Software Defined Networking Journal

Name: **Anas Ansari**
Class: **MSc Computer Science**
Roll No: **CS22001**

•

INDEX

Index No	Practical	Date
0	Setup of GNS3	14-09-2022
1	IP SLA	23-09-2022
2	Implement IPv4 ACLs 1. Standard 2. Extended	24-09-2022
3	1. Implement a GRE Tunnel 2. Implement VTP 3. Implement NAT	29-10-2022
4	Implement Inter-VLAN Routing	30-09-2022
5	Observe STP Topology Changes and Implement RSTP	20-10-2022
6	1. Implement EtherChannel 2. Tune and Optimize EtherChannel Operations	29-10-2022
7	1. Implement Single-Area OSPFv2 2. Implement Multi-Area OSPFv2 3. OSPFv2 Route Summarization and Filtering 4. Implement Multiarea OSPFv3	15-09-2022
8	Implement BGP Communities 1. Implement eBGP for Ipv4(only perform part 1 and part 2 from doc) 2.Implement MP(Multi-protocol) -BGP (meaning it supports multiple protocol eg: IPv6,ospf etc)-part 1& 2 3. Implement BGP path Manipulation	8-11-2022
9	Implement IP Sec Site -to -Site VPNs	11-11-2022

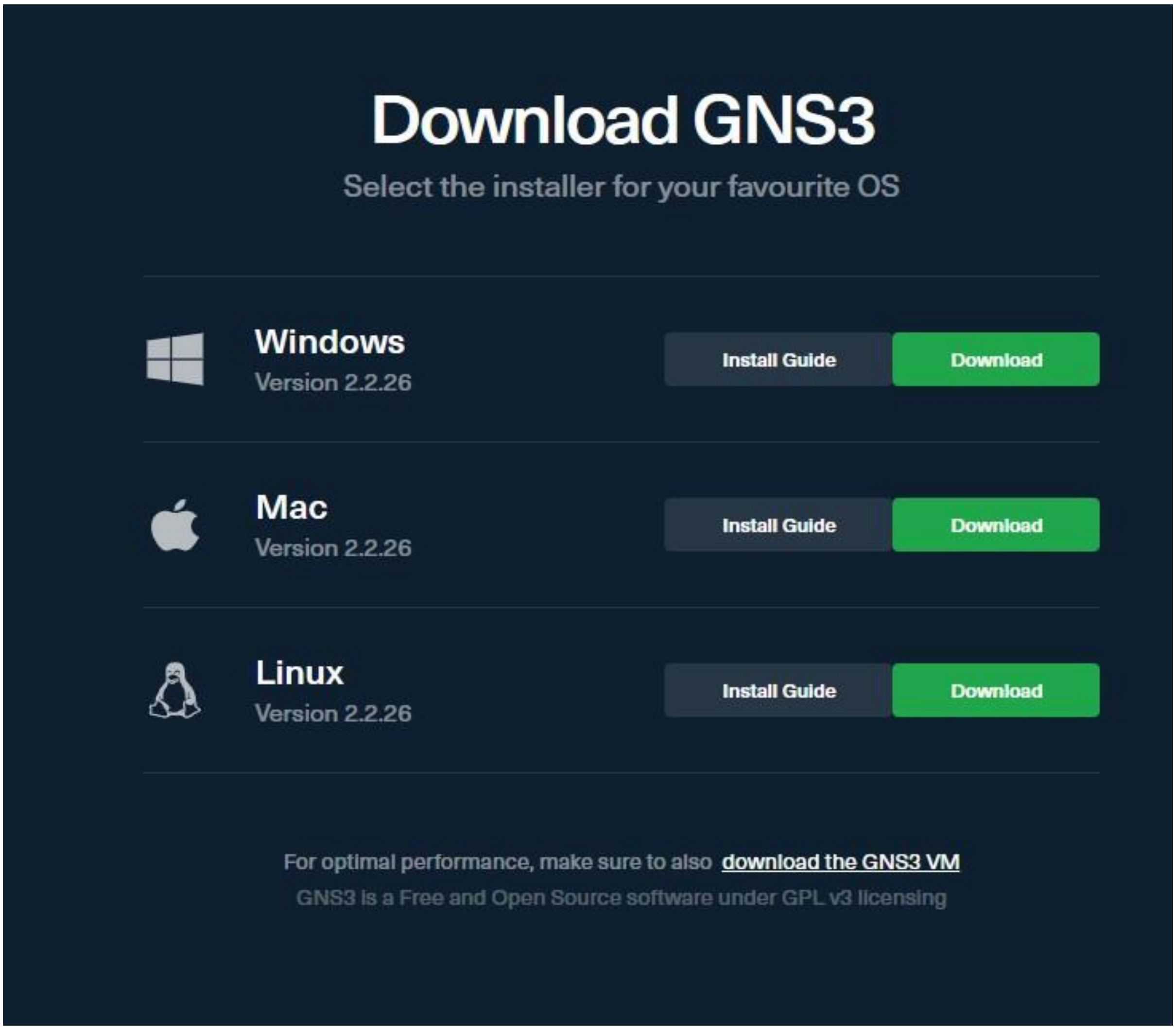
•

Practical 0:

Step 1: Download from the link

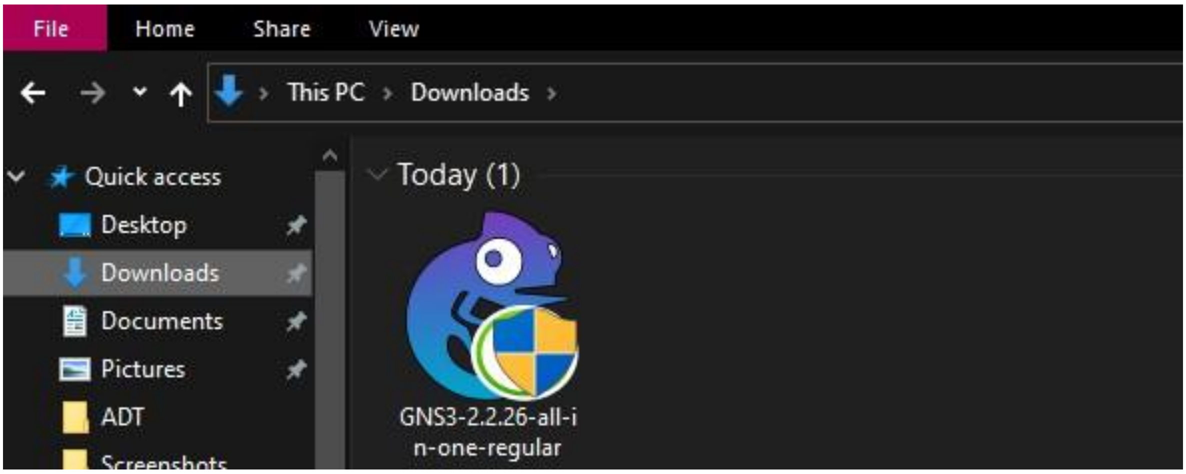
https://www.gns3.com/software/download

Step 2: Select your system



Step 3: Open the downloaded File:

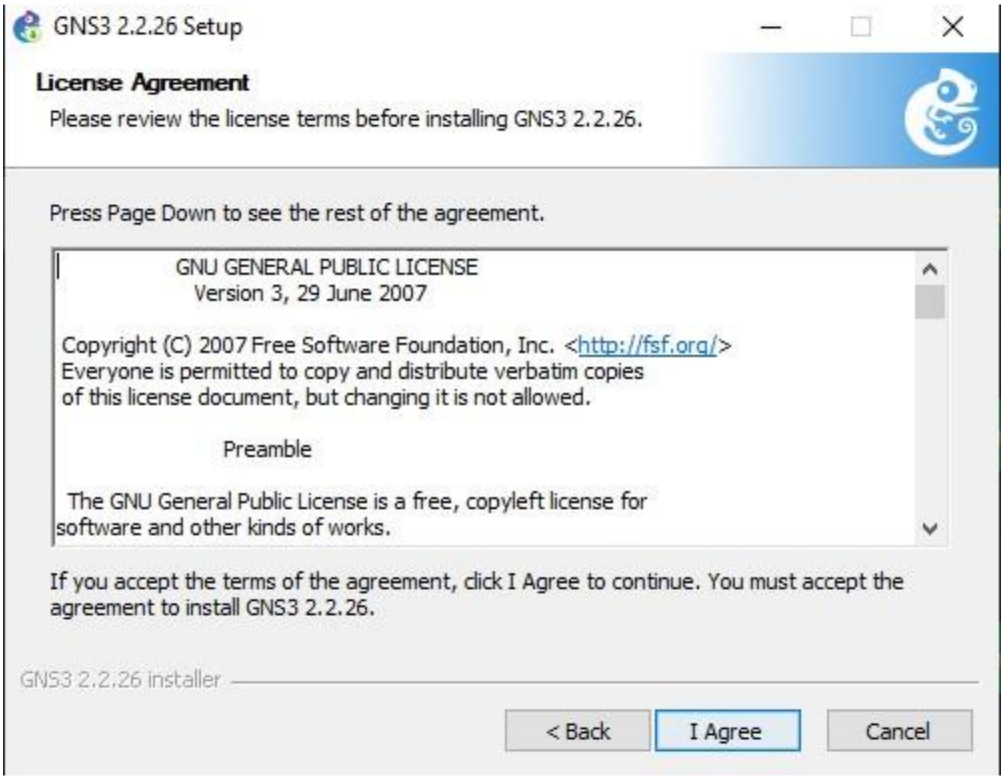
•

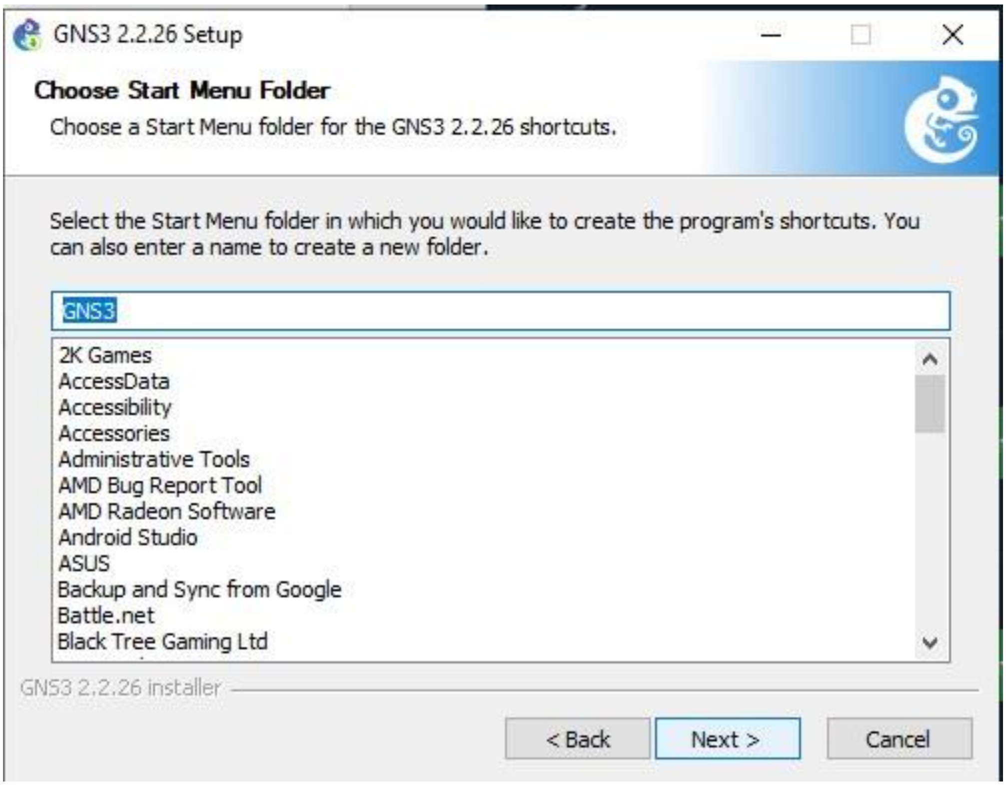


Step 4: Run the setup

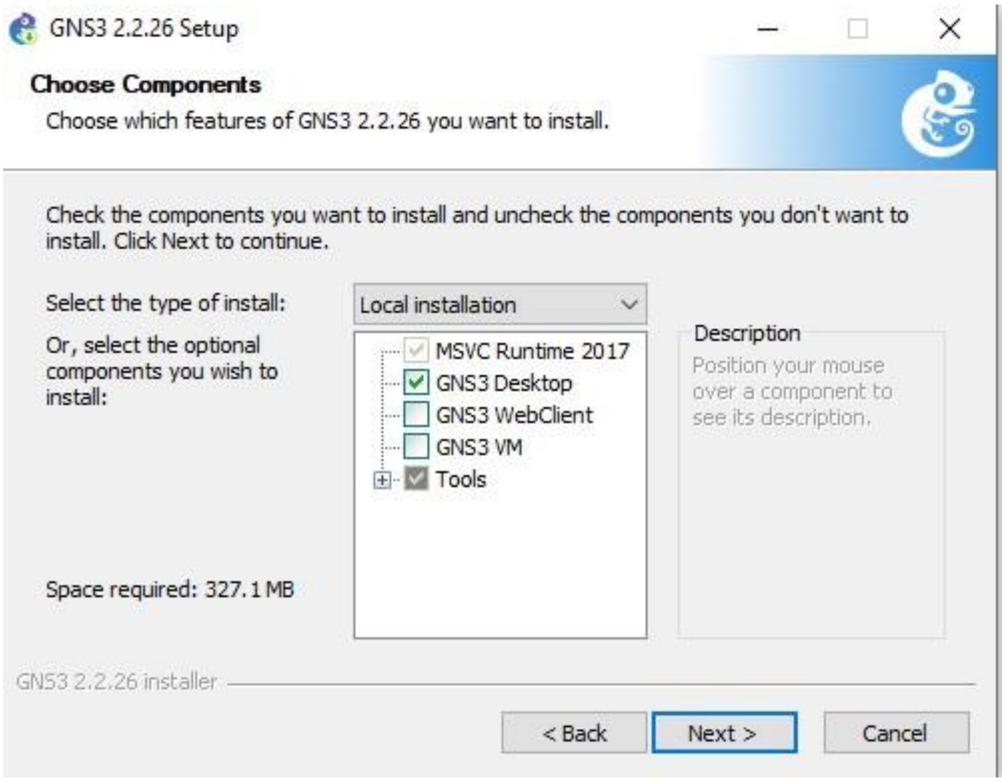
And then , Install According to the Screenshot :

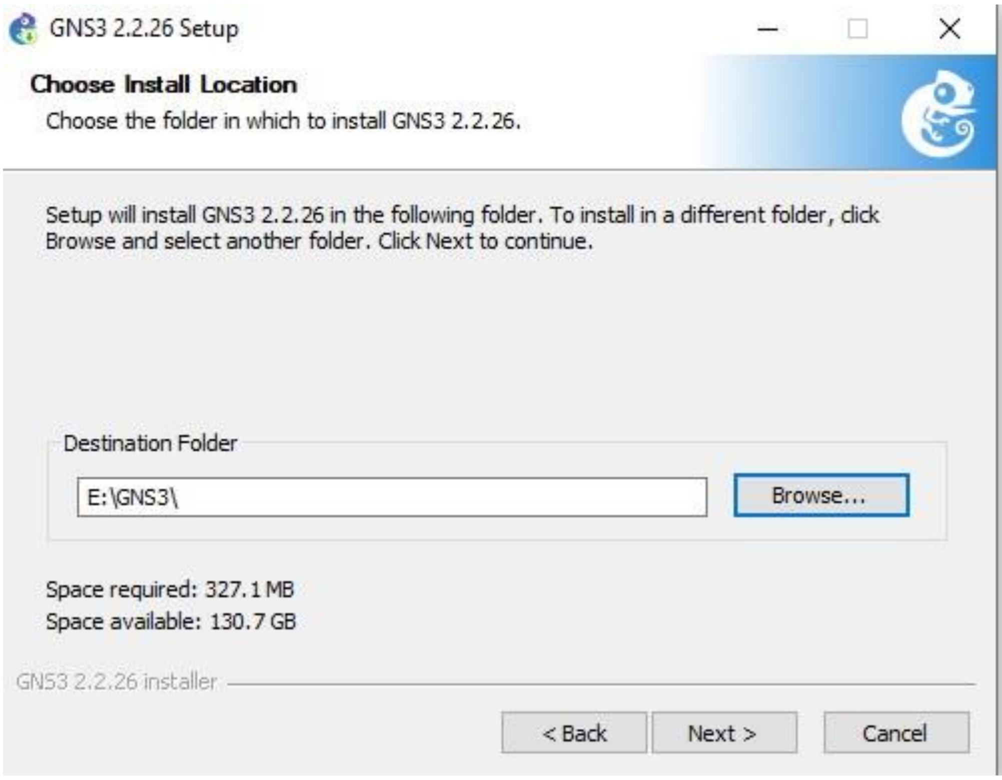
•

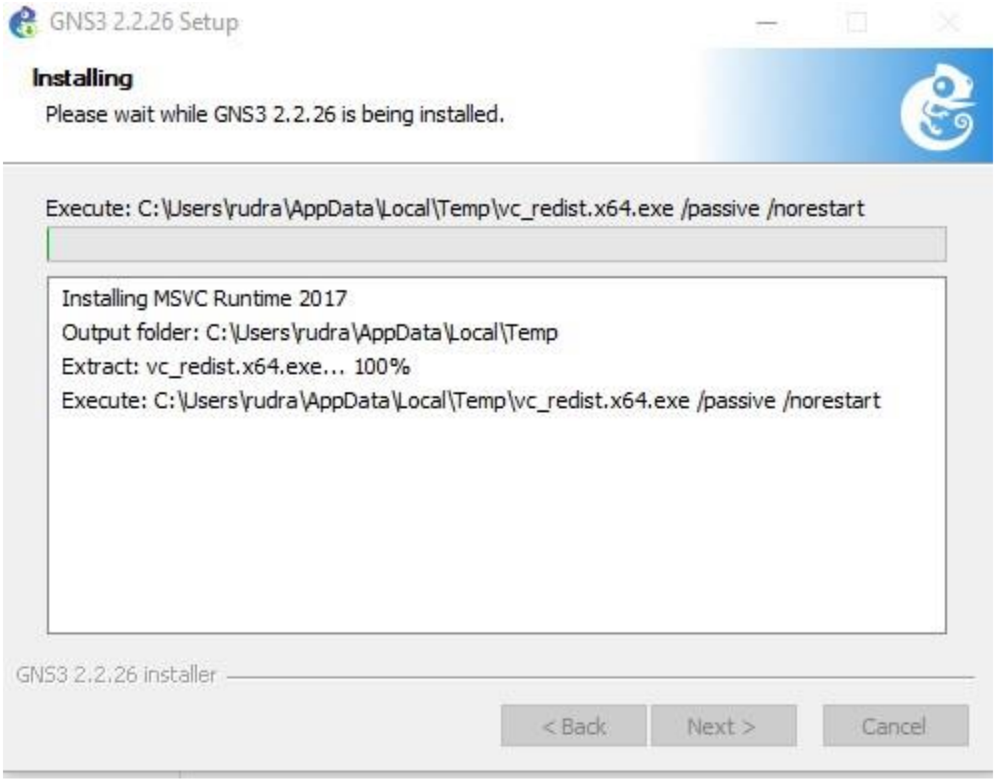

GNS3 2.2.26 Setup
License Agreement
Please review the license terms before installing GNS3 2.2.26.
Press Page Down to see the rest of the agreement.
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
The GNU General Public License is a free, copyleft license for
software and other kinds of works.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install GNS3 2.2.26.
GNS3 2.2.26 installer
< Back
I Agree
Cancel

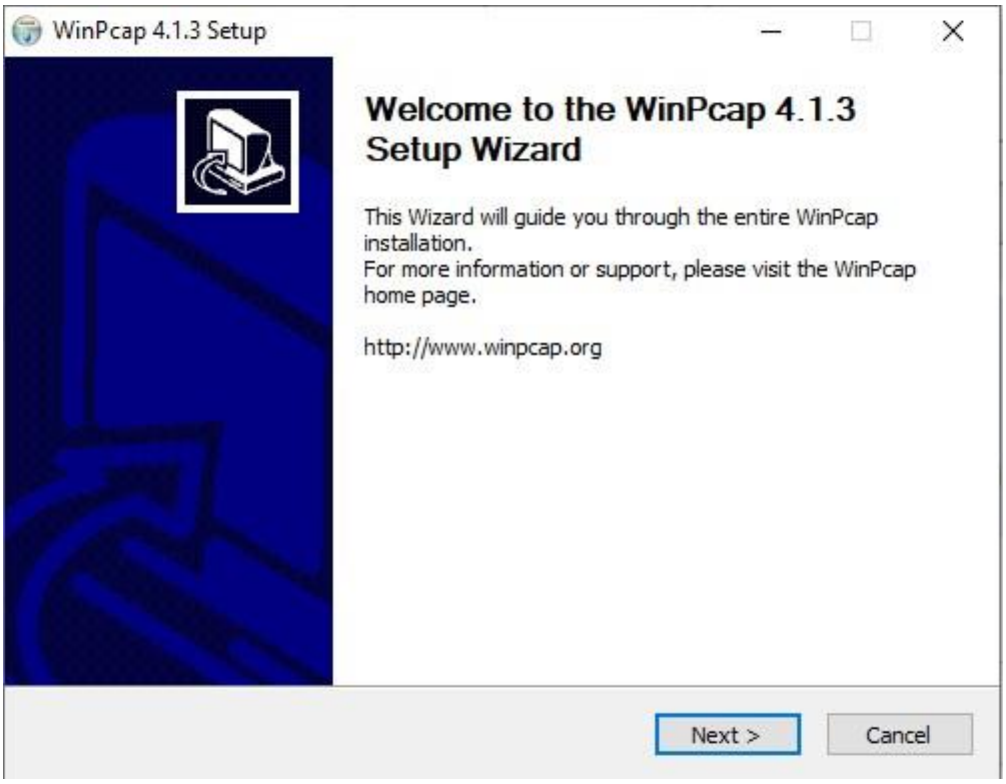

GNS3 2.2.26 Setup
Choose Start Menu Folder
Choose a Start Menu folder for the GNS3 2.2.26 shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
GNS3
2K Games
AccessData
Accessibility
Accessories
Administrative Tools
AMD Bug Report Tool
AMD Radeon Software
Android Studio
ASUS
Backup and Sync from Google
Battle.net
Black Tree Gaming Ltd
GNS3 2.2.26 installer
< Back
Next >
Cancel

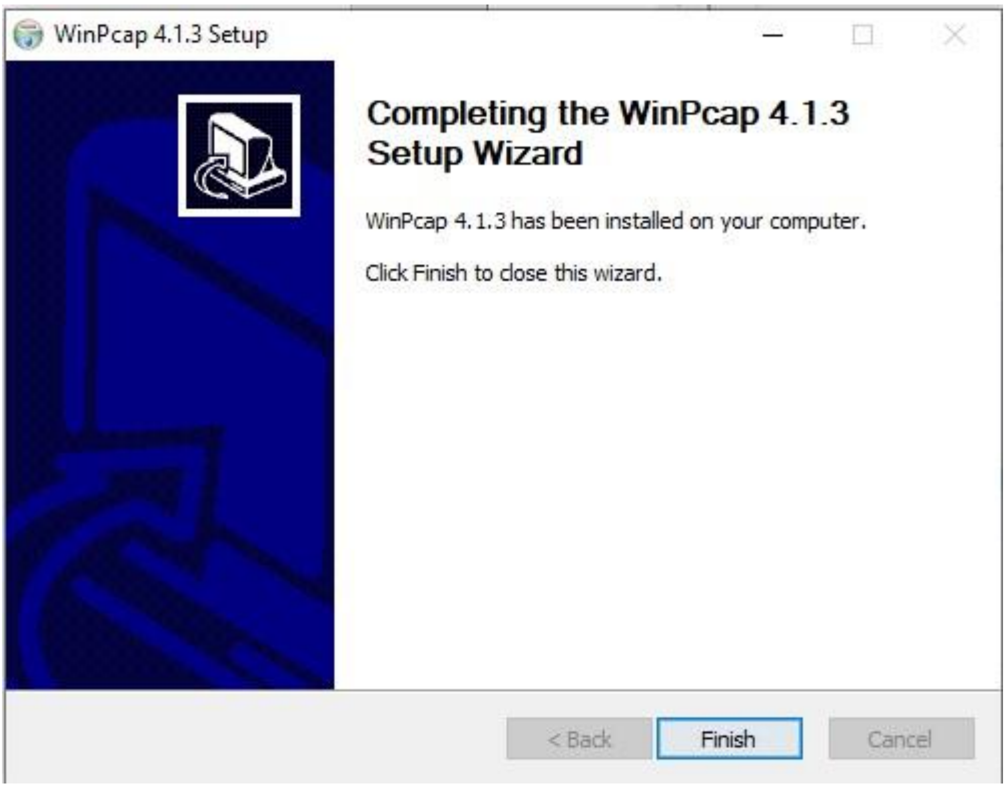
•


GNS3 2.2.26 Setup
Choose Components
Choose which features of GNS3 2.2.26 you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select the type of install:
Local installation
Or, select the optional components you wish to install:
MSVC Runtime 2017
GNS3 Desktop
GNS3 WebClient
GNS3 VM
Tools
Description
Position your mouse over a component to see its description.
Space required: 327.1 MB
GNS3 2.2.26 installer
< Back
Next >
Cancel

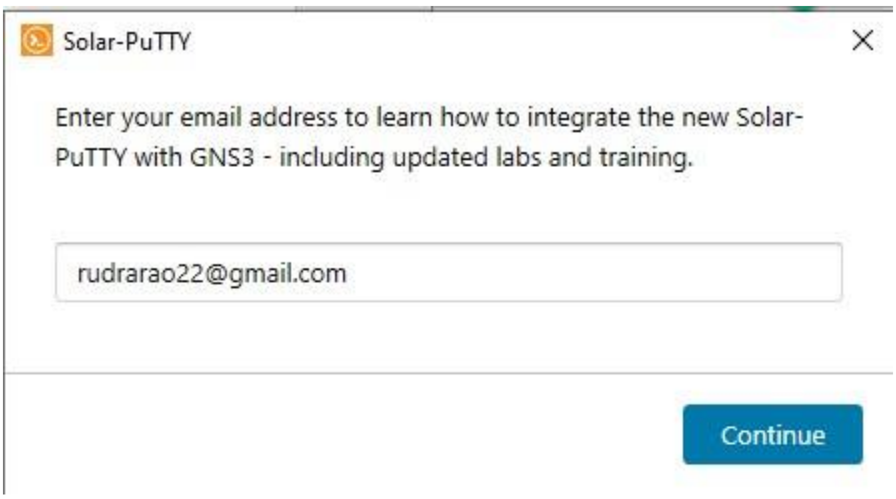

GNS3 2.2.26 Setup
Choose Install Location
Choose the folder in which to install GNS3 2.2.26.
Setup will install GNS3 2.2.26 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
E:\GNS3\
Browse...
Space required: 327.1 MB
Space available: 130.7 GB
GNS3 2.2.26 installer
< Back
Next >
Cancel

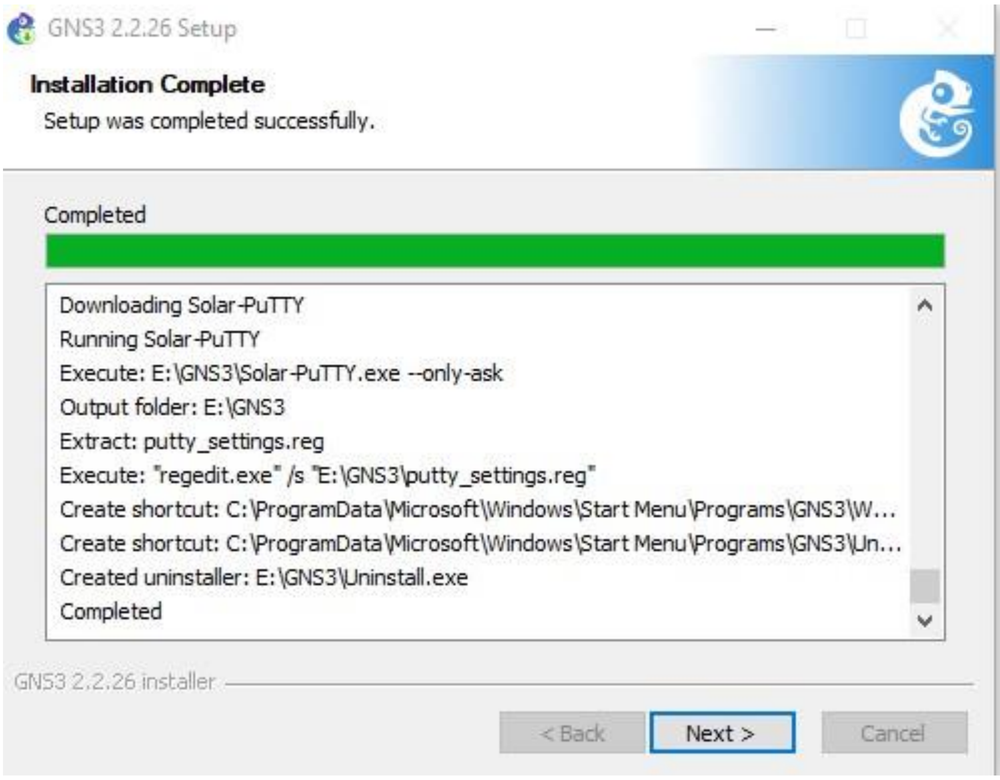
- 
GNS3 2.2.26 Setup
Installing
Please wait while GNS3 2.2.26 is being installed.
Execute: C:\Users\rudra\AppData\Local\Temp\vc_redist.x64.exe /passive /norestart
Installing MSVC Runtime 2017
Output folder: C:\Users\rudra\AppData\Local\Temp
Extract: vc_redist.x64.exe... 100%
Execute: C:\Users\rudra\AppData\Local\Temp\vc_redist.x64.exe /passive /norestart
GNS3 2.2.26 installer
< Back
Next >
Cancel

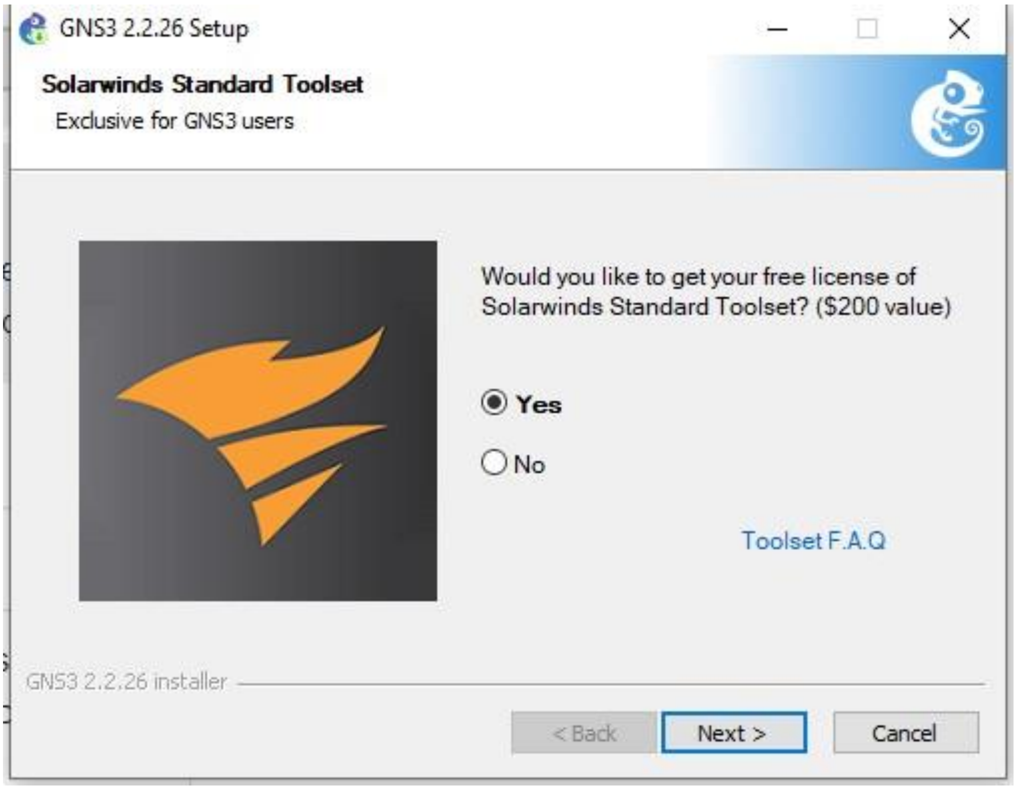

WinPcap 4.1.3 Setup
Welcome to the WinPcap 4.1.3 Setup Wizard
This Wizard will guide you through the entire WinPcap installation.
For more information or support, please visit the WinPcap home page.
http://www.winpcap.org
Next >
Cancel

- 
WinPcap 4.1.3 Setup
Completing the WinPcap 4.1.3 Setup Wizard
WinPcap 4.1.3 has been installed on your computer.
Click Finish to close this wizard.
< Back
Finish
Cancel

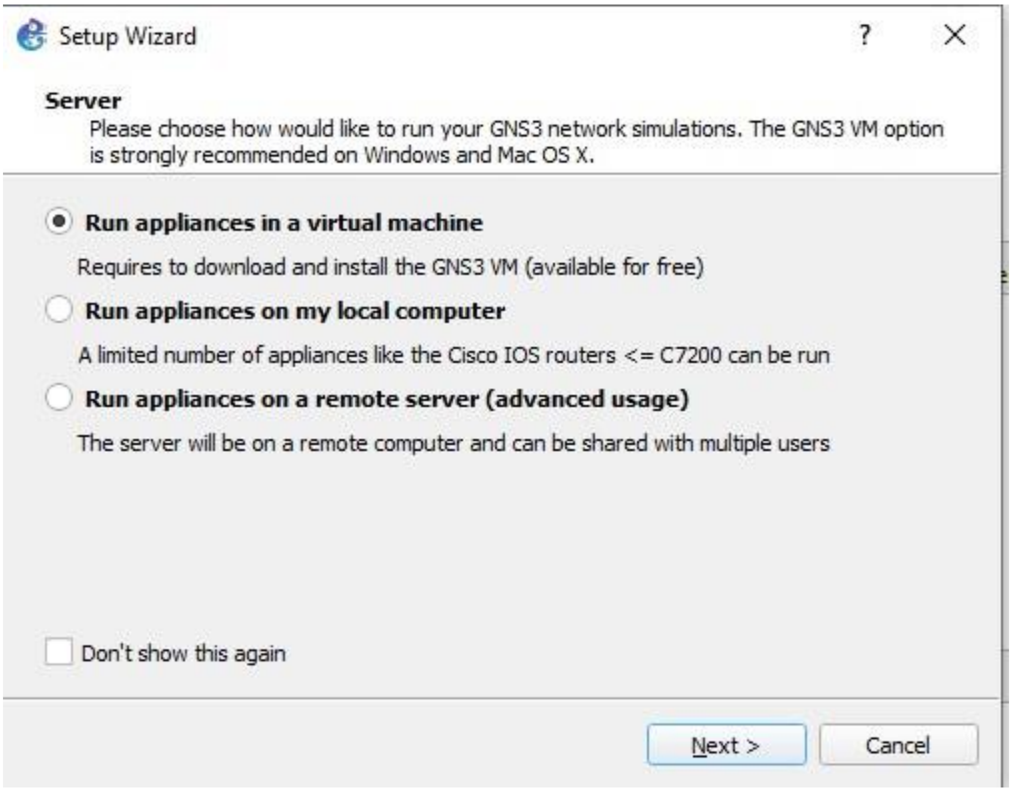

License Agreement and Privacy Notice
I agree to the End User License Agreement and the Privacy Notice *
Accept
Cancel

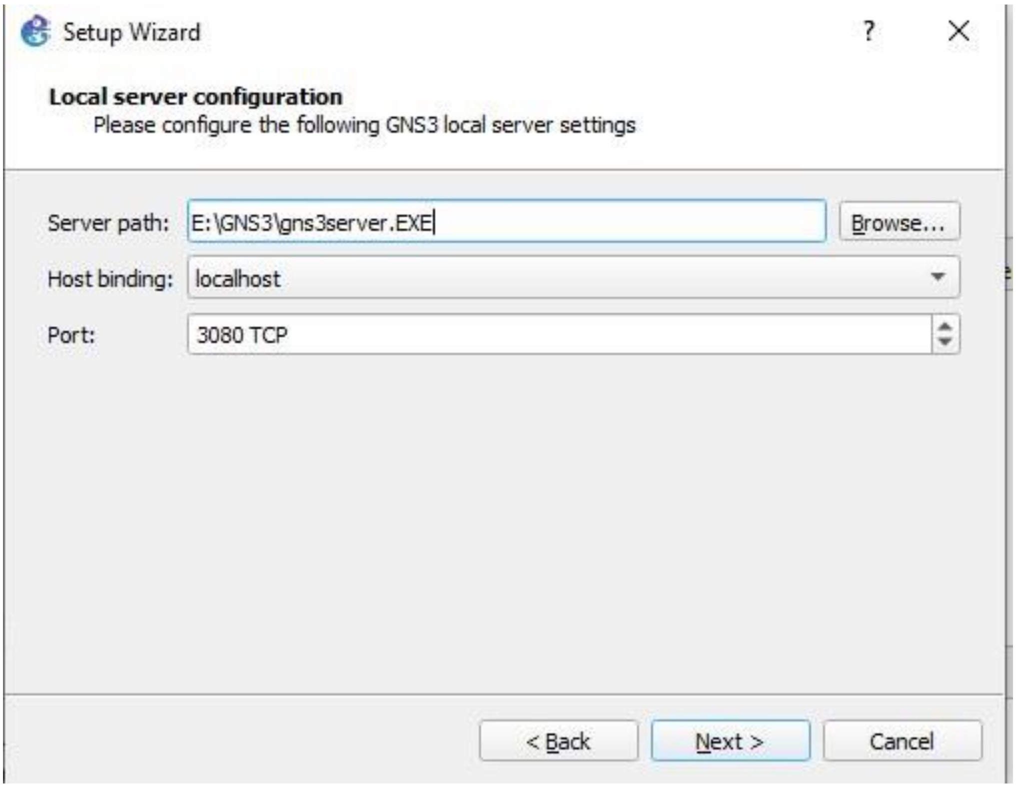

Solar-PuTTY
Enter your email address to learn how to integrate the new Solar-PuTTY with GNS3 - including updated labs and training.
rudrarao22@gmail.com
Continue


GNS3 2.2.26 Setup
Installation Complete
Setup was completed successfully.
Completed
Downloading Solar-PuTTY
Running Solar-PuTTY
Execute: E:\GNS3\Solar-PuTTY.exe --only-ask
Output folder: E:\GNS3
Extract: putty_settings.reg
Execute: "regedit.exe" /s "E:\GNS3\putty_settings.reg"
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\GNS3\W...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\GNS3\Un...
Created uninstaller: E:\GNS3\Uninstall.exe
Completed
GNS3 2.2.26 installer
< Back
Next >
Cancel


GNS3 2.2.26 Setup
Solarwinds Standard Toolset
Exclusive for GNS3 users
Would you like to get your free license of Solarwinds Standard Toolset? ($200 value)
Yes
No
Toolset F.A.Q
GNS3 2.2.26 installer
< Back
Next >
Cancel


GNS3 2.2.26 Setup
GNS3
Completing GNS3 2.2.26 Setup
GNS3 2.2.26 has been installed on your computer.
Click Finish to close Setup.
Start GNS3
< Back
Finish
Cancel

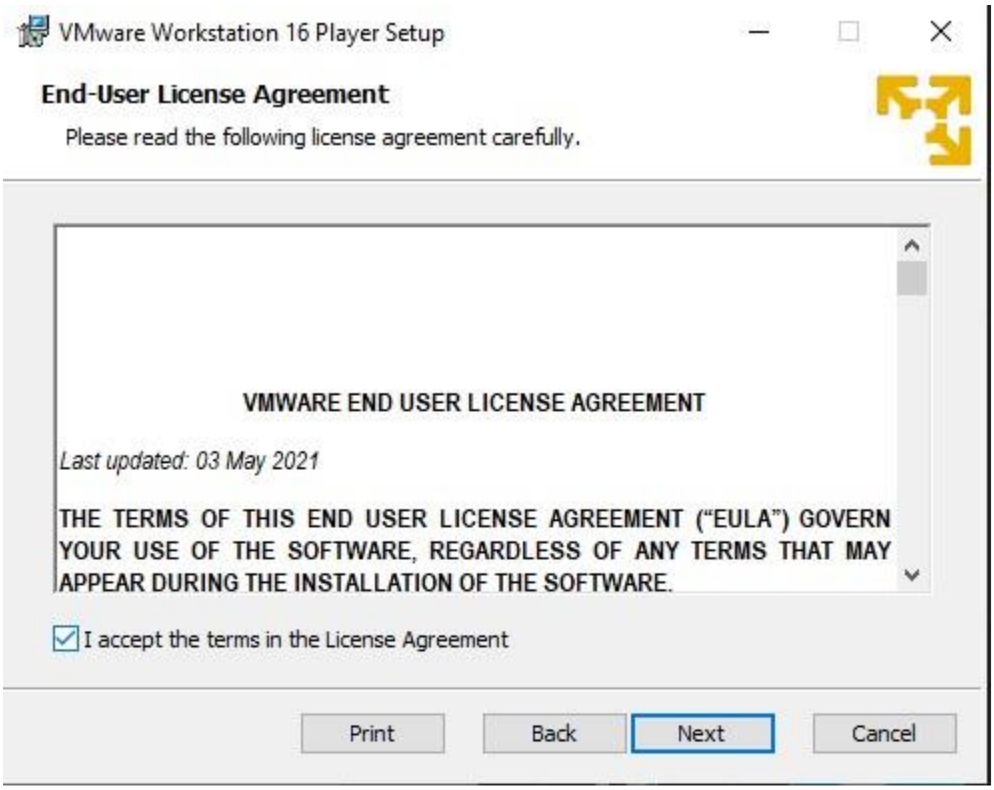

Setup Wizard
Server
Please choose how would like to run your GNS3 network simulations. The GNS3 VM option is strongly recommended on Windows and Mac OS X.
Run appliances in a virtual machine
Requires to download and install the GNS3 VM (available for free)
Run appliances on my local computer
A limited number of appliances like the Cisco IOS routers <= C7200 can be run
Run appliances on a remote server (advanced usage)
The server will be on a remote computer and can be shared with multiple users
Don't show this again
Next >
Cancel

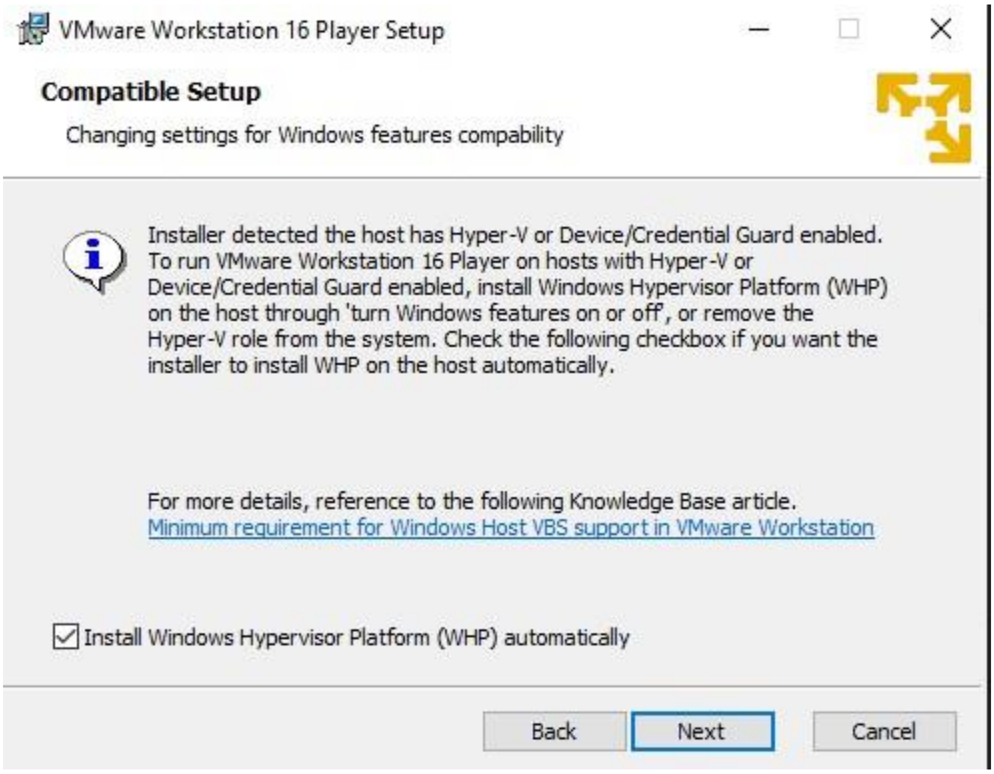


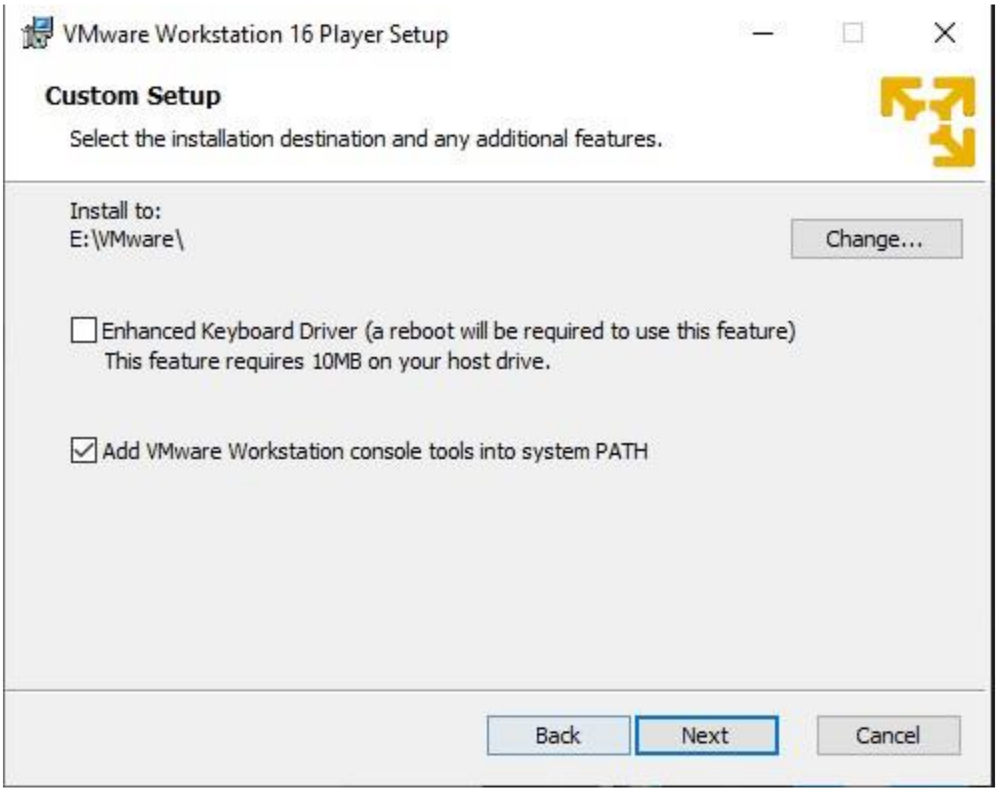
Step 5: Install VMware

-

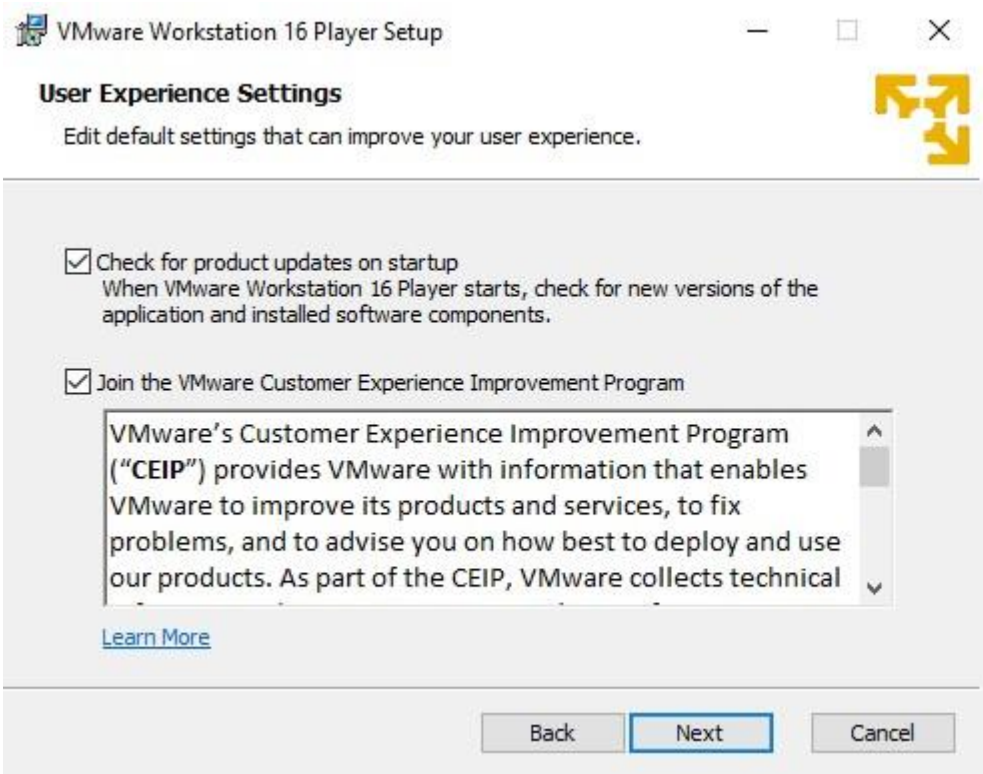

VMware Workstation 16 Player Setup
VMWARE
WORKSTATION
PLAYER™ 16
Welcome to the VMware Workstation 16 Player Setup Wizard
The Setup Wizard will install VMware Workstation 16 Player on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Copyright 1998-2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at:
http://www.vmware.com/go/patents
Next
Cancel

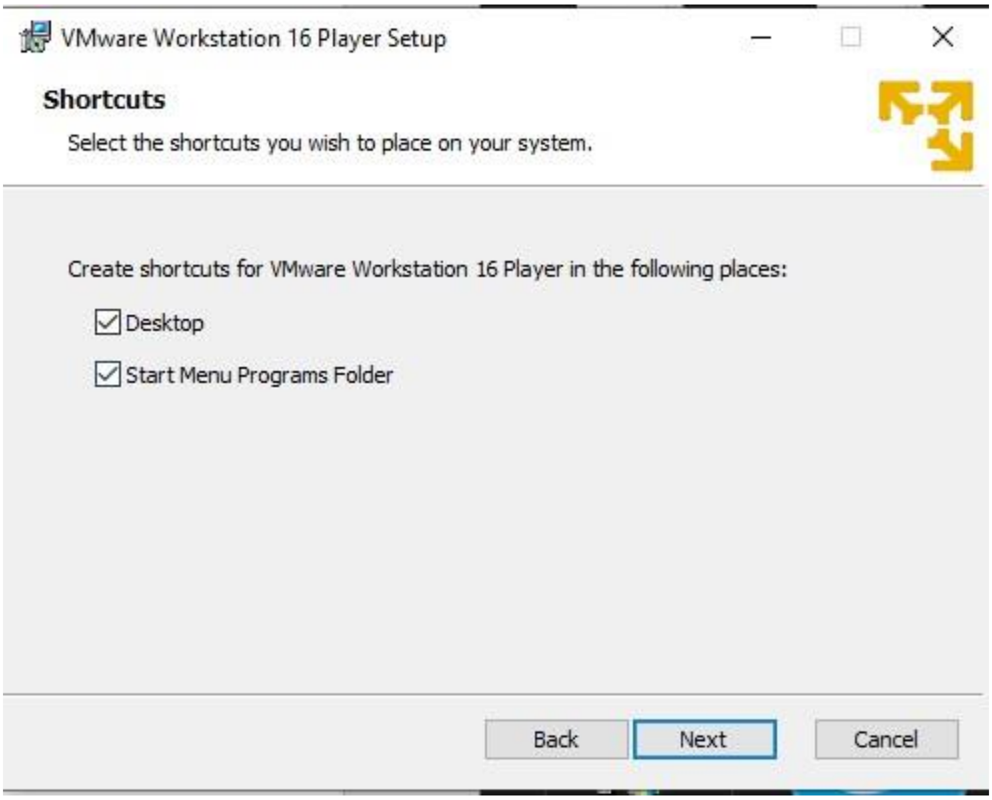

VMware Workstation 16 Player Setup
End-User License Agreement
Please read the following license agreement carefully.
VMWARE END USER LICENSE AGREEMENT
Last updated: 03 May 2021
THE TERMS OF THIS END USER LICENSE AGREEMENT ("EULA") GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE SOFTWARE.
I accept the terms in the License Agreement
Print
Back
Next
Cancel

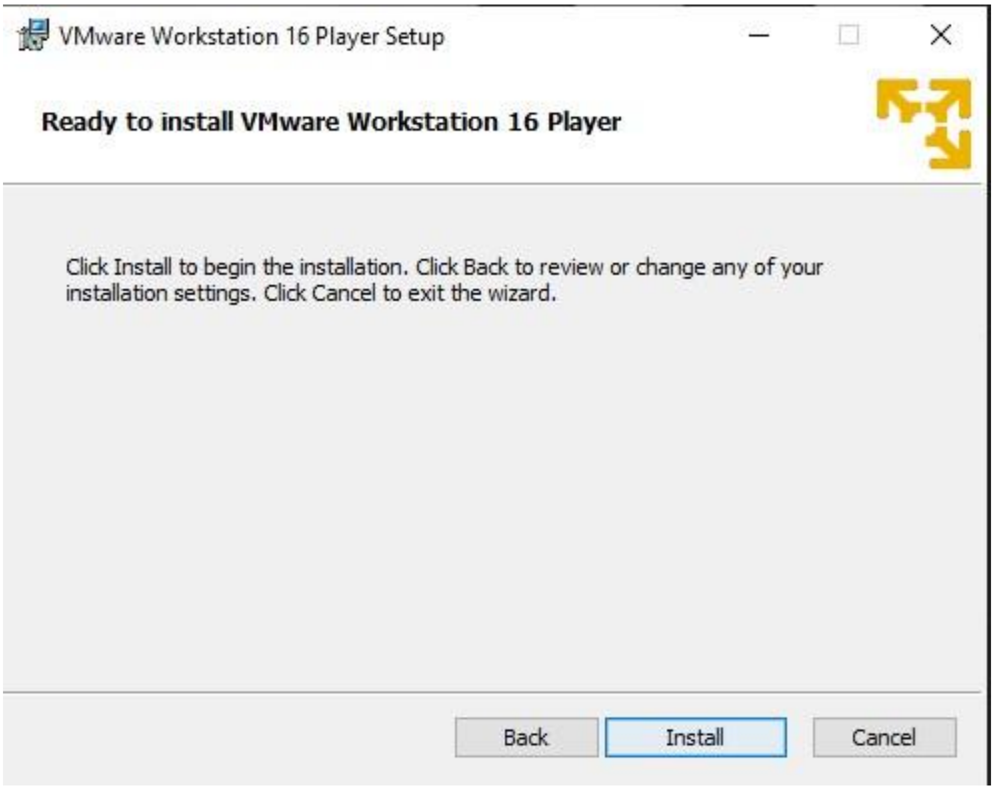
- 
VMware Workstation 16 Player Setup
Compatible Setup
Changing settings for Windows features compability
Installer detected the host has Hyper-V or Device/Credential Guard enabled. To run VMware Workstation 16 Player on hosts with Hyper-V or Device/Credential Guard enabled, install Windows Hypervisor Platform (WHP) on the host through 'turn Windows features on or off', or remove the Hyper-V role from the system. Check the following checkbox if you want the installer to install WHP on the host automatically.
For more details, reference to the following Knowledge Base article.
Minimum requirement for Windows Host VBS support in VMware Workstation
Install Windows Hypervisor Platform (WHP) automatically
Back
Next
Cancel

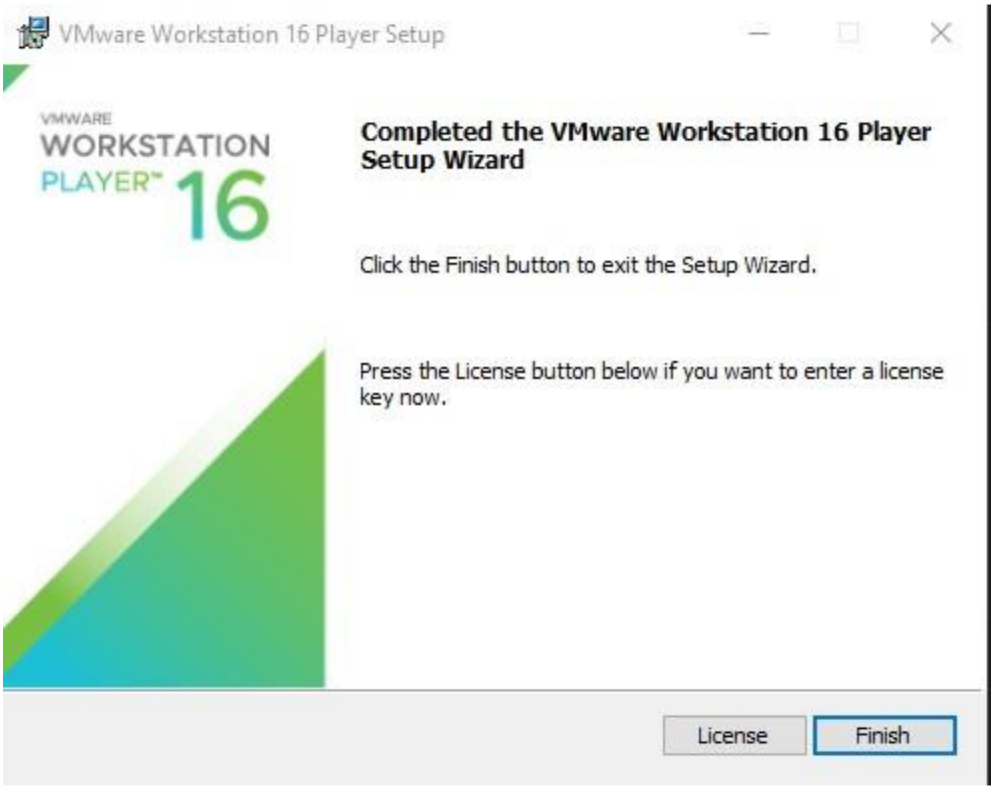

VMware Workstation 16 Player Setup
Custom Setup
Select the installation destination and any additional features.
Install to:
E:\VMware\
Change...
Enhanced Keyboard Driver (a reboot will be required to use this feature)
This feature requires 10MB on your host drive.
Add VMware Workstation console tools into system PATH
Back
Next
Cancel

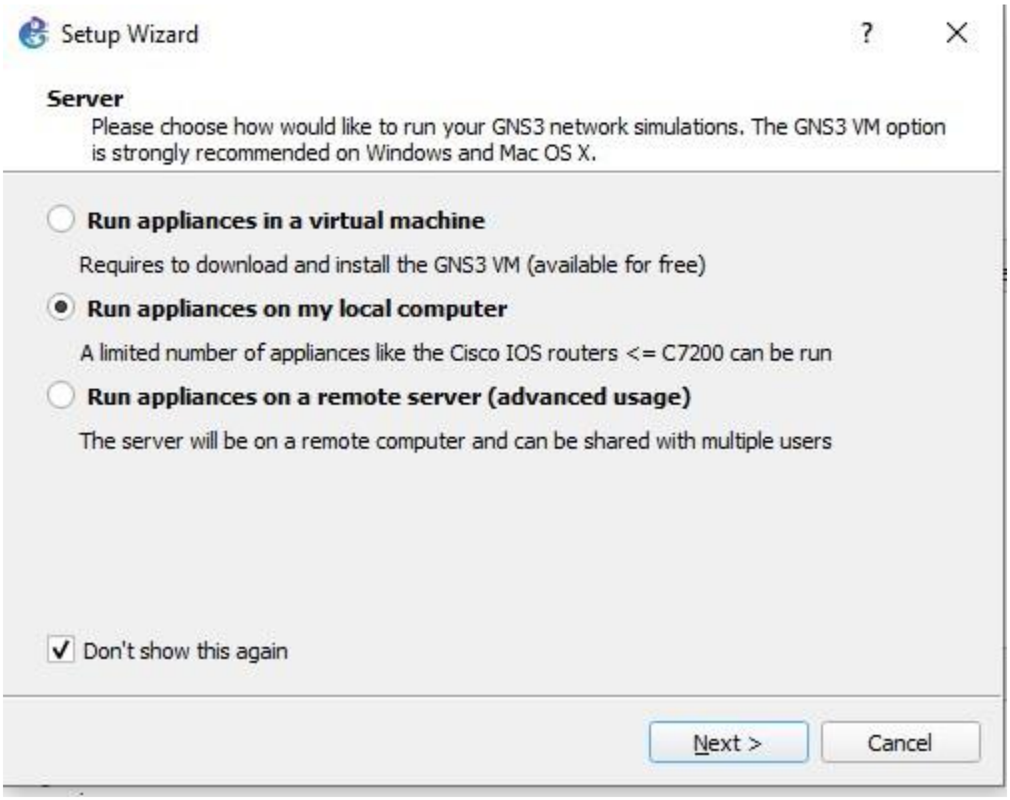
•

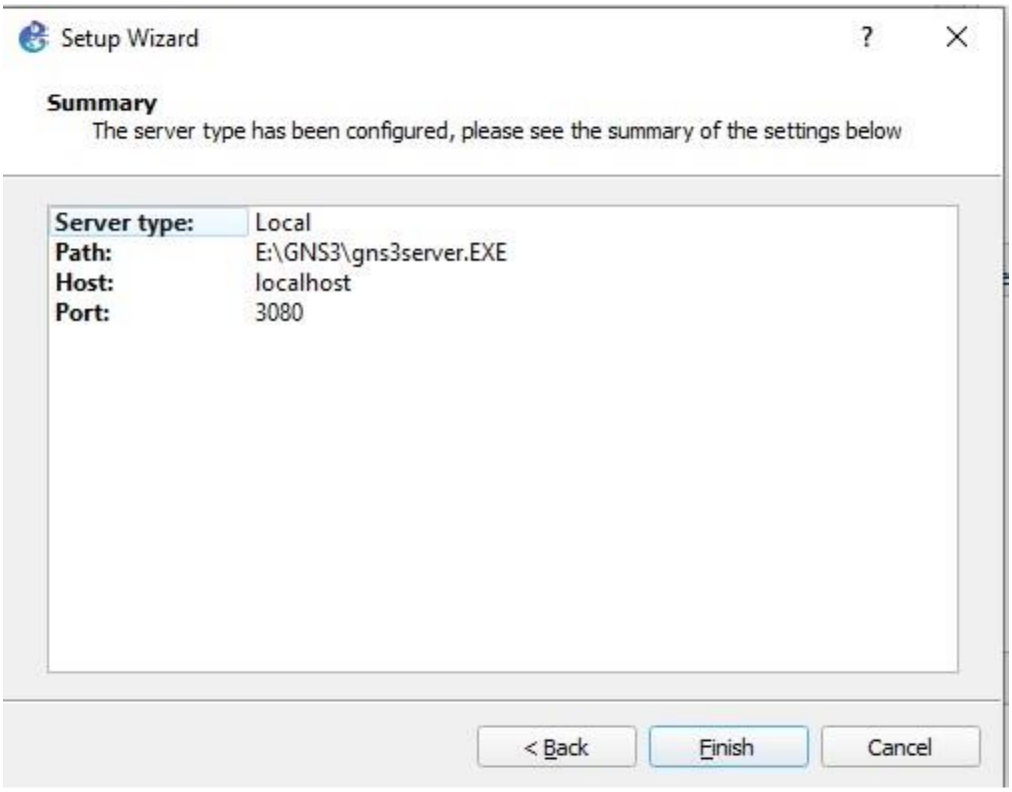

VMware Workstation 16 Player Setup
User Experience Settings
Edit default settings that can improve your user experience.
Check for product updates on startup
When VMware Workstation 16 Player starts, check for new versions of the application and installed software components.
Join the VMware Customer Experience Improvement Program
VMware's Customer Experience Improvement Program ("CEIP") provides VMware with information that enables VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products. As part of the CEIP, VMware collects technical
Learn More
Back
Next
Cancel


VMware Workstation 16 Player Setup
Shortcuts
Select the shortcuts you wish to place on your system.
Create shortcuts for VMware Workstation 16 Player in the following places:
Desktop
Start Menu Programs Folder
Back
Next
Cancel

- 
VMware Workstation 16 Player Setup
Ready to install VMware Workstation 16 Player
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel


VMware Workstation 16 Player Setup
VMWARE
WORKSTATION
PLAYER™ 16
Completed the VMware Workstation 16 Player Setup Wizard
Click the Finish button to exit the Setup Wizard.
Press the License button below if you want to enter a license key now.
License
Finish

- 
Setup Wizard
?
Server
Please choose how would like to run your GNS3 network simulations. The GNS3 VM option is strongly recommended on Windows and Mac OS X.
Run appliances in a virtual machine
Requires to download and install the GNS3 VM (available for free)
Run appliances on my local computer
A limited number of appliances like the Cisco IOS routers <= C7200 can be run
Run appliances on a remote server (advanced usage)
The server will be on a remote computer and can be shared with multiple users
Don't show this again
Next >
Cancel

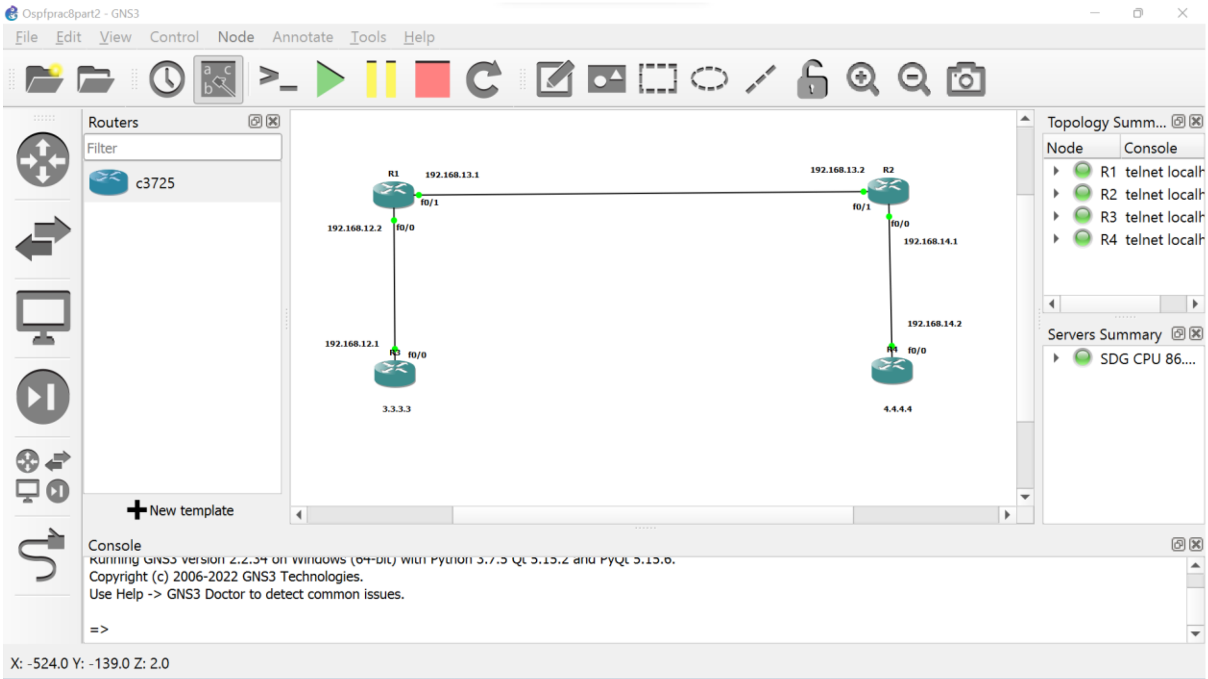

Setup Wizard
?
Summary
The server type has been configured, please see the summary of the settings below
Server type: Local
Path: E:\GNS3\gns3server.EXE
Host: localhost
Port: 3080
< Back
Finish
Cancel

•

SDN Practical 2

Aim: Implement IPv4 ACLs 1. Standard 2. Extended

Step 1: Design the Topology



Step 2: Explain what ACL is and how we apply it in the current system.

• Access-list (ACL) is a set of rules defined for controlling network traffic and reducing network attacks. ACLs are used to filter traffic based on the set of rules defined for the incoming or outgoing of the network.

• Layer 3 Security – Router blocks the IP address, which means that ACL is applied.

• ACL is also called as packet filtering firewall

• There are two main types of ACL:

1. Standard:

- Standard access-list is implemented using source IP address only.

- Standard Access-list is generally applied close to destination (but not always).

- Standard access-list uses the range 1-99 and extended range 1300-1999.

- In a standard access list, the whole network or sub-network is denied.

2. Extended:

- In the Extended access list, packet filtering takes place on the basis of source

•

IP address, destination IP address, port numbers.

- Extended access-list is generally applied close to the source but not always.

- Extended ACL is created from 100 – 199 & extended range 2000 – 2699.

- In an extended access list, particular services will be permitted or denied.

• To enable ACL on our network we use Routing protocols.

Note: As the destination is not reachable from PC 1 to Router 2 (means via connection is not reachable) so we apply a routing protocol.

• There are many routing protocols:

1. Static

2. Dynamic:

- RIP, OSPF, EIGRP.

- Here we apply RIP. After applying RIP all the Routers and PCs are able to communicate and ping each other.

Step 3 : Configure the System.

Here we configure Router 1 (R1).

We set the ip address for the router and its various connections.

For the PC 1 connection we use the FastEthernet 0/0 interface and set its IP-address to '192.168.1.50'

Then we configure the serial 1/0 interface and set its IP-address to 10.1.1.1 with 255.255.255.0 as its subnet mask.

Note: Always remember to set the 'no shutdown' attribute for each interface. This way the protocol state will be changed to up rather than down

Also after setting up the configuration, it is advised to ping a nearby device to check whether the connection has been established.

-

```
 changed state to down
R1#
R1#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface FastEthernet 0/0
R1(config-if)#ip address 192.168.1.50 255.255.255.0
R1(config-if)#exit
R1(config)#exit
R1#
*Mar  1 00:07:41.435: %SYS-5-CONFIG_I: Configured from console by console
R1#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface Serial1/0
R1(config-if)#ip address 10.1.1.1 255.255.2555.0
                                              ^
% Invalid input detected at '^' marker.

R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#exit
R1#
*Mar  1 00:09:32.971: %SYS-5-CONFIG_I: Configured from console by console
R1#wr
Building configuration...
[OK]
R1#
R1#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface FastEthernet0/0
R1(config-if)#no shut
R1(config-if)#
*Mar  1 00:11:01.511: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:11:02.511: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#exit
R1(config)#interface Serial1/0
R1(config-if)#o shut
               ^
% Invalid input detected at '^' marker.

R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Mar  1 00:11:47.875: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:11:48.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R1(config)#exit
R1#wr
Building configuration...

*Mar  1 00:12:09.959: %SYS-5-CONFIG_I: Configured from console by console[OK]
R1#
*Mar  1 00:12:12.395: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to down
R1#
R1#
*Mar  1 00:21:22.387: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
```

```
R1#copy run
R1#copy running-config start
Destination filename [startup-config]?
Building configuration...
[OK]
R1#
R1#router rip
    ^
```

PC 1:

PC 1 can be used to help send data to other PCs on the network via its router.

In this case PC 1 is connected to R1 via the fastEthernet 0/0 connection.

Here we set the Ip-address of the PC to ‘192.168.1.1/24’ with a gateway of ‘192.168.1.50’.

-

```
PC1 : 192.168.1.1 255.255.255.0 gateway 192.168.1.50

PC1> ping 192.168.1.50
84 bytes from 192.168.1.50 icmp_seq=1 ttl=255 time=30.250 ms
84 bytes from 192.168.1.50 icmp_seq=2 ttl=255 time=15.379 ms
84 bytes from 192.168.1.50 icmp_seq=3 ttl=255 time=15.194 ms
84 bytes from 192.168.1.50 icmp_seq=4 ttl=255 time=15.230 ms
84 bytes from 192.168.1.50 icmp_seq=5 ttl=255 time=15.258 ms

PC1> ping 192.168.2.50
*192.168.1.50 icmp_seq=1 ttl=255 time=30.123 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=2 ttl=255 time=15.175 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=3 ttl=255 time=15.089 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=4 ttl=255 time=30.109 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=5 ttl=255 time=15.063 ms (ICMP type:3, code:1, Destination host unreachable)

PC1> ahow
Bad command: "ahow". Use ? for help.

PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.1.1/24
GATEWAY     : 192.168.1.50
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10024
RHOST:PORT  : 127.0.0.1:10025
MTU:        : 1500

PC1> ping 192.168.3.50
*11.1.1.2 icmp_seq=1 ttl=253 time=74.898 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=2 ttl=253 time=75.166 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=3 ttl=253 time=90.037 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=4 ttl=253 time=75.123 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=5 ttl=253 time=105.161 ms (ICMP type:3, code:13, Communication administratively prohibited)

PC1> ping 192.168.2.50
84 bytes from 192.168.2.50 icmp_seq=1 ttl=254 time=45.039 ms
84 bytes from 192.168.2.50 icmp_seq=2 ttl=254 time=45.223 ms
84 bytes from 192.168.2.50 icmp_seq=3 ttl=254 time=60.184 ms
84 bytes from 192.168.2.50 icmp_seq=4 ttl=254 time=60.236 ms
84 bytes from 192.168.2.50 icmp_seq=5 ttl=254 time=60.121 ms

PC1> ping 192.168.2.1
84 bytes from 192.168.2.1 icmp_seq=1 ttl=62 time=90.130 ms
84 bytes from 192.168.2.1 icmp_seq=2 ttl=62 time=75.076 ms
84 bytes from 192.168.2.1 icmp_seq=3 ttl=62 time=60.145 ms
84 bytes from 192.168.2.1 icmp_seq=4 ttl=62 time=60.206 ms
84 bytes from 192.168.2.1 icmp_seq=5 ttl=62 time=75.280 ms

PC1> sh ip

NAME        : PC1[1]
IP/MASK     : 192.168.1.1/24
GATEWAY     : 192.168.1.50
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10024
RHOST:PORT  : 127.0.0.1:10025
MTU:        : 1500

PC1> z
```

solarwinds | Solar-PuTTY *free tool*

Router 2:

Here we configure Router 2 (R3).

R3 is connected to three devices (2 routers – R1 and R3, and one PC – PC 2).

It is connected to PC 2 via the fastEthernet 0/0 with its Ip-address set to ‘192.168.2.50’ with a subnet mask of ‘255.255.255.0’

It is connected to R1 via the interface serial 1/0 with its Ip-address set to ‘10.1.1.2’ with a subnet mask of ‘255.255.255.0’.

- It is connected to R2 via the interface serial 1/1 with its Ip-address set to ‘11.1.1.1’ with a subnet mask of ‘255.255.255.0’

Note: Always remember to set the ‘no shutdown’ attribute for each interface. This way the protocol state will be changed to up rather than down

Also after setting up the configuration, it is advised to ping a nearby device to check whether the connection has been established.

```
R2#
R2#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface FastEthernet0/0
R2(config-if)#ip address 192.168.2.50 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Mar  1 00:19:47.083: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:19:48.083: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config)#interface Serial1/0
R2(config-if)#ip address 10.1.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Mar  1 00:21:14.471: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:21:15.471: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R2(config)#interface Serial1/1
R2(config-if)#ip address 11.1.1.1 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Mar  1 00:21:47.575: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Mar  1 00:21:48.575: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
R2(config)#end
R2#
*Mar  1 00:21:59.479: %SYS-5-CONFIG_I: Configured from console by console
R2#wr
Building configuration...
[OK]
R2#
*Mar  1 00:22:12.087: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to down
R2#wr
Building configuration...
[OK]
```

```
R2#copy run
R2#copy running-config start
Destination filename [startup-config]?
Building configuration...
[OK]
R2#
*Mar  1 00:25:52.079: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
R2#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 10.0.0.0
R2(config-router)#network 11.0.0.0
R2(config-router)#network 192.168.2.0
R2(config-router)#exit
R2(config)#exit
R2#
*Mar  1 00:35:52.779: %SYS-5-CONFIG_I: Configured from console by console
R2#
R2#wr
Building configuration...
[OK]
R2#
R2#
R2#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#access-list 141 deny icmp host 192.168.2.1 host 192.168.3.1
R2(config)#do show access-list
Extended IP access list 141
    10 deny icmp host 192.168.2.1 host 192.168.3.1
R2(config)#access-list 141 permit icmp host any any
Translating "any"
                                             ^
% Invalid input detected at '^' marker.

R2(config)#access-list 141 permit icmp any any
R2(config)#do show access-list
Extended IP access list 141
    10 deny icmp host 192.168.2.1 host 192.168.3.1
    20 permit icmp any any
R2(config)#interface Serial1/1
R2(config-if)#ip access
R2(config-if)#ip access-group 141 out
R2(config-if)#do show access-list
Extended IP access list 141
    10 deny icmp host 192.168.2.1 host 192.168.3.1
    20 permit icmp any any
R2(config-if)#exit
R2(config)#exit
R2#
*Mar  1 00:59:18.739: %SYS-5-CONFIG_I: Configured from console by console
R2#
```

solarwinds | Solar-PuTTY free tool

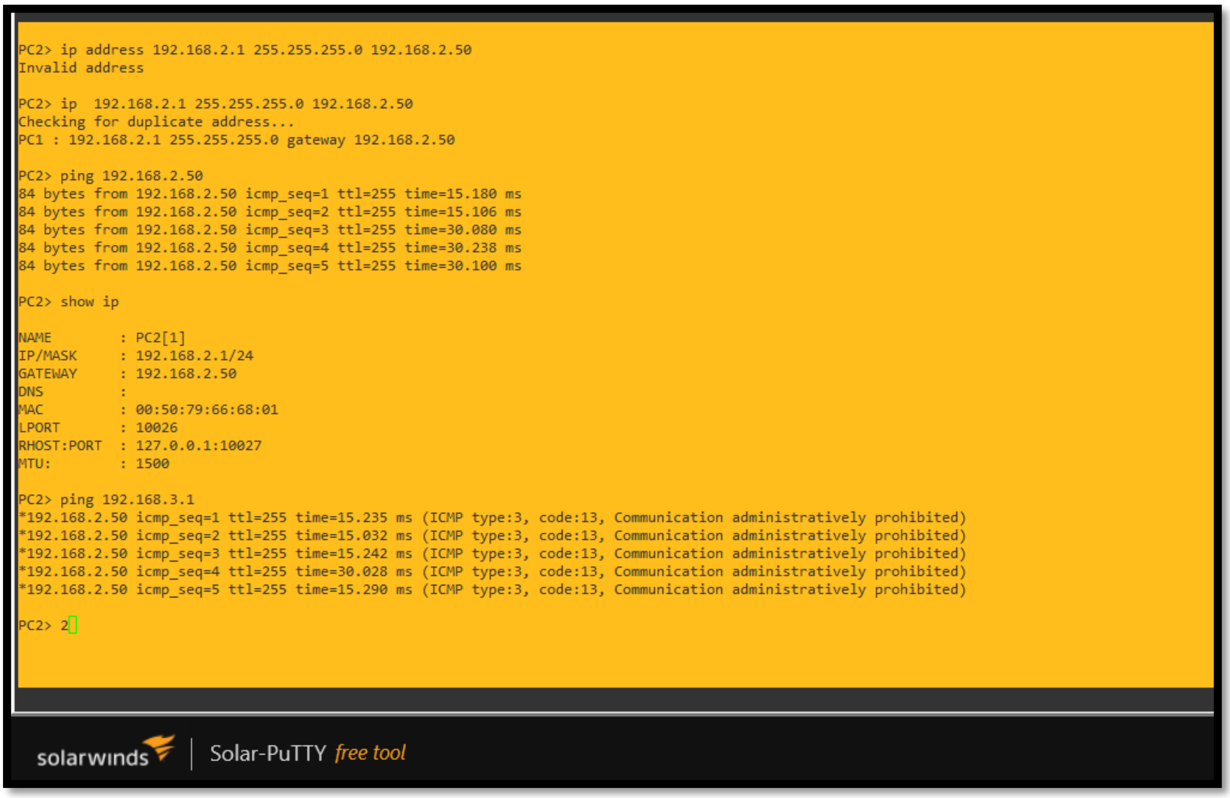
PC 2:

PC 2 can be used to help send data to other PCs on the network via its router.

-

In this case PC 2 is connected to R3 via the fastEthernet 0/0 connection.

Here we set the Ip-address of the PC to ‘192.168.2.1/24’ with a gateway of ‘192.168.2.50



```
PC2> ip address 192.168.2.1 255.255.255.0 192.168.2.50
Invalid address

PC2> ip  192.168.2.1 255.255.255.0 192.168.2.50
Checking for duplicate address...
PC1 : 192.168.2.1 255.255.255.0 gateway 192.168.2.50

PC2> ping 192.168.2.50
84 bytes from 192.168.2.50 icmp_seq=1 ttl=255 time=15.180 ms
84 bytes from 192.168.2.50 icmp_seq=2 ttl=255 time=15.106 ms
84 bytes from 192.168.2.50 icmp_seq=3 ttl=255 time=30.080 ms
84 bytes from 192.168.2.50 icmp_seq=4 ttl=255 time=30.238 ms
84 bytes from 192.168.2.50 icmp_seq=5 ttl=255 time=30.100 ms

PC2> show ip

NAME        : PC2[1]
IP/MASK     : 192.168.2.1/24
GATEWAY     : 192.168.2.50
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 10026
RHOST:PORT  : 127.0.0.1:10027
MTU:        : 1500

PC2> ping 192.168.3.1
*192.168.2.50 icmp_seq=1 ttl=255 time=15.235 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=2 ttl=255 time=15.032 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=3 ttl=255 time=15.242 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=4 ttl=255 time=30.028 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=5 ttl=255 time=15.290 ms (ICMP type:3, code:13, Communication administratively prohibited)

PC2> 2
```


Router 3:

Here we configure Router 3 (R2).

R3 is connected to two devices (router – R3, and PC – PC 3).

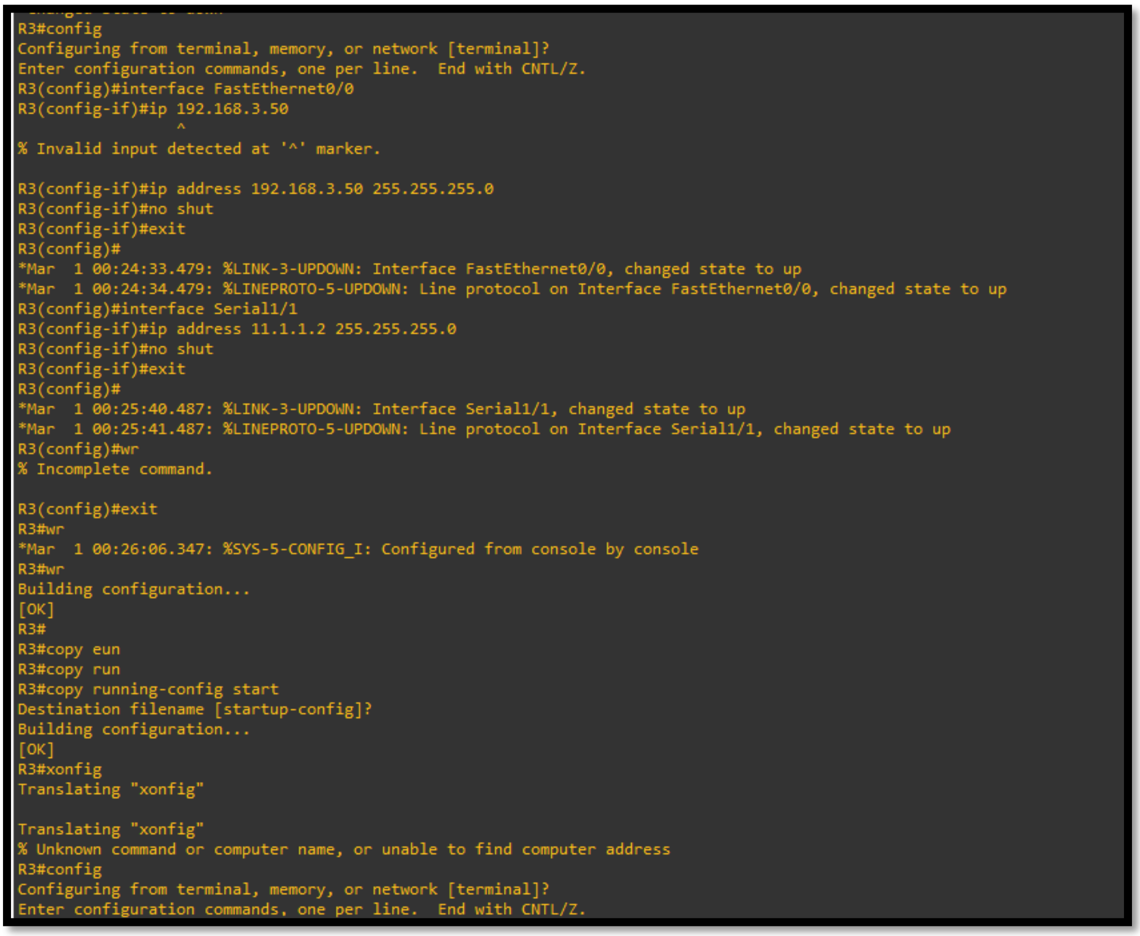
It is connected to PC 3 via the fastEthernet 0/0 with its Ip-address set to ‘192.168.2.50’ with a subnet mask of ‘255.255.255.0’.

Then we configure the serial 1/0 interface and set its IP-address to 10.1.1.1 with 255.255.255.0 as its subnet mask

Note: Always remember to set the ‘no shutdown’ attribute for each interface. This way the protocol state will be changed to up rather than down.

Also after setting up the configuration, it is advised to ping a nearby device to check whether the connection has been established.

•



```
R3#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface FastEthernet0/0
R3(config-if)#ip 192.168.3.50
                 ^
% Invalid input detected at '^' marker.

R3(config-if)#ip address 192.168.3.50 255.255.255.0
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Mar  1 00:24:33.479: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:24:34.479: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config)#interface Serial1/1
R3(config-if)#ip address 11.1.1.2 255.255.255.0
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Mar  1 00:25:40.487: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Mar  1 00:25:41.487: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
R3(config)#wr
% Incomplete command.

R3(config)#exit
R3#wr
*Mar  1 00:26:06.347: %SYS-5-CONFIG_I: Configured from console by console
R3#wr
Building configuration...
[OK]
R3#
R3#copy eun
R3#copy run
R3#copy running-config start
Destination filename [startup-config]?
Building configuration...
[OK]
R3#xonfig
Translating "xonfig"

Translating "xonfig"
% Unknown command or computer name, or unable to find computer address
R3#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
```

PC 3:

PC 3 can be used to help send data to other PCs on the network via its router.

In this case PC 3 is connected to R2 via the fastEthernet 0/0 connection.

Here we set the Ip-address of the PC to ‘192.168.3.1/24’ with a gateway of ‘192.168.3.50’

•

```
PC3> ip 192.168.3.1 255.255.255.0 192.168.3.50
Checking for duplicate address...
PC1 : 192.168.3.1 255.255.255.0 gateway 192.168.3.50

PC3> ping 192.168.3.50
84 bytes from 192.168.3.50 icmp_seq=1 ttl=255 time=15.019 ms
84 bytes from 192.168.3.50 icmp_seq=2 ttl=255 time=15.027 ms
84 bytes from 192.168.3.50 icmp_seq=3 ttl=255 time=15.227 ms
84 bytes from 192.168.3.50 icmp_seq=4 ttl=255 time=15.218 ms
84 bytes from 192.168.3.50 icmp_seq=5 ttl=255 time=15.288 ms

PC3> show ip

NAME        : PC3[1]
IP/MASK     : 192.168.3.1/24
GATEWAY     : 192.168.3.50
DNS         :
MAC         : 00:50:79:66:68:02
LPORT       : 10028
RHOST:PORT  : 127.0.0.1:10029
MTU:        : 1500

PC3> ping 192.168.3.1
192.168.3.1 icmp_seq=1 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=2 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=3 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=4 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=5 ttl=64 time=0.001 ms

PC3> ping 192.168.2.1
192.168.2.1 icmp_seq=1 timeout
192.168.2.1 icmp_seq=2 timeout
192.168.2.1 icmp_seq=3 timeout
192.168.2.1 icmp_seq=4 timeout
192.168.2.1 icmp_seq=5 timeout

PC3> ping 192.168.3.1
192.168.3.1 icmp_seq=1 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=2 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=3 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=4 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=5 ttl=64 time=0.001 ms

PC3>
```


Step 4:

Configure and test the ACL on the network system.

• To initially apply the routing protocol for the system:

To configure the network route for the Router R1, we use the RIP protocol. Using router rip – version 2 we set the network route PC 1 and Router 1

```
R1#copy run
R1#copy running-config start
Destination filename [startup-config]?
Building configuration...
[OK]
R1#
R1#router rip
    ^
% Invalid input detected at '^' marker.

R1#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#dire
                    ^
% Invalid input detected at '^' marker.

R1(config-router)#dire
R1(config-router)#netw
R1(config-router)#network 192.168.1.0
R1(config-router)#network 10.0.0.0
R1(config-router)#exit
R1(config)#exit
R1#
*Mar  1 00:33:28.119: %SYS-5-CONFIG_I: Configured from console by console
R1#wr
Building configuration...
[OK]
R1#
R1#
```

•

To configure the network route for the Router 2, we use the RIP protocol. Using router rip – version 2 we set the network route PC 2 and Router 2. This includes the two serial connections (serial 1/0, serial 1/1) and the fastEthernet 0/0 connection.

```
R2#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 10.0.0.0
R2(config-router)#network 11.0.0.0
R2(config-router)#network 192.168.2.0
R2(config-router)#exit
R2(config)#exit
R2#
*Mar  1 00:35:52.779: %SYS-5-CONFIG_I: Configured from console by console
R2#
R2#wr
Building configuration...
[OK]
R2#
```

To check whether the routing has been done and the connection has been made, we use a simple ping. If it returns a successful message, we can assume that the routing of the network has been done for all the routers.

```
R3#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router rip
R3(config-router)#do ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 76/336/1100 ms
R3(config-router)#do ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/79/92 ms
R3(config-router)#exit
R3(config)#exit
R3#wr
Building configuration...
[OK]
R3#
*Mar  1 00:45:43.739: %SYS-5-CONFIG_I: Configured from console by console
R3#access-list 10 deny host 192.168.1.1
          ^
% Invalid input detected at '^' marker.
```

1. To apply Standard ACL:

Here the scenario we are going to use is that whatever data that is being sent by PC 1 should not be received by PC 3, so based on the features of a standard ASL, we know that it needs to be place near the destination.

So by checking the network topology we can say that the best place to create a standard ASL is in Router 2 . We should also note that the number of the standard ASL should be in the range of 1-99, in our case we have chosen 10.

So by configuring the access list to 'deny', we can deny all communication from PC 1 to PC 2.

```
R2(config)#access-list 10 deny host 192.168.1.1
R2(config)#exit
```

```
R2#show access-list
Standard IP access list 10
    10 deny   192.168.1.1
```

Note:

-

The ‘deny’ keyword tends to deny access to the entire network, so we have to configure the access list for the serial interface 1/0 to allow data to be received from other devices other than PC 1.

To achieve this we use the ‘permit’ keyword in the access-list configuration

```
R2(config)#access-list 10 permit any
R2(config)#interface fas
R2(config)#interface serial 1/0
R2(config-if)#ip acc
R2(config-if)#ip acces
R2(config-if)#ip access-group 10 in
R2(config-if)#exit
R2(config)#exit
R2#wr
*Mar  1 01:23:08.271: %SYS-5-CONFIG_I: Configured from console by console
R2#write
Building configuration...
[OK]
```

Here we can observe that when PC 1 tries to send data to PC 3, its access is denied.

```
PC1> ping 192.168.3.50
*11.1.1.2 icmp_seq=1 ttl=253 time=52.192 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=2 ttl=253 time=45.104 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=3 ttl=253 time=52.174 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=4 ttl=253 time=49.092 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.2 icmp_seq=5 ttl=253 time=48.184 ms (ICMP type:3, code:13, Communication administratively prohibited)
```

But when PC 1 tries to send data to PC 2, its access is not denied.

```
PC1> ping 192.168.2.1
192.168.2.1 icmp_seq=1 timeout
192.168.2.1 icmp_seq=2 timeout
84 bytes from 192.168.2.1 icmp_seq=3 ttl=62 time=36.971 ms
84 bytes from 192.168.2.1 icmp_seq=4 ttl=62 time=40.185 ms
84 bytes from 192.168.2.1 icmp_seq=5 ttl=62 time=41.151 ms
```

But when PC 3 tries to send data to PC 1, its access is not denied.

```
PC3> ping 192.168.3.1
192.168.3.1 icmp_seq=1 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=2 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=3 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=4 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=5 ttl=64 time=0.001 ms
```

Based on this observation we can assume that we have successfully used a standard access list.

2. To apply Extended ACL:

Here the scenario we are going to use is that PC 2 should not be able to send data to PC 3.

To achieve this we are going to use Extended ASL. T

he concept of extended ASL states that we need to place it close to its source. So in terms of the network topology and the position of PC 2, the closest place that is near the source would be Router 2 (R3). We should also note that the number of the extended ASL should be in the range of 100-199, in our case we have chosen 141.

So by configuring the access list to ‘deny’, we can deny all communication from PC 2 to PC 3.

-

The difference between the standard ASL and the extended ASL in this regard is that in extended ASL we have to mention both the source and the destination, whereas in standard ASL we have to only mention the destination.

```
R3(config)#access-list 141 deny icmp host 192.168.2.1 host 192.168.3.1
R3(config)#do show access-list
Extended IP access list 141
    10 deny icmp host 192.168.2.1 host 192.168.3.1
```

Note: The ‘deny’ keyword tends to deny access to the entire network, so here we have to initially give permission for other devices to send data to PC 2 and then group the output that is provided by PC 2 via the serial interface 1/1 in order to prevent it from sending any data to PC 3. To achieve this we use the ‘permit’ keyword in the accesslist configuration.

```
R3(config)#access-list 141 permit icmp any any
R3(config)#do show access-list
Extended IP access list 141
    10 deny icmp host 192.168.2.1 host 192.168.3.1
    20 permit icmp any any
R3(config)#interface serial 1/1
R3(config-if)#ip access-gr
R3(config-if)#ip access-group 141 out
R3(config-if)#do show access-list
Extended IP access list 141
    10 deny icmp host 192.168.2.1 host 192.168.3.1
    20 permit icmp any any
R3(config-if)#exit
R3(config)#exit
R3#
```

Here we can observe that when PC 2 tries to send data to PC 3, its access is denied.

```
PC2> ping 192.168.3.1
*192.168.2.50 icmp_seq=1 ttl=255 time=19.206 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=2 ttl=255 time=8.113 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=3 ttl=255 time=6.304 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=4 ttl=255 time=1.128 ms (ICMP type:3, code:13, Communication administratively prohibited)
*192.168.2.50 icmp_seq=5 ttl=255 time=1.194 ms (ICMP type:3, code:13, Communication administratively prohibited)
```

Similarly when PC 3 tries to send data to PC 2, nothing gets through since all communication has been blocked between them.

```
PC3> ping 192.168.2.1
192.168.2.1 icmp_seq=1 timeout
192.168.2.1 icmp_seq=2 timeout
192.168.2.1 icmp_seq=3 timeout
192.168.2.1 icmp_seq=4 timeout
192.168.2.1 icmp_seq=5 timeout
```

But when PC 1 tries to send data to PC 2, its access is not denied.

```
PC1> ping 192.168.2.1
192.168.2.1 icmp_seq=1 timeout
192.168.2.1 icmp_seq=2 timeout
84 bytes from 192.168.2.1 icmp_seq=3 ttl=62 time=36.971 ms
84 bytes from 192.168.2.1 icmp_seq=4 ttl=62 time=40.185 ms
84 bytes from 192.168.2.1 icmp_seq=5 ttl=62 time=41.151 ms
```

Similarly when PC 3 tries to communicate with PC 1, its access is not denied

-

```
PC3> ping 192.168.3.1
192.168.3.1 icmp_seq=1 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=2 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=3 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=4 ttl=64 time=0.001 ms
192.168.3.1 icmp_seq=5 ttl=64 time=0.001 ms
```

Based on this observation we can assume that we have successfully used an extended access list

•

SDN Practical 3

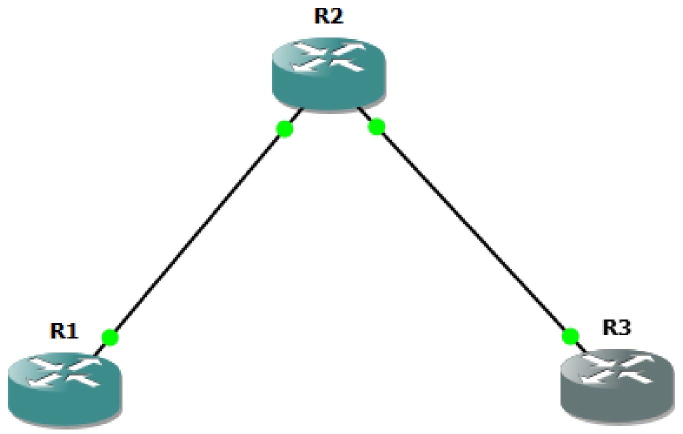
Aim: -

1. Implement a GRE Tunnel

2. Implement NAT

1. Implement a GRE Tunnel

Step 1: Topology



Step 2: Configure the routers

R1

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#banner motd #R1, Implement GRE tunnel#
R1(config)#line con 0
R1(config-line)#exec-timeout 0 0
R1(config-line)#logging synchronous
R1(config-line)#exit
R1(config)#int f0/0
R1(config-if)#ip add 192.168.0.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#
R1(config)#int loopback 0
R1(config-if)#ip add 10.0.0.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#
R1(config)#int loopback 1
R1(config-if)#ip add 172.16.0.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#
R1(config)#router ospf 1
R1(config-router)#network 192.168.0.0 0.0.0.255 area 0
R1(config-router)#network 10.0.0.0 0.0.0.255 area 0
R1(config-router)#exit
R1(config)#exit
```

•

R2

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#hostname R2
R2(config)#banner motd #R2, Implement GRE tunnel#
R2(config)#line con 0
R2(config-line)#exec-timeout 0 0
R2(config-line)#logging synchronous
R2(config-line)#exit
R2(config)#int f0/0
R2(config-if)#ip add 192.168.0.2 255.255.255.0
R2(config-if)#no sh
R2(config-if)#exit
R2(config)#
R2(config)#int  f0/1
R2(config-if)#ip add 192.168.1.1 255.255.255.0
R2(config-if)#no sh
R2(config-if)#exit
R2(config)#
R2(config)#router ospf 1
R2(config-router)#network 192.168.0.0 0.0.0.255 area 0
R2(config-router)#network 192.168.1.0 0.0.0.255 area 0
R2(config-router)#exit
R2(config)#exit
```

R3

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname R3
R3(config)#banner motd #R3, Implement GRE tunnel#
R3(config)#line con 0
R3(config-line)#exec-timeout 0 0
R3(config-line)#logging synchronous
R3(config-line)#exit
R3(config)#int f0/0
R3(config-if)#ip add 192.168.1.2 255.255.255.0
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#
R3(config)#int  loopback 0
R3(config-if)#ip add 10.0.1.1 255.255.255.0
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#
R3(config)#int  loopback 1
R3(config-if)#ip add 172.16.1.1 255.255.255.0
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#
R3(config)#router ospf 1
R3(config-router)#network 192.168.1.0 0.0.0.255 area 0
R3(config-router)#network 10.0.1.0 0.0.0.255 area 0
R3(config-router)#exit
R3(config)#exit
```

•

Step 3: Verify reachability between R1 and R3.

R3 -> R1

```
R3#ping 192.168.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/44/84 ms
R3#ping 10.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/49/84 ms
R3#
```

R1 -> R3

```
R1#ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/64/96 ms
R1#ping 10.0.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/51/68 ms
R1#
```

Step 4:Create an GRE tunnel between R1 and R3.

R1

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int tunnel 0
R1(config-if)#ip add 100.100.100.1 255.255.255.252
R1(config-if)#tunnel source loopback 0
R1(config-if)#tunnel destination 10.0.1.1
R1(config-if)#exit
R1(config)#
R1(config)#ip route 172.16.1.0 255.255.255.0 tunnel 0
*Mar  1 00:15:29.371: %SYS-5-CONFIG_I: Configured from console
R1(config)#ip route 172.16.1.0 255.255.255.0 tunnel 0
*Mar  1 00:15:32.043: %LINEPROTO-5-UPDOWN: Line protocol on Int
R1(config)#ip route 172.16.1.0 255.255.255.0 tunnel 0
R1(config)#
```

-

R3

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#int tunnel 0
R3(config-if)#ip add 100.100.100.2 255.255.255.252
R3(config-if)#tunnel source loopback 0
R3(config-if)#tunnel destination 10.0.0.1
R3(config-if)#exit
R3(config)#
R3(config)#ip route 172.16.0.0 255.255.255.0 tunnel 0
R3(config)#
R3(config)#
*Mar  1 00:15:36.335: %LINEPROTO-5-UPDOWN: Line protocol on In
R3(config)#
```

```
R1#show int tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 100.100.100.1/30
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.1 (Loopback0), destination 10.0.1.1
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:02:05, output 00:02:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     5 packets input, 620 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5 packets output, 620 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
R1#
```

Step 5: verify ping between R1 and R3

R1 -> R3

•

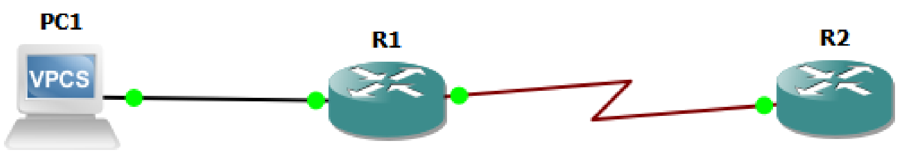
```
R1#ping 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 se
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4
R1#
```

R3 -> R1

```
R3#ping 172.16.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/44/60 ms
R3#
```

2. Implement NAT

Step 1: Topology



Step 2: Configure ISP (Router 2)

-

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#hostname ISP
ISP(config)#enable password cisco
ISP(config)#enable secret class
ISP(config)#line console 0
ISP(config-line)#password cisco
ISP(config-line)#login
ISP(config-line)#exit
ISP(config)#line vty 0 4
ISP(config-line)#password cisco
ISP(config-line)#login
ISP(config-line)#exit
ISP(config)#int loopback 0
ISP(config-if)#ip add 172.16.1.1 255.255.255.255
ISP(config-if)#no sh
ISP(config-if)#exit
ISP(config)#int serial 0/0
ISP(config-if)#ip add 200.2.2.17 255.255.255.252
ISP(config-if)#no sh
ISP(config-if)#clockrate 64000
ISP(config-if)#
```

Step 3: Configure Gateway

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname Gateway
Gateway(config)#enable password cisco
Gateway(config)#enable secret class
Gateway(config)#line console 0
Gateway(config-line)#password cisco
Gateway(config-line)#login
Gateway(config-line)#exit
Gateway(config)#line vty 0 4
Gateway(config-line)#password cisco
Gateway(config-line)#login
Gateway(config-line)#exit
Gateway(config)#int f0/0
Gateway(config-if)#ip add 10.10.10.1 255.255.255.0
Gateway(config-if)#no sh
Gateway(config-if)#exit
Gateway(config)#
Gateway(config)#int serial 0/0
Gateway(config-if)#ip add 200.2.2.18 255.255.255.252
Gateway(config-if)#no sh
Gateway(config-if)#exit
```

Step 4: Configure PC

```
PC1> ip 10.10.10.100/24 10.10.10.1
Checking for duplicate address...
PC1 : 10.10.10.100 255.255.255.0 gateway 10.10.10.1
```

Step 5: Check connectivity from pc to gateway

-

```
PC1> ping 200.2.2.18

84 bytes from 200.2.2.18 icmp_seq=1 ttl=255 time=10.963 ms
84 bytes from 200.2.2.18 icmp_seq=2 ttl=255 time=0.879 ms
84 bytes from 200.2.2.18 icmp_seq=3 ttl=255 time=10.229 ms
84 bytes from 200.2.2.18 icmp_seq=4 ttl=255 time=4.151 ms
84 bytes from 200.2.2.18 icmp_seq=5 ttl=255 time=1.095 ms

PC1>
```

Step 6: Create a static route from the ISP to the Gateway router.

```
ISP#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#ip route 199.99.9.32 255.255.255.224 200.2.2.18
ISP(config)#
```

Step 7: Add a default route from the Gateway router to the ISP router.

```
Gateway#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Gateway(config)#ip route 0.0.0.0 0.0.0.0 200.2.2.17
Gateway(config)#
```

Step 8: Define the pool of usable public IP addresses

ip nat pool public-access 199.99.9.40 199.99.9.62 netmask 255.255.255.224

```
Gateway(config)#ip nat pool public-access ?
  A.B.C.D        Start IP address
  netmask        Specify the network mask
  prefix-length  Specify the prefix length

Gateway(config)#$cess 199.99.9.40 199.99.9.62 netmask 255.255.255.224
Gateway(config)#
```

Step 9: Define an access list that will match the inside private IP addresses

```
Gateway(config)#access-list 1 permit 10.10.10.0 0.0.0.255
Gateway(config)#
```

Step 10: Define the NAT translation from inside list to outside pool

-

```
Gateway(config)#ip nat inside source list 1 pool public-access
Gateway(config)#
```

Step 11: Specify the interfaces

```
Gateway(config)#int f 0/0
Gateway(config-if)#ip nat inside
Gateway(config-if)#exit
Gateway(config)#int serial 0/0
Gateway(config-if)#ip nat outside
Gateway(config-if)#exit
```

Step 12: Testing the configuration

From the PC, ping 200.2.2.17

```
PC1> ping 200.2.2.17

84 bytes from 200.2.2.17 icmp_seq=1 ttl=254 time=10.060 ms
84 bytes from 200.2.2.17 icmp_seq=2 ttl=254 time=9.578 ms
84 bytes from 200.2.2.17 icmp_seq=3 ttl=254 time=9.521 ms
84 bytes from 200.2.2.17 icmp_seq=4 ttl=254 time=3.792 ms
84 bytes from 200.2.2.17 icmp_seq=5 ttl=254 time=10.327 ms

PC1> 200.2.2.17
```

Check the NAT translation on the Gateway router,

```
Gateway#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 199.99.9.40        10.10.10.100       ---                ---
Gateway#
```

•

SDN Practical 4 :

Aim: Implement Inter-VLAN Routing.

Step 1: What is Inter-VLAN Routing.

- Inter-VLAN routing can be defined as a way to forward traffic between different VLAN by implementing a router in the network.
- VLAN logically segments the switch into different subnets, when a router is connected to the switch, an administrator can configure the router to forward the traffic between the various VLANs configured on the switch.
- The user nodes in the VLANs forwards traffic to the router which then forwards the traffic to the destination network regardless of the VLAN configured on the switch.

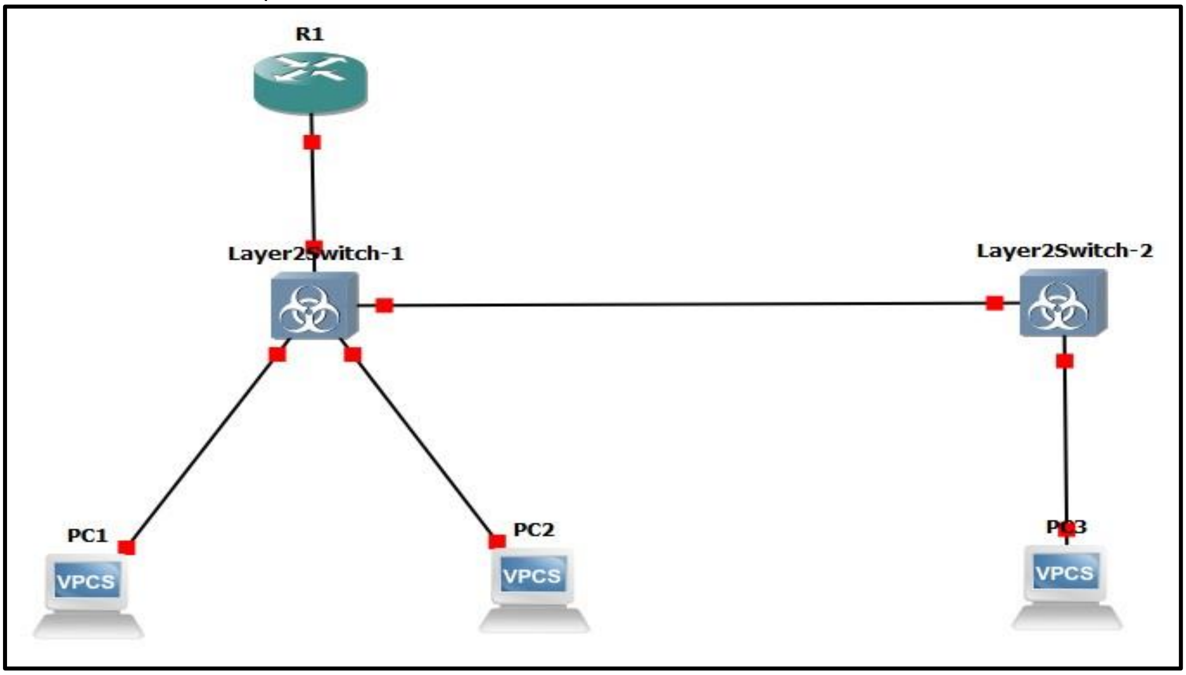
Step 2: Download and install The Layer 2 Switch.

- You can download the Layer 2 Switch from this link: https://www.sysnettechsolutions.com/en/download-vios-l2-image-for-gns3/
- To install it on GNS3 follow the steps given here: https://www.sysnettechsolutions.com/en/add-layer-2-switch-in-gns3/

Step 3 Design the network Topology and set its configuration..

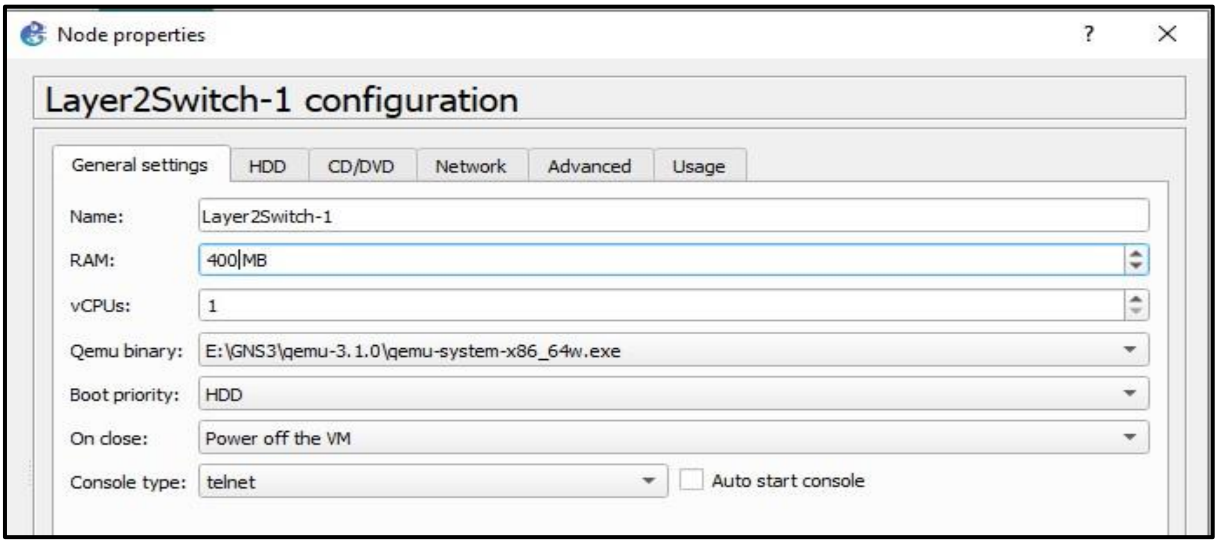
- Create the basic network design.

- Then we set up the wires.

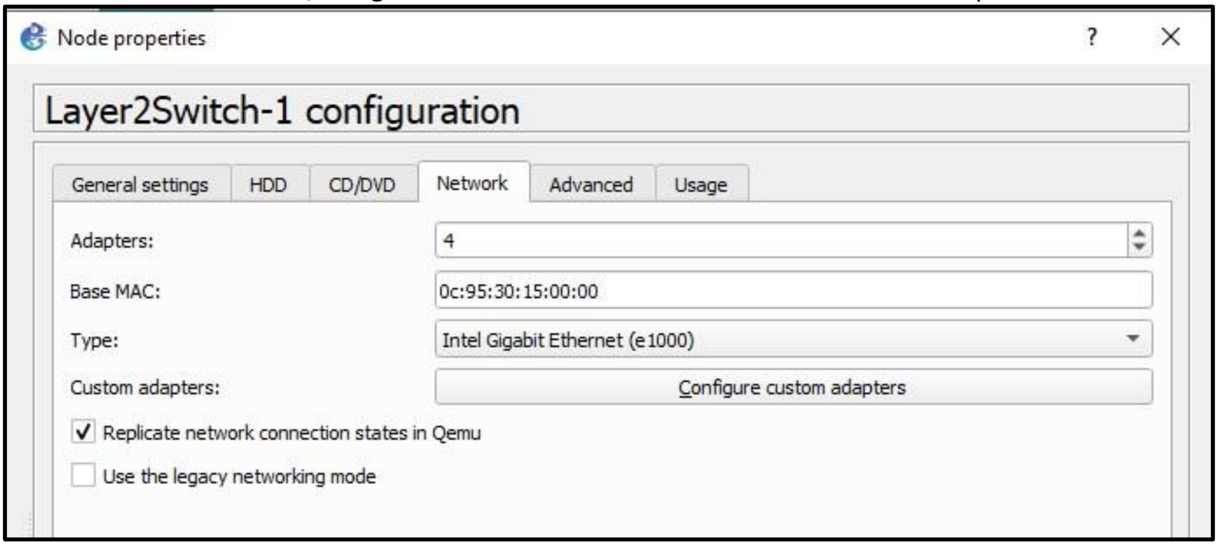


•

Once that is done, configure the Layer2Switch-1. Open its configuration menu. Once that is done set the Cisco L2 switch RAM to 400 MB in the general settings tab.

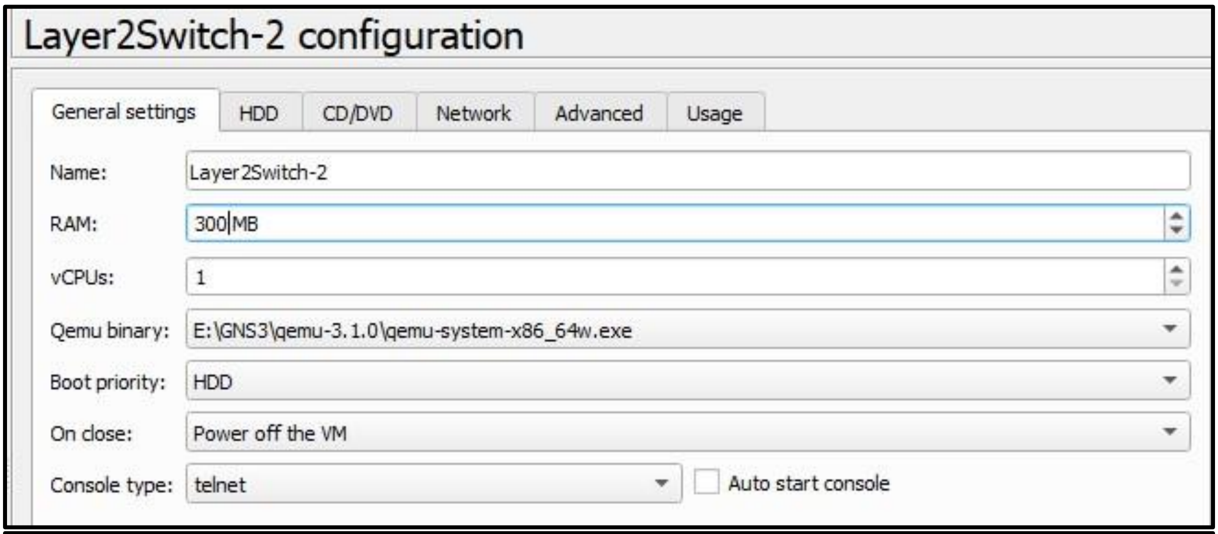


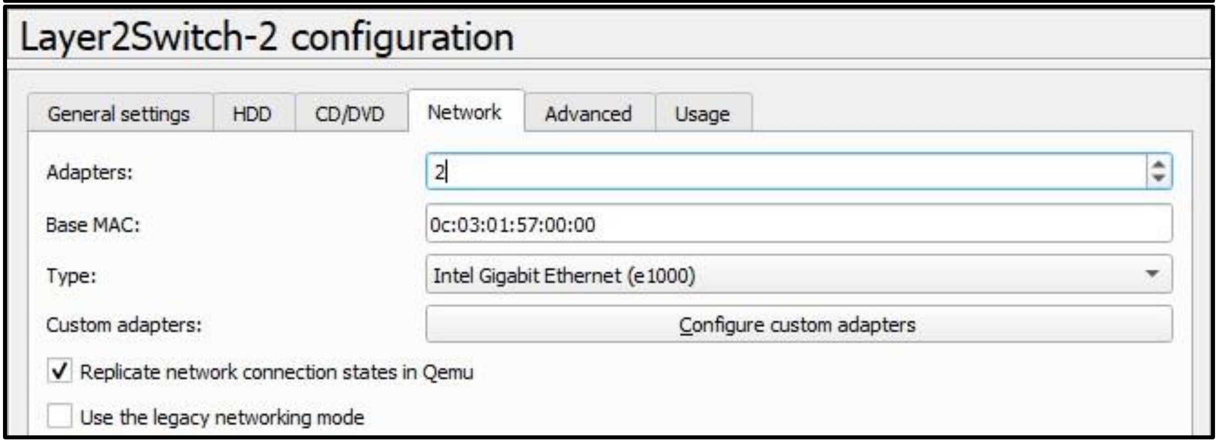
- Once that is done, navigate to the network tab and set the number of adapters to 4.



- Once that is done click on 'Apply' followed by 'OK'.

  Then we configure the Layer 2 Switch-2.
- Set its RAM to 300 MB and its number of adapters to 2.





- Once that is done click on ‘Apply’ followed by ‘OK’.

Then we configure the ip address of PC 1.

-

```
PC1> ip 192.168.5.5/24 192.168.5.1
Checking for duplicate address...
PC1 : 192.168.5.5 255.255.255.0 gateway 192.168.5.1

PC1>
```

```
PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.5.5/24
GATEWAY     : 192.168.5.1
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10014
RHOST:PORT  : 127.0.0.1:10015
MTU:        : 1500

PC1>
```

- Then we configure the ip address of PC 2.

```
PC2> ip 192.168.10.10/24 192.168.10.1
Checking for duplicate address...
PC1 : 192.168.10.10 255.255.255.0 gateway 192.168.10.1

PC2>
```

```
PC2> show ip

NAME        : PC2[1]
IP/MASK     : 192.168.10.10/24
GATEWAY     : 192.168.10.1
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 10012
RHOST:PORT  : 127.0.0.1:10013
MTU:        : 1500

PC2>
```

-

Then we configure the ip address of PC 3.

```
PC3> ip 192.168.5.10/24 192.168.5.1
Checking for duplicate address...
PC1 : 192.168.5.10 255.255.255.0 gateway 192.168.5.1

PC3> show ip

NAME        : PC3[1]
IP/MASK     : 192.168.5.10/24
GATEWAY     : 192.168.5.1
DNS         :
MAC         : 00:50:79:66:68:02
LPORT       : 10016
RHOST:PORT  : 127.0.0.1:10017
MTU:        : 1500

PC3>
```

- Now we configure the L2 Switch-1.

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#vlan 5
vIOS-L2-01(config-vlan)#name IT
vIOS-L2-01(config-vlan)#exit
vIOS-L2-01(config)#vlan 10
vIOS-L2-01(config-vlan)#name SALES
vIOS-L2-01(config-vlan)#exit
vIOS-L2-01(config)#end
vIOS-L2-01#wr
*Dec 20 12:48:32.218: %SYS-5-CONFIG_I: Configured from console by console
vIOS-L2-01#wr
Building configuration...
Compressed configuration from 5041 bytes to 1946 bytes[OK]
*Dec 20 12:48:47.950: %GRUB-5-CONFIG_WRITING: GRUB configuration is being updated on disk. Pleas.
vIOS-L2-01#
*Dec 20 12:48:55.608: %GRUB-5-CONFIG_WRITTEN: GRUB configuration was written to disk successfull.
vIOS-L2-01#
```

- Then we configure the L2 Switch -2.

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#vlan 5
vIOS-L2-01(config-vlan)#name IT
vIOS-L2-01(config-vlan)#exit
vIOS-L2-01(config)#end
vIOS-L2-01#wr
Building configuration...
Compressed configuration from 4857 bytes to 1850 bytes
*Dec 20 12:50:04.637: %SYS-5-CONFIG_I: Configured from console by console[OK]
vIOS-L2-01#
*Dec 20 12:50:13.225: %GRUB-5-CONFIG_WRITING: GRUB configuration is being updated on disk. Pleas.
vIOS-L2-01#
*Dec 20 12:50:14.435: %GRUB-5-CONFIG_WRITTEN: GRUB configuration was written to disk successfull.
vIOS-L2-01#
```

- Now we have to configure trunk and the access interface for L2 Switch-1.

```
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#interface gigabitethernet 0/1
vIOS-L2-01(config-if)#switchport mode  access
vIOS-L2-01(config-if)#switchport access vlan 5
vIOS-L2-01(config-if)#exit
vIOS-L2-01(config)#interface gigabitethernet 0/2
vIOS-L2-01(config-if)#switchport mode access
vIOS-L2-01(config-if)#switchport access vlan 10
vIOS-L2-01(config-if)#exit
vIOS-L2-01(config)#interface gigabitethernet 0/3
vIOS-L2-01(config-if)#switchport trunk encapsulation dot1q
vIOS-L2-01(config-if)#switchport mode trunk
vIOS-L2-01(config-if)#exit
vIOS-L2-01(config)#interface gigabitethernet 0/0
vIOS-L2-01(config-if)#switchport trunk encapsulation dot1q
vIOS-L2-01(config-if)#switchport mode trunk
vIOS-L2-01(config-if)#exit
vIOS-L2-01(config)#exit
vIOS-L2-01#wr
*Dec 20 12:53:40.046: %SYS-5-CONFIG_I: Configured from console by console
Building configuration...
Compressed configuration from 5193 bytes to 2026 bytes[OK]
*Dec 20 12:53:52.125: %GRUB-5-CONFIG_WRITING: GRUB configuration is being updated on disk. Pleas.
vIOS-L2-01#
*Dec 20 12:53:59.161: %GRUB-5-CONFIG_WRITTEN: GRUB configuration was written to disk successfull.
vIOS-L2-01#
```

- Now we have to configure trunk and the access interface for L2 Switch-2.

•

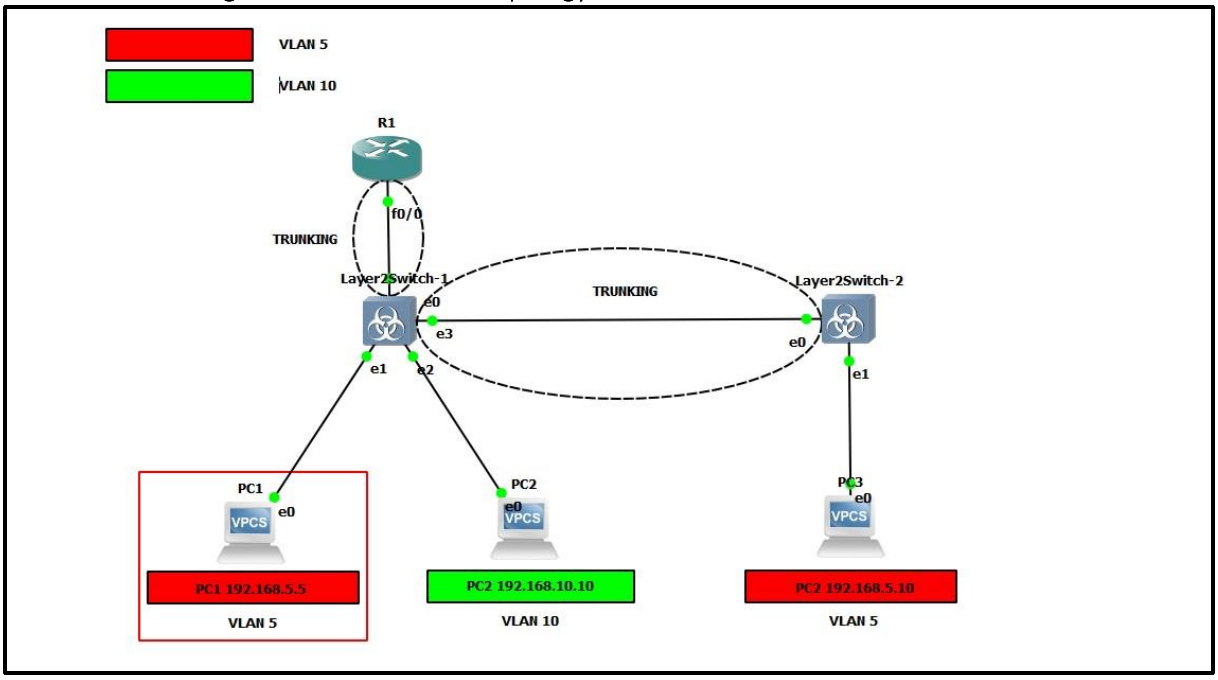
```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#interface gigabitethernet 0/1
vIOS-L2-01(config-if)#switchport mode access
vIOS-L2-01(config-if)#switchport access vlan 5
vIOS-L2-01(config-if)#exit
vIOS-L2-01(config)#interface gigabitethernet 0/0
vIOS-L2-01(config-if)#switchport trunk encapsulation dot1q
vIOS-L2-01(config-if)#switchport mode trunk
vIOS-L2-01(config-if)#end
vIOS-L2-01#wr
Building configuration...
Compressed configuration from 4921 bytes to 1893 bytes
*Dec 20 13:19:40.753: %SYS-5-CONFIG_I: Configured from console by console[OK]
*Dec 20 13:19:48.198: %GRUB-5-CONFIG_WRITING: GRUB configuration is being updated on disk. Pleas.
vIOS-L2-01#
*Dec 20 13:19:49.566: %GRUB-5-CONFIG_WRITTEN: GRUB configuration was written to disk successfull.
vIOS-L2-01#
```

Now we will configure Router 1 (R1).

```
R1#
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface fastEthernet 0/0
R1(config-if)#no shutdown
R1(config-if)#exit
*Mar  1 01:34:06.155: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 01:34:07.155: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed st
ate to up
R1(config-if)#exit
R1(config)#interface fastEthernet 0/0.5
R1(config-subif)#encapsulation dot1q 5
R1(config-subif)#ip ad
*Mar  1 01:35:00.603: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R1(config-subif)#ip address 19
% Incomplete command.

R1(config-subif)#ip address 192.168.5.1 255.255.255.0
R1(config-subif)#no shutdown
R1(config-subif)#exit
R1(config)#interface fastEthernet 0/0.10
R1(config-subif)#encapsulation dot1q 10
R1(config-subif)#ip address 192.168.10.1 255.255.255.0
R1(config-subif)#no shutdown
R1(config-subif)#end
R1#
*Mar  1 01:38:02.299: %SYS-5-CONFIG_I: Configured from console by console
R1#wr
Building configuration...
[OK]
R1#
```

•

- After configuration the network topology should look similar to this.



Step 4: Test the network.

- Now we ping the PC1 to the V2 member PC2 and test the connection. Check that the routing between VLAN's is working properly.

•

```
PC1> ping 192.168.5.1
84 bytes from 192.168.5.1 icmp_seq=1 ttl=255 time=20.456 ms
84 bytes from 192.168.5.1 icmp_seq=2 ttl=255 time=69.384 ms
84 bytes from 192.168.5.1 icmp_seq=3 ttl=255 time=98.680 ms
84 bytes from 192.168.5.1 icmp_seq=4 ttl=255 time=82.232 ms
84 bytes from 192.168.5.1 icmp_seq=5 ttl=255 time=75.012 ms

PC1> ping 192.168.10.1
84 bytes from 192.168.10.1 icmp_seq=1 ttl=255 time=67.339 ms
84 bytes from 192.168.10.1 icmp_seq=2 ttl=255 time=176.354 ms
84 bytes from 192.168.10.1 icmp_seq=3 ttl=255 time=63.449 ms
84 bytes from 192.168.10.1 icmp_seq=4 ttl=255 time=29.438 ms
84 bytes from 192.168.10.1 icmp_seq=5 ttl=255 time=123.141 ms

PC1> ping 192.168.5.10
84 bytes from 192.168.5.10 icmp_seq=1 ttl=64 time=91.187 ms
84 bytes from 192.168.5.10 icmp_seq=2 ttl=64 time=25.624 ms
84 bytes from 192.168.5.10 icmp_seq=3 ttl=64 time=136.011 ms
84 bytes from 192.168.5.10 icmp_seq=4 ttl=64 time=176.647 ms
84 bytes from 192.168.5.10 icmp_seq=5 ttl=64 time=77.996 ms

PC1> ping 192.168.10.10
192.168.10.10 icmp_seq=1 timeout
192.168.10.10 icmp_seq=2 timeout
84 bytes from 192.168.10.10 icmp_seq=3 ttl=63 time=136.551 ms
84 bytes from 192.168.10.10 icmp_seq=4 ttl=63 time=74.037 ms
84 bytes from 192.168.10.10 icmp_seq=5 ttl=63 time=104.810 ms

PC1>
```

- Now we ping from PC2 to PCs that are a member of VLAN5.

```
PC2> ping 192.168.10.1
84 bytes from 192.168.10.1 icmp_seq=1 ttl=255 time=31.528 ms
84 bytes from 192.168.10.1 icmp_seq=2 ttl=255 time=38.096 ms
84 bytes from 192.168.10.1 icmp_seq=3 ttl=255 time=43.545 ms
84 bytes from 192.168.10.1 icmp_seq=4 ttl=255 time=26.123 ms
84 bytes from 192.168.10.1 icmp_seq=5 ttl=255 time=97.194 ms

PC2> ping 192.168.5.5
192.168.5.5 icmp_seq=1 timeout
192.168.5.5 icmp_seq=2 timeout
84 bytes from 192.168.5.5 icmp_seq=3 ttl=63 time=50.090 ms
84 bytes from 192.168.5.5 icmp_seq=4 ttl=63 time=92.386 ms
84 bytes from 192.168.5.5 icmp_seq=5 ttl=63 time=146.650 ms

PC2> ping 192.168.5.10
192.168.5.10 icmp_seq=1 timeout
84 bytes from 192.168.5.10 icmp_seq=2 ttl=63 time=245.134 ms
84 bytes from 192.168.5.10 icmp_seq=3 ttl=63 time=149.907 ms
84 bytes from 192.168.5.10 icmp_seq=4 ttl=63 time=119.149 ms
84 bytes from 192.168.5.10 icmp_seq=5 ttl=63 time=186.286 ms

PC2>
```

Similarly we ping from PC3 to other PCs that are a member of VLAN10.

•

```
PC3> ping 192.168.5.1
84 bytes from 192.168.5.1 icmp_seq=1 ttl=255 time=99.233 ms
84 bytes from 192.168.5.1 icmp_seq=2 ttl=255 time=116.125 ms
84 bytes from 192.168.5.1 icmp_seq=3 ttl=255 time=115.733 ms
84 bytes from 192.168.5.1 icmp_seq=4 ttl=255 time=109.011 ms
84 bytes from 192.168.5.1 icmp_seq=5 ttl=255 time=46.019 ms

PC3> ping 192.168.10.1
84 bytes from 192.168.10.1 icmp_seq=1 ttl=255 time=117.311 ms
84 bytes from 192.168.10.1 icmp_seq=2 ttl=255 time=75.237 ms
84 bytes from 192.168.10.1 icmp_seq=3 ttl=255 time=138.979 ms
84 bytes from 192.168.10.1 icmp_seq=4 ttl=255 time=51.093 ms
84 bytes from 192.168.10.1 icmp_seq=5 ttl=255 time=77.665 ms

PC3> ping 192.168.5.5
84 bytes from 192.168.5.5 icmp_seq=1 ttl=64 time=61.317 ms
84 bytes from 192.168.5.5 icmp_seq=2 ttl=64 time=184.710 ms
84 bytes from 192.168.5.5 icmp_seq=3 ttl=64 time=66.613 ms
84 bytes from 192.168.5.5 icmp_seq=4 ttl=64 time=26.954 ms
84 bytes from 192.168.5.5 icmp_seq=5 ttl=64 time=35.832 ms

PC3> ping 192.168.10.10
192.168.10.10 icmp_seq=1 timeout
192.168.10.10 icmp_seq=2 timeout
84 bytes from 192.168.10.10 icmp_seq=3 ttl=63 time=180.705 ms
84 bytes from 192.168.10.10 icmp_seq=4 ttl=63 time=151.416 ms
84 bytes from 192.168.10.10 icmp_seq=5 ttl=63 time=230.598 ms

PC3>
```

- Now we check that whether PC1 has been configured properly.

```
PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.5.5/24
GATEWAY     : 192.168.5.1
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10014
RHOST:PORT  : 127.0.0.1:10015
MTU:        : 1500

PC1>
```

- Now we check that whether PC2 has been configured properly.

-

```
PC2> show ip

NAME        : PC2[1]
IP/MASK     : 192.168.10.10/24
GATEWAY     : 192.168.10.1
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 10012
RHOST:PORT  : 127.0.0.1:10013
MTU:        : 1500

PC2>
```

- Now we check that whether PC3 has been configured properly.

```
PC3> show ip

NAME        : PC3[1]
IP/MASK     : 192.168.5.10/24
GATEWAY     : 192.168.5.1
DNS         :
MAC         : 00:50:79:66:68:02
LPORT       : 10016
RHOST:PORT  : 127.0.0.1:10017
MTU:        : 1500

PC3>
```

- Now we check the brief of VLAN on Layer 2 Switch – 1.

```
vIOS-L2-01#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active
5    IT                               active    Gi0/1
10   SALES                            active    Gi0/2
100  VLAN100                          active
200  VLAN0200                         active
300  VLAN0300                         active
1002 fddi-default                     act/unsup
1003 trcrf-default                    act/unsup
1004 fddinet-default                  act/unsup
1005 trbrf-default                    act/unsup
vIOS-L2-01#
```

Now we check the brief of VLAN on Layer 2 Switch – 2.

•

```
vIOS-L2-01>enable
vIOS-L2-01#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active
5    IT                               active    Gi0/1
100  VLAN100                          active
200  VLAN0200                         active
300  VLAN0300                         active
1002 fddi-default                     act/unsup
1003 trcrf-default                    act/unsup
1004 fddinet-default                  act/unsup
1005 trbrf-default                    act/unsup
vIOS-L2-01#
```

- Now we check the running configuration of the Layer 2 Switch – 1.

```
vIOS-L2-01#show running-config
Building configuration...

Current configuration : 5193 bytes
!
! Last configuration change at 13:17:03 UTC Mon Dec 20 2021
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service compress-config
!
hostname vIOS-L2-01
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!
!
!
!
!
vtp domain CISCO-vIOS
vtp mode transparent
!
!
!
ip cef
```

•

```
no ipv6 cef
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 5
 name IT
!
vlan 10
 name SALES
!
vlan 100
 name VLAN100
!
vlan 200,300
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/1
 switchport access vlan 5
 switchport trunk encapsulation dot1q
```

```
 switchport autostate exclude
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/2
 switchport access vlan 10
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/3
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
control-plane
!
banner exec ^C
```

-

Now we check the running configuration of the Layer 2 Switch – 2.

```
vIOS-L2-01#show running-config
Building configuration...

Current configuration : 4921 bytes
!
! Last configuration change at 13:19:40 UTC Mon Dec 20 2021
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service compress-config
!
hostname vIOS-L2-01
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!
!
!
!
!
vtp domain CISCO-vIOS
vtp mode transparent
!
!
```

-

```
ip cef
no ipv6 cef
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 5
 name IT
!
vlan 100
 name VLAN100
!
vlan 200,300
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/1
 switchport access vlan 5
 switchport trunk encapsulation dot1q
 media-type rj45
```

```
 negotiation auto
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
control-plane
!
banner exec ^C
```

-

Now we see the interfaces that have been set to the trunk mode in Layer 2 Switch – 1.

```
vIOS-L2-01#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1
Gi0/3       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094
Gi0/3       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,5,10,100,200,300
Gi0/3       1,5,10,100,200,300

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       1,5,10,100,200,300
Gi0/3       1,5,10,100,200,300
vIOS-L2-01#
```

- Now we see the interfaces that have been set to the trunk mode in Layer 2 Switch – 2.

```
vIOS-L2-01#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,5,100,200,300

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       1,5,100,200,300
vIOS-L2-01#
```

- Now we see the overall running configuration of Router 1 (R1).

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

Current configuration : 1805 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
ip tcp synwait-time 5
!
!
!
```

```
no ip domain lookup
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.5
 encapsulation dot1Q 5
 ip address 192.168.5.1 255.255.255.0
!
interface FastEthernet0/0.10
 encapsulation dot1Q 10
 ip address 192.168.10.1 255.255.255.0
!
interface Serial0/0
 no ip address
```

```
 no ip address
 shutdown
 clock rate 2000000
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/2
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/3
```

```
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial1/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
```

```
 no ip address
 shutdown
 serial restart-delay 0
!
!
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end

R1#
```

Practical 5:

Aim: Observe STP Topology Changes and Implement RSTP
1. Implement Advanced STP Modifications and Mechanisms
2. Implement MST

1. Implement Advanced STP Modifications and Mechanisms

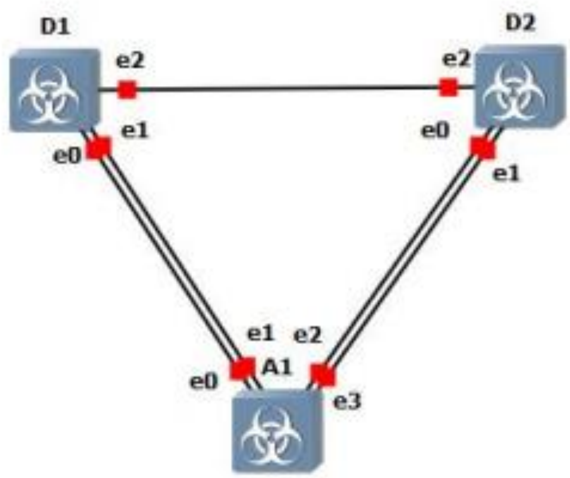
What is STP?

Spanning Tree Protocol (STP) is a Layer 2 network protocol used to prevent looping within a network topology. STP was created to avoid the problems that arise when computers exchange data on a local area network (LAN) that contains redundant paths. If the flow of traffic is not carefully monitored and controlled, the data can be caught in a loop that circles around network segments, affecting performance and bringing traffic to a near halt.

Networks are often configured with redundant paths when connecting network segments. Although redundancy can help protect against disaster, it can also lead to bridge or switch looping. Looping occurs when data travels from a source to a destination along redundant
paths and the data begins to circle around the same paths, becoming amplified and resulting in a broadcast storm.

STP can help prevent bridge looping on LANs that include redundant links. Without STP, it would be difficult to implement that redundancy and still avoid network looping. STP monitors all network links, identifies redundant connections, and disables the ports that can lead to looping.

Step 1: Design the Topology.



Step 2: Configure the network.

- **Configure Switch D1:**

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname D1
D1(config)#spanning-tree mode pvst
D1(config)#banner motd #D1, STP Topology Changes and RSTP Lab#
D1(config)#line con 0
D1(config-line)#exec-timeout 0 0
D1(config-line)#logging synchronous
D1(config-line)#exit
D1(config)#interface range gi0/0-2
D1(config-if-range)#switchport trunk encapsulation dot1q
D1(config-if-range)#switchport mode trunk
D1(config-if-range)#no shutdown
D1(config-if-range)#exit
D1(config)#vlan 2
D1(config-vlan)#name SecondVLAN
D1(config-vlan)#exit
D1(config)#interface vlan 1
D1(config-if)#ip address 10.0.0.1 255
*Jan  6 07:40:56.029: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to d0
D1(config-if)#ip address 10.0.0.1 255.0.0.0
D1(config-if)#no shutdown
D1(config-if)#exit
D1(config)#
*Jan  6 07:41:06.851: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Jan  6 07:41:07.927: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
D1(config)#
```

- **Configure Switch D2:**

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname D2
D2(config)#banner motd #D2, STP Topology Change and RSTP Lab#
D2(config)#spanning-tree mode pvst
D2(config)#line con 0
D2(config-line)#exec-timeout 0 0
D2(config-line)#logging synchronous
D2(config-line)#exit
D2(config)#interface range gi0/0-2
D2(config-if-range)#switchport trunk encapsulation dot1q
D2(config-if-range)#switchport mode trunk
D2(config-if-range)#no shutdown
D2(config-if-range)#exit
D2(config)#vlan 2
D2(config-vlan)#name SecondVLAN
D2(config-vlan)#exit
D2(config)#interface vlan 1
D2(config-if)#ip address 10.1
*Jan  6 07:42:42.744: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to d
D2(config-if)#ip address 10.0.0.2 255.0.0.0
D2(config-if)#no shutdown
D2(config-if)#exit
D2(config)#
*Jan  6 07:42:58.439: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Jan  6 07:43:00.055: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
D2(config)#
```

• **Configure Layer 2 Switch 1 (A1):**

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname A1
A1(config)#banner motd #A1, STP Topology Change and RSTP Lab#
A1(config)#spanning-tree mode pvst
A1(config)#line con 0
A1(config-line)#exec-timeout 0 0
A1(config-line)#logging synchronous
A1(config-line)#exit
A1(config)#interface range gi0/0-3
A1(config-if-range)#switchport trunk encapsulation dot1q
A1(config-if-range)#switchport mode trunk
A1(config-if-range)#no shutdown
A1(config-if-range)#exit
A1(config)#vlan 2
A1(config-vlan)#name SecondVLAN
A1(config-vlan)#exit
A1(config)#interface vlan 1
A1(config-if)#ip address
*Jan  6 07:44:33.203: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to d0
A1(config-if)#ip address 10.0.0.3 255.0.0.0
A1(config-if)#no shutdown
A1(config-if)#exit
A1(config)#
*Jan  6 07:44:50.587: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Jan  6 07:44:51.834: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
A1(config)#
```

Step 3: Discover the Default Spanning Tree.

• **Find the root bridge.**

On A1, issue the command show spanning-tree root and observe what the output tells you about the root bridge. Amongst the lab devices being used to document this lab, A1 shows the root id with a cost of 19 and the root port as interface FastEthernet 0/1 for both VLAN1 and VLAN2.

```
A1#show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0002         32770 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0100         32868 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0200         32968 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0300         33068 0ca6.45e6.0000         4    2   20  15  Gi0/0
A1#
```

The root bridge is elected based upon which switch has the highest Bridge ID (BID). The BID is made up of a configurable priority value (which defaults to 32768) and the base MAC address for the switch. Use the command show spanning-tree root to gather that information from your switches to support the root bridge decision.

```
D1#show spanning-tree root

                                          Root    Hello Max Fwd
Vlan                 Root ID              Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 0ca6.45e6.0000         0      2   20  15
VLAN0002         32770 0ca6.45e6.0000         0      2   20  15
VLAN0100         32868 0ca6.45e6.0000         0      2   20  15
VLAN0200         32968 0ca6.45e6.0000         0      2   20  15
VLAN0300         33068 0ca6.45e6.0000         0      2   20  15
D2#show spanning-tree root

                                          Root    Hello Max Fwd
Vlan                 Root ID              Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 0ca6.45e6.0000         4      2   20  15  Gi0/2
VLAN0002         32770 0ca6.45e6.0000         4      2   20  15  Gi0/2
VLAN0100         32868 0ca6.45e6.0000         4      2   20  15  Gi0/2
VLAN0200         32968 0ca6.45e6.0000         4      2   20  15  Gi0/2
VLAN0300         33068 0ca6.45e6.0000         4      2   20  15  Gi0/2
D2#
A1#show spanning-tree root

                                          Root    Hello Max Fwd
Vlan                 Root ID              Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 0ca6.45e6.0000         4      2   20  15  Gi0/0
VLAN0002         32770 0ca6.45e6.0000         4      2   20  15  Gi0/0
VLAN0100         32868 0ca6.45e6.0000         4      2   20  15  Gi0/0
VLAN0200         32968 0ca6.45e6.0000         4      2   20  15  Gi0/0
VLAN0300         33068 0ca6.45e6.0000         4      2   20  15  Gi0/0
A1#
```

What are the base MAC addresses for the devices we are using? Issue the command show version include MAC (capitalized exactly like that) on each switch.

- **Find the Root Port for each switch.**

  As we saw in the previous output of show spanning-tree root on each switch, the Path Cost can be different amongst switches. In this case, the path cost from A1 to D1 is 19, reflecting connectivity via a FastEthernet port, while the path cost from D2 to D1 is 4, reflecting connectivity via a GigabitEthernet port.

```
D1#show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 0ca6.45e6.0000         0    2   20  15
VLAN0002         32770 0ca6.45e6.0000         0    2   20  15
VLAN0100         32868 0ca6.45e6.0000         0    2   20  15
VLAN0200         32968 0ca6.45e6.0000         0    2   20  15
VLAN0300         33068 0ca6.45e6.0000         0    2   20  15
D1#
D2#show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 0ca6.45e6.0000         4    2   20  15  Gi0/2
VLAN0002         32770 0ca6.45e6.0000         4    2   20  15  Gi0/2
VLAN0100         32868 0ca6.45e6.0000         4    2   20  15  Gi0/2
VLAN0200         32968 0ca6.45e6.0000         4    2   20  15  Gi0/2
VLAN0300         33068 0ca6.45e6.0000         4    2   20  15  Gi0/2
D2#
A1#show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32769 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0002         32770 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0100         32868 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0200         32968 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0300         33068 0ca6.45e6.0000         4    2   20  15  Gi0/0
A1#
```

These are direct connections to the root, so port cost and path cost are the same. This can be seen in the output of show spanning-tree.

```
A1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0ca6.45e6.0000
             Cost        4
             Port        1 (GigabitEthernet0/0)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0ca9.d6ca.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    Shr
Gi0/1               Altn BLK 4         128.2    Shr
Gi0/2               Desg FWD 4         128.3    Shr
Gi0/3               Desg FWD 4         128.4    Shr
```

Our topology does not really illustrate the difference between port cost and path cost very well, so we will introduce a change in the network to achieve this. At D1, shutdown the g1/0/1 interface. The result of this is that D2 will have to change the port it considers root, and we will then see the difference between port cost and path cost.

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface gi0/2
D1(config-if)#shutdown
D1(config-if)#
*Jan  6 07:51:59.728: %LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to administrn
*Jan  6 07:52:00.962: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changgn
D1(config-if)#
```

On D2, issue the command show spanning-tree and you will see the port cost and path cost values separating themselves.

```
D2#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0ca6.45e6.0000
             Cost        8
             Port        1 (GigabitEthernet0/0)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0cbb.4d3b.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root LIS 4         128.1    Shr
Gi0/1               Altn BLK 4         128.2    Shr
Gi0/2               Desg FWD 4         128.3    Shr
```

- **Identify Designated Ports.**

  If you have not already done so, issue the no shutdown command for D1 interface g1/0/1. This will restore our full topology and allow for the non-root attached segment to exist (the links between A1 and D2).

```
D1(config)#interface gi0/2
D1(config-if)#no shutdown
D1(config-if)#exit
D1(config)#
*Jan  6 07:53:44.558: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
*Jan  6 07:53:45.728: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changp
D1(config)#
```

On D2, issue the show spanning-tree command, and you will see that there are two ports now identified as being in the Designated Port role.

And now look at the segments from the A1 side. Issue the show spanning-tree command on A1.

You may have noticed in the previous output that the two links from A1 to D1 were not being used.

Step 4: Implement and Observe Rapid Spanning Tree Protocol.

- **In Part 3, you will implement Rapid Spanning Tree Protocol (RSTP) on all the switches. Using the same basic rules, RSTP speeds up convergence significantly.**

On D2, issue the debug spanning-tree events command, and then issue the shutdown command for interface g1/0/1 and observe the output.

```
D2#debug spanning-tree events
Spanning Tree event debugging is on
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#interface gi0/2
D2(config-if)#shutdown
D2(config-if)#
*Jan  6 07:55:31.934: STP: VLAN0001 new root port Gi0/0, cost 8
*Jan  6 07:55:31.938: STP: VLAN0001 Gi0/0 -> listening
*Jan  6 07:55:31.939: STP[1]: Generating TC trap for port GigabitEthernet0/2
*Jan  6 07:55:31.941: STP: VLAN0002 new root port Gi0/0, cost 8
*Jan  6 07:55:31.941: STP: VLAN0002 Gi0/0 -> listening
*Jan  6 07:55:31.942: STP[2]: Generating TC trap for port GigabitEthernet0/2
*Jan  6 07:55:31.942: STP: VLAN0100 new root port Gi0/0, cost 8
*Jan  6 07:55:31.942: STP: VLAN0100 Gi0/0 -> listening
*Jan  6 07:55:31.942: STP[100]: Generating TC trap for port GigabitEthernet0/2
*Jan  6 07:55:31.942: STP: VLAN0200 new root port Gi0/0, cost 8
*Jan  6 07:55:31.942: STP: VLAN0200 Gi0/0 -> listening
*Jan  6 07:55:31.942: STP[200]: Generating TC trap for port GigabitEthernet0/2
*Jan  6 07:55:31.943: STP: VLAN0300 new root port Gi0/0, cost 8
*Jan  6 07:55:31.944: STP: VLAN0300 Gi0/0 -> listening
*Jan  6 07:55:31.944: STP[300]: Generating TC trap for port GigabitEthernet0/2
D2(config-if)#
*Jan  6 07:55:34.313: %LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to administrn
*Jan  6 07:55:34.320: STP: VLAN0001 sent Topology Change Notice on Gi0/0
*Jan  6 07:55:34.321: STP: VLAN0002 sent Topology Change Notice on Gi0/0
*Jan  6 07:55:34.322: STP: VLAN0100 sent Topology Change Notice on Gi0/0
*Jan  6 07:55:34.322: STP: VLAN0200 sent Topology Change Notice on Gi0/0
*Jan  6 07:55:34.322: STP: VLAN0300 sent Topology Change Notice on Gi0/0
D2(config-if)#
*Jan  6 07:55:35.378: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changn
D2(config-if)#
```

```
D2(config-if)#
*Jan  6 07:55:47.011: STP: VLAN0001 Gi0/0 -> learning
*Jan  6 07:55:47.012: STP: VLAN0002 Gi0/0 -> learning
*Jan  6 07:55:47.012: STP: VLAN0100 Gi0/0 -> learning
*Jan  6 07:55:47.012: STP: VLAN0200 Gi0/0 -> learning
*Jan  6 07:55:47.012: STP: VLAN0300 Gi0/0 -> learning
D2(config-if)#
```

On D1, change the spanning tree mode to rapid-pvst:

```
D1(config)#spanning-tree mode rapid-pvst
D1(config)#
```

On D1, issue the command show spanning-tree.

```
D1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     0ca6.45e6.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0ca6.45e6.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr Peer(STP)
Gi0/1               Desg FWD 4         128.2    Shr Peer(STP)
Gi0/2               Desg FWD 4         128.3    Shr



D1#
```

The type values tell the story. Rapid spanning tree is backwards compatible with common spanning tree. It achieves this backwards compatibility by falling back to using the timers and settings of common spanning tree. In other words, we will not see the benefits of rapid spanning tree if only one switch is running it.

On D2 and A1, change the spanning tree mode to rapid spanning tree.

For D2:(Note that I have only enabled the debugging mode for D2, this is why the output for D2 is different from D1 and A1 when the spanning-tree mode is set to rstp.

```
D2(config)#spanning-tree mode rapid-pvst
D2(config)#
*Jan  6 07:57:31.699: setting bridge id (which=3) prio 32769 prio cfg 32768 sysid 1 (on) id 8000
*Jan  6 07:57:31.700: RSTP(1): initializing port Gi0/0
*Jan  6 07:57:31.702: RSTP(1): Gi0/0 is now designated
*Jan  6 07:57:31.705: RSTP(1): initializing port Gi0/1
*Jan  6 07:57:31.705: RSTP(1): Gi0/1 is now designated
*Jan  6 07:57:31.706: setting bridge id (which=3) prio 32770 prio cfg 32768 sysid 2 (on) id 8000
*Jan  6 07:57:31.706: RSTP(2): initializing port Gi0/0
*Jan  6 07:57:31.706: RSTP(2): Gi0/0 is now designated
*Jan  6 07:57:31.707: RSTP(2): initializing port Gi0/1
*Jan  6 07:57:31.708: RSTP(2): Gi0/1 is now designated
*Jan  6 07:57:31.708: setting bridge id (which=3) prio 32868 prio cfg 32768 sysid 100 (on) id 80
*Jan  6 07:57:31.709: RSTP(100): initializing port Gi0/0
*Jan  6 07:57:31.710: RSTP(100): Gi0/0 is now designated
*Jan  6 07:57:31.712: RSTP(100): initializing port Gi0/1
*Jan  6 07:57:31.712: RSTP(100): Gi0/1 is now designated
*Jan  6 07:57:31.712: setting bridge id (which=3) prio 32968 prio cfg 32768 sysid 200 (on) id 80
*Jan  6 07:57:31.712: RSTP(200): initializing port Gi0/0
*Jan  6 07:57:31.712: RSTP(200): Gi0/0 is now designated
*Jan  6 07:57:31.715: RSTP(200): initializing port Gi0/1
```

```
*Jan  6 07:57:31.715: RSTP(200): Gi0/1 is now designated
*Jan  6 07:57:31.717: setting bridge id (which=3) prio 33068 prio cfg 32768 sysid 300 (on) id 80
*Jan  6 07:57:31.717: RSTP(300): initializing port Gi0/0
*Jan  6 07:57:31.717: RSTP(300): Gi0/0 is now designated
*Jan  6 07:57:31.717: RSTP(300): initializing port Gi0/1
*Jan  6 07:57:31.717: RSTP(300): Gi0/1 is now designated
*Jan  6 07:57:31.727: RSTP(1): transmitting a proposal on Gi0/0
*Jan  6 07:57:31.728: RSTP(1): transmitting a proposal on Gi0/1
*Jan  6 07:57:31.730: RSTP(2): transmitting a proposal on Gi0/0
*Jan  6 07:57:31.730: RSTP(2): transmitting a proposal on Gi0/1
*Jan  6 07:57:31.731:
D2(config)#RSTP(100): transmitting a proposal on Gi0/0
*Jan  6 07:57:31.733: RSTP(100): transmitting a proposal on Gi0/1
*Jan  6 07:57:31.734: RSTP(200): transmitting a proposal on Gi0/0
*Jan  6 07:57:31.734: RSTP(200): transmitting a proposal on Gi0/1
*Jan  6 07:57:31.736: RSTP(300): transmitting a proposal on Gi0/0
*Jan  6 07:57:31.736: RSTP(300): transmitting a proposal on Gi0/1
*Jan  6 07:57:32.887: RSTP(1): transmitting a proposal on Gi0/0
```

```
*Jan  6 07:57:32.888: RSTP(1): transmitting a proposal on Gi0/1
*Jan  6 07:57:32.888: RSTP(2): transmitting a proposal on Gi0/0
*Jan  6 07:57:32.888: RSTP(2): transmitting a proposal on Gi0/1
*Jan  6 07:57:32.891: RSTP(100): transmitting a proposal on Gi0/0
*Jan  6 07:57:32.893: RSTP(100): transmitting a proposal on Gi0/1
*Jan  6 07:57:32.893: RSTP(200): transmitting a proposal on Gi0/0
*Jan  6 07:57:32.893: RSTP(200): transmitting a proposal on Gi0/1
*Jan  6 07:57:32.894: RSTP(300): transmitting a proposal on Gi0/0
*Jan  6 07:57:32.894: RSTP(300): transmitting a proposal on Gi0/1
*Jan  6 07:57:33.455: RSTP(1): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:33.456: RSTP(1): Gi0/0 is now root port
*Jan  6 07:57:33.457: RSTP(1): syncing port Gi0/1
*Jan  6 07:57:33.461: RSTP(2): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:33.461: RSTP(2): Gi0/0 is now root port
*Jan  6 07:57:33.461: RSTP(2): syncing port Gi0/1
*Jan  6 07:57:33.461: RSTP(100): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:33.461: RSTP(100): Gi0/0 is now root port
*Jan  6 07:57:33.461: RSTP(100): syncing port Gi0/1
```

```
*Jan  6 07:57:33.462: RSTP(200): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:33.462: RSTP(200): Gi0/0 is now root port
*Jan  6 07:57:33.462: RSTP(200): syncing port Gi0/1
*Jan  6 07:57:33.463: RSTP(300): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:33.463: RSTP(300): Gi0/0 is now root port
*Jan  6 07:57:33.463: RSTP(300): syncing port Gi0/1
*Jan  6 07:57:33.464: RS
D2(config)#TP(1): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:33.465: RSTP(1): Gi0/1 is now alternate
*Jan  6 07:57:33.469: RSTP(2): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:33.469: RSTP(2): Gi0/1 is now alternate
*Jan  6 07:57:33.470: RSTP(100): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:33.470: RSTP(100): Gi0/1 is now alternate
D2(config)#
*Jan  6 07:57:33.470: RSTP(200): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:33.470: RSTP(200): Gi0/1 is now alternate
*Jan  6 07:57:33.470: RSTP(300): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:33.470: RSTP(300): Gi0/1 is now alternate
*Jan  6 07:57:33.480: STP[1]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:33.480: STP[2]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:33.481: STP[100]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:33.481: STP[200]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:33.481: STP[300]: Generating TC trap for port GigabitEthernet0/0
D2(config)#
*Jan  6 07:57:37.638: RSTP(1): Gi0/0 received a tc ack
*Jan  6 07:57:37.638: RSTP(2): Gi0/0 received a tc ack
*Jan  6 07:57:37.638: RSTP(100): Gi0/0 received a tc ack
*Jan  6 07:57:37.639: RSTP(200): Gi0/0 received a tc ack
*Jan  6 07:57:37.639: RSTP(300): Gi0/0 received a tc ack
D2(config)#
*Jan  6 07:57:43.255: RSTP(1): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:43.255: RSTP(1): Gi0/1 is now root port
*Jan  6 07:57:43.257: RSTP(1): Gi0/0 blocked by re-root
*Jan  6 07:57:43.257: RSTP(1): Gi0/0 is now designated
*Jan  6 07:57:43.258: RSTP(1): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:43.258: RSTP(2): updt roles, received superior bpdu on Gi0/0
```

```
*Jan  6 07:57:43.258: RSTP(2): Gi0/1 is now root port
*Jan  6 07:57:43.258: RSTP(2): Gi0/0 blocked by re-root
*Jan  6 07:57:43.258: RSTP(2): Gi0/0 is now designated
*Jan  6 07:57:43.258: RSTP(100): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:43.258: RSTP(100): Gi0/1 is now root port
*Jan  6 07:57:43.258: RSTP(100): Gi0/0 blocked by re-root
*Jan  6 07:57:43.258: RSTP(100): Gi0/0 is now designated
*Jan  6 07:57:43.258: RSTP(200): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:43.258: RSTP(200): Gi0/1 is now root port
*Jan  6 07:57:43.258: RSTP(200): Gi0/0 blocked by re-root
*Jan  6 07:57:43.258: RSTP(200): Gi0/0 is now designated
*Jan  6 07:57:43.259: RSTP(300): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:43.259: RSTP(300): Gi0/1 is now root port
*Jan  6 07:57:43.259: RSTP(300): Gi0/0 blocked by re-root
*Jan  6 07:57:43.259: RSTP(300): Gi0/0 is now designated
*Jan  6 07:57:43.259: RSTP(2): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:43.259: RSTP(100): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:43.259: RSTP(200): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:43.259: RSTP(300): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:43.259: RSTP(1): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.260: RSTP(2): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.260: RSTP(100): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.263: RSTP(200): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.265: RSTP(300): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.278: STP[
D2(config)#1]: Generating TC trap for port GigabitEthernet0/1
*Jan  6 07:57:43.279: STP[2]: Generating TC trap for port GigabitEthernet0/1
*Jan  6 07:57:43.280: STP[100]: Generating TC trap for port GigabitEthernet0/1
*Jan  6 07:57:43.280: STP[200]: Generating TC trap for port GigabitEthernet0/1
*Jan  6 07:57:43.280: STP[300]: Generating TC trap for port GigabitEthernet0/1
*Jan  6 07:57:43.291: RSTP(1): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.292: RSTP(2): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.294: RSTP(100): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.294: RSTP(200): transmitting a proposal on Gi0/0
*Jan  6 07:57:43.295: RSTP(300): transmitting a proposal on Gi0/0
*Jan  6 07:57:45.449: RSTP(1): updt roles, received superior bpdu on Gi0/0
```

```
*Jan  6 07:57:45.451: RSTP(1): Gi0/0 is now root port
*Jan  6 07:57:45.452: RSTP(1): Gi0/1 blocked by re-root
*Jan  6 07:57:45.452: RSTP(1): Gi0/1 is now designated
*Jan  6 07:57:45.452: RSTP(2): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:45.452: RSTP(2): Gi0/0 is now root port
*Jan  6 07:57:45.452: RSTP(2): Gi0/1 blocked by re-root
*Jan  6 07:57:45.452: RSTP(2): Gi0/1 is now designated
*Jan  6 07:57:45.452: RSTP(100): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:45.452: RSTP(100): Gi0/0 is now root port
*Jan  6 07:57:45.452: RSTP(100): Gi0/1 blocked by re-root
*Jan  6 07:57:45.453: RSTP(100): Gi0/1 is now designated
*Jan  6 07:57:45.454: RSTP(200): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:45.454: RSTP(200): Gi0/0 is now root port
*Jan  6 07:57:45.454: RSTP(200): Gi0/1 blocked by re-root
*Jan  6 07:57:45.454: RSTP(200): Gi0/1 is now designated
*Jan  6 07:57:45.454: RSTP(300): updt roles, received superior bpdu on Gi0/0
*Jan  6 07:57:45.454: RSTP(300): Gi0/0 is now root port
*Jan  6 07:57:45.454: RSTP(300): Gi0/1 blocked by re-root
*Jan  6 07:57:45.454: RSTP(300): Gi0/1 is now designated
*Jan  6 07:57:45.455: RSTP(1): updt roles, receiv
D2(config)#ed superior bpdu on Gi0/1
*Jan  6 07:57:45.455: RSTP(1): Gi0/1 is now alternate
*Jan  6 07:57:45.455: RSTP(2): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:45.456: RSTP(2): Gi0/1 is now alternate
*Jan  6 07:57:45.456: RSTP(100): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:45.456: RSTP(100): Gi0/1 is now alternate
*Jan  6 07:57:45.456: RSTP(200): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:45.456: RSTP(200): Gi0/1 is now alternate
*Jan  6 07:57:45.456: RSTP(300): updt roles, received superior bpdu on Gi0/1
*Jan  6 07:57:45.458: RSTP(300): Gi0/1 is now alternate
*Jan  6 07:57:45.465: STP[1]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:45.465: STP[2]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:45.465: STP[100]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:45.465: STP[200]: Generating TC trap for port GigabitEthernet0/0
*Jan  6 07:57:45.465: STP[300]: Generating TC trap for port GigabitEthernet0/0
D2(config)#
```

For A1:

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#spanning-tree mode rapid-pvst
A1(config)#
```

2. Implement MST.

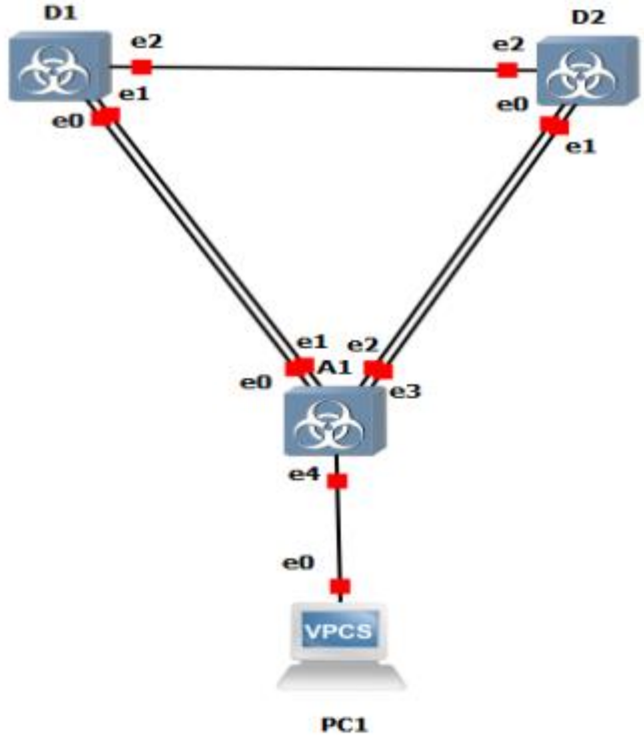
What is MST?

Multiple Spanning Tree Protocol (MSTP), initially defined in IEEE 802.1s and later included in IEEE 802.1Q, supports mapping of multiple VLANs onto a single spanning-tree instance. This reduces the number of spanning-tree instances required in a switched network with many VLANs.

Although RSTP provides faster convergence time than STP does, it still does not solve a problem inherent in STP: all VLANs within a LAN must share the same spanning tree. To solve this problem, the QFX Series products use Multiple Spanning Tree Protocol (MSTP) to create a loop-free topology in networks with multiple spanning-tree regions. An MSTP region allows a group of bridges to be modeled as a single bridge. An MSTP region contains multiple spanning-tree instances (MSTIs). MSTIs provide

different paths for different VLANs. This functionality facilitates more efficient load sharing across redundant links.
An MSTP region can support up to 64 MSTIs, and each instance can support from 1 through 4094 VLANs.

Step 1: Design the Topology.



Step 2: Configure the devices.

- **Configure Switch D1:**

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname D1
D1(config)#banner motd #D1, Multiple Spanning Tree#
D1(config)#spannin-tree mode rapid-pvst
                  ^
% Invalid input detected at '^' marker.

D1(config)#spanning-tree mode rapid-pvst
D1(config)#line con 0
D1(config-line)#exec-timeout 0 0
D1(config-line)#logging synchronous
D1(config-line)#exit
D1(config)#interface range gi0/0-2
D1(config-if-range)#switchport trunk encapsulation dot1q
D1(config-if-range)#switchport mode trunk
D1(config-if-range)#no shutdown
D1(config-if-range)#exit
D1(config)#vlan 2
D1(config-vlan)#name SecondVLAN
D1(config-vlan)#exit
D1(config)#vlan 3
D1(config-vlan)#name ThirdVLAN
D1(config-vlan)#exit
D1(config)#vlan 4
D1(config-vlan)#name FourthVLAN
D1(config-vlan)#exit
D1(config)#vlan 5
D1(config-vlan)#name FifthVLAN
D1(config-vlan)#exit
D1(config)#
```

• **Configure Switch D2:**

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname D2
D2(config)#banner motd #D2, Multiple Spanning Tree#
D2(config)#spanning-tree mode rapid-pvst
D2(config)#line con 0
D2(config-line)#exec-timeout 0 0
D2(config-line)#logging synchronous
D2(config-line)#exit
D2(config)#interface range gi0/0-2
D2(config-if-range)#switchport trunk encapsulation dot1q
D2(config-if-range)#switchport mode trunk
D2(config-if-range)#no shutdown
D2(config-if-range)#exit
D2(config)#vlan 2
D2(config-vlan)#name SecondVLAN
D2(config-vlan)#exit
D2(config)#vlan 3
D2(config-vlan)#name ThirdVLAN
D2(config-vlan)#exit
D2(config)#vlan 4
D2(config-vlan)#name FourthVLAN
D2(config-vlan)#exit
D2(config)#vlan 5
D2(config-vlan)#name FifthVLAN
D2(config-vlan)#exit
D2(config)#
```

• **Configure Switch A1:**

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname A1
A1(config)#banner motd #A1, Multiple Spanning Tree#
A1(config)#spanning-tree mode rapid-pvst
A1(config)#line con 0
A1(config-line)#exec-timeout 0 0
A1(config-line)#logging synchronous
A1(config-line)#exit
```

```
A1(config)#interface range gi0/0-3
A1(config-if-range)#switchport trunk encapsulation dot1q
A1(config-if-range)#switchport mode trunk
A1(config-if-range)#no shutdown
A1(config-if-range)#exit
A1(config)#interface gi1/0
A1(config-if)#no shutdown
A1(config-if)#exit
A1(config)#vlan 2
A1(config-vlan)#name SecondVLAN
A1(config-vlan)#exit
A1(config)#vlan 3
A1(config-vlan)#name ThirdVLAN
A1(config-vlan)#exit
A1(config)#vlan 4
A1(config-vlan)#name FourthVLAN
A1(config-vlan)#exit
A1(config)#vlan 5
A1(config-vlan)#name FifthVLAN
A1(config-vlan)#exit
A1(config)#
```

Step 3: Implement and Observe MST.

• **Configure MST on D1 and D2.**

On D1 and D2, issue the command spanning-tree mode mst.

```
D1(config)#spanning-tree mode mst
D1(config)#
```

```
D2(config)#spanning-tree mode mst
D2(config)#
```

At this point, with no MST-specific configuration, MST Instance 0 is operational for all VLANs. Issue the command show spanning-tree and you will see in the output that the spanning tree information is about MST 0. Issue the command show spanning-tree mst and you will see the MST-specific STP information that is specific to MST 0 only. Take note of the information displayed for interfaces g1/0/5 and g1/0/6 because they are connected to a switch that is not running MST. Their type is listed as P2p Bound (PVST).

```
D1#show spanning-tree

MST0
  Spanning tree enabled protocol mstp
  Root ID    Priority    32768
             Address     0ca6.45e6.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     0ca6.45e6.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg LRN 20000     128.1    Shr Bound(PVST)
Gi0/1               Desg LRN 20000     128.2    Shr Bound(PVST)
Gi0/2               Desg LRN 20000     128.3    Shr


D1#
```

```
D1#show spanning-tree mst

##### MST0    vlans mapped:   1-4094
Bridge        address 0ca6.45e6.0000  priority      32768 (32768 sysid 0)
Root          this switch for the CIST
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Desg FWD 20000     128.1    Shr Bound(PVST)
Gi0/1            Desg FWD 20000     128.2    Shr Bound(PVST)
Gi0/2            Desg FWD 20000     128.3    Shr

D1#
```

The basic behaviour of MST is the same as spanning tree, where a root bridge must be selected, then root ports, and finally best paths to the root bridge from all non-root bridges. In the current network, we can see that D1 has been elected the root bridge. The bridge priority defaults to 32768, so the election is based on D1 having a lower base MAC address. The switch elected as root may be different in your lab topology, but the rules for election remain the same. Issue the command show spanning-tree root on switch A1. Switch A1 is running five instances of spanning tree.

```
A1#show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0002         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0003         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0004         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0005         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0100         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0200         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
VLAN0300         32768 0ca6.45e6.0000         4    2   20  15  Gi0/0
A1#
```

Issue the show spanning-tree root command on D2 and the output will be different. This is because with MST, only one instance of the spanning-tree algorithm runs, regardless of the number of VLANs mapped to it.

```
D2#show spanning-tree root

                                        Root    Hello Max Fwd
MST Instance           Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
MST0             32768 0ca6.45e6.0000         0    2   20  15  Gi0/2
D2#
```

Configure A1 to use MST.

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#spanning-tree mode mst
A1(config)#
```

Step 4: Configure, Tune and Verify Basic MST Operation.

In the last part, you configured all three switches to run MST. In this part, you will further configure, tune, and verify MST to support the unique topological requirements.

- **Create and verify an MST configuration.**

Enter MST configuration mode using the command spanning-tree mst configuration.

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#spanning-tree mst configuration
D1(config-mst)#
```

Configure an MST region name. Our example will be CCNPv8.

```
D1(config-mst)#name CCNPv8
D1(config-mst)#
```

Configure an MST configuration revision number. Our example will be 1.

```
D1(config-mst)#revision 1
D1(config-mst)#
```

Configure instance 1 to include VLAN 2.

```
D1(config-mst)#instance 1 vlan 2
D1(config-mst)#
```

Configure instance 2 to include VLAN 4.

```
D1(config-mst)#instance 2 vlan 4
D1(config-mst)#
```

Commit the configuration by typing exit and returning to global configuration mode.

```
D1(config-mst)#exit
D1(config)#end
D1#
*Jan  6 09:20:13.619: %SYS-5-CONFIG_I: Configured from console by console
D1#
```

Issue the command show spanning-tree mst to verify the configuration is in place.

```
D1#show spanning-tree mst

##### MST0    vlans mapped:   1,3,5-4094
Bridge        address 0ca6.45e6.0000  priority      32768 (32768 sysid 0)
Root          this switch for the CIST
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Desg LRN 20000     128.1    Shr
Gi0/1            Desg LRN 20000     128.2    Shr
Gi0/2            Desg LRN 20000     128.3    Shr

##### MST1    vlans mapped:   2
Bridge        address 0ca6.45e6.0000  priority      32769 (32768 sysid 1)
Root          this switch for MST1

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Desg LRN 20000     128.1    Shr
Gi0/1            Desg LRN 20000     128.2    Shr
Gi0/2            Desg LRN 20000     128.3    Shr

##### MST2    vlans mapped:   4
Bridge        address 0ca6.45e6.0000  priority      32770 (32768 sysid 2)
Root          this switch for MST2

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Desg LRN 20000     128.1    Shr
Gi0/1            Desg LRN 20000     128.2    Shr
Gi0/2            Desg LRN 20000     128.3    Shr

D1#
```

This configuration does not propagate to other switches. Each switch exchanges digest information summarizing the VLAN-to-Instance mappings it has configured. If a switch receives a BPDU with a different digest, it assumes that the sender is in a different MST region. The output below is what A1 shows in the topology used to create this lab. Notice that the ports connected to D1 are classified as P2p Bound (RSTP).

```
A1#show spanning-tree mst

##### MST0    vlans mapped:   1-4094
Bridge        address 0ca9.d6ca.0000  priority      32768 (32768 sysid 0)
Root          address 0ca6.45e6.0000  priority      32768 (32768 sysid 0)
              port    Gi0/0           path cost     20000
Regional Root this switch
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Root FWD 20000     128.1    Shr Bound(RSTP)
Gi0/1            Altn BLK 20000     128.2    Shr Bound(RSTP)
Gi0/2            Desg LRN 20000     128.3    Shr
Gi0/3            Desg FWD 20000     128.4    Shr
Gi1/0            Desg FWD 20000     128.5    Shr

A1#
```

```
A1#show spanning-tree mst configuration digest
Name      []
Revision  0     Instances configured 1
Digest          0xAC36177F50283CD4B83821D8AB26DE62
Pre-std Digest  0xBB3B6C15EF8D089BB55ED10D24DF44DE
A1#
```

And here is the digest from D1:

```
D1#show spanning-tree mst configuration digest
Name      [CCNPv8]
Revision  1     Instances configured 3
Digest          0x746D865FEAD726D8F401F9396B8B62DA
Pre-std Digest  0xDE5D7C8B79A99142EBC0A1C265ED7B05
D1#
```

Configure MST on D1, change the revision number to 2 and add VLAN 3 to instance 1 and VLAN 5 to instance 2.

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#spanning-tree mst configuration
D1(config-mst)#show current
Current MST configuration
Name      [CCNPv8]
Revision  1     Instances configured 3

Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1,3,5-4094
1         2
2         4
-------------------------------------------------------------------------------
D1(config-mst)#revision 2
D1(config-mst)#instance 1 vlan 3
D1(config-mst)#instance 2 vlan 5
D1(config-mst)#show pending
Pending MST configuration
Name      [CCNPv8]
Revision  2     Instances configured 3

Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1,6-4094
1         2-3
2         4-5
-------------------------------------------------------------------------------
D1(config-mst)#exit
D1(config)#end
D1#
*Jan  6 09:22:17.157: %SYS-5-CONFIG_I: Configured from console by console
D1#
```

Now configure D2 and A1 with the same configuration settings (name CCNPv8, revision2, instance 1 vlans 2-3, instance 2 vlans 4-5). After completing the configuration on D2 and A1, the output of show spanning-tree mst on A1 should be similar to the following output. D1 in this case is the root for all 3 instances. D2:

```
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#spanning-tree mst configuration
D2(config-mst)#name CCNPv8
D2(config-mst)#revision 2
D2(config-mst)#instance 1 vlan 2
D2(config-mst)#instance 2 vlan 4
D2(config-mst)#instance 1 vlan 3
D2(config-mst)#instance 2 vlan 5
D2(config-mst)#exit
D2(config)#end
D2#
*Jan  6 09:24:25.738: %SYS-5-CONFIG_I: Configured from console by console
D2#
```

A1:

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#spanning-tree mst configuration
A1(config-mst)#name CCNPv8
A1(config-mst)#revision 2
A1(config-mst)#instance 1 vlan 2
A1(config-mst)#instance 2 vlan 4
A1(config-mst)#instance 1 vlan 3
A1(config-mst)#instance 2 vlan 5
A1(config-mst)#exit
A1(config)#end
A1#
*Jan  6 09:21:55.632: %SYS-5-CONFIG_I: Configured from console by console
A1#
```

```
A1#show spanning-tree mst

##### MST0    vlans mapped:   1,6-4094
Bridge        address 0ca9.d6ca.0000  priority      32768 (32768 sysid 0)
Root          address 0ca6.45e6.0000  priority      32768 (32768 sysid 0)
              port    Gi0/0           path cost     0
Regional Root address 0ca6.45e6.0000  priority      32768 (32768 sysid 0)
                                      internal cost 20000     rem hops 19
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Root FWD 20000     128.1    Shr
Gi0/1            Altn BLK 20000     128.2    Shr
Gi0/2            Desg BLK 20000     128.3    Shr
Gi0/3            Desg BLK 20000     128.4    Shr
Gi1/0            Desg LRN 20000     128.5    Shr

##### MST1    vlans mapped:   2-3
Bridge        address 0ca9.d6ca.0000  priority      32769 (32768 sysid 1)
Root          address 0ca6.45e6.0000  priority      32769 (32768 sysid 1)
              port    Gi0/0           cost          20000     rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Root FWD 20000     128.1    Shr
Gi0/1            Altn BLK 20000     128.2    Shr
Gi0/2            Desg BLK 20000     128.3    Shr
Gi0/3            Desg BLK 20000     128.4    Shr

##### MST2    vlans mapped:   4-5
Bridge        address 0ca9.d6ca.0000  priority      32770 (32768 sysid 2)
Root          address 0ca6.45e6.0000  priority      32770 (32768 sysid 2)
              port    Gi0/0           cost          20000     rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Root FWD 20000     128.1    Shr
Gi0/1            Altn BLK 20000     128.2    Shr
Gi0/2            Desg BLK 20000     128.3    Shr
Gi0/3            Desg BLK 20000     128.4    Shr

A1#
```

- **Controlling the Root Bridge.**

Just like with PVST+ and Rapid PVST+, the current root bridge was elected based on the lowest Bridge ID (consisting of the Priority, extended system ID equal to the VLAN ID, and base MAC address values).
With the priority and extended system IDs being identical, the root bridge's MAC is numerically smaller than the local bridge's MAC. The result is that in a completely un-configured network, one single switch will be elected as the root bridge. The resulting choice of switch may or may not be desirable.
With PVST+ or Rapid PVST+, root bridge selection is done for each VLAN. With MST, the root bridge is based on instances.
There are two basic ways to manipulate the configuration to control the location of the root bridge:

The spanning-tree mst instance-id priority value command can be used to manually set a priority value

The spanning-tree mst instance-id root { primary | secondary } command can be used to automatically set a priority value.

Modify D1 and D2 so that D1 is elected the primary root bridge for MST Instance 1 and D2 is elected the primary root bridge for MST Instance 2. D1 should be elected as the secondary root bridge for MST
Instance 2, and D2 should be elected as the secondary root bridge for MST Instance 1. You will need to make configuration changes on both D1 and D2. The commands used at D1:

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#spanning-tree mst 1 root primary
```

```
D1(config)#spanning-tree mst 2 root secondary
D1(config)#
```

After you have configured both D1 and D2, go to A1 and issue the command show spanning-tree root. In this output, you will see the root bridges differentiated.

```
A1#show spanning-tree root

                                        Root    Hello Max Fwd
MST Instance           Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
MST0             32768 0ca6.45e6.0000         0    2   20  15  Gi0/0
MST1             24577 0ca6.45e6.0000     20000    2   20  15  Gi0/0
MST2             28674 0ca6.45e6.0000     20000    2   20  15  Gi0/0
A1#
```

- **Adjust port cost values to impact root and designated port selection.** As the network is implemented right now, there are two direct paths between switch A1 and the root bridge for each MST. Path and port costs are evaluated to determine the shortest path to the root bridge. In the case where there are multiple equal cost paths to the root bridge, additional attributes must be evaluated. In our case, the lower

interface number (for example, F0/1) is chosen as the Root Port, and the higher interface number (for example, F0/2) is put into a spanning tree Blocking state. You can see which ports are blocked with the show spanning-tree vlan-id command or the show spanning-tree blockedports command. For now, examine VLAN 1 on D1.

On A1, issue the commands show spanning-tree vlan 1 and show spanning-tree blockedports.

```
A1#show spanning-tree mst 1

##### MST1    vlans mapped:   2-3
Bridge        address 0ca9.d6ca.0000  priority      32769 (32768 sysid 1)
Root          address 0ca6.45e6.0000  priority      24577 (24576 sysid 1)
              port    Gi0/0           cost          20000     rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Root FWD 20000     128.1    Shr
Gi0/1            Altn BLK 20000     128.2    Shr
Gi0/2            Desg FWD 20000     128.3    Shr
Gi0/3            Desg FWD 20000     128.4    Shr

A1#show spanning-tree blockedports

Name                 Blocked Interfaces List
-------------------- ------------------------------------
MST0                 Gi0/1
MST1                 Gi0/1
MST2                 Gi0/1

Number of blocked ports (segments) in the system : 3

A1#
```

On A1, shutdown interfaces F0/1 and F0/2, assign a new port cost of 1000 to F0/2 using the spanning- tree mst 1 cost value command, and then issue the no shutdown command on the ports.

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#interface range gi0/0-1
A1(config-if-range)#shutdown
A1(config-if-range)#exit
A1(config)#
*Jan  6 09:25:20.674: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to administrn
*Jan  6 09:25:20.675: %LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to administrn
*Jan  6 09:25:21.801: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changn
*Jan  6 09:25:21.805: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changn
A1(config)#interface gi0/1
A1(config-if)#spanning-tree mst 1 cost 1000
A1(config-if)#exit
A1(config)#interface range gi0/0-1
A1(config-if-range)#no shutdown
A1(config-if-range)#exit
A1(config)#
*Jan  6 09:25:56.279: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Jan  6 09:25:56.283: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
*Jan  6 09:25:57.724: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changp
*Jan  6 09:25:57.725: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changp
A1(config)#
```

Now verify that this impacts root port selection on A1 using the show spanning-tree vlan 1 and show spanning-tree blockedports commands.

```
A1#show spanning-tree mst 1

##### MST1    vlans mapped:   2-3
Bridge        address 0ca9.d6ca.0000  priority      32769 (32768 sysid 1)
Root          address 0ca6.45e6.0000  priority      24577 (24576 sysid 1)
              port    Gi0/1           cost          1000      rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Altn BLK 20000     128.1    Shr
Gi0/1            Root FWD 1000      128.2    Shr
Gi0/2            Desg FWD 20000     128.3    Shr
Gi0/3            Desg FWD 20000     128.4    Shr

A1#show spanning-tree blockedports

Name                 Blocked Interfaces List
-------------------- ------------------------------------
MST0                 Gi0/1
MST1                 Gi0/0
MST2                 Gi0/1

Number of blocked ports (segments) in the system : 3

A1#
```

- **Adjust port priority values to impact root port selection.**

The next method to impact root port selection is configured on the root bridge itself. In our current network topology, A1 has two connections to the root bridge for MST Instance 2, switch D2. The root port has been selected, in this case based on the lowest port ID. Port ID is made up of two values, labelled as Prio (Priority) and Nbr (Number).
The port priority can be any value between 0 and 240, in increments of 16 (older switches may allow setting the priority in different increments).

On A1, issue the command show spanning-tree mst 2 and take note of the port ID values listed.

```
A1#show spanning-tree mst 2

##### MST2    vlans mapped:   4-5
Bridge        address 0ca9.d6ca.0000  priority      32770 (32768 sysid 2)
Root          address 0ca6.45e6.0000  priority      28674 (28672 sysid 2)
              port    Gi0/0           cost          20000     rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Root FWD 20000     128.1    Shr
Gi0/1            Altn BLK 20000     128.2    Shr
Gi0/2            Desg FWD 20000     128.3    Shr
Gi0/3            Desg FWD 20000     128.4    Shr

A1#
```

Modify the port priority of D2 interface G1/0/6 so that it becomes the preferred port by issuing the spanning-tree mst 2 port-priority value interface configuration command. Use a value of 64.

```
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#interface range gi0/0-1
D2(config-if-range)#shutdown
D2(config-if-range)#exit
D2(config)#
*Jan  6 09:31:29.166: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to administrn
*Jan  6 09:31:29.273: %LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to administrn
*Jan  6 09:31:30.166: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changn
D2(config)#
*Jan  6 09:31:30.273: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changn
D2(config)#interface gi0/1
D2(config-if)#spanning-tree mst 2 port-priority 64
D2(config-if)#exit
D2(config)#interface range gi0/0-1
D2(config-if-range)#no shutdown
D2(config-if-range)#exit
D2(config)#
*Jan  6 09:31:59.043: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Jan  6 09:31:59.178: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
D2(config)#
*Jan  6 09:32:00.044: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changp
*Jan  6 09:32:00.178: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changp
D2(config)#
```

On A1, issue the show spanning-tree mst 2 command and you will see that Fa0/4 is now the selected root port. This selection is based on the lower priority value of D2 interface G1/0/6. Notice that the lower priority value does not appear in any A1 output.

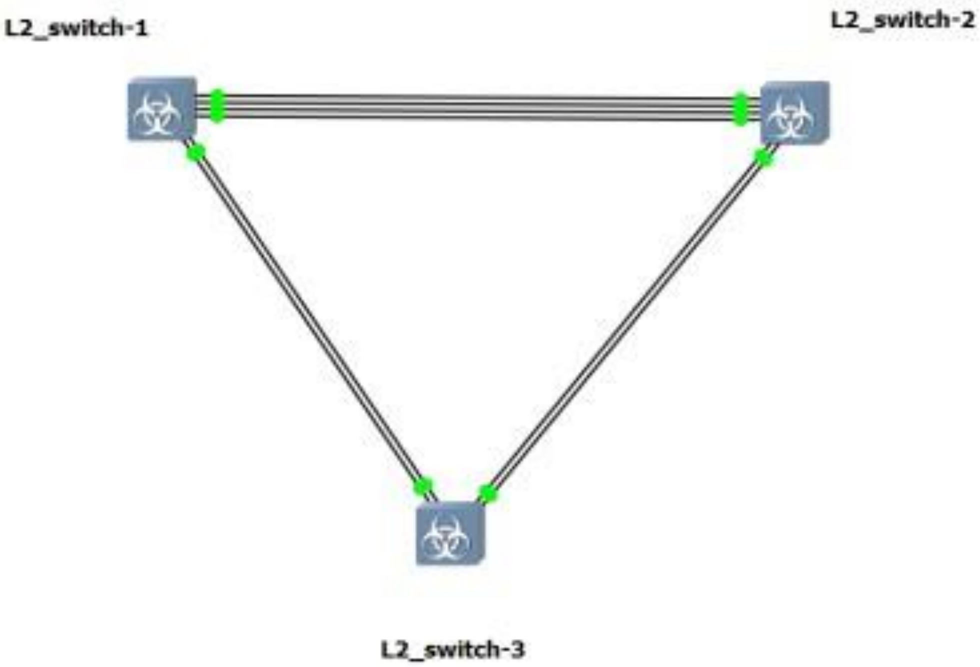
```
A1#show spanning-tree mst 2

##### MST2    vlans mapped:   4-5
Bridge        address 0ca9.d6ca.0000  priority      32770 (32768 sysid 2)
Root          address 0ca6.45e6.0000  priority      28674 (28672 sysid 2)
              port    Gi0/0           cost          20000     rem hops 19

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/0            Root FWD 20000     128.1    Shr
Gi0/1            Altn BLK 20000     128.2    Shr
Gi0/2            Desg FWD 20000     128.3    Shr
Gi0/3            Desg FWD 20000     128.4    Shr

A1#
```

SDN Practical 6

Aim:
1. Implement EtherChannel
2. Tune and Optimize EtherChannel Operations

1. **Implement EtherChannel.**
   What is EtherChannel?
   EtherChannel is a port link aggregation technology in which multiple physical port links are grouped into one logical link. It is used to provide high speed links and redundancy. Maximum of 8 links can be aggregated to form a single logical link.
   EtherChannel protocols – To form an EtherChannel, there are 2 protocols, port aggregation protocol (PAgP) and link aggregation control protocol (LACP).
   Port Aggregation Protocol (PAgP) –
   Port Aggregation Protocol is a Cisco proprietary protocol used to form an EtherChannel.
   There are different modes in which you can configure your interface. These are namely:
   ON: In this mode, the interface will be a part of EtherChannel, but no negotiation takes place.
   Desirable: In this mode, the interface will continuously attempt to convert other side interface into an EtherChannel.
   Auto: In this mode, the interface will become a part of EtherChannel if and only if it is requested by the opposite interface.
   Off: No EtherChannel configured on the interface.

Step 1: Design the Topology.



Step 2: Configure the network.

- **Examine the default port status and manipulate DTP.**
  If the switches are in their default configuration, this connection between the two switches defaults to be an access port in VLAN 1, which can be seen in the output of show interfaces g1/0/5 switchport and show interfaces f0/1 switchport.
  Switch D1:

```
D1#show interfaces f1/2 switchport
Name: Fa1/2
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Disabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
D1#
```

```
A1#show interfaces f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Disabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
A1#
```

Switch A1:

```
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname A1
A1(config)#end
A1#show interfaces gi0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: isl
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
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
Capture Mode Disabled
Capture VLANs Allowed: ALL

 --More--
*Jan  7 17:55:26.234: %SYS-5-CONFIG_I: Configured from console Appliance trust: noe
A1#
A1#
```

Change the administrative mode of interface f0/1 on A1 to Dynamic Desirable with the interface configuration command switchport mode dynamic desirable. After a few moments, check the interface switchport status and you will see that it is in trunk mode. The output of show interfaces trunk will show the protocol as desirable. The output of show interfaces trunk on D1 will show auto.

```
A1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       desirable        n-isl          trunking      1
Gi0/1       desirable        n-isl          trunking      1
Gi0/2       desirable        n-isl          trunking      1
Gi0/3       desirable        n-isl          trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094
Gi0/1       1-4094
Gi0/2       1-4094
Gi0/3       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,100,200,300
Gi0/1       1,100,200,300
Gi0/2       1,100,200,300
Gi0/3       1,100,200,300

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       100,200,300
Gi0/1       100,200,300
Gi0/2       1,100,200,300

A1#
```

```
D1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       desirable        n-isl          trunking      1
Gi0/1       desirable        n-isl          trunking      1
Gi0/2       desirable        n-isl          trunking      1
Gi0/3       desirable        n-isl          trunking      1
Gi1/0       desirable        n-isl          trunking      1
Gi1/1       desirable        n-isl          trunking      1
```

DTP datagrams continue to be sent if the port is set statically to trunk mode. However, if the port is set statically to the access mode, both sending and processing DTP datagrams on that port are deactivated. To see this, configure D1 interface g1/0/6 with the switchport mode trunk command. After a few moments, you should once again see that A1 has automatically negotiated a trunk, this time between f0/2 and D1 g1/0/6.

```
D1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface f1/3
D1(config-if)#switchport trunk encapsulation dot1q
D1(config-if)#switchport mode trunk
D1(config-if)#end
D1#
```

```
A1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       desirable        n-isl          trunking      1
Gi0/1       desirable        n-802.1q       trunking      1
Gi0/2       desirable        n-isl          trunking      1
Gi0/3       desirable        n-isl          trunking      1
```

On A1, shutdown interfaces f0/1 and f0/2 if necessary. Then go to D1 and configure interfaces g1/0/5 and g1/0/6 as trunks with the additional command switchport nonegotiate. A few moments after you re- enable the interfaces at A1, you will see that they do not form trunks with D1.

```
A1(config)#
A1(config)#interface range gi0/0-1
A1(config-if-range)#shutdown
A1(config-if-range)#exit
A1(config)#
*Jan  6 11:58:48.410: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to administrn
*Jan  6 11:58:48.412: %LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to administrn
*Jan  6 11:58:49.409: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changn
*Jan  6 11:58:49.413: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changn
```

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface range gi1/0-1
D1(config-if-range)#switchport trunk encapsulation dot1q
D1(config-if-range)#switchport mode trunk
*Jan  6 11:50:47.138: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changn
*Jan  6 11:50:48.606: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changp
```

```
A1(config)#interface range gi0/0-1
A1(config-if-range)#no shutdown
A1(config-if-range)#exit
A1(config)#
*Jan  6 11:59:24.267: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Jan  6 11:59:24.277: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
*Jan  6 11:59:25.688: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changgp
*Jan  6 11:59:25.690: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changgp
A1(config)#
```

```
A1#show interface trunk
```

```
A1#show interfaces gi0/0 switchport | i Mode
Administrative Mode: dynamic desirable
Operational Mode: trunk
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Capture Mode Disabled
A1#
```

At each switch, issue the global configuration command default interface range first-int-id - last-int-id to reset the interfaces back to their defaults.

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#default interface range gi0/0-1
A1(config)#exit
*Jan  6 12:00:40.311: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to reset
*Jan  6 12:00:40.312: %LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to reset
*Jan  6 12:00:41.371: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changn
*Jan  6 12:00:41.372: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changn
A1(config)#exit
A1#
*Jan  6 12:00:43.704: %SYS-5-CONFIG_I: Configured from console by console
A1#
```

```
D1(config)#default interface range gi1/0-1
D1(config)#
*Jan  6 11:59:34.165: %LINK-5-CHANGED: Interface GigabitEthernet1/0, changed state to reset
*Jan  6 11:59:35.667: %LINK-5-CHANGED: Interface GigabitEthernet1/1, changed state to reset
*Jan  6 11:59:35.671: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changn
D1(config)#
*Jan  6 11:59:37.134: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1, changn
D1(config)#
```

- **Configure Basic Device Settings**
  Switch D1:

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname D1
D1(config)#line con 0
D1(config-line)#exec-timeout 0 0
D1(config-line)#logging synchronous
D1(config-line)#exit
D1(config)#interface range gi1/0-5
                                 ^
% Invalid input detected at '^' marker.

D1(config)#interface range gi0/0-3
D1(config-if-range)#switchport trunk encapsulation dot1q
D1(config-if-range)#switchport mode trunk
D1(config-if-range)#no shutdown
D1(config-if-range)#exit
D1(config)#interface range gi1/0-1
D1(config-if-range)#switchport trunk encapsulation dot1q
D1(config-if-range)#switchport
*Jan  6 16:29:42.543: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changn
*Jan  6 16:29:42.563: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1, changn
*Jan  6 16:29:43.855: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1, changp
*Jan  6 16:29:43.857: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changk
D1(config-if-range)#switchport mode trunk
D1(config-if-range)#no shutdown
D1(config-if-range)#exit
D1(config)#
```

Switch D2:

```
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname D2
D2(config)#banner motd #D2, Implement EtherChannel#
D2(config)#line con 0
D2(config-line)#exec-timeout 0 0
D2(config-line)#logging synchronous
D2(config-line)#exit
D2(config)#interface range gi0/0-3
D2(config-if-range)#switchport trunk encapsulation dot1q
D2(config-if-range)#switchport mode trunk
D2(config-if-range)#no shutdown
D2(config-if-range)#exit
D2(config)#interface range gi1/0-1
D2(config-if-range)#switchport trunk encapsulation dot1q
D2(config-if-range)#switchport mode trunk
D2(config-if-range)#no shutdown
D2(config-if-range)#exit
D2(config)#
```

Switch A1:

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#banner motd #A1, Implement EtherChannel#
A1(config)#line con 0
A1(config-line)#exec-timeout 0 0
A1(config-line)#logging synchronous
A1(config-line)#exit
A1(config)#interface range gi0/0-3
A1(config-if-range)#switchport trunk encapsulation dot1q
A1(config-if-range)#switchport mode trunk
A1(config-if-range)#no shutdown
A1(config-if-range)#exit
A1(config)#
```

Step 3: Configure Static EtherChannel.

In this part, you will configure an EtherChannel without a negotiation protocol. This is against best practices because there is no health check mechanism when the port-channel is statically set to on. The focus for this part is to establish the process for creating and modifying the EtherChannel bundle. For this part you will work with D2 and A1.

- **Configure and verify trunking between D2 and A1.**

  Configure the ports interconnecting D2 and A1 as static trunk ports with the switchport nonegotiate command (the startup configuration has the commands to make them a trunk).

  Verify the trunks have formed.

```
A1#show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1
Gi0/1       on               802.1q         trunking      1
Gi0/2       on               802.1q         trunking      1
Gi0/3       on               802.1q         trunking      1
```

- **Configure and verify a static EtherChannel link between D2 and A1.**

  Add the command channel-group 1 mode on to all the trunk interfaces between D2 and A1.

```
D2(config)#interface range gi1/0-1
D2(config-if-range)#channel-group 1 mode on
Creating a port-channel interface Port-channel 1

D2(config-if-range)#
```

```
D2(config-if-range)#
*Jan  6 16:38:12.208: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed stt
D2(config-if-range)#exit
D2(config)#
```

```
A1(config)#interface range gi0/2-3
A1(config-if-range)#channel-group 1 mode on
Creating a port-channel interface Port-channel 1

A1(config-if-range)#e
*Jan  6 16:40:12.351: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed std
A1(config-if-range)#exit
A1(config)#
```

Verify the EtherChannel has formed by examining the output of the show etherchannel summary command. Also check the spanning tree status. You will see that there is a change to the topology because Po1 replaced interfaces F0/3 and F0/4 with a lower cost.

```
A1#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)          -        Gi0/2(P)    Gi0/3(P)

A1#
```

```
A1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0c93.321a.0000
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0c93.321a.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/1               Desg FWD 4         128.2    Shr
Po1                 Desg FWD 3         128.65   Shr


A1#
```

- **Make a change to the EtherChannel.**
  With very few exceptions, changes to the EtherChannel configuration (whether a negotiation protocol is used or not) must be made at the port-channel level. Changes you make directly to the member interfaces of a port-channel may create synchronization issues that will cause the group to fail or underperform.

  On D2 and A1, create VLAN 999 with the name NATIVE_VLAN.

```
D2(config)#vlan 999
D2(config-vlan)#name NATIVE_VLAN
D2(config-vlan)#exit
D2(config)#
```

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#vlan 999
A1(config-vlan)#name NATIVE_VLAN
A1(config-vlan)#exit
A1(config)#
```

  On D2 and A1, modify interface port-channel 1 so that it uses VLAN 999 as the native VLAN.

```
D2(config)#interface port-channel 1
D2(config-if)#switchport trunk native vlan 999
D2(config-if)#end
D2#
```

```
A1(config)#interface port-channel 1
A1(config-if)#switchport trunk native vlan 999
A1(config-if)#end
A1#
```

  Verify the change has been applied by examining the output of show interfaces trunk.

```
A1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1
Gi0/1       on               802.1q         trunking      1
Po1         on               802.1q         trunking      999

Port        Vlans allowed on trunk
Gi0/0       1-4094
Gi0/1       1-4094
Po1         1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,100,200,300,999
Gi0/1       1,100,200,300,999
Po1         1,100,200,300,999

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       1,100,200,300,999
Gi0/1       1,100,200,300,999
Po1         1,100,200,300,999
A1#
```

Step 4: Implement EtherChannel Using PAgP.

In this part you will configure an EtherChannel using the Cisco proprietary Port Aggregation Protocol, or PAgP. PAgP works between Cisco switches only. The protocol has two modes - Desirable or Auto. These modes work in a fashion similar to modes of the same name in Dynamic Trunking Protocol; Desirable actively communicates a desire to build an EtherChannel bundle, while Auto passively agrees to a bundle if the switch at the other end desires it. Therefore, if both ends are configured in Auto mode, the bundle will not form.

Additionally, PAgP can be configured for non-silent operation. Normally, PAgP operates in silent mode, and will add interfaces to a bundle without having received PAgP packets from the connected device. An example might be when you are connecting a PAgP bundle to a file server. The file server is not sending traffic, and so the bundle will never be formed. Silent mode, which is the default, would allow the switch to build and use the bundle. The recommended configuration is to add the non-silent option when building connections between PAgP-capable devices. For this part you will work with D1 and A1.

- **Configure and verify trunking between D1 and A1.**

  Verify the trunks are still working.

```
A1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1
Gi0/1       on               802.1q         trunking      1
Po1         on               802.1q         trunking      999
```

- **Configure and verify an EtherChannel using PAgP between D1 and A1.**

  Add the command channel-group 2 mode desirable non-silent to all the trunk interfaces between D1 and A1.

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#interface range gi0/0-1
A1(config-if-range)#channel-group 2 mode desirable non-silent
Creating a port-channel interface Port-channel 2

A1(config-if-range)#end
A1#
```

Verify the EtherChannel has formed by examining the output of the show etherchannel summary command.

```
A1#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         -        Gi0/2(P)    Gi0/3(P)
2      Po2(SD)         PAgP     Gi0/0(I)    Gi0/1(I)

A1#
```

- **Make a change to the EtherChannel.**

  Re-iterating step 3 in Part 2, with very few exceptions, changes to the EtherChannel configuration must be made at the port-channel level. Changes you make directly to the member interfaces of a port-channel may create synchronization issues that will cause the group to fail or underperform.

  On D1, create VLAN 999 with the name NATIVE_VLAN.

```
D1(config)#vlan 999
D1(config-vlan)#name NATIVE_VLAN
D1(config-vlan)#exit
D1(config)#
```

On D1 and A1, modify interface port-channel 2 so that it uses VLAN 999 as the native VLAN.

```
D1(config)#interface port-channel 3
D1(config-if)#switchport trunk native vlan 999
D1(config-if)#end
D1#
```

```
A1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#interface range gi0/0-1
A1(config-if-range)#channel-group 3 mode desirable non-silent
Creating a port-channel interface Port-channel 3

A1(config-if-range)#end
A1#
*Jan  6 16:54:22.356: %SYS-5-CONFIG_I: Configured from console by console
A1#
```

Verify the change has been applied by examining the output of show interfaces trunk | i Port|Po3.

```
D1#show interfaces trunk | i Port| Po3
Port        Mode             Encapsulation  Status        Native vlan
Port        Vlans allowed on trunk
Port        Vlans allowed and active in management domain
Port        Vlans in spanning tree forwarding state and not pruned
D1#
```

Step 5: Implement EtherChannel using LACP.

In this part, you will configure an EtherChannel using the open standard Link Aggregation Control Protocol, or LACP. This protocol also has two modes – Active and Passive. These modes work in a similar fashion to modes of PAgP; the Active mode actively communicates a desire to build an EtherChannel bundle, while the Passive mode passively agrees to a bundle if the switch at the other end initiates it. Therefore, if both ends are configured in passive mode, the bundle will not form. For this part you will work with D1 and D2.

- **Configure and verify trunking between D1 and D2.**

  Verify the trunks are still operational.

```
D2#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1
Gi0/1       on               802.1q         trunking      1
Gi0/2       on               802.1q         trunking      1
Gi0/3       on               802.1q         trunking      1
Po1         on               802.1q         trunking      999
```

- **Configure and verify an EtherChannel using LACP between D1 and D2.**

  Add the command channel-group 4 mode active to all the trunk interfaces between D1 and D2.

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface range gi0/0-3
D1(config-if-range)#channel-group 4 mode active
Creating a port-channel interface Port-channel 4

D1(config-if-range)#
```

```
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#interface range gi0/0-3
D2(config-if-range)#channel-group 4 mode active
Creating a port-channel interface Port-channel 4
```

Verify the EtherChannel has formed by examining the output of the show etherchannel summary command. Also check the spanning tree status. You will see that the two interfaces are no longer referenced by Spanning Tree, but the port-channel is. Because there is only one (logical) trunk between D1 and D1, there are no Spanning Tree blocked ports.

```
D1#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 4
Number of aggregators:           4

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SD)         -
2      Po2(RD)         -
3      Po3(SD)         -        Gi1/0(D)    Gi1/1(D)
4      Po4(SU)         LACP     Gi0/0(P)    Gi0/1(P)    Gi0/2(P)
                                Gi0/3(P)

D1#
```

```
D2#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 2
Number of aggregators:           2

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         -        Gi1/0(P)    Gi1/1(P)
4      Po4(SU)         LACP     Gi0/0(P)    Gi0/1(P)    Gi0/2(P)
                                Gi0/3(P)

D2#
```

- **Make a change to the EtherChannel.**
  Once again, it is important to understand that changes to the EtherChannel configuration must be made at the port-channel level. Changes you make directly to the member interfaces of a port-channel may create synchronization issues that will cause the group to fail or underperform.

  On D1 and D2, modify interface port-channel 4 so that it uses VLAN 999 as the native VLAN.

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface port-channel 4
D1(config-if)#switchport trunk native vlan 999
D1(config-if)#exit
D1(config)#end
D1#
```

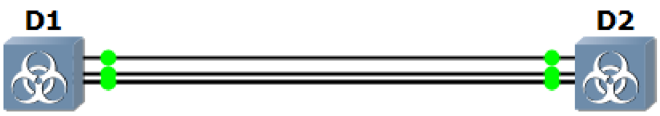
```
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#interface port-channel
*Jan  6 16:58:05.921: %PLATFORM-5-SIGNATURE_VERIFIED: Image 'flash0:/vios_l2-adventerprisek9-m'n
D2(config)#interface port-channel 4
D2(config-if)#switchport trunk native vlan 999
D2(config-if)#exit
D2(config)#end
D2#
```

Verify the change has been applied by examining the output of show interfaces trunk | i Port|Po4

```
D1#show interfaces trunk | i Port|Po4
Port        Mode             Encapsulation  Status        Native vlan
Po4         on               802.1q         trunking      999
Port        Vlans allowed on trunk
Po4         1-4094
Port        Vlans allowed and active in management domain
Po4         1,100,200,300,999
Port        Vlans in spanning tree forwarding state and not pruned
Po4         1,100,200,300,999
D1#
```

```
D2#show interfaces trunk | i Port|Po4
Port        Mode             Encapsulation  Status        Native vlan
Po4         on               802.1q         trunking      999
Port        Vlans allowed on trunk
Po4         1-4094
Port        Vlans allowed and active in management domain
Po4         1,100,200,300,999
Port        Vlans in spanning tree forwarding state and not pruned
Po4         1,100,200,300,999
D2#
```

2. **Tune and Optimize EtherChannel Operations.**
   Step 1: Design the Topology.



Step 2: Configure the network.

- **Switch D1:**

```
D1(config)#hostname D1
D1(config)#banner motd #D1, Tuning EtherChannel#
D1(config)#spanning-tree mode rapid-pvst
D1(config)#line con 0
D1(config-line)#exec-timeout 0 0
D1(config-line)#logging synchronous
D1(config-line)#exit
D1(config)#interface range gi0/0-3
D1(config-if-range)#switchport trunk encapsulation dot1q
D1(config-if-range)#switchport mode trunk
D1(config-if-range)#no shutdown
D1(config-if-range)#exit
```

- **Switch D2:**

```
vIOS-L2-01>
vIOS-L2-01>
vIOS-L2-01>enable
vIOS-L2-01#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
vIOS-L2-01(config)#hostname D2
D2(config)#banner motd #D2, Tuning EtherChannel#
D2(config)#spanning-tree mode rapid-pvst
D2(config)#line con 0
D2(config-line)#exec-timeout 0 0
D2(config-line)#logging synchronous
D2(config-line)#exit
D2(config)#interface range gi0/0-3
D2(config-if-range)#switchport trunk encapsulation dot1q
D2(config-if-range)#switchport mode trunk
D2(config-if-range)#no shutdown
D2(config-if-range)#exit
D2(config)#
```

Step 3: Tune LACP-based EtherChannel's.

An EtherChannel bundle using LACP as its negotiation protocol can have as many as 16 assigned members with 8 active ports passing traffic, and the other 8 ports on standby. The switches involved in a LACP bundle negotiate a master/slave relationship and the designated master switch makes the decisions on which members are active and which are in "hot standby" mode when the number of members in the bundle exceeds 8.

The minimum and maximum number of ports allowed to be involved in a port channel can be managed through configuration as well.

In this part of the lab, you will do just that. For the group of links connecting D1 and D2, you will set up an EtherChannel bundle using LACP as the negotiation protocol, with a minimum of 2 links and a maximum of 3. As a part of this configuration, you will control which switch is the master. Next, you will enable LACP fast packets, reducing the time out period from 30 seconds to 1 second.

- **Configure master switch criteria.**

  Each switch connected using LACP has a system ID value. Those numbers are compared, and the switch with the lowest number is considered the master. The system ID value is a combination of a system priority that defaults to 32768 and the base MAC address. Unlike spanning tree, the priority value for LACP does not have be scaled by multiples of 4096.

  Use the show lacp sys-id privileged EXEC command to see what the sys-id value is for D1 and D2. Based on the output shown below, we can deduce that D1 would be the bundle master if all default values remained unchanged.

```
D1#show lacp sys-id
32768, 0c67.92fa.0000
D1#
D1#
```

```
D2#show lacp sys-id
32768, 0cc1.b1d6.0000
D2#
```

On D2, modify the lacp sys-id by changing the system priority. Use the lacp system-priority value global configuration command to set the value to 1, and then verify that the value has been changed.

```
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#lacp system-priority 1
D2(config)#exit
D2#show lacp sys-id
1, 0cc1.b1d6.0000
D2#
```

- **Configure bundle size and member preferences.**
  By default, interfaces are selected to be included in the active bundle based on their interface id. For a given configuration, the lower numbered interfaces are added to the bundle until the bundle has reached its maximum size. Any interfaces that remain are put in hot standby mode.

  Issue the shutdown command for the interfaces connecting D1 and D2.

```
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface range gi0/0-3
D1(config-if-range)#shutdown
D1(config-if-range)#
*Jan  7 17:10:45.728: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to administratively down
*Jan  7 17:10:45.730: %LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to administratively down
*Jan  7 17:10:45.731: %LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to administratively down
*Jan  7 17:10:46.158: %LINK-5-CHANGED: Interface GigabitEthernet0/3, changed state to administratively down
*Jan  7 17:10:46.799: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down
*Jan  7 17:10:46.799: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Jan  7 17:10:46.799: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to down
D1(config-if-range)#
*Jan  7 17:10:47.210: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/3, changed state to down
D1(config-if-range)#
```

```
*Jan  7 17:10:09.481: %SYS-5-CONFIG_I: Configured from console by console
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#interface range gi0/0-3
D2(config-if-range)#shutdown
D2(config-if-range)#
*Jan  7 17:10:51.148: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to administratively down
*Jan  7 17:10:51.728: %LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to administratively down
*Jan  7 17:10:51.728: %LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to administratively down
*Jan  7 17:10:51.728: %LINK-5-CHANGED: Interface GigabitEthernet0/3, changed state to administratively down
*Jan  7 17:10:52.473: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down
D2(config-if-range)#
*Jan  7 17:10:52.978: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Jan  7 17:10:52.981: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to down
*Jan  7 17:10:52.982: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/3, changed state to down
```

Configure the connections between D1 and D2 into a single LACP EtherChannel bundle. Use Channel Group number 12 and the Active mode. Configure the interfaces for LACP Fast.

```
D1(config-if-range)#channel-group 12 mode active
Creating a port-channel interface Port-channel 12

D1(config-if-range)#
```

```
D2(config-if-range)#channel-group 12 mode active
Creating a port-channel interface Port-channel 12

D2(config-if-range)#
```

Issue the no shutdown command for the interfaces connecting D1 and D2.

```
D1(config-if-range)#no shutdown
D1(config-if-range)#exit
D1(config)#
*Jan  7 17:13:30.292: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Jan  7 17:13:30.296: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
*Jan  7 17:13:30.298: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
*Jan  7 17:13:30.301: %LINK-3-UPDOWN: Interface GigabitEthernet0/3, changed state to up
*Jan  7 17:13:31.339: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
*Jan  7 17:13:31.343: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
*Jan  7 17:13:31.343: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up
*Jan  7 17:13:31.343: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/3, changed state to up
D1(config)#
```

```
D2(config-if-range)#no shutdown
D2(config-if-range)#exit
D2(config)#
*Jan  7 17:11:38.483: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Jan  7 17:11:38.492: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
*Jan  7 17:11:38.495: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
*Jan  7 17:11:38.498: %LINK-3-UPDOWN: Interface GigabitEthernet0/3, changed state to up
*Jan  7 17:11:39.541: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
*Jan  7 17:11:39.541: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
*Jan  7 17:11:39.542: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up
*Jan  7 17:11:39.542: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/3, changed state to up
D2(config)#
```

On D1 and D2, configure the port-channel 12 interface with a LACP minimum bundle size of 2 interfaces, and the maximum bundle size of 3 interfaces.

```
D1(config)#interface port-channel 12
D1(config-if)#lacp max-bundle 3
D1(config-if)#end
D1#
```

```
D2(config)#interface port-channel 12
D2(config-if)#lacp
*Jan  7 17:11:56.488: %LINK-3-UPDOWN: Interface Port-channel12, changed state to down
*Jan  7 17:11:58.241: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel12, changed state to down max-bundle 3
D2(config-if)#lacp max-bundle 3
*Jan  7 17:12:05.495: %IDBMAN-3-INVALIDAGGPORTBANDWIDTH: Port-channel12(16 / 0) has an invalid bandwidth value of 0
D2(config-if)#
*Jan  7 17:12:07.760: %IDBMAN-3-INVALIDAGGPORTBANDWIDTH: Port-channel12(16 / 0) has an invalid bandwidth value of 0
*Jan  7 17:12:07.767: %LINK-3-UPDOWN: Interface Port-channel12, changed state to upend
D2#
```

Verify that the EtherChannel bundle has formed and take note of the ports that are included versus the port that is in hot standby mode.

```
D1#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
12     Po12(SU)        LACP      Gi0/0(P)    Gi0/1(P)    Gi0/2(P)
                                 Gi0/3(w)

D1#
```

```
D2#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
12     Po12(SU)        LACP      Gi0/0(P)    Gi0/1(P)    Gi0/2(P)
                                 Gi0/3(H)

D2#
```

Verify the mode, state, and rate of LACPDU being sent for port members.

```
D2#show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 12
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi0/0     SA      bndl      32768         0xC       0xC     0x1         0x3D
Gi0/1     SA      bndl      32768         0xC       0xC     0x2         0x3D
Gi0/2     SA      bndl      32768         0xC       0xC     0x3         0x3D
Gi0/3     SA      hot-sby   32768         0xC       0xC     0x4         0x5
D2#
```

```
D1#show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 12
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi0/0     SA      bndl      32768         0xC       0xC     0x1         0x3D
Gi0/1     SA      bndl      32768         0xC       0xC     0x2         0x3D
Gi0/2     SA      bndl      32768         0xC       0xC     0x3         0x3D
Gi0/3     SA        -       32768         0xC       0xC     0x4         0x5
D1#
```

Step 4: Explore EtherChannel Load Balancing.

The load balancing method used to send traffic through an EtherChannel is a global setting on the switch. All EtherChannel’s on a given switch will use the method selected for that switch. The load balancing methods used at either end of an EtherChannel bundle do not have to match.

The available methods, as well as the default method used, varies by hardware platform. By default, Cisco Catalyst 3650 and Catalyst 2960 switches load-balance using the source MAC address.

```
Gi0/3      SA        -        32768          0xC        0xC      0x4
D1#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address

D1#
```

```
Gi0/3      SA       not-sby   32768          0xC        0xC      0x4
D2#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address

D2#
```

The full form of the command to show what interface the load-balancing algorithm will choose is show platform software fed etherchannel [switch switch-number] channel-group-number {group-mask | load- balance mac src-mac dst-mac [ip src-ip dst-ip [port src-port dst-port]]}. Select a channel-group and specify the source and destination address (in the form of a MAC, IP, or Port number) and the switch tells you what the exit port will be. The example that follows shows that traffic with the same source MAC always exits using the same interface.
Use this command on your switches to verify how the default load-balancing scheme is working, and then try out the other load-balancing mechanisms to see how the behaviour changes.

SDN Practical 7

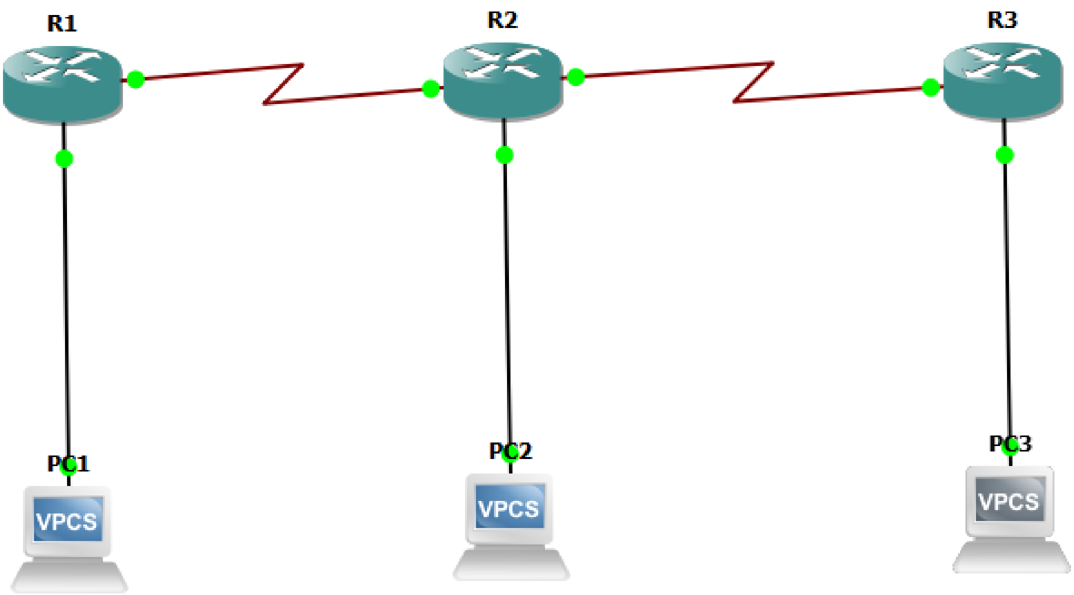
Aim : OSPF Implementation

1. **Implement Single Area OSPFv2**
2. **Implement Multi-Area OSPFv2**
3. **OSPFv2 Route Summarization and Filtering**
4. **Implement Multi-Area OSPFv3**

1. Implement Single Area OSPFv2 :

Step 1 : For Single Area OSPFv2, I will use 3 routers and 3 virtual PCs.

Step 2 : Design of the Topology is as follows :-



Step 3 : Start all the devices and then configure the network using console.

- Configuring Router 1.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 192.168.1.0 255.255.255.0
Bad mask /24 for address 192.168.1.0
R1(config-if)#ip address 192.168.1.50 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#ex
*Mar  1 00:15:54.331: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:15:55.331: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#exit
R1(config)#interface serial 1/0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
*Mar  1 00:18:00.663: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:18:01.663: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R1(config)#
```

- Configuring Router 2.

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface fastEthernet 0/0
R2(config-if)#ip address 192.168.2.50 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
*Mar  1 00:22:37.611: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:22:38.611: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
R2(config)#interface serial 1/0
R2(config-if)#ip address 10.1.1.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#ip address 10.1.1.2 255.255.255.0
*Mar  1 00:23:37.955: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:23:38.955: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R2(config-if)#exit
R2(config)#interface serial 1/1
R2(config-if)#ip address 11.1.1.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit
*Mar  1 00:24:20.175: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Mar  1 00:24:21.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
R2(config-if)#exit
R2(config)#
```

- Configuring Router 3.

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface fastEthernet 0/0
R3(config-if)#ip address 192.168.3.50 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#ex
*Mar  1 00:27:58.383: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:27:59.383: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to
R3(config-if)#exit
R3(config)#interface serial 1/0
R3(config-if)#ip address 11.1.1.2 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
*Mar  1 00:29:00.599: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:29:01.599: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R3(config-if)#exit
R3(config)#
```

- Configure PC 1:

```
PC1> ip 192.168.1.1/24 192.168.1.50
Checking for duplicate address...
PC1 : 192.168.1.1 255.255.255.0 gateway 192.168.1.50

PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.1.1/24
GATEWAY     : 192.168.1.50
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10026
RHOST:PORT  : 127.0.0.1:10027
MTU:        : 1500
```

- Configure PC 2:

```
PC2> ip 192.168.2.1/24 192.168.2.50
Checking for duplicate address...
PC1 : 192.168.2.1 255.255.255.0 gateway 192.168.2.50

PC2> show ip

NAME        : PC2[1]
IP/MASK     : 192.168.2.1/24
GATEWAY     : 192.168.2.50
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 10024
RHOST:PORT  : 127.0.0.1:10025
MTU:        : 1500
```

- Configure PC 3:

```
PC3> ip 192.168.3.1/24 192.168.3.50
Checking for duplicate address...
PC1 : 192.168.3.1 255.255.255.0 gateway 192.168.3.50

PC3> show ip

NAME        : PC3[1]
IP/MASK     : 192.168.3.1/24
GATEWAY     : 192.168.3.50
DNS         :
MAC         : 00:50:79:66:68:02
LPORT       : 10028
RHOST:PORT  : 127.0.0.1:10029
MTU:        : 1500
```

- Check whether all the interfaces are up.

```
*Mar  1 00:02:02.031: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip interface brief | include manual
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.1.50    YES NVRAM  up                    up
Serial0/0                  unassigned      YES NVRAM  administratively down down
FastEthernet0/1            unassigned      YES NVRAM  administratively down down
Serial0/1                  unassigned      YES NVRAM  administratively down down
Serial0/2                  unassigned      YES NVRAM  administratively down down
Serial0/3                  unassigned      YES NVRAM  administratively down down
Serial1/0                  10.1.1.1        YES NVRAM  up                    up
Serial1/1                  unassigned      YES NVRAM  administratively down down
Serial1/2                  unassigned      YES NVRAM  administratively down down
Serial1/3                  unassigned      YES NVRAM  administratively down down
Serial2/0                  unassigned      YES NVRAM  administratively down down
Serial2/1                  unassigned      YES NVRAM  administratively down down
Serial2/2                  unassigned      YES NVRAM  administratively down down
Serial2/3                  unassigned      YES NVRAM  administratively down down
R1#
```

```
R2#show ip interface brief | include manual
FastEthernet0/0            192.168.2.50    YES manual up                    up
Serial1/0                  10.1.1.2        YES manual up                    up
Serial1/1                  11.1.1.1        YES manual up                    up
R2#
```

```
R3#show ip interface brief | include manual
FastEthernet0/0            192.168.3.50    YES manual up                    up
Serial1/0                  11.1.1.2        YES manual up                    up
R3#
```

Step 4: Configure and test the Single – Area OSPFv2 for IPv4 on R1, R2 and R3.

- Configure Single – Area OSPFv2 on R1:

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.50 0.0.0.7 area 1
R1(config-router)#network 10.1.1.1 0.0.0.3 area 1
R1(config-router)#exit
R1(config)#exit
```

- Test the Single – Area OSPFv2 on R1:

```
R1#show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.1.50
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.0 0.0.0.3 area 1
    192.168.1.48 0.0.0.7 area 1
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 110)
```

This shows the neighbour of Router 1 (which is only possible via OSPF)

```
R1#
*Mar  1 00:19:18.659: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.50 on Serial1/0 from LOADING to FULL, Loading Done
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.2.50      0   FULL/  -        00:00:33    10.1.1.2        Serial1/0
R1#
```

This will show the various routes used by OSPF. The ‘O’ at the start indicates a single area network.

```
R1#show ip route ospf
     11.0.0.0/24 is subnetted, 1 subnets
O       11.1.1.0 [110/128] via 10.1.1.2, 00:00:18, Serial1/0
O    192.168.2.0/24 [110/74] via 10.1.1.2, 00:00:18, Serial1/0
O    192.168.3.0/24 [110/138] via 10.1.1.2, 00:00:18, Serial1/0
R1#
```

- Configure Single – Area OSPFv2 on R2:

```
R2#
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#network 192.168.2.50 0.0.0.7 area 1
R2(config-router)#network 10.1.1.1 0.0.0.3 area 1
R2(config-router)#
*Mar  1 00:18:18.743: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.50 on Serial1/0 from LOADING to FULL, Loading Done
R2(config-router)#network 11.1.1.1 0.0.0.3 area 1
R2(config-router)#exit
R2(config)#exit
R2#
*Mar  1 00:20:03.455: %SYS-5-CONFIG_I: Configured from console by console
```

- Test the Single – Area OSPFv2 on R1:

  This will show the various routes used by OSPF. The ‘O’ at the start indicates a single area network.

```
R2#show ip route ospf
O    192.168.1.0/24 [110/74] via 10.1.1.1, 00:01:05, Serial1/0
O    192.168.3.0/24 [110/74] via 11.1.1.2, 00:01:05, Serial1/1
R2#
```

  This shows the neighbour of Router 2 (which is only possible via OSPF)

```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.3.50      0   FULL/  -        00:00:34    11.1.1.2        Serial1/1
192.168.1.50      0   FULL/  -        00:00:37    10.1.1.1        Serial1/0
R2#
```

- Configure Single – Area OSPFv2 on R3:

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router ospf 1
R3(config-router)#network 192.168.3.50 0.0.0.7 area 1
R3(config-router)#network 11.1.1.2 0.0.0.3 area 1
R3(config-router)#
*Mar  1 00:21:05.563: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.50 on Serial1/0 from LOADING to FULL, Loa
R3(config-router)#exit
R3(config)#exit
R3#
*Mar  1 00:21:34.279: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

- Test the Single – Area OSPFv2 on R3:

```
R3#show ip protocol
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.3.50
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    11.1.1.0 0.0.0.3 area 1
    192.168.3.48 0.0.0.7 area 1
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.50         110      00:02:28
    192.168.1.50         110      00:02:28
  Distance: (default is 110)
```

  This shows the neighbour of Router 3 (which is only possible via OSPF).

```
R3#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.2.50      0   FULL/  -        00:00:37    11.1.1.1        Serial1/0
R3#
```

This will show the various routes used by OSPF. The ‘O’ at the start indicates a single area network.

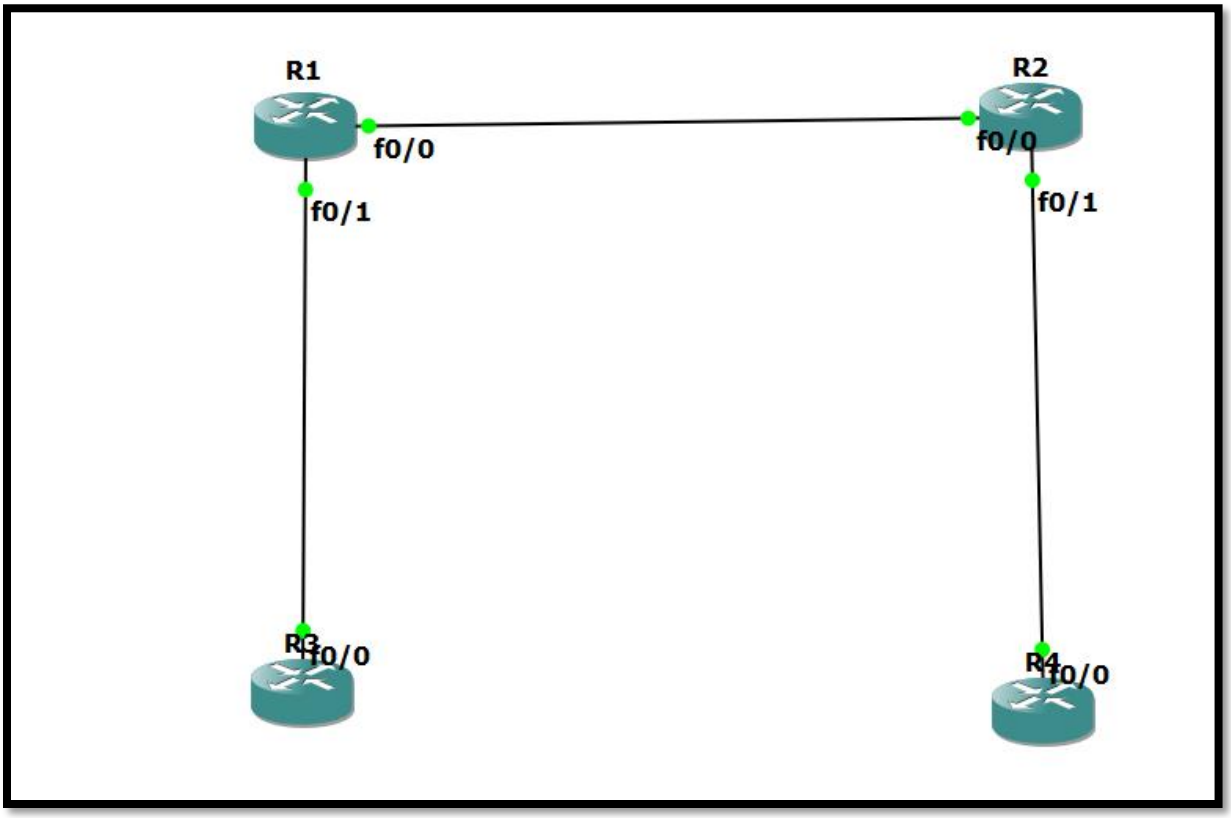
```
R3#show ip route ospf
     10.0.0.0/24 is subnetted, 1 subnets
O       10.1.1.0 [110/128] via 11.1.1.1, 00:03:38, Serial1/0
O    192.168.1.0/24 [110/138] via 11.1.1.1, 00:03:38, Serial1/0
O    192.168.2.0/24 [110/74] via 11.1.1.1, 00:03:38, Serial1/0
R3#
```

1. **Implement Multi-Area OSPFv2:**

Step 1: Assess the components.

- We use 4 routers.

Step 2: Design the Topology.



Step 3: Build and configure the network.

- Router 1 (R1):

```
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip cef
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 192.168.12.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
*Mar  1 00:01:07.751: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R1(config)#
*Mar  1 00:01:08.751: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config)#interface fastEthernet 0/1
R1(config-if)#ip address 192.168.13.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#e
*Mar  1 00:01:50.999: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:01:51.999: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R1(config-if)#exit
R1(config)#
```

- Router 2 (R2):

```
R2(config)#ip cef
R2(config)#interface fastEt
R2(config)#interface fastEthernet 0/0
R2(config-if)#ip address 192.168.12.2 255.255.255.0
R2(config-if)#exit
R2(config)#interface fastEthernet 0/1
R2(config-if)#ip address 192.168.24.2 255.255.255.0
R2(config-if)#exit
R2(config)#
```

```
*Mar  1 00:05:30.135: %SYS-5-CONFIG_I: Configured from console by console
R2#
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface fastEthernet 0/0
R2(config-if)#no shutdown
R2(config-if)#e
*Mar  1 00:06:24.351: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:06:25.351: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
R2(config)#interface fastEthernet 0/1
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
*Mar  1 00:06:47.627: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:06:48.627: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R2(config)#exit
R2#
*Mar  1 00:07:00.351: %SYS-5-CONFIG_I: Configured from console by console
R2#
```

- Router 3 (R3):

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#ip cef
R3(config)#interface Loopback0
R3(config-if)#ip addre
*Mar  1 00:02:31.983: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R3(config-if)#ip address 3.3.3.3 255.255.255.255
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#interface fastEth
R3(config)#interface fastEthernet 0/0
R3(config-if)#ip address 192.168.13.3 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
*Mar  1 00:03:52.139: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:03:53.139: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config)#
```

- Router 4 (R4):

```
R4#
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#
R4(config)#hostname R4
R4(config)#ip cef
R4(config)#interface Loopback0
R4(config-if)#ip addre
*Mar  1 00:03:28.275: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R4(config-if)#ip address 4.4.4.4 255.255.255.255
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#interface fastEthe
R4(config)#interface fastEthernet 0/0
R4(config-if)#ip address 192.168.24.4 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#
*Mar  1 00:04:16.819: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:04:17.883: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R4(config)#
R4(config)#
```

Step 4: Configure and test the system for Multi – Area OSPFv2.

- Router 1 (R1):

```
R1(config)#
R1(config)#
R1(config)#router ospf 1
R1(config-router)#network 192.168.12.0 0.0.0.255 area 0
R1(config-router)#network 192.168..13.0 0.0.0.255 area 1
                                   ^
% Invalid input detected at '^' marker.

R1(config-router)#network 192.168.13.0 0.0.0.255 area 1
R1(config-router)#end
R1#
*Mar  1 00:22:06.523: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

To verify Multi – Area OSPFv2 for Router 1.
R1 has formed a neighbour adjacency with R2 and R3.

```
R1#
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.24.2      1   FULL/BDR        00:00:37    192.168.12.2    FastEthernet0/0
3.3.3.3           1   FULL/BDR        00:00:33    192.168.13.3    FastEthernet0/1
R1#show ip route ospf
```

We see three OSPF entries. The first one is for 3.3.3.3/32, the loopback interface of R3. It shows up with an O since this is an intra-area route. R1 has also learned about 4.4.4.4/32 and 192.168.24.0/24. These two entries show up as O IA since they are inter-area routes.

```
R1#show ip route ospf
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/11] via 192.168.13.3, 00:01:37, FastEthernet0/1
     4.0.0.0/32 is subnetted, 1 subnets
O IA    4.4.4.4 [110/21] via 192.168.12.2, 00:00:50, FastEthernet0/0
O IA 192.168.24.0/24 [110/20] via 192.168.12.2, 00:01:37, FastEthernet0/0
R1#
```

- Router 2 (R2):

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#network 192.168.12.0 0.0.0.255 area 0
R2(config-router)#network 1
*Mar  1 00:20:10.447: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.13.1 on FastEthernet0/0 from LOADING to FULL, Loading Done
R2(config-router)#network 192.168.24.0 0.0.0.255 area 5
R2(config-router)#network 192.168.24.0 0.0.0.255 area 2
R2(config-router)#
*Mar  1 00:20:29.151: %OSPF-6-AREACHG: 192.168.24.0/24 changed from area 5 to area 2
R2(config-router)#end
R2#
R2#
*Mar  1 00:20:35.291: %SYS-5-CONFIG_I: Configured from console by console
R2#
```

To verify Multi – Area OSPFv2 for Router 2.
R2 has formed neighbour adjacencies with R1 and R4.

```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.13.1      1   FULL/DR         00:00:35    192.168.12.1    FastEthernet0/0
4.4.4.4           1   FULL/BDR        00:00:38    192.168.24.4    FastEthernet0/1
R2#
```

We see that R2 has learned about 3.3.3.3/32 and 192.168.13.0/24 which area inter-area routes. 4.4.4.4/32 is an intra-area route.

```
R2#show ip route ospf
O IA 192.168.13.0/24 [110/20] via 192.168.12.1, 00:02:20, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/21] via 192.168.12.1, 00:02:20, FastEthernet0/0
     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/11] via 192.168.24.4, 00:02:20, FastEthernet0/1
R2#
```

- Router 3 (R3):

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router ospf 1
R3(config-router)#network 3.3.3.3 0.0.0.0 area 1
R3(config-router)#network 192.168.13.0 0.0.0.255 area 1
R3(config-router)#end
R3#
*Mar  1 00:19:08.251: %SYS-5-CONFIG_I: Configured from console by console
R3#
*Mar  1 00:19:13.899: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.13.1 on FastEthernet0/0 from LOADING to FULL, Loading Done
R3#
```

To verify Multi – Area OSPFv2 for Router 3.
R3 has formed neighbour adjacency with R1.

```
R3#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.13.1      1   FULL/DR         00:00:39    192.168.13.1    FastEthernet0/0
R3#
```

Everything that R3 has learned is from another area, that's why we only see inter-area routes here.

```
R3#
R3#show ip route ospf
O IA 192.168.12.0/24 [110/20] via 192.168.13.1, 00:03:36, FastEthernet0/0
     4.0.0.0/32 is subnetted, 1 subnets
O IA    4.4.4.4 [110/31] via 192.168.13.1, 00:02:53, FastEthernet0/0
O IA 192.168.24.0/24 [110/30] via 192.168.13.1, 00:03:36, FastEthernet0/0
R3#
```

- Router 4 (R4):

```
R4#
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#router ospf 1
R4(config-router)#network 4.4.4.4 0.0.0.0 area 2
R4(config-router)#network 192.168.24.0 0.0.0.255 area 2
R4(config-router)#end
R4#
*Mar  1 00:17:15.543: %SYS-5-CONFIG_I: Configured from console by console
R4#
```

To verify Multi – Area OSPFv2 for Router 4.
R4 has formed neighbour adjacency with R2.

```
R4#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.24.2      1   FULL/DR         00:00:30    192.168.24.2    FastEthernet0/0
R4#
```

Everything that R4 has learned is from another area, that's why we only see inter-area routes here.

```
R4#
R4#show ip route ospf
O IA 192.168.12.0/24 [110/20] via 192.168.24.2, 00:03:41, FastEthernet0/0
O IA 192.168.13.0/24 [110/30] via 192.168.24.2, 00:03:41, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O IA    3.3.3.3 [110/31] via 192.168.24.2, 00:03:41, FastEthernet0/0
R4#
```

- Just to be sure, let's try a quick ping between R3 and R4 to prove that our multi-area OSPF configuration is working

```
R3#ping 4.4.4.4 source 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 3.3.3.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 92/106/140 ms
R3#
```

2. OSPFv2 Route Summarization and Filtering

Step 1: Follow the same Topology as the Multi – Area OSPFv2.

Step 2: Add more loopbacks to Router 3 and configure the OSPF accordingly.

```
R3(config)#
R3(config)#interface Loopback 1
R3(config-if)#
*Mar  1 00:15:40.863: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
R3(config-if)#ip address 172.17.1.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
R3(config)#
R3(config)#interface Loopback2
R3(config-if)#
R3(config-if)#
*Mar  1 00:16:10.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback2, changed state to up
R3(config-if)#ip address 172.17.2.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
R3(config)#interface Loopback3
R3(config-if)#ip ad
*Mar  1 00:16:46.959: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback3, changed state to up
R3(config-if)#ip address 172.17.3.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
R3(config)#interface Loopback0
R3(config-if)#ip address 172.17.0.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
R3(config)#interface Loopback5
R3(config-if)#ip add
*Mar  1 00:17:56.823: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback5, changed state to up
R3(config-if)#ip address 172.17.4.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit
```

```
R3(config)#router ospf 1
R3(config-router)#
R3(config-router)#network 172.17.0.1 0.0.0.255 area 1
R3(config-router)#network 172.17.1.1 0.0.0.255 area 1
R3(config-router)#network 172.17.2.1 0.0.0.255 area 1
R3(config-router)#network 172.13.3.1 0.0.0.255 area 1
R3(config-router)#network 172.17.3.1 0.0.0.255 area 1
R3(config-router)#network 172.17.4.1 0.0.0.255 area 1
R3(config-router)#
R3(config-router)#end
R3#
*Mar  1 00:22:32.787: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

If we perform ‘show ip route’ on R2 we can see all the loopbacks of R3. Over here nothing is being summarized.

```
R2#show ip route
Codes: 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

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
O IA 192.168.13.0/24 [110/20] via 192.168.12.1, 00:13:02, FastEthernet0/0
     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/11] via 192.168.24.4, 00:13:02, FastEthernet0/1
C    192.168.24.0/24 is directly connected, FastEthernet0/1
     172.17.0.0/32 is subnetted, 5 subnets
O IA    172.17.4.1 [110/21] via 192.168.12.1, 00:00:25, FastEthernet0/0
O IA    172.17.0.1 [110/21] via 192.168.12.1, 00:03:44, FastEthernet0/0
O IA    172.17.1.1 [110/21] via 192.168.12.1, 00:03:15, FastEthernet0/0
O IA    172.17.2.1 [110/21] via 192.168.12.1, 00:01:39, FastEthernet0/0
O IA    172.17.3.1 [110/21] via 192.168.12.1, 00:00:46, FastEthernet0/0
R2#
```

But if we want to summarize the data being received, we will perform the summary operation on R1.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#area 1 range 172.17.0.0 255.255.252.0
R1(config-router)#end
R1#
*Mar  1 00:27:47.167: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

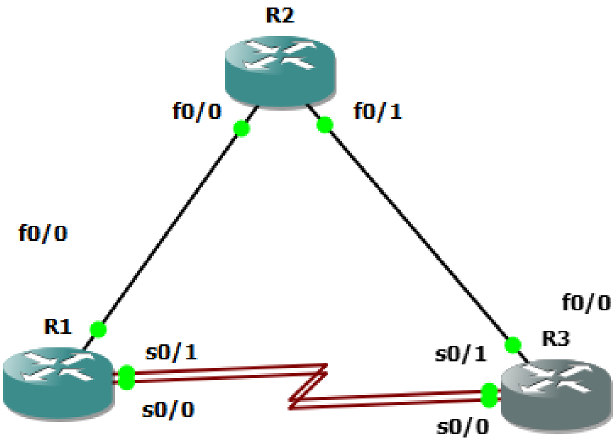
Here we can see that the network area is sub-netted into 2 subnets rather than 5. From this we can see OSPF summarization based on the range of the subnets and their masks.

```
R2#
R2#show ip route
Codes: 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

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
O IA 192.168.13.0/24 [110/20] via 192.168.12.1, 00:15:04, FastEthernet0/0
     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/11] via 192.168.24.4, 00:15:04, FastEthernet0/1
C    192.168.24.0/24 is directly connected, FastEthernet0/1
     172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
```

SDN Practical 8

Aim: Implement BGP Communities
1. Implement eBGP for Ipv4
2.Implement MP(Multi protocol) -BGP(meaning it supports multiple protocol eg:IPv6,ospf etc)-part 1& 2
3. Implement BGP path Manipulation

1. Implement eBGP for Ipv4
Step 1: Topology



Step 2:Configuring routers
R1

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#no ip domain lookup
R1(config)#line con 0
R1(config-line)#logging sync
R1(config-line)#exec-time 0 0
R1(config-line)#exit
R1(config)#int loopback 0
R1(config-if)#ip add 192.168.1.1 255.255.255.224
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#int loopback 1
R1(config-if)#ip add 192.168.1.65 255.255.255.192
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#int f0/0
R1(config-if)#ip add 10.1.2.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#int serial 0/0
R1(config-if)#ip add 10.1.3.1 255.255.255.128
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#int serial 0/1
R1(config-if)#ip add 10.1.3.129 255.255.255.128
R1(config-if)#no sh
R1(config-if)#exit
*Mar  1 00:02:06.011: %LINEPROTO-5-UPDOWN: Line protocol on Interface Lo
*Mar  1 00:02:06.447: %LINEPROTO-5-UPDOWN: Line protocol on Interface Lo
R1(config-if)#exit
R1(config)#
```

R2

```
R2#conf t
Enter configuration commands, one per line.  End with CN
R2(config)#hostname R2
R2(config)#no ip domain lookup
R2(config)#line con 0
R2(config-line)#logging sync
R2(config-line)#exec-time 0 0
R2(config-line)#exit
R2(config)#int loopback 0
R2(config-if)#ip add 192.168.2.1 255.255.255.224
R2(config-if)#no sh
R2(config-if)#exit
R2(config)#int loopback 1
R2(config-if)#ip add 192.168.2.65 255.255.255.192
R2(config-if)#no sh
R2(config-if)#exit
R2(config)#int f0/0
R2(config-if)#ip add 10.1.2.2 255.255.255.0
R2(config-if)#no sh
R2(config-if)#exit
R2(config)#int f0/1
R2(config-if)#ip add 10.2.3.2 255.255.255.0
R2(config-if)#no sh
R2(config-if)#exit
*Mar  1 00:02:44.659: %LINEPROTO-5-UPDOWN: Line protocol
*Mar  1 00:02:45.083: %LINEPROTO-5-UPDOWN: Line protocol
R2(config-if)#exit
R2(config)#
```

R3

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname R3
R3(config)#no ip domain lookup
R3(config)#line con 0
R3(config-line)#logging sync
R3(config-line)#exec-time 0 0
R3(config-line)#exit
R3(config)#int loopback 0
R3(config-if)#ip add 192.168.3.1 255.255.255.224
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#int loopback 1
R3(config-if)#ip add 192.168.3.65 255.255.255.192
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#int f0/0
R3(config-if)#ip add 10.2.3.3 255.255.255.0
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#int serial 0/0
R3(config-if)#ip add 10.1.3.3 255.255.255.128
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#int serial 0/1
R3(config-if)#ip add 10.1.3.130 255.255.255.128
R3(config-if)#no sh
R3(config-if)#exit
*Mar  1 00:08:01.075: %SYS-5-CONFIG_I: Configured from console by console
R3(config-if)#exit
R3(config)#
```

Step 3: Implement BGP and neighbor relationships on R1

R1

```
R1(config)#router bgp 1000
R1(config-router)#bgp router-id 1.1.1.1
R1(config-router)#neighbor 10.1.2.2 remote-as 500
R1(config-router)#neighbor 10.1.3.3 remote-as 300
R1(config-router)#neighbor 10.1.3.130 remote-as 300
R1(config-router)#network 192.168.1.0 mask 255.255.255.224
R1(config-router)#network 192.168.1.64 mask 255.255.255.192
R1(config-router)#
```

R2

```
R2(config)#router bgp 500
R2(config-router)#bgp router-id 2.2.2.2
R2(config-router)#neighbor 10.1.2.1 remote-as 1000
R2(config-router)#neighbor 10.2.3.3 remote-as 300
R2(config-router)#network 192.168.2.0 mask 255.255.255.224
R2(config-router)#network 192.168.2.64 mask 255.255.255.192
R2(config-router)#
```

R3

```
R2(config)#router bgp 300
R2(config-router)#bgp router-id 3.3.3.3
R2(config-router)#no bgp default ipv4-unicast
R2(config-router)#neighbor 10.2.3.2 remote-as 500
R2(config-router)#neighbor 10.1.3.1 remote-as 1000
R2(config-router)#neighbor 10.1.3.129 remote-as 1000
R2(config-router)#
```

Step 4: Verifying BGP neighbor relationships. (patience 5/10 sec)

R1

```
R1#show ip route bgp
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B       192.168.2.64/26 [20/0] via 10.1.2.2, 00:06:15
B       192.168.2.0/27 [20/0] via 10.1.2.2, 00:06:15
R1#
```

R2

```
R2#show ip route bgp
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
B       192.168.1.64/26 [20/0] via 10.1.2.1, 00:06:25
B       192.168.1.0/27 [20/0] via 10.1.2.1, 00:06:25
R2#
```

R3

```
R3#show ip route bgp

R3#
```

Notice the BGP state between R2 and R1 is established, while the BGP state between R2 and R3 is idle.

```
R3(config)#router bgp 300
R3(config-router)#address-family ipv4
R3(config-router-af)#
R3(config-router-af)#neighbor 10.1.3.1 activate
R3(config-router-af)#neighbor 10.1.3.129 activate
R3(config-router-af)#neighbor 10.2.3.2 activate
R3(config-router-af)#network 192.168.3.0 mask 255.255.255.224
R3(config-router-af)#network 192.168.3.64 mask 255.255.255.192
R3(config-router-af)#
```

BGP in R3

```
R3#show ip route bgp
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
B       192.168.1.64/26 [20/0] via 10.1.3.1, 00:00:47
B       192.168.1.0/27 [20/0] via 10.1.3.1, 00:00:47
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B       192.168.2.64/26 [20/0] via 10.2.3.2, 00:00:16
B       192.168.2.0/27 [20/0] via 10.2.3.2, 00:00:16
R3#
*Mar  1 00:26:47.883: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

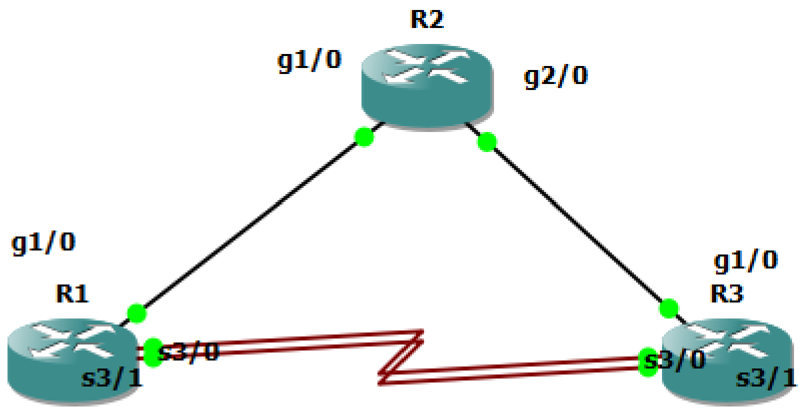
Step 5: Verifying BGP operations.

```
R2#show ip bgp
BGP table version is 9, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  192.168.1.0/27   10.2.3.3                               0 300 1000 i
*>                  10.1.2.1                 0             0 1000 i
*  192.168.1.64/26  10.2.3.3                               0 300 1000 i
*>                  10.1.2.1                 0             0 1000 i
*> 192.168.2.0/27   0.0.0.0                  0         32768 i
*> 192.168.2.64/26  0.0.0.0                  0         32768 i
*> 192.168.3.0/27   10.2.3.3                 0             0 300 i
*                   10.1.2.1                               0 1000 300 i
*> 192.168.3.64/26  10.2.3.3                 0             0 300 i
*                   10.1.2.1                               0 1000 300 i
```

```
R2#show ip bgp neighbors
BGP neighbor is 10.1.2.1,  remote AS 1000, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:23:18
  Last read 00:00:18, last write 00:00:18, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                5          4
    Keepalives:            25         25
    Route Refresh:          0          0
    Total:                 31         30
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 9, neighbor version 9/0
 Output queue size : 0
 --More--
```

2. Aim : Implement eBGP for ipv4.

Topology :



- **Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing**

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

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

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each router.

a. Console into each router, enter global configuration mode, and apply the basic settings and interface addressing.

Router R1:

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface Loopback0
R1(config-if)#
*Dec  1 14:04:56.991: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R1(config-if)#ip address 192.168.1.1 255.255.255.224
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#interface Loopback1
R1(config-if)#
*Dec  1 14:05:53.275: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
R1(config-if)#ip address 192.168.1.65 255.255.255.192
R1(config-if)# no shut
R1(config-if)#exit
R1(config)#interface g1/0
R1(config-if)#ip address 10.1.2.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Dec  1 14:06:57.947: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Dec  1 14:06:58.947: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
R1(config)#interface s3/0
R1(config-if)#ip address 10.1.3.1 255.255.255.128
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Dec  1 14:07:28.907: %LINK-3-UPDOWN: Interface Serial3/0, changed state to up
R1(config)#
*Dec  1 14:07:29.911: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
R1(config)#interface s3/1
R1(config-if)#ip address 10.1.3.129 255.255.255.128
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
```

Router R2 :

- ### Configure Lookbacks :

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface Loopback0
R2(config-if)#
*Dec  1 14:08:31.779: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R2(config-if)#ip address 192.168.2.1 255.255.255.224
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#interface Loopback1
R2(config-if)#
*Dec  1 14:08:51.135: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
R2(config-if)#ip address 192.168.2.65 255.255.255.192
R2(config-if)#no shut
R2(config-if)#exit
```

```
R2(config)#interface g1/0
R2(config-if)#ip address 10.1.2.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
*Dec  1 14:09:47.343: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Dec  1 14:09:48.343: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
R2(config-if)#exit
R2(config)#interface g2/0
R2(config-if)#$tocol on Interface GigabitEthernet1/0, changed state to up
N: Line protocol on Interface GigabitEthernet1/0, changed state to up
 ^
% Invalid input detected at '^' marker.

R2(config-if)#R2(config-if)#exit
               ^
% Invalid input detected at '^' marker.

R2(config-if)#ip address 10.2.3.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Dec  1 14:10:33.723: %LINK-3-UPDOWN: Interface GigabitEthernet2/0, changed state to up
*Dec  1 14:10:34.723: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0, changed state to up
```

Router R3 :

- ## Configure Lookbacks :

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface Loopback0
R3(config-if)#
*Dec  1 14:11:03.495: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R3(config-if)#ip address 192.168.3.1 255.255.255.224
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
R3(config)#R3(config-if)#no shut
            ^
% Invalid input detected at '^' marker.

R3(config)#interface Loopback1
R3(config-if)#
*Dec  1 14:11:27.903: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
R3(config-if)#ip address 192.168.3.65 255.255.255.192
R3(config-if)#no shut
R3(config-if)#exit
```

```
R3(config)#int g1/0
R3(config-if)#ip address 10.2.3.3 255.255.255.0
R3(config-if)#negotiation auto
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Dec  1 14:12:23.367: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Dec  1 14:12:24.367: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
R3(config)#int s3/0
R3(config-if)#ip address 10.1.3.3 255.255.255.128
R3(config-if)# no shut
R3(config-if)#
*Dec  1 14:12:50.295: %LINK-3-UPDOWN: Interface Serial3/0, changed state to up
R3(config-if)#
*Dec  1 14:12:51.303: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
R3(config-if)#exit
R3(config)#int s3/1
R3(config-if)#ip address 10.1.3.130 255.255.255.128
R3(config-if)# no shut
R3(config-if)#exit
R3(config)#
*Dec  1 14:13:18.703: %LINK-3-UPDOWN: Interface Serial3/1, changed state to up
R3(config)#
*Dec  1 14:13:19.711: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/1, changed state to up
```

Part 2: Configure and Verify eBGP for IPv4 on all Routers

Step 1: Implement BGP and neighbor relationships on R1.

a. Enter BGP configuration mode from global configuration mode, specifying AS 1000.
b. Configure the BGP router-id for R1.

```
R1(config)#router bgp 1000
R1(config-router)#
R1(config-router)#bgp router-id 1.1.1.1
```

c. Based on the topology diagram, configure all the designated neighbors for R1.

```
R1(config-router)#neighbor 10.1.2.2 remote-as 500
R1(config-router)#neighbor 10.1.3.3 remote-as 300
R1(config-router)#neighbor 10.1.3.130 remote-as 300
R1(config-router)#
```

d. Configure R1 to advertise the IPv4 prefixes local to ASN 1000.

```
R1(config-router)#network 192.168.1.0 mask 255.255.255.224
R1(config-router)#network 192.168.1.64 mask 255.255.255.192
```

Step 2: Implement BGP and neighbor relationships on R2.

a. Enter BGP configuration mode from global configuration mode, specifying AS 500.
b. Configure the BGP router-id for R2.
c. Based on the topology diagram, configure all the designated neighbors for R2.
d. Configure R2 to advertise the IPv4 prefixes local to ASN 500.

```
R2(config)#router bgp 500
R2(config-router)#bgp router-id 2.2.2.2
R2(config-router)#neighbor 10.1.2.1 remote-as 1000
R2(config-router)#
*Dec  1 14:15:07.675: %BGP-5-ADJCHANGE: neighbor 10.1.2.1 Up
R2(config-router)#neighbor 10.2.3.3 remote-as 300
R2(config-router)#network 192.168.2.0 mask 255.255.255.224
R2(config-router)#network 192.168.2.64 mask 255.255.255.192
```

Step 3: Implement BGP and neighbor relationships on R3.

a. Enter BGP configuration mode from global configuration mode, specifying AS 300.
b. Configure the BGP router-id for R3.
c. Unlike the configuration on R1 and R2, disable the default IPv4 unicast behavior.

```
R3(config)#router bgp 300
R3(config-router)#bgp router-id 3.3.3.3
R3(config-router)#no bgp default ipv4-unicast
```

The default behavior in IOS is `bgp default ipv4-unicast`. Routers R1 and R2 were configured using this default behavior. The bgp default ipv4-unicast command enables the automatic exchange of IPv4 address family prefixes. When this command is disabled using `no bgp default ipv4-unicast`, bgp neighbors must be activated within IPv4 address family (AF) configuration mode. BGP network commands must also be configured within IPv4 AF mode.
d. Based on the topology diagram, configure all the designated neighbors for R3.

```
R3(config-router)#neighbor 10.2.3.2 remote-as 500
R3(config-router)#neighbor 10.1.3.1 remote-as 1000
R3(config-router)#neighbor 10.1.3.129 remote-as 1000
```

Step 4: Verifying BGP neighbor relationships.

a. Examine the routing tables on each router. Notice that R1 and R2 are receiving BGP prefixes from each other but not receiving BGP prefixes from R3. And R3 is not receiving any prefixes from R1 or R2. This is because R3 was configured using `no bgp default ipv4-unicast` and the interfaces must be activated within IPv4 address configuration mode.

Router R1 :

```
R1#show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:00:56
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:00:56
R1#
*Dec  1 14:21:16.251: %BGP-5-ADJCHANGE: neighbor 10.1.3.3 Up
R1#
*Dec  1 14:21:23.447: %BGP-5-ADJCHANGE: neighbor 10.1.3.130 Up
```

Router R2 :

```
R2#show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.1.0/27 [20/0] via 10.1.2.1, 00:01:25
B        192.168.1.64/26 [20/0] via 10.1.2.1, 00:01:25
```

Router R3 :

```
R3#show ip route bgp | begin Gateway
Gateway of last resort is not set
```

b. This can be further verified by examining the BGP neighbor adjacencies on R2. Notice the BGP state between R2 and R1 is established, while the BGP state between R2 and R3 is idle.

```
R2#show ip bgp neighbors
BGP neighbor is 10.1.2.1,  remote AS 1000, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:03:02
  Last read 00:00:09, last write 00:00:05, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                2          2
    Keepalives:             4          4
    Route Refresh:          0          0
    Total:                  7          7
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  Session: 10.1.2.1
  BGP table version 5, neighbor version 5/0
  Output queue size : 0
  Index 1, Advertise bit 0
  1 update-group member
```

c. The interfaces on R3 need to be activated in IPv4 AF configuration mode. The `neighbor activate` command in IPv4 AF configuration mode is required to enable the exchange of BGP information between neighbors. This will enable R3 to form an established neighbor adjacency with both R1 and R2. Additionally, because `bgp default ipv4-unicast` is disabled, `network` commands must be configured in IPv4 AF configuration mode.

```
R3(config)#router bgp 300
R3(config-router)#address-family ipv4
R3(config-router-af)#neighbor 10.1.3.1 activate
R3(config-router-af)#
*Dec  1 14:21:16.395: %BGP-5-ADJCHANGE: neighbor 10.1.3.1 Up
R3(config-router-af)#neighbor 10.1.3.129 activate
R3(config-router-af)#
*Dec  1 14:21:23.591: %BGP-5-ADJCHANGE: neighbor 10.1.3.129 Up
R3(config-router-af)#neighbor 10.2.3.2 activate
R3(config-router-af)#network 192.168.3.0 mask 255.255.255.224
R3(config-router-af)#
*Dec  1 14:21:35.023: %BGP-5-ADJCHANGE: neighbor 10.2.3.2 Up
R3(config-router-af)#network 192.168.3.64 mask 255.255.255.192
R3(config-router-af)#exit
R3(config-router)#exit
R3(config)#exit
R3#
*Dec  1 14:22:17.779: %SYS-5-CONFIG_I: Configured from console by console
```

d. Verify that all BGP speakers are receiving prefixes from their neighbors. The prefixes from R3 are highlighted in the routing tables of R1 and R2.

```
R1#show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:05:33
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:05:33
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.3.0/27 [20/0] via 10.1.3.3, 00:00:05
B        192.168.3.64/26 [20/0] via 10.1.3.3, 00:00:05
R1#show running-config | section bgp
router bgp 1000
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 network 192.168.1.0 mask 255.255.255.224
 network 192.168.1.64 mask 255.255.255.192
 neighbor 10.1.2.2 remote-as 500
 neighbor 10.1.3.3 remote-as 300
 neighbor 10.1.3.130 remote-as 300
```

```
R2#show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.1.0/27 [20/0] via 10.1.2.1, 00:05:39
B        192.168.1.64/26 [20/0] via 10.1.2.1, 00:05:39
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.3.0/27 [20/0] via 10.2.3.3, 00:00:11
B        192.168.3.64/26 [20/0] via 10.2.3.3, 00:00:11
```

```
R3#show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.1.0/27 [20/0] via 10.1.3.1, 00:01:11
B        192.168.1.64/26 [20/0] via 10.1.3.1, 00:01:11
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.2.0/27 [20/0] via 10.2.3.2, 00:00:52
B        192.168.2.64/26 [20/0] via 10.2.3.2, 00:00:52
```

e. Verify that the BGP state between R2 and R3 has now been established.

```
R2#show ip bgp neighbors | begin BGP neighbor is 10.2.3.3
BGP neighbor is 10.2.3.3,  remote AS 300, external link
  BGP version 4, remote router ID 3.3.3.3
  BGP state = Established, up for 00:01:27
  Last read 00:00:24, last write 00:00:19, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
```

Step 5: Examining the running-configs.

Examine the running-configs on all three routers. Because router R3 was configured using `no bgp default ipv4-unicast` command, notice that the network commands were automatically entered under the IPv4 AF. This is the same configuration mode where the neighbors were activated to exchange BGP information.

```
R1#show running-config | section bgp
router bgp 1000
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 network 192.168.1.0 mask 255.255.255.224
 network 192.168.1.64 mask 255.255.255.192
 neighbor 10.1.2.2 remote-as 500
 neighbor 10.1.3.3 remote-as 300
 neighbor 10.1.3.130 remote-as 300
```

```
R2#show running-config | section bgp
router bgp 500
 bgp router-id 2.2.2.2
 bgp log-neighbor-changes
 network 192.168.2.0 mask 255.255.255.224
 network 192.168.2.64 mask 255.255.255.192
 neighbor 10.1.2.1 remote-as 1000
 neighbor 10.2.3.3 remote-as 300
```

```
R3#show running-config | section bgp
router bgp 300
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.1.3.1 remote-as 1000
 neighbor 10.1.3.129 remote-as 1000
 neighbor 10.2.3.2 remote-as 500
 !
 address-family ipv4
  network 192.168.3.0 mask 255.255.255.224
  network 192.168.3.64 mask 255.255.255.192
  neighbor 10.1.3.1 activate
  neighbor 10.1.3.129 activate
  neighbor 10.2.3.2 activate
 exit-address-family
```

Step 6: Verifying BGP operations.

a. To verify the BGP operation on R2, issue the `show ip bgp` command.

```
R2#show ip bgp
BGP table version is 7, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *   192.168.1.0/27   10.2.3.3                               0 300 1000 i
 *>                   10.1.2.1                 0             0 1000 i
 *   192.168.1.64/26  10.2.3.3                               0 300 1000 i
 *>                   10.1.2.1                 0             0 1000 i
 *>  192.168.2.0/27   0.0.0.0                  0         32768 i
 *>  192.168.2.64/26  0.0.0.0                  0         32768 i
 *   192.168.3.0/27   10.1.2.1                               0 1000 300 i
 *>                   10.2.3.3                 0             0 300 i
 *   192.168.3.64/26  10.1.2.1                               0 1000 300 i
 *>                   10.2.3.3                 0             0 300 i
```

b. Use the `show ip bgp ip-prefix` command to display all the paths for a specific route and the BGP path attributes for that route.

```
R2#show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/27, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 2
  300 1000
    10.2.3.3 from 10.2.3.3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  1000
    10.1.2.1 from 10.1.2.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

c. Examine the BGP neighbor relationships on R2 using the `show ip bgp neighbors` command.

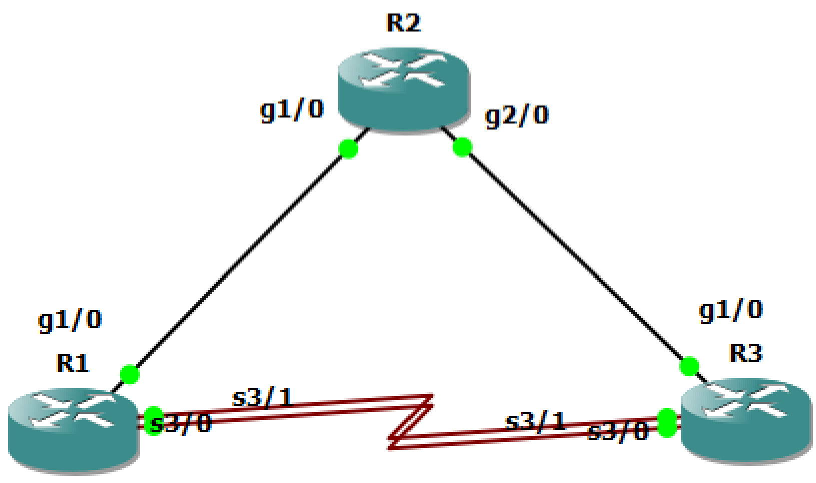
```
R2#show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/27, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 2
  300 1000
    10.2.3.3 from 10.2.3.3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  1000
    10.1.2.1 from 10.1.2.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
R2#show ip bgp neighbors
BGP neighbor is 10.1.2.1,  remote AS 1000, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:12:41
  Last read 00:00:44, last write 00:00:05, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
```

```
                        Sent       Rcvd
   Opens:                  1          1
   Notifications:          0          0
   Updates:                3          3
   Keepalives:            15         14
   Route Refresh:          0          0
   Total:                 19         18
 Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
 Session: 10.1.2.1
 BGP table version 7, neighbor version 7/0
 Output queue size : 0
 Index 1, Advertise bit 0
 1 update-group member
 Slow-peer detection is disabled
 Slow-peer split-update-group dynamic is disabled
                                Sent       Rcvd
 Prefix activity:               ----       ----
   Prefixes Current:               6          4 (Consumes 320 bytes)
   Prefixes Total:                 8          4
   Implicit Withdraw:              2          0
   Explicit Withdraw:              0          0
   Used as bestpath:             n/a          2
   Used as multipath:            n/a          0

                                  Outbound    Inbound
 Local Policy Denied Prefixes:    --------    -------
   Bestpath from this peer:              2        n/a
   Total:                                2          0
 Number of NLRIs in the update sent: max 2, min 0
 Last detected as dynamic slow peer: never
 Dynamic slow peer recovered: never
 Refresh Epoch: 1
 Last Sent Refresh Start-of-rib: never
 Last Sent Refresh End-of-rib: never
 Last Received Refresh Start-of-rib: never
```

```
  Refresh-In took 11 seconds
                                        Sent       Rcvd
        Refresh activity:               ----       ----
          Refresh Start-of-RIB           1          1
          Refresh End-of-RIB             1          1

  Address tracking is enabled, the RIB does have a route to 10.2.3.3
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 1
Local host: 10.2.3.2, Local port: 179
Foreign host: 10.2.3.3, Foreign port: 22819
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x176D8C):
Timer          Starts    Wakeups            Next
Retrans            10          0             0x0
TimeWait            0          0             0x0
AckHold            10          6             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
ProcessQ            0          0             0x0

iss: 3489576355  snduna: 3489576808  sndnxt: 3489576808
irs: 2871417030  rcvnxt: 2871417544

sndwnd:  15932  scale:      0  maxrcvwnd:  16384
rcvwnd:  15871  scale:      0  delrcvwnd:    513

SRTT: 737 ms, RTTO: 2506 ms, RTV: 1769 ms, KRTT: 0 ms
minRTT: 28 ms, maxRTT: 1000 ms, ACK hold: 200 ms
Status Flags: passive open, gen tcbs
Option Flags: nagle, path mtu capable
IP Precedence value : 6

Datagrams (max data segment is 1460 bytes):
Rcvd: 23 (out of order: 0), with data: 12, total data bytes: 513
Sent: 22 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with data: 12, total data bytes: 452

 Packets received in fast path: 0, fast processed: 0, slow path: 0
 fast lock acquisition failures: 0, slow path: 0
TCP Semaphore      0x6A631434  FREE
```

3. Aim : Implementation of BGP Path Manipulation.

Topology :



Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

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

Attach the devices as shown in the topology diagram, and cable as necessary.

- Step 2: Configure basic settings for each router.

a. Console into each router, enter global configuration mode, and apply the basic settings and interface addressing.
Router R1 :

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ipv6 unicast-routing
R1(config)#interface g1/0
R1(config-if)#ip address 10.1.2.1 255.255.255.0
R1(config-if)#ipv6 address fe80::1:1 link-local
R1(config-if)#ipv6 address 2001:db8:acad:1012::1/64
R1(config-if)#no shut
R1(config-if)#
*Dec  1 15:35:40.975: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Dec  1 15:35:41.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
R1(config-if)#exit
R1(config)#int s3/0
R1(config-if)#ip address 10.1.3.1 255.255.255.128
R1(config-if)#ipv6 address fe80::1:2 link-local
R1(config-if)#ipv6 address 2001:db8:acad:1013::1/64
R1(config-if)#no shut
R1(config-if)#exit
*Dec  1 15:36:17.287: %LINK-3-UPDOWN: Interface Serial3/0, changed state to up
R1(config-if)#exit
*Dec  1 15:36:18.303: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
R1(config-if)#exit
R1(config)#int s3/1
R1(config-if)#ip address 10.1.3.129 255.255.255.128
R1(config-if)#
*Dec  1 15:36:43.595: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to down
R1(config-if)#ipv6 address fe80::1:3 link-local
R1(config-if)#ipv6 address fe80::1:3 link-local
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Dec  1 15:37:08.391: %LINK-3-UPDOWN: Interface Serial3/1, changed state to up
R1(config)#
```

```
R1(config)#int lo0
R1(config-if)#
*Dec  1 15:37:21.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R1(config-if)#ip address 192.168.1.1 255.255.255.224
R1(config-if)#ipv6 address fe80::1:4 link-local
R1(config-if)#
*Dec  1 15:37:33.611: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/1, changed state to down
R1(config-if)#ipv6 address fe80::1:4 link-local
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#int lo1
R1(config-if)#
*Dec  1 15:37:53.727: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
R1(config-if)#ip address 192.168.1.65 255.255.255.192
R1(config-if)#ip address 192.168.1.65 255.255.255.192
R1(config-if)#ipv6 address 2001:db8:acad:1001::1/64
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Dec  1 15:43:43.603: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
R1(config)#
*Dec  1 15:44:33.607: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/1, changed state to up
```

Router R2 :

```
R2(config)#ipv6 unicast-routing
R2(config)#int g1/0
R2(config-if)#ip address 10.1.2.2 255.255.255.0
R2(config-if)#ip address 10.1.2.2 255.255.255.0
R2(config-if)#ipv6 address fe80::2:1 link-local
R2(config-if)#ipv6 address 2001:db8:acad:1012::2/64
R2(config-if)#no shut
R2(config-if)#e
*Dec  1 15:39:41.435: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Dec  1 15:39:42.435: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
R2(config-if)#exit
R2(config)#int g2/0
R2(config-if)#ip address 10.2.3.2 255.255.255.0
R2(config-if)#ipv6 address fe80::2:2 link-local
R2(config-if)#ipv6 address 2001:db8:acad:1023::2/64
R2(config-if)#no shut
R2(config-if)#ex
*Dec  1 15:40:22.323: %LINK-3-UPDOWN: Interface GigabitEthernet2/0, changed state to up
*Dec  1 15:40:23.323: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0, changed state to up
R2(config-if)#exit
R2(config)#int lo0
R2(config-if)#
*Dec  1 15:40:40.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R2(config-if)#ip address 192.168.2.1 255.255.255.224
R2(config-if)#ipv6 address fe80::2:3 link-local
R2(config-if)#ipv6 address 2001:db8:acad:2000::1/64
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#int lo1
R2(config-if)#
*Dec  1 15:41:14.195: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
R2(config-if)#ip address 192.168.2.65 255.255.255.192
R2(config-if)#ipv6 address fe80::2:4 link-local
R2(config-if)#ipv6 address 2001:db8:acad:2001::1/64
R2(config-if)#no shut
R2(config-if)#exit
```

Router R3 :

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#int g1/0
R3(config-if)#exit
R3(config)#ipv6 unicast-routing
R3(config)#int g1/0
R3(config-if)#ip address 10.2.3.3 255.255.255.0
R3(config-if)#ipv6 address fe80::3:1 link-local
R3(config-if)#ipv6 address 2001:db8:acad:1023::3/64
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Dec  1 15:42:53.423: %LINK-3-UPDOWN: Interface GigabitEthernet1/0, changed state to up
*Dec  1 15:42:54.423: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
```

```
R3(config)#int s3/0
R3(config-if)#ip address 10.1.3.3 255.255.255.128
R3(config-if)#ipv6 address fe80::3:2 link-local
R3(config-if)#ipv6 address 2001:db8:acad:1013::3/64
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Dec  1 15:43:33.611: %LINK-3-UPDOWN: Interface Serial3/0, changed state to up
R3(config)#
*Dec  1 15:43:34.627: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
R3(config)#int s3/1
R3(config-if)#ip address 10.1.3.130 255.255.255.128
R3(config-if)#ipv6 address fe80::3:3 link-local
R3(config-if)#ipv6 address 2001:db8:acad:1014::3/64
R3(config-if)#no shut
R3(config-if)#exi
*Dec  1 15:44:21.023: %LINK-3-UPDOWN: Interface Serial3/1, changed state to up
R3(config-if)#exit
R3(config)#
*Dec  1 15:44:22.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/1, changed state to up
```

```
R3(config)#int lo0
R3(config-if)#
*Dec  1 15:44:29.727: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R3(config-if)#ip address 192.168.3.1 255.255.255.224
R3(config-if)#ipv6 address fe80::3:4 link-local
R3(config-if)#ipv6 address 2001:db8:acad:3000::1/64
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#int lo1
R3(config-if)#
*Dec  1 15:45:31.879: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
R3(config-if)#ip address 192.168.3.65 255.255.255.192
R3(config-if)#ipv6 address fe80::3:5 link-local
R3(config-if)#ipv6 address 2001:db8:acad:3001::1/64
R3(config-if)#no shut
R3(config-if)#exit
```

Part 2: Configure and Verify Multi-Protocol BGP on all Routers

In Part 2, you will configure and verify Multi-Protocol BGP on all routers to achieve full connectivity between the routers. The text below provides you with the complete configuration for R1. You will use this to inform your configuration of R2 and R3. The configuration being used here is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: On R1, create the core BGP configuration.

a. Enter BGP configuration mode from global configuration mode, specifying AS 6500

```
R1(config)#
*Dec  1 15:44:33.607: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/1, changed state to up
R1(config)#router bgp 6500
```

b. Configure the BGP router-id for R1.

```
R1(config-router)#bgp router-id 1.1.1.1
```

c. Disable the default IPv4 unicast address family behavior.

```
R1(config-router)#no bgp default ipv4-unicast
```

d. Based on the topology diagram, configure all the designated neighbors for R1.

```
R1(config-router)#neighbor 10.1.2.2 remote-as 500
R1(config-router)#neighbor 10.1.3.3 remote-as 300
R1(config-router)#neighbor 10.1.3.130 remote-as 300
R1(config-router)#neighbor 2001:db8:acad:1012::2 remote-as 500
R1(config-router)#neighbor 2001:db8:acad:1013::3 remote-as 300
R1(config-router)#neighbor 2001:db8:acad:1014::3 remote-as 300
```

Step 2: On R1, configure the IPv4 unicast address family.

a. Enter the IPv4 unicast address family configuration mode.

```
R1(config-router)#address-family ipv4 unicast
```

b. Configure network statements for the IPv4 networks attached to interfaces loopback0 and loopback1. Remember that BGP does not work the same way that an IGP does, and that the network statement has no impact on neighbor adjacency; it is used solely for advertising purposes.

```
R1(config-router-af)#network 192.168.1.0 mask 255.255.255.224
R1(config-router-af)#network 192.168.1.64 mask 255.255.255.192
```

c. Deactivate the IPv6 neighbors and activate the IPv4 neighbors.

```
R1(config-router-af)#no neighbor 2001:db8:acad:1012::2 activate
R1(config-router-af)#no neighbor 2001:db8:acad:1013::3 activate
R1(config-router-af)#no neighbor 2001:db8:acad:1014::3 activate
R1(config-router-af)#neighbor 10.1.2.2 activate
R1(config-router-af)#neighbor 10.1.3.3 activate
R1(config-router-af)#neighbor 10.1.3.130 activate
```

Step 3: On R1, configure the IPv6 unicast address family.

a. Enter the IPv6 unicast address family configuration mode.

```
R1(config-router-af)#address-family ipv6 unicast
```

b. Configure network statements for the IPv6 networks that are attached to interfaces loopback0 and loopback1. Remember that BGP does not work the same way that an IGP does; therefore, the network statement has no impact on neighbor adjacency; it is used solely for advertising purposes.

```
R1(config-router-af)#network 2001:db8:acad:1000::/64
R1(config-router-af)#network 2001:db8:acad:1001::/64
```

c. Activate the IPv6 neighbors that are configured for BGP.

```
R1(config-router-af)#neighbor 2001:db8:acad:1012::2 activate
R1(config-router-af)#neighbor 2001:db8:acad:1013::3 activate
R1(config-router-af)#neighbor 2001:db8:acad:1014::3 activate
```

Step 4: Configure MP-BGP on R2 and R3 as you did in the previous step.

Router R2 :

```
R2(config)#router bgp 500
R2(config-router)#bgp router-id 2.2.2.2
R2(config-router)#no bgp default ipv4-unicast
R2(config-router)#neighbor 10.1.2.1 remote-as 6500
R2(config-router)#neighbor 10.2.3.3 remote-as 300
R2(config-router)#neighbor 2001:db8:acad:1012::1 remote-as 6500
R2(config-router)#neighbor 2001:db8:acad:1023::3 remote-as 300
R2(config-router)#address-family ipv4
R2(config-router-af)#network 192.168.2.0 mask 255.255.255.224
R2(config-router-af)#network 192.168.2.64 mask 255.255.255.192
R2(config-router-af)#neighbor 10.1.2.1 activate
R2(config-router-af)#neighbor 10.2.3.3 activate
R2(config-router-af)#
*Dec  1 15:54:39.699: %BGP-5-ADJCHANGE: neighbor 10.1.2.1 Up
R2(config-router-af)#no neighbor 2001:db8:acad:1012::1 activate
R2(config-router-af)#no neighbor 2001:db8:acad:1023::3 activate
R2(config-router-af)#exit
```

```
R2(config-router)#address-family ipv6
R2(config-router-af)#network 2001:db8:acad:2000::/64
R2(config-router-af)#network 2001:db8:acad:2001::/64
R2(config-router-af)#neighbor 2001:db8:acad:1012::1 activate
R2(config-router-af)#
*Dec  1 15:55:37.663: %BGP-5-ADJCHANGE: neighbor 2001:DB8:ACAD:1012::1 Up
R2(config-router-af)#neighbor 2001:db8:acad:1023::3 activate
R2(config-router-af)#exit
R2(config-router)#
*Dec  1 15:58:05.515: %BGP-5-ADJCHANGE: neighbor 10.2.3.3 Up
R2(config-router)#
*Dec  1 15:59:12.043: %BGP-5-ADJCHANGE: neighbor 2001:DB8:ACAD:1023::3 Up
```

Router R3 :

```
R3(config)#router bgp 300
R3(config-router)#bgp router-id 3.3.3.3
R3(config-router)#no bgp default ipv4-unicast
R3(config-router)#neighbor 10.1.3.1 remote-as 6500
R3(config-router)#neighbor 10.1.3.129 remote-as 6500
R3(config-router)#neighbor 10.2.3.2 remote-as 500
R3(config-router)#neighbor 2001:db8:acad:1013::1 remote-as 6500
R3(config-router)#neighbor 2001:db8:acad:1014::1 remote-as 6500
R3(config-router)#neighbor 2001:db8:acad:1023::2 remote-as 500
R3(config-router)#address-family ipv4
R3(config-router-af)#network 192.168.3.0 mask 255.255.255.224
R3(config-router-af)#network 192.168.3.64 mask 255.255.255.192
R3(config-router-af)#neighbor 10.1.3.1 activate
R3(config-router-af)#
*Dec  1 15:57:53.927: %BGP-5-ADJCHANGE: neighbor 10.1.3.1 Up
R3(config-router-af)#neighbor 10.1.3.129 activate
R3(config-router-af)#
*Dec  1 15:58:01.135: %BGP-5-ADJCHANGE: neighbor 10.1.3.129 Up
R3(config-router-af)#neighbor 10.2.3.2 activate
R3(config-router-af)#
*Dec  1 15:58:05.603: %BGP-5-ADJCHANGE: neighbor 10.2.3.2 Up
R3(config-router-af)#no neighbor 2001:db8:acad:1013::1 activate
R3(config-router-af)#no neighbor 2001:db8:acad:1014::1 activate
R3(config-router-af)#no neighbor 2001:db8:acad:1023::2 activate
R3(config-router-af)#exit
R3(config-router)#address-family ipv6
R3(config-router-af)#network 2001:db8:acad:3000::/64
R3(config-router-af)#network 2001:db8:acad:3001::/64
R3(config-router-af)#neighbor 2001:db8:acad:1013::1 activate
R3(config-router-af)#neighbor 2001:db8:acad:1014::1 activate
R3(config-router-af)#
*Dec  1 15:58:58.411: %BGP-5-ADJCHANGE: neighbor 2001:DB8:ACAD:1013::1 Up
R3(config-router-af)#neighbor 2001:db8:acad:1023::2 activate
R3(config-router-af)#exit
*Dec  1 15:59:12.123: %BGP-5-ADJCHANGE: neighbor 2001:DB8:ACAD:1023::2 Up
```

Step 5: Verify that MP-BGP is operational.

a. Use the `show bgp ipv4 unicast summary` and `show bgp ipv6 unicast summary` commands to verify that BGP has established three IPv4 and three IPv6 adjacencies and received four prefixes from each neighbor.

```
R1#show bgp ipv4 unicast summary
BGP router identifier 1.1.1.1, local AS number 6500
BGP table version is 7, main routing table version 7
6 network entries using 864 bytes of memory
14 path entries using 1120 bytes of memory
5/3 BGP path/bestpath attribute entries using 720 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2800 total bytes of memory
BGP activity 11/0 prefixes, 23/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.2.2        4          500      11      11        7    0    0 00:05:26        4
10.1.3.3        4          300       8      11        7    0    0 00:02:12        4
10.1.3.130      4          300      10      11        7    0    0 00:02:05        4
R1#show bgp ipv6 unicast summary
BGP router identifier 1.1.1.1, local AS number 6500
BGP table version is 6, main routing table version 6
5 network entries using 840 bytes of memory
9 path entries using 936 bytes of memory
5/3 BGP path/bestpath attribute entries using 720 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2592 total bytes of memory
BGP activity 11/0 prefixes, 23/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:DB8:ACAD:1012::2
                4          500      11      10        6    0    0 00:04:48        4
2001:DB8:ACAD:1013::3
                4          300       8      10        6    0    0 00:01:27        4
2001:DB8:ACAD:1014::3
                4          300       0       0        1    0    0 never    Idle
```

b. Use the `show bgp ipv4 unicast` and `show bgp ipv6 unicast` commands to view the specified BGP tables. Note that R1 has multiple paths to each destination network. Take note of the next hop address for the destination networks marked with the “>” symbol.

```
R1#show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  192.168.1.0/27   0.0.0.0                  0         32768 i
 *>  192.168.1.64/26  0.0.0.0                  0         32768 i
 *   192.168.2.0/27   10.1.3.130                             0 300 500 i
 *                    10.1.3.3                               0 300 500 i
 *>                   10.1.2.2                 0             0 500 i
 *   192.168.2.64/26  10.1.3.130                             0 300 500 i
 *                    10.1.3.3                               0 300 500 i
 *>                   10.1.2.2                 0             0 500 i
 *   192.168.3.0/27   10.1.2.2                               0 500 300 i
 *                    10.1.3.130               0             0 300 i
 *>                   10.1.3.3                 0             0 300 i
 *   192.168.3.64/26  10.1.2.2                               0 500 300 i
 *                    10.1.3.130               0             0 300 i
 *>                   10.1.3.3                 0             0 300 i
```

```
R1#show bgp ipv6 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  2001:DB8:ACAD:1001::/64
                       ::                       0         32768 i
 *   2001:DB8:ACAD:2000::/64
                       2001:DB8:ACAD:1013::3
                                                              0 300 500 i
 *>                    2001:DB8:ACAD:1012::2
                                                0             0 500 i
 *   2001:DB8:ACAD:2001::/64
                       2001:DB8:ACAD:1013::3
                                                              0 300 500 i
 *>                    2001:DB8:ACAD:1012::2
                                                0             0 500 i
 *   2001:DB8:ACAD:3000::/64
                       2001:DB8:ACAD:1012::2
                                                              0 500 300 i
 *>                    2001:DB8:ACAD:1013::3
                                                0             0 300 i
 *   2001:DB8:ACAD:3001::/64
                       2001:DB8:ACAD:1012::2
                                                              0 500 300 i
 *>                    2001:DB8:ACAD:1013::3
                                                0             0 300 i
```

c. Use the `show ip route bgp` and `show ipv6 route bgp` commands to view the routing tables. Note that there is only one route to each destination, and that the routes included in the routing table have the same next hop as those with the ">" symbol in the BGP tables.

```
R1#show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:06:23
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:06:23
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.3.0/27 [20/0] via 10.1.3.3, 00:03:08
B        192.168.3.64/26 [20/0] via 10.1.3.3, 00:03:08
```

```
R1#show ipv6 route bgp
IPv6 Routing Table - default - 11 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, la - LISP alt
       lr - LISP site-registrations, ld - LISP dyn-eid, a - Application
B   2001:DB8:ACAD:2000::/64 [20/0]
     via FE80::2:1, GigabitEthernet1/0
B   2001:DB8:ACAD:2001::/64 [20/0]
     via FE80::2:1, GigabitEthernet1/0
B   2001:DB8:ACAD:3000::/64 [20/0]
     via FE80::3:2, Serial3/0
B   2001:DB8:ACAD:3001::/64 [20/0]
     via FE80::3:2, Serial3/0
```

Part 3: Configure and Verify BGP Path Manipulation Settings on all Routers

In Part 3, you will configure path manipulation tools for BGP. The way these tools are being used here is not meant to represent best practice, but to assess your ability to complete the required configurations.

Step 1: Configure ACL-based route filtering.

In this step, you will configure R3 so that it only sends ASN300 networks to R1; it will not tell R1 that it knows about the networks in ASN200.

a. On R1, issue the command `show bgp ipv4 unicast | i 300` to see what prefixes ASN300 is sharing via BGP. Take note of those prefixes that do not originate in ASN300.

```
R1#show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  192.168.1.0/27   0.0.0.0                  0         32768 i
 *>  192.168.1.64/26  0.0.0.0                  0         32768 i
 *>  192.168.2.0/27   10.1.2.2                 0             0 500 i
 *>  192.168.2.64/26  10.1.2.2                 0             0 500 i
 *>  192.168.3.0/27   10.1.3.130               0    250      0 300 i
 *                    10.1.3.3                 0             0 300 i
 *>  192.168.3.64/26  10.1.3.130               0    250      0 300 i
 *                    10.1.3.3                 0             0 300 i
```

b. On R3, configure an access list designed to match the source address and mask of the networks belonging to ASN300:

```
R3(config)#ip access-list extended ALLOWED_TO_R1
R3(config-ext-nacl)#permit ip 192.168.3.0 0.0.0.0 255.255.255.224 0.0.0.0
R3(config-ext-nacl)#permit ip 192.168.3.64 0.0.0.0 255.255.255.192 0.0.0.0
R3(config-ext-nacl)#exit
```

c. On R3, apply the ALLOWED_TO_R1 ACL as a distribute list to the IPv4 neighbor adjacencies with R1.

```
R3(config)#router bgp 300
R3(config-router)#address-family ipv4 unicast
R3(config-router-af)#neighbor 10.1.3.1 distribute-list ALLOWED_TO_R1 out
R3(config-router-af)#neighbor 10.1.3.129 distribute-list ALLOWED_TO_R1 out
R3(config-router-af)#end
R3#
*Dec  1 16:06:59.579: %SYS-5-CONFIG_I: Configured from console by console
```

d. Perform a reset of the IPv4 adjacency with R1 for the outbound traffic without tearing down the session.

```
R3#clear bgp ipv4 unicast 6500 out
```

e. On R1, issue the command `show bgp ipv4 unicast | i 300` to see what prefixes routes ASN300 is now sharing via BGP. All of the prefixes should now originate in ASN300:

```
R1#show bgp ipv4 unicast | i 300
 *>  192.168.3.0/27   10.1.3.130               0    250      0 300 i
 *                    10.1.3.3                 0             0 300 i
 *>  192.168.3.64/26  10.1.3.130               0    250      0 300 i
 *                    10.1.3.3                 0             0 300 i
```

Step 2: Configure prefix-list-based route filtering.

In this step, you will configure R1 so that it only accepts ASN500 networks from R2; it will not accept information about ASN300 networks from R2.

a. On R1, issue the command `show bgp ipv4 unicast | begin 192.168.3` to see what prefixes ASN500 is sharing via BGP. Take note of those prefixes that do not originate in ASN500.

```
R1#show bgp ipv4 unicast | begin 192.168.3
 *   192.168.3.0/27   10.1.2.2                              0 500 300 i
 *                    10.1.3.130               0             0 300 i
 *>                   10.1.3.3                 0             0 300 i
 *   192.168.3.64/26  10.1.2.2                              0 500 300 i
 *                    10.1.3.130               0             0 300 i
 *>                   10.1.3.3                 0             0 300 i
```

b. On R1, configure a prefix list designed to match the source address and mask of networks belonging to ASN500.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip prefix-list ALLOWED_FROM_R2 seq 5 permit 192.168.2.0/24 le 27
```

c. Apply the ALLOWED_FROM_R2 prefix list to the IPv4 neighbor adjacencies for R2.

```
R1(config)#router bgp 6500
R1(config-router)#address-family ipv4 unicast
R1(config-router-af)#neighbor 10.1.2.2 filter-list 1 out
R1(config-router-af)#end
R1#
```

d. Perform a reset of the IPv4 adjacency with R2 for the inbound traffic without tearing down the session.

```
R1#clear bgp ipv4 unicast 500 out
```

e. On R1, issue the command `show bgp ipv4 unicast | i 500` to see what prefixes routes ASN500 is now sharing via BGP. All of the prefixes should now originate in ASN500.

```
R1#show bgp ipv4 unicast | i 500
 *>  192.168.2.0/27   10.1.2.2                 0             0 500 i
 *>  192.168.2.64/26  10.1.2.2                 0             0 500 i
R1#
```

Step 3: Configure an AS-PATH ACL to filter routes being advertised.

In this step, you will configure R1 so that it only sends ASN100 networks to R2; it will not forward information about prefixes from any other ASN to ASN500.

a. On R2, issue the command `show bgp ipv4 unicast | begin Network` to see what prefixes ASN6500 is sharing via BGP. Take note of those prefixes that do not originate in ASN6500. Advertising these routes could set ASN6500 up as a transit AS, and that is not a desirable scenario.

```
R2#show bgp ipv4 unicast | begin Network
*Dec  1 16:34:49.979: %SYS-5-CONFIG_I: Configured from console by console
R2#show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *   192.168.1.0/27   10.2.3.3                               0 300 6500 i
 *>                   10.1.2.1                 0             0 6500 i
 *   192.168.1.64/26  10.2.3.3                               0 300 6500 i
 *>                   10.1.2.1                 0             0 6500 i
 *>  192.168.2.0/27   0.0.0.0                  0         32768 i
 *>  192.168.2.64/26  0.0.0.0                  0         32768 i
 *>  192.168.3.0/27   10.2.3.3                 0             0 300 i
 *                    10.1.2.1                               0 6500 300 i
 *>  192.168.3.64/26  10.2.3.3                 0             0 300 i
 *                    10.1.2.1                               0 6500 300 i
R2#show bgp ipv4 unicast | i 6500
 *   192.168.1.0/27   10.2.3.3                               0 300 6500 i
 *>                   10.1.2.1                 0             0 6500 i
 *   192.168.1.64/26  10.2.3.3                               0 300 6500 i
 *>                   10.1.2.1                 0             0 6500 i
```

b. On R1, configure AS-PATH ACL to match the routes from the local ASN.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip as-path access-list 1 permit ^$
```

c. On R1, apply the AS-PATH ACL as a filter-list on the adjacency configured with R2.

```
R1(config)#router bgp 6500
R1(config-router)#address-family ipv4 unicast
R1(config-router-af)#neighbor 10.1.2.2 filter-list 1 out
R1(config-router-af)#end
R1#
*Dec  2 16:08:38.711: %SYS-5-CONFIG_I: Configured from console by console
```

d. On R1, perform a reset of the IPv4 adjacency with R2 for the outbound traffic without tearing down the session.

```
R1#clear bgp ipv4 unicast 500 out
```

e. On R2, issue the command `show bgp ipv4 unicast | i 6500` to see what prefixes routes ASN6500 is now sharing via BGP. All of the prefixes should now originate in ASN6500.

```
R2#show bgp ipv4 unicast | i 6500
 *   192.168.1.0/27   10.2.3.3                               0 300 6500 i
 *>                   10.1.2.1                 0             0 6500 i
 *   192.168.1.64/26  10.2.3.3                               0 300 6500 i
 *>                   10.1.2.1                 0             0 6500 i
```

Step 4: Configure IPv6 prefix-list-based route filtering.

In this step, you will configure R1 so that it only accepts ASN500 IPv6 networks from R2. It will not accept information about ASN300 IPv6 networks from R2.

a. On R1, issue the command `show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes` to see what IPv6 prefixes ASN500 is sharing via BGP. Take note of those IPv6 prefixes that do not originate in ASN500.

```
R1#show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  2001:DB8:ACAD:2000::/64
                       2001:DB8:ACAD:1012::2
                                                0             0 500 i
 *>  2001:DB8:ACAD:2001::/64
                       2001:DB8:ACAD:1012::2
                                                0             0 500 i
 *   2001:DB8:ACAD:3000::/64
                       2001:DB8:ACAD:1012::2
                                                              0 500 300 i
 *   2001:DB8:ACAD:3001::/64
                       2001:DB8:ACAD:1012::2
                                                              0 500 300 i

Total number of prefixes 4
```

b. On R1, configure an IPv6 prefix list designed to match the source address and mask of networks belonging to ASN500.

```
R1(config)#$-list IPV6_ALLOWED_FROM_R2 seq 5 permit 2001:db8:acad:2000::/64
R1(config)#ipv6 prefix-list IPV6_ALLOWED_FROM_R2 seq 10 permit 2001:db8:acad:2$
```

c. Apply the IPV6_ALLOWED_FROM_R2 prefix list to the IPv6 neighbor adjacencies for R2.

```
R1(config)#router bgp 6500
R1(config-router)#address-family ipv6 unicast
R1(config-router-af)#$01:db8:acad:1012::2 prefix-list IPV6_ALLOWED_FROM_R2 in
R1(config-router-af)#end
R1#
*Dec  2 16:10:21.391: %SYS-5-CONFIG_I: Configured from console by console
```

d. Perform a reset of the IPv6 adjacency with R2 for the inbound traffic without tearing down the session.

R1# clear bgp ipv6 unicast 500 in

e. On R1, issue the command `show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes` to see what IPv6 prefixes routes ASN500 is now sharing via BGP. All of the IPv6 prefixes should now originate in ASN500.

```
R1#show bgp ipv6 unicast neighbors 2001:db8:acad:1012::2 routes
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  2001:DB8:ACAD:2000::/64
                       2001:DB8:ACAD:1012::2
                                                0             0 500 i
 *>  2001:DB8:ACAD:2001::/64
                       2001:DB8:ACAD:1012::2
                                                0             0 500 i

Total number of prefixes 2
```

f. Configure and apply an IPv6 filter to do the same thing on the adjacency with ASN300.

Step 5: Configure BGP path attribute manipulation to effect routing.

In this step, you will configure R1 so that it prefers the next-hop address of 192.168.3.130 over 192.168.3.3, which would normally be the preferred path to ASN300 networks. You will do this by using a prefix list to identify the destination networks and then use a route map to match the prefix list and set the matched networks to have a local preference of 250.

a. On R1, issue the command `show ip route bgp` and take note of the next hop addresses for the 192.168.3.0/27 and 192.168.3.64/26 networks. Then issue the command `show bpg ipv4 unicast` and note that the 10.1.3.130 is a valid next hop (It's just not the best next hop, according to the BGP path selection algorithm.) Lastly, issue the command `show bgp ipv4 unicast 192.168.3.0` to see details about all the paths available and which one was selected.

```
R1#show bgp ipv4 unicast 192.168.3.0
BGP routing table entry for 192.168.3.0/27, version 6
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 3
  300
    10.1.3.130 from 10.1.3.130 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  300
    10.1.3.3 from 10.1.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

b. On R1, configure a prefix list designed to match the source address and mask of networks belonging to ASN300.

R1(config)# ip prefix-list PREFERRED_IPV4_PATH seq 5 permit 192.168.3.0/24 le 27

c. Create a route-map named USE_THIS_PATH_FOR_IPV4 that matches on the prefix list you just created and sets the local preference to 250.

```
R1(config)#route-map USE_THIS_PATH_FOR_IPV4 permit 10
R1(config-route-map)#match ip address prefix-list PERFERRED_IPV4_PATH
R1(config-route-map)#set local-preference 250
R1(config-route-map)#exit
```

d. Next, apply this route map to the BGP neighbor 10.1.3.130.

```
R1(config)#router bgp 6500
R1(config-router)#address-family ipv4 unicast
R1(config-router-af)#neighbor 10.1.3.130 route-map USE_THIS_PATH_FOR_IPV4 in
R1(config-router-af)#end
R1#
*Dec  2 16:12:27.027: %SYS-5-CONFIG_I: Configured from console by console
```

e. Perform a reset of the IPv4 adjacency with R3 for the inbound traffic without tearing down the session.

R1# clear bgp ipv4 unicast 300 in

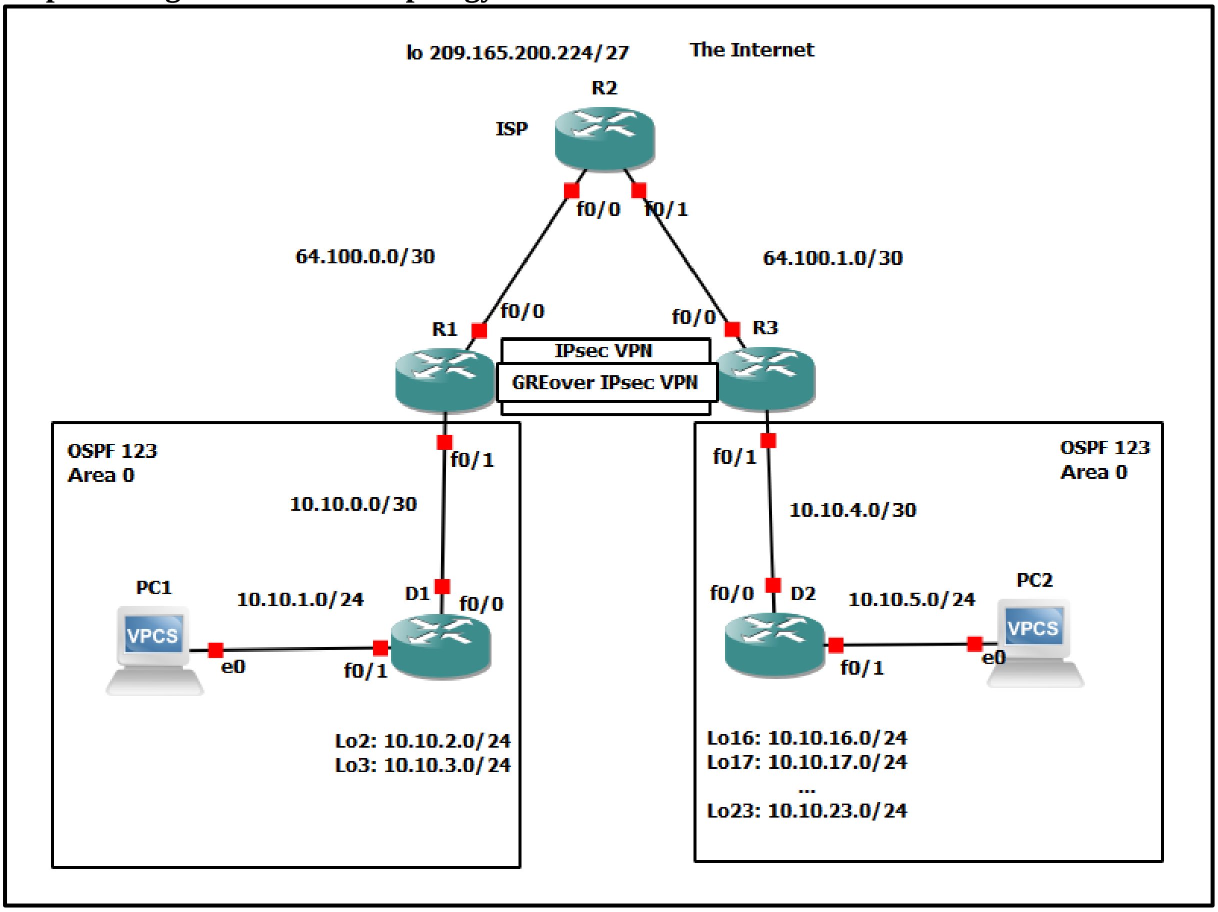
f. On R1, issue the command `show ip route bgp` and take note of the next hop addresses for the 192.168.3.0/27 and 192.168.3.64/26 networks; it should be 10.1.3.130 for both. Issue the command `show bgp ipv4 unicast` and you should see the local preference value in the appropriate column.

```
R1#show ip route bgp | begin Gateway
Gateway of last resort is not set

      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.2.0/27 [20/0] via 10.1.2.2, 00:14:31
B        192.168.2.64/26 [20/0] via 10.1.2.2, 00:14:31
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
B        192.168.3.0/27 [20/0] via 10.1.3.130, 00:00:05
B        192.168.3.64/26 [20/0] via 10.1.3.130, 00:00:05
R1#show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  192.168.1.0/27   0.0.0.0                  0         32768 i
 *>  192.168.1.64/26  0.0.0.0                  0         32768 i
 *>  192.168.2.0/27   10.1.2.2                 0             0 500 i
 *>  192.168.2.64/26  10.1.2.2                 0             0 500 i
 *>  192.168.3.0/27   10.1.3.130               0    250      0 300 i
 *                    10.1.3.3                 0             0 300 i
 *>  192.168.3.64/26  10.1.3.130               0    250      0 300 i
 *                    10.1.3.3                 0             0 300 i
```

SDN Practical 9

Aim: Implement IP Sec Site -to -Site VPNs
What is IP Sec VPN?
IPsec is a group of protocols that are used together to set up encrypted connections between devices. It helps keep data sent over public networks secure. IPsec is often used to set up VPNs, and it works by encrypting IP packets, along with authenticating the source where the packets come from.
Users can access an IPsec VPN by logging into a VPN application, or "client." This typically requires the user to have installed the application on their device.
VPN logins are usually password-based. While data sent over a VPN is encrypted, if user passwords are compromised, attackers can log into the VPN and steal this encrypted data. Using two-factor authentication (2FA) can strengthen IPsec VPN security, since stealing a password alone will no longer give an attacker access.

Step 1: Design the network topology.



- **GRE over IPsec VPN using a Crypto Map – This is a common method of enabling multicast traffic over the VPN.**
- **GRE over IPsec VPN using an IPsec Tunnel Profile – This is a newer method of implementing GRE over IPsec using IPsec profiles.**

Step 2: Configure the network.

- **Router 1 (R1):**

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#no ip domain lookup
R1(config)#line con 0
R1(config-line)#logging sync
R1(config-line)#exec-time 0 0
R1(config-line)#exit
R1(config)#$ #This id R1, Implement GRE over IPSec Site To Site VPN#
R1(config)#interface fastEthernet 0/0
R1(config-if)#description Connection to R2
R1(config-if)#ip address 64.100.0.2 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface fastEthernet 0/1
R1(config-if)#description Connection to D1
R1(config-if)#ip address 10.10.0.1 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#router ospf 123
R1(config-router)#router-id 1.1.1.1
R1(config-router)#auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent across all routers.
R1(config-router)#network 10.10.0.0 0.0.0.3 area 0
R1(config-router)#default-information originate
R1(config-router)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 64.
*Mar  1 00:00:35.959: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:36.459: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:00:36.959: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:37.459: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up100.0.1
R1(config)#end
R1#
*Mar  1 00:00:46.019: %SYS-5-CONFIG_I: Configured from console by console
R1#
*Mar  1 00:01:45.263: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.2 on FastEthernet0/1 from LOADING to FULL, Loading Done
R1#
```

- **Router 2 (R2):**

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#no ip domain lookup
R2(config)#line con 0
R2(config-line)#logging sync
R2(config-line)#exec-time 0 0
R2(config-line)#exit
R2(config)#$ #This is R2. Implement GRE over IpSec Site-To-Site VPN#
R2(config)#interface fastEthernet 0/0
R2(config-if)#description Connection to R1
R2(config-if)#ip address 64.100.0.1 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface fastEthernet 0/1
R2(config-if)#description Connection to R3
R2(config-if)#ip address 64.100.1.1 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface loopback 0
R2(config-if)#description Internet simulated address
R2(config-if)#ip address 209.165.200.225 255.255.255.224
R2(config-if)#exit
R2(config)#ip route 0.0.0.0 0.0.0.0 Loopback0
R2(config)#ip route 10.10.0.0 255.255.252.0 64.100.0.2
R2(config)#ip route 10.10.4.0 255.255.252.0 64.100.1.2
R2(config)#ip route 10.10.16.0 255.255.248.0 64.100.1.2
R2(config)#end
R2#
*Mar  1 00:00:40.163: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:00:40.259: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R2#
*Mar  1 00:00:40.843: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:41.211: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:00:41.843: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2#
*Mar  1 00:00:42.211: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R2#
```

- **Router 3 (R3):**

```
R3(config)#no ip domain lookup
R3(config)#line con 0
R3(config-line)#logging sync
R3(config-line)#exec-time 0 0
R3(config-line)#exit
R3(config)#$ #This is R#, Implement GRE over IPSEC Site - To - Site VPN#
R3(config)#interface f0/0
R3(config-if)#description Connnection to R2
R3(config-if)#ip address 64.100.1.2 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#interface fastEthernet 0/1
R3(config-if)#description Connection to D2
R3(config-if)#ip address 10.10.4.1 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#ip route 0.0.0.0 0.0.0.0 64.100.1.1
R3(config)#router ospf 123
R3(config-router)#router-id 3.3.3.1
R3(config-router)#auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent across all routers.
R3(config-router)#network 10.10.4.0 0.0.0.3 area 0
R3(config-router)#default-information originate
R3(config-router)#exit
R3(config)#end
R3#
*Mar  1 00:00:46.175: %SYS-5-CONFIG_I: Configured from console by console
R3#
*Mar  1 00:00:46.615: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:47.071: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:00:47.615: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3#
*Mar  1 00:00:48.071: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R3#
*Mar  1 00:01:36.179: %OSPF-5-ADJCHG: Process 123, Nbr 3.3.3.2 on FastEthernet0/1 from LOADING to FULL, Loading Done
R3#
```

- **Router 4 (D1):**

```
D1#
D1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#no ip domain lookup
D1(config)#line con 0
D1(config-line)#exec-timeout 0 0
D1(config-line)#logging synchronous
D1(config-line)#exit
D1(config)#$ #This is D1, Implement GRE over IPSEC Site - To - Site VPN#
D1(config)#interface fastEthernet 0/0
D1(config-if)#description Connection to R1
D1(config-if)#ip address 10.10.0.2 255.255.255.252
D1(config-if)#no shutdown
D1(config-if)#exit
D1(config)#interface fastEthernet 0/1
D1(config-if)#description Connection to PC1
D1(config-if)#ip address 10.10.1.1 255.255.255.0
D1(config-if)#no shutdown
D1(config-if)#exit
D1(config)#interface Loopback 2
D1(config-if)#description Loopback to simulate an OSPF network
D1(config-if)#ip address 10.10.2.1 255.255.255.0
D1(config-if)#ip ospf network point-to-point
D1(config-if)#exit
D1(config)#interface Loopback 3
D1(config-if)#description Loopback to simulate an OSPF network
D1(config-if)#ip address 10.10.3.1 255.255.255.0
D1(config-if)#ip ospf network point-to-point
D1(config-if)#exit
D1(config)#ip routing
D1(confi
```

```
D1(confi
*Mar  1 00:01:21.023: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback2, changed state to up
*Mar  1 00:01:21.583: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:01:21.651: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback3, changed state to up
*Mar  1 00:01:21.899: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:01:22.583: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Mar  1 00:01:22.899: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to upg)#router ospf 123
D1(config-router)#router-id 1.1.1.2
D1(config-router)#auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent across all routers.
D1(config-router)#network 10.10.0.0 0.0.3.255 area 0
D1(config-router)#exit
D1(config)#end
*Mar  1 00:01:35.287: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.1 on FastEthernet0/0 from LOADING to FULL, Loading Done
D1(config)#end
D1#
*Mar  1 00:01:39.351: %SYS-5-CONFIG_I: Configured from console by console
D1#
```

- **Router 5 (D2):**

```
D2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
D2(config)#no ip domain lookup
D2(config)#line con 0
D2(config-line)#logging synchronous
D2(config-line)#exec-timeout 0 0
D2(config-line)#$ #This is D2, Implement GRE over IPSEC Site - To - Site VPN#
D2(config)#interface fastEthernet 0/0
D2(config-if)#description Connection to R3
D2(config-if)#ip address 10.10.4.2 255.255.255.252
D2(config-if)#no shutdown
D2(config-if)#exit
D2(config)#interface fastEthernet 0/1
D2(config-if)#description Connection to PC2
D2(config-if)#ip address 10.10.5.1 255.255.255.0
D2(config-if)#no shutdown
D2(config-if)#exit
D2(config)#interface Loopback 16
D2(config-if)#description Loopback to simulate an OSPF network
D2(config-if)#ip address 10.10.16.1 255.255.255.0
D2(config-if)#ip ospf network point-to-point
D2(config-if)#exit
D2(config)#interface Loopback 17
D2(config-if)#description Loopback to simulate an OSPF network
D2(config-if)#ip address 10.10.17.1 255.255.255.0
D2(config-if)#ip ospf network point-to-point
D2(config-if)#exit
D2(config)#interface Loopback 18
D2(config-if)#description Loopback to simulate an OSPF network
D2(config-if)#ip address 10.10.18.1 255.255.255.0
D2(config-if)#ip ospf network point-to-point
D2(config-if)#exit
D2(config)#int
*Mar  1 00:00:58.863: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback16, changed state to up
*Mar  1 00:00:59.391: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:59.495: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback17, changed state to up
*Mar  1 00:00:59.659: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback18, changed state to up
```

```
D2(config-if)#ip ospf network point-to-point
D2(config-if)#exit
D2(config)#interface Loopback 21
D2(config-if)#description Loopback to simulate an OSPF network
D2(config-if)#ip address 10.10.21.1 255.255.255.0
D2(config-if)#ip ospf network point-to-point
D2(config-if)#exit
D2(config)#interface Loopback 22
D2(config-if)#description Loopback to simulate an OSPF network
D2(config-if)#ip address 10.10.22.1 255.255.255.0
D2(config-if)#ip ospf network point-to-point
D2(config-if)#exit
D2(config)#interface Loopback 23
D2(config-if)#description Loopback to simulate an OSPF network
D2(config-if)#ip address 10.10.23.1 255.255.255.0
D2(config-if)#ip osp
*Mar  1 00:01:03.931: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback19, changed state to up
*Mar  1 00:01:04.099: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback20, changed state to up
*Mar  1 00:01:04.267: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback21, changed state to up
*Mar  1 00:01:04.435: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback22, changed state to up
*Mar  1 00:01:04.599: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback23, changed state to upf network point-to-point
D2(config-if)#exit
D2(config)#ip routing
D2(config)#router ospf 123
D2(config-router)#router-id 3.3.3.2
D2(config-router)#auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent across all routers.
D2(config-router)#network 10.10.4.0 0.0.1.255 area 0
D2(config-router)#network 10.10.16.0 0.0.7.255 area 0
D2(config-router)#exit
D2(config)#end
D2#
*Mar  1 00:01:12.127: %SYS-5-CONFIG_I: Configured from console by console
D2#
*Mar  1 00:01:16.203: %OSPF-5-ADJCHG: Process 123, Nbr 3.3.3.1 on FastEthernet0/0 from LOADING to FULL, Loading Done
D2#
```

- **PC1:**

```
PC1> ip 10.10.1.10/24 10.10.1.1
Checking for duplicate address...
PC1 : 10.10.1.10 255.255.255.0 gateway 10.10.1.1

PC1> show ip

NAME        : PC1[1]
IP/MASK     : 10.10.1.10/24
GATEWAY     : 10.10.1.1
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10035
RHOST:PORT  : 127.0.0.1:10036
MTU:        : 1500
```

- **PC2:**

```
PC2> ip 10.10.5.10/24 10.10.5.1
Checking for duplicate address...
PC1 : 10.10.5.10 255.255.255.0 gateway 10.10.5.1

PC2> show ip

NAME        : PC2[1]
IP/MASK     : 10.10.5.10/24
GATEWAY     : 10.10.5.1
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 10037
RHOST:PORT  : 127.0.0.1:10038
MTU:        : 1500

PC2>
```

Step 3: On PC1, verify end-to-end connectivity.

From PC1, ping PC3(10.10.5.10),

```
PC1> ping 10.10.5.10
10.10.5.10 icmp_seq=1 timeout
84 bytes from 10.10.5.10 icmp_seq=2 ttl=59 time=165.928 ms
84 bytes from 10.10.5.10 icmp_seq=3 ttl=59 time=166.898 ms
84 bytes from 10.10.5.10 icmp_seq=4 ttl=59 time=210.310 ms
84 bytes from 10.10.5.10 icmp_seq=5 ttl=59 time=166.057 ms
```

From PC1, ping the first loopback on D3 (10.10.16.1).

```
PC1> ping 10.10.16.1
84 bytes from 10.10.16.1 icmp_seq=1 ttl=251 time=136.169 ms
84 bytes from 10.10.16.1 icmp_seq=2 ttl=251 time=136.467 ms
84 bytes from 10.10.16.1 icmp_seq=3 ttl=251 time=135.927 ms
84 bytes from 10.10.16.1 icmp_seq=4 ttl=251 time=136.417 ms
84 bytes from 10.10.16.1 icmp_seq=5 ttl=251 time=137.554 ms

PC1>
```

Finally, from PC1, ping the default gateway loopback on R2 (209.165.200.225).

```
PC1> ping 209.165.200.225
84 bytes from 209.165.200.225 icmp_seq=1 ttl=253 time=77.614 ms
84 bytes from 209.165.200.225 icmp_seq=2 ttl=253 time=76.150 ms
84 bytes from 209.165.200.225 icmp_seq=3 ttl=253 time=76.291 ms
84 bytes from 209.165.200.225 icmp_seq=4 ttl=253 time=76.200 ms
84 bytes from 209.165.200.225 icmp_seq=5 ttl=253 time=76.179 ms

PC1>
```

Step 4: Verify the routing table of R1 and R3.

Verify the OSPF routing table of R1.

```
R1#show ip route ospf
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O       10.10.1.0/24 [110/200] via 10.10.0.2, 00:08:02, FastEthernet0/1
O       10.10.2.0/24 [110/101] via 10.10.0.2, 00:08:02, FastEthernet0/1
O       10.10.3.0/24 [110/101] via 10.10.0.2, 00:08:02, FastEthernet0/1
R1#
```

Verify the routing table of R3.

```
*Mar  1 00:01:36.179: %OSPF-5-ADJCHG: Process 123, Nbr 3.3.3.2 on FastEthernet0/1 from LOAD
R3#show ip route ospf
     10.0.0.0/8 is variably subnetted, 10 subnets, 2 masks
O       10.10.5.0/24 [110/200] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.16.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.17.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.18.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.19.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.20.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.21.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.22.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
O       10.10.23.0/24 [110/101] via 10.10.4.2, 00:07:56, FastEthernet0/1
R3#
```

Step 5: Configure GRE over IPsec using a Crypto Map on R1.

- **On R1, configure the ISAKMP policy and pre-shared key.**

Like site-to-site VPNs using crypto maps, GRE over IPsec also requires an ISAKMP policy configuration and pre-shared key configured.

In this lab, we will use the following parameters for the ISAKMP policy 10 on R1:

o Encryption: aes 256
o Hash: sha256
o Authentication method: pre-share key
o Diffie-Hellman group: 14
o Lifetime: 3600 seconds (60 minutes / 1 hour)

Configure ISAKMP policy 10 on R1:

```
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#hash sha
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 1
R1(config-isakmp)#lifetime 3600
R1(config-isakmp)#exit
R1(config)#
```

Configure the pre-shared key of cisco123 on R1. This command points to the remote peer R3 G0/0/0 IP address.

```
R1(config)#crypto isakmp key cisco123 address 64.100.1.2
R1(config)#
```

- **On R1, configure the transform set and VPN ACL.**

  Create a transform set called GRE-VPN using AES 256 cipher with ESP and the SHA 256 hash function.

```
R1(config)#crypto ipsec transform-set GRE-VPN esp-aes 256 esp-sha-hmac
R1(cfg-crypto-trans)#
```

Unlike a site-to-site IPsec VPN, the transform must use transport mode. The mode command is used to identify the type of tunnel that will be established. The default is mode tunnel mode. However, GRE over IPsec should be configured using the mode transport command.

```
R1(cfg-crypto-trans)#mode transport
R1(cfg-crypto-trans)#exit
R1(config)#
```

Next, create a named extended ACL called GRE-VPN-ACL that makes the tunnel interface traffic interesting.

```
R1(config)#ip access-list extended GRE-VPN-ACL
R1(config-ext-nacl)#permit gre host 64.100.0.2 host 64.100.1.2
R1(config-ext-nacl)#exit
R1(config)#
```

- **On R1, configure the crypto map and apply it to the interface.**

  Create a crypto map called GRE-CMAP that associates the new GRE-VPN-ACL, transform set, and peer.

```
R1(config)#crypto map GRE-CMAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R1(config-crypto-map)#match address GRE-VPN-ACL
R1(config-crypto-map)#set transform-set GRE-VPN
R1(config-crypto-map)#set peer 64.100.1.2
R1(config-crypto-map)#exit
R1(config)#
```

Finally, assign a crypto map called GRE-MAP on G0/0/0.

```
R1(config)#interface fastEthernet 0/0
R1(config-if)#crypto map GRE-CMAP
R1(config-if)#exit
R1(config)#
*Mar  1 00:17:18.843: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config)#
```

- **On R1, configure the GRE tunnel interface.**
  Configure a GRE tunnel interface as shown. To enable GRE on the tunnel interface, the tunnel mode gre ipv4 command is required. However, this command is enabled by default and will therefore not be configured in our example.

```
R1(config)#interface Tunnel 1
R1(config-if)#bandwidth 4000
R1(config-if)#ip address 172.16.1.1 255.255.255.252
R1(config-if)#ip mtu 1400
R1(config-if)#tunnel source 64.100.0.2
R1(config-if)#tunnel destination 64.100.1.2
R1(config-if)#end
R1#
*Mar  1 00:18:42.883: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:18:43.779: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state to up
R1#
```

Step 6: Configure GRE over IPsec using a Tunnel IPsec Profile on R3.

In this part, we will configure GRE over IPsec using tunnel IPsec profiles on R3.

- **On R3, configure the ISAKMP policy, pre-shared key, and transform set.**
  In this step, we will configure the same parameters for the ISAKMP policy 10 that we configured on R1.

  Configure ISAKMP policy 10 on R3:

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#crypto isakmp policy 10
R3(config-isakmp)#encryption aes 256
R3(config-isakmp)#hash sha
R3(config-isakmp)#authentication pre-share
R3(config-isakmp)#group 1
R3(config-isakmp)#lifetime 3600
R3(config-isakmp)#exit
R3(config)#
```

  Configure the pre-shared key of cisco123 on R1. This command points to the remote peer R3 G0/0/0 IP address.

```
R3(config)#crypto isakmp key cisco123 address 64.100.0.2
R3(config)#
```

  Create a new transform set called GRE-VPN using the same security parameters and transport mode that we configured on R1. Also configure the mode transport command.

```
R3(config)#crypto ipsec transform-set GRE-VPN esp-aes 256 esp-sha-hmac
R3(cfg-crypto-trans)#mode transport
R3(cfg-crypto-trans)#exit
R3(config)#
```

- **On R3, configure the IPsec profile.**
  Instead of a crypto map, we will configure an IPsec profile called GRE-PROFILE using the crypto ipsec profile ipsec-profile-name global configuration command.

```
R3(config)#crypto ipsec profile GRE-profile
R3(ipsec-profile)#
```

In IPsec profile configuration mode, specify the transform set to be negotiated using the set transform-set transform-set-name command. Multiple transform sets can be specified in order of priority. The fist transform-set-name specified is the highest priority.

```
R3(ipsec-profile)#
R3(ipsec-profile)#set transform-set GRE-VPN
R3(ipsec-profile)#exit
R3(config)#
```

- **On R3, configure the tunnel interface.**
  On R3, configure a GRE tunnel interface.

```
R3(config)#interface Tunnel 1
R3(config-if)#bandwidth 4000
R3(config-if)#ip address 172.16.1.2 255.255.255.252
R3(config-if)#ip mtu 1400
R3(config-if)#tunnel source 64.100.1.2
R3(config-if)#tunnel destination 64.100.0.2
R3(config-if)#
*Mar  1 00:23:50.731: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state to down
R3(config-if)#
*Mar  1 00:23:52.259: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state to up
R3(config-if)#
```

Apply the IPsec profile GRE-PROFILE to the Tunnel 1 interface using the tunnel protection ipsec profile profile-name command.

```
R3(config-if)#tunnel protection ipsec profile GRE-profile
R3(config-if)#end
R3#
```

- **On R1 and R3, enable OSPF routing on the tunnel interface.**
  Verify that the GRE over IPsec VPN is operational.
  On R1, perform an extended ping to the R3 10.10.16.1 interface.

```
R1#ping 10.10.16.1 source 10.10.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.16.1, timeout is 2 seconds:
Packet sent with a source address of 10.10.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/107/140 ms
R1#
```

The pings are successful, and it appears that the VPN is operational. On R1, verify the IPsec SA encrypted and decrypted statistics.

```
R1#show crypto ipsec sa | include encrypt | decrypt
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
R1#
```

From D1, trace the path taken to the R3 10.10.16.1 interface.

```
D1#trace 10.10.16.1

Type escape sequence to abort.
Tracing the route to 10.10.16.1

  1 10.10.0.1 16 msec 32 msec 28 msec
  2 64.100.0.1 104 msec 60 msec 48 msec
  3 64.100.1.2 104 msec 72 msec 96 msec
  4 10.10.4.2 104 msec 124 msec 136 msec
D1#
```

On R1, configure OSPF to advertise the tunnel interfaces.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 123
R1(config-router)#network 172.16.1.0 0.0.0.3 area 0
R1(config-router)#
```

On R3, configure OSPF to advertise the tunnel interfaces.

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router ospf 123
R3(config-router)#network 172.16.1.0 0.0.0.3 area 0
R3(config-router)#
*Mar  1 00:30:50.807: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.1 on Tunnel1 from LOADING to FULL, Loading Done
R3(config-router)#
```

Step 7: Verify the GRE over IPsec Tunnel on R1 and R3

Now that the GRE over IPsec has been configured, we must verify that the tunnel interfaces are correctly enabled, that the crypto session is active, and then generate traffic to confirm it is traversing securely over the IPsec tunnel.

- **On R1 and R3, verify the tunnel interfaces.**
  Use the show interfaces tunnel 1 command to verify the interface settings.

```
R1#show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.1/30
  MTU 1514 bytes, BW 4000 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 64.100.0.2, destination 64.100.1.2
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:01, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     12 packets input, 1596 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

On R3, use the show interfaces tunnel 1 command to verify the interface settings.

```
R3#
R3#$tunnel 1 | include is up | Internet address | Enc | Tunnel protocol
Tunnel1 is up, line protocol is up
  Internet address is 172.16.1.2/30
  Tunnel protocol/transport GRE/IP
R3#
```

```
R3#show inter tunnel 1 | include is up|Internet address|Enc|Tunnel protocol
Tunnel1 is up, line protocol is up
  Internet address is 172.16.1.2/30
  Encapsulation TUNNEL, loopback not set
  Tunnel protocol/transport GRE/IP
R3#
```

- **On R1 and R3, verify the crypto settings.**
  On R1, use the show crypto session command to verify the operation of the VPN tunnel.

```
R1#
R1#show crypto session
Crypto session current status

Interface: FastEthernet0/0
Session status: UP-ACTIVE
Peer: 64.100.1.2 port 500
  IKE SA: local 64.100.0.2/500 remote 64.100.1.2/500 Active
  IPSEC FLOW: permit 47 host 64.100.0.2 host 64.100.1.2
        Active SAs: 2, origin: crypto map

R1#
```

On R3, use the show crypto session command to verify the operation of the VPN tunnel.

```
R3#show crypto session
Crypto session current status

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 64.100.0.2 port 500
  IKE SA: local 64.100.1.2/500 remote 64.100.0.2/500 Active
  IPSEC FLOW: permit 47 host 64.100.1.2 host 64.100.0.2
        Active SAs: 2, origin: crypto map

R3#
```

- **On R1 and R3, verify OSPF routing.**

On R1 and R3, verify which interfaces are configured for OSPF using the show ip ospf interface brief command.

```
R1#
R1#show ip ospf interface brief
Interface      PID    Area            IP Address/Mask    Cost  State Nbrs F/C
Tu1            123    0               172.16.1.1/30      250   P2P   1/1
Fa0/1          123    0               10.10.0.1/30       100   DR    1/1
R1#
```

```
R3#
R3#show ip ospf interface brief
Interface      PID    Area            IP Address/Mask    Cost  State Nbrs F/C
Tu1            123    0               172.16.1.2/30      250   P2P   1/1
Fa0/1          123    0               10.10.4.1/30       100   DR    1/1
R3#
```

On R1 and R3, verify the OSPF neighbours using the show ip ospf interface brief command.

```
R1#
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
3.3.3.1           0   FULL/  -        00:00:39    172.16.1.2      Tunnel1
1.1.1.2           1   FULL/BDR        00:00:34    10.10.0.2       FastEthernet0/1
R1#
```

```
R3#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           0   FULL/  -        00:00:30    172.16.1.1      Tunnel1
3.3.3.2           1   FULL/BDR        00:00:32    10.10.4.2       FastEthernet0/1
R3#
```

Verify the R1 routing table for OSPF routes.

```
R1#show ip route ospf
     10.0.0.0/8 is variably subnetted, 14 subnets, 2 masks
O       10.10.1.0/24 [110/200] via 10.10.0.2, 00:06:08, FastEthernet0/1
O       10.10.2.0/24 [110/101] via 10.10.0.2, 00:06:08, FastEthernet0/1
O       10.10.3.0/24 [110/101] via 10.10.0.2, 00:06:08, FastEthernet0/1
O       10.10.4.0/30 [110/350] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.5.0/24 [110/450] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.16.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.17.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.18.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.19.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.20.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.21.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.22.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
O       10.10.23.0/24 [110/351] via 172.16.1.2, 00:06:08, Tunnel1
R1#
```

Verify the R3 routing table for OSPF routes.

```
R3#show ip route ospf
     10.0.0.0/8 is variably subnetted, 14 subnets, 2 masks
O       10.10.0.0/30 [110/350] via 172.16.1.1, 00:06:28, Tunnel1
O       10.10.1.0/24 [110/450] via 172.16.1.1, 00:06:28, Tunnel1
O       10.10.2.0/24 [110/351] via 172.16.1.1, 00:06:28, Tunnel1
O       10.10.3.0/24 [110/351] via 172.16.1.1, 00:06:28, Tunnel1
O       10.10.5.0/24 [110/200] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.16.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.17.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.18.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.19.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.20.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.21.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.22.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
O       10.10.23.0/24 [110/101] via 10.10.4.2, 00:06:28, FastEthernet0/1
R3#
```

Verify that there is an operational logical point-to-point link between R1 and R3 using the GRE tunnel interface.

```
R1#show ip route 172.16.0.0
Routing entry for 172.16.0.0/30, 1 known subnets
  Attached (1 connections)

C       172.16.1.0 is directly connected, Tunnel1
R1#
```

```
R3#show ip route 172.16.0.0
Routing entry for 172.16.0.0/30, 1 known subnets
  Attached (1 connections)

C       172.16.1.0 is directly connected, Tunnel1
R3#
```

- **Test the GRE over IPsec VPN tunnel.**
  From D1, trace the path taken to the R3 10.10.16.1 interface.

```
D1#trace 10.10.16.1

Type escape sequence to abort.
Tracing the route to 10.10.16.1

  1 10.10.0.1 56 msec 32 msec 36 msec
  2 172.16.1.2 68 msec 92 msec 104 msec
  3 10.10.4.2 148 msec 140 msec 96 msec
D1#
```

On R1, verify the IPsec SA encrypted and decrypted statistics.

```
R1#show crypto ipsec sa | include encrypt | decrypt
    #pkts decaps: 70, #pkts decrypt: 70, #pkts verify: 70
R1#
```

The output verifies that the GRE over IPsec VPN tunnel is properly encrypting traffic between both sites. The packets encrypted include the trace packets along with OSPF packets.