (MSTP) instance ID. The MAC address field initially contains the MAC address of the sending switch.

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 BPDUs contain the switch BID and the root ID.

The switch with the lowest BID will become the root bridge. At first, all switches declare themselves as the root bridge. Eventually, the switches exchange BPDUs, and agree on one root bridge.

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

There is a root bridge elected for each spanning tree instance. It is possible to have multiple distinct root bridges for different sets of VLANs. If all ports on all switches are members of VLAN 1, then there is only one spanning tree instance. The extended system ID includes the VLAN ID, and plays a role in how spanning tree instances are determined.

The BID consists of a configurable bridge priority number and a MAC address. Bridge priority is a value between 0 and 65,535. The default is 32,768. If two or more switches have the same priority, the switch with the lowest MAC address will become the root bridge.

Note: The reason the bridge priority value in Figure 1 displays 32,769 instead of the default value of 32,768 is because STA algorithm also adds the default VLAN number (VLAN 1) to the priority value.

Spanning Tree Algorithm: Root Path Cost

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

Best Paths to the Root Bridge

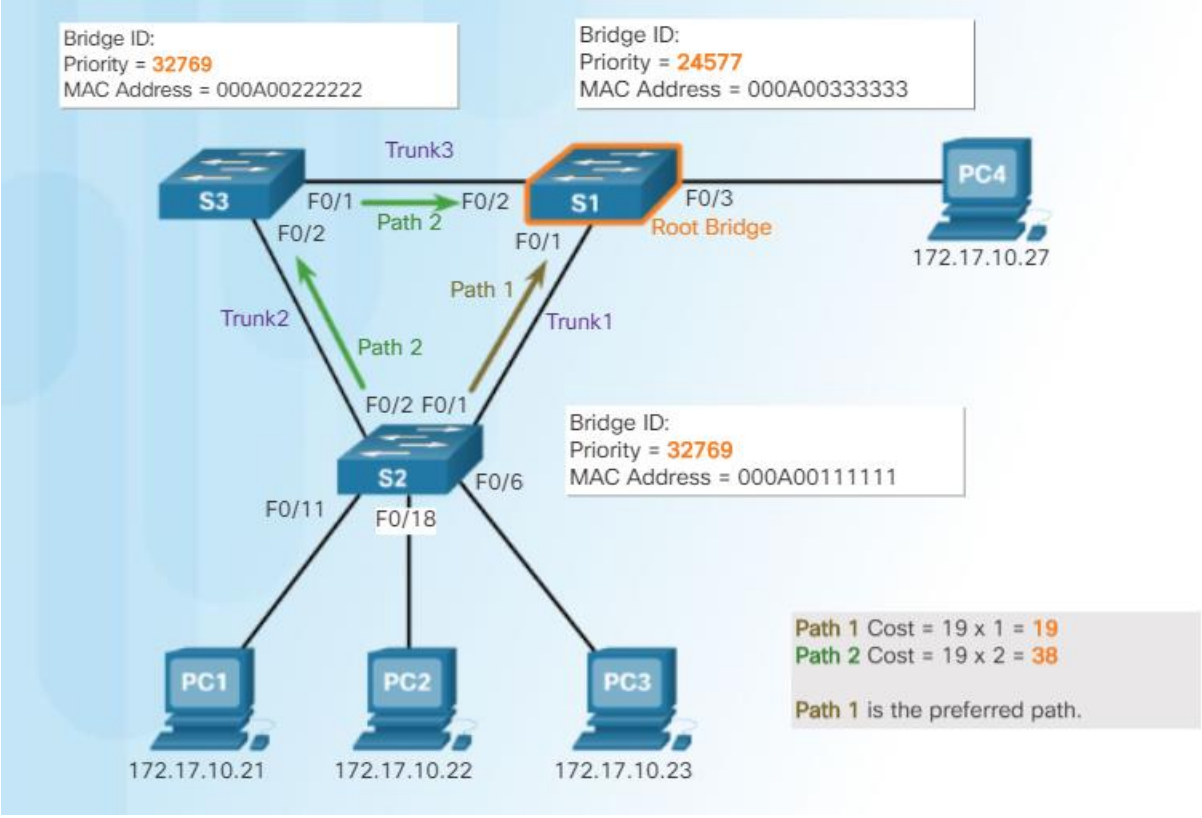
Link Speed	Cost (Revised IEEE Specification)	Cost (Previous IEEE Specification)
10 Gb/s	2	1
1 Gb/s	4	1
100 Mb/s	19	10
10 Mb/s	100	100

Configure Port Cost

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

Reset Port Cost

```
S2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
S2(config)# interface f0/1
S2(config-if)# no spanning-tree cost
S2(config-if)# end
S2#
```




```
S2# show spanning-tree

VLAN001
   Spanning tree enabled protocol ieee
   Root ID     Priority  24577
               Address      000A.0033.3333
               Cost         19
               Port         1
               Hello Time   2 sec Max Age 20 sec Forward Delay 15 sec

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

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

Port Role Decisions for RSTP

Port Role Decisions for RSTP

In the example, switch S1 is the root bridge. Switches S2 and S3 have root ports configured for the ports connecting back to S1.

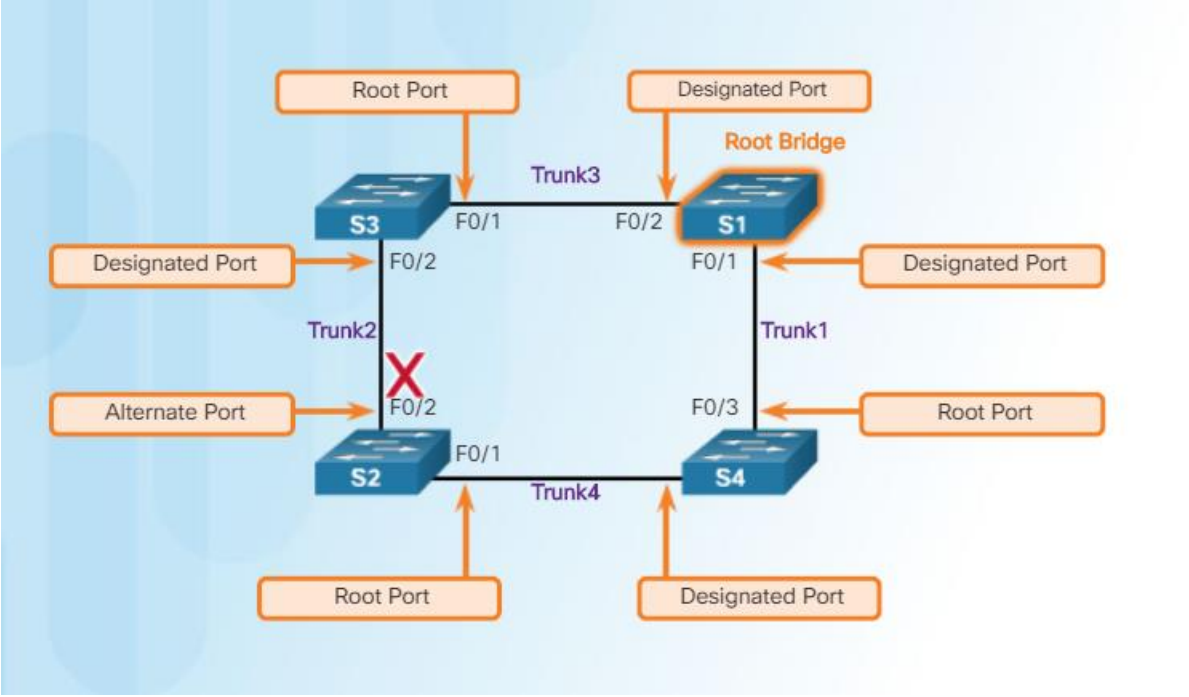
After STP has determined which switch port serves in the root port role on each switch, STP needs to decide which ports have the designated and alternate roles.

The root bridge automatically configures all of its switch ports in the designated role. Other switches in the topology configure their non-root ports as designated or alternate ports.

Designated ports are configured for all LAN segments. When two switches are connected to the same LAN segment, and root ports have already been defined, the two switches have to decide which port to configure as a designated port and which port remains the alternate port.

The switches on the LAN segment exchange BPDU frames, which contain the switch BID. Generally, the switch with the lower BID has its port configured as a designated port while the switch with the higher BID has its port configured as an alternate port. However, keep in mind that the first priority is the lowest path cost to the root bridge and that the sender's BID is used only if the port costs are equal.

Each switch determines which port roles are assigned to each of its ports to create the loop-free spanning tree.



Extended System ID

The bridge ID (BID) is used to determine the root bridge on a network. The BID field of a BPDU frame contains three separate fields:

- Bridge priority
- Extended system ID

- MAC address

Each field is used during the root bridge election.

Bridge Priority

The bridge priority is a customizable value that can be used to influence which switch becomes the root bridge. The switch with the lowest priority, which implies the lowest BID, becomes the root bridge because a lower priority value takes precedence. For example, to ensure that a specific switch is always the root bridge, set the priority to a lower value than the rest of the switches on the network. The default priority value for all Cisco switches is the decimal value 32768. The range is 0 to 61440 in increments of 4096. Valid priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected. A bridge priority of 0 takes precedence over all other bridge priorities.

Extended System ID

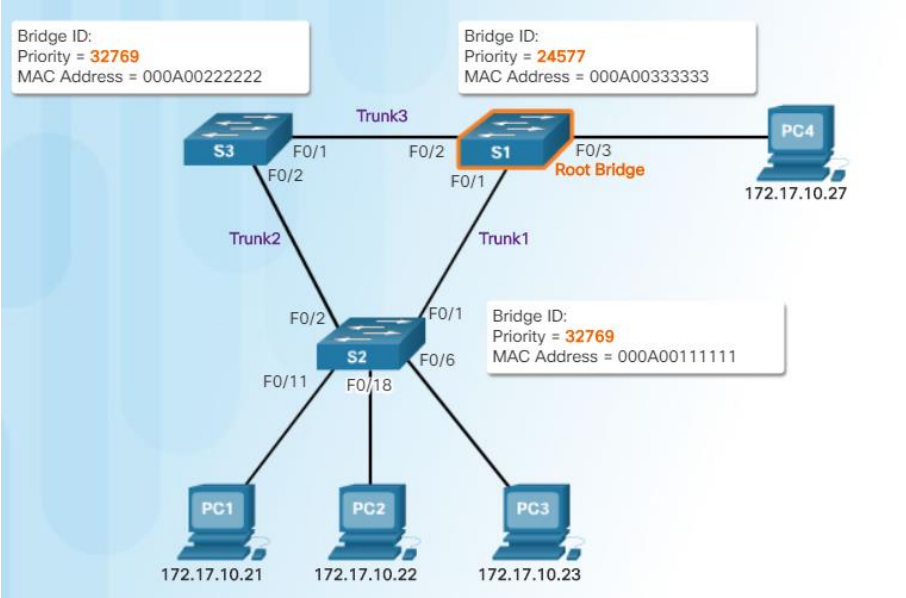
Early implementations of IEEE 802.1D were designed for networks that did not use VLANs. There was a single common spanning tree across all switches. For this reason, in older Cisco switches, the extended system ID could be omitted in BPDU frames. As VLANs became common for network infrastructure segmentation,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

As shown in Figure 1, the bridge priority field is 2 bytes or 16-bits in length. 4-bits are used for the bridge priority and 12-bits are used for the extended system ID, which identifies the VLAN participating in this particular STP process. Using these 12 bits for the extended system ID reduces the bridge priority to 4 bits. This process reserves the rightmost 12 bits for the VLAN ID and the far left 4 bits for the bridge priority. This explains why the bridge priority value can only be configured in multiples of 4096, or 2^12. If the far left bits are 0001, then the bridge priority is 4096. If the far left bits are 1111, then the bridge priority is 61440 (= 15 x 4096). The Catalyst 2960 and 3560 Series switches do not allow the configuration of a bridge priority of 65536 (= 16 x 4096) because it assumes use of a 5th bit that is unavailable due to the use of the extended system ID.

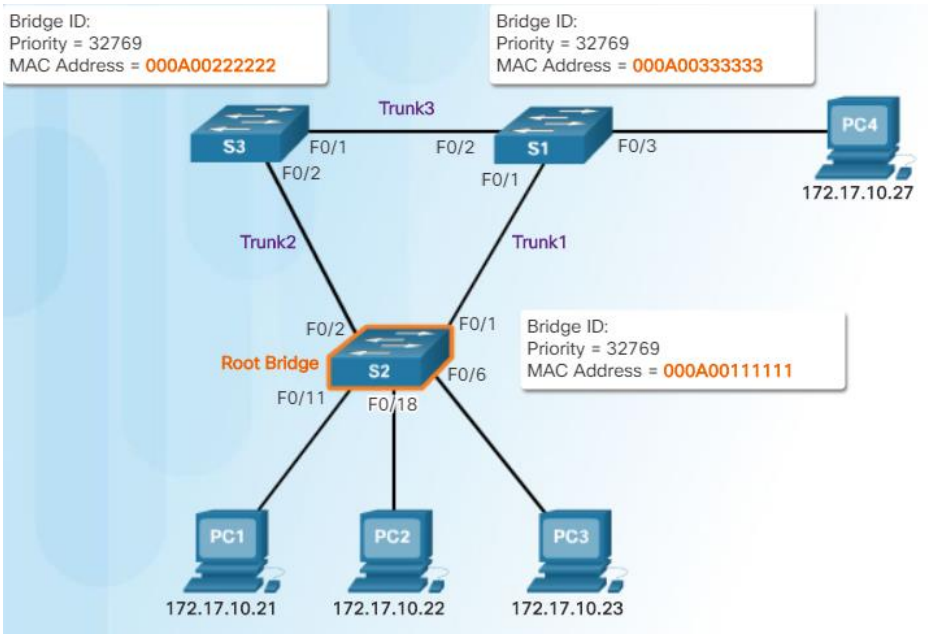
The extended system ID value is a decimal value added to the bridge priority value in the BID to identify the priority and VLAN of the BPDU frame.

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. Initially, all switches are configured with the same default priority value. The MAC address is then the deciding factor as to which switch is going to 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. This also ensures that the addition of new switches to the network does not trigger a new spanning tree election, which can disrupt network communication while a new root bridge is being selected.

S1 has a lower priority than the other switches. Therefore, it is preferred as the root bridge for that spanning tree instance.



When all switches are configured with the same priority, as is the case with all switches kept in the default configuration with a priority of 32768, the MAC address becomes the deciding factor as to which switch becomes the root bridge, as shown in Figure



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

The MAC address with the lowest hexadecimal value is considered to be the preferred root bridge. In the example, S2 has the lowest value for its MAC address and is, therefore, designated as the root bridge for that spanning tree instance.

Cisco Proprietary	PVST • Uses the Cisco proprietary ISL trunking protocol • Each VLAN has an instance of spanning tree • Ability to load balance traffic at layer-2 • Includes extensions BackboneFast, UplinkFast, and PortFast
	PVST+ • Supports ISL and IEEE 802.1Q trunking • Supports Cisco proprietary STP extensions • Adds BPDU guard and Root guard enhancements
	rapid-PVST+ • Based on IEEE802.1w standard • Has faster convergence than 802.1D
IEEE Standard	RSTP • Introduced in 1982 provides faster convergence than 802.1D • Implements generic versions of the Cisco proprietary STP extensions • IEEE has incorporated RSTP into 802.1D, identifying the specification as IEEE 802.1D-2004
	MSTP • Multiple VLANs can be mapped to the same spanning-tree instance • Inspired by the Cisco Multiple Instances Spanning Tree Protocol (MISTP) • IEEE 802.1Q-2003 now includes MSTP

Protocol	Standard	Resources Needed	Convergence	Tree Calculation
STP	802.1D	Low	Slow	All VLANs
PVST+	Cisco	High	Slow	Per VLAN
RSTP	802.1w	Medium	Fast	All VLANs
Rapid PVST+	Cisco	Very high	Fast	Per VLAN
MSTP	802.1s, Cisco	Medium or high	Fast	Per Instance

Configuring and Verifying the Bridge ID

When an administrator wants a specific switch to become a root bridge, the bridge priority value must be adjusted to ensure it is lower than the bridge priority values of all the other switches on the network. There are two different methods to configure the bridge priority value on a Cisco Catalyst switch.

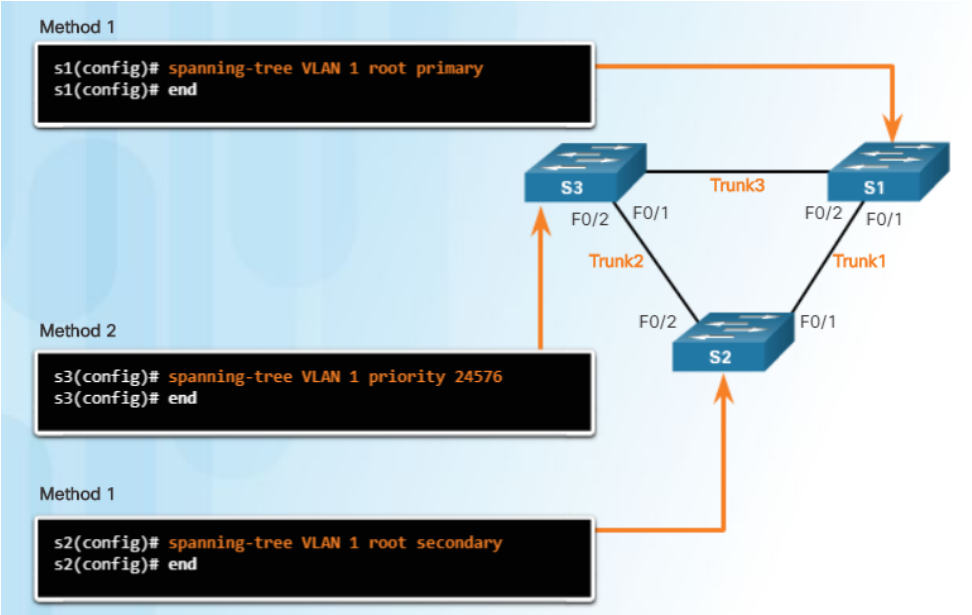
Method 1

To ensure that the switch has the lowest bridge priority value, use the **spanning-tree vlan** *vlan-id* **root primary** command in global configuration mode. The priority for the switch is set to the predefined value of 24,576 or to the highest multiple of 4,096, less than the lowest bridge priority detected on the network.

If an alternate root bridge is desired, use the**spanning-tree vlan** *vlan-id* **root secondary** global configuration mode command. This command sets the priority for the switch to the predefined value of

28,672. This ensures that the alternate switch becomes the root bridge if the primary root bridge fails. This assumes that the rest of the switches in the network have the default 32,768 priority value defined.

In Figure , S1 has been assigned as the primary root bridge using the **spanning-tree vlan 1 root primary** command, and S2 has been configured as the secondary root bridge using the **spanning-tree vlan 1 root secondary** command.



Method 2

Another method for configuring the bridge priority value is using the **spanning-tree vlan** *vlan-id* **priority** *value* global configuration mode command. This command gives more granular control over the bridge priority value. The priority value is configured in increments of 4,096 between 0 and 61,440.

In the example, S3 has been assigned a bridge priority value of 24,576 using the **spanning-tree vlan 1 priority 24576** command.

To verify the bridge priority of a switch, use the**show spanning-tree** command. In Figure, the priority of the switch has been set to 24,576. Also notice that the switch is designated as the root bridge for the spanning tree instance.

```
S3# show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     000A.0033.3333
              This bridge is the root
              Hello Time  2 sec Max Age 20 sec Forward Delay 15 sec
  Bridge ID   Priority    24577  (priority  24576  sys-id-ext  1)
              Address     000A.0033.3333
              Hello Time  2 sec Max Age 20 sec Forward Delay 15 sec
              Aging Time  300

Interface     Role    Sts    Cost     Prio.Nbr   Type
------------  -----   ----   -----    ---------- ------
Fa0/1         Desg    FWD    4        128.1      p2p
Fa0/2         Desg    FWD    4        128.2      p2p
S3#
```

PVST+ Load Balancing

The topology in Figure 1 shows three switches with 802.1Q trunks connecting them. There are two VLANs, 10 and 20, that are being trunked across these links. The goal is to configure S3 as the root bridge for VLAN 20 and S1 as the root bridge for VLAN 10. Port F0/3 on S2 is the forwarding port for VLAN 20 and the blocking port for VLAN 10. Port F0/2 on S2 is the forwarding port for VLAN 10 and the blocking port for VLAN 20.

```
S3(config)# spanning-tree vlan 20 root primary
```

This command forces S3 to be the primary root for VLAN 20.

```
S3(config)# spanning-tree vlan 10 root secondary
```

This command forces S3 to be the secondary root for VLAN 10.

```
S1(config)# spanning-tree vlan 10 root primary
```

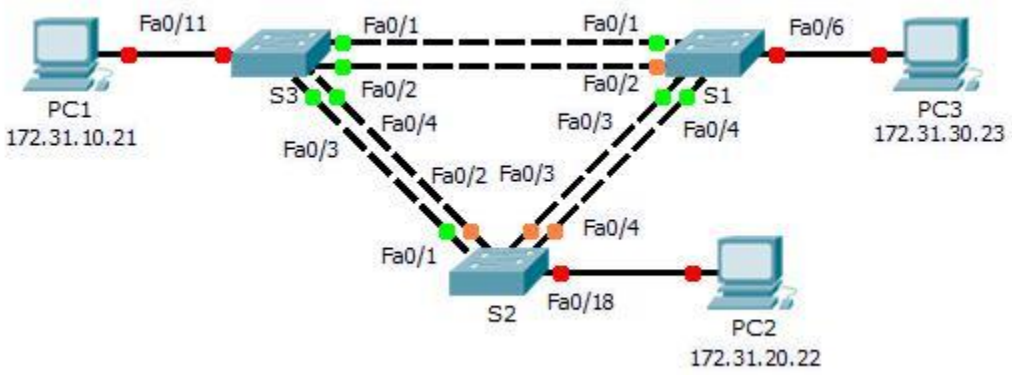
This command forces S1 to be the primary root for VLAN 10.

```
S1(config)# spanning-tree vlan 20 root secondary
```

This command forces S1 to be the secondary root for VLAN 20.

Packet Tracer – Configuring PVST+

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
S1	VLAN 99	172.31.99.1	255.255.255.0	N/A
S2	VLAN 99	172.31.99.2	255.255.255.0	N/A
S3	VLAN 99	172.31.99.3	255.255.255.0	N/A
PC1	NIC	172.31.10.21	255.255.255.0	172.31.10.254
PC2	NIC	172.31.20.22	255.255.255.0	172.31.20.254
PC3	NIC	172.31.30.23	255.255.255.0	172.31.30.254

Switch Port Assignment Specifications

Ports	Assignments	Network
S1 F0/6	VLAN 30	172.17.30.0/24
S2 F0/18	VLAN 20	172.17.20.0/24
S3 F0/11	VLAN 10	172.17.10.0/24

Objectives

Part 1: Configure VLANs

Part 2: Configure Spanning Tree PVST+ and Load Balancing

Part 3: Configure PortFast and BPDU Guard

Background

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

Part 1: Configure VLANs

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

Refer to the topology diagram to determine which switch ports (**S1, S2,** and **S3**) are activated for end-user device access. These three ports will be configured for access mode and enabled with the **no shutdown** command.

Step 2: Create VLANs.

Using the appropriate command, create VLANs 10, 20, 30, 40, 50, 60, 70, 80, and 99 on all of the switches.

Step 3: Assign VLANs to switch ports.

Port assignments are listed in the table at the beginning of the activity. Save your configurations after assigning switch ports to the VLANs.

Step 4: Verify the VLANs.

Use the **show vlan brief** command on all switches to verify that all VLANs are registered in the VLAN table.

Step 5: Assign the trunks to native VLAN 99.

Use the appropriate command to configure ports F0/1 to F0/4 on each switch as trunk ports, and assign these trunk ports to native VLAN 99.

Step 6: Configure the management interface on all three switches with an address.

Verify that the switches are correctly configured by pinging between them.

Part 2: Configure Spanning Tree PVST+ and Load Balancing

Because there is a separate instance of the spanning tree for every active VLAN, a separate root election is conducted for each instance. If the default switch priorities are used in root selection, the same root is elected for every spanning tree instance, as we have seen. This could lead to an inferior design. Some reasons to control the selection of the root switch include:

- The root switch is responsible for generating BPDUs for STP 802.1D and is the focal point for spanning tree to control traffic. The root switch must be capable of handling this additional load.
- The placement of the root defines the active switched paths in the network. Random placement is likely to lead to suboptimal paths. Ideally the root is in the distribution layer.
- Consider the topology used in this activity. Of the six trunks configured, only three are carrying traffic. While this prevents loops, it is a waste of resources. Because the root can be defined on the basis of the VLAN, you can have some ports blocking for one VLAN and forwarding for another. This is demonstrated below.

Step 1: Configure STP mode.

Use the **spanning-tree mode** command to configure the switches so they use PVST as the STP mode.

Step 2: Configure Spanning Tree PVST+ load balancing.

d. Configure **S1** to be the primary root for VLANs 1, 10, 30, 50, and 70. Configure **S3** to be the primary root for VLANs 20, 40, 60, 80, and 99. Configure **S2** to be the secondary root for all VLANs.

e. Verify your configurations using the **show spanning-tree** command.

Part 3: Configure PortFast and BPDU Guard

Step 1: Configure PortFast on the switches.

PortFast causes a port to enter the forwarding state almost immediately by dramatically decreasing the time of the listening and learning states. PortFast minimizes the time it takes for the server or workstation to come online. Configure PortFast on the switch interfaces that are connected to PCs.

Step 2: Configure BPDU guard on the switches.

The STP PortFast BPDU guard enhancement allows network designers to enforce the STP domain borders and keep the active topology predictable. The devices behind the ports that have STP PortFast enabled are unable to influence the STP topology. At the reception of BPDUs, the BPDU guard operation disables the port that has PortFast configured. The BPDU guard transitions the port into the err-disable state, and a message appears on the console. Configure BPDU guard on switch interfaces that are connected to PCs.

Step 3: Verify your configuration.

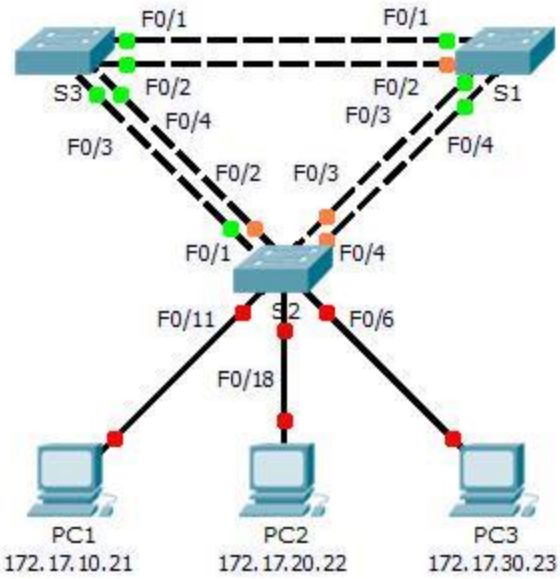
Use the **show running-configuration** command to verify your configuration.

Cisco IOS Command Syntax

Enter global configuration mode.	`configure terminal`
Configure Rapid PVST+ spanning-tree mode.	`spanning-tree mode rapid-pvst`
Enter interface configuration mode and specify an interface to configure. Valid interfaces include physical ports, VLANs, and port channels.	`interface` *`interface-id`*
Specify that the link type for this port is point-to-point.	`spanning-tree link-type point-to-point`
Return to privileged EXEC mode.	`end`
Clear all detected STP.	`clear spanning-tree detected-protocols`

Packet Tracer – Configuring Rapid PVST+

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
S1	VLAN 99	172.17.99.11	255.255.255.0	N/A
S2	VLAN 99	172.17.99.12	255.255.255.0	N/A
S3	VLAN 99	172.17.99.13	255.255.255.0	N/A
PC1	NIC	172.17.10.21	255.255.255.0	172.17.10.254
PC2	NIC	172.17.20.22	255.255.255.0	172.17.20.254
PC3	NIC	172.17.30.23	255.255.255.0	172.17.30.254

Switch Port Assignment Specifications

Ports	Assignments	Network
S2 F0/6	VLAN 30	172.17.30.0/24
S2 F0/18	VLAN 20	172.17.20.0/24
S2 F0/11	VLAN 10	172.17.10.0/24

Objectives

Part 1: Configure VLANs

Part 2: Configure Rapid Spanning Tree PVST+ Load balancing

Part 3: Configure PortFast and BPDU Guard

Background

In this activity, you will configure VLANs and trunks, Rapid Spanning Tree PVST+, primary and secondary root bridges, and examine the configuration results. You will also optimize the network by configuring PortFast, and BPDU Guard on edge ports.

Part 1: Configure VLANs

Step 1: Enable the user ports on S2 in access mode.

Refer to the topology diagram to determine which switch ports on **S2** are activated for end-user device access. These three ports will be configured for access mode and enabled with the **no shutdown** command.

Step 2: Create VLANs.

Using the appropriate command, create VLANs 10, 20, 30, 40, 50, 60, 70, 80, and 99 on all of the switches.

Step 3: Assign VLANs to switch ports.

Port assignments are listed in the table at the beginning of the activity. Save your configurations after assigning switch ports to the VLANs.

Step 4: Verify the VLANs.

Use the **show vlan brief** command on all switches to verify that all VLANs are registered in the VLAN table.

Step 5: Assign the trunks to native VLAN 99.

Use the appropriate command to configure ports F0/1 to F0/4 on each switch as trunk ports and assign these trunk ports to native VLAN 99.

Step 6: Configure the management interface on all three switches with an address.

Verify that the switches are correctly configured by pinging between them.

Part 2: Configure Rapid Spanning Tree PVST+ Load Balancing

The Rapid Spanning Tree Protocol (RSTP; IEEE 802.1w) can be seen as an evolution of the 802.1D standard more so than a revolution. The 802.1D terminology remains primarily the same. Most parameters have been left unchanged so users familiar with 802.1D can rapidly configure the new protocol comfortably. In most cases, RSTP performs better than proprietary extensions of Cisco without any additional configuration.
802.1w can also revert back to 802.1D in order to interoperate with legacy bridges on a per-port basis.

Step 1: Configure STP mode.

Use the **spanning-tree mode** command to configure the switches to use rapid PVST as the STP mode.

Step 2: Configure Rapid Spanning Tree PVST+ load balancing.

Configure **S1** to be the primary root for VLANs 1, 10, 30, 50, and 70. Configure **S3** to be the primary root for VLANs 20, 40, 60, 80, and 99. Configure **S2** to be the secondary root for all of the VLANs.

Verify your configurations by using the **show spanning-tree** command.

Part 3: Configure PortFast and BPDU Guard

Step 1: Configuring PortFast on S2.

PortFast causes a port to enter the forwarding state almost immediately by dramatically decreasing the time of the listening and learning states. PortFast minimizes the time it takes for the server or workstation to come online. Configure PortFast on **S2** interfaces that are connected to PCs.

Step 2: Configuring BPDU Guard on S2.

The STP PortFast BPDU Guard enhancement allows network designers to enforce the STP domain borders and keep the active topology predictable. The devices behind the ports that have STP PortFast enabled are not able to influence the STP topology. At the reception of BPDUs, the BPDU Guard operation disables the port that has PortFast configured. The BPDU Guard transitions the port into err-disable state, and a message appears on the console. Configure BPDU Guard on **S2** interfaces that are connected to PCs.

Step 3: Verify your configuration.

Use the **show run** command to verify your configuration.

Lab8: Link aggregation: EtherChannel

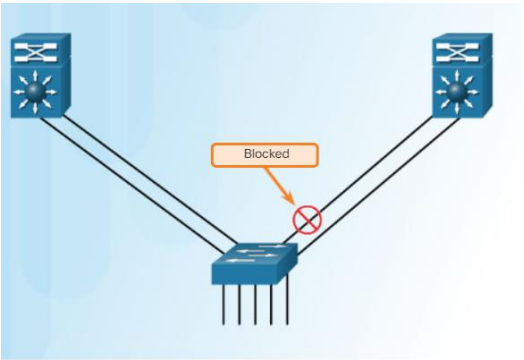
Introduction to Link Aggregation

In the figure, traffic coming from several links (usually 100 or 1000 Mb/s) aggregates on the access switch and must be sent to distribution switches. Because of the traffic aggregation, links with higher bandwidth must be available between the access and distribution switches.

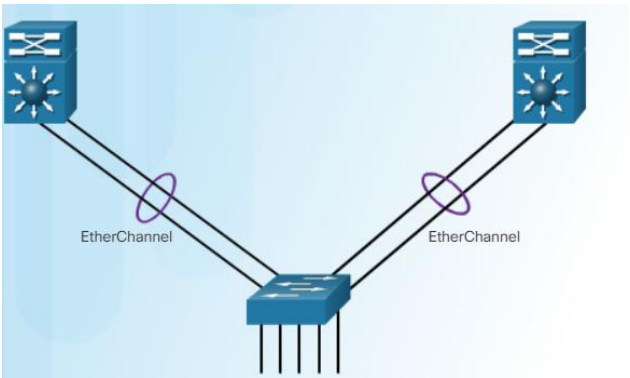
It may be possible to use faster links, such as 10 Gb/s, on the aggregated link between the access and distribution layer switches. However, adding faster links is expensive. Additionally, as the speed increases on the access links, even the fastest possible port on the aggregated link is no longer fast enough to aggregate the traffic coming from all access links.

It is also possible to combine the number of physical links between the switches to increase the overall speed of switch-to-switch communication. However, by default, STP is enabled on Layer 2 devices such as switches. STP will block redundant links to prevent switching loops.

For these reasons, the best solution is to implement an EtherChannel configuration



By default, STP will block redundant links.



Advantages of EtherChannel

1. Most configuration tasks can be done on the EtherChannel interface instead of on each individual port, ensuring configuration consistency throughout the links.
2. 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.
3. 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 to destination MAC load balancing, or source IP to destination IP load balancing.
4. EtherChannel creates an aggregation that is seen as one logical link
5. 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.

Implementation Restrictions

1. Interface types cannot be mixed; Fast Ethernet and Gigabit Ethernet cannot be mixed within a single EtherChannel.
2. The EtherChannel provides full-duplex bandwidth up to 800 Mb/s (Fast EtherChannel) or 8 Gb/s (Gigabit EtherChannel) between one switch and another switch or host.
3. Each EtherChannel can consist of up to eight compatibly-configured Ethernet ports. The Cisco IOS switch can currently support six EtherChannels
4. EtherChannel creates a one-to-one relationship; one EtherChannel link connects only two devices. An EtherChannel link can be created between two switches (A trunked port can be part of an EtherChannel bundle) or an EtherChannel link can be created between an EtherChannel-enabled server and a switch.
5. 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.
6. Each EtherChannel has a logical port channel interface, A configuration applied to the port channel interface affects all physical interfaces that are assigned to that interface.

Port Aggregation Protocol

- EtherChannels can be formed using one of two protocols, PAgP or LACP.
- PAgP is a Cisco-proprietary protocol that aids in the automatic creation of EtherChannel links. When an EtherChannel link is configured using PAgP, PAgP packets are sent between EtherChannel-capable ports to negotiate the forming of a channel. When PAgP identifies matched Ethernet links, it groups the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a **single port**.
- When enabled, PAgP also manages the EtherChannel. PAgP packets are sent every 30 seconds. PAgP checks for configuration consistency and manages link additions and failures between two switches. It ensures that when an EtherChannel is created, all ports have the same type of configuration.
- **Note**: In EtherChannel, it is mandatory that all ports have the same speed, duplex setting, and VLAN information. Any port modification after the creation of the channel also changes all other channel ports.

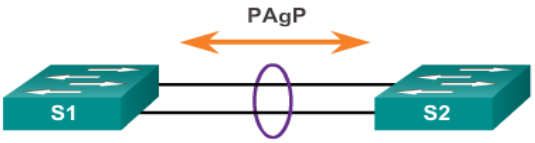
Port Aggregation Protocol

- PAgP helps create the EtherChannel link by detecting the configuration of each side and ensuring that links are compatible so that the EtherChannel link can be enabled when needed. The figure shows the modes for PAgP.

  On - This mode forces the interface to channel without PAgP. Interfaces configured in the on mode do not exchange PAgP packets.

  PAgP desirable - This mode places an interface in an active negotiating state in which the interface initiates negotiations with other interfaces by sending PAgP packets.

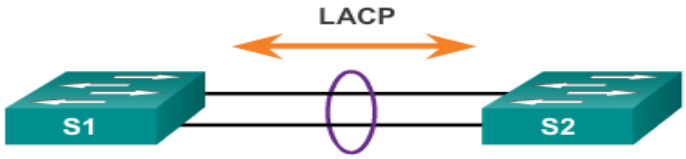
  PAgP auto - This PAgP mode places an interface in a passive negotiating state in which the interface responds to the PAgP packets that it receives.
- The modes must be compatible on each side, each pair of modes represnet a case as the following figure.



S1	S2	Channel Establishment
On	On	Yes
Auto/Desirable	Desirable	Yes
On/Auto/Desirable	Not Configured	No
On	Desirable	No
Auto/On	Auto	No

Link Aggregation Control Protocol

- 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 peer. It performs a function similar to PAgP with Cisco EtherChannel. Because LACP is an IEEE standard, it can be used to facilitate EtherChannels in multivendor environments. On Cisco devices, both protocols are supported.
- Note: LACP was originally defined as IEEE 802.3ad. However, LACP is now defined in the newer IEEE 802.1AX standard for local and metropolitan area networks.



S1	S2	Channel Establishment
On	On	Yes
Active/Passive	Active	Yes
On/Active/Passive	Not Configured	No
On	Active	No
Passive/On	Passive	No

LACP provides the same negotiation benefits as PAgP. 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 figure shows the modes for LACP.

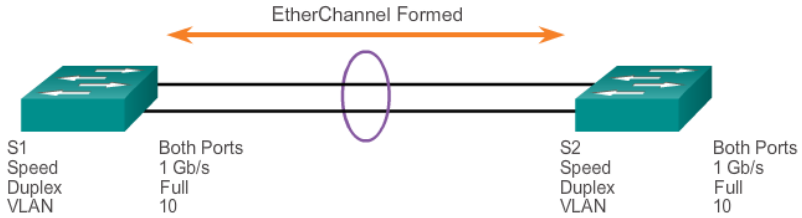
- **On** - This mode forces the interface to channel without LACP. Interfaces configured in the on mode do not exchange LACP packets.
- **LACP active** - This LACP mode places a port in an active negotiating state. In this state, the port initiates negotiations with other ports by sending LACP packets.
- **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.

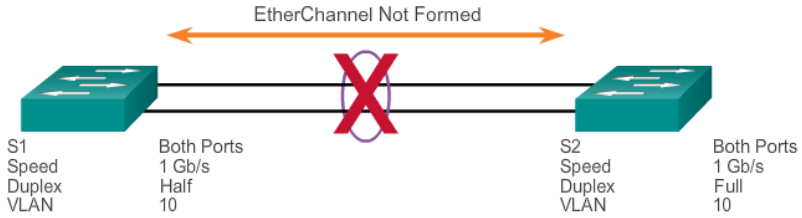
Link Aggregation Configuration

The following guidelines and restrictions are useful for configuring EtherChannel:

- **EtherChannel support** - All Ethernet interfaces on all modules must support EtherChannel with no requirement that interfaces be physically contiguous, or on the same module.

- **Speed and duplex** - Configure all interfaces in an EtherChannel to operate at the same speed and in the same duplex mode, as shown in the figure.
- **VLAN match** - All interfaces in the EtherChannel bundle must be assigned to the same VLAN, or be configured as a trunk (also shown in the figure).
- **Range of allowed 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.





Configuring Interfaces

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

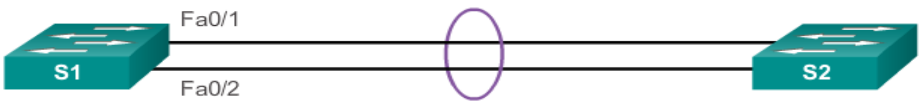
A good practice is to start by shutting down those interfaces, so that any incomplete configuration does not create activity on the link.

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

Note: EtherChannel is disabled by default.

```
S1(config)# interface range FastEthernet0/1 - 2
S1(config-if-range)# channel-group 1 mode active
Creating a port-channel interface Port-channel 1
S1(config-if-range)# interface port-channel 1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk allowed vlan 1,2,20
```

Creates EtherChannel and configures trunk.



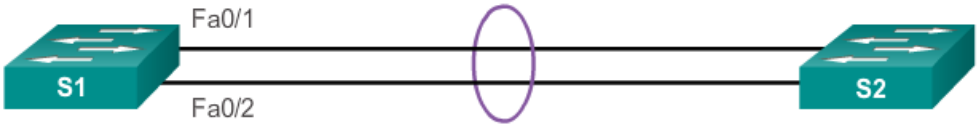
Verifying EtherChannel

- There are a number of commands to verify an EtherChannel configuration. First, the **show interface port-channel** command displays the general status of the port channel interface.

In the Figure, the Port Channel 1 interface is up.

```
S1# show interface port-channel1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 0cd9.96e8.8a02 (bia
0cd9.96e8.8a02)
  MTU 1500 bytes, BW 200000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
<Output omitted>
```

Verifies the interface status.



- When several port channel interfaces are configured on the same device, the **show etherchannel summary** command used to simply display one line of information per port channel. I

n the figure the switch has one EtherChannel configured; group 1 uses LACP.

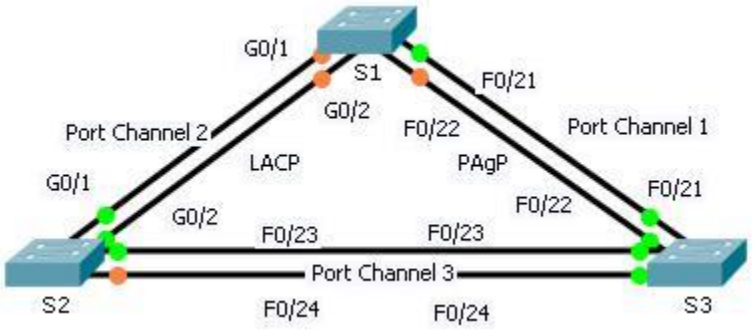
The interface bundle consists of the FastEthernet0/1 and FastEthernet0/2 interfaces. The group is a Layer 2 EtherChannel and that it is in use, as indicated by the letters SU next to the port channel number.

- The **show etherchannel port-channel** command used to display information about a specific port channel interface, as shown in the figure .

In the example, the Port Channel 1 interface consists of two physical interfaces, FastEthernet0/1 and FastEthernet0/2. It uses LACP in active mode. It is properly connected to another switch with a compatible configuration, which is why the port channel is said to be in use.

Packet Tracer – Configuring EtherChannel

Topology



Objectives

Part 1: Configure Basic Switch Settings

Part 2: Configure an EtherChannel with Cisco PAgP

Part 3: Configure an 802.3ad LACP EtherChannel

Part 4: Configure a Redundant EtherChannel Link

Background

Three switches have just been installed. There are redundant uplinks between the switches. Usually, only one of these links could be used; otherwise, a bridging loop might occur. However, using only one link utilizes only half of the available bandwidth. EtherChannel allows up to eight redundant links to be bundled together into one logical link. In this lab, you will configure Port Aggregation Protocol (PAgP), a Cisco EtherChannel protocol, and Link Aggregation Control Protocol (LACP), an IEEE 802.3ad open standard version of EtherChannel.

Part 1: Configure Basic Switch Settings

Step 1: Configure basic switch parameters.

n. Assign each switch a hostname according to the topology diagram.

o. Configure all required ports as trunks, depending on the connections between devices.

Note: If the ports are configured with dynamic auto mode, and you do not set the mode of the ports to trunk, the links do not form trunks and remain access ports. The default mode on a 2960 switch is dynamic auto.

Part 2: Configure an EtherChannel with Cisco PAgP

Note: When configuring EtherChannels, it is recommended to shut down the physical ports being grouped on both devices before configuring them into channel groups. Otherwise, the EtherChannel Misconfig Guard may place these ports into err-disabled state. The ports and port channels can be re-enabled after EtherChannel is configured.

Step 1: Configure Port Channel 1.

o. The first EtherChannel created for this activity aggregates ports F0/22 and F0/21 between **S1** and **S3**. Use the **show interfaces trunk** command to ensure that you have an active trunk link for those two links.

p. On both switches, add ports F0/21 and F0/22 to Port Channel 1 with the **channel-group 1 mode desirable** command. The **mode desirable** option enables the switch to actively negotiate to form a PAgP link.

q. Configure the logical interface to become a trunk by first entering the **interface port-channel** *number* command and then the **switchport mode trunk** command. Add this configuration to both switches.

Step 2: Verify Port Channel 1 status.

i. Issue the **show etherchannel summary** command to verify that EtherChannel is working on both switches. This command displays the type of EtherChannel, the ports utilized, and port states.

j. If the EtherChannel does not come up, shut down the physical interfaces on both ends of the EtherChannel and then bring them back up again. This involves using the **shutdown** command on those interfaces, followed by a **no shutdown** command a few seconds later.

> The **show interfaces trunk** and **show spanning-tree** commands also show the port channel as one logical link.

Part 3: Configure an 802.3ad LACP EtherChannel

Step 1: Configure Port Channel 2.

j. In 2000, the IEEE released 802.3ad, which is an open standard version of EtherChannel. Using the previous commands, configure the link between **S1** and **S2** on ports G0/1 and G0/2 as an LACP EtherChannel. You must use a different port channel number on **S1** than 1, because you already used that in the previous step. To configure a port channel as LACP, use the interface configuration mode **channel-group** *number* **mode active** command. Active mode indicates that the switch actively tries to negotiate that link as LACP, as opposed to PAgP.

Step 2: Verify Port Channel 2 status.

• Use the **show** commands from Part 1 Step 2 to verify the status of Port Channel 2. Look for the protocol used by each port.

Part 4: Configure a Redundant EtherChannel Link

Step 1: Configure Port Channel 3.

There are various ways to enter the **channel-group** *number* **mode** command:

```
S2(config)# interface range f0/23 - 24
S2(config-if-range)# channel-group 3 mode ?
  active     Enable LACP unconditionally
  auto       Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on         Enable Etherchannel only
  passive    Enable LACP only if a LACP device is detected
```

• On switch **S2**, add ports F0/23 and F0/24 to Port Channel 3 with the **channel-group 3 mode passive** command. The **passive** option indicates that you want the switch to use LACP only if another LACP device is detected. Statically configure Port Channel 3 as a trunk interface.

k. On switch **S3**, add ports F0/23 and F0/24 to Port Channel 3 with the **channel-group 3 mode active** command. The **active** option indicates that you want the switch to use LACP unconditionally. Statically configure Port Channel 3 as a trunk interface.

Step 2: Verify Port Channel 3 status.

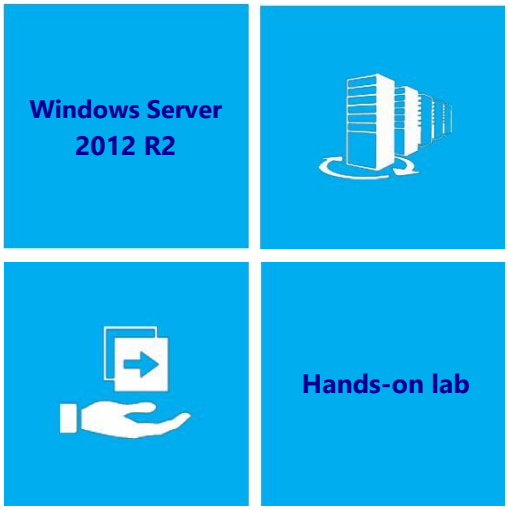
m. Use the **show** commands from Part 1 Step 2 to verify the status of Port Channel 3. Look for the protocol used by each port.

n. Port Channel 2 is not operative because spanning tree protocol placed some ports into blocking mode. Unfortunately, those ports were Gigabit ports. To restore these ports, configure **S1** to be **primary** root for VLAN 1 or set the priority to **24576**.

Windows Server 2012 R2

Microsoft

Windows Server 2012 R2 – IP Address Management (IPAM)



IP Address Management (IPAM) provides a single console to plan, design and administer network services and IP address spaces – physical and virtual. In this lab, you will learn more about how you can use IPAM in your organization to manage physical and virtual address space, delegate permissions in a multi-user environment, perform advanced DHCP configurations, and leverage IPAM PowerShell cmdlets for automating routine operations.

Introduction

Estimated time to complete this lab

Objectives

After completing this lab, you will be able to:

- Configure role-based access control and delegated administration.
- Manage policy-based DHCP addresses and option assignments by using IPAM.
- Automate IP address lifecycle management.
- Manage DHCP MAC address filters by using IPAM.
- Manager DHCP superscopes by using IPAM.

Prerequisites

Before working on this lab, you must have:

1. Experience working with Windows Server (any version).
2. Experience with network and server administration.
3. An understanding of TCP/IP, in particular DHCP.

Overview of the lab

IP Address Management (IPAM) provides a single console to plan, design and administer network services and IP address spaces – physical and virtual. In this lab, you will learn more about how you can use IPAM in your organization to manage physical and virtual address space, delegate permissions in a multi-user environment, perform advanced DHCP configurations, and leverage IPAM PowerShell cmdlets for automating routine operations.

Virtual machine technology

This lab is completed using virtual machines that run on Windows Server 2012 Hyper-V technology. To log on to the virtual machines, press CTRL+ALT+END and enter your logon credentials.

Computers in this lab

This lab uses computers as described in the following table. Before you begin the lab, you must ensure that the virtual machines are started and then log on to the computers.

Computer	Role	Configuration
DC	Domain controller	Domain controller with Active Directory and DNS

Computer	Role	Configuration
SERVER1	DHCP server used in lab exercises	Server with DHCP
SERVER2	DHCP server used in lab exercises	Server with DHCP
SERVER3	IPAM server	Server with IPAM
Admin	Client computer for the lab	Windows 8.1 client with RSAT

All user accounts in this lab use the password **Passw0rd!**

Note regarding pre-release software

Portions of this lab may include software that is not yet released, and as such may still contain active or known issues. While every effort has been made to ensure this lab functions as written, unknown or unanticipated results may be encountered as a result of using pre-release software.

Note regarding user account control

Some steps in this lab may be subject to user account control. User account control is a technology which provides additional security to computers by requesting that users confirm actions that require administrative rights. Tasks that generate a user account control confirmation are denoted using a shield icon. If you encounter a shield icon, confirm your action by selecting the appropriate button in the dialog box that is presented.

Note on activation

The virtual machines for these labs may have been built by using software that has not been activated. This is by design in the lab to prevent the redistribution of activated software. The unactivated state of software has been taken into account in the design of the lab. Consequently, the lab is in no way affected by this state. For operating systems other than Windows 8.1, click Cancel or Close if prompted by an activation dialog box. If you are prompted by an Activate screen for Windows 8.1, press the Windows key to display the Start screen.

Exercise 1: Installing and Configuring IPAM

In this exercise, you will install and configure an IPAM server, and then configure it to manage IP services including DHCP, DNS, and Active Directory. IPAM deployment includes many schedule-based activities which run automatically to populate data and configure servers. These tasks include scheduled tasks and group policy refreshes. During this exercise, you will force many of these tasks to run automatically and also perform some manual tasks to speed up the overall deployment.

Install and configure DHCP on Server1 and Server2

In this task, you will use a script to install and configure DHCP on Server2 and Server3. The script will first switch the computers to use static IP addresses, and then install the DHCP role. Finally it will perform DHCP initial configuration which includes adding local groups and performing Active Directory authorization.

- Begin this task logged onto **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. On the taskbar, right-click the **Windows PowerShell** icon, and then click **Windows PowerShell ISE**.
2. In Windows PowerShell ISE, open the file **C:\LabFiles\IPAM-Setup.ps1**.
3. Press F5, and then press ENTER to run the file.
   - This file will set both servers to use static IP addresses, and then install and configure the DHCP service.
   - After a few minutes, Windows PowerShell will display the running script below the Windows PowerShell window.
   - Windows PowerShell ISE will indicate when the tasks are completed.

Install IPAM on Server3

In this task, you will install the IPAM service on Server3.

- Begin this task logged onto **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. On **Server3**, in Server Manager, click **Manage**, and then click **Add Roles and Features**.
2. On the Before you Begin page, click **Next**.
3. On the Installation Type page, click **Next**.
4. On the Server Selection page, click **Next**.
5. On the Server Roles page, click **Next**.
6. On the Features page, check **IP Address Management (IPAM) Server**, click **Add Features**, and then click **Next**.
7. Click **Install**, and then when the installation completes, click **Close**.

Configure IPAM on Server3

In this task, you will perform the initial configuration of IPAM on Server3.

✎ Begin this task logged onto **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, click **IPAM**.
2. In IPAM, click **Provision the IPAM server**.
3. On the Before you begin page, click **Next**.
4. On the Configure database page, click **Next**.
5. On the Select provisioning method page, in GPO name prefix, type **IPAM**, and then click **Next**.
6. Click **Apply**, and then when provisioning completes, click **Close**.
7. Open **Windows PowerShell**.
8. Type the following command, and then press ENTER.

   📌 This command will create the IPAM GPOs.

   ```
   ↪ Invoke-IpamGpoProvisioning –Domain contoso.com –GpoPrefixName IPAM –IpamServerFqdn
     server3.contoso.com –Force
   ```

9. In Server Manager, select **IPAM**, and then click **Configure Server Discovery**.
10. In Configure Server Discovery, click **Add**, and then click **OK**.
11. Click **Start server discovery**.

    ⚠ **IMPORTANT**: Server discovery will take a few minutes, possibly longer, to complete. You must wait for this to partially complete before proceeding. You can review the progress of server discovery by clicking the task notification icon next to Manage in the Server Manager Toolbar area.

12. In the navigation tree, click **Server Inventory**. When you see DC listed, it is safe to proceed to the next step.

    📌 Press F5 to refresh this view.

Configure IPAM managed servers

In this task, you will configure DC, Server1, and Server2 to be managed by IPAM. This will involve rebooting the DC, and forcing GPO refresh on all servers. These steps are examples of the steps mentioned in the exercise introduction to speed up the IPAM deployment process in a lab environment. They are not necessary in a production environment where you are able to wait for scheduled tasks to happen.

✎ Begin this task logged onto **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Inventory, click **Tasks**, and then click **Add Server**.

2. Type **Server1**, and then click **Verify**.
3. Check **DHCP Server**, and then set the Manageability status to **Managed**.
4. Click **OK**.
5. In Server Inventory, click **Tasks**, and then click **Add Server**.
6. Type **Server2**, and then click **Verify**.
7. Check **DHCP Server**, and then set the Manageability status to **Managed**.
8. Click **OK**.
9. Right-click **DC**, and then click **Edit Server**.
10. Set the Manageability status to **Managed**, and then click **OK**.
11. In Windows PowerShell, type the following command, and then press ENTER.

```
↪ ICM DC, Server1, Server2 {GPUpdate /Force}
```

12. In Server Manager, in Server Inventory, press the CTRL key, and then click all three servers.
13. Right-click the selected servers, and then click **Refresh Server Access Status**.
    - **IMPORTANT**: Wait for the task to complete before moving to the next step. To check the status of the task, on the Activity bar, click the More link.
14. In Server Inventory, press F5 to refresh the view.
    - All three servers will show a status of Managed, an access status of Unblocked, and a green icon if you completed this task correctly.
15. In Server Manager, in Server Inventory, press the CTRL key, and then click all three servers.
16. Right-click the selected servers, and then click **Retrieve All Server Data**.
    - **IMPORTANT**: Wait for the task to complete before moving to the next step. To check the status of the task, on the Activity bar, click the More link.

Exercise 2: Configure Administrative Roles for IPAM Operations

In this exercise, you will configure role-based administration to ensure that an administrator has permission to edit a particular DHCP scope but does not have the ability to modify other IPAM settings.

RBAC Concepts

Role

Role is a collection of IPAM operations. A role can be associated with a Windows user or group through an access policy (see below). The operations that a user can execute are determined by the role. IPAM provides several built-in user roles. Administrators can create new roles as per their business requirements. For example, an administrator can create a Block Admin role that would contain only edit and delete operations for IP address blocks.

Access scope

While a role determines what operations a user can execute, it does not tell what IPAM entities (IP address blocks, IP address range, DHCP server, or DHCP scopes) the user has access to. This is where access scopes come into play. Access scopes define administrative domains in IPAM. IPAM has a built-in access scope called Global. By default, all IPAM entities fall under the Global access scope; however an administrator can create more access scopes as child access scopes of Global based on business requirements. For example, based on geography, an administrator can create Europe and Asia as new access scopes under Global. The administrator can then select any IPAM entity (or group of entities using multi-select) and assign Global\Asia or Global\Europe access scopes to them.

A user or group is associated with an access scope through an access policy (see below).

Access policy

An access policy brings a role and an access scope together and associates these with a Windows user or group. Through an access policy an administrator can pick a Windows user or group and specify a role for the user as well as an access scope in which the role is applicable, effectively specifying what operation a user is allowed to execute and on what IPAM entities these operations can be executed.

For example, an administrator can define an access policy for User1 with a role of Block Admin and an access scope of Global\Asia. User1 will be able to edit and delete IP address blocks that lie under the Global\Asia access scope. User1 will not be able to edit or delete the IP address blocks that lie under the Global\Europe access scope nor will User1 be able to execute any operation other than edit or delete of IP address blocks.

Create a new DHCP scope

In this task, you will create a new DHCP scope name Singapore Lab DHCP Scope.

✎ Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, in the left navigation pane, click **IPAM**.
2. Under Monitor and Manage, click **DNS and DHCP Servers**.
3. In the contents pane, ensure **DHCP** is selected in the Server Type field.
4. In the View field, select **Server Properties**.
   - 📌 This will cause a list of the 3 DHCP servers in the lab environment to appear.
5. Right-click **DC.contoso.com**, and then click **Create DHCP Scope**.
6. On the General Properties page, enter the following information in the form:

Property	Value
Scope name	**Singapore Lab Scope**
Start IP address	**40.40.1.0**
End IP address	**40.40.1.100**
Subnet Mask	**255.0.0.0**

7. Scroll to the bottom of the form, and then click **OK**.

Create a new role-based user role for administration of the new scope

In this task, you will create a new role that contains only the Edit DHCP Scope operation user role. ✎

Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, in the left navigation pane, select IPAM, and then click **Access Control**.
   - 📌 You may have to scroll down to see this item.
2. Below Access Control, right-click **Role**, and then click **Add User Role**.
3. In the Add or Edit Role dialog box, in Name, type **DHCPScopeEditor**.
4. In the Operations area, expand **DHCP Scope Operations**, and then select **Edit DHCP scope**.
5. Click **OK**.
6. In the left navigation pane, right-click **Access Scopes**, and then click **Add Access Scope**.
7. In the Add Access Scope dialog box, click **New**.
8. In the Add Access Scope dialog box, in Name, type **SingaporeLab**.
9. Click **Add**, and then click **OK**.
10. In the left navigation pane, click **DHCP Scopes**.

11. In the Current view field, select **Scope Properties**.
12. In Filter, type **Singapore lab**.
    - The display is filtered to show only the Singapore Lab Scope.
13. Right-click **Singapore Lab Scope**, and then click **Set Access Scope**.
14. In the Set Access Scope dialog box, clear the **Inherit access scope from parent** check box.
    - **CAUTION**: Ensure you clear the check box.
15. Select **Singapore Lab**, and then click **OK**.
16. In the left navigation pane, click **Access Control**.
17. Right-click **Access Policies**, and then click **Add Access Policy**.
18. In the Add Access Policy dialog box, click **Add**.
19. In the Select User, Computer, or Group dialog box, click **Locations**.
20. In the Locations dialog box, click **Entire Directory**, and then click **OK**.
21. In the Enter the object name to select field, type **BenSmith**, and then click **OK**.
22. In the Add Access Policy dialog box, on the left-hand side, click **Access Settings**.
23. Under Access Settings, click **New**.
24. In the New Setting tile that appears below, in Select role, select **DHCPScopeEditor**.
25. In the Select the access scope for the role, click **SingaporeLab**.
26. Click **Add Setting**.
    - **IMPORTANT**: Ensure you add the setting in the above step. You may have to scroll down to see the button.
27. Click **OK**.

Exercise 3: Configure DHCP Policy-Based Assignment using IPAM

DHCP Policy-Based Assignment (PBA) is a powerful feature for IPV4 networks in DHCP server that gives you greater control over your network and the devices accessing it. It allows you to use tools known as DHCP policies or simply policies to identify and group together these devices based on attributes like MAC Address, Vendor Class, and User Class. You can then control the leases (IP addresses) and DHCP options (configuration information) that are assigned to these devices or clients. For example, you can use DHCP policies to match the MAC address of clients and make sure that all virtual machines (VMs) accessing your network are assigned addresses from a specific IP range or are assigned some specific DHCP options.

DHCP policies can be configured at the server level or scope (subnet) level. Until now, policies were accessible only for individual DHCP servers via the management interfaces for the DHCP server role. With the IPAM feature in Windows Server 2012 R2, you can create and manage policies centrally across multiple DHCP servers. You can create a policy for multiple servers or scopes in a single operation. You can also copy policies for one server or scope to another.

In this exercise, you will see how IPAM allows you to centrally manage policies in an easy and simple manner.

Configure new DHCP policies and import existing DHCP policies

You can create new policies on a DHCP server or scope. You can also copy existing policies on a DHCP server or scope by importing them from some other server or scope. These operations are available in the management view of DHCP servers and scopes. In this task, you will create new DHCP policies as well as importing them from another server.

- Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, in the left navigation pane, click **IPAM**.
2. Under Monitor and Manage, click **DNS and DHCP Servers**.
3. In the contents pane, in the Server Type field, ensure **DHCP** is selected.
4. In the View field, select **Server Properties**.
   - In the details pane at the bottom, you can view Policies by clicking on the Policies tab.
5. Right-click **Server1.contoso.com**.
   - Configure Policy can be used for configuring a new policy on the selected server(s). The Import Policy can be used for copying an existing policy from a different server or scope. The operation Activate Policies can be used to turn ON application of all policies configured the selected server. The operation Deactivate Policies can be used to turn OFF application of all policies configured on the selected server.
6. Click **Configure DHCP Policy**.

7. On the Configure Server policy page, in Name, type **Test Policy**.
8. In the left pane, click **Conditions** if it is not already selected.
9. Under Policy Conditions, click **New**.
10. In the New Condition tile, ensure **Vendor-Class** is selected as the Criteria, **Equals** as the Operator, and **Microsoft Options** as the Value, and then click **Add**.
11. Click **Add Condition**.
12. In the left pane, select **DNS Updates** if it is not already selected.
    - Notice the configuration options that are available for DNS Dynamic Updates. These include the DNS registration settings and DHCP options that you want to apply to the clients that match this policy.
13. Click **OK**.
    - Now that you have configured a policy, you can import this policy to another server.
14. In Server Manager, right-click **Server2.contoso.com**, and then click **Import DHCP Policy**.
    - The Import Policy dialog box provides you with the option to import policies at either the server or the scope level. You can configure or import policies for scopes from the management view for DHCP scopes. For scope policies, you can specify IP address ranges from which the leases will be allocated to the clients.
15. In the Import Policy dialog box, in Select Server, select **Server1.contoso.com**.
16. In Select Policy, select **Test Policy**, and then click **OK**.
17. In the details pane, click the **Policies** tab.
    - Here you can see the various policies that are assigned to each DHCP server. If you do not see the policy you just added, refresh the view.

Exercise 4: Automating IP Address Lifecycle Management

IPAM's new Windows PowerShell interface makes it painless to automate IP address lifecycle management. It provides a rich set of cmdlets that allow you to perform all management functions for your IP addresses, ranges, and blocks. You can leverage these to write scripts to integrate IPAM with various orchestrators to provision network properties for various physical and virtual devices and servers. This saves time, eliminates manual intervention, and reduces operating cost. In this exercise, you will learn how to use some of the IPAM PowerShell cmdlets to introduce more automation into your environment.

> **IMPORTANT**: The IPAM PowerShell cmdlets that are used in this exercise may be found in C:\LabFiles\IPAM-Automation.ps1 on ADMIN. Some of the commands are lengthy and complex. As such, you may prefer to copy this file to Server3 and then copy and paste the commands from the text file to the IPAM PowerShell console, or run them directly from Windows PowerShell ISE.

Managing IP address ranges using IPAM PowerShell cmdlets

In this task, you will learn how to use the Get-IpamRange cmdlet to perform a variety of administrative actions.

Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. On the taskbar, click the **Windows PowerShell** icon to open the Windows PowerShell console.
2. In the Windows PowerShell console, type the following command, and then press ENTER.

```
Get-IpamRange –AddressFamily IPv4 –AddressCategory Private
```

   This cmdlet lists all IPv4 private address ranges in IPAM. By default, the output shows you the **NetworkId**, **StartIp** address and **EndIp** address of the range, the service which is managing this range, and whether a given IP range is overlapping with another range. You can use the **Format-List** cmdlet to display more details for any given range.

3. In the Windows PowerShell console, type the following two commands, pressing ENTER after each one.

```
$a=Get-IpamRange –AddressFamily IPv4 –AddressCategory Private
$a[0]|fl *
```

   The above cmdlets provide more detail of each IP range.

4. In the Windows PowerShell console, type the following command, and then press ENTER.

```
Get-IpamRange –AddressFamily IPv4 –AddressCategory Private|where-object {$_.PercentageUtilized –gt 10}
```

- The above cmdlet lists all IPv4 private address ranges with utilization greater than 10%. As shown in the next cmdlet, you can use the **Export-Csv** Windows PowerShell cmdlet to generate a comma-separated file with all the ranges with a utilization greater than 10%.
- In a production environment, you would likely be more interested in utilization percentages that are greater than some value closer to 100% or percentages that are less than some value closer to 0%.

5. In the Windows PowerShell console, type the following two commands, and then press ENTER.

```
↪ Get-IpamRange –AddressFamily IPv4 –AddressCategory Private|where-object
  {$_.PercentageUtilized –gt 10}|Export-Csv –Path
  "C:\Users\Administrator.Contoso\Desktop\OverUtilizedRanges.csv" –NoTypeInformation
  –Force
↪ notepad "C:\Users\Administrator.contoso\Desktop\OverUtilizedRanges.csv"
```

- This will open a Notepad file containing all the over-utilized IPv4 private address ranges. Similarly, you can generate reports about conflicting IP address ranges as shown in next example.

6. In the Windows PowerShell console, type the following command, and then press ENTER.

```
↪ Get-IpamRange –AddressFamily IPv4 –AddressCategory Private|where-object {$_.Overlapping
  –eq "True"}
```

- You may not see any output from this command. This is expected in this environment, as there are no overlapping scopes.
- The next series of commands show how to find a free IP address, and then assign it to a device.

7. In the Windows PowerShell console, type the following two commands, pressing ENTER after each one.

```
↪ $range = Get-IpamRange –StartIPAddress 10.0.0.10 -EndIPAddress 10.0.0.20
↪ Find-IpamFreeAddress –InputObject $range -TestReachability
```

- This cmdlet will output an unutilized IP address from the IP address range with a start IP address of 10.20.1.100 and an end IP address of 10.20.1.200. After you have verified address availability, you can assign the address to a device. To do this, you will first add this IP address to IPAM, and then create a corresponding reservation for this IP.

8. In the Windows PowerShell console, type the following four commands, pressing ENTER after each one.

```
↪ $range = Get-IpamRange –StartIPAddress 10.0.0.10 -EndIPAddress 10.0.0.20

↪ $freeip = Find-IpamFreeAddress –InputObject $range –TestReachability

↪ $ip = Add-IpamAddress –IpAddress $freeIp.Address –ManagedByService
  $range.ManagedByService –ServiceInstance $range.ServiceInstance –DeviceType Printer
```

```
–IpAddressState In-Use –AssignmentType Dynamic –MacAddress "AA-BB-CC-DD-EE-
FF" –ReservationServer $range.DhcpServerName –ReservationName "B3_F1_Printer_HP"
–ReservationType Both –ReservationDescription "Reservation for printer on first floor
of building 3" –ClientID "B3F1" –PassThru

↪ $ip|fl *
```

📌 This cmdlet will add an IP address to the IPAM system. You can observe in the output of the last cmdlet that the **ReservationScopeName** and **ReservationScopeId** fields have been added automatically. You will still need to create the reservation on DHCP server using the cmdlet in the following step.

9. In the Windows PowerShell console, type the following command, and then press ENTER.

```
↪ Add-DhcpServerv4Reservation –ComputerName $ip.ReservationServer –IPAddress
$ip.IPAddress –ClientId $ip.MacAddress –ScopeId $ip.ReservationScopeID –Name
$ip.ReservationName –Description $ip.ReservationDescription -PassThru
```

📌 Now that you have provisioned a device, you will deprovision it and reclaim the IP address. As a part of this process, you will remove the reservation from the DHCP server, and then delete the IP address from the IPAM server.

10. In the Windows PowerShell console, type the following command, and then press ENTER.

```
↪ Remove-DhcpServerv4Reservation –ComputerName $ip.ReservationServer –
IPAddress $ip.IPAddress -PassThru
```

📌 This command removes the reservation. The command in the following step deletes the IP address from the IPAM system.

11. In the Windows PowerShell console, type the following command, and then press ENTER.

```
↪ Remove-IpamAddress –InputObject $ip -Force
```

Exercise 5: Administering DHCP Failover using IPAM

In this exercise, you will learn how to administer DHCP failover using IPAM. Tasks that you will perform in this exercise include creating, viewing, and editing a failover relationship, replicating scopes, removing scopes, and other related administrative tasks.

Create, view, and edit a failover relationship

In this task, you will create, view, and edit a failover relationship.

- Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, in the left navigation pane, click **IPAM**.
2. Under Monitor and Manage, click **DHCP Scopes**.
3. In the contents pane, in Current view, select **Scope Properties**.
   - You may have to clear Singapore Lab from the filter to view all the available scopes.
4. Click the **Scope ID** column to order the list by scope ID.
5. Press the CTRL key, and then select both the Scope IDs **192.168.10.0** and **192.168.11.0**.
   - Both of these scopes are configured on **DC.contoso.com**.
6. Right-click the two selected scopes, and then click **Configure DHCP Failover**.
7. On the Configure Failover Relationship page, ensure **Create new relationship** is selected.
8. For Partner server, select **Server1.contoso.com**.
9. In Relationship name, type **Server1-DC**.
10. In Secret, type **secret**.
11. Click **Apply**.
    - After clicking Apply, the property sheet will automatically shift to the Summary pane. In this pane, you can see the result of the Configure Failover operation.
12. Click **OK**.
    - The IPAM management list (ML) will refresh to display the relationship details for the scopes.
13. In the contents pane, scroll to the right to view the relationship details.
14. In the left navigation pane, click **DNS and DHCP Servers**.
15. Ensure that Server Type is set to **DHCP**, and then in View, select **Failover Relationships**.
16. Right-click **Server1-DC**, and then click **Edit DHCP Failover Relationship**.
17. Under Advanced Properties, change the Percentage of Server to **60**.
18. Click **Apply**, and then click **OK**.

Replicate configuration of failover scopes to partner servers

In this task, you will learn how to replicate DHCP failover server settings.

✎ Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, in the left navigation pane, click **IPAM**.
2. Under Monitor and Manage, click **DNS and DHCP Servers**.
3. Ensure that Server Type is set to **DHCP**, and then in View, select **Server Properties**.
4. Right-click **DC.contoso.com**, and then click **Replicate DHCP Server**.
5. In the Replicate Server dialog box, click **OK**.
   - 📌 A dialog box appears showing the status of the replication.
6. When the replication has finished, click **Close**.

Replicate multiple scopes of a server

In this task, you will learn how to replicate multiple server scopes.

✎ Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, in the left navigation pane, click **IPAM**.
2. Under Monitor and Manage, click **DHCP Scopes**.
3. In the contents pane, in Current view, select **Scope Properties**.
4. Right-click the **Scope ID** column heading, point to **Group by**, and then click **Server Name**.
5. Press the CTRL key, and then under DC.contoso.com, select both of the Scope IDs **192.168.10.0** and **192.168.11.0**.
6. Right-click the selected scopes, and then click **Replicate DHCP Scope**.
7. Click **OK**.
8. In the Replicate Scopes dialog box, click **OK**.
   - 📌 A dialog box appears showing the status of the replication.
9. When the replication has finished, click **Close**.

Remove scopes from a failover relationship and delete the failover relationship

In this task, you will learn how to remove scopes from a failover relationship and how to delete the failover relationship.

✎ Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, select **IPAM**, select **Monitor and Manage**, and then click **DHCP Scopes**.
2. In the contents pane, in Current view, select **Scope Properties**.
3. Select the **CorpNet** and **CorpNet2** scopes.
4. Right-click the two selected scopes, and then click **Remove DHCP Failover Configuration**.
5. In the Remove Failover Configuration dialog box, click **OK**.
6. When the task shows as completed, click **Close**.

Exercise 6: Managing DHCP MAC Address Filters using IPAM

MAC address-based filtering or link layer-based filtering for DHCP enables administrators to control network access based on media access control (MAC) address, providing a low-level security method. You can create MAC address-based filters to specify which MAC addresses are allowed on the network and which are denied access.

A DHCP server maintains allow and deny lists of MAC addresses. If you add MAC addresses to the allow list and then enable the list, only these MAC addresses will be granted an IP address by the DHCP server. If you add MAC addresses to the deny list and then enable the list, these MAC addresses will be denied service by the DHCP server. You can enable both allow and deny lists, in which case the deny list takes precedence. This means that the DHCP server provides DHCP services only to clients whose MAC addresses are in the allow list, provided that no corresponding matches are in the deny list.

You can use wildcards to allow or deny network access based on vendor MAC prefixes.

Link layer filtering is currently available for IPv4 address only.

In this exercise, you will see how IPAM allows you to centrally manage MAC address filters in an easy and simple manner.

Add DHCP MAC address filters

In this task, you will add MAC address filters to allow or deny lists on one or more DHCP servers.

- Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, select IPAM, select Monitor and Manage, and then click **DNS and DHCP Servers**.
2. Ensure that Server Type is set to **DHCP**, and then in View, select **Server Properties**.
3. Select one of the DHCP servers, and then review the information in the details pane.
   - The Allow MAC Address Filters and the Deny MAC Address Filters are disabled.
4. Right-click **DC.contoso.com**, and then click **Edit DHCP Server Properties**.
   - You could also select multiple DHCP servers.
5. In Edit DHCP Server Properties, in the left pane, click **MAC Address Filters**.
   - Under MAC Address Filters, you will find the check boxes to enable allow or deny lists.
6. Select the check boxes to enable the **allow** and **deny** lists.
7. Click **OK**.
8. Right-click **DC.contoso.com**, and then click **Add DHCP MAC Address Filter**.
   - You could also select multiple DHCP servers.
9. In the Add MAC Address Filter dialog box, click **Deny**.

10. In MAC Address, type **00155D***.
    - Wildcards can be used to filter MAC addresses based on a pattern, such as a vendor ID in the MAC address. Wildcards can only be used if there is an even number of characters.
11. In Description, type **MAC Address Filter for Hyper-V Virtual Machines**.
12. Click **Add MAC Address Filter**.
13. In the Add MAC Address Filter dialog box, click **OK**.
14. In Server Manager, click **View**, and then select **Filters**.
    - Here you can manage filters from DNS and DHCP servers.
15. Right-click the filter created in the previous steps.
    - You can delete the filter, edit the filter, or change it from a deny filter to an allow filter or vice versa depending on its state.
16. Click **Delete**.

Exercise 7: Managing DHCP using IPAM

Consider a situation in which the available address pool for a currently active scope is nearly depleted, and still more computers are expected to be added to the network. In this situation, you can use superscopes that allow a DHCP server to provide leases from more than one scope to clients on a single physical network. The scopes in the same superscope can share IP addresses and give leases to clients on each other's subnet. You can add the depleted scope, along with another scope, to a superscope.

Superscopes can also help you resolve other deployment issues such as migrating clients over time to a new scope, for example, to renumber the current IP network from an address range used in an existing active scope to a new IP network range of addresses used in a new scope.

At present, superscopes are available for IPv4 addresses only.

Add and manage DHCP superscopes

In this task, you will see how you can create and manage DHCP superscopes using IPAM.

- Begin this task logged on to **Server3** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, under Monitor and Manage, click **DNS and DHCP Servers**.
2. Ensure that Server Type is set to **DHCP**, and then in View, select **Scope Properties**.
   - You are now in the management view for DHCP scopes. Alternatively, you can select DHCP Scopes in the navigation tab, and then in Current view, select Scope Properties.
3. Scroll to the right to see the Superscope name column.
   - None of the scopes is a member of a superscope. Consequently, the entries in the column are blank.
4. In Filter, type **192.168**.
   - This causes all the scopes starting with 192.168 to be filtered in the display.
5. Right-click the **192.168.10.0** scope, and then click **Add to DHCP Superscope**.
6. In the Add to Superscope dialog box, in Superscope name, type **Dublin Superscope**.
7. Click **OK**.
8. Right-click the **192.168.11.0** scope, and then click **Add to DHCP Superscope**.
9. In the Add to Superscope dialog box, click **Use existing superscope**, and then select **Dublin Superscope**.
10. Click **OK**.
    - You have now created a superscope and added two scopes to it. Each scope can now service each other's clients.
11. In View, select **Superscope Properties**.

12. In the details pane, click the **Superscope Properties** and the **DHCP Scopes** tabs, and then review the information that is presented.
13. In the upper pane, right-click **Dublin Superscope**, note the options presented in the menu, and then click **Delete**.

    The operations you can perform on superscopes include:

    Rename Superscope: Rename a superscope.

    Create DHCP Scope: Create a new DHCP scope to add to this superscope. This is particularly useful when you feel that utilization of the superscope is high and you need another scope to cater to the clients of the member scopes of that superscope.

    Configure Failover: Configure a failover relationship for the member scopes of the superscope. See the section on managing a failover relationship.

    Remove Failover Configuration: Remove the member scopes of the superscope from failover relationships. See the section on managing a failover relationship.

    Set Access Scope: Restrict the permissions on the superscope to only certain users. See the section on restricting user permissions for editing scopes.

    Activate DHCP Superscope: Activate all the member scopes of the superscope.

    Deactivate DHCP Superscope: Deactivate all the member scopes of the superscope.

    Delete: Delete the superscope. The member scopes are not deleted but are simply removed from the superscope.

This is the end of the lab

LAB 10. Internet Information Services (IIS 8, Web Server) in Windows Server 2012

Microsoft has provide Internet Information Services (IIS) for more than 15 years. IIS 8 has been released and installed in Windows Server 2012. IIS 8.0 is far more scalable, more appropriate for cloud and virtual systems, and more integral to Microsoft's application and programming environment.

The new features of IIS 8 included Centralized SSL Certificate Support: SSL Scalability and Manageability, CPU Throttling: Sand-boxing Sites and Applications, Application Initialization, WebSocket Protocol Support, FTP Logon Attempt Restrictions, Dynamic IP Address Restrictions, Using ASP.NET 3.5 and ASP.NET 4.5, and Multicore Scaling on NUMA Hardware.

10.1 Installing Web Server (IIS 8) in Windows Server 2012

Open **Server Manager**, and under Manage menu, select **Add Roles and Features**

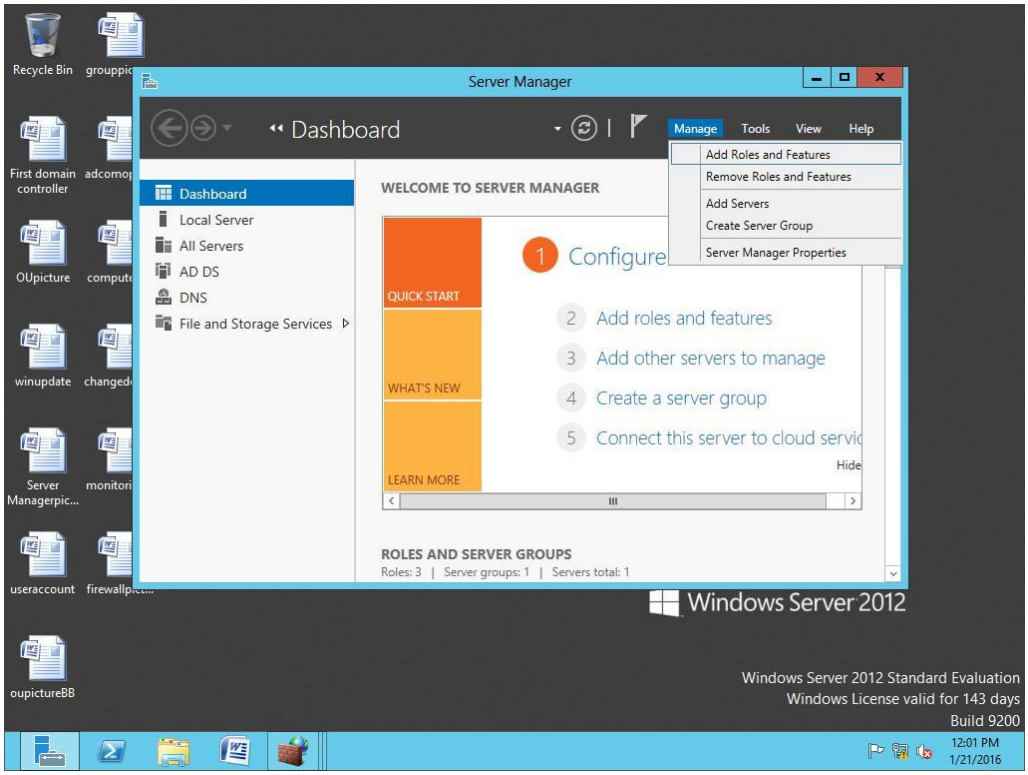


Fig. 1 Add Roles and Features

83

Click **Add Roles and Features**

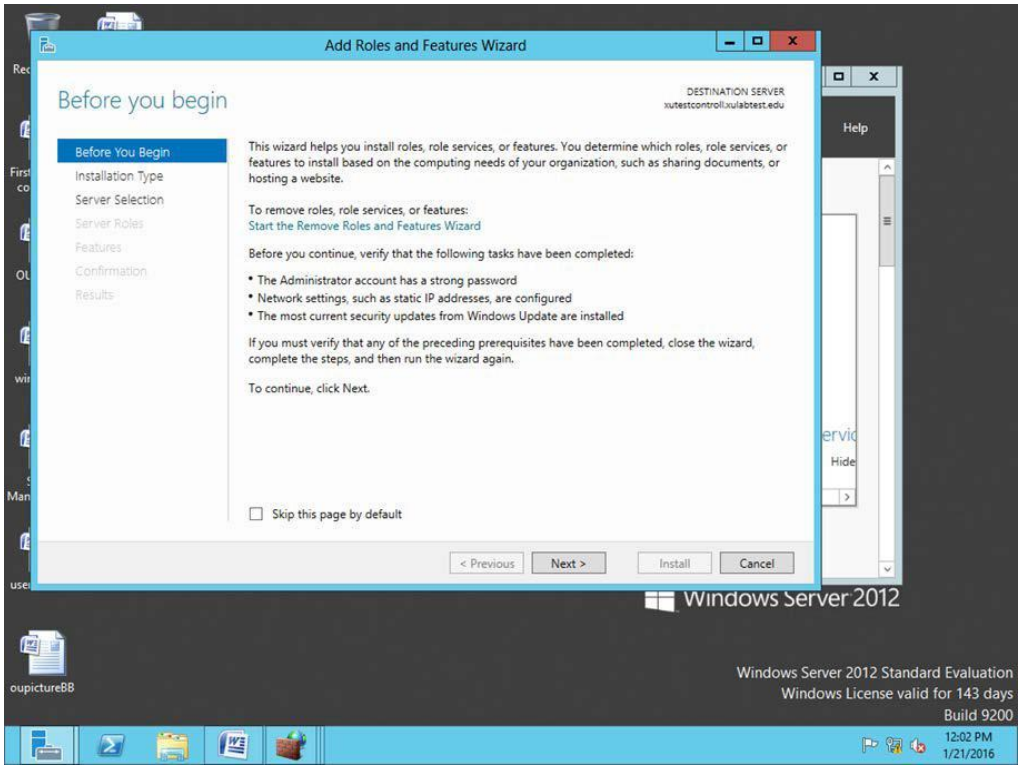


Fig.2 Before installation of Web Server

Click **Next,** Select **Role-based or Feature-based Installation**

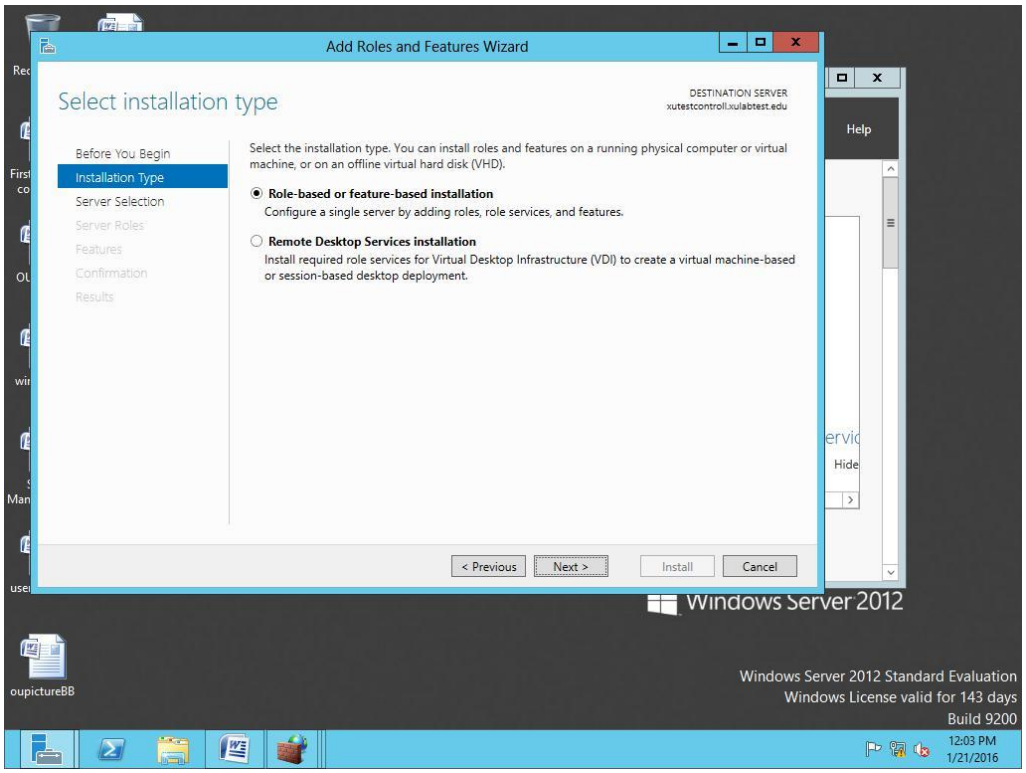


Fig.3 Select installation type

Select Server that you want to install Web Server

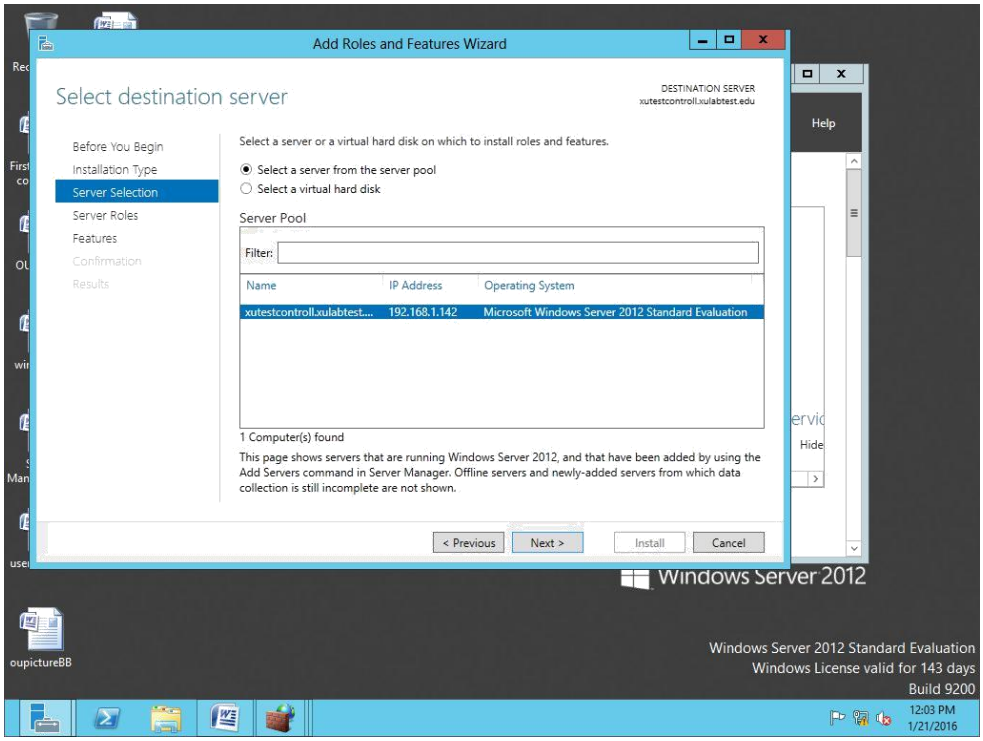


Fig. 4 Select Server to install Web Server

Select **Web Server (IIS),** Add features that are required to Web Server (IIS)

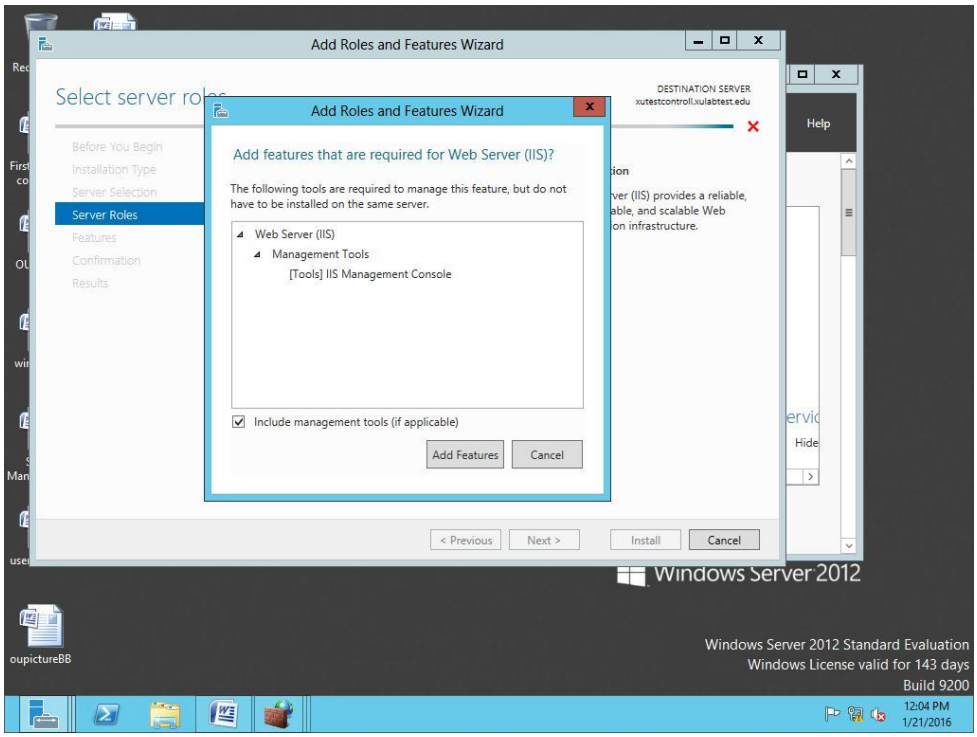


Fig. 5 Add features that are required to Web Server (IIS)

Click **Add Features**

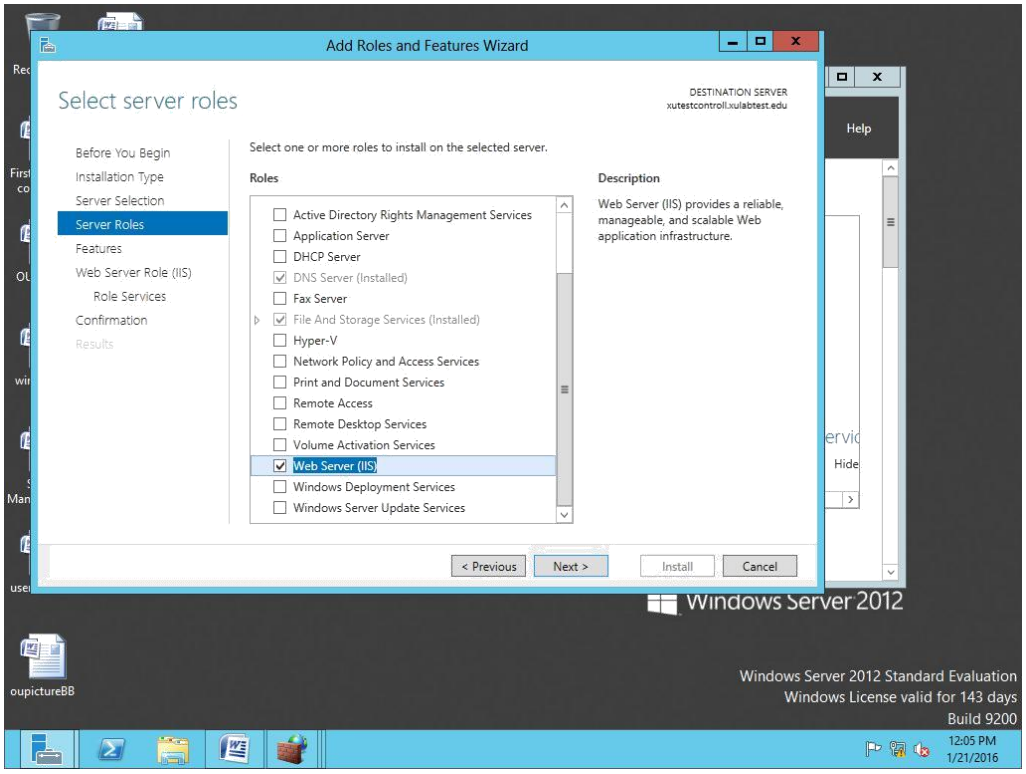


Fig. 6 Server Roles: Web Server (IIS)

Click Next, Select Features, **ASP.NET 4.5**

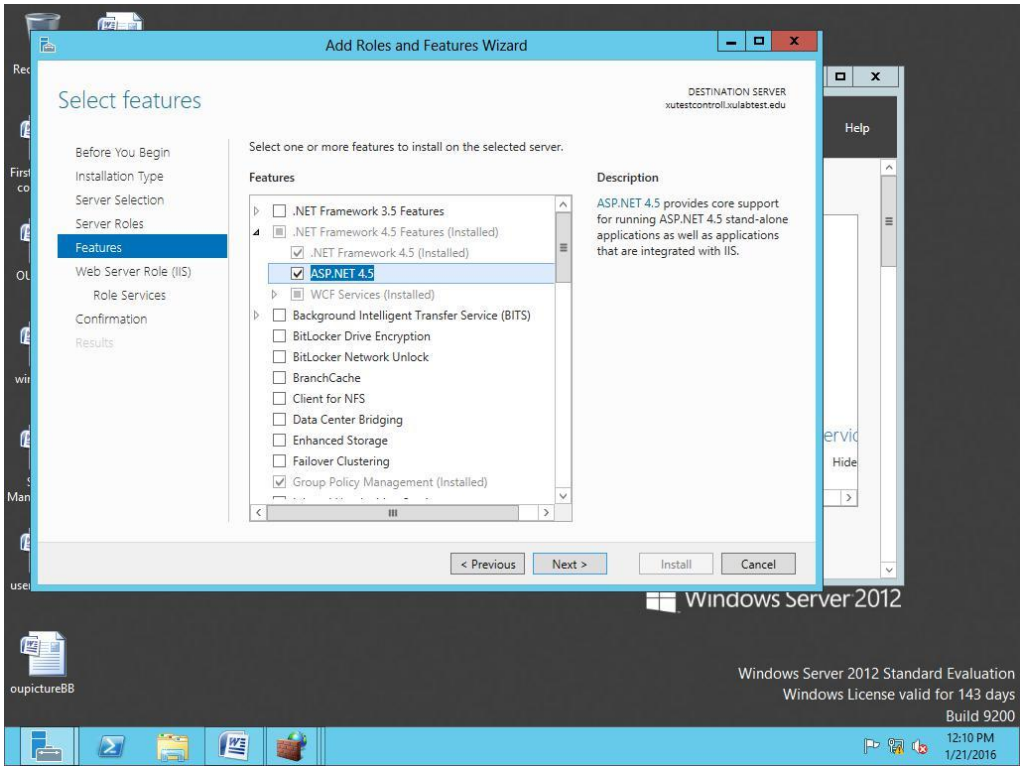


Fig. 7 Add ASP.NET 4.5 Feature

Click **Next**. Web Server Role (IIS)

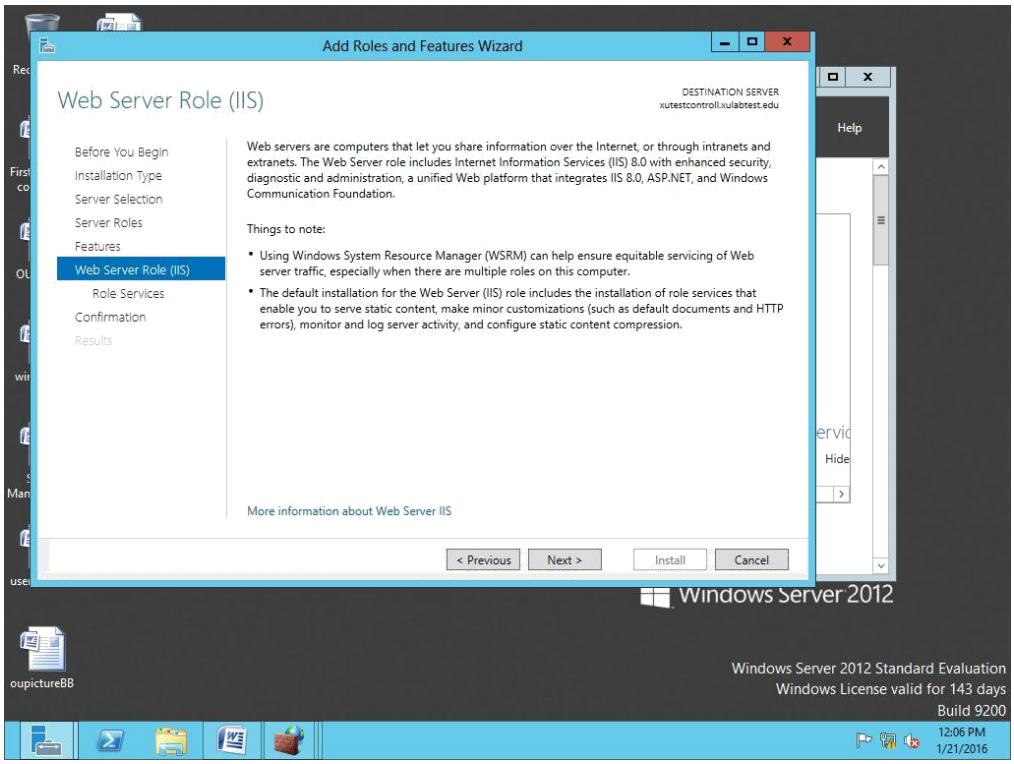


Fig.8 Web Server Role (IIS)

Click **Next**, and select role services

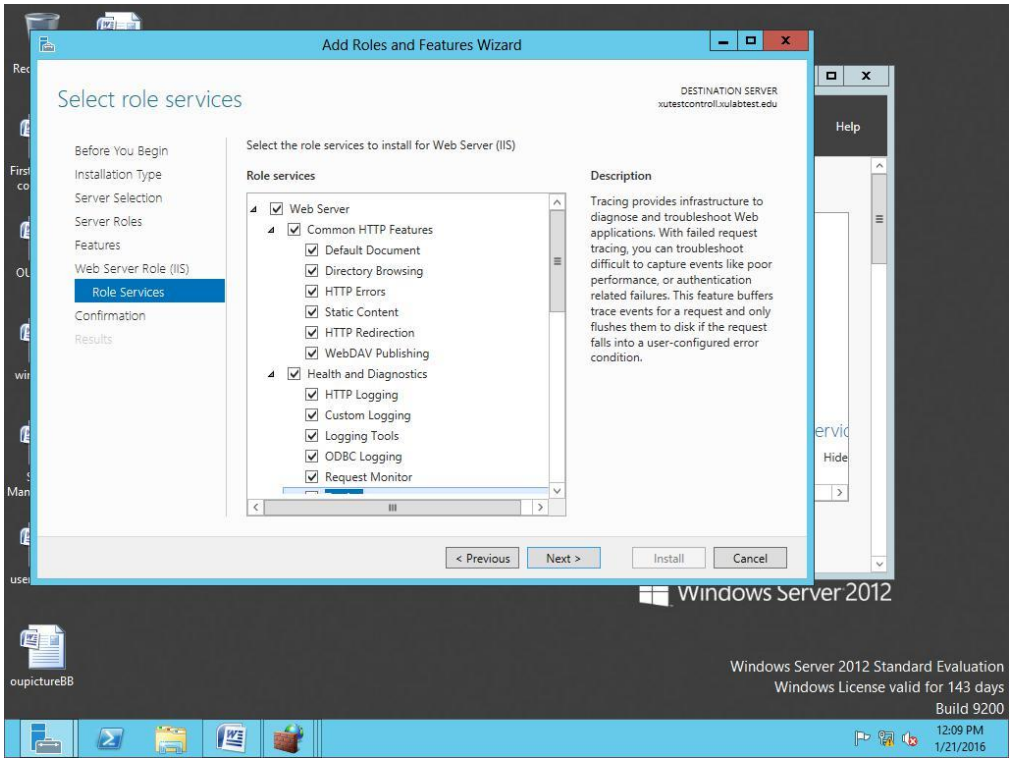


Fig. 9 select role services

Click Next, and confirmation

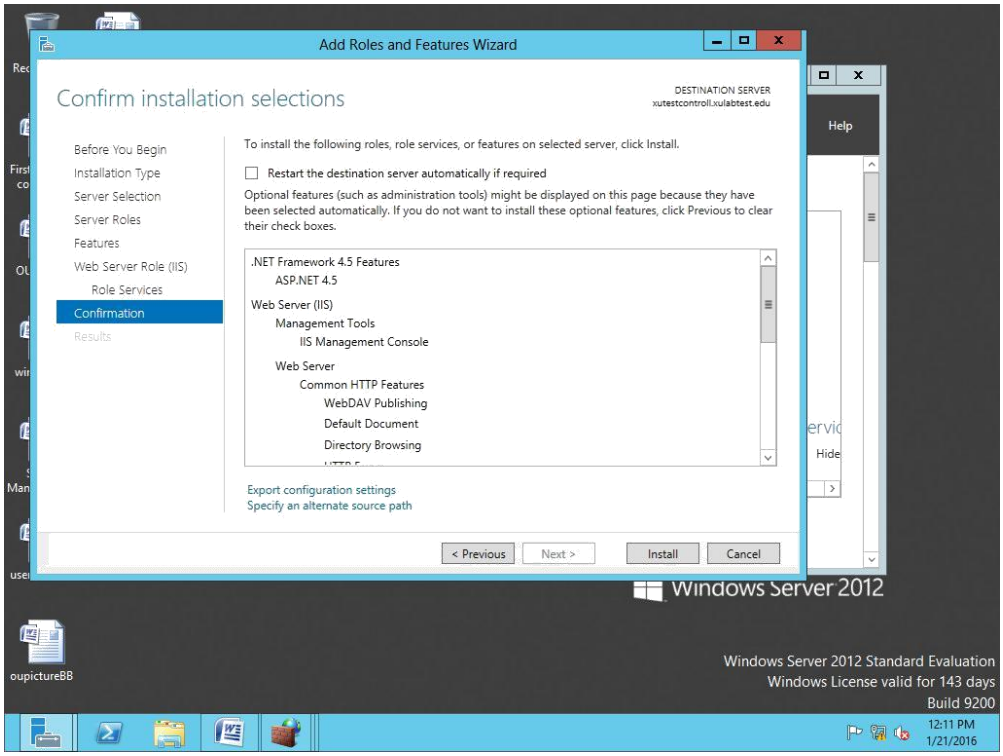


Fig. 10 Confirmation for Web Server installation

Click **Install**, begin to install Web Server (IIS)

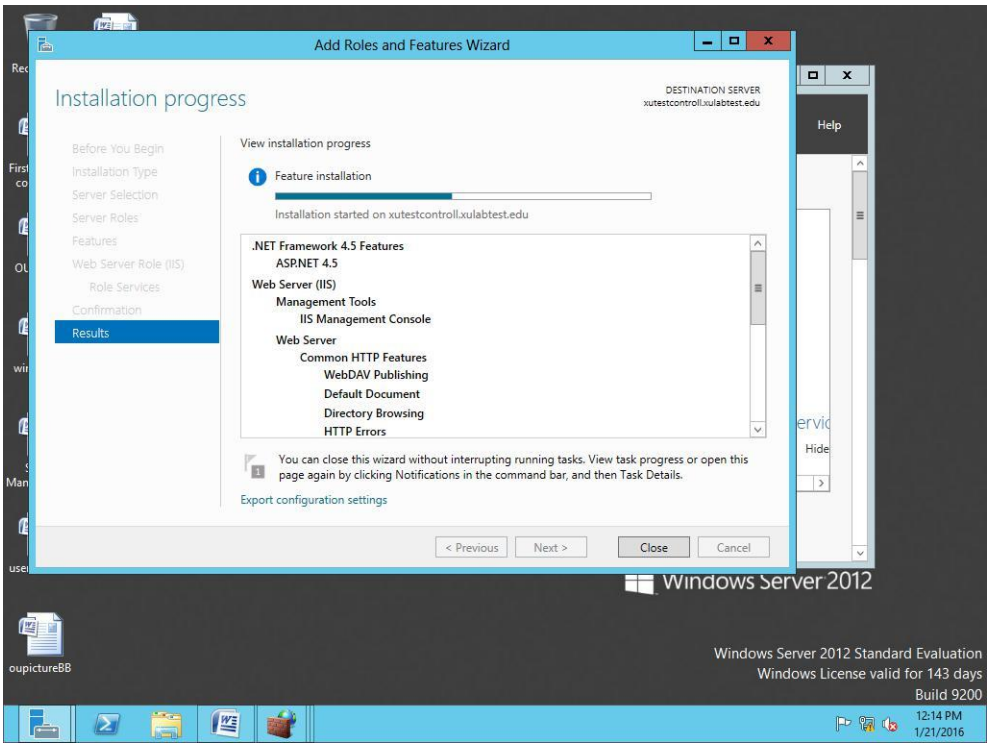


Fig.11 Installation progress

When the IIS installation completes, the wizard reflects the installation status:

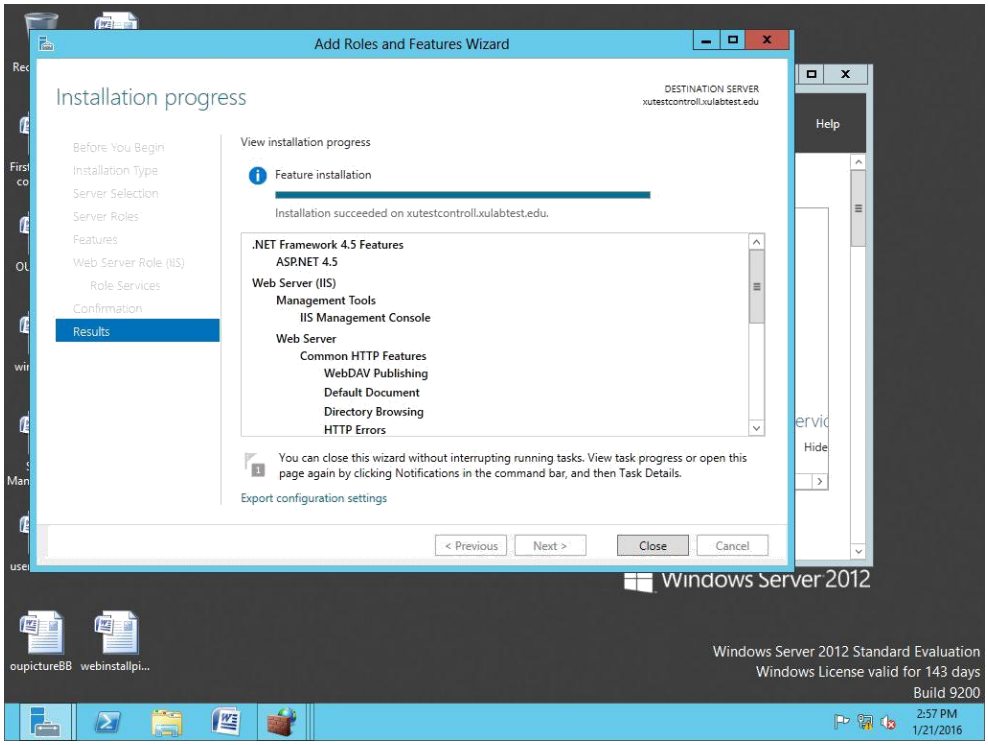


Fig. 12 IIS installation completes

Click **Close** to exit the wizard.

You are successful to install Web Server (IIS8) in the Windows Server 2012.

10.2 Testing Web Server (IIS 8)

After you are successful to install IIS 8, you can try to test the Web Server IIS 8. You type http://localhost/ in the Internet Explorer, you can see the webpage as follows:

Fig. 13 Internet Information Services (IIS 8)

10.3 Internet Information Services (IIS) Manager

Internet Information Services (IIS) Manager is a useful tool for Web Server Management.

10.3.1 Accessing Internet Information Services (IIS 8) Manager

- On the Start screen, click **Control Panel**. Click **System and Security**, and then click **Administrative Tools**. In the **Administrative Tools** window, double-click **Internet Information Services (IIS) Manager**.

- Another way to access Internet Information Services (IIS 8) Manager from Server Manager.

Click **Tools** in Dashboard of Server Manager, Click **Internet Information Services (IIS) Manager** in Pop down menu.

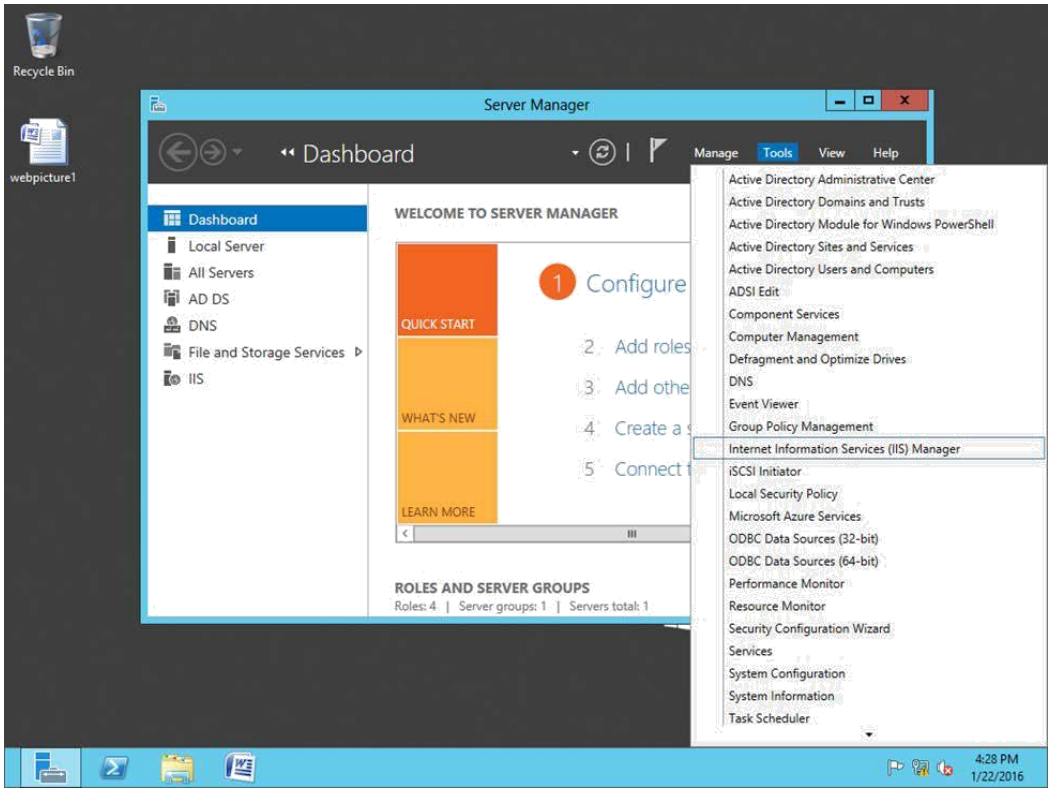


Fig. 14 Access Internet Information Services (IIS 8) Manager from Server Manager

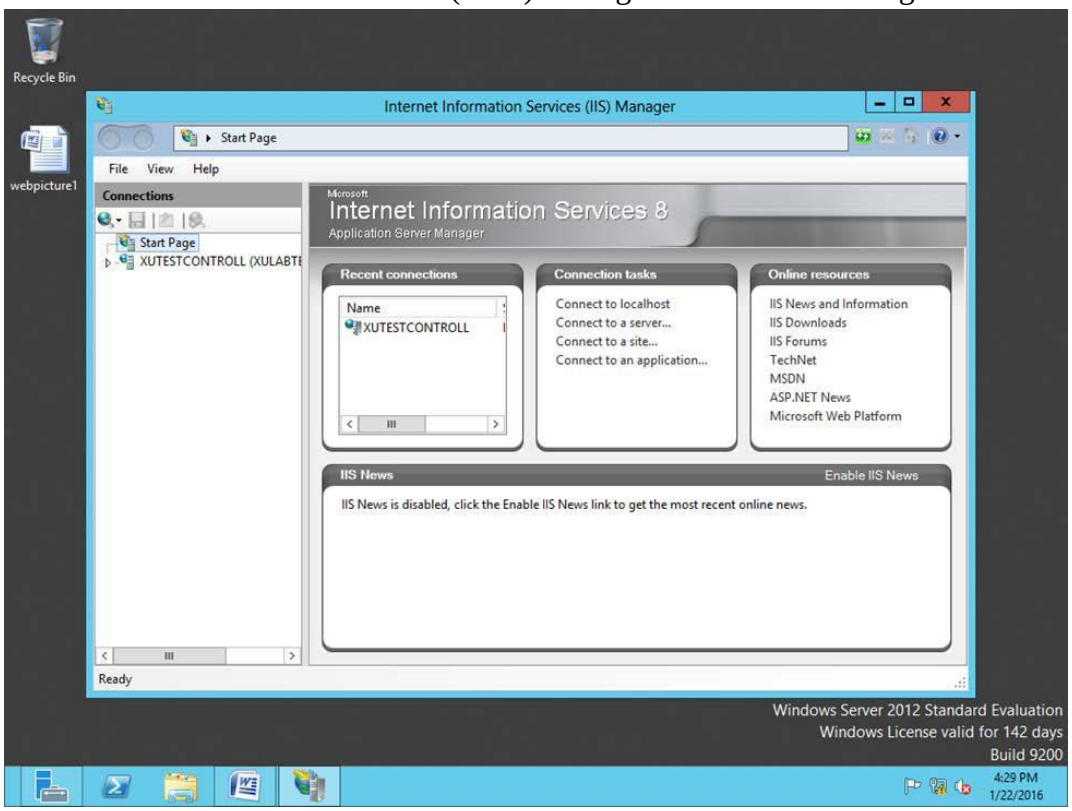


Fig. 15 Internet Information Services (IIS) Manager Administration Interface

10.3.2 Introduction to Internet Information Services (IIS) Manager Administration Interface

The Introduction to Internet Information Services (IIS) Manager included Connection, Connection tasks and Online resources.

In Connection section, you can see Start Page and Web Server.

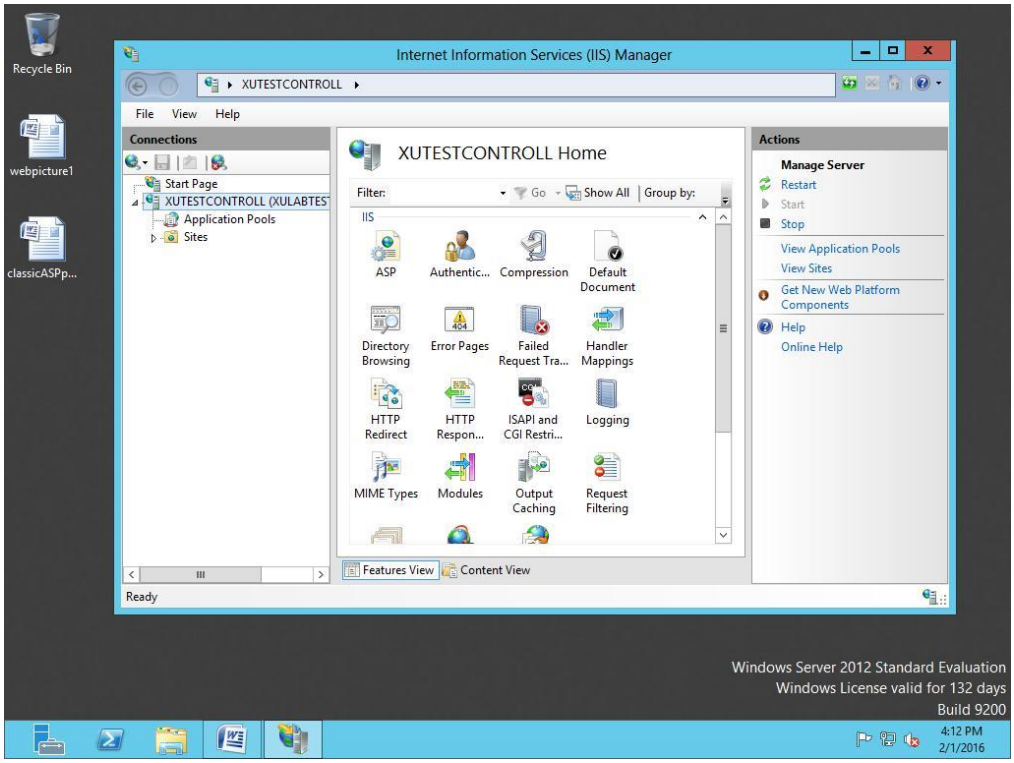


Fig. 16 Web Server Home

Click Web Server, you can see Server Home that included Features View and Content View. In Action window, you can manage server, and view Application Pools and Sites.

10.3.3 File, View and Help menu in IIS 8 Manager

Click **File**, you can see Connect to a Server, Site and Application menu.

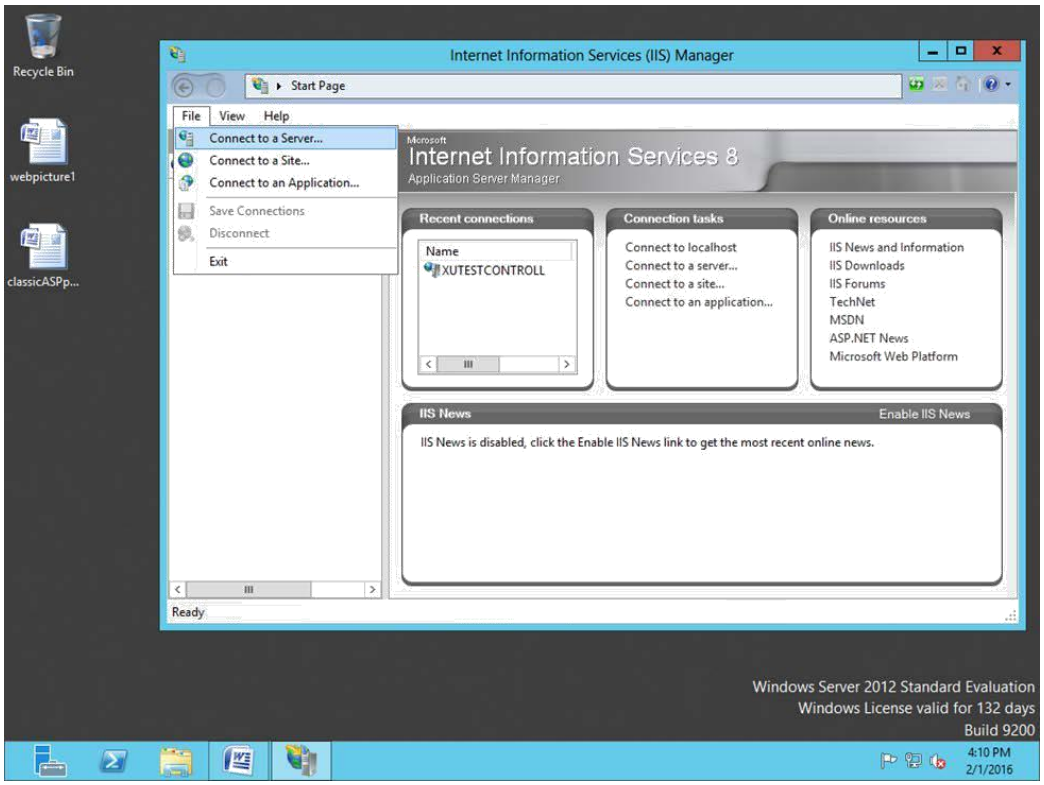


Fig. 17 File Pop down menu

Click **View**, you can view Web Server, Application Pools and Site by Group and Sort.

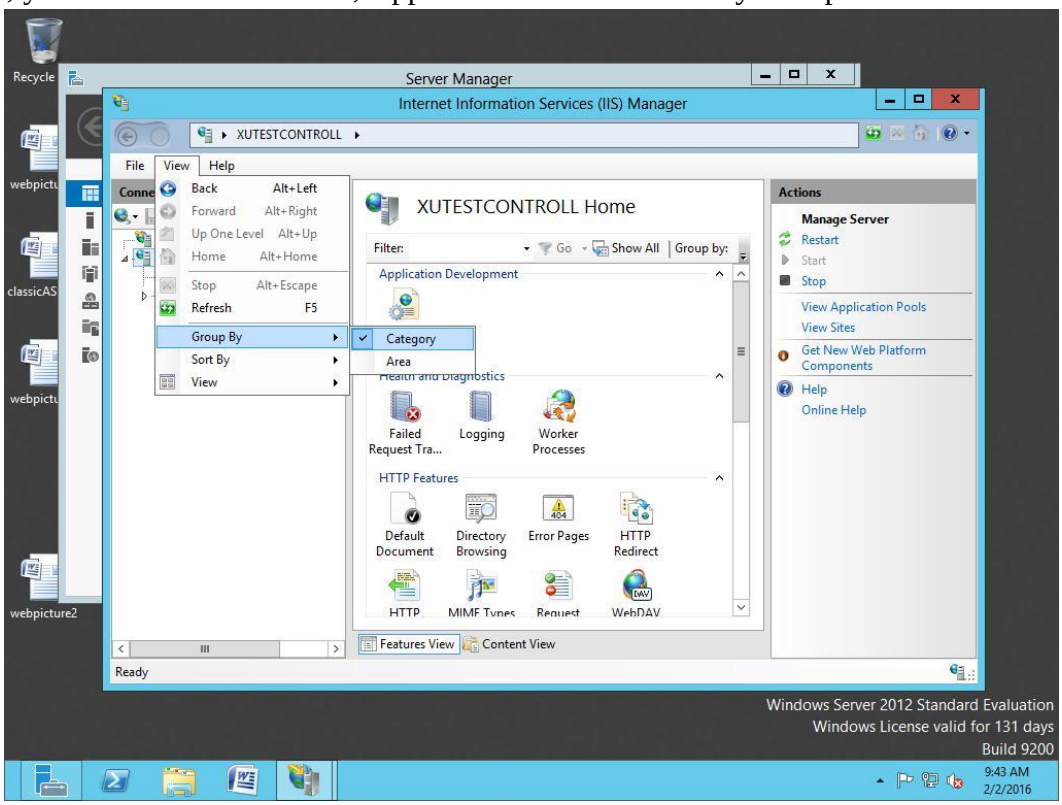


Fig. 18 View Web Server Home

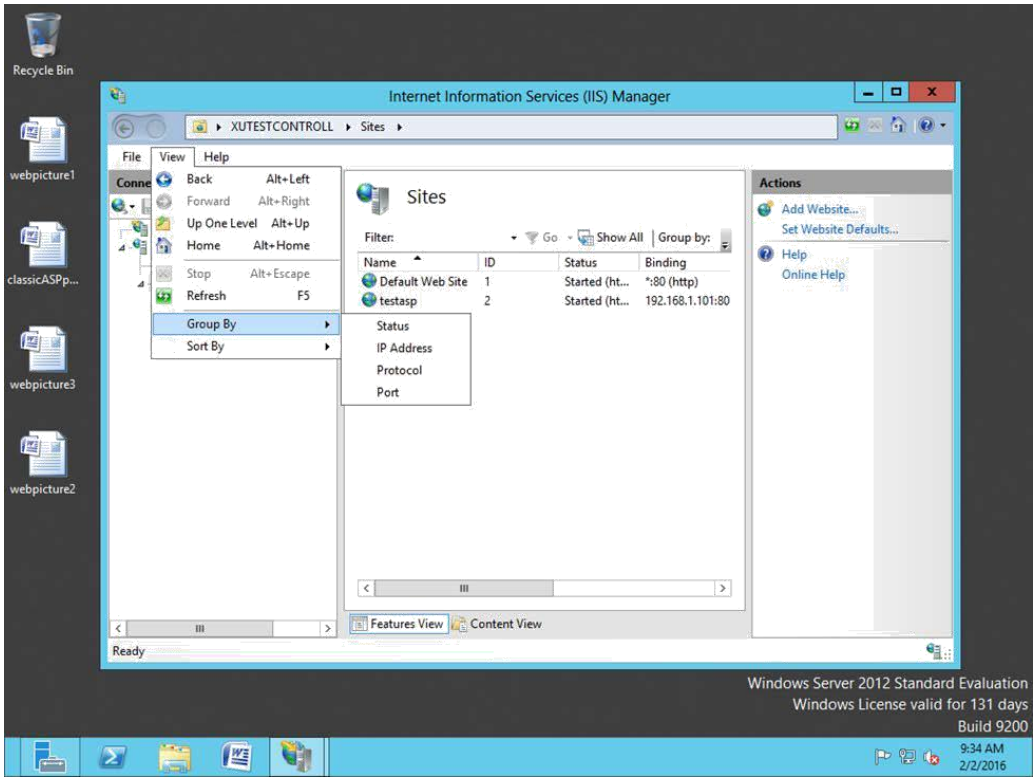


Fig.19 View Web Site

Click **Help**, you can get IIS Help, TechNet, MSDN, .NET and KBs online.

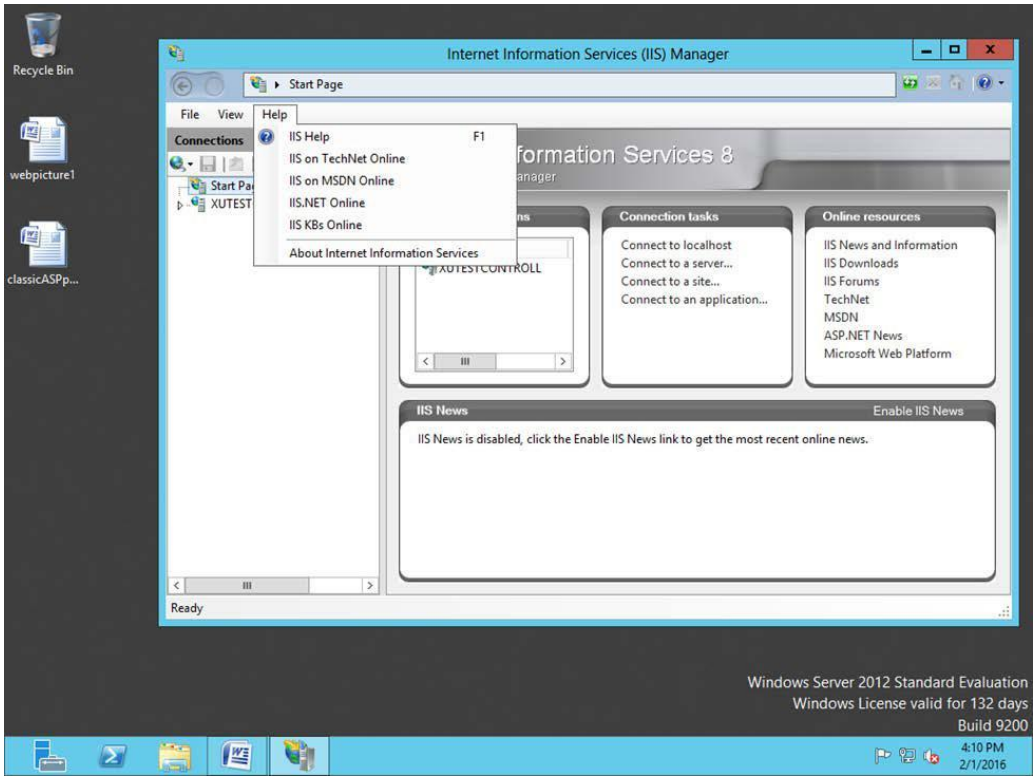


Fig. 20 Help Pop down menu

10.4 Working with classic ASP Web Homepage

We can load classic ASP sources codes to IIS 8 in Microsoft Windows Server 2012 or 2012 R2. The work steps are the following:

10.4.1 Installing Classic ASP

The classic version of ASP is not installed by default. The classic ASP page could not display in the browser, and got HTTP 404 errors.

To support and configure ASP applications on the Web server, you must install the classic ASP module, and use the following steps in Windows server 2012 and 2012 R2.

Click **Server Manager**

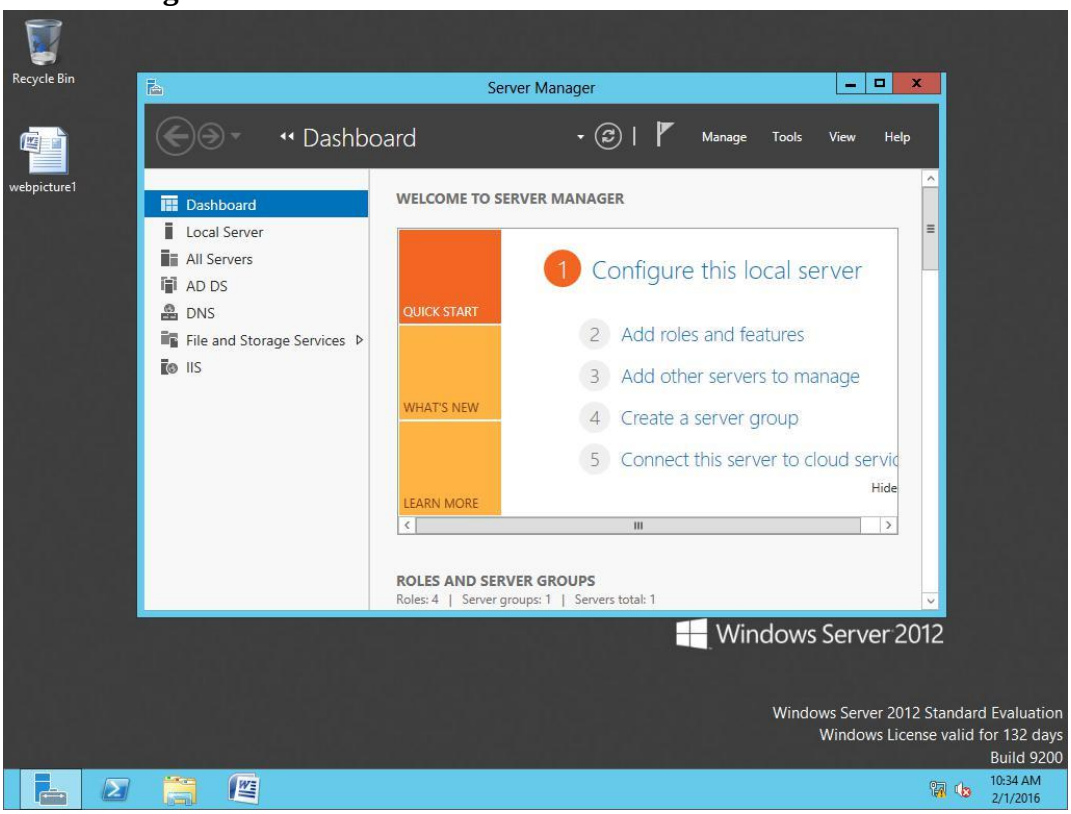


Fig. 21 Server Manager

In **Server Manager**, click the **Manage** menu, and then click **Add Roles and Features**.

In the **Add Roles and Features** wizard, click **Next**. Select the installation type and click **Next**. Select the destination server and click **Next**.

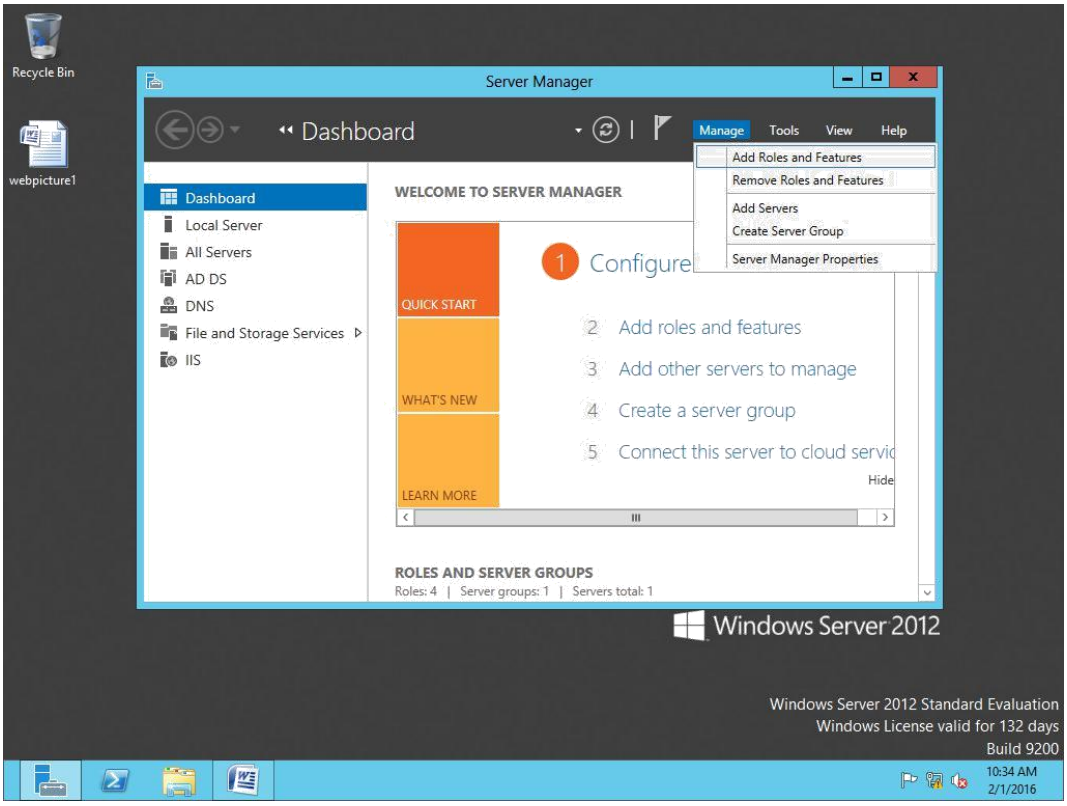


Fig. 22 Add Roles and Features in Server Manager

On the **Server Role*s*** *page*, *expand* ***Web Server (IIS)***, *expand* ***Web Server***, *and then expand* **Application Development,** select **ASP**

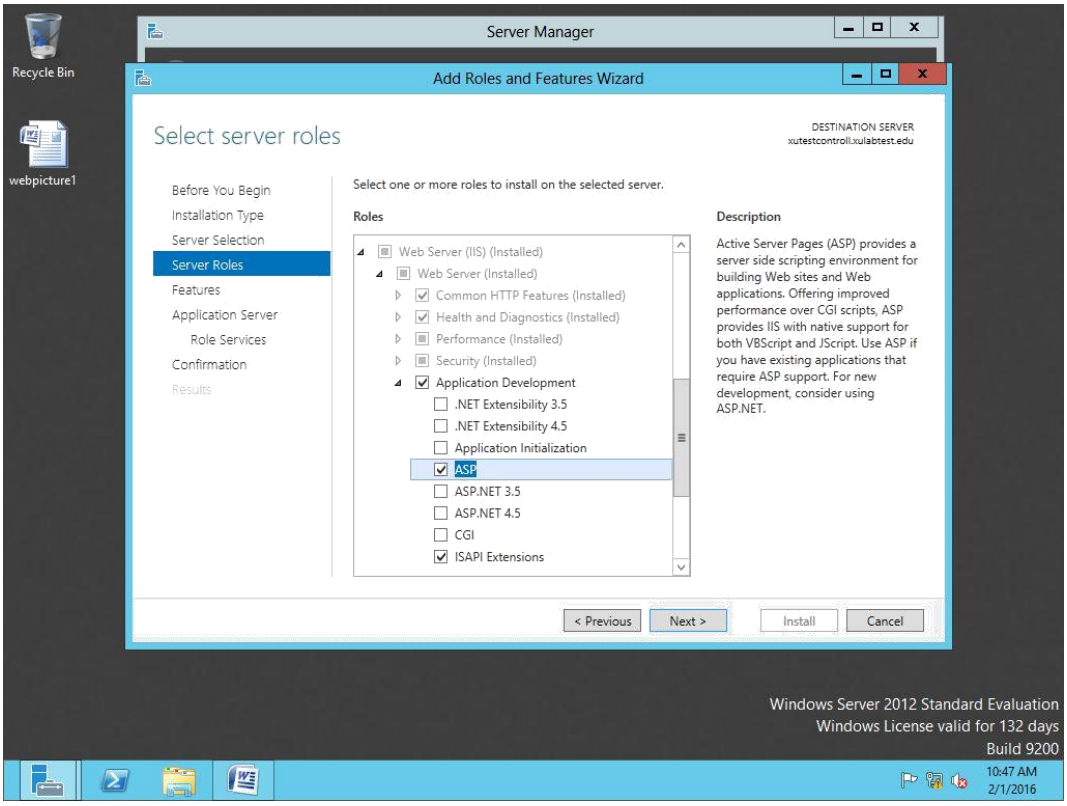


Fig.23 Select **ASP** on the **Server Roles**

On the **Server Roles** page**, ASP** and **ISAPI Extensions** should be selected. Click **Next.**

On the **Features** page, click **Next**.

On the **Confirmation** page, click **Install**.

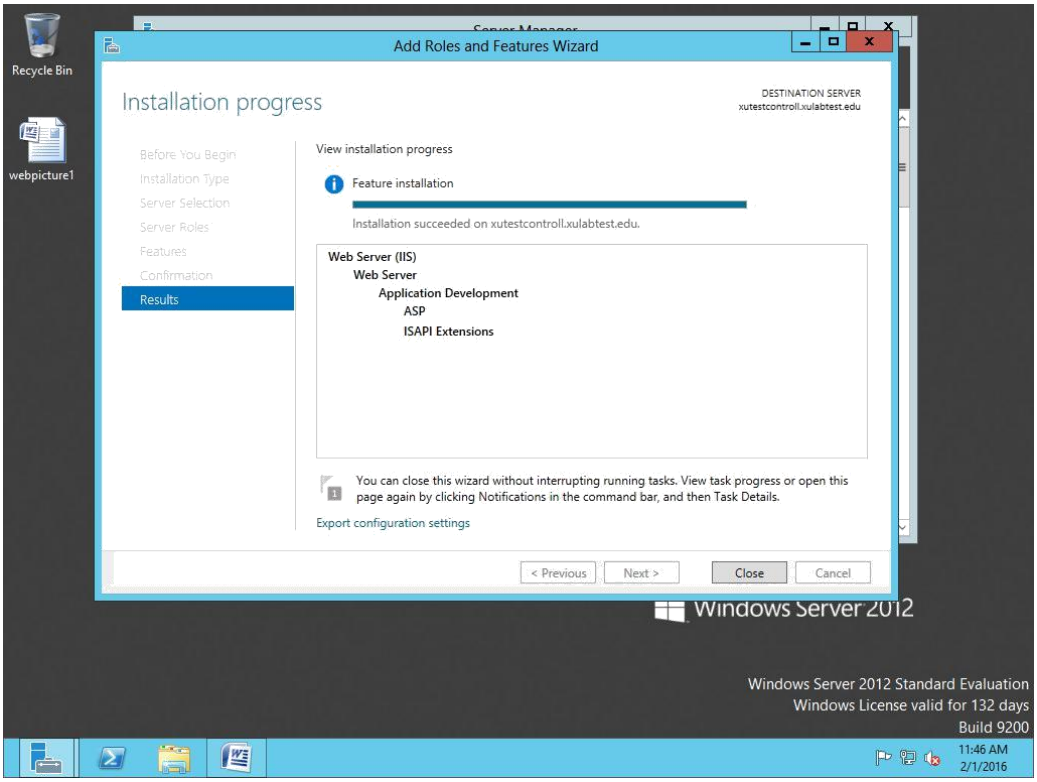


Fig. 24 ASP has been successful to install on Web Server

10.4.2 Adding a classic ASP Website

After installing classic ASP, you need to create a Website. The work steps are the following:

Open **IIS Manager**.

On the **Server Manager Dashboard**, click the **Tools** menu, and then click **Internet Information Services (IIS) Manager**

Click right-side mouse on Web Server, and click **Add Website**

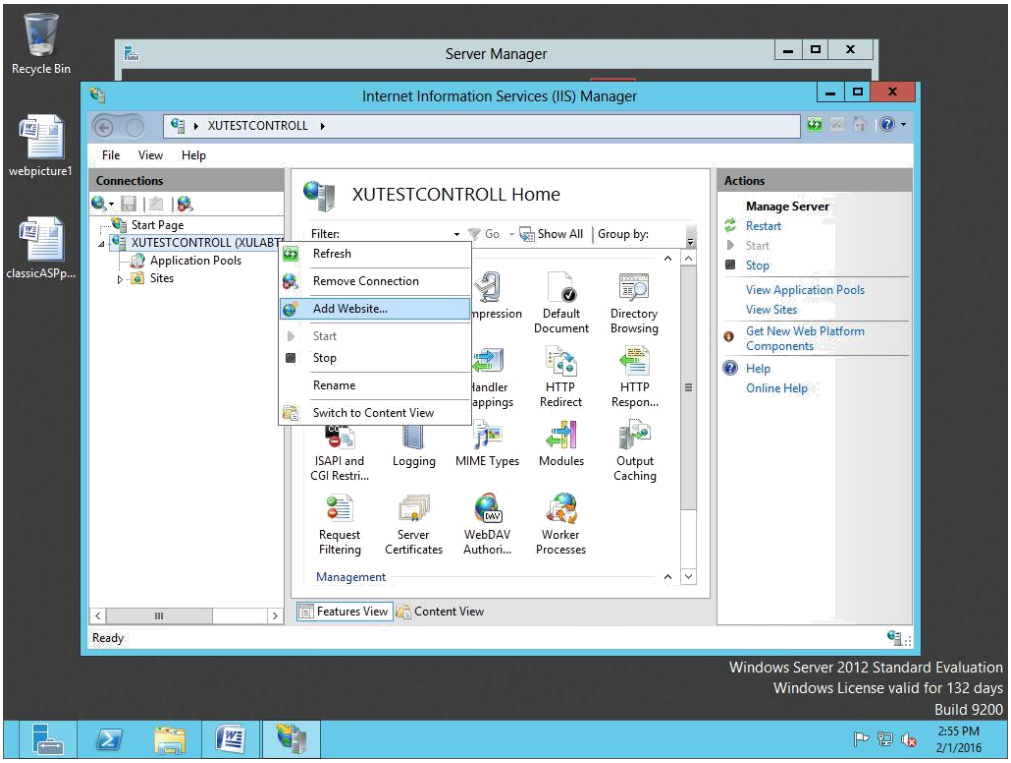


Fig.25 Add Website

You need to type Site name, physical path, IP address and port number, and mark Start Website immediately. Click **OK**.

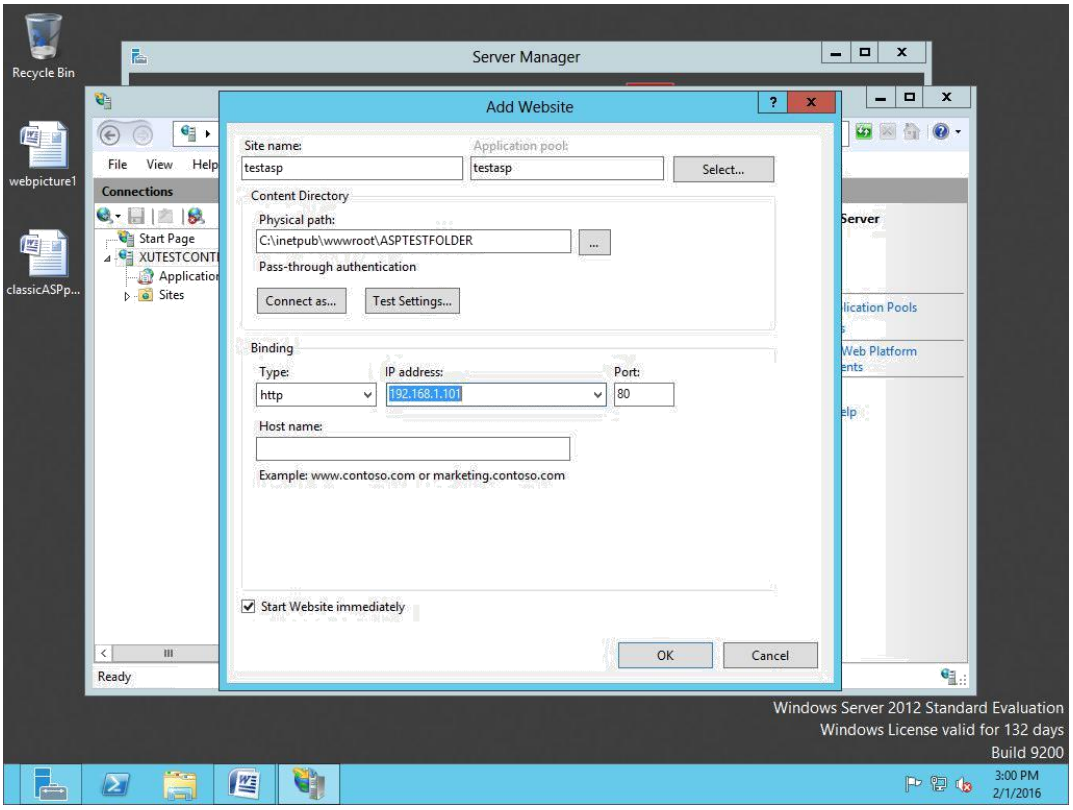


Fig. 26 Input the information of Website

The new website has been created. You can see new Website testasp under Site.

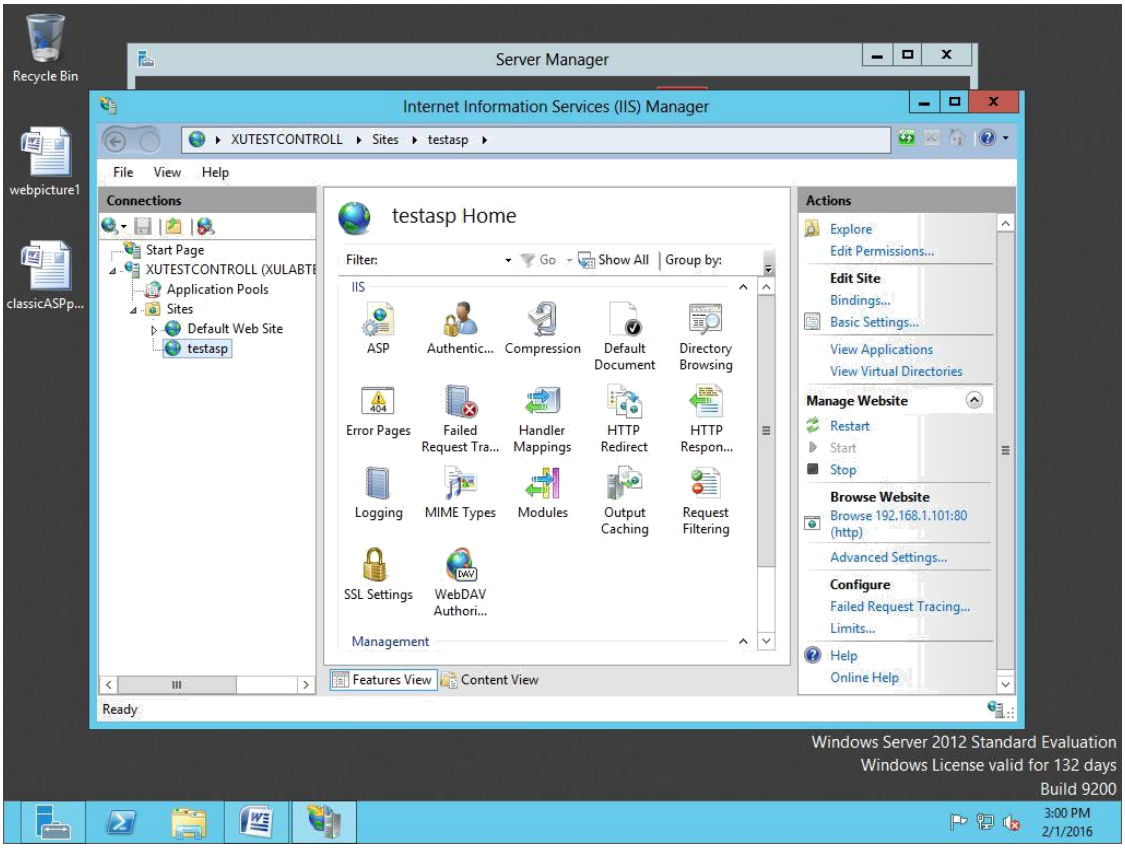


Fig. 27 New Website under Site

10.4.3 Working with classic ASP Web page

Now you can load ASP sources codes in the new Website.

Copying all ASP files to the folder of the new Website.

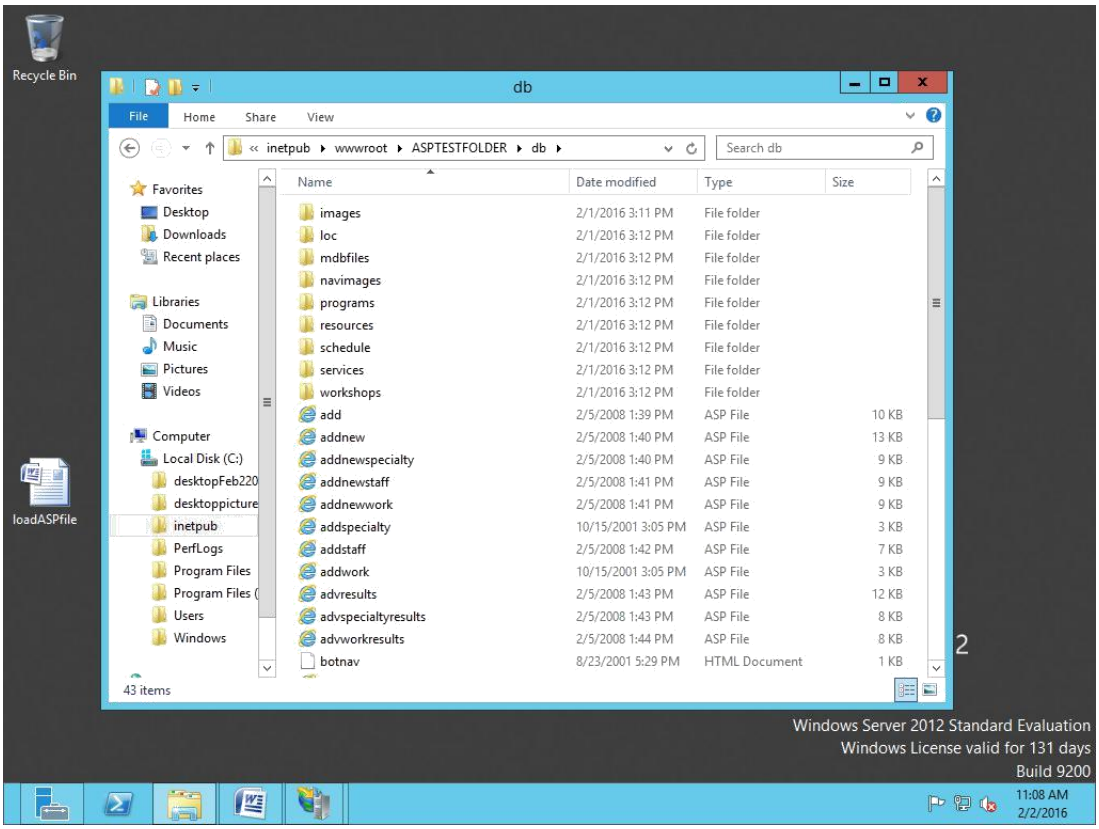


Fig. 28 All ASP sources codes to the folder of the new Website

In the IIS manager, all ASP files in the new Website

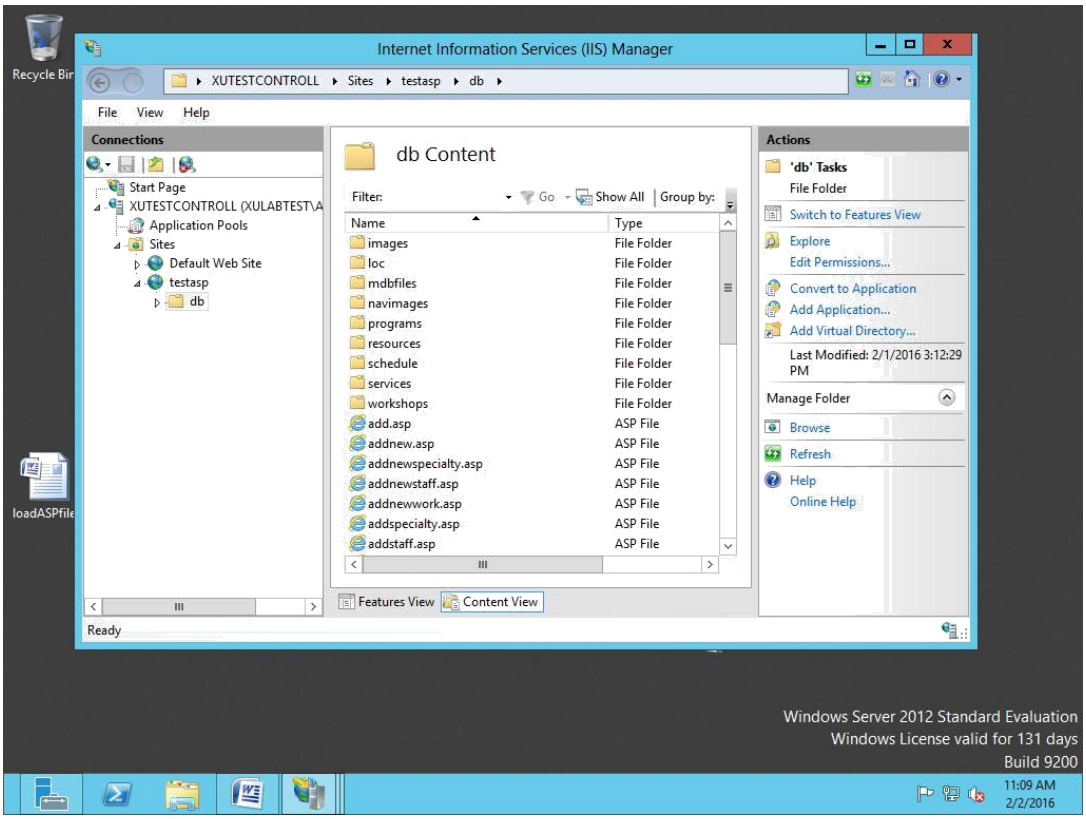


Fig.29 ASP file in the new Website Classic ASP Web page was displayed by Internet Explorer

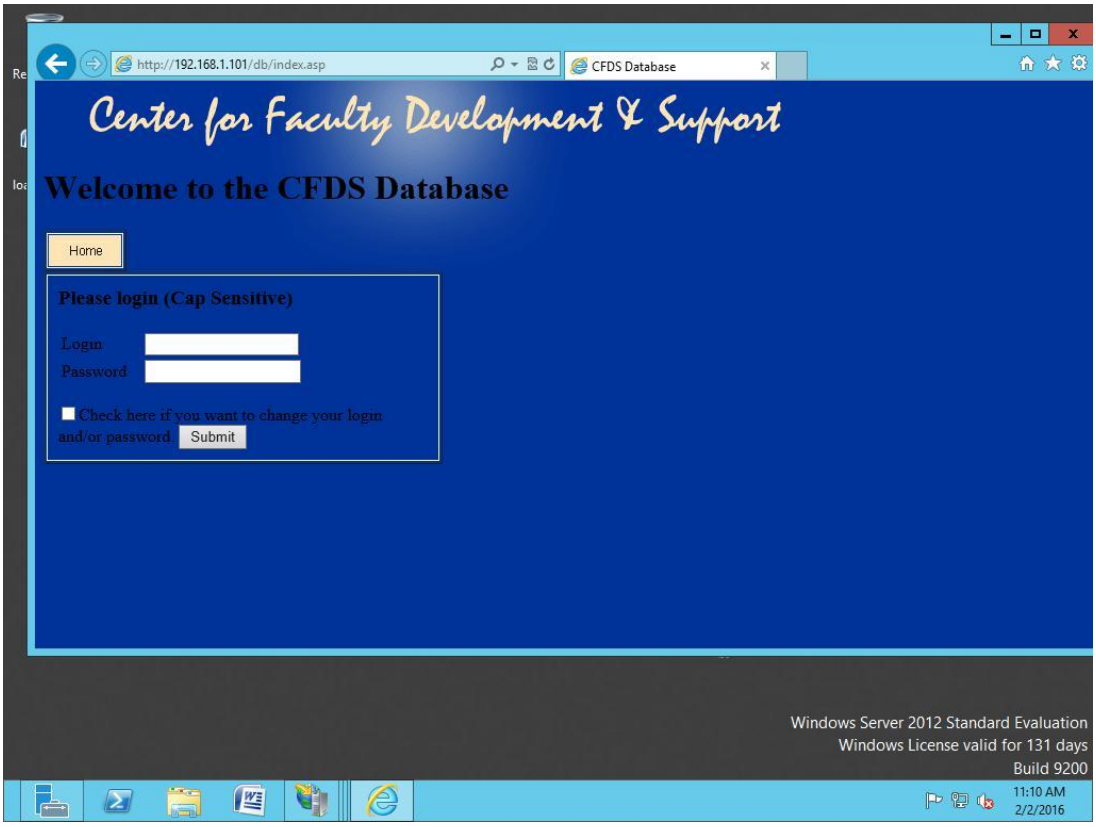


Fig. 30 Classic ASP Webpage displayed on IIS 8

LAB 11 Managing the Firewall in Windows Server 2012

The managing windows firewall on Windows Server 2012 is important tasks to the server administrators. This tasks include managing the firewall settings and creating custom inbound and outbound firewall rules.

We apply for Windows Firewall with Advanced Security correctly, and will reduce the risk of network security threats, safeguard sensitive data and intellectual property and extend the value of existing investments.

11.1 Accessing Windows Firewall with Advanced Security managing console

The Windows Firewall with Advanced Security is a host-based firewall that runs on Windows Server 2012 and is turned on by default. Firewall settings within Windows Server 2012 are managed from within the Windows Firewall MMC (Microsoft Management Console). To review and set Firewall settings perform the following:

Step 1: Open the **Server Manager** from the task bar.

Step 2: **Tools menu** and select **Windows Firewall with Advanced Security**.

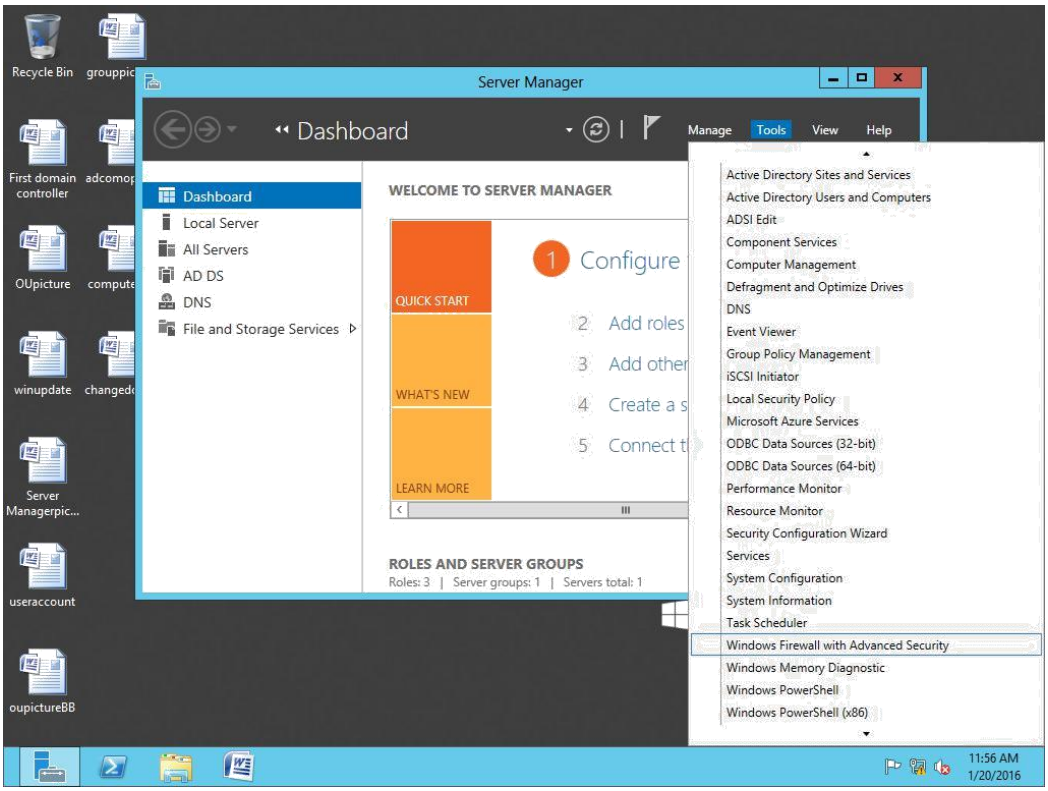


Fig. 1 Accessing Windows Firewall with Advanced Security in Server Manager

We need to review the current configuration settings by **Windows Firewall Properties** from the MMC landing page.

Click **Windows Firewall with Advanced Security** to get **Windows Firewall with Advanced Security on local computer.**

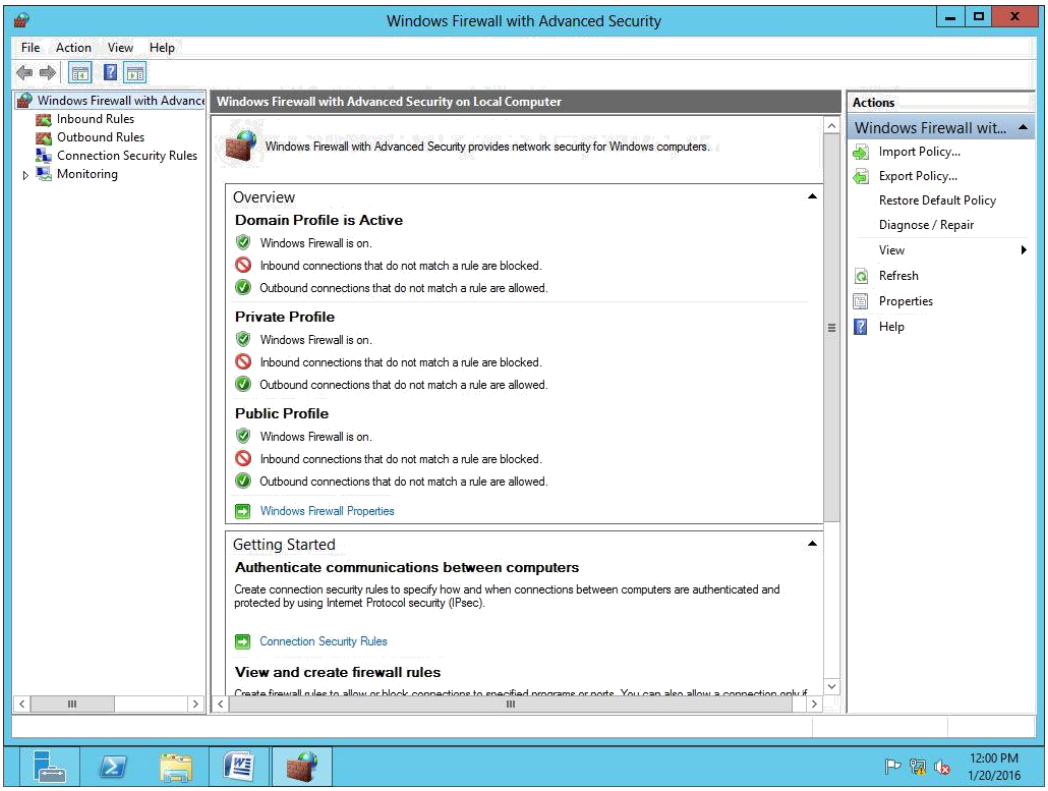


Fig. 2 Windows Firewall with Advanced Security

Right-click Windows Firewall with Advanced Security that located on above left side, to get Windows Firewall with Advanced Security on local computer.

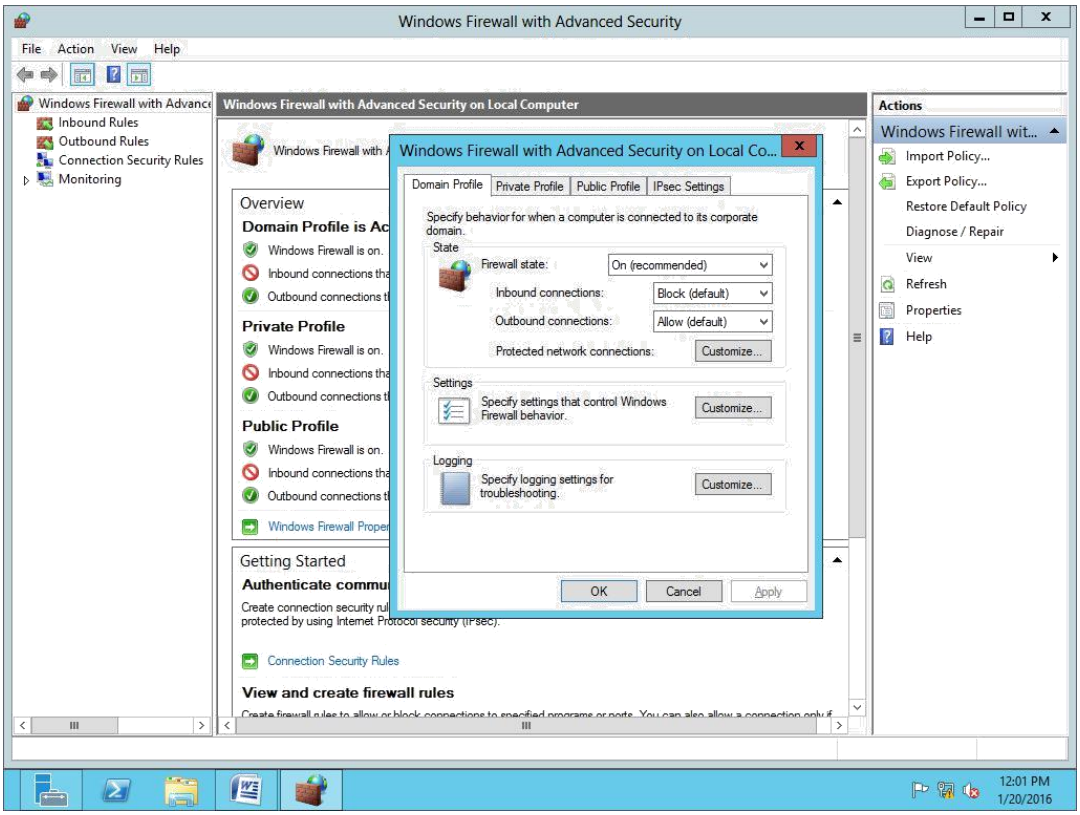


Fig.3 Windows Firewall with Advanced Security on local computer

There are three firewall profiles, **Domain, Private, and Public** and one **IPSec settings**.

11.2 Creating and configuring new rule in inbound

Click above left side **Inbound Rules**, all Inbound Rules will be listed on right window. Right-click the rule, the managing rules menu will be showed. The managing rules will included Enable Roles, Cut, Copy, Delete and Properties.

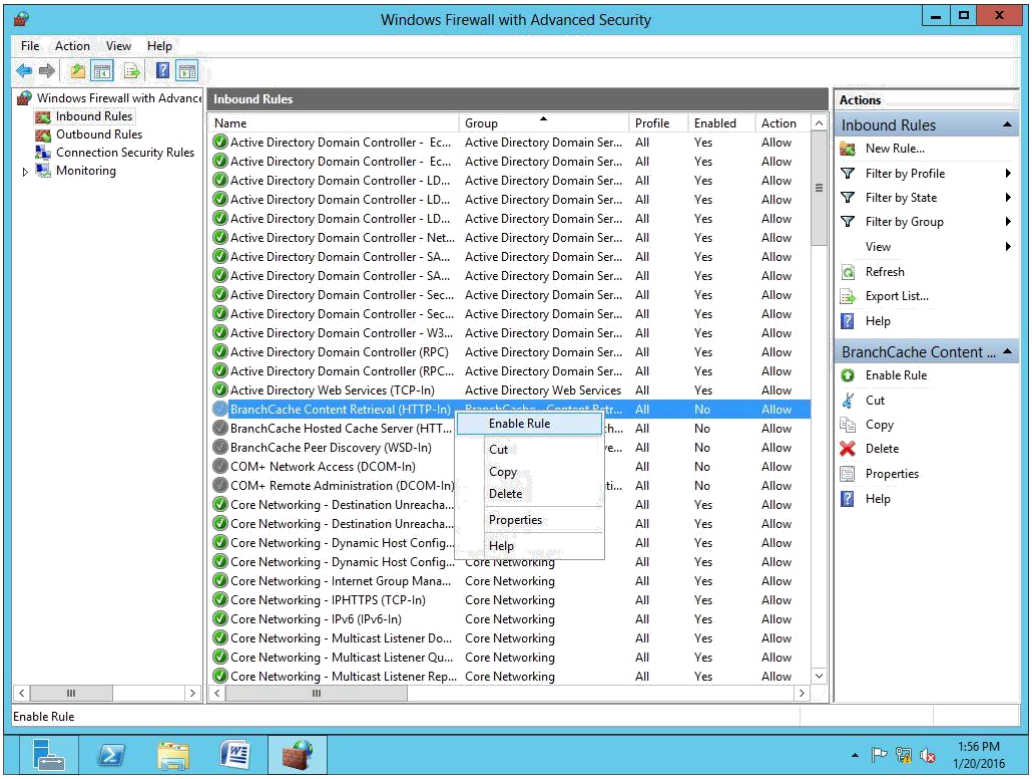


Fig.4 Inbound rules list

Click right side **New Rule…** under **Inbound Rules**. New Inbound Rule Wizard comes.

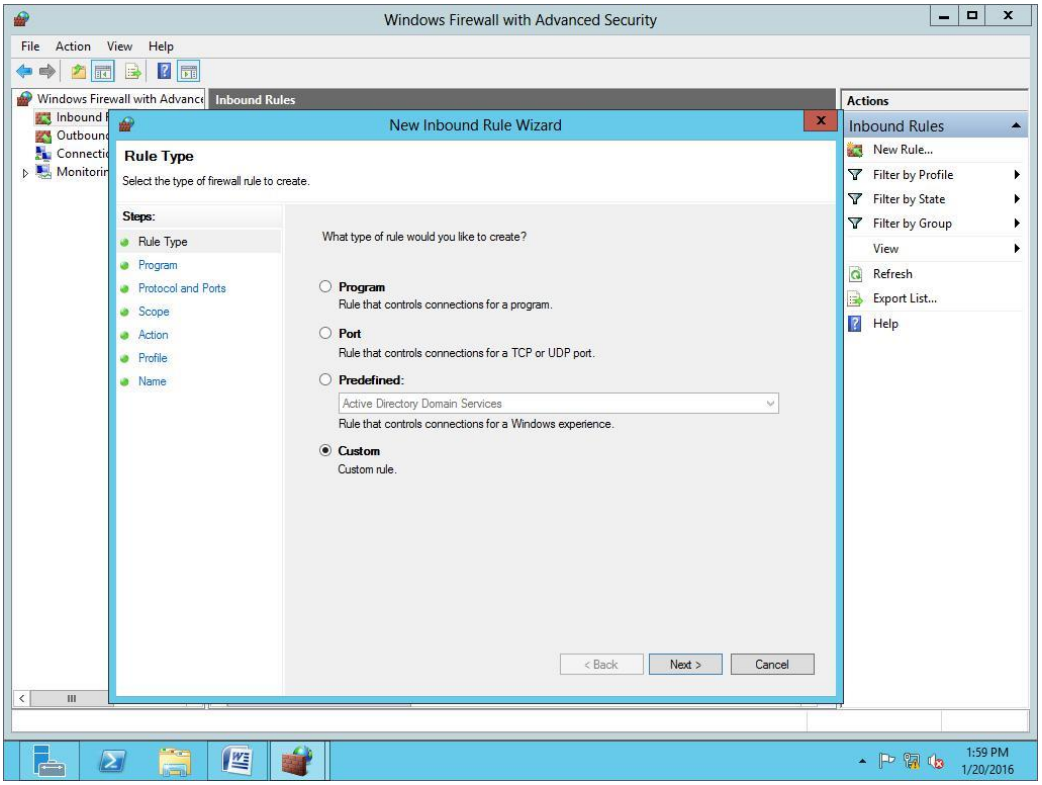


Fig. 5 New Inbound Rule Wizard and Rule Type

Click **Custom** and **Next**

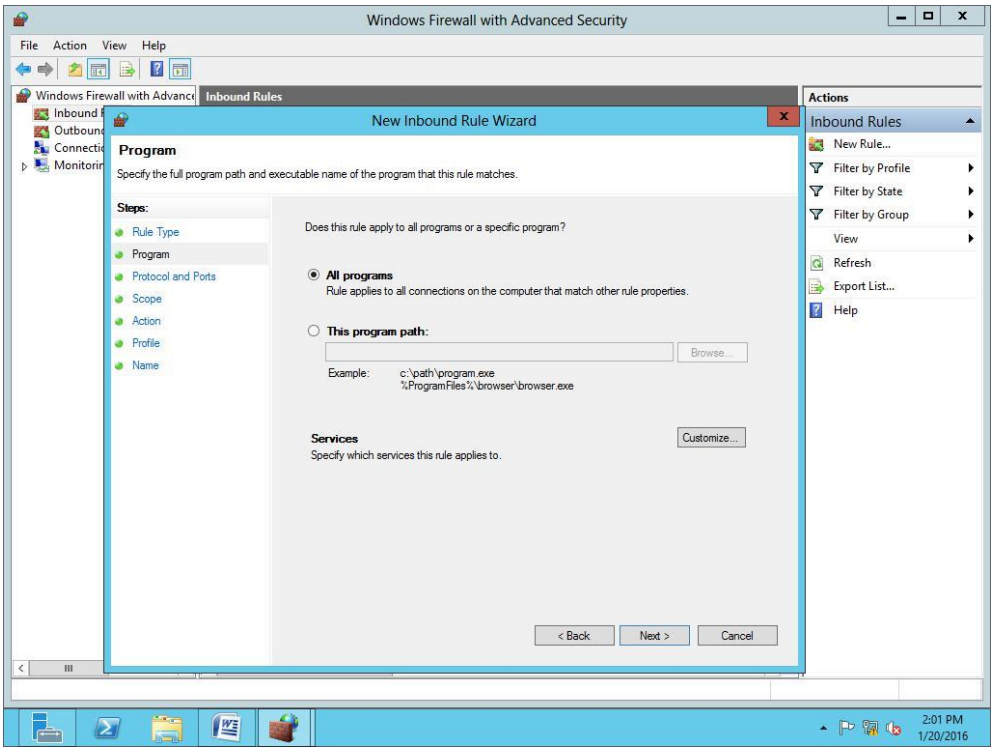


Fig.6 Rule apply to all programs

Click **Next.** Now you should choose protocol and ports

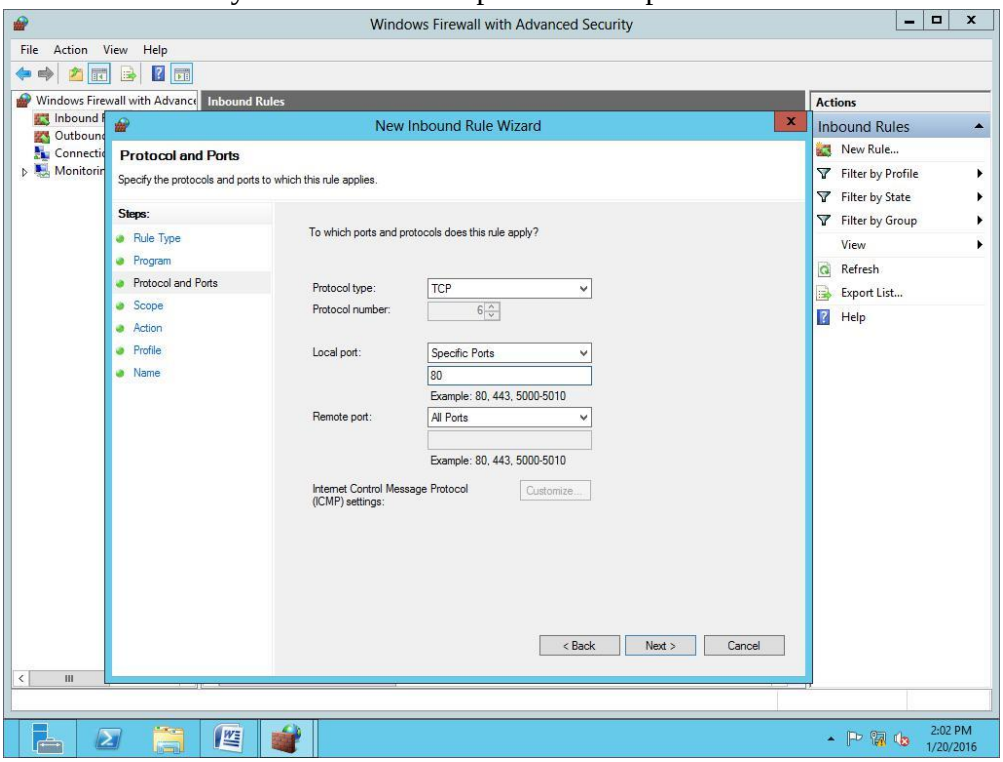


Fig. 7 Choose protocol and ports

Click **Next**, and choose local IP addresses.

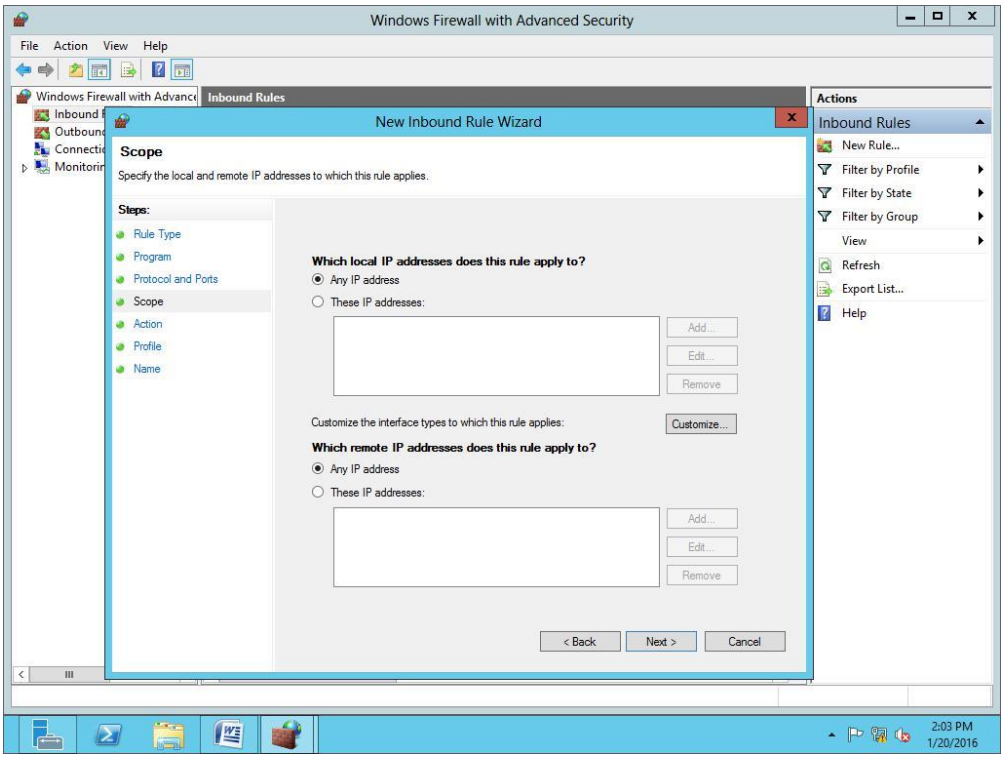


Fig. 8 Choose local IP addresses

Click **Next**, and choose connection matches.

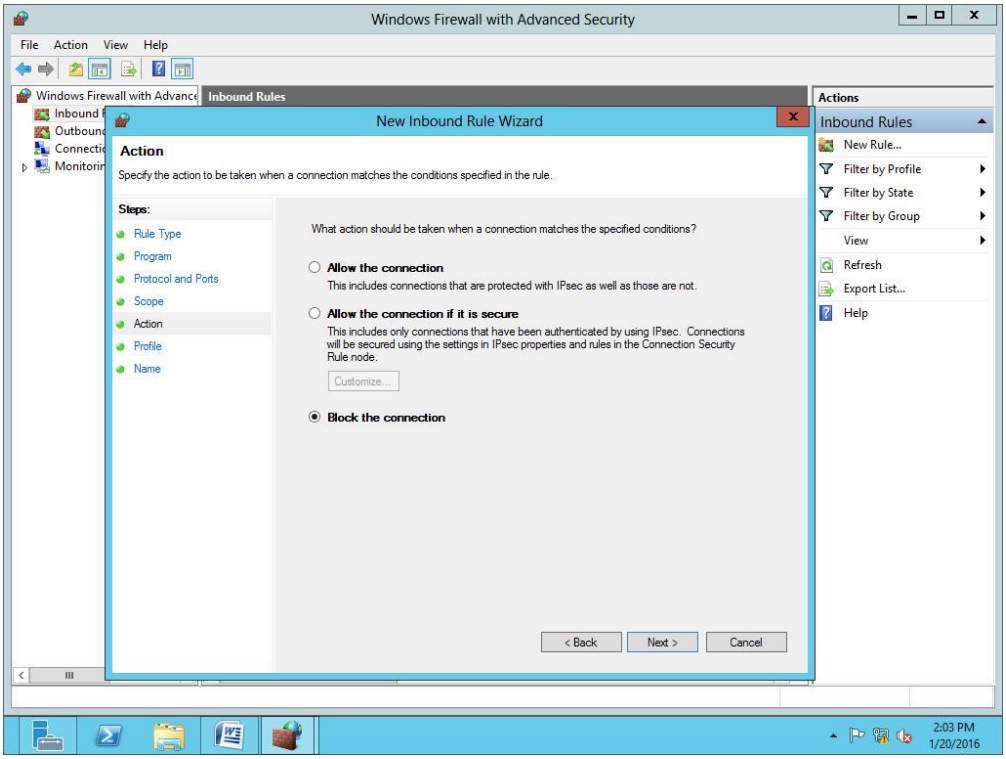


Fig. 9 Action for rule

Click **Next**. And rule apply

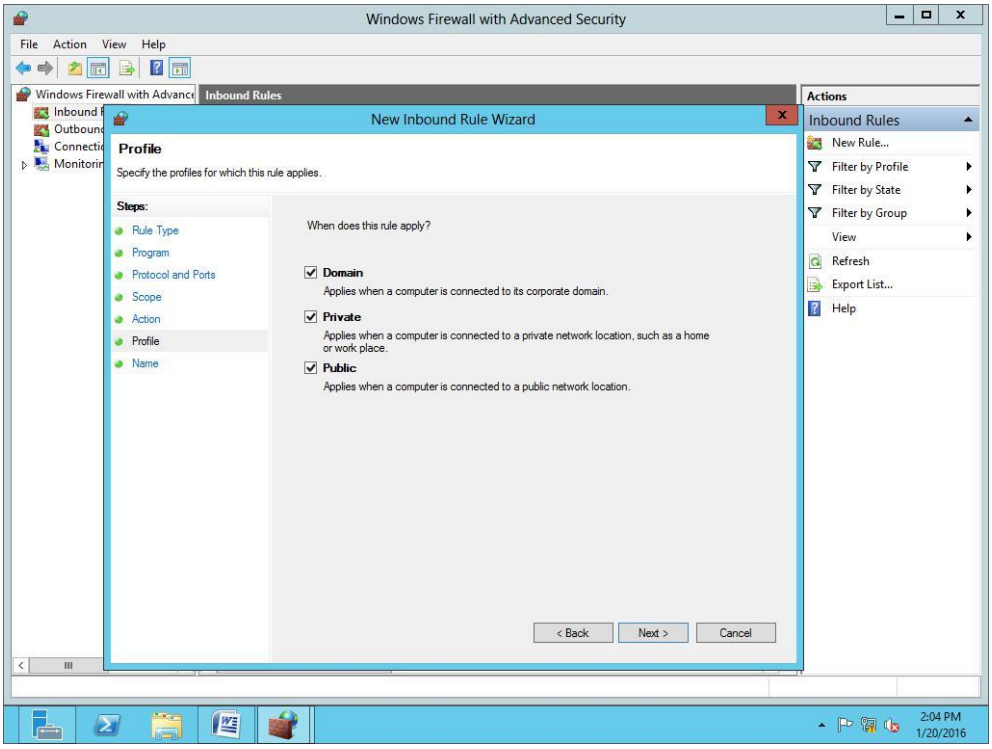


Fig. 10 Rule apply

Click Next, Typing name and description

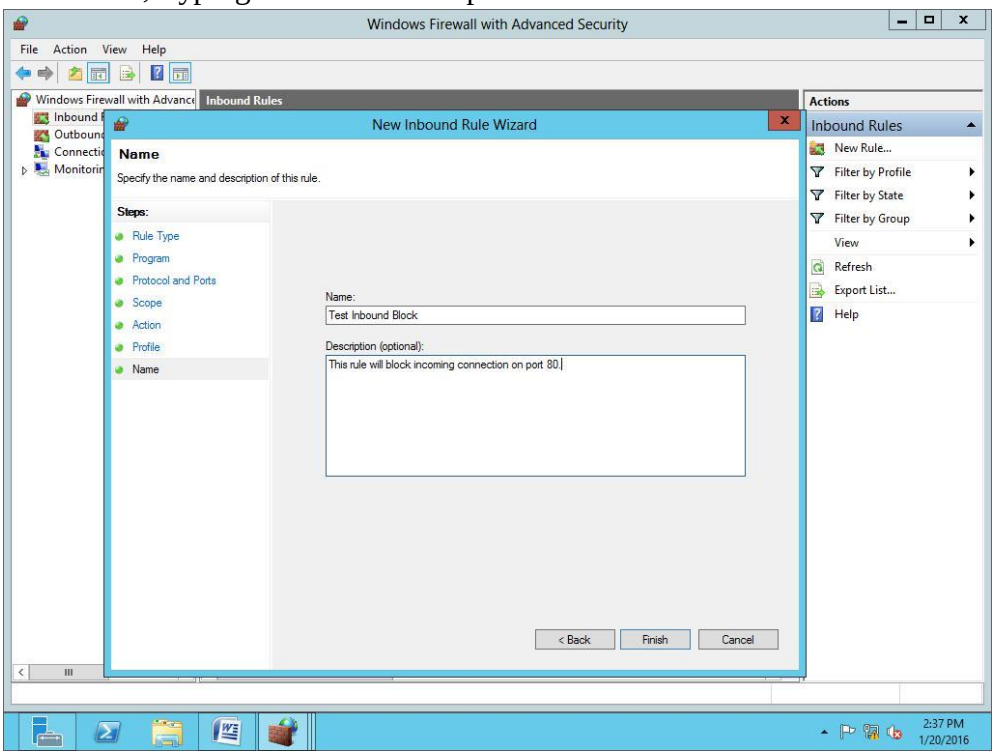


Fig. 11 Typing name and description of the rule

Click **Finish**, the new rule is listed in Inbound Rules

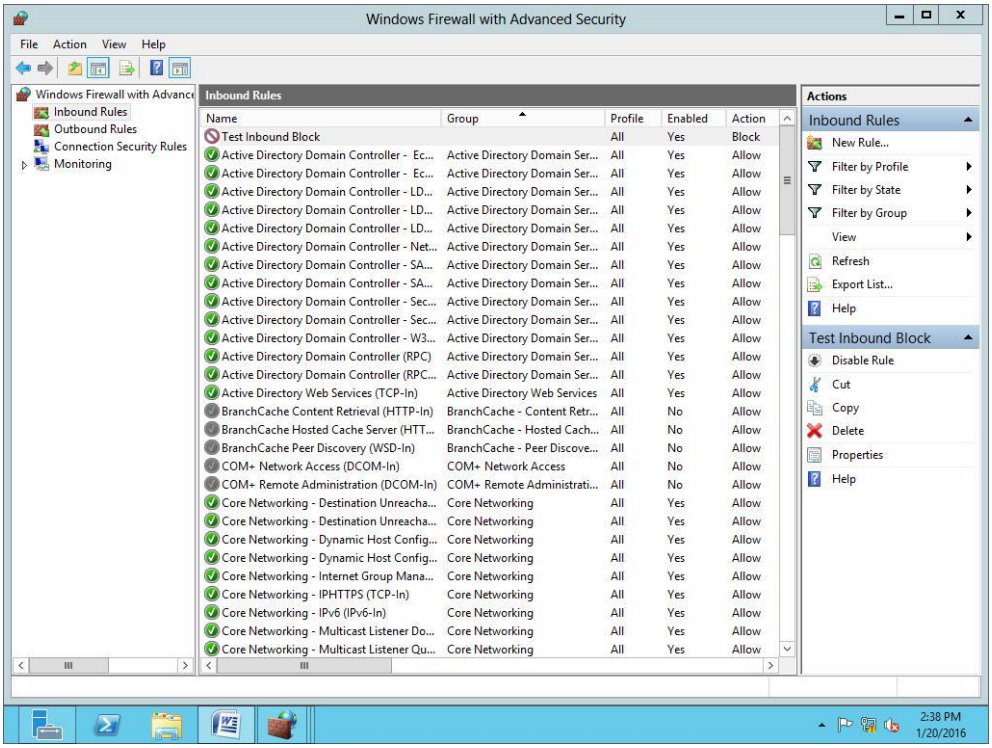


Fig. 12 New rule is listed in Inbound Rules Click-right new rule name, click Properties. The rule properties will be showed.

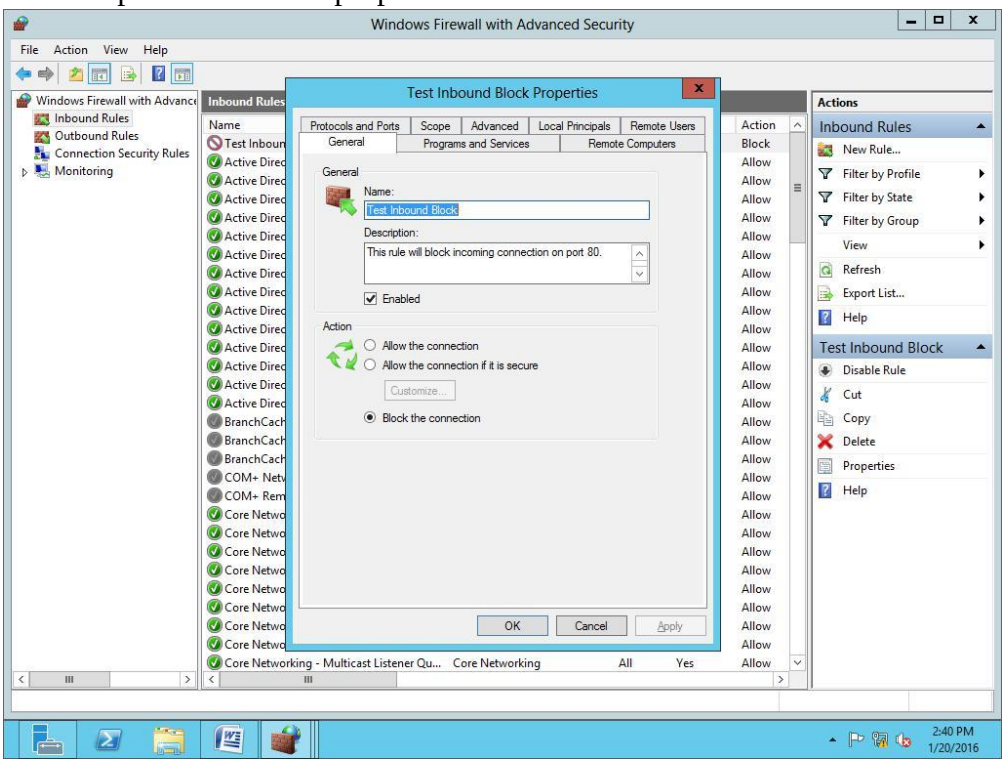


Fig. 13 New rule properties

Click Right on new rule, the managing menu will be showed. You can Disable Rule, Cut, Copy, and Delete.

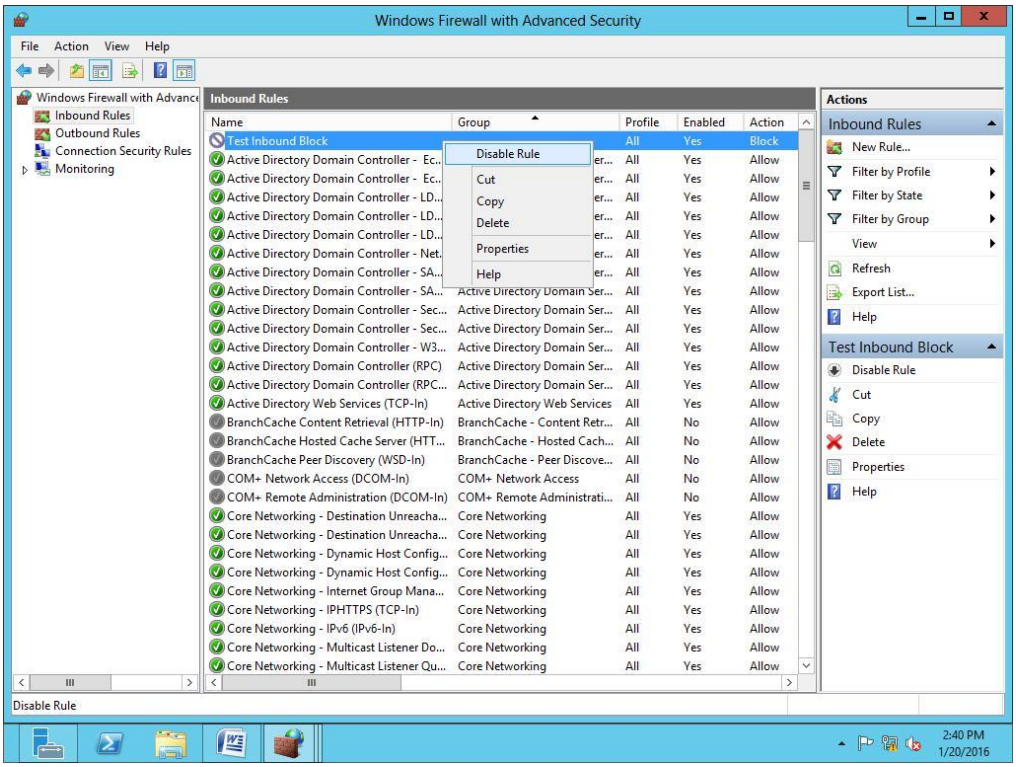


Fig. 14 Disable Rule, cut, Copy, and Delete rule

11.3 Monitoring in the Windows Firewall with Advanced Security

The Monitoring in the Windows Firewall with Advanced Security allows us to monitor the active firewall rules and connection security rules on the computer.

The Monitoring included viewing active firewall rules, connection security rules and security association.

Click **Monitoring**, and get monitoring console.

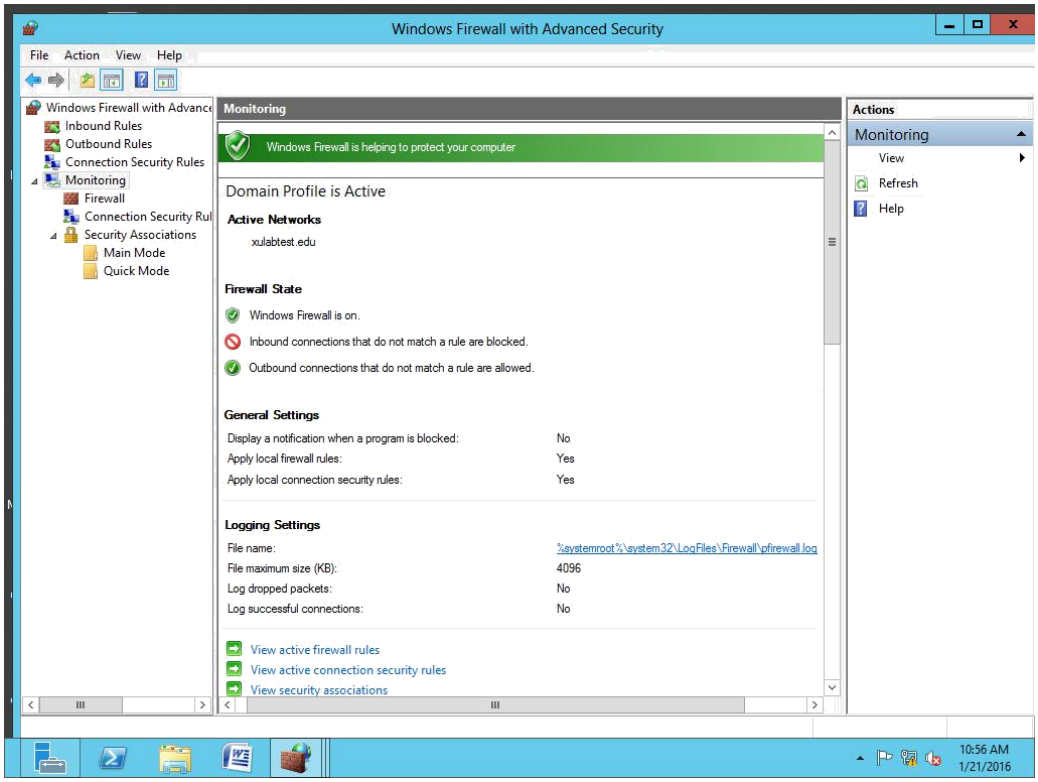


Fig.15 Monitoring console

Click **active firewall rules**, and will list active firewall rules

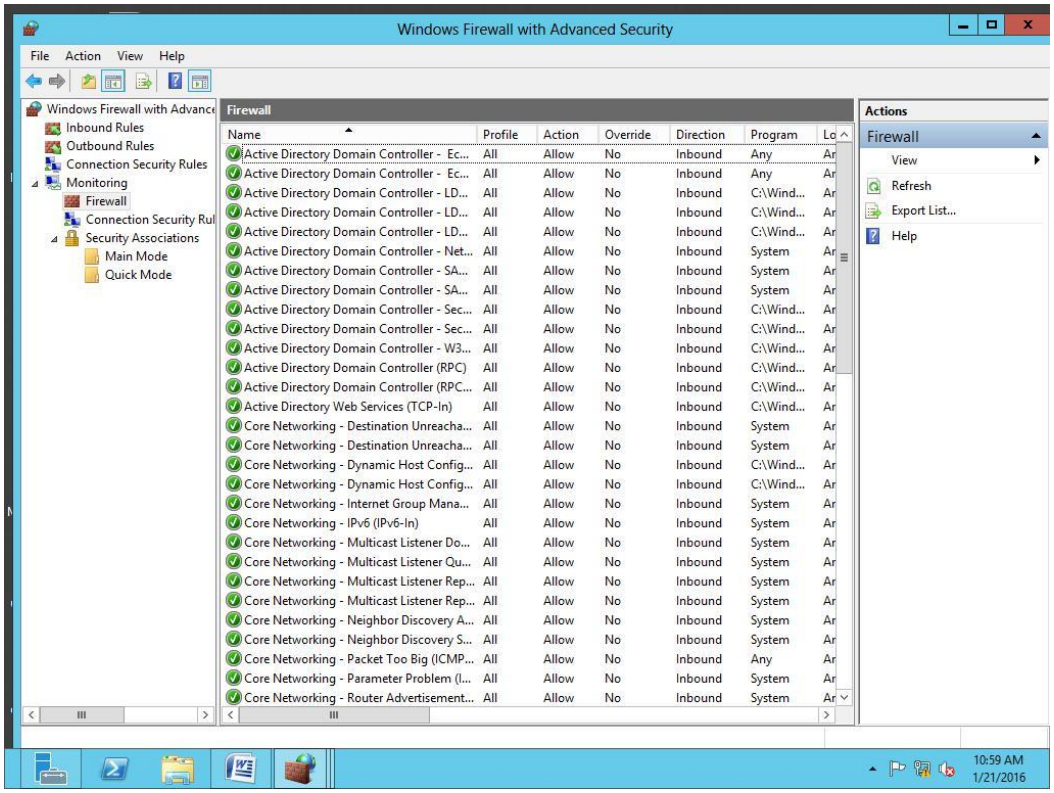


Fig. 16 Viewing active firewall rules

Click **connection security rules,** and list Connection security rules

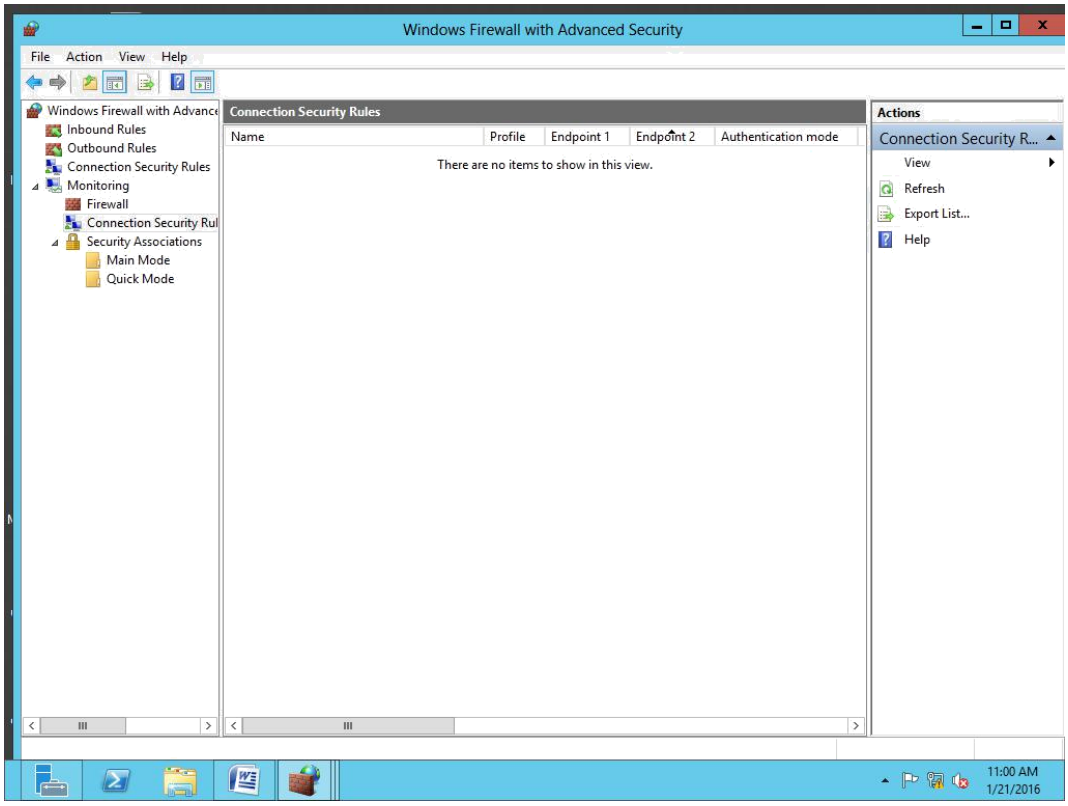


Fig. 17 Viewing connection security rules

Click view security association, and will list security association

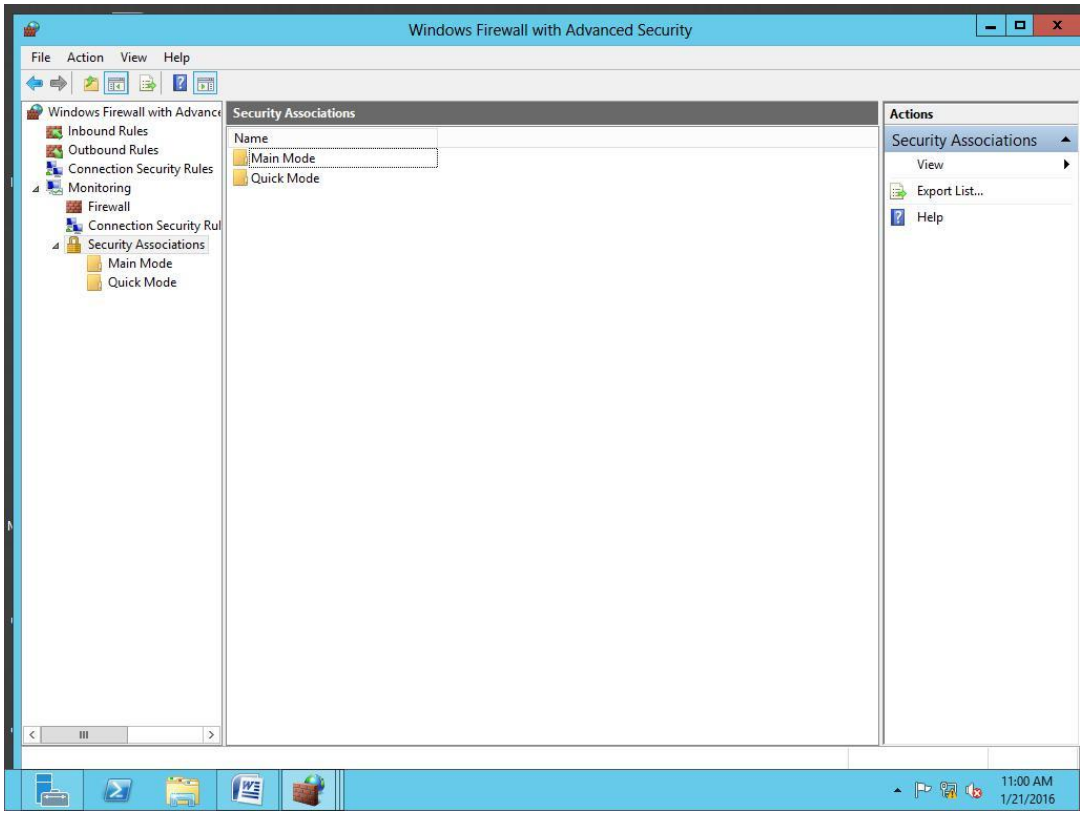


Fig.18 Viewing security association

The security association lists active main mode and quick mode security associations (SAs). The main mode lists all of the main mode SAs with detailed information about their settings and endpoints. We also can use this folder to view the IP addresses of the endpoints and the methods and algorithms that were used for authentication. The quick mode lists all of the quick mode SAs with detailed information about their settings and endpoints. We can use this folder to view the IP addresses of the endpoints and the integrity and encryption algorithms in use to protect traffic exchanged between the two endpoints.