CCNA- Portable Command Guide

Requirement	Cisco Command
Enter privilege mode	Router>**enable**
Return to user mod	Router# **disable**
Enter the configuration mode	Router# **configuration terminal**
To add hostname for router or switch	Router(config)#**hostname router_name**
To display the motd banner	Router(config)#**banner motd "type message here"**
Password Encryption	
Set a console password to cisco	Router(config)#**line con 0** Router(config-line)#**login** Router(config-line)#**password cisco**
Set a telnet password	Router(config)#**line vty 0 15** Router(config-line)#**login** Router(config-line)#**password cisco**
Set the enable password to cisco	Router(config)#**enable password cisco**
Set the enable secret password. This password overrides the enable password and is encrypted within the config file	Router(config)#**enable secret class**
Configuring a Gigabit Ethernet Interface with IPv4	
Moves to gigabitethernet 0/0 interface configuration mode	Router(config)#**interface gigabitethernet 0/0**
Optional descriptor of the link is locally significant	Router(config-if)#**description Accounting LAN**
Assigns address and subnet mask to interface	Router(config-if)#**ip address 192.168.20.1 255.255.255.0**
Turns interface on	Router(config-if)#**no shutdown**
Configuring a Gigabit Ethernet Interface with IPv6	
• Enables the forwarding of IPv6 unicast datagrams globally on the router	Router (config)#**ipv6 unicast-routing**
• Moves to gigabitethernet 0/0 interface configuration mode	Router(config)#**interface gigabitethernet 0/0**
• Assigns an IPv6 address to this interface	Router (config-if)#**ipv6 address 2001:db8:c003:1104::1/64**
• Optional descriptor of the link is locally significant	Router(config-if)#**description Accounting LAN**

• Configures a specific link-local IPv6 address	Router(configif)#**ipv6 address fe80::2 link-local**
• Turns interface on	Router(config-if)#**no shutdown**
Basic Security Practices	
• Encrypt all passwords in the configuration file:	R(Config)# **service password-encryption**
A specific amount of time using the command • This command will block login attempts for 120 seconds if there are three failed login attempts within 60 seconds	R(Config)# **login block-for** 120 **attempts 3 within 60**
• Security passwords min-length	R(Config)# **Security passwords min-length** 10
• **Exec timeout on a router** ○ For Console line:	R(Config)# Line console 0 R(Config-lin)# exec-timeout **10** R(Config-lin)# exit
○ For VTY line:	R(Config)# Line VTY 0 15 R(Config-lin)# exec-timeout **10** R(Config-lin)# exit
SSH (A Cisco device to support SSH using four steps)	
• **Step 1**: Configure the IP domain name.	R(config) # **ip domain-name** cisco.com
• **Step 2:** Generate one-way secret keys.	R(config) # crypto key generate rsa **press Enter** **1024**
• **Step 3:** Verify or create a local database entry. • Create a user **Bop** with a privilege level of **15** using the encrypted password for **Class**. OR • Create a user **Bop** with password for **Class**.	R(config) # username **Bop** privilege **15** Secret **Class** OR R(config) # username **Bop** password **Class**
• **Step 4:** Enable VTY inbound SSH sessions	R(config) # Line vty 0 4 R(config-line) # login local R(config-line) # transport input ssh R(config-line) # exit

SSH version 2	R(config) # ip ssh version 2
limited to 2 authentication attempts	R(config) # ip ssh authentication-retries 2
a 60 second timeout	R(config) # ip ssh time-out 60

• **To Disable DNS lookup:** • To decrease user delays if no **DNS** server is configured.	R(Config)# no ip domain-lookup
To save the current configuration from DRAM (**running-config**) to NVRAM (**startup-config)**	Router# **Copy running-config startup-config**
To save the current configuration from DRAM to TFTP Server	Router# **Copy running-config** tftp: Address or name of remote host [ ]? 192.168.1.20
Configuring a SVI Interface with IPv4 on a Switch	
Moves to VLAN interface configuration mode	Switch (config)#**interface VLAN 1**
Assigns address and subnet mask to interface	Switch (config-if)#**ip address 192.168.0.1 255.255.255.0**
• Configure the default gateway.	Switch(config)#**ip default-gateway** 192.168.0.1

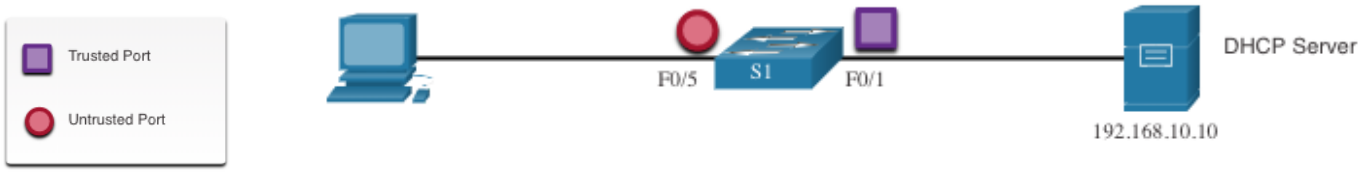
- **<u>Common show commands include:</u>**
    - **show running-config**
    - **show interfaces**
    - **show ip interface brief**
    - **show arp**
    - **show ip route**
    - **show protocols**
    - **show version**
- When using windows, use the **tracert** command.
- When performing a trace from a router CLI, use the **traceroute** command.
- On a Windows computer, the IP address of the default gateway can be viewed by using the **ipconfig** command.
    - The **ipconfig /all** command can be used to view the MAC address as well as other important details regarding the Layer 3 addressing of the device.
    - The **ipconfig /displaydns** command displays all of the cached DNS entries on a Windows computer system.
- On a Windows computer, the **arp -a** command lists all devices currently stored in the ARP cache of a particular host.
- The arp cache can be cleared using the command **arp-d**
- **show cdp neighbors detail**
    - To disable CDP globally, use the global configuration command **no cdp run**. To disable CDP on an interface, use the interface command **no cdp enable**.
- Use the **show ip route** command to verify that the default route has been set.

Switch Port Security

VLAN	
Moves to interface configuration mode.	Switch(config)#**interface fastethernet 0/1**
Enables port security on the interface.	Switch(config-if)#**switchport port-security**
Sets a maximum limit of four MAC addresses that will be allowed on this port.	Switch(config-if)#**switchport port-security maximum 4**
Static MAC Addresses Sets a specific secure MAC address 1234.5678.90ab. You can add additional secure MAC addresses up to the maximum value configured.	Switch(config-if)#**switchport port-security mac-address 1234.5678.90ab**
Sticky MAC Addresses Converts all dynamic port security learned MAC addresses to sticky secure MAC addresses.	Switch(config-if)#**switchport port-security mac-address sticky**
security violation • Configures port security to shut down the interface if a security violation occurs. • **NOTE** In shutdown mode, the port is errdisabled, a log entry is made, and manual intervention or errdisable recovery must be used to reenable the interface.	Switch(config-if)#**switchport port-security violation shutdown**
security violation Configures port security to restrict mode if a security violation occurs. **NOTE** In restrict mode, frames from a nonallowed address are dropped, and a log entry is made. The interface remains operational.	Switch(config-if)#**switchport port-security violation restrict**
security violation Configures port security to protect mode if a security violation occurs. **NOTE** In protect mode, frames from a nonallowed address are dropped, but no log entry is made. The interface remains operational.	Switch(config-if)#**switchport port-security violation protect**

Mitigate DHCP Attacks

DHCP Snooping Configuration Example



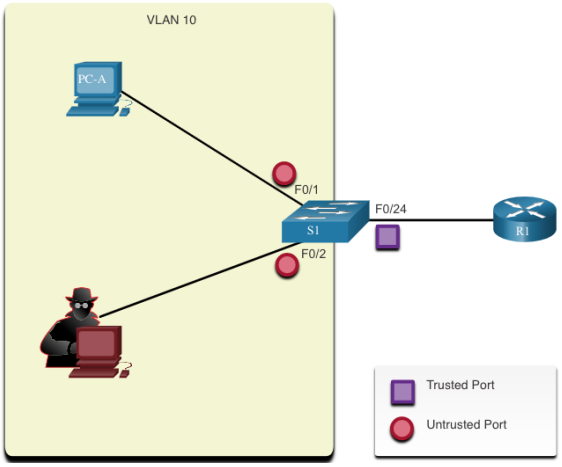
1. **DHCP snooping is first enabled on S1.**
2. **The upstream interface to the DHCP server is explicitly trusted.**
3. **F0/5 to F0/24 are untrusted and are, therefore, rate limited to six packets per second.**
4. **Finally, DHCP snooping is enabled on VLANS 5, 10, 50, 51, and 52.**

```
S1(config)# ip dhcp snooping
S1(config)# interface f0/1
S1(config-if)# ip dhcp snooping trust
S1(config-if)# exit
S1(config)# interface range f0/5 - 24
S1(config-if-range)# ip dhcp snooping limit rate 6
S1(config-if)# exit
S1(config)# ip dhcp snooping vlan 5,10,50-52
S1(config)# end
S1#
```

Mitigate ARP Attacks

Dynamic ARP Inspection

- DHCP snooping is enabled because DAI requires the DHCP snooping binding table to operate.
- Next, DHCP snooping and ARP inspection are enabled for the PCs on VLAN10.
- The uplink port to the router is trusted, and therefore, is configured as trusted for DHCP snooping and ARP inspection.




```
S1(config)# ip dhcp snooping
S1(config)# ip dhcp snooping vlan 10
S1(config)# ip arp inspection vlan 10
S1(config)# interface fa0/24
S1(config-if)# ip dhcp snooping trust
S1(config-if)# ip arp inspection trust
```

The **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]} global configuration command is used to configure DAI to drop ARP packets when the IP addresses are invalid.

- It can be used when the MAC addresses in the body of the ARP packets do not match the addresses that are specified in the Ethernet header.
- Notice in the following example how only one command can be configured.

Therefore, entering multiple **ip arp inspection validate** commands overwrites the previous command.

- To include more than one validation method, enter them on the same command line as shown in the output.

```
S1(config)# ip arp inspection validate ?
  dst-mac  Validate destination MAC address
  ip       Validate IP addresses
  src-mac  Validate source MAC address
S1(config)# ip arp inspection validate src-mac
S1(config)# ip arp inspection validate dst-mac
S1(config)# ip arp inspection validate ip
S1(config)# do show run | include validate
ip arp inspection validate ip
S1(config)# ip arp inspection validate src-mac dst-mac ip
S1(config)# do show run | include validate
ip arp inspection validate src-mac dst-mac ip
S1(config)#
```

VLAN

Creating Static VLANs	
• **Creating Static VLANs**	Switch(config)#**vlan 3** Switch(config-vlan)#**name Engineering** Switch(config-vlan)#**exit**
Assigning Ports as access to VLANs	
• **Assigning Ports to VLANs**	Switch(config)#**interface fastethernet 0/1** Switch(config-if)#**switchport mode access** Switch(config-if)#**switchport access vlan 10**
• **Using the range Command**	Switch(config)#**interface range fastethernet 0/1 – 9** Switch(config-if-range)#**switchport mode access** Switch(config-if-range)#**switchport access vlan 10**
VLAN Trunking Protocol	
Puts the interface into permanent trunking mode and negotiates to convert the link into a trunk link.	Switch(config)#**interface fastethernet 0/1** Switch(config-if)#**switchport mode trunk** Switch(config-if)#**switchport trunk native VLAN 99** Switch(config-if)#**switchport trunk allowed VLAN 10,20,30,99**
Verifying VLAN Information and Erasing VLAN	
Verifying VLAN Information ▪ Displays VLAN information ▪ Displays VLAN information in brief ▪ Displays information about VLAN 2 only ▪ Displays information about VLAN ▪ named marketing only ▪ Displays interface characteristics for the specified VLAN ▪ Displays VLAN information for all interfaces	Switch#**show vlan** Switch#**show vlan brief** Switch#**show vlan id 2** Switch#**show vlan name marketing** Switch#**show interfaces vlan *x*** Switch#**show interfaces switchport**
Erasing VLAN Configurations • Removes the entire VLAN database from flash.	Switch#**delete flash:vlan.dat**
• Moves to interface configuration mode. • Removes port from VLAN 5 and reassigns it to VLAN 1—the default VLAN.	Switch(config)#**interface fastethernet 0/5** Switch(config-if)#**no switchport access vlan 5**
• Removes VLAN 5 from the VLAN database.	Switch(config)#**no vlan 5**

Inter-VLAN Communication Using an External Router: Router-on-a-Stick	
Moves to interface configuration mode. Enables the interface.	Router(config)#**interface gigabitethernet 0/0** Router(config-if)#**no shutdown** Router(config-if)#**exit**
▪ Creates subinterface 0/0.10 and moves to subinterface configuration mode. ▪ (Optional) Sets the locally significant description of the subinterface. ▪ Assigns VLAN 10 to this subinterface. This subinterface will use the 802.1q trunking protocol. ▪ Assigns the IP address and netmask.	Router(config-subif)#**interface gigabitethernet 0/0.10** Router(config-subif)#**description HR VLAN 10** Router(config-subif)#**encapsulation dot1q 10** Router(config-subif)#**ip address 192.168.10.1 255.255.255.0** Router(config-subif)# **exit**
For subinterface of Native VLAN ▪ Creates subinterface 0/0.99 and moves to subinterface configuration mode. ▪ (Optional) Sets the locally significant description of the subinterface. ▪ Assigns VLAN 99 to this subinterface. VLAN 99 will be the native VLAN. This subinterface will use the 802.1q trunking protocol. ▪ Assigns the IP address and netmask.	Router(config-if)#**interface gigabitethernet 0/0.99** Router(config-subif)#**description Management VLAN 99** Router(config-subif)#**encapsulation dot1q 99 native** Router(config-subif)#**ip address 192.168.1.1 255.255.255.0** Router(config-subif)# **exit**

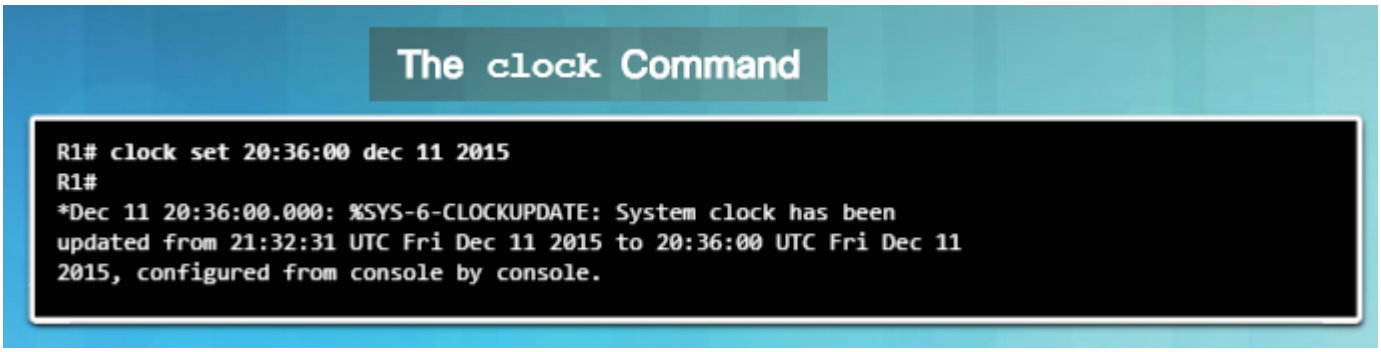
Dynamic Host Configuration Protocol (DHCPv4)

Configuring a DHCP Server on an IOS Router	
▪ Creates a DHCP pool named internal. The name can be anything of your choosing. ▪ Defines the range of addresses to be leased. ▪ Defines the address of the default router for the client. ▪ Defines the address of the Domain Name System (DNS) server for the client ▪ Defines the address of the NetBIOS server for the client. ▪ Defines the domain name for the client. ▪ Returns to global configuration mode.	Router(config)#**ip dhcp pool internal** Router(dhcp-config)#**network 172.16.10.0 255.255.255.0** Router(dhcp-config)#**defaultrouter 172.16.10.1** Router(dhcp-config)#**dns-server 172.16.10.10** Router(dhcp-config)#**domain-name cisco.com** Router(dhcp-config)#**exit**
▪ Specifies the range of addresses not to be leased out to clients.	Router(config)#**ip dhcp excluded-address 172.16.10.1 172.16.10.9**
Configuring a DHCP Helper Address	
▪ Moves to interface configuration mode. ▪ DHCP broadcasts will be forwarded as a unicast to this specific address rather than be dropped by the router.	Router(config)#**interface gigabitethernet 0/0** Router(config-if)#**ip helper-address 172.16.20.2**
DHCP Client on a Cisco IOS Software Ethernet Interface	
▪ Moves to interface configuration mode ▪ Specifies that the interface acquire an IP address through DHCP	Router(config)# **interface gigabitethernet 0/0** Router(config-if)# **ip address dhcp**

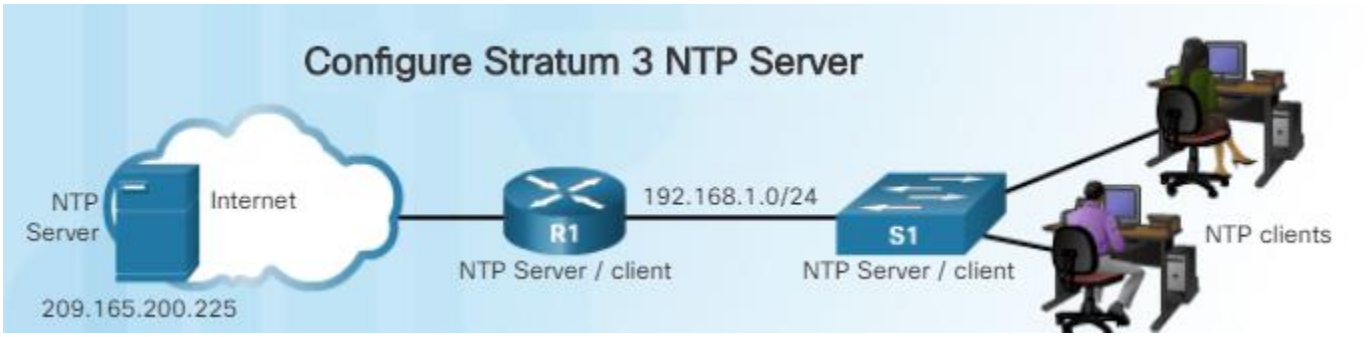
Setting the System Clock

Typically, the date and time settings on a router or switch can be set using one of two methods:

- Manually configure the date and time, as shown in the figure
- Configure the Network Time Protocol (NTP)



Configure and Verify NTP



- **Configure R1 to use an external public NTP server with an IP address of 209.165.200.225.**
  R1# configure terminal
  R1(config)# ntp server 209.165.200.225
  R1(config)# end

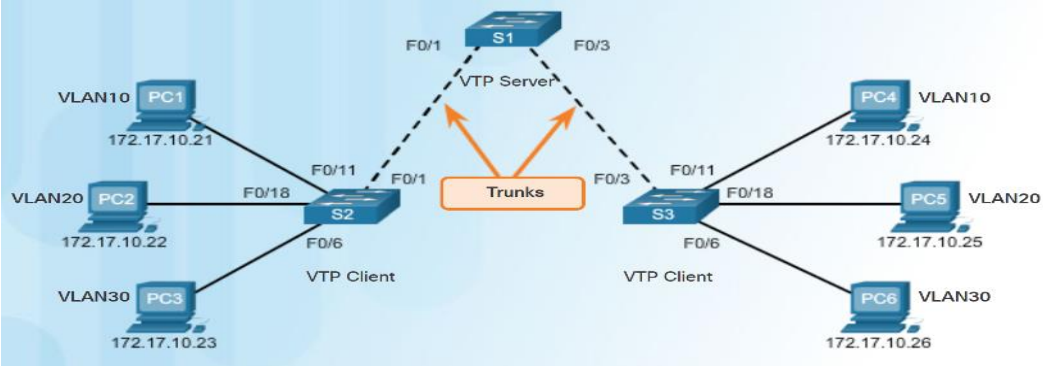
- **Verify that R1 is associated with the NTP server at IP address 209.165.200.225.**
  R1# show ntp associations

- **Verify that R1 is synchronized with the NTP server at IP address 209.165.200.225.**
  R1# show ntp status

VTP, Extended VLANs, and DTP

1. VLAN Trunking Protocol (VTP):

Reduces administration in a switched network. A switch in VTP server mode can manage additions, deletions and renaming of VLANs across the domain.



VTP Configuration	
1. **Configure the VTP Server.**	S1(config)# vtp mode server
2. **Configure the VTP Domain Name and Password.**	S1(config)# vtp domain CCNA S1(config)# vtp password cisco
3. **Configure the VTP Clients.**	S2(config)# vtp mode client S2(config)# vtp domain CCNA S2(config)# vtp password cisco
4. **Configure VLANs on the VTP Server.**	S1(config)# vlan 10 S1(config-vlan)# name Red
5. **Verify the VTP Clients have received the new VLAN information.**	S2# show vtp status S2# show vtp password

2. Extended VLANs

- Extended range VLANs are identified by a VLAN ID between 1006 and 4094.
- To configure an extended VLAN on a 2960 switch it must be set to VTP transparent mode. (By default 2960 switches do not support Extended range VLANs.)

Configuring Extended VLANs	
6. **Configure the VTP** transparent mode**.**	S1(config)# vtp mode transparent
7. Create Extended VLAN	S1(config)# VLAN 2000 S1(config-vlan)# end

Dynamic Trunking Protocol (DTP)

Trunk negotiation is managed by the Dynamic Trunking Protocol (DTP). DTP is a Cisco proprietary protocol

	Dynamic Auto	Dynamic Desirable	Trunk	Access
Dynamic Auto	Access	Trunk	Trunk	Access
Dynamic Desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Limited Connectivity
Access	Access	Access	Limited Connectivity	Access

DTP Configuration	
• **Switchport mode access** - interface becomes a nontrunk interface.	S1(config)# **Switchport mode access**
• **Switchport mode dynamic auto** - interface becomes a trunk if the neighboring interface is set to trunk or desirable mode.	S1(config)# **Switchport mode dynamic auto**
• **Switchport mode dynamic desirable** - interface becomes a trunk if the neighboring interface is set to trunk, desirable, or dynamic auto mode.	S1(config)# **Switchport mode dynamic desirable**
• **Switchport mode trunk** - interface becomes a trunk even if the neighboring interface is not a trunk interface.	S1(config)# **Switchport mode trunk**
• **Switchport nonegotiate** - prevents the interface from generating DTP frames.	S1(config)# **Switchport nonegotiate**
• Use show dtp interface to verify DTP.	S1# **show dtp interface**

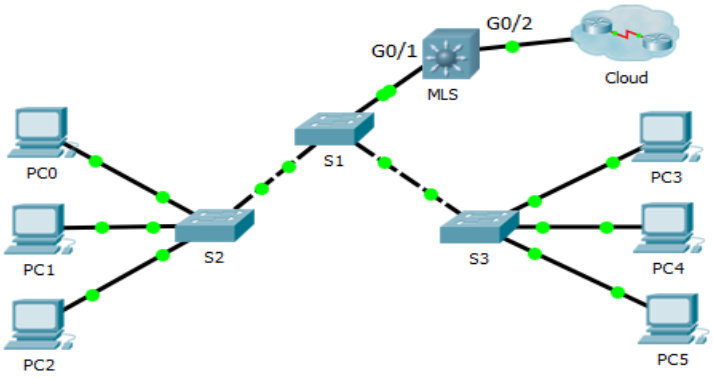
Layer 3 Switching

Multilayer switches provide high-packet processing rates using hardware-based switching

- Catalyst multilayer switches support the following types of Layer 3 interfaces:

  •**Routed port** - A layer 3 interface

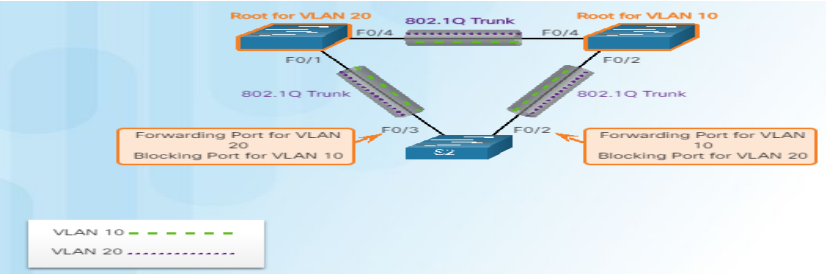
  •**Switch virtual interface (SVI) -** Virtual Interface for inter- VLAN routing



Layer 3 Switching Configuration	
• **Configure G0/2 as a routed port and assign an IP address**	MLS(config)# interface g0/2
	MLS(config-if)# no switchport
	MLS(config-if)# ip address 209.165.200.225 255.255.255.252
• **Configure SVI on MLS.**	MLS(config)# interface vlan 10
	MLS(config-if)# ip address 192.168.10.254 255.255.255.0
	MLS(config)# interface vlan 20
	MLS(config-if)# ip address 192.168.20.254 255.255.255.0
• **Enable routing.**	MLS(config)# ip routing

STP

Spanning Tree Protocol (STP) is a Layer 2 protocol that helps especially when there are redundant links.



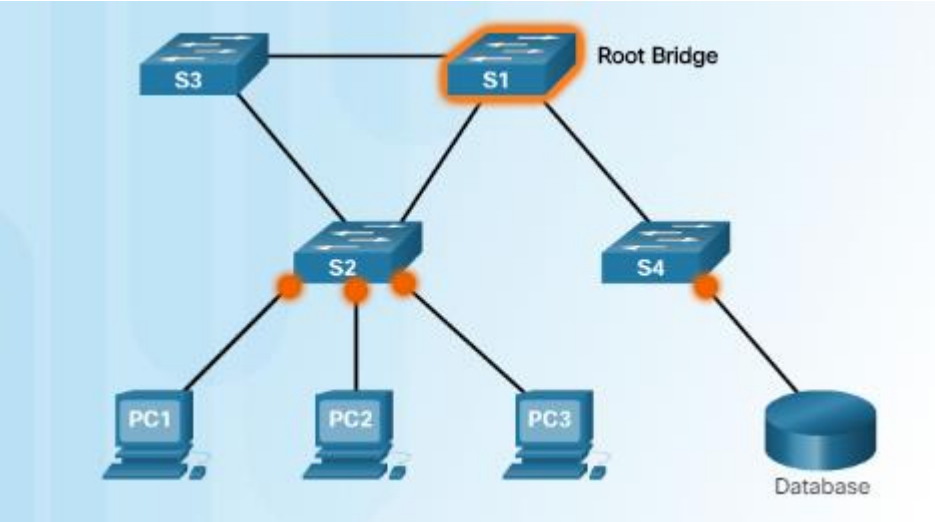
Changing the Spanning-Tree Mode	
Enables PVST. This is the default setting.	Switch(config)#**spanning-tree mode pvst**
Enables Rapid PVST+.	Switch(config)#**spanning-tree mode rapid-pvst**
Configuring the Root primary Switch	
Switch recalculates timers along with priority to allow the switch to become the root switch for VLAN 5.	Switch(config)#**spanning-tree vlan 5 root primary**
Configuring the Root primary Switch	
Switch recalculates timers along with priority to allow the switch to become the root switch for VLAN 5 should the primary root switch fail.	Switch(config)#**spanning-tree vlan 10 root secondary**
Configuring by the Switch Priority	
Configures the switch priority of VLAN 5 to 24576	Switch(config)#spanning-tree vlan 5 priority 24576
Optional STP Configurations	
PortFast	
Enters interface range configuration mode.	Switch(config**)#interface range fastethernet 0/1 – 5**
Enables PortFast on an access port.	Switch(config-if)#**spanning-tree portfast**
BPDU Guard	
Enters interface range configuration mode.	Switch(config**)#interface range fastethernet 0/1 – 5**
Enables BPDU Guard on the interface.	Switch(config-if)#**spanning-tree bpduguard enable**

STP

Verifying STP	
Displays STP information	Switch#**show spanning-tree**
Displays STP information on active interfaces only	Switch#**show spanning-tree active**
Displays a brief status of the STP	Switch#**show spanning-tree brief**
Displays a detailed summary of interface information	Switch#**show spanning-tree detail**
Displays STP information for interface gigabitethernet 0/1	Switch#**show spanning-tree interface gigabitethernet 0/1**
Displays a summary of port states	Switch#**show spanning-tree summary**

- **CAUTION** Cisco recommends caution when using this command. Cisco further recommends that the **spanning-tree vlan** *x* **root primary** or the **spanning-tree vlan** *x* **root secondary** command be used instead to modify the switch priority.

Edge Ports



- PortFast is used on ports that have end devices attached.
  - Puts a port in the forwarding state
  - Allows DHCP to work properly
- BPDU Guard disables a port that has PortFast configured on it if a BPDU is received

EtherChannel

- **EtherChannel:**
  - EtherChannel groups multiple physical ports into one or more logical EtherChannel links.

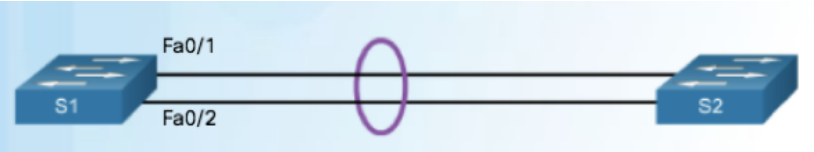
Link Aggregation Protocols:

1- Port Aggregation Protocol (PAgP)

Cisco-proprietary protocol

2- Link Aggregation Control Protocol (LACP)

Is part of IEEE (IEEE.3AD)



Interface Modes in EtherChannel

Mode	Protocol	Description
On	None	Forces the interface into an EtherChannel without PAgP or LACP. Channel only exists if connected to another interface group also in On mode.
Auto	PAgP	Places the interface into a passive negotiating state—will respond to PAgP packets but will not initiate PAgP negotiation.
Desirable	PAgP	Places the interface into an active negotiating state—will send PAgP packets to start negotiations.
Passive	LACP	Places the interface into a passive negotiating state—will respond to LACP packets but will not initiate LACP negotiation.
Active	LACP	Places the interface into an active negotiating state—will send LACP packets to start negotiations.

- To create a channel in PAgP, sides must be set to
  - Auto-Desirable
  - Desirable-Desirable
- To create a channel in LACP, sides must be set to
  - Active-Active
  - Active-Passive

- This configuration creates EtherChannel with LACP and configures trunking.
  - Step 1: Specify the interfaces that compose the EtherChannel group.
  - Step 2: Create the port channel interface with the **channel-group** command in **active** mode. (Channel group number needs to be selected.)
  - Step 3: Change Layer 2 settings in port channel interface configuration mode.

EtherChannel with LACP and configures	
• Places the interface into an active negotiating state—will send LACP packets to start negotiations. • Configure Port Channel	• S1(config)#interface range fastethernet 0/1 – 2 S1 (config-if)#channel-group 1 mode Active S1 (config-if)#exit • S1(config)#interface port-channel 1 S1(config-if)# Switchport mode Trunk S1(config-if)# switchport trunk native vlan 999 S1(config-if)# switchport trunk allowed vlan 1,10,20 S1 (config-if)#exit
• Places the interface into a passive negotiating state—will respond to LACP packets but will not initiate LACP negotiation. • Configure Port Channel	• S1(config)#interface range fastethernet 0/1 – 2 S1 (config-if)#channel-group 1 mode passive S1 (config-if)#exit • S1(config)#interface port-channel 1 S1(config-if)# Switchport mode Trunk S1(config-if)# switchport trunk native vlan 999 S1(config-if)# switchport trunk allowed vlan 1,10,20 S1 (config-if)#exit
EtherChannel with PAgP and configures	
• Places the interface into an active negotiating state—will send PAgP packets to start negotiations. • Configure Port Channel	• S1(config)#interface range fastethernet 0/1 – 2 S1 (config-if)#channel-group 1 mode Desirable S1 (config-if)#exit • S1(config)#interface port-channel 1 S1(config-if)# Switchport mode Trunk S1(config-if)# switchport trunk native vlan 999 S1(config-if)# switchport trunk allowed vlan 1,10,20
• Places the interface into a passive negotiating state—will respond to PAgP packets but will not initiate PAgP negotiation. • Configure Port Channel	• S1(config)#interface range fastethernet 0/1 – 2 S1 (config-if)#channel-group 1 mode Auto S1 (config-if)#exit • S1(config)#interface port-channel 1 S1(config-if)# Switchport mode Trunk S1(config-if)# switchport trunk native vlan 999 S1(config-if)# switchport trunk allowed vlan 1,10,20

Verifying EtherChannel	
Displays all EtherChannel information	Switch#show etherchannel
Displays port channel information	Switch#show etherchannel 1 portchannel
Displays a summary of EtherChannel information	Switch#show etherchannel summary
Displays the general status of EtherChannel 1	Switch#show interface port-channel 1
Shows PAgP neighbor information	Switch#show pagp neighbor

HSRP

First Hop Redundancy Protocols

- To prevent a single point of failure at the default gateway, implement a virtual router.
- **First Hop Redundancy Protocols:**
  - Hot Standby Router Protocol (HSRP)
  - Virtual Router Redundancy Protocol (VRRP)
  - Gateway Load Balancing Protocol (GLBP)

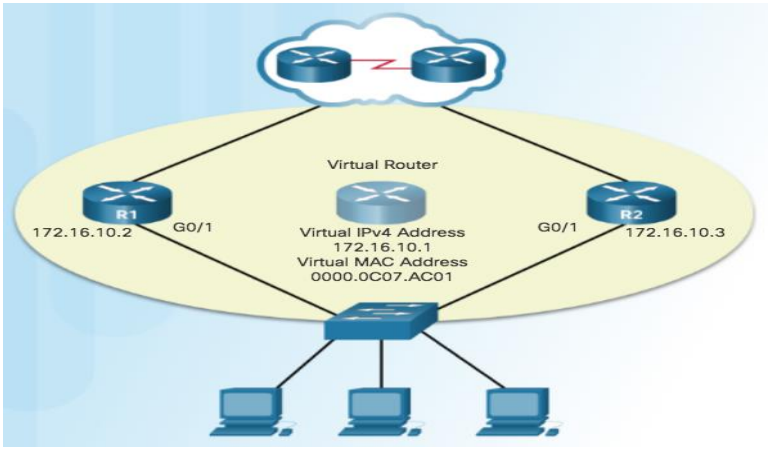
HSRP Configuration Commands

Step 1. Configure HSRP version 2.

Step 2. Configure the virtual IP address for the group.

Step 3. Configure the priority for the desired active router to be greater than 100.

Step 4. Configure the active router to preempt the standby router in cases where the active router comes online after the standby router.

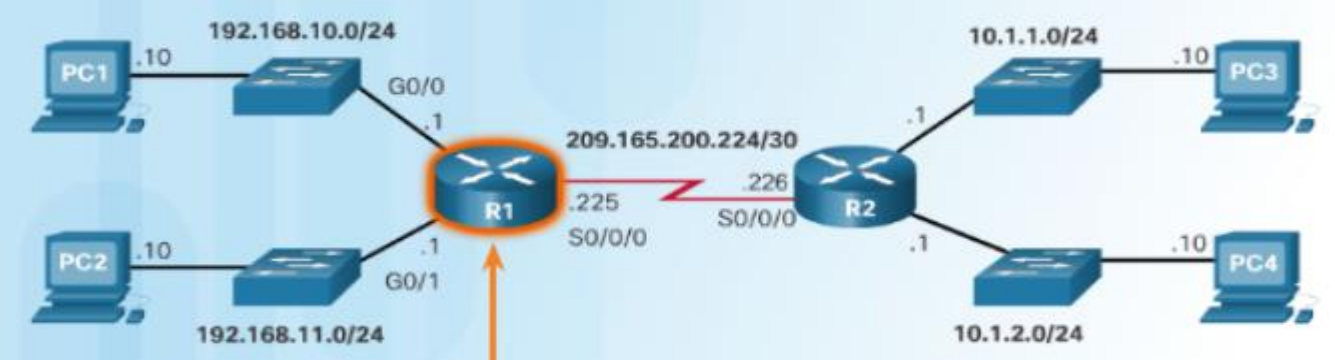



```
R1(config)# interface g0/1
R1(config-if)# ip address 172.16.10.2 255.255.255.0
R1(config-if)# standby version 2
R1(config-if)# standby 1 ip 172.16.10.1
R1(config-if)# standby 1 priority 150
R1(config-if)# standby 1 preempt
R1(config-if)# no shutdown
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R2(config)# interface g0/1
R2(config-if)# ip address 172.16.10.3 255.255.255.0
R2(config-if)# standby version 2
R2(config-if)# standby 1 ip 172.16.10.1
R2(config-if)# no shutdown
```

Static Route

There are two common types of static routes in the routing table:

- Static route to a specific network
- Default static route



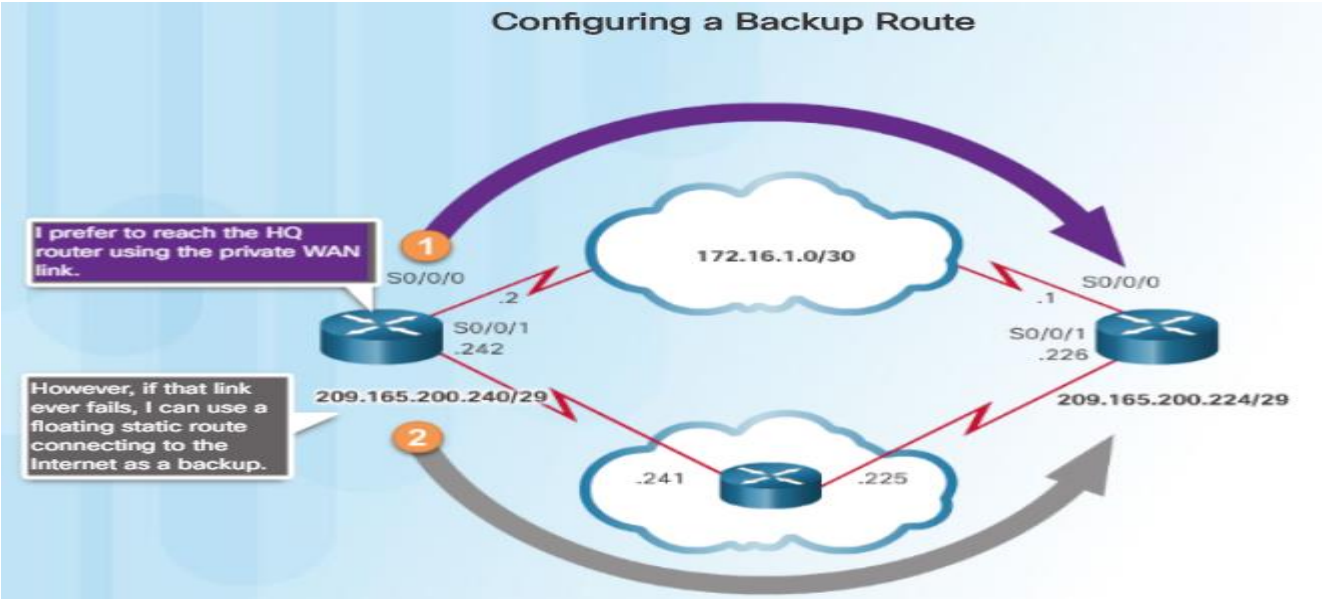
Configuring a Static Route on a Router

When using the **ip route** command, you can identify where packets should be routed in two ways:

- The next-hop address
- The exit interface

Configuring a Static Route on a Router	
10.1.1.0 = destination network. 255.255.255.0 = subnet mask. 209.165.200.226 = next-hop address. Read this to say, "To get to the destination network of 10.1.1.0, with a subnet mask of 255.255.255.0, send all packets to 209.165.200.224."	R1(config)#**ip route** **10.1.1.0** **255.255.255.0** **209.165.200.226**
10.1.1.0 = destination network. 255.255.255.0 = subnet mask. 209.165.200.226 = next-hop address. Read this to say, "To get to the destination network of 10.1.1.0, with a subnet mask of 255.255.255.0, send all packets out interface serial 0/0/0."	R1(config)#**ip route** **10.1.1.0** **255.255.255.0 serial 0/0/0**
Configuring a Default Route on a Router	
Send all packets destined for networks not in my routing table to next hop IP **209.165.200.226**.	Router(config)# **ip route 0.0.0.0 0.0.0.0** **209.165.200.226**
Send all packets destined for networks not in my routing table out my exit Interface serial 0/0 interface.	Router(config)# **ip route 0.0.0.0 0.0.0.0** **serial 0/0/0**

Floating Static Route



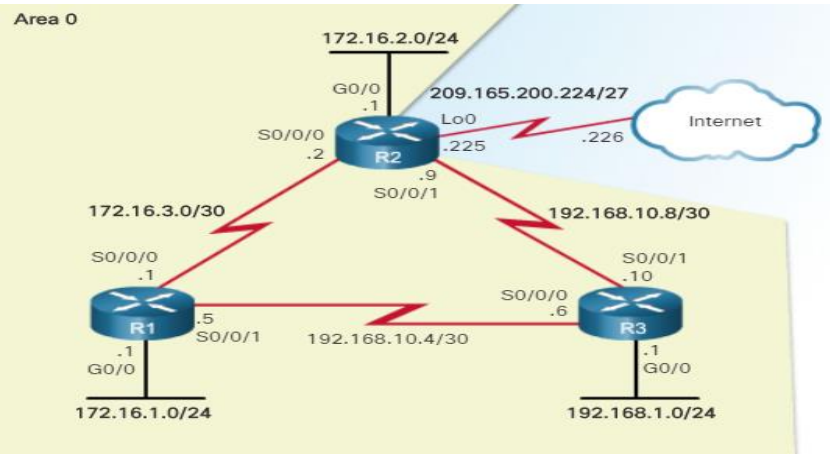
Floating Static Route	
Configured a Primary static route. Because no administrative distance is configured, the default value (1) is used for this static route.	**Router(config)#ip route 209.165.200.224 255.255.255.248 S0/0/0**
Configured a floating static route with an administrative distance of 5	**Router(config)#ip route 209.165.200.224 255.255.255.248 S0/0/1 5**

Verifying Static Routes	
To display the contents of the IP routing table, enter the following command:	• Router# **show ip route** • Router# **show ip route static** • Router# **show ip route** *networ*

Dynamic Routing

Open Shortest Path First (OSPF)

1. Single-Area OSPF

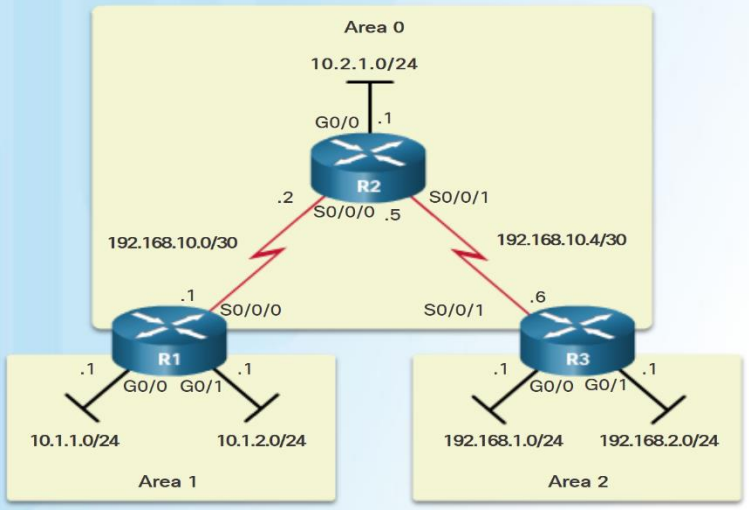


Configuring single Area OSPF	
Starts OSPF process 100. The process ID is any positive integer value between 1 and 65,535. The process ID *is not related to* the OSPF area	R2(config)#**router ospf 100**
Router ID Sets the router ID to 10.1.1.1. If this command is used on an OSPF router process that is already active (has neighbors), the new router ID is used at the next reload or at a manual OSPF process restart.	Router(config-router)# **router-id 10.1.1.1**
• OSPF advertises interfaces, not networks. Uses the wildcard mask to determine which interfaces to advertise. OR • Read this line to say "Any interface with an exact address of 172.168.10.9, 172.16.3.1 and 172.16.2.2 is to be put into area 0."	R2(config-router)# **network 172.16.2.0 0.0.0.255 area 0** R2(config-router)# **network 172.16.3.0 0.0.0.3 area 0** R2(config-router)# **network 192.168.10.8 0.0.0.3 area 0** OR R2(config-router)# **network 172.16.2.2 0.0.0.0 area 0**

	R2(config-router)# **network 172.16.3.1 0.0.0.0 area 0** R2(config-router)# **network 192.168.10.9 0.0.0.0 area 0**
Passive Interfaces 1. Disables the sending of routing updates on this interface. 2. Disables the sending of routing updates out all interfaces.Enables routing updates to be sent out interface serial 0/0/1, thereby allowing neighbor adjacencies to form.	Router(configrouter)# **passive-interface fastethernet 0/0** OR Router(configrouter)# **passive-interface default** Router(config-router)#**no passive-interface serial 0/0/1**
OSPF auto-cost reference-bandwidth Changes the reference bandwidth that OSPF uses to calculate the cost of an interface. Accurate to 1 Gbps	Router(config-router)# **auto-cost reference-bandwidth 1000**
Propagating a Default Route Creates a default route. Starts OSPF process 100. Sets the default route to be propagated to all OSPF routers.	Router(config)#**ip route 0.0.0.0 0.0.0.0 s0/0/0** Router(config)#**router ospf 100** Router(config-router)#**defaultinformation originate**
Timers Changes the Hello Interval timer to 20 seconds. Changes the Dead Interval timer to 80 seconds. **NOTE** Hello and Dead Interval timers must match for routers to become neighbors.	Router(config)# interface S0/0/0 Router(config-if)#**ip ospf hello-interval timer 20** Router(config-if)#**ip ospf dead-interval 80**
Modifying Cost Metrics Changes the router to interface configuration mode. ▪ If you change the bandwidth, OSPF recalculates the cost of the link. Or ▪ Changes the cost to a value of 1564.	Router(config)#**interface serial 0/0/0** Router(configif)# **bandwidth 128** Router(config-if)#**ip ospf cost 1564**

NOTE The cost of a link is determined by dividing the reference bandwidth by the interface bandwidth. The bandwidth of the interface is a number between 1 and 10,000,000. The unit of measurement is kilobits. The cost is a number between 1 and 65,535. The cost has no unit of measurement—it is just a number.	

Multi-Area OSPF



Configuring Multi-Area OSPF	
Starts OSPF process 1. The process ID is any positive integer value between 1 and 65,535. The process ID is *not* related to the OSPF area.	R1(config)# **router ospf 1**
Read this line to say "Any interface with an address of 172.16.10. *x* is to be put into area 0."	Router(config-router)# **network** 192.168.10.0 0.0.0.3 **area** 0
• Uses the wildcard mask to determine which interfaces to advertise. OR • Read this line to say "Any interface with an exact	Router(config-router)# **network** 10.1.1.0 0.0.0.255 **area** 1 Router(config-router)# **network** 10.1.2.0 0.0.0.255 **area** 1 OR Router(config-router)# **network** 10.1.1.1 0.0.0.0 **area** 1

address of 10.1.1.1 and 10.1.2.1 is to be put into area 1.”	Router(config-router)# network 10.1.2.1 0.0.0.0 area 1
▪ Commands to verify multiarea OSPFv2	• show ip ospf neighbor • show ip ospf • show ip ospf interface • Show ip protocols • show ip ospf interface brief • show ip route ospf • show ip ospf database

Device Discovery with CDP

- For Cisco devices, CDP is enabled by default.

• Device Discovery with CDP	
To enable CDP globally for all the supported interfaces on the device.	R1(config)# **cdp run**
To enable CDP on the specific interface again,	R1(config)# **interface g 0/1** R1(config-if)# **cdp enable**
• Use the **show** command to display the interfaces that are CDP-enabled on a device. • Use the **show** command can be used to determine the network layout	R1# **show cdp interface** R1# **show cdp neighbors** R1# **show cdp neighbors detail**

Device Discovery with LLDP

- Link Layer Discovery Protocol (LLDP) is a vendor-neutral neighbor discovery protocol similar to CDP.

• Device Discovery with CDP	
• To enable LLDP globally for all the supported interfaces on the device, command in the global configuration mode.	Switch(config)# **lldp run**
• To enable CDP on the specific interface again,	Switch(config)# **interface gigabitethernet 0/1** Switch(config-if)# **lldp transmit** Switch(config-if)# **lldp receive**
• With LLDP enabled, device neighbors can be discovered	S1# **show lldp neighbors** S1# **show lldp neighbors detail**

Router and Switch File Maintenance

Using TFTP to Back Up and Restore a Configuration

Follow these steps to back up the running configuration to a TFTP server:

Step 1. Enter the **copy running-config tftp** command.
Step 2. Enter the IP address of the host where the configuration file will be stored.
Step 3. Enter the name to assign to the configuration file.
Step 4. Press Enter to confirm each choice.

Use the following steps to restore the running configuration from a TFTP server:

Step 1. Enter the **copy tftp running-config** command.
Step 2. Enter the IP address of the host where the configuration file is stored.
Step 3. Enter the name to assign to the configuration file.
Step 4. Press **Enter** to confirm each choice.

```
R1# copy running-config tftp
Remote host []?192.168.10.254
Name of the configuration file to write[R1-config]? R1-Jan-2019
Write file R1-Jan-2019 to 192.168.10.254? [confirm]
Writing R1-Jan-2019 !!!!!! [OK]
```

Using USB to Back Up and Restore a Configuration

```
R1# copy running-config usbflash0:
Destination filename [running-config]? R1-Config
%Warning:There is a file already existing with this name
Do you want to over write? [confirm]
5024 bytes copied in 1.796 secs (2797 bytes/sec)
R1#
```

Password Recovery Procedures

Passwords on devices are used to prevent unauthorized access. For encrypted passwords, such as the enable secret passwords, the passwords must be replaced after recovery. Depending on the device, the detailed procedure for password recovery varies.

However, all the password recovery procedures follow the same principle:

Step 1. Enter the ROMMON mode. With console access, a user can access the ROMMON mode by using a **break sequence** during the boot up process or removing the external flash memory when the device is powered off.

When successful, the **rommon 1 >** prompt displays, as shown in the example.

Step 2. Change the configuration register. The **confreg 0x2142** command allows the user to set the configuration register to 0x2142, which causes the device to ignore the startup config file during startup.

Step 3. Copy the startup-config to the running-config.

CAUTION: Do not enter **copy running-config startup-config**. This command erases your original startup configuration.

Step 4. Change the password.

Step 5. Save the running-config as the new startup-config. After the new passwords are configured, change the configuration register back to 0x2102

Step 6. Reload the device.

```
Readonly ROMMON initialized
monitor: command "boot" aborted due to user interrupt
rommon 1 >
rommon 1 > confreg 0x2142
rommon 2 > reset
System Bootstrap, Version 15.0(1r)M9, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2010 by cisco Systems, Inc.
(output omitted)
Router# copy startup-config running-config
Destination filename [running-config]?
1450 bytes copied in 0.156 secs (9295 bytes/sec)
R1#
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# enable secret cisco

R1(config)# config-register 0x2102
R1(config)# end
R1# copy running-config startup-config
Destination filename [startup-config]? Building configuration...
[OK]
R1#
R1#Reload
```

IOS Image Management

TFTP Servers as a Backup Location

```
R1# copy flash: tftp:
Source filename []? isr4200-universalk9_ias.16.09.04.SPA.bin
Address or name of remote host []? 172.16.1.100
Destination filename [isr4200-universalk9_ias.16.09.04.SPA.bin]?
Writing isr4200-universalk9_ias.16.09.04.SPA.bin...
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
(output omitted)
517153193 bytes copied in 863.468 secs (269058 bytes/sec)
```

Copy an IOS Image to a Device Example

```
R1# copy tftp: flash:
Address or name of remote host []? 2001:DB8:CAFE:100::99
Source filename []? isr4200-universalk9_ias.16.09.04.SPA.bin
Destination filename [isr4200-universalk9_ias.16.09.04.SPA.bin]?
Accessing tftp://2001:DB8:CAFE:100::99/ isr4200- universalk9_ias.16.09.04.SPA.bin...
Loading isr4200-universalk9_ias.16.09.04.SPA.bin from 2001:DB8:CAFE:100::99 (via
GigabitEthernet0/0/0): !!!!!!!!!!!!!!!!!!!

[OK - 517153193 bytes]
517153193 bytes copied in 868.128 secs (265652 bytes/sec)
```

The boot system Command

```
R1# configure terminal
R1(config)# boot system flash0:isr4200-universalk9_ias.16.09.04.SPA.bin
R1(config)# exit
R1# copy running-config startup-config
R1# reload
```

ACLs for IPv4 Configuration

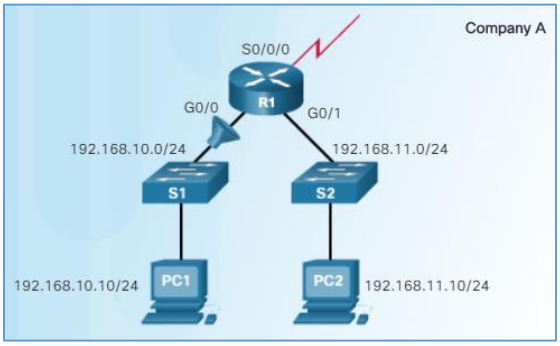
1.Standard IPv4 ACL

a. Numbered Standard IPv4 ACL Syntax

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

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

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



Apply a Standard IPv4 ACL

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

Numbered Standard ACL Example

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

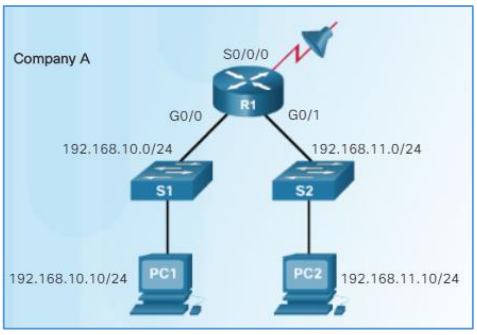
```
R1(config)# access-list 10 remark ACE permits ONLY host 192.168.10.10 to the internet
R1(config)# access-list 10 permit host 192.168.10.10
R1(config)# do show access-lists
Standard IP access list 10
    10 permit 192.168.10.10
R1(config)#
```

```
R1(config)# access-list 10 remark ACE permits all host in LAN 2
R1(config)# access-list 10 permit 192.168.20.0 0.0.0.255
R1(config)# do show access-lists
Standard IP access list 10
    10 permit 192.168.10.10
    20 permit 192.168.20.0, wildcard bits 0.0.0.255
R1(config)#
```

```
R1(config)# interface Serial 0/1/0
R1(config-if)# ip access-group 10 out
R1(config-if)# end
R1#
```

b. Named Standard ACL

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




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

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

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

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

• Secure VTY Ports with a Standard IPv4 ACL

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

Modify IPv4 ACLs

There are two methods to use when modifying an ACL:

1. Use a text editor.
    - To correct an error in an ACL:
    - Copy the ACL from the running configuration and paste it into the text editor.
    - Make the necessary edits or changes.
    - Remove the previously configured ACL on the router.
    - Copy and paste the edited ACL back to the router.
2. Use sequence numbers.

An ACL ACE can be deleted or added using the ACL sequence numbers.

- Use the **ip access-list standard** command to edit an ACL.
- Statements cannot be overwritten using an existing sequence number. The current statement must be deleted first with the **no 10** command. Then the correct ACE can be added using sequence number.

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

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

Named ACLs can also use sequence numbers to delete and add ACEs. In the example an ACE is added to deny hosts 192.168.10.11.

```
R1# show access-lists
Standard IP access list NO-ACCESS
    10 deny   192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
```

```
R1# configure terminal
R1(config)# ip access-list standard NO-ACCESS
R1(config-std-nacl)# 15 deny 192.168.10.5
R1(config-std-nacl)# end
R1#
R1# show access-lists
Standard IP access list NO-ACCESS
    15 deny   192.168.10.5
    10 deny   192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Extended IPv4 ACLs

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

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

Protocols and Port Numbers Configuration Examples

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

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

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

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

Apply a Numbered Extended IPv4 ACL

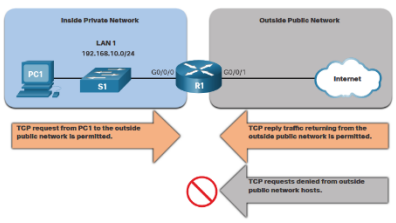
- In this example, the ACL permits both HTTP and HTTPS traffic from the 192.168.10.0 network to go to any destination.
- Extended ACLs can be applied in various locations. However, they are commonly applied close to the source. Here ACL 110 is applied inbound on the R1 G0/0/0 interface.

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

TCP Established Extended ACL

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

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




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

Named Extended IPv4 ACL

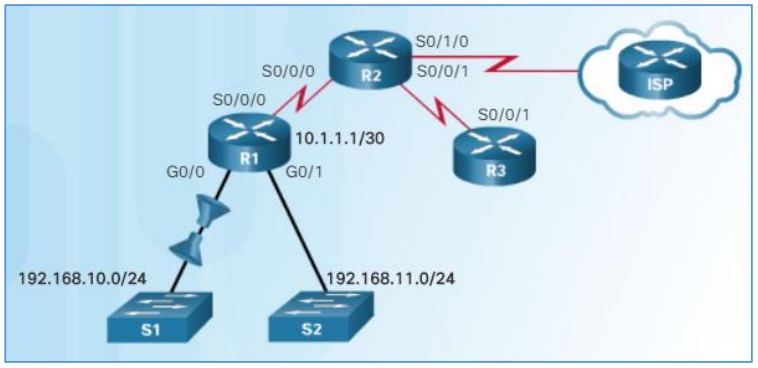
Naming an ACL makes it easier to understand its function. To create a named extended ACL, use the **ip access-list extended** configuration command.

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

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

For example:

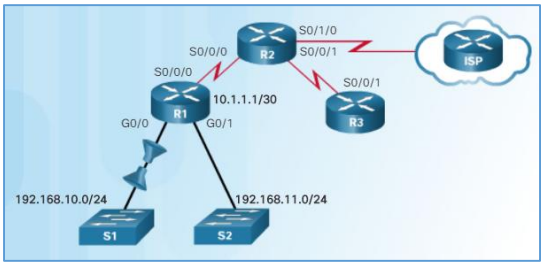
ACL 103 allows requests to port 80 and 443.
ACL 104 allows established HTTP and HTTPS replies.
The **established** parameter allows only responses to traffic that originates from the 192.168.10.0/24 network to return to that network.




```
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
```

		deny		source net. IP	source-wildcard	destination net. IP	destination-wildcard		
		or		or		or			port-name
access-list	ACL-#	permit	protocol	Host	source IP	Host	destination IP	eq	or
		or		or		or			port-no.
		remark		any		any			

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



- The **show ip interface** command is used to verify the ACL on the interface and the direction in which it was applied.
- The **show access-lists** command can be used to confirm that the ACLs work as expected. The command displays statistic counters that increase whenever an ACE is matched.
  Note: Traffic must be generated to verify the operation of the ACL.

Network Address Translation (NAT)

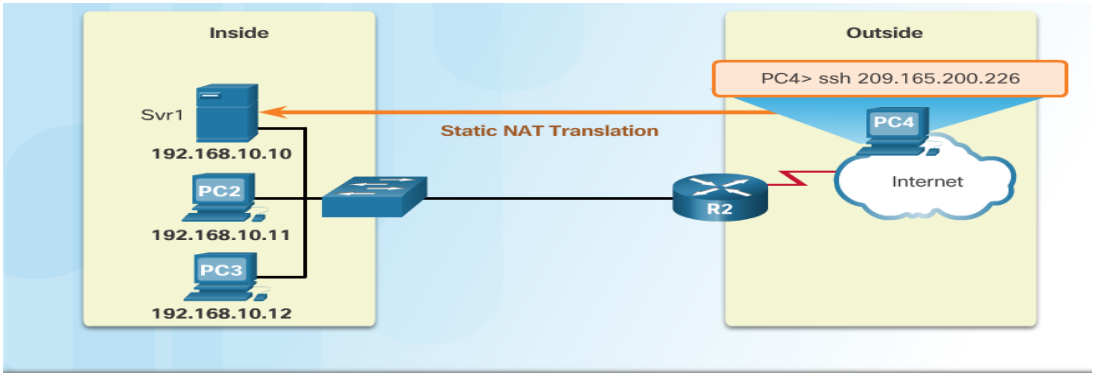
- NAT is used to translate private IP addresses used inside a company to public addresses that can be routed over the Internet.

Private Addresses

Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0–10.255.255.255	10.0.0.0/8
B	172.16.0.0–172.31.255.255	172.16.0.0/12
C	192.168.0.0–192.168.255.255	192.168.0.0/16

1. Static NAT

Static address translation (static NAT) assigns one public IP address to one private IP address



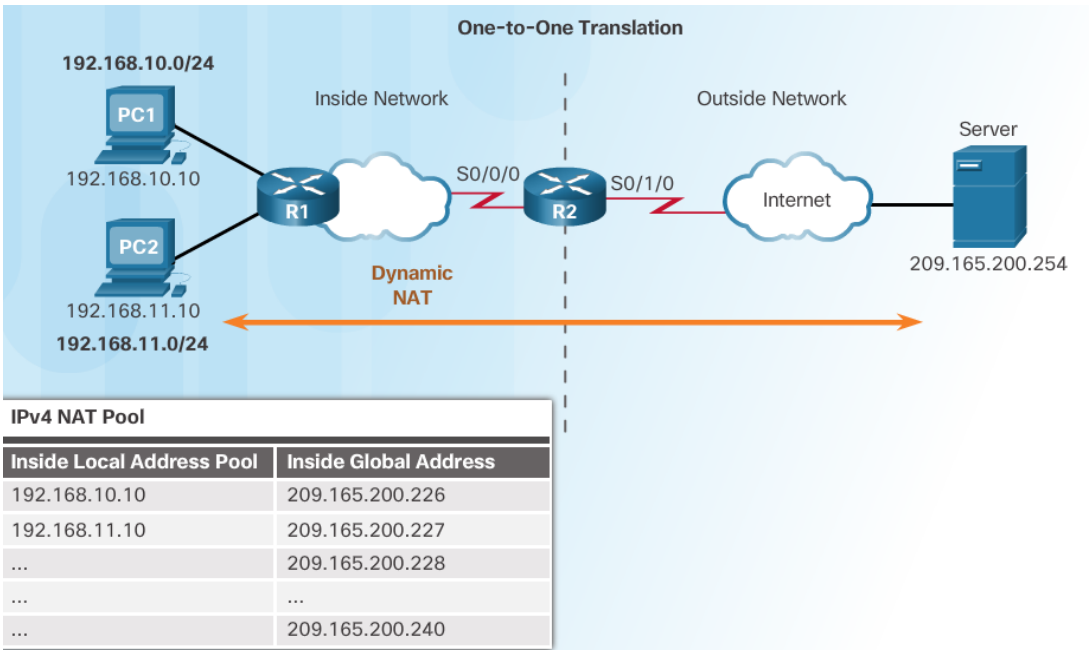
Static NAT Table

Inside Local Address	Inside Global Address – Addresses reachable via R2
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228

Configuring Static NAT: One Private to One Permanent Public Address Translation	
▪ Permanently translates the inside address of **192.168.10.10** to a public address of **209.165.200.226**. ▪ Use the command for each of the private IP addresses you want to statically map to a public address.	R2(config)#**ip nat inside source static 192.168.10.10 209.165.200.226**
▪ Define which interfaces are inside (contain the private addresses).	R2(config)#**interface gigabitethernet 0/0** R2(config-if)#**ip nat inside**
▪ Define the outside interface (the interface leading to the public network).	R2(config)#**interface serial 0/0/0** R2(config-if)#**ip nat outside**

2. **Dynamic NAT** –

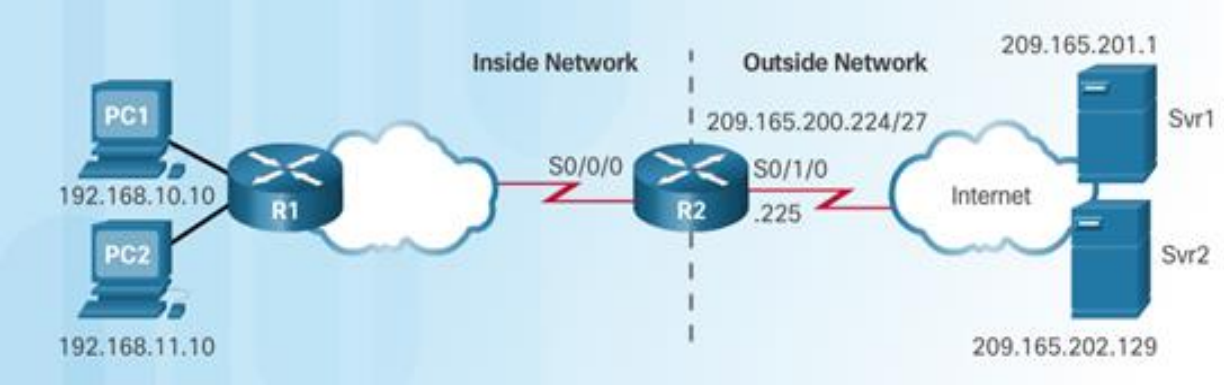
Dynamic NAT assigns a public IP address from a pool of addresses to each packet that originates from a device that has a private IP address assigned when that packet is destined to a network outside the company.



Configuring Dynamic NAT	
▪ Defines the following: The name of the pool is scott. (The name of the pool can be anything.) The start of the pool is 209.165.200.226. The end of the pool is 209.165.200.240. The subnet mask is 255.255.255.224.	R2(config)#**ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224**
▪ Create an access control list (ACL) that will identify which private IP addresses will be translated.	R2(config)#**access-list 1 permit 192.168.0.0 0.0.255.255**
▪ Link the ACL to the pool of addresses (create the translation).	R2(config)#**ip nat inside source list 1 pool NAT-POOL1**
▪ Define which interfaces are inside (contain the private addresses).	R2(config)#**interface gigabitethernet 0/0** R2(config-if)#**ip nat inside**
▪ Define the outside interface (the interface leading to the public network).	R2(config)#**interface Serial 0/0/0** R2(config-if)#**ip nat outside**

3. Port Address Translation (PAT)

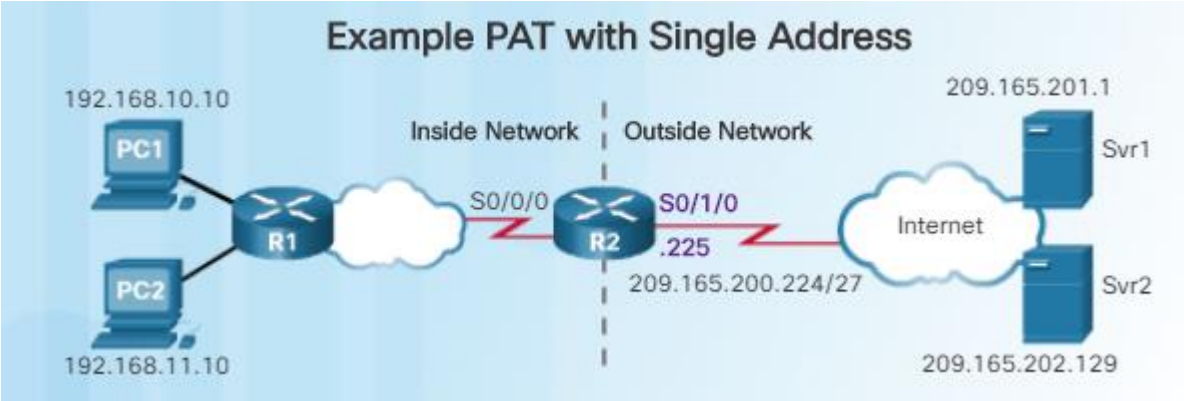
PAT (otherwise known as NAT overload) can use one public IPv4 address to allow thousands of private IPv4 addresses to communicate with outside network devices.



a. Configuring PAT: Address Pool

Configuring PAT: Address Pool	
▪ Defines the following: The name of the pool is scott. (The name of the pool can be anything.) The start of the pool is 209.165.200.226. The end of the pool is 209.165.200.240. The subnet mask is 255.255.255.224.	R2(config)#**ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224**
▪ Create an access control list (ACL) that will identify which private IP addresses will be translated.	R2(config)#**access-list 1 permit 192.168.0.0 0.0.255.255**
▪ Link the ACL to the pool of addresses (create the translation).	R2(config)#**ip nat inside source list 1 pool NAT-POOL1 Overload**
▪ Define which interfaces are inside (contain the private addresses).	R2(config)#**interface gigabitethernet 0/0** R2(config-if)#**ip nat inside**
▪ Define the outside interface (the interface leading to the public network).	R2(config)#**interface Serial 0/0/0** R2(config-if)#**ip nat outside**

b. **Configuring PAT: Single Pool**



Configuring PAT: Single Address	
▪ Defines the following: The name of the pool is scott. (The name of the pool can be anything.) The start of the pool is 209.165.200.226. The end of the pool is 209.165.200.240. The subnet mask is 255.255.255.224.	R2(config)#**ip nat pool NAT-POOL1 209.165.200.225 209.165.200.225 netmask 255.255.255.224**
▪ Create an access control list (ACL) that will identify which private IP addresses will be translated.	R2(config)#**access-list 1 permit 192.168.0.0 0.0.255.255**
▪ Link the ACL to the pool of addresses (create the translation).	R2(config)#**ip nat inside source list 1 pool NAT-POOL1 Overload**
▪ Define which interfaces are inside (contain the private addresses).	R2(config)#**interface gigabitethernet 0/0** R2(config-if)#**ip nat inside**
▪ Define the outside interface (the interface leading to the public network).	R2(config)#**interface Serial 0/0/0** R2(config-if)#**ip nat outside**

OR

Configuring PAT: Single Address	
▪ Create an access control list (ACL) that will identify which private IP addresses will be translated.	R2(config)#**access-list 1 permit 192.168.0.0 0.0.255.255**
▪ Link the ACL to the pool of addresses (create the translation).	R2(config)#**ip nat inside source list 1 interface** serial 0/1/0 **Overload**
▪ Define which interfaces are inside (contain the private addresses).	R2(config)#**interface gigabitethernet 0/0** R2(config-if)#**ip nat inside**
▪ Define the outside interface (the interface leading to the public network).	R2(config)#**interface Serial 0/0/0** R2(config-if)#**ip nat outside**