

List of Experiments

1. Understanding Working of a Switched Ethernet using CISCO packet tracer.
2. Configuration of a Switched Ethernet using CISCO packet tracer.
3. Design a MAN network using a Single Router and Configure Router with CLI Mode using CISCO packet tracer.
4. Design and Configuring WAN using three Routers with Static Routing using CISCO packet tracer.
5. Create 2 VLANs using 2 switches and configuring it using CISCO packet tracer.
6. Serial and Point to Point router's connection using HDLC, PPP with PAP and CHAP using CISCO packet tracer.
7. Simulation & Analysis of RIP Routing Protocols using CISCO packet tracer.
8. Simulation & Analysis of OSPF Routing Protocols using CISCO packet tracer.
9. Simulation & Analysis of Transport Layer Protocol (TCP) and User Datagram Protocol (UDP) using CISCO packet tracer.
10. Configuring and analysis of DHCP and DNS using CISCO packet tracer.
11. Analysing FTP Traffic using Wireshark.
12. Analysing HTTP Traffic using Wireshark.

Software Requirement

1. Cisco Packet Tracer
Official Site: <https://www.netacad.com/courses/packet-tracer>
Download Link: <https://www.computernetworkingnotes.com/ccna-study-guide/download-packet-tracer-for-windows-and-linux.html>
2. Operating System: Windows

Lab 1: Basic Networking Commands

1. Ping

Ping is used to testing a network host capacity to interact with another host. Just enter the command Ping, followed by the target host's name or IP address. The ping utilities seem to be the most common network tool. This is performed by using the Internet Control Message Protocol, which allows the echo packet to be sent to the destination host and a listening mechanism. If the destination host reply to the requesting host, that means the host is reachable. This utility usually gives a basic image of where there may be a specific networking issue,

2. NetStat

Netstat is a Common TCP – IP networking command-line method present in most Windows, Linux, UNIX, and other operating systems. The netstat provides the statistics and information in the use of the current TCP-IP Connection network about the protocol.

Options are as follows-

- a: This will display all connection and ports
- b: Shows the executable involved in each connection or hearing port
- e: This protocol will combine with the -s and display the ethernet statistics
- n: This will display the address and the port number in the form of numerical
- o: It will display the ID of each connection for the ownership process.
- r: It will display the routing table
- v: When used in combination with -b, the link or hearing port sequence for every executable is shown.

3. Ip Config

The command IP config will display basic details about the device's IP address configuration. Just type IP config in the Windows prompt and the IP, subnet mask and default gateway that the current device will be presented. If you have to see full information, then type on command prompt config-all and then you will see full information. There are also choices to assist you in resolving DNS and DHCP issues.

Ipconfig

Ipconfig/all

4. Hostname

To communicate with each and other, the computer needs a unique address. A hostname can be alphabetic or alphanumeric and contain specific symbols used specifically to define a specific node or device in the network. For example, a hostname should have a domain name

(TLD) of the top-level and a distance between one and 63 characters when used in a domain name system (DNS) or on the Internet.

To get only the computer name, run the following command:

```
hostname -s
```

Similarly, if a user wants to find out which domain system is running, then use the following command.

```
hostname -d
```

The IP address for the hostname can also be retrieved by using the following command.”

```
hostname -i
```

5. Tracert

The tracert command is a Command Prompt command which is used to get the network packet being sent and received and the number of hops required for that packet to reach to target. This command can also be referred to as a traceroute. It provides several details about the path that a packet takes from the source to the specified destination.

The tracert command is available for the Command Prompt in all Windows operating systems.

The syntax for Tracert Command

```
tracert [-d] [-h MaxHops] [-w TimeOut] target
```

Options for tracert Command are as follows-

target: This is the destination, either an IP address or hostname.

-d: This option prevents Tracert from resolving IP addresses to hostnames to get faster results.

-h MaxHops: This Tracert option specifies the maximum number of hops in the search for the target. If the MaxHops option is not specified the target has not been found by 30 hops, then the tracert command will stop looking.

-w timeout: A timeout value must be specified while executing this ping command. It adjusts the amount of time in milliseconds.

6. Nslookup

The Nslookup, which stands for name server lookup command, is a network utility command used to obtain information about internet servers. It provides name server information for the DNS (Domain Name System), i.e. the default DNS server's name and IP Address.

The syntax for Nslookup is as follows.

```
Nslookup
```

```
or
```

```
Nslookup [domain_name]
```

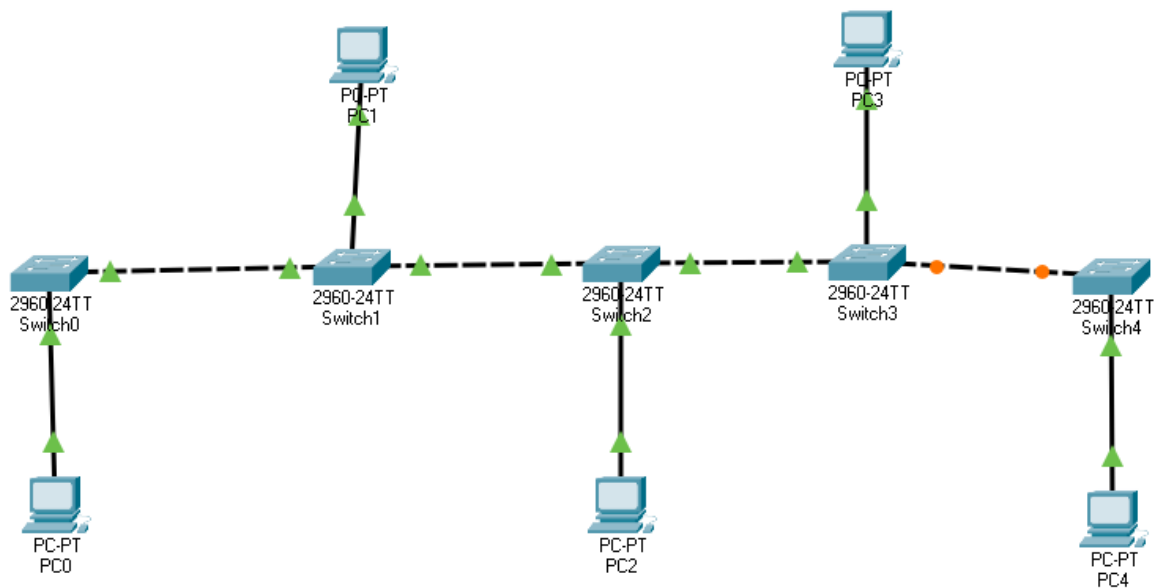
7. Route

In IP networks, routing tables are used to direct packets from one subnet to another. The Route command provides the device's routing tables. To get this result, just type route print. The Route command returns the routing table, and the user can make changes by Commands such as Route Add, Route Delete, and Route Change, which allows modifying the routing table as a requirement.

8. ARP

ARP stands for Address Resolution Protocol. Although network communications can readily be thought of as an IP address, the packet delivery depends ultimately on the media access control (MAC). This is where the protocol for address resolution comes into effect. You can add the remote host IP address, which is an arp -a command, in case you have issues to communicate with a given host. The ARP command provides information like Address, Flags, Mask, IFace, Hardware Type, Hardware Address, etc.

Lab 2: Design a Bus Topology network using Switches.



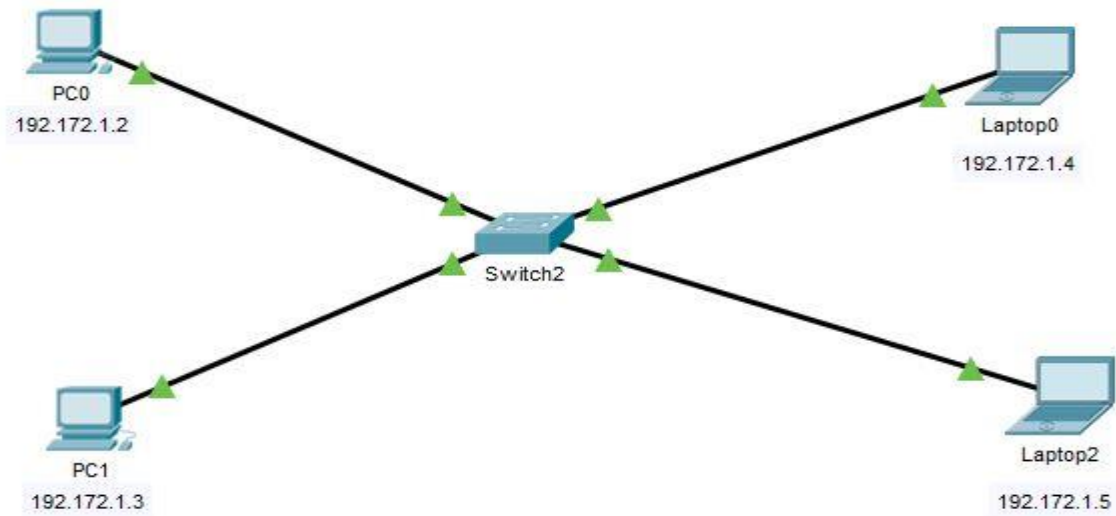
Objectives:

1. Design a BUS Topology using switches with PCs.
2. Verify the connectivity.

Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.172.1.2	255.255.255.0
PC2	NIC	192.172.1.3	255.255.255.0
PC3	NIC	192.172.1.4	255.255.255.0
PC4	NIC	192.172.1.5	255.255.255.0
PC5	NIC	192.172.1.6	255.255.255.0

Lab 3: Design a LAN network using a Single Switch.



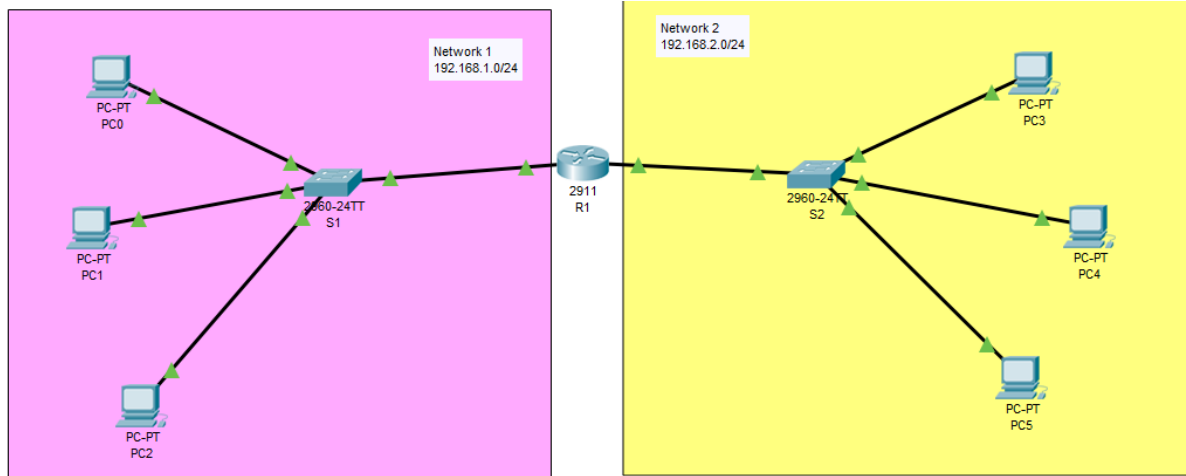
Objectives:

1. Design a LAN using a switch with four PCs.
2. Verify the connectivity.

Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.172.1.2	255.255.255.0
PC2	NIC	192.172.1.3	255.255.255.0
PC3	NIC	192.172.1.4	255.255.255.0
PC4	NIC	192.172.1.5	255.255.255.0

Lab 4: Design a MAN network using a Single Router and Configure Router with CLI Mode.



Objectives:

1. Design two LAN using two switches with three PCs each.
2. Add one Router to connect two LAN networks
3. Verify the connectivity.

Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.168.1.5	255.255.255.0
PC2	NIC	192.168.1.6	255.255.255.0
PC3	NIC	192.168.1.7	255.255.255.0
PC4	NIC	192.168.2.5	255.255.255.0
PC5	NIC	192.168.2.6	255.255.255.0
PC6	NIC	192.168.2.7	255.255.255.0
Router 1	NIC	192.168.1.1	255.255.255.0
Router 1 2 nd int	INC	192.168.2.1	255.255.255.0
LAN 1		192.168.1.0	255.255.255.0
LAN 2		192.168.2.0	255.255.255.0

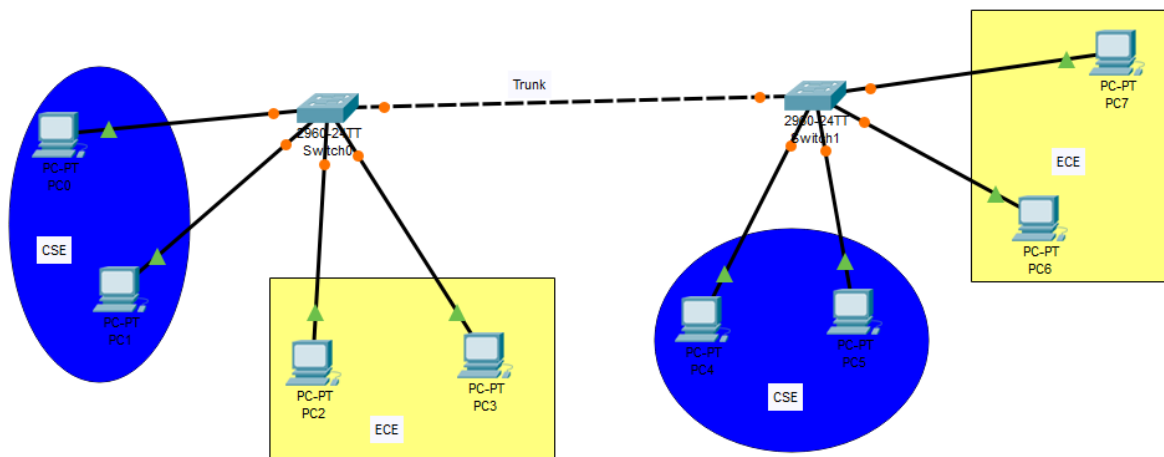
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

```
Router>en
Router>enable
Router#sh ip brief
^
% Invalid input detected at '^' marker.
Router#sh
Router#show ip
Router#show ip briefly
^
% Invalid input detected at '^' marker.
Router#show ip brief
^
% Invalid input detected at '^' marker.
Router#show ip int brief
Interface IP-Address OK? Method Status Protocol
GigabitEthernet0/0 unassigned YES unset administratively down down
GigabitEthernet0/1 unassigned YES unset administratively down down
GigabitEthernet0/2 unassigned YES unset administratively down down
Vlan1 unassigned YES unset administratively down down
Router#conf t
Router#conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int
Router(config)#interface gi
Router(config)#interface gigabitEthernet 0/0
Router(config-if)#ip ad
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#no shutdown
```

Lab 5: Creating and configuring 2 VLANs



Addressing Table:

Device	Interface	IP Address	Subnet Mask	VLAN
PC1	NIC	172.17.10.21	255.255.255.0	10
PC2	NIC	172.17.10.22	255.255.255.0	10
PC3	NIC	172.17.20.23	255.255.255.0	20
PC4	NIC	172.17.20.24	255.255.255.0	20
PC5	NIC	172.17.10.25	255.255.255.0	10
PC6	NIC	172.17.10.26	255.255.255.0	10
PC7	NIC	172.17.20.27	255.255.255.0	20
PC8	INC	172.17.20.28	255.255.255.0	20

Objectives:

3. Verify the default VLAN configuration
4. Configure the VLANs
5. Assign VLANs to ports.

Background / Scenario

Modern switches use virtual local-area networks (VLANs) to improve network performance by separating large Layer 2 broadcast domains into smaller ones. VLANs can also be used as a security measure by controlling which hosts can communicate. In general, VLANs make it easier to design a network to support the goals of an organization.

VLAN trunks are used to span VLANs across multiple devices. Trunks allow the traffic from multiple VLANs to travel over a single link, while keeping the VLAN identification and segmentation intact. In this lab, you will create VLANs on both switches in the topology, assign VLANs to switch access ports, verify that VLANs are working as expected, and then create a VLAN trunk between the two switches to allow hosts in the same VLAN to communicate through the trunk, regardless of which switch the host is actually attached to.

1. Verify the default VLAN configuration

```
Switch>en
Switch>enable
Switch#sh
Switch#show vlan
```

VLAN Name Status Ports

```
-----
1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4
Fa0/5, Fa0/6, Fa0/7, Fa0/8
Fa0/9, Fa0/10, Fa0/11, Fa0/12
Fa0/13, Fa0/14, Fa0/15, Fa0/16
Fa0/17, Fa0/18, Fa0/19, Fa0/20
Fa0/21, Fa0/22, Fa0/23, Fa0/24
Gig0/1, Gig0/2
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active
```

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2

```
-----
1 enet 100001 1500 - - - - 0 0
1002 fddi 101002 1500 - - - - 0 0
1003 tr 101003 1500 - - - - 0 0
1004 fdnet 101004 1500 - - - ieee - 0 0
1005et 101005 1500 - - - ibm - 0 0
```

2. Configure the VLANs

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: Initialize and reload the switches as necessary.

Step 3: Configure basic settings for each switch.

- Console into the switch and enter global configuration mode.
- Copy the following basic configuration and paste it to the running-configuration on the switch.

```
no ip domain-lookup
service password-encryption
enable secret class
banner motd #
Unauthorized access is strictly prohibited. #
line con 0
password cisco
login
logging synchronous
line vty 0 15
```

```
password cisco
logging synchronous
login exit
```

- Configure the host name as shown in the topology.
- Configure the IP address listed in the Addressing Table for VLAN 1 on the switch.
- Administratively deactivate all unused ports on the switch.
- Copy the running configuration to the startup configuration.

Step 4: Test connectivity.

Verify that the PC hosts can ping one another. Note: It may be necessary to disable the PCs firewall to ping between PCs.

PC1 can ping PC5

PC2 can Ping PC6

PC3 can ping PC7

PC4 can ping PC8

Pings to PCs with others fail.

Creat VLANs on the Switches

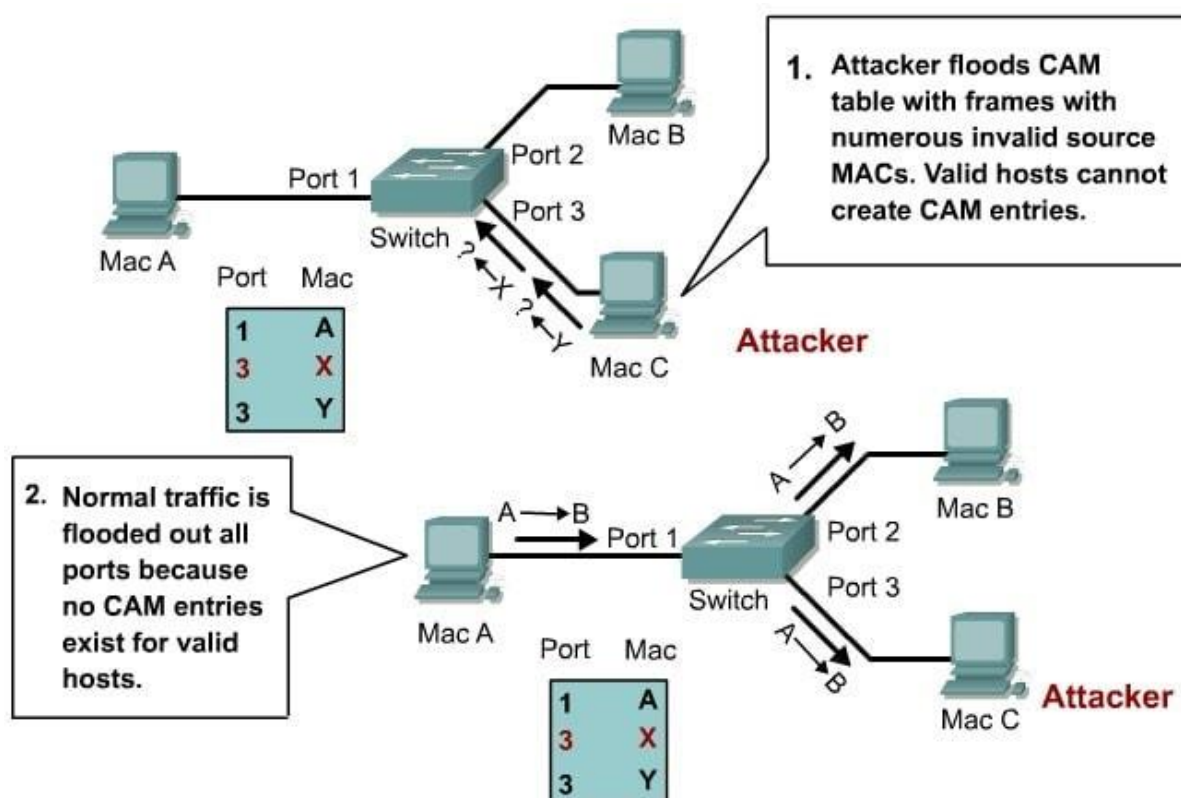
- a. Create the VLANs on S1.
- b. S1(config)# vlan 10
- c. S1(config-vlan)# name Student
- d. S1(config-vlan)# vlan 20
- e. S1(config-vlan)# name Faculty
- f. S1(config-vlan)# vlan 99
- g. S1(config-vlan)# name Management
- h. S1(config-vlan)# end

Create the same VLANs on S2 and s3

Add 2-3 screenshots

Lab 6: Port-Security Configuration to prevent MAC flooding attack

Switch port Security is a **network security** feature that associates specific MAC addresses of devices (such as PCs) with specific interfaces on a switch. This will enable you to restrict access to a given switch interface so that only the authorized devices can use it. If an unauthorized device is connected to the same port, you can define the action that the switch will take, such as discarding the traffic, sending an alert, or shutting down the port.



Now let's configure port security in Packet Tracer.

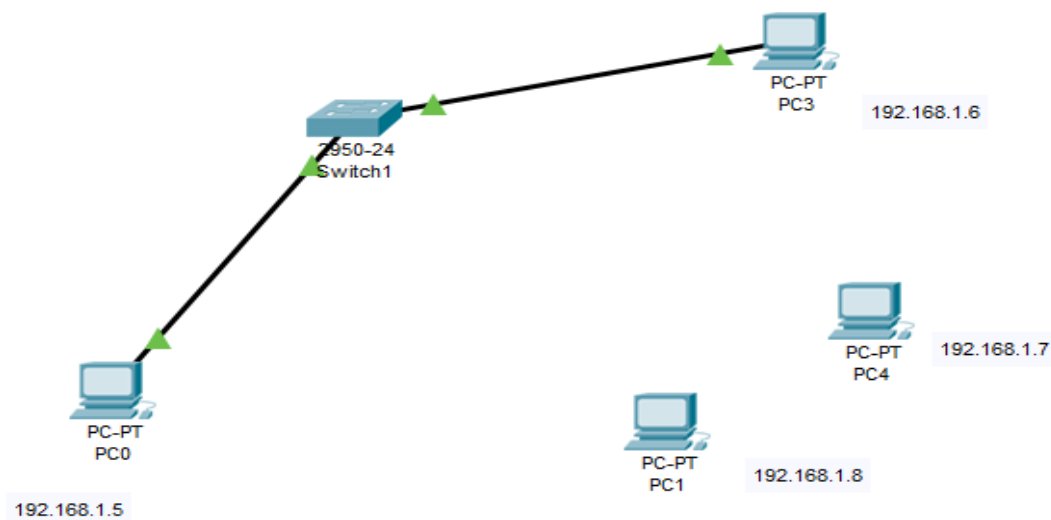
1. Build the network topology:

PC1 connects to fa0/1 and PC2 to fa0/2 of the switch

2. Now configure switch port security on switch interfaces.

We'll configure port security interfaces on fa0/1 and fa0/2. To do this, we'll:

2. Configure the port as an **access port**
3. Enable **port security**
4. Define which **MAC addresses** are allowed to send frames through this interface.



The *sticky* keyword instructs the switch to **dynamically** learn the MAC address of the currently connected host.

You can add these two optional commands.

- defining the action that the switch will take when a frame from an unauthorized device is received. This is done using the `switchport port-security violation {protect | restrict | shutdown}` interface command. All three options discard the traffic from the unauthorized device.
- defining the maximum number of MAC addresses that can be received on the port using the `switchport port-security maximum NUMBER` interface submode command

Port Security Violation Modes:

Security Violation Modes					
Violation Mode	Forwards Traffic	Sends Syslog Message	Displays Error Message	Increases Violation Counter	Shuts Down Port
Protect	No	No	No	No	No
Restrict	No	Yes	No	Yes	No
Shutdown	No	No	No	Yes	Yes

d



switchport port security example

8

In our topology **PC0** is connected with **F0/1** port of switch. Enter following commands to secure **F0/1** port

- Switch>enable
- Switch#configure terminal
- Switch(config)#interface fastethernet 0/1
- Switch(config-if)#switchport mode access
- Switch(config-if)#switchport port-security
- Switch(config-if)#switchport port-security maximum 1
- Switch(config-if)#switchport port-security violation shutdown
- Switch(config-if)#switchport port-security mac-address sticky
- Switch(config-if)#ex
- Switch(config)#ex



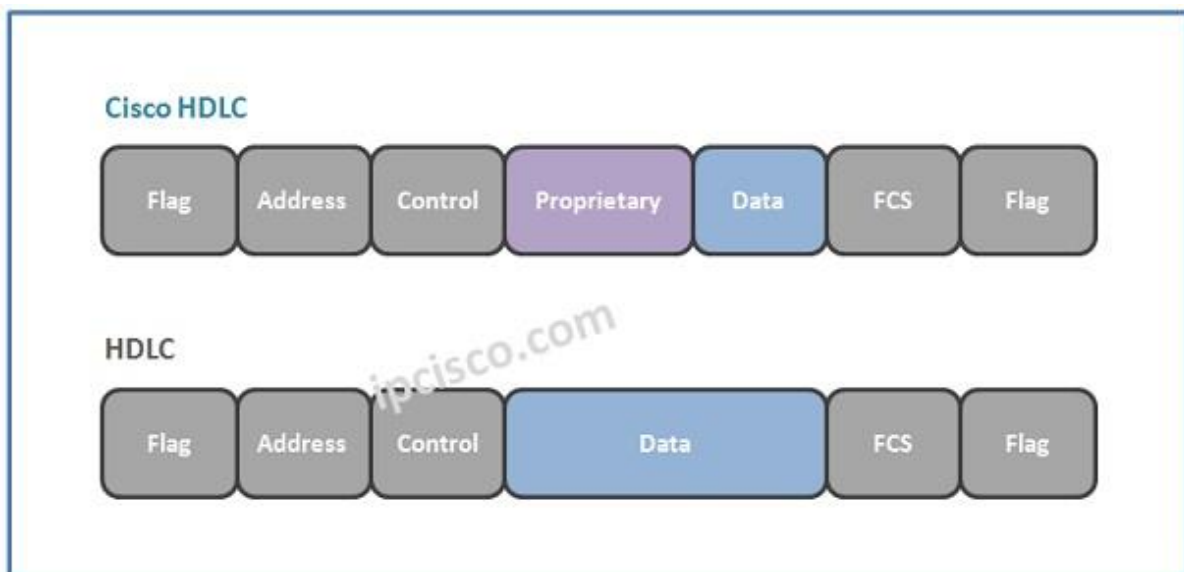
Networks and Communication Department

```
Switch# Show port-security
Switch# sh mac-address-table
Switch# sh ip int brief
```

Lab 7: Router Serial Point to Point Connection HDLC, PPP with PAP and CHAP

HDLC (High-level Data Link Control) is a WAN protocol intended to perform the encapsulation of the data in the data link layer. The encapsulation of the data means to change the format of the data. HDLC protocol is developed by IBM and submitted to the ANSI and ISO for the acceptance as the international standards.

HDLC has two versions. One of them is the standard one and the other is the Cisco proprietary version. The frame of standard version and Cisco proprietary version is similar. Only in Cisco proprietary HDLC, there is one additional proprietary field. Below, you can check both of the frames:



Cisco HDLC is the default enabled WAN protocol of Cisco for Point-to-Point WAN links. And we can use Cisco HDLC only between Cisco devices. Other vendor devices cannot use Cisco HDLC.

Lastly, there is no Authentication mechanism in HDLC. So, security is a concern for this WAN protocol.

PPP (Point to Point Protocol) is also a WAN Encapsulation Protocol that is based on HDLC but we can say that it is the enhanced version of HDLC. There are many additional features in **PPP** if we compare with HDLC.

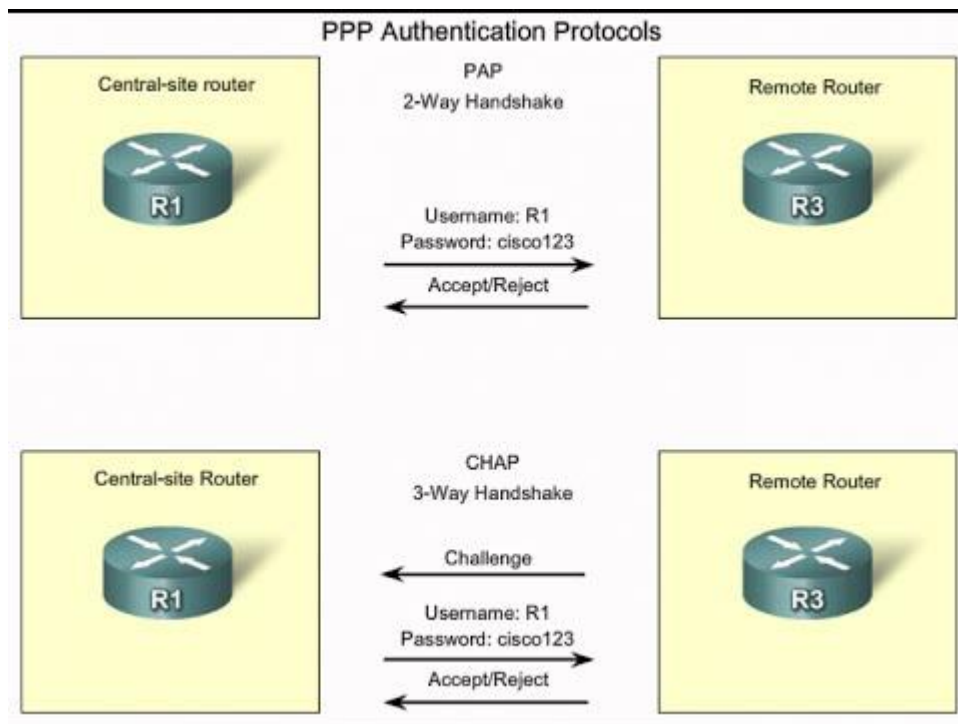
PPP supports two Authentication Protocols. These Authentication Protocols are:

- **PAP (Password Authentication Protocol)**
- **CHAP (Challenge Handshake Authentication Protocol)**

PAP (Password Authentication Protocol) is the simplest Authentication method. It uses 2-way handshake. Both ends send the passwords in “**clear text**” in this method. And passwords are exchanged only at the beginning.

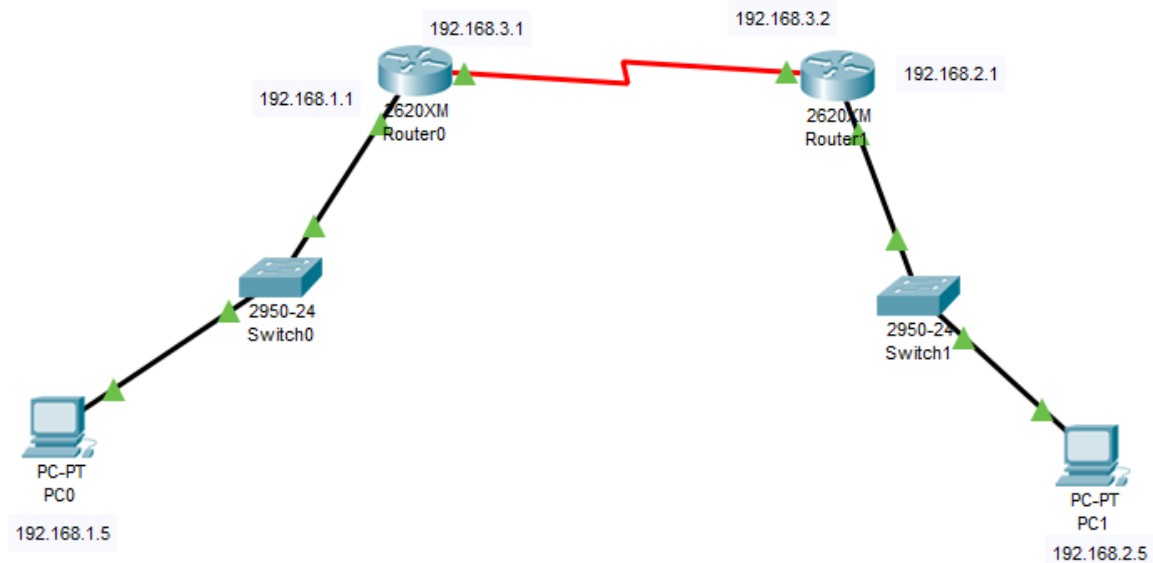
CHAP (Challenge Handshake Authentication Protocol) is the more complex Authentications method. CHAP uses 3-way handshake and with this mechanism it checks the remote node periodically. CHAP uses MD5 hash. One end sends “**Hash**” to other node and the other node also sends a hash. If the hashes are same, then the communication starts.

PAP v/s CHAP



P A P V E R S U S C H A P

PAP	CHAP
A password based authentication protocol used by Point to Point Protocol (PPP) to validate users	A communication protocol that authenticates a user or network host to an authenticating entity
Stands for Password Authentication Protocol	Stands for Challenge Handshake Authentication Protocol
During link establishment, PAP stops working after establishing the authentication, which can lead to attacks on the network	CHAP conducts periodic challenges to make sure that the remote host still has valid password value
Not secure like CHAP	Provide better security than PAP
	Visit www.PEDIAA.com



By default, HDLC encapsulation is works in CISCO devices. In case it is not there
Then by simple command we can configure:

Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.168.1.5	255.255.255.0
PC2	NIC	192.168.2.5	255.255.255.0
Router1 Fast Ethernet 0/1	NIC	192.168.1.1	255.255.255.0
Router1 Serial 0/0	NIC	192.168.3.1	255.255.255.0
Router2 Fast Ethernet 0/1	NIC	192.168.2.1	255.255.255.0
Router2 Serial 0/0	NIC	192.168.3.2	255.255.255.0

Objectives:

1. Design a WAN with 2 Routers
2. Configure the Routers
3. Apply PPP (PAP and CHAP) authentication on it.
4. Verify Connection.

Router Configuration :

```
Router> enable
```

```
Router# configure terminal
```

```
Router(config)# interface gigabitethernet 0/1
```

```
Router(config-if)# ip address 192.168.1.2 255.255.255.0
```

```
Router(config)# ip route 192.168.1.0 255.255.0.0 192.168.2.2
```

```
Router(config-if)# no shutdown
```

```
Router#show interfaces serial 0/0/0
```

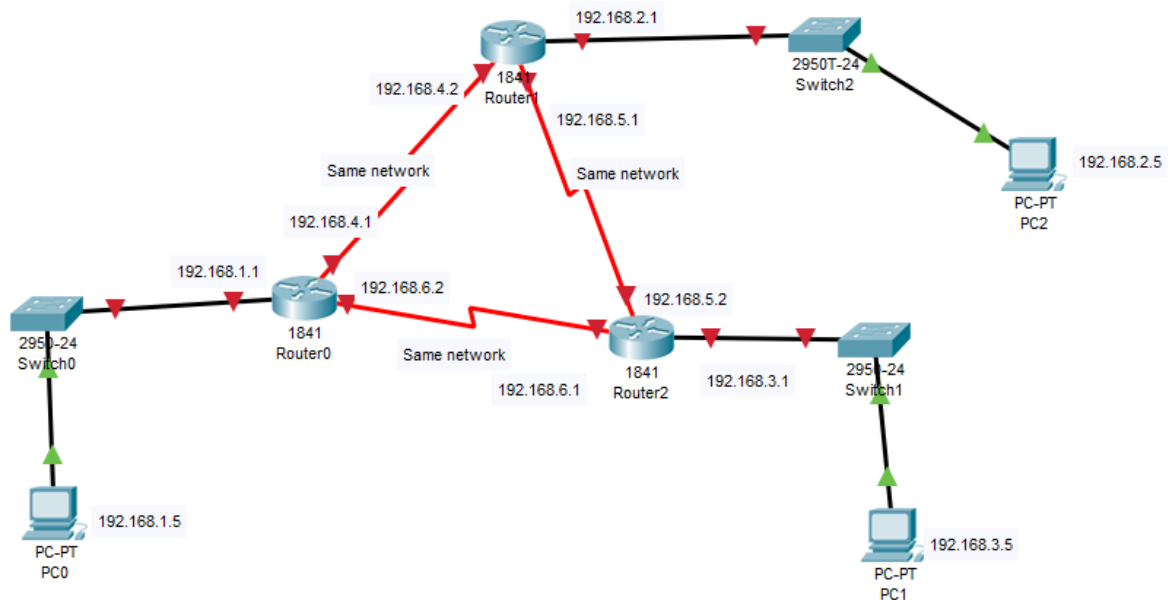
```
Router#configure terminal
Router(config)#interface serial 0/0/0
Router(config-if)#encapsulation ppp
Router(config)#exit
Router#show interfaces serial 0/0/0
```

```
Router#configure terminal
Router(config)#interface serial 0/0/0
Router(config-if)#encapsulation hdlc
Router(config-if)#shutdown
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit
Router#show interfaces serial 0/0/0
```

```
Router>enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#username R2 password vinita
R1(config)#interface serial 0/0/0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication chap
R1(config-if)#exit
R1(config)#
```

```
Router>enable
Router#configure terminal
Router(config)#hostname R2
R2(config)#username R1 password vinita
R2(config)#interface serial 0/0/0
R2(config-if)#encapsulation ppp
R2(config-if)#ppp authentication chap
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/0,
changed state to up
R2(config)#
```

Lab 8: Design and Configuring WAN using three Routers using serial DCE cables with Static Routing



Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.168.1.5	255.255.255.0
PC2	NIC	192.168.2.5	255.255.255.0
PC3	NIC	192.168.3.5	255.255.255.0
Router 1 FastEthernet 0/0		192.168.1.1	255.255.255.0
Router 1 Serial 0/0		192.168.4.1	255.255.255.0
Router 1 Serial 0/1		192.168.6.2	255.255.255.0
Router 2 FastEthernet 0/0	NIC	192.168.2.1	255.255.255.0
Router 2 Serial 0/0	INC	192.168.4.2	255.255.255.0
Router 2 Serial 0/1	INC	192.168.5.1	255.255.255.0
Router 3 FastEthernet 0/0	INC	192.168.3.1	255.255.255.0
Router 3 Serial 0/0	INC	192.168.5.2	255.255.255.0
Router 3 Serial 0/1	INC	192.168.6.1	255.255.255.0

Objectives:

1. Design a WAN network using three router, three switch and 3 PCs
2. Configure 3 Routers with static routing
3. Verify the connectivity by simulation

Static Routing Table:

Router 1				
Network	192.168.2.0	11.0.0.0	192.168.3.0	12.0.0.0
Musk	255.255.255.0	255.0.0.0	255.255.255.0	255.0.0.0
Next hop	11.0.0.2	11.0.0.2	11.0.0.2	11.0.0.2
Router 2				
Network	192.168.1.0	11.0.0.0	192.168.3.0	12.0.0.0
Musk	255.255.255.0	255.0.0.0	255.255.255.0	255.0.0.0
Next hop	11.0.0.1	11.0.0.1	12.0.0.2	12.0.0.2
Router 2				
Network	192.168.1.0	11.0.0.0	12.0.0.0	192.168.2.0
Musk	255.255.255.0	255.0.0.0	255.0.0.0	255.255.255.0
Next hop	12.0.0.1	12.0.0.1	12.0.0.1	12.0.0.1

Router Configuration:

```
Router>enable
```

```
Router#
```

```
Router#configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Router(config)#interface FastEthernet0/0
```

```
Router(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
Router(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
Router(config-if)#no shutdown
```

```
Router(config-if)#
```

```
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

```
Router(config-if)#exit
```

```
Router(config)#interface Serial0/0
```

```
Router(config-if)#ip address 11.0.0.1 255.0.0.0
```

```
Router(config-if)#ip address 11.0.0.1 255.0.0.0
```

```
Router(config-if)#no shutdown
```

```
Router(config-if)#
```

```
Router(config-if)#exit
```

```
Router(config)#
```

```
Router(config)#ip route 192.168.2.0 255.255.255.0 11.0.0.2
```

```
Router(config)#ip route 11.0.0.0 255.0.0.0 11.0.0.2
```

```
Router(config)#ip route 192.168.3.0 255.255.255.0 11.0.0.2
```

```
Router(config)#ip route 12.0.0.0 255.255.255.0 11.0.0.2
```

```
Router(config)#no ip route 12.0.0.0 255.255.255.0 11.0.0.2
```

```
Router(config)#ip route 12.0.0.0 255.0.0.0 11.0.0.2
```

```
Router(config)#
```

```
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
```

Lab 9: Design a network using fixed length Subnetting for a class C Ipv4 address and configure it in Router.

192.168.1.0 /27_I

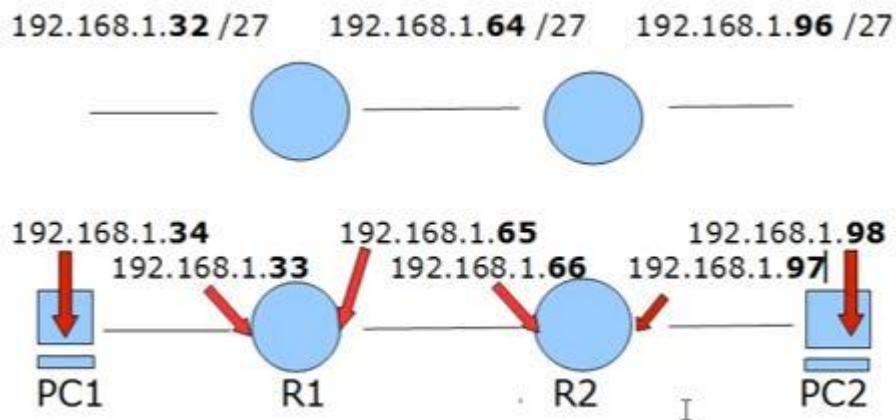
255.255.255.224

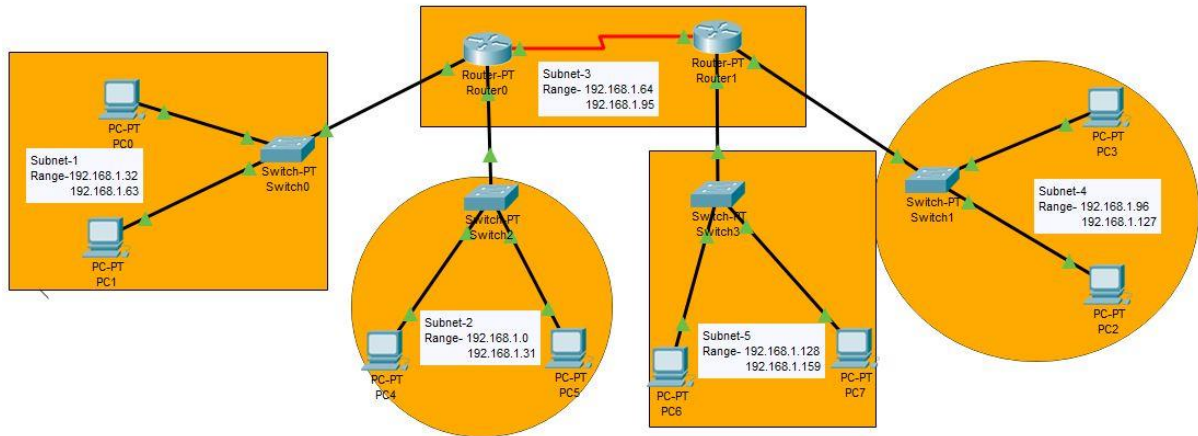
11111111.11111111.11111111.11**1**00000

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
 128 64 **32** 16 8 4 2 1 magic # 32

Networks

0 – 31	128 – 159
32 – 63	160 – 191
64 – 95	192 – 223
96 – 127	224 – 255





Objectives:

1. Design Subnet network using two routers and switches with two PCs each.
2. Configure Subnet
3. Verify the connectivity.

Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.168.1.34	255.255.255.224
PC2	NIC	192.168.1.35	255.255.255.224
PC3	NIC	192.168.1.99	255.255.255.224
PC4	NIC	192.168.1.100	255.255.255.224
Router 1 fa0/1	NIC	192.168.1.33	255.255.255.224
Router 1 Serial 0/2	INC	192.168.1.65	255.255.255.224
Router 2 Serial 0/2		192.168.1.66	255.255.255.224
Router 2 fa0/1		192.168.1.97	255.255.255.224

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

```
Router>en
Router>enable
Router#sh ip brief
^
```

% Invalid input detected at '^' marker.

Router#sh

Router#show ip

Router#show ip briefly

^

% Invalid input detected at '^' marker.

Router#show ip brief

^

% Invalid input detected at '^' marker.

Router#show ip int brief

Interface IP-Address OK? Method Status Protocol

GigabitEthernet0/0 unassigned YES unset administratively down down

GigabitEthernet0/1 unassigned YES unset administratively down down

GigabitEthernet0/2 unassigned YES unset administratively down down

Vlan1 unassigned YES unset administratively down down

Router#conf t

Router#conf terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#int

Router(config)#interface gi

Router(config)#interface gigabitEthernet 0/0

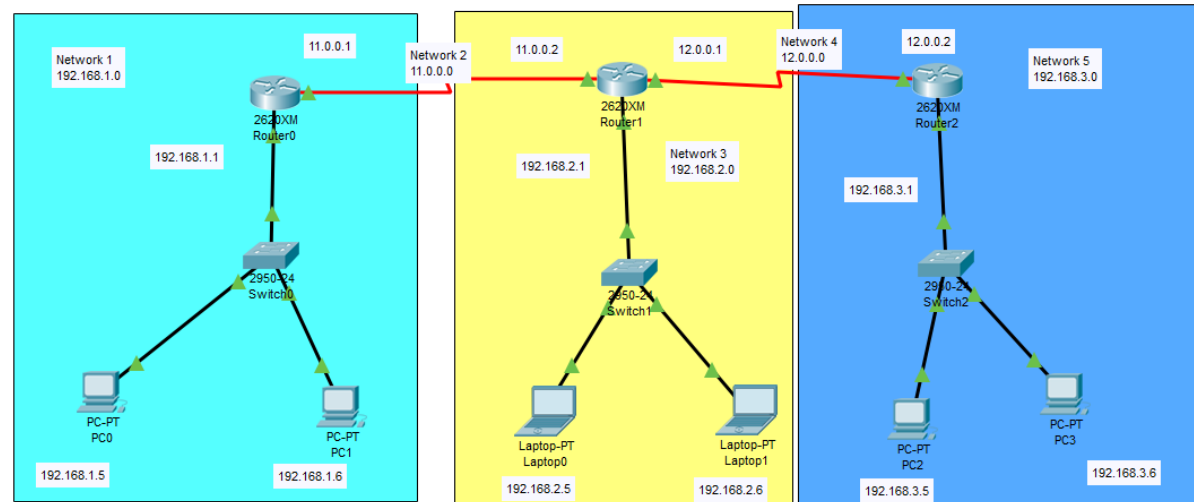
Router(config-if)#ip ad

Router(config-if)#ip address 192.168.1.1 255.255.255.0

Router(config-if)#no shut

Router(config-if)#no shutdown

Lab 10: Design and Configuring WAN using three Routers with Static Routing



Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.168.1.5	255.255.255.0
PC2	NIC	192.168.1.6	255.255.255.0
PC3	NIC	192.168.2.5	255.255.255.0
PC4	NIC	192.168.2.6	255.255.255.0
PC5	NIC	192.168.3.5	255.255.255.0
PC6	NIC	192.168.3.6	255.255.255.0
Router 1 FastEthernet 0/0	NIC	192.168.1.1	255.255.255.0
Router 1 Serial 0/0	INC	11.0.0.1	255.0.0.0
Router 2 Serial 0/0	INC	11.0.0.2	255.0.0.0
Router 2 Serial 0/1	INC	12.0.0.1	255.0.0.0
Router 2 FastEthernet 0/0	INC	192.168.2.1	255.255.255.0
Router 3 Serial 0/0	INC	12.0.0.2	255.0.0.0
Router 3 FastEthernet 0/0	INC	192.168.3.1	255.255.255.0

Objectives:

1. Design a WAN network using three router, three switch and 6 PCs
2. Configure 3 Routers with static routing
3. Verify the connectivity by simulation

Static Routing Table:

Router 1				
Network	192.168.2.0	11.0.0.0	192.168.3.0	12.0.0.0
Musk	255.255.255.0	255.0.0.0	255.255.255.0	255.0.0.0
Next hop	11.0.0.2	11.0.0.2	11.0.0.2	11.0.0.2
Router 2				
Network	192.168.1.0	11.0.0.0	192.168.3.0	12.0.0.0
Musk	255.255.255.0	255.0.0.0	255.255.255.0	255.0.0.0
Next hop	11.0.0.1	11.0.0.1	12.0.0.2	12.0.0.2
Router 2				
Network	192.168.1.0	11.0.0.0	12.0.0.0	192.168.2.0
Musk	255.255.255.0	255.0.0.0	255.0.0.0	255.255.255.0
Next hop	12.0.0.1	12.0.0.1	12.0.0.1	12.0.0.1

Router Configuration:

```
Router>enable
```

```
Router#
```

```
Router#configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Router(config)#interface FastEthernet0/0
```

```
Router(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
Router(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
Router(config-if)#no shutdown
```

```
Router(config-if)#
```

```
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

```
Router(config-if)#exit
```

```
Router(config)#interface Serial0/0
```

```
Router(config-if)#ip address 11.0.0.1 255.0.0.0
```

```
Router(config-if)#ip address 11.0.0.1 255.0.0.0
```

```
Router(config-if)#no shutdown
```

```
Router(config-if)#
```

```
Router(config-if)#exit
```

```
Router(config)#
```

```
Router(config)#ip route 192.168.2.0 255.255.255.0 11.0.0.2
```

```
Router(config)#ip route 11.0.0.0 255.0.0.0 11.0.0.2
```

```
Router(config)#ip route 192.168.3.0 255.255.255.0 11.0.0.2
```

```
Router(config)#ip route 12.0.0.0 255.255.255.0 11.0.0.2
```

```
Router(config)#no ip route 12.0.0.0 255.255.255.0 11.0.0.2
```

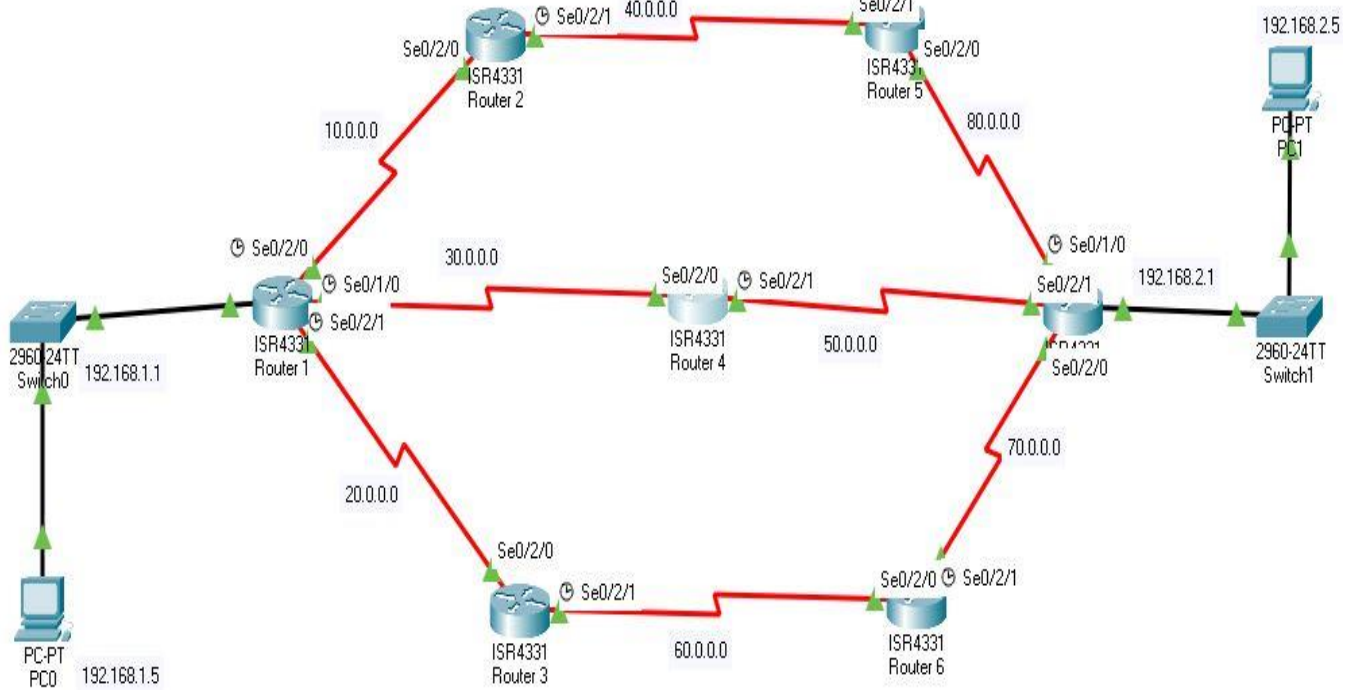
```
Router(config)#ip route 12.0.0.0 255.0.0.0 11.0.0.2
```

```
Router(config)#
```

```
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
```

Lab 11: Design and configure a network using RIP Routing protocol.



Objectives:

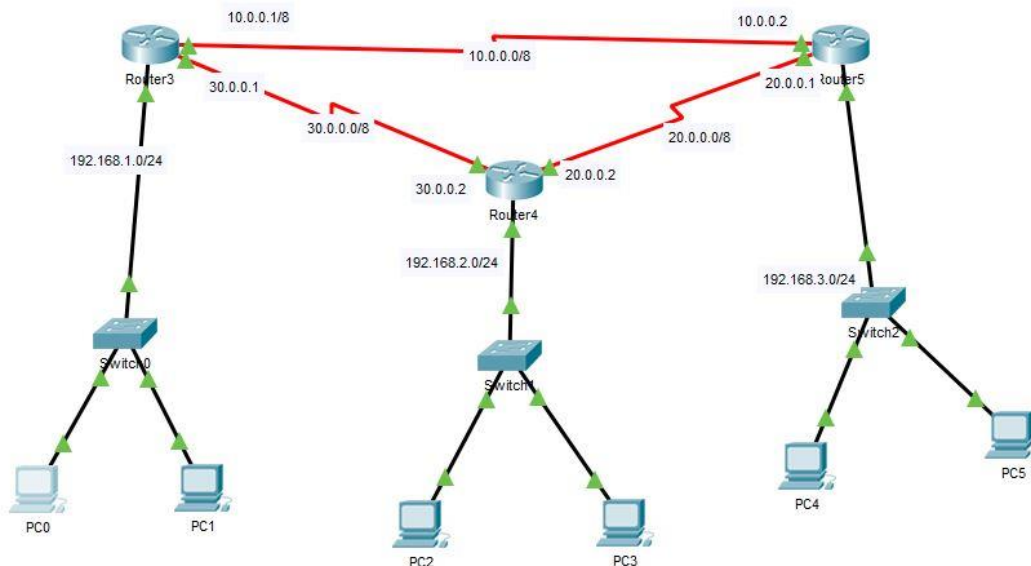
1. Design a network using seven routers and switches with one source and one destination PC
2. Configure all the routers with RIP routing.
3. Verify the connectivity.

Addressing Table:

Device	Interface	IP Address	Subnet Mask
PC1	NIC	192.168.1.5	255.255.255.0
PC2	NIC	192.168.2.5	255.255.255.0
Router 1 gi 0/0/0		192.168.1.1	255.255.255.0
Router 1 Serial 0/2/0		10.0.0.1	255.0.0.0
Router 1 Serial 0/2/1		20.0.0.1	255.0.0.0
Router 1 Serial 0/1/0		30.0.0.1	255.0.0.0
Router 2 Serial 0/2/0		10.0.0.2	255.0.0.0
Router 2 Serial 0/2/1		40.0.0.1	255.0.0.0
Router 3 Serial 0/2/0		20.0.0.2	255.0.0.0
Router 3 Serial 0/2/1		60.0.0.1	255.0.0.0
Router 4 Serial 0/2/0		30.0.0.2	255.0.0.0
Router 4 Serial 0/2/1		50.0.0.1	255.0.0.0
Router 5 Serial 0/2/0		80.0.0.1	255.0.0.0
Router 5 Serial 0/2/1		40.0.0.2	255.0.0.0
Router 6 Serial 0/2/0		60.0.0.2	255.0.0.0
Router 6 Serial 0/2/0		70.0.0.1	255.0.0.0
Router 7 Serial 0/2/0		70.0.0.2	255.0.0.0
Router 7 Serial 0/2/1		50.0.0.2	255.0.0.0
Router 7 Serial 0/1/0		80.0.0.2	255.0.0.0
Router 7 gi 0/0/0		192.168.2.1	255.255.255.0

Configuration:

Lab 12: Design and configure a network using OSPF Routing protocol.



Objectives:

1. Design a network using three routers and switches with two PCs each.
2. Configure all the routers with OSPF routing.
3. Verify the connectivity.

Addressing Table:

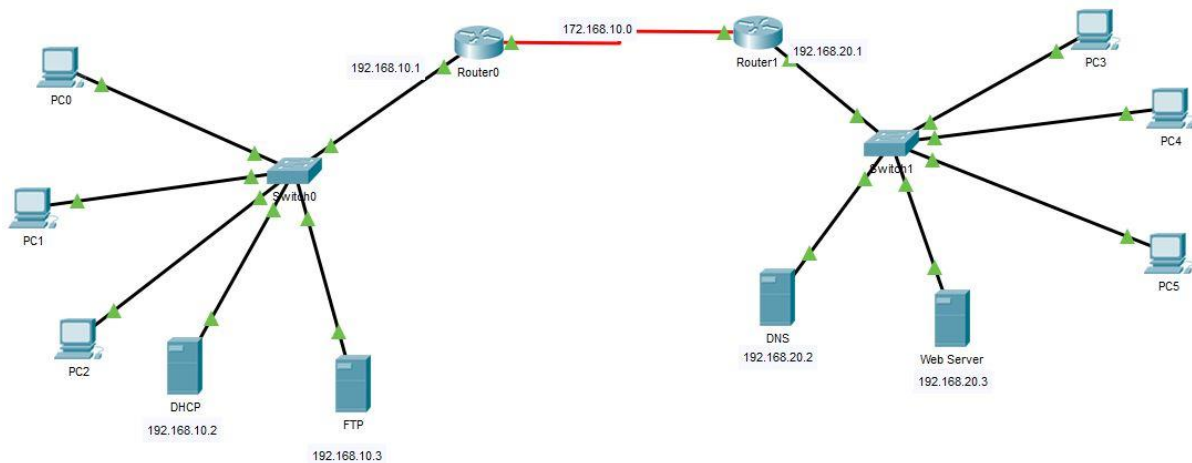
Device	Area	IP Address	wild card bits	Subnet Mask
PC1	0	192.168.1.5		255.255.255.0
PC2	0	192.168.1.6		255.255.255.0
PC3	0	192.168.2.5		255.255.255.0
PC4	0	192.168.2.6		255.255.255.0
PC5	0	192.168.3.5		255.255.255.0
PC6	0	192.168.3.6		255.255.255.0
Router 1 fa0/1	0	192.168.1.1	0.0.0.255	255.255.255.0
Router 1 Serial 2/0	0	10.0.0.1	0.255.255.255	255.0.0.0
Router 1 Serial 2/0	0	10.0.0.2	0.255.255.255	255.0.0.0
Router 2 fa0/1	0	192.168.2.1	0.0.0.255	255.255.255.0
Router 2 Serial 2/0	0	20.0.0.1	0.255.255.255	255.0.0.0
Router 2 Serial 3/0	0	20.0.0.2	0.255.255.255	255.0.0.0
Router 3 fa 0/1	0	192.168.3.1	0.0.0.255	255.255.255.0
Router 3 Serial 2/0	0	30.0.0.1	0.255.255.255	255.0.0.0
Router 3 Serial 2/0	0	30.0.0.2	0.255.255.255	255.0.0.0

Configuration:

```
Router(config)#router ospf ?
<1-65535> Process ID
Router(config)#router ospf 10
Router(config-router)#router
Router(config-router)#router-id 1.1.1.1
Router(config-router)#net
Router(config-router)#network 192.168.1.0?
A.B.C.D
Router(config-router)#network 192.168.1.0
% Incomplete command.
Router(config-router)#network 192.168.1.0 ?
A.B.C.D OSPF wild card bits
Router(config-router)#network 192.168.1.0 0.0.0.255
% Incomplete command.
Router(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#pass
Router(config-router)#passive-interface gi
Router(config-router)#passive-interface gigabitEthernet 0/0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#cop
Router#copy r
Router#copy running-config st
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#!
```

Lab 13: Design and configure a network using DHCP, DNS, Web Server and FTP Protocol.



Objectives:

1. Design a network using two routers and switches with 3 - 4 PCs each and one DHCP, DNS, Web, and FTP server in the network.
2. Configure all the routers with any routing scheme.
3. Configure all the servers.
4. Verify the servers working.

Addressing Table:

Device	IP Address	Subnet Mask
PC1		
PC2		
PC3		
PC4		
PC5		
PC6		
DHCP Server	192.168.10.2	255.255.255.0
FTP Server	192.168.10.3	255.255.255.0
DNS Server	192.168.20.2	255.255.255.0
Web/HTTP Server	192.168.20.3	255.255.255.0
Router 1 giga0/0/0	192.168.10.1	255.255.255.0
Router 1 Serial 0/1/0	172.168.10.1	255.255.0.0
Router 2 giga0/0/0	192.168.20.1	255.255.255.0
Router 2 Serial 0/1/0	172.168.10.2	255.255.0.0