



MITSoE S.Y CSE

**MIT Art, Design and Technology University, Rajbaug,  
Pune MIT School of Engineering**  
Department of Computer Science & Engineering

**LAB MANUAL**

Subject: Programming Lab-IV (DCCN) (Second Year  
CSE) Subject Code: 18BTCS412/18BTNS412/18BTIS412  
Academic Year: 2020-2021

Prepared by: Prof. Sagar. P. Jaikar

Course Code	Course Title			Category	
18BTCS412	Programming Lab-IV (DCCN)			Core	
Contact Hours per Week			CA	FE	Credits
L	T	D/P			
0	0	4	40	60	1
<b>Prerequisite:</b> Data Communication & Computer Networks.					
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>· To understand basic Modulation Techniques (Data Communication.)</li> <li>· To understand basic networking commands &amp; utilities.</li> <li>· To study different networking protocols</li> <li>· To explore various real world applications based on Computer Networks</li> </ul>					

### Programming Lab-IV (DCCN)- Assignments

#### Assignment 1

To study types of cables (Twisted Pair Cable, Co-axial cable, fiber optic cable), connectors (RJ-45, BNC connector) and Networking devices(Hub, Switch, Router, Gateway).

#### Assignment 2

To study and analyze various modulation Technique

- A) To study and analyze Amplitude Modulation.
- B) To study and analyze Frequency Modulation.
- C) To study and analyze Pulse code Modulation.

#### Assignment 3

To Study networking commands with the analysis of commands

- 1) Ping 2) Tracert 3) ARP 4) Netstat 5) NSlookup 6) IPConfig 7) Whois 8) FTP **Assignment 4**

Implement Program to determine IP Address class, Network ID & Host ID of an IPv4 Address.

### **Assignment 5**

To implement the entire Topologies configuration in Cisco packet tracer (Star, Mesh, Bus, Ring, Hybrid Topology) **Assignment 6**

Implement simple Static Routing & Dynamic routing in Cisco Packet Tracer.

### **Assignment 7**

To Design & Configure VLAN by using Packet Tracer

### **Assignment 8**

To Configure Routing Protocols (RIP/OSPF/BGP)

### **Assignment 9**

Design & configure small network by using Sub net.

### **Assignment 10**

To design & implement Socket Programming using TCP

### **Assignment 11**

To design & implement Socket Programming using UDP

### **Course Outcomes:**

After successful completion of the course, the students would be able to:

- Configure various routing protocols from TCP/IP Protocol Stack.
- Design and Configure a Local Area Network
- Administrate the Basic Computer Networking Problems.
- Apply and Analyze the Advanced Computer Networking Practices as per the industry needs.

### **REFERENCES**

1. Andrew S. Tanenbaum, David J. Wethrall, Computer Network, Pearson Education, ISBN: 978-0- 13-212695-3
2. Behrouz A. Forouzan, TCP/IP Protocol Suite, McGraw Hill Education, ISBN: 978-0-07-070652-1, 4th Edition

### **Experiment No.: 1 Date:**

**Title: To study types of cables** (Twisted Pair Cable, Co-axial cable, fiber optic cable), connectors (RJ-45,

BNC connector) and Networking devices(Hub, Switch, Router, Gateway).

**AIM: To study types of cables** (Twisted Pair Cable, Co-axial cable, fiber optic cable), connectors (RJ-45, BNC connector) and Networking devices(Hub, Switch, Router, Gateway).

### **OBJECTIVES:**

Study of (cables, connectors, topologies, switches/ hubs, crimping tool, IP addressing scheme, Subnetting,

### **THEORY:**

#### **LAN**

A local area network (LAN) is a computer network that interconnects computers within a limited area such as a home, school, computer laboratory, or office building using network media to communicate with one another and share resources such as printers.

#### **I] Cables**

Cable is the medium through which information usually moves from one network device to another. The type of cable chosen for a network is related to the network's topology, protocol, and size. There are several types of cable which are commonly used with LANs.

- Twisted Pair Cable
- Coaxial Cable
- Fiber Optic Cable

#### **1) Twisted Pair Cable**

In its simplest form, twisted-pair cable consists of two insulated strands of copper wire twisted around each other.

##### **Unshielded Twisted Pair Cable**

UTP, using the 10BaseT specification, is the most popular type of twisted-pair cable and is fast becoming the most popular LAN cabling. The maximum cable length segment is 100 meters, about 328 feet. Traditional UTP cable, as shown in Figure 1, consists of two insulated copper wires.

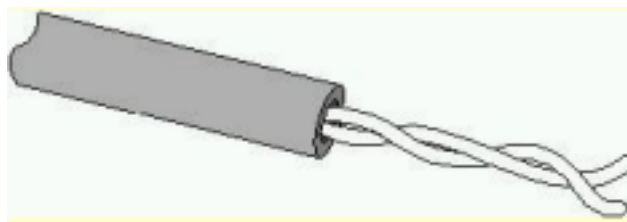


Fig.1 Unshielded Twisted Pair Cable

##### **Shielded Twisted Pair Cable**

STP cable uses a woven copper-braid jacket that is more protective and of a higher quality than the jacket used by UTP. Figure 2 shows a two-twisted-pair STP cable. STP also uses a foil wrap around each of the wire pairs. This gives STP excellent shielding to protect the transmitted data from outside interference, which in turn allows it to support higher transmission rates over longer distances than UTP.

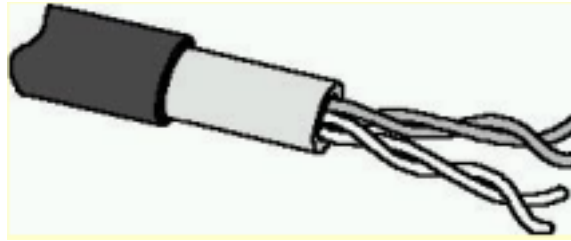


Fig.2 Shielded Twisted Pair Cable

#### Twisted Pair Cable Connector (RJ45 Connector)

Twisted-pair cabling uses RJ-45 telephone connectors to connect to a computer. The RJ-45 connector houses eight cable connections. An RJ-45 connector is shown in Figure 3.

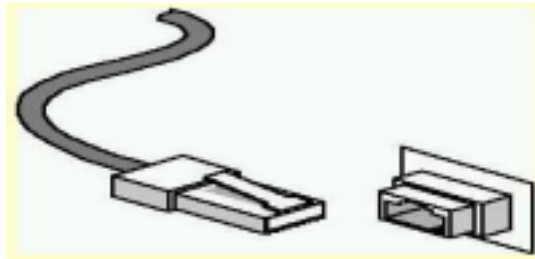


Fig.3 RJ45 Connector

## **2) Coaxial cable**

In its simplest form, coaxial cable consists of a core of copper wire surrounded by insulation, a braided metal shielding, and an outer cover. Figure 2.1 shows the various components that make up a coaxial cable.

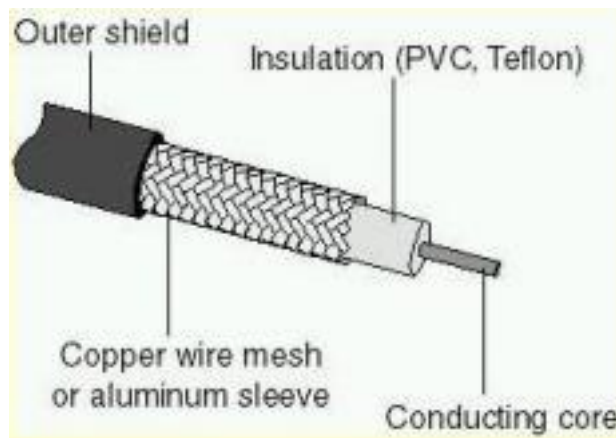


Fig.4 Coaxial Cable

#### Coaxial Cable Connector (BNC Connector)

Coaxial cable uses a connection component, known as a BNC connector, to make the connections between the cable and the computers. Figure 2.7 shows a BNC connector.

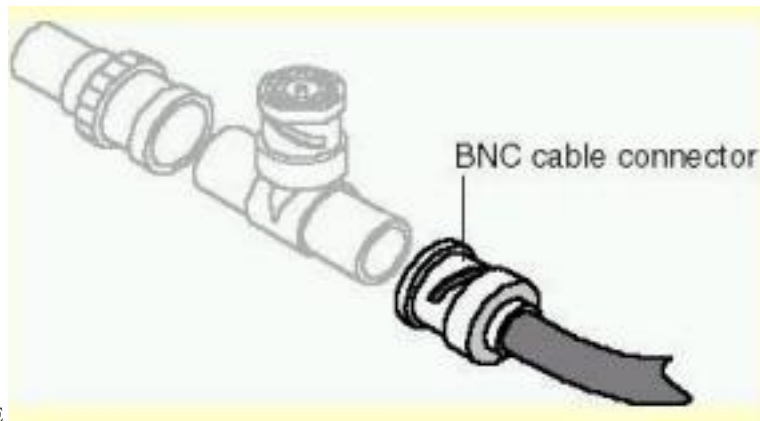


Fig.5 BNC Connector

### 3) Fiber optic Cable

In fiber-optic cable, optical fibers carry digital data signals in the form of modulated pulses of light. An optical fiber consists of an extremely thin cylinder of glass, called the core, surrounded by a concentric layer of glass, known as the cladding.

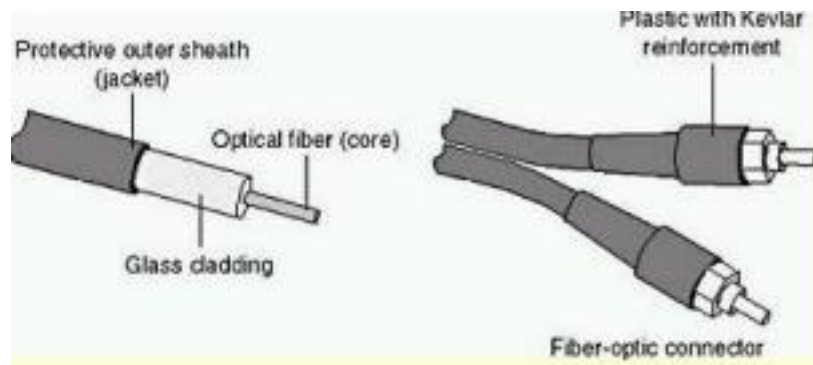
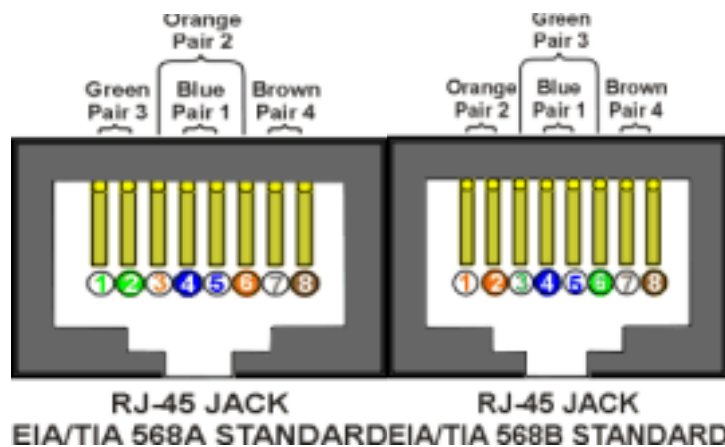


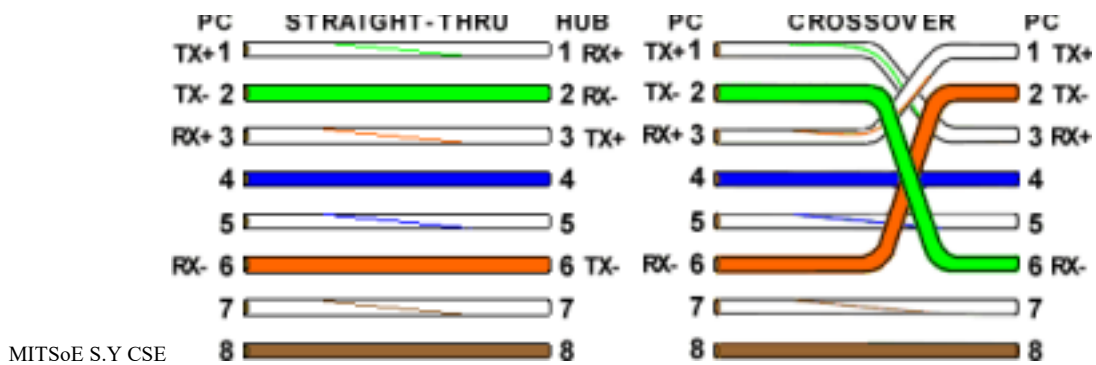
Fig.6 Fiber optic Cable

### Color-Code Standards:

Two wires color-code standards apply: EIA/TIA 568A and EIA/TIA 568B. The codes are commonly depicted with RJ-45 jacks as follows:



If we apply the 568A color code and show all eight wires, our pin-out looks like this:



Note that pins 4, 5, 7, and 8 and the blue and brown pairs are not used in either standard.

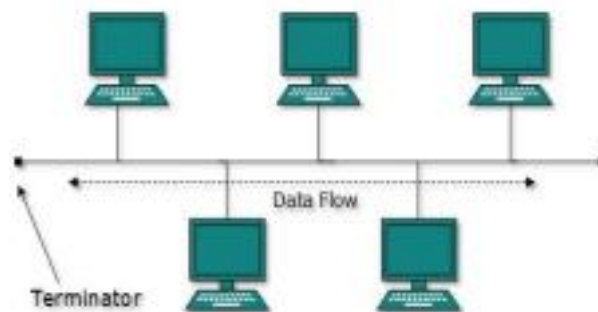
## II] Network Topologies

**Network topology** is the arrangement of the various elements (links, nodes, etc.) of a computer network. Essentially, it is the topological structure of a network, and may be depicted physically or logically. Physical topology refers to the placement of the network's various components, including device location and cable installation, while logical topology shows how data flows within a network, regardless of its physical design.

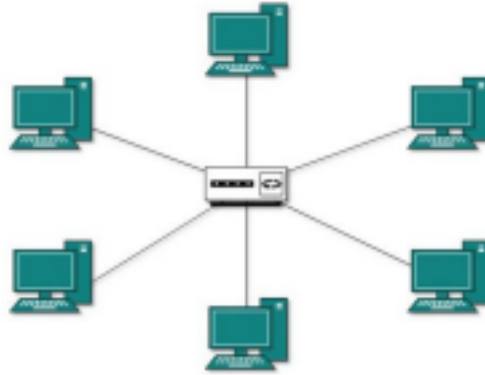
**Point-to-point:** Point-to-point networks contains exactly two hosts (computer or switches or routers or servers) connected back to back using a single piece of cable.



**Bus Topology:** In local area networks where bus topology is used, each node is connected to a single cable. Each computer or server is connected to the single bus cable.



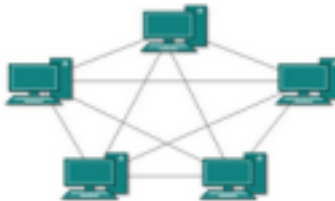
**Star Topology:** All hosts in star topology are connected to a central device, known as Hub device, using a point-to-point connection. That is, there exists a point to point connection between hosts and Hub.



**Ring Topology:** In ring topology, each host machine connects to exactly two other machines, creating a circular network structure. When one host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts.



**Mesh Topology:** In this type of topology, a host is connected to one or two or more than two hosts. This topology may have hosts having point-to-point connection to every other host or may also have hosts which are having point to point connection to few hosts only.



**Tree Topology:** This topology divides the network in to multiple levels/layers of network.





**Hybrid Topology:** A network structure whose design contains more than one topology is said to be Hybrid Topology.

### III] Hub

Hub is a common connection point for devices in a network. Hubs are commonly used to connect segments of a LAN. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets.

### IV] Switch

In networks, Switch is a device that filters and forwards packets between LAN segments. Switches operate at the data link layer (layer 2) and sometimes the network layer (layer 3) of the OSI Reference Model and therefore support any packet protocol. LANs that use switches to join segments are called switched LANs or, in the case of Ethernet networks, switched Ethernet LANs.

### V] IP address classes

In the early days of the Internet, the IANA (Internet Assigned Numbers Authority) defined five classes of public IP addresses as shown below.

<b>Class</b>	<b>Theoretical Address Range</b>	<b>Binary Start</b>	<b>Used for</b>
<b>A</b>	0.0.0.0 to 127.255.255.255	0	Very large networks
<b>B</b>	128.0.0.0 to 191.255.255.255	10	Medium networks
<b>C</b>	192.0.0.0 to 223.255.255.255	110	Small networks
<b>D</b>	224.0.0.0 to 239.255.255.255	1110	Multicast
<b>E</b>	240.0.0.0 to 255.255.255.255	1111	Experimental

### VI] Subnetting

Subnetting enables the network administrator to further divide the host part of the address into two or more subnets. In this case, a part of the host address is reserved to identify the particular subnet. Subnetting is a method for getting the most out of the limited 32-bit IPv4 addressing space and reducing the size of the routing tables in a large internetwork. With any address class, subnetting provides a means of

allocating a part of the host address space to network addresses, which lets you have more networks. The part of the host address space allocated to new network addresses is known as the subnet number. In addition to making more efficient use of the IPv4 address space, subnetting has several administrative benefits. Routing can become very complicated as the number of networks grows. A small organization, for example, might give each local network a class C number. As the organization grows, administering a number of different network numbers could become complicated. A better idea is to allocate a few class B network numbers to each major division in an organization. For instance, you could allocate one to Engineering, one to Operations, and so on. Then, you could divide each class B network into additional networks, using the additional network numbers gained by subnetting. This can also reduce the amount of routing information that must be communicated among routers.

### **CONCLUSIONS:**

We have studied networking different cables, devices and topologies.

## **Title: Basic TCP/IP utilities and commands**

**AIM:** To study basic TCP/IP utilities and commands. (eg: ping, ifconfig, tracer, arp, tcpdump, whois, host, netsat, nslookup, ftp, telnet etc...)

### **OBJECTIVES:**

To learn the basic TCP/IP utilities and commands that are commonly employed to help set up, configure and maintain TCP/IP internetworks. These utilities allows a network administrator to perform functions such as checking the identity of a host; verifying connectivity between two hosts; checking the path of routers between devices; examining the configuration of a computer; looking up a DNS domain name; and much more.

### **THEORY:**

#### **1. Ping**

PING Verifies connections to local or remote computers (Ping stands for Packet InterNet Groper, an excellent IP troubleshooting tool)

The PING utility tests connectivity between two hosts. PING uses a special protocol called the Internet Control Message Protocol (ICMP) to determine whether the remote machine (website, server, etc.) can receive the test packet and reply.

Also a great way to verify whether you have TCP/IP installed and your Network Card is working.

We'll start by Pinging the loopback address (127.0.0.1) to verify that TCP/IP is installed and configured correctly on the local computer.

#### **Ping 127.0.0.1**

This tells us that TCP/IP is working as well as Network Card.

To test out connectivity to a website all you have to do is **ping espn.com**

The results should tell us if the connection was successful or if we had any lost packets. Packet loss describes a condition in which data packets appear to be transmitted correctly at one end of a connection, but never arrive at the other. Why? Well, there are a few possibilities.

The network connection might be poor and packets get damaged in transit or the packet was dropped at a router because of internet congestion. Some Internet Web servers may be configured to disregard ping requests for

security purposes.

Note the IP address of espn.com — 199.181.132.250. We can also ping this address and get the same result.

However, Ping is not just used to test websites. It can also test connectivity to various servers: DNS, DHCP, your Print server, etc

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS][ -r count] [-s count] [[-j host-list] | [-k host-list]] [-w timeout] destination-list

-t Ping the specified host until stopped. To see statistics and continue - type Control-Break; To stop - type Control-C.

-a Resolve addresses to hostnames.

-n count Count Number of echo requests to send.

-l size Send buffer size.

-f Don't Fragment flag in packet.

-i ttl TTL Time To Live.

-v tos TOS Type Of Service.

-r count Record route for count hops.

-s count Timestamp for count hops.

-j host-list Loose source route along host-list.

-k host-list Strict source route along host-list

-w timeout Timeout in milliseconds to wait for each reply.

Examples:

1. I'm pinging 127.0.0.1 which is self. The 127.0.0.1 is called loopback. Thus when receiving replies I know my basic TCP/IP setup is working. The time provided is the roundtrip times and the "Time to Live" is the hop count for the packets being sent. The roundtrip time here is very short since all I'm doing is a wrap around to self

C:\>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<10ms TTL=128

Reply from 127.0.0.1: bytes=32 time<10ms TTL=128

Reply from 127.0.0.1: bytes=32 time<10ms TTL=128

Reply from 127.0.0.1: bytes=32 time<10ms TTL=128

Ping statistics for 127.0.0.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

Example 2:

Here I'm pinging web site using the IP address; normally, you would do this after having done one above. The first established that your basic setup is fine. This establishes that your internet connection is working fine

C:\>ping 207.159.129.102

Pinging 207.159.129.102 with 32 bytes of data:

```
Reply from 207.159.129.102: bytes=32 time=328ms TTL=250
Reply from 207.159.129.102: bytes=32 time=251ms TTL=250
Reply from 207.159.129.102: bytes=32 time=358ms TTL=250
Reply from 207.159.129.102: bytes=32 time=296ms TTL=250
```

Ping statistics for 207.159.129.102:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 251ms, Maximum = 358ms, Average = 308ms

It is not uncommon to get a few "request Timed out" responses.

### Example 3

Here I'm pinging the same place as above. If the two first worked and this does not, the most common problem is the DNS setup. It may also be caused by duplicates of some of the winsock files in win95.

```
C:\>ping www.Winfiles.com
```

Pinging www3-pool.Winfiles.com [207.159.129.102] with 32 bytes of data:

```
Reply from 207.159.129.102: bytes=32 time=202ms TTL=250
Reply from 207.159.129.102: bytes=32 time=245ms TTL=250
Reply from 207.159.129.102: bytes=32 time=330ms TTL=250
Reply from 207.159.129.102: bytes=32 time=217ms TTL=250
```

Ping statistics for 207.159.129.102:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 202ms, Maximum = 330ms, Average = 248ms

## 2. Tracert

Tracert is very similar to Ping, except that Tracert identifies pathways taken along each hop, rather than the time it takes for each packet to return (ping). It is a nice little utility which can be used quite effectively for diagnosis of networks and routes. It can also be used to find IP addresses for items you only know by name. If I have trouble connecting to a remote host I will use Tracert to see where that connection fails. Any information sent from a source computer must travel through many computers / servers / routers (they're all the same thing, essentially) before it reaches a destination. It may not be your computer but something that is down along the

way. It can also tell you if communication is slow because a link has gone down between you and the destination.

If you know there are normally 4 routers but Tracert returns 8 responses, you know your packets are taking an indirect route due to a link being down.

Tracert is a TCP/IP utility which determines the route taken. It does this by sending out packets with varying TTL (time to live). Each way station along the route is supposed to decrease the TTL value by 1 before passing it on. When the count reaches Zero, the router will return respond to the sender that the time was exceeded. Thus, the first packet is sent with a TTL (hop count) of 1 and then incremented until the destination is reached.

Some routers just drops packets with a Zero count and thus becomes invisible to

tracert. tracert [-d] [-h maximum\_hops] [-j host-list] [-w timeout] target\_name

-d	Do not to resolve addresses to host names specifies the max TTL (hop count) to use to find target specifies a route along the host list - loose Waits for the timeout milliseconds for repsonse specifies the name of the destination
-h maximum_hops	
-j host-list	
-w timeout	
target_name	

C:\>tracert www.hildrum.com

Tracing route to hildrum.com [207.159.136.230]  
over a maximum of 30 hops:

1 1007 ms 839 ms 1477 ms max44.seattle.wa.ms.uu.net [207.76.5.50]

```

2 1148 ms 745 ms 155 ms ar1.seattle.wa.ms.uu.net [207.76.5.3]
3 168 ms 193 ms 159 ms Fddi0-0.CR2.SEA1.Alter.Net [137.39.33.42]
4 * 194 ms 151 ms Fddi1-0.GW2.SEA1.Alter.Net [137.39.42.194]
5 203 ms 162 ms 184 ms lightrealm-gw.customer.ALTER.NET [157.130.176.50]
6 220 ms 216 ms 183 ms hildrum.com [207.159.136.230]

```

Trace complete.

### 3. ARP

The ARP utility helps diagnose problems associated with the Address Resolution Protocol (ARP). TCP/IP hosts use ARP to determine the physical (MAC) address that corresponds with a specific IP address. Type arp with the – a option to display IP addresses that have been resolved to MAC addresses recently.

ARP stands for Address Resolution Protocol. This provides IP to Ethernet addresses. Each hardware card has an address coded in. This allows deletion and addition to the ARP cache. The switches to be used can be obtained by just typing arp at a DOS command prompt.

ARP -s inet\_addr eth\_addr [if\_addr]

ARP -d inet\_addr [if\_addr]

ARP -a [inet\_addr] [-N if\_addr]

-a	Displays current ARP entries by interrogating the current protocol data. If inet_addr is specified, the IP and Physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.
-g	same as -a
-N	if_addr Displays the ARP entries for the network interface specified by if_addr. Deletes the host specified by inet_addr.
-d	Adds the host and associates the Internet address inet_addr with the Physical address eth_addr. The Physical address is given as 6 hexadecimal bytes separated by hyphens. The entry is permanent.
-s	



eth\_addr Specifies a physical address.

if\_addr If present, this specifies the Internet address of the interface whose address translation table should be modified. If not present, the first applicable interface will be used.

C:\>arp -g

Interface: 153.34.131.179 on Interface 3

Internet Address	Physical Address	Type
16.1.0.4	20-53-52-43-00-00	dynamic
128.173.14.71	20-53-52-43-00-00	dynamic
129.132.98.11	20-53-52-43-00-00	dynamic
192.31.216.8	20-53-52-43-00-00	dynamic
204.118.34.6	20-53-52-43-00-00	dynamic
204.118.34.22	20-53-52-43-00-00	dynamic
204.123.2.72	20-53-52-43-00-00	dynamic
204.255.246.18	20-53-52-43-00-00	dynamic
208.215.43.40	20-53-52-43-00-00	dynamic

#### 4. Netstat

Netstat (Network Statistics) displays network connections (both incoming and outgoing), routing tables, and a number of network interface statistics. It's a helpful tool in finding problems and determining the amount of traffic on the network as a performance measurement.

This utility provides the connection both the local and remote, ports and the state of the connection. It has several switches which may be found by typing netstat /?

It provides the IP addresses and the ports of the remote computer(S) to which the socket is connected. If a port has not been established it is indicated by a \*. It shows the port numbers as well as IP address for the local computer.

It provides the type of protocol being used for the connection(s). It provides a status of the connection. Is it

established?? Is it closed?? Or is it waiting?? And more

NETSTAT [-a] [-e] [-n] [-s] [-p proto] [-r] [interval]

-a Displays all connections and listening ports.

-e Displays Ethernet statistics. This may be combined with the -s option.

-n Displays addresses and port numbers in numerical form.

-p proto Shows connections for the protocol specified by proto; proto may be TCP or UDP. If used with the -s option to display per-protocol statistics, proto may be TCP, UDP, or IP.

-r Displays the routing table.

-s Displays per-protocol statistics. By default, statistics are shown for TCP, UDP and IP; the -p option may be used to specify a subset of the default.

interval Redisplays selected statistics, pausing interval seconds between each display. Press CTRL+C to stop redisplaying statistics. If omitted, netstat will print the current configuration information once.

C:\>netstat

#### Active Connections

Proto	Local Address	Foreign Address	State
TCP	dummy:3174	www.microsoft.com:80	ESTABLISHED
TCP	dummy:3175	www.microsoft.com:80	ESTABLISHED
TCP	dummy:3176	www.microsoft.com:80	ESTABLISHED
TCP	dummy:3177	www.microsoft.com:80	ESTABLISHED
TCP	dummy:3178	208.8.204.14:telnet	ESTABLISHED
TCP	dummy:3181	chat.msn.com:6667	ESTABLISHED
TCP	dummy:3182	hildrum.com:ftp	ESTABLISHED
TCP	dummy:3183	hildrum.com:ftp-data	ESTABLISHED

## 5. Nbtstat

Nbtstat (NetBios over TCP/IP) enables you to check information about NetBios names. It helps us view the NetBios name cache (nbtstat -c) which shows the NetBios names and the corresponding IP address that has been resolved (nbtstat -r) by a particular host as well as the names that have been registered by the local system (nbtstat -n).

**6.**

## **NSLookup**

NSLookup provides a command-line utility for diagnosing DNS problems. In its most basic usage, NSLookup returns the IP address with the matching host name.

## **7. IPConfig**

Not part of the TCP/IP utilities but it is useful to show current TCP/IP settings.

The IPConfig command line utility will show detailed information about the network you are connected to. It also helps with reconfiguration of your IP address through release and renew.

Let's say you want to know what you're IP address is — **ipconfig** is what you type in the command

prompt.

**ipconfig** will give a quick view of you IP address, your subnet mask and default gateway. **ipconfig /all** will give you more detailed information.

Through **ipconfig /all** we can find DNS servers, if we have DHCP enabled, MAC Address, along with other helpful information. All good things to know if we have trouble getting connected to the internet.

#### SYNTAX

**ipconfig** [/? | /all | /renew *adapter* | /release *adapter* | /flushdns | /displaydns | /registerdns | /showclassid *adapter* | /setclassid *adapter classid*

The adapter connection name can use wildcard characters (\* and ?).

#### OPTIONS

`/?` Displays this help message

`/all` Displays full configuration information

`/release` Releases the IP address for the specified adapter    `/renew`

Renews the IP address for the specified adapter

`/flushdns` Purges the DNS Resolver cache

`/registerdns` Refreshes all DHCP leases and reregisters DNS names

`/displaydns` Displays the contents of the DNS Resolver Cache

19

MITSoE S.Y CSE

`/showclassid` Displays all the DHCP ClassIds allowed for the specified adapter

`/setclassid` Modifies the DHCP ClassId

The default (with no parameters specified) is to display only the IP address, subnet mask, and default gateway for each adapter that is bound to TCP/IP.

For **/all**, Ipconfig displays all of the current TCP/IP configuration values, including the IP address, subnet mask, default gateway, and Windows Internet Naming Service (WINS) and DNS configuration.

For **/release** and **/renew**, if no adapter name is specified, the IP address leases for all adapters that are bound to TCP/IP are released or renewed.

For **/setclassid**, if no ClassId is specified, the ClassId is removed.

## EXAMPLES

```
ipconfig Show information
```

```
ipconfig /all Show detailed information
```

```
ipconfig /renew Renew all adapters
```

```
ipconfig /renew EL* Renew any connection whose name starts EL
```

```
ipconfig /release *Con* Release all matching connections, for example, "Local Area Connection 1" or "Local Area Connection2"
```

## 8. tcpdump

tcpdump command will work on most flavors of unix operating system. tcpdump allows us to save the packets that are captured, so that we can use it for future analysis. The saved file can be viewed by the same tcpdump command. We can also use open source software like wireshark to read the tcpdump pcap files.

Options:

- **-i any** : Listen on all interfaces just to see if you're seeing any traffic.
- **-n** : Don't resolve hostnames.
- **-nn** : Don't resolve hostnames *or* port names.
- **-x** : Show the packet's *contents* in both hex and ASCII.
- **-xx** : Same as **-x**, but also shows the ethernet header.
- **-v**, **-vv**, **-vvv** : Increase the amount of packet information you get back.
- **-c** : Only get *x* number of packets and then stop.
- **-s** : Define the *snaplength* (size) of the capture in bytes. Use **-s0** to get everything, unless you are intentionally capturing less.
- **-S** : Print absolute sequence numbers.

20

MITSoE S.Y CSE

- **-e** : Get the ethernet header as well.
- **-q** : Show less protocol information.
- **-E** : Decrypt IPSEC traffic by providing an encryption key.

The default snaplength as of tcpdump 4.0 has changed from 68 bytes to 96 bytes. While this will give you more of a packet to see, it still won't get everything. Use **-s 1514** to get full coverage

Examples:

- **host** // look for traffic based on IP address (also works with hostname if you're not using **-n**) #  
tcpdump host 1.2.3.4
- **src, dst** // find traffic from only a source or destination (eliminates one side of a **host** conversation) #  
tcpdump src 2.3.4.5  
# tcpdump dst 3.4.5.6
- **net** // capture an entire network using CIDR notation

```
# tcpdump net 1.2.3.0/24
```

- **proto** // works for tcp, udp, and icmp. Note that you don't have to type proto  
# tcpdump icmp
- **port** // see only traffic to or from a certain port  
# tcpdump port 3389
- **src, dst port** // filter based on the source or destination port  
# tcpdump src port 1025  
# tcpdump dst port 389
- **src/dst, port, protocol** // combine all three  
# tcpdump src port 1025 and tcp  
# tcpdump udp and src port 53

## 9. whois Command

To make it easier for administrators to find information about domains in this large distributed database, modern TCP/IP implementations generally come with an intelligent version of the whois utility. It is able to accept as input the name of a domain and automatically locate the appropriate registry in which that domain's information is located.

The utility is usually used as follows:

```
whois [-h <whois-host>] <domain>
```

In the above syntax, the term `—domain` represents the name about which registration information is requested. The administrator can use the `—h` parameter to force the program to query a particular whois server, but again, this is usually not required.

## 10. FTP Command

To connect to another computer using FTP at the MS-DOS prompt, command line, or Linux shell type **FTP** and press enter. Once in FTP> Type:

```
open ftp.example.com
```

In the above example, you'd substitute example.com for the name of your domain you want to connect to. In addition to the domain name the IP address of the computer you're trying to connect to can also be typed in, for example, open 192.168.1.12.

Once connected you will be asked for a username and password. If these are entered properly you'll be successfully connected to the server where you can browse the files, send files, or receive files depending on your rights. Some servers may also allow anonymous logins you can connect to these computers using guest or e-mail address.

### Send and receive a file in FTP

To get files from the server onto your own computer use the get command as shown in the below example. In this example you'd get the file myfile.htm.

```
get myfile.htm
```

To send a file from your computer to the computer you are connected to assuming you have the rights use the send command as shown in the below example. In this example we're sending the myfile.htm to the directory we're currently in.

send myfile.htm

It is important to realize that the files being sent must be in your local working directory. In other words the directory you were in when you typed the FTP command. If you want to change to the directory that contains your files use the lcd command. For example, on Windows you'd type lcd c:\windows to set the local directory to the Windows directory.

## FTP Commands

### Command Information

Using this command you will have the capability of toggling back and forth between the operating system and ftp. Once back in the operating system, typing exit will take you back to the FTP

!

command line.

? Access the Help screen. **append** Append

text to a local file. **ascii** Switch to ASCII

transfer mode **bell** Turns bell mode on or off.

**binary** Switches to binary transfer mode. **bye**

Exits from FTP.

**cd** Changes directory.

**close** Exits from FTP.

**delete** Deletes a file.

**debug** Sets debugging on or off.

22

MITSoE S.Y CSE

Lists files if connected.

dir -C = Will list the files in wide format.

dir -l = Lists the files in bare format in alphabetic

order **dir**

dir -r = Lists directory in reverse alphabetic order.

dir -R = Lists all files in current directory and sub

directories. dir -S = Lists files in bare format in alphabetic order.

**disconnect** Exits from FTP.

**get** Get file from the computer connected to.

**glob** Sets globbing on or off. When turned off the file name in the put and get commands is taken literally and wildcards will not be looked at.

**hash** Sets hash mark printing on or off. When turned on for each 1024 bytes of data received a

hash mark (#) is displayed. **help** Access the Help screen and displays information about command if

command typed after help. **lcd** Displays local directory if typed alone or if path typed after lcd will change

local directory. **literal** Sends a literal command to the connected computer with an expected one line

response. **ls** Lists files of the remotely connected computer.

**mdelete** Multiple delete.

**mdir** Lists contents of multiple remote directories.

**mget** Get multiple files.  
**mkdir** Make directory.  
**mls** Lists contents of multiple remote directories.  
**mput** Sent multiple files  
**open** Opens address.  
**prompt** Enables or disables the prompt.  
**put** Send one file  
**pwd** Print working directory  
**quit** Exits from FTP.  
**quote** Same as the literal command.  
**recv** Receive file.  
**remotehelp** Get help from remote server.  
**rename** Renames a file.  
**rmdir** Removes a directory on the remote computer.  
**send** Send single file.  
**status** Shows status of currently enabled and disabled options  
**trace** Toggles packet tracing.  
**Type** Set file transfer type.  
**user** Send new user information.  
**verbose** Sets verbose on or off.

### **CONCLUSIONS:**

We have studied different TCP/IP utilities.

### **Experiment No.: 4. Date:**

**Title: Implement Program to determine IP Address class, Network ID & Host ID of an IPv4 Address.**

**AIM:** Write a program to identify IP Address class, Network ID & Host ID of an IPv4 Address

### **OBJECTIVES:**

To learn various IP address classes and to understand Network ID & Host ID

### **THEORY:**

In the early days of the Internet, the IANA (Internet Assigned Numbers Authority) defined five classes of public IP addresses as shown below.

Class	Theoretical Address Range	Binary Start	Used for
A	0.0.0.0 to 127.255.255.255	0	Very large networks
B			Medium networks
C	128.0.0.0 to 191.255.255.255	10	Small networks
D	192.0.0.0 to 223.255.255.255	110	Multicast



E	224.0.0.0 to 239.255.255.255	1110	Experimental
	240.0.0.0 to 255.255.255.255	1111	

### Source Code: C

```
// C program to determine class, Network
// and Host ID of an IPv4 address
#include<stdio.h>
#include<string.h>

// Function to find out the Class
char findClass(char str[])
{
    // storing first octet in arr[] variable
    char arr[4];
    int i = 0;
    while (str[i] != '.')
    {
        arr[i] = str[i];
        i++;
    }
    i--;

    // converting str[] variable into number for
    // comparison
    int ip = 0, j = 1;
    while (i >= 0)
```

24

MITSoE S.Y CSE

```
{
    ip = ip + (str[i] - '0') * j;
    j = j * 10;
    i--;
}
```

```
// Class A
if (ip >= 1 && ip <= 126)
    return 'A';
```

```
// Class B
else if (ip >= 128 && ip <= 191)
    return 'B';
```

```
// Class C
else if (ip >= 192 && ip <= 223)
    return 'C';
```

```
// Class D
else if (ip >= 224 && ip <= 239)
    return 'D';
```

```

// Class E
else
    return 'E';
}

// Function to separate Network ID as well
as // Host ID and print them
void separate(char str[], char ipClass)
{
    // Initializing network and host array to
    NULL char network[12], host[12];
    for (int k = 0; k < 12; k++)
        network[k] = host[k] = '\0';

    // for class A, only first octet is Network
    ID // and rest are Host ID
    if (ipClass == 'A')
    {
        int i = 0, j = 0;
        while (str[j] != '.')
            network[i++] = str[j++];
        i = 0;
        j++;
        while (str[j] != '\0')
            host[i++] = str[j++];
        printf("Network ID is %s\n", network);
        printf("Host ID is %s\n", host);
    }
}

```

25

MITSoE S.Y CSE

```

}

// for class B, first two octet are Network
ID // and rest are Host ID
else if (ipClass == 'B')
{
    int i = 0, j = 0, dotCount = 0;

    // storing in network[] up to 2nd dot
    // dotCount keeps track of number of
    // dots or octets passed
    while (dotCount < 2)
    {
        network[i++] = str[j++];
        if (str[j] == '.')
            dotCount++;
    }
    i = 0;
    j++;

    while (str[j] != '\0')
        host[i++] = str[j++];

    printf("Network ID is %s\n",
network); printf("Host ID is %s\n",

```

```

        host);
    }

// for class C, first three octet are Network
ID // and rest are Host ID
else if (ipClass == 'C')
{
    int i = 0, j = 0, dotCount = 0;

    // storing in network[] up to 3rd dot
    // dotCount keeps track of number of
    // dots or octets passed
    while (dotCount < 3)
    {
        network[i++] = str[j++];
        if (str[j] == '.')
            dotCount++;
    }

    i = 0;
    j++;

    while (str[j] != '\0')
        host[i++] = str[j++];
}

```

26

MITSoE S.Y CSE

```

        printf("Network ID is %s\n", network);
        printf("Host ID is %s\n", host);
    }

// Class D and E are not divided in Network
// and Host ID
else
    printf("In this Class, IP address is not"
           " divided into Network and Host ID\n");
}

// Driver function is to test above function
int main()
{
    char str[] = "192.226.12.11";
    char ipClass = findClass(str);
    printf("Given IP address belongs to Class %c\n", ipClass);
    separate(str, ipClass);
    return 0;
}

```

### Output:

Given IP address belongs to Class C

Network ID is 192.226.12

Host ID is 11

**CONCLUSION:**

We have studied various IP address classes.

27

MITSOE S.Y CSE **Experiment No.: 5. Date:**

**Title: To implement the entire Topologies configuration in Cisco packet tracer. (Star, Mesh, Bus, Ring, Hybrid Topology.)**

**AIM:** To implement the entire Topologies configuration in Cisco packet tracer. (Star, Mesh, Bus, Ring, Hybrid Topology.)

**OBJECTIVES:**

To learn various networking topologies and to implement them in cisco packet tracer

**SOFTWARES USED:**

Cisco Packet Tracer

**THEORY:**

**Network topology** is the arrangement of the various elements (links, nodes, etc.) of a computer network. Essentially, it is the topological structure of a network, and may be depicted physically or logically. Physical topology refers to the placement of the network's various components, including device location and cable installation, while logical topology shows how data flows within a network, regardless of its physical design.

**Point-to-point:** Point-to-point networks contains exactly two hosts (computer or switches or routers or servers) connected back to back using a single piece of cable.

**Bus Topology:** In local area networks where bus topology is used, each node is connected to a single cable. Each computer or server is connected to the single bus cable.

**Star Topology:** All hosts in star topology are connected to a central device, known as Hub device, using a point-to-point connection. That is, there exists a point to point connection between hosts and Hub.

**Ring Topology:** In ring topology, each host machine connects to exactly two other machines, creating a circular network structure. When one host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts.

**Mesh Topology:** In this type of topology, a host is connected to one or two or more than two hosts. This topology may have hosts having point-to-point connection to every other host or may also have hosts which are having point to point connection to few hosts only.

**Tree Topology:** This topology divides the network in to multiple levels/layers of network.

**Hybrid Topology:** A network structure whose design contains more than one topology is said to be Hybrid Topology.

**CONCLUSIONS:**

We have successfully configured all network topologies in cisco packet tracer

**Experiment No.: 6. Date: Title: Implement simple Static Routing & Dynamic routing in Cisco**

**Packet Tracer. AIM:** Using a Network Simulator (e.g. packet tracer) configure Static Routing and  
Dynamic Routing

**OBJECTIVES:**

To learn various routing protocols and get familiar with network simulator like Cisco Packet Tracer.

**SOFTWARES USED:**

Cisco Packet Tracer

**THEORY:****Router**

A Router is a layer 3 network device that moves data between different network segments and can look into a packet header to determine the best path for the packet to travel. Routers can connect network segments that use different protocols. They also allow all users in a network to share a single connection to the Internet or a WAN. It is used to improve network performance by:-

- segmenting the network and creating separate collision & broadcast domains.
- reducing competition for bandwidth.
- Broadcasts are not forwarded to other network segments.
- Increases security by using Access Lists.

**Router Components (Internal)****ROM**

ROM is used to store the router's bootstrap startup program, operating system software, and power-on diagnostic tests programs. In order to perform ROM upgrades you remove and replace pluggable chips on the motherboard

**Flash Memory**

It holds operating system image(s). Flash memory is erasable, reprogrammable ROM. You can perform Cisco® IOS software upgrades without having to remove and replace chips. Flash content is retained when you switch off or restart the router.

**RAM**

RAM is used to store operational information such as routing tables, router's running configuration file. RAM also provides caching and packet buffering capabilities. Its contents are lost when you switch off or restart the router.

**NVRAM**

NVRAM (nonvolatile RAM), is used to store the router's startup configuration file. It does not lose data when power is switched off. So the contents of the startup configuration file are maintained even when you switch off or restart the router.

**Network Interfaces**



The router's network interfaces are located on the motherboard or on separate interface modules. You configure Ethernet or Token Ring interfaces to allow connection to a LAN. The synchronous serial interfaces are configured to allow connection to WANs. You can also configure ISDN BRI interfaces to allow connection to an ISDN WAN.

<b>BASIC MODE CHANGING COMMANDS</b>	
router> enable	Move from User to Privilege mode. Prompt changes from Routername> to routername#
router# configure terminal	Changes the routers interface from Privileged mode to Global Configuration mode. Prompt becomes Routername(config)#
router(config)#CTRL Z	Will exit Global configuration mode and return to Privileged mode.
router(config)#exit	Will exit the level of configuration and drop you down one level or back to privileged mode.
router# copy running-config startup config	Copies the Running-config (ram) to the Startup-config (nvram). The configuration in NVRAM will be saved when the router is powered off
router(config)#no	No followed by any command will negate or reverse the command. To unset or set the opposite behavior of a command.
router(config)#hostname Lab-B	Name the router Lab-B Name is case sensitive
Lab-B(config)#enable secret class	Sets the encrypted version of the routers password to —class  Secret password overrides standard password.
	#enable password cisco Sets standard clear text password for router
<b>INTERFACE CONFIGURATION –FAST ETHERNET PORT</b>	
Lab-B(config)#interface fastethernet 0/0	Interface FastEthernet 0/0 Changes the configuration mode from Global to Interface for the FastEthernet (100 Mps)
Lab-B(config-if)#ip address 219.17.100.1 255.255.255.0	Assigns the IP address 219.17.100.1 to the interface. Subnet mask for Class C address.
Lab-B(config-if)#description Connected to LAN B	Provides a description to an interface.

Lab-B(config-if)#no shutdown Shutdown is the actual command – no shutdown is the most popular use of the command	Enables the interface. By default all interface are shutdown. You must use —no shutdownl to remove the shutdown command
<b>INTERFACE CONFIGURATION –SERIAL PORT</b>	
Lab-B(config)#interface serial 0/0/0	Interface Serial 0/0/0 Changes the configuration mode from Global to Interface for the Serial port.

32

MITSoE S.Y CSE

Lab-B(config-if)#ip address 199.6.13.1 255.255.255.0	Assigns the IP address 199.6.13.1 to the interface. Subnet mask for Class C address.
Lab-B(config-if)#clock rate 56000	For Serial interfaces the DCE side of the interface cable must have the clock rate set. This controls the speed of the serial connection
Lab-B(config-if)#no shutdown	Enables the interface. By default all interface are shutdown. You must use —no shutdownl to remove the shutdown command

## I. Static Routing

Static routing occurs when you manually add routes in each router's routing table. There are advantages and disadvantages to static routing, but that's true for all routing processes.

### Static routing has the following advantages:

- There is no overhead on the router CPU.
- There is no bandwidth usage between routers.
- It adds security because the administrator can choose to allow routing access to certain networks only.

### Static routing has the following disadvantages:

- The administrator must really understand the internetwork and how each router is connected in order to configure routes correctly.
- If a network is added to the internetwork, the administrator has to add a route to it on all routers—manually.
- It's not possible in large networks because maintaining it would be a full-time job in itself.

Command syntax for static route:

```
ip route [destination_network] [mask] [next-hop_address or exit_interface] [administrative_distance]
[permanent]
```

**ip route** : The command used to create the static route.

**destination\_network** : The network you're placing in the routing table.

**mask** : The subnet mask being used on the network.

**next-hop\_address** : The address of the next-hop router that will receive the packet and forward it to the remote network.

**exit\_interface** : Used in place of the next-hop address if you want, and shows up as a directly connected route.

**administrative\_distance** : By default, static routes have an administrative distance of 1 (or even 0 if you use an exit interface instead of a next-hop address).

**permanent Keyword (Optional)** : Without the permanent keyword in a static route statement, a static route will be removed if an interface goes down. Adding the permanent keyword to a static route statement will keep the static routes in the routing table even if the interface goes down and the directly connected networks are removed.

## **CONFIGURATION:**

Configure IP address to routers go to global configuration mode in R1 and R2 configure connected interfaces. In Router 1

Interface Fastethernet0/0 in global configuration mode

```
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
```

Interface Serial 2/0

```
R1(config)#interface serial 2/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
```

```
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
```

In Router 2

Interface FastEthernet 0/0

34

MITSoE S.Y CSE

```
R2(config)#interface fastEthernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
R2(config-if)#exit
```

Interface Serial 2/0

```
R2(config)#interface serial 2/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
R2(config-if)#exit
```

Assign IP address for both PC's with appropriate ip and subnetmask and default gateway. Now configure both routers with static route. By default, Routers know only directly connected networks here Router 1 knows only 10.0.0.0 and 20.0.0.0 it doesn't know the 30.0.0.0 like this R2 doesn't know about 10.0.0.0. So we are going to add static route to this both routers

R1(config)#ip route Destination Network | Destination N/W SubnetMask | Next Hop Address

In Router R1, just give this command, In this case Destination is 30.0.0.0 and its subnet mask is 255.0.0.0 next hop address is 20.0.0.2

```
R1(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
```

In Router R2

```
R2(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
```

That's it! Now both routers know all networks, check by ping IP address of hosts.

## DYNAMIC ROUTING

### II. RIPv2 Routing Protocol

RIPv2 has one great improvement over RIPv1, instead of sending just the network address in its routing updates it can send the subnet mask and the next hop address as well, this means that it can advertise non-classful subnets. In this way, RIPv2 supports VLSM and CIDR. Mostly RIPv2 is very similar to RIPv1, it is a Distance Vector routing protocol, it uses hop count as its metric, with a maximum distance of 15 hops.

### RIPv2 Differences from RIPv1

- VLSM and CIDR - is supported by sending the subnet mask and the next hop address in its routing updates.
- Multicasts - its routing updates will not be broadcasted to 255.255.255.255 like RIPv1
- Authentication - RIPv2 supports md5 authentication

- Updates - RIPv2 sends and receives version 2 updates only. RIPv1 sends version 1 updates and receives both 1 and 2, however version 2 information is ignored.

### **RIPv2 Similarities to RIPv1**

- Auto Summarizes by default (You will need to turn this off if you have discontinuous networks)
- Distance Vector Protocol
- Hop Count is the metric with a maximum of 15 hops, 16 is infinity and is dropped.

### **CONFIGURATION:**

---

1841 Series Router0 (R1)

IP address

Connected With

2811 Series Router0 (R4) IP address

Pc0
-----

FastEthernet0/0
-----------------

FastEthernet0/0	Serial0/0/0
10.0.0.1	20.0.0.1

Connected With Pc1 R3 on Serial 0/0

<b>2621XM Series Router0 (R3)</b>	FastEthernet0/0 Serial0/0/0
IP address 30.0.0.2 40.0.0.1	
Connected With FastEthernet0/0 R4 on Serial 0/0/0	<b>2620XM Series Router1 (R2)</b>
	FastEthernet0/0 Serial0/0
IP address 30.0.0.1 20.0.0.2	
Connected With R3 on FastEthernet0/0 R1 on Serial 0/0/0	

36

MITSøE S.Y CSE **PC-PT PC0**

IP address

Connected With **PC-PT PC1**

10.0.0.2
----------

R1 on FastEthernet0/0
-----------------------

FastEthernet0	Default Gateway
---------------	-----------------

	FastEthernet0	Default Gateway
IP address	50.0.0.2	50.0.0.1
Connected With	R4 on FastEthernet0/0	

**To configure and enable rip routing on R1 follow these commands**

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router rip
R1(config-router)#network 10.0.0.0
R1(config-router)#network 20.0.0.0
R1(config-router)#exit
R1(config)#
```

**To configure and enable rip routing on R2 follow these command**

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
```

```
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config)#router rip
R2(config-router)#network 20.0.0.0
R2(config-router)#network 30.0.0.0
R2(config-router)#exit
R2(config)#
```

**To configure and enable rip routing on R3 follow these commands**

```

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up %LINEPROTO-5-
UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to
up
R3(config)#router rip
R3(config-router)#network 30.0.0.0
R3(config-router)#network 40.0.0.0
R3(config-router)#exit
R3(config)#

```

To configure and enable rip routing on R4 follow these commands

```

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to
up
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%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

```

38

MITS&E S.Y CSE

```

R4(config)#router rip
R4(config-router)#network 40.0.0.0
R4(config-router)#network 50.0.0.0
R4(config-router)#exit
R4(config)#

```

**You can verify that RIP is running successfully via `show ip protocols` command in privilege mode.**

```

R1#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 2 seconds
Invalid after 180 seconds, hold down 180, flushed after
240 Outgoing update filter list for all interfaces is not
set Incoming update filter list for all interfaces is not
set Redistributing: rip
Default version control: send version 1, receive any version
Interface Send Recv Triggered RIP Key-chain FastEthernet0/0
1 2 1

```



```
Serial0/0/0 1 2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
    10.0.0.0
    20.0.0.0
Passive Interface(s):
Routing Information Sources:
    Gateway Distance Last Update
    20.0.0.2 120 00:00:20
Distance: (default is 120)
R1#
```

### **CONCLUSIONS:**

We have successfully configured Router for static and dynamic routing.

**Experiment No.: 7. Date: Title: To Design & Configure VLAN by using Packet Tracer.**

**AIM: To Design & Configure VLAN**

### **OBJECTIVES:**

To implement VLAN & allow host to communicate within same VLAN

### **SOFTWARES USED:**

Cisco Packet Tracer

### **THEORY:**

#### **Virtual LAN:**

A Local Area Network (LAN) was originally defined as a network of computers located within the same area. Today, Local Area Networks are defined as a single broadcast domain. This means that if a user broadcasts information on his/her LAN, the broadcast will be received by every other user on the LAN. Broadcasts are prevented from leaving a LAN by using a router. The disadvantage of this method is routers usually take more time to process incoming data compared to a bridge or a switch. More importantly, the formation of broadcast

domains depends on the physical connection of the devices in the network. Virtual Local Area Networks (VLAN's) were developed as an alternative solution to using routers to contain broadcast traffic.

VLAN offers a number of advantages over traditional LAN's. They are

### **1) Performance**

In networks where traffic consists of a high percentage of broadcasts and multicasts, VLAN's can reduce the need to send such traffic to unnecessary destinations.

### **2) Formation of Virtual Workgroups**

Nowadays, it is common to find cross-functional product development teams with members from different departments such as marketing, sales, accounting, and research. These workgroups are usually formed for a short period of time. During this period, communication between members of the workgroup will be high. To contain broadcasts and multicasts within the workgroup, a VLAN can be set up for them.

### **3) Simplified Administration**

VLAN's will only really be felt when good management tools are created which can allow network managers to drag and drop users into different VLAN's or to set up aliases.

### **4) Reduced Cost**

VLAN's can be used to create broadcast domains which eliminate the need for expensive routers.

### **5) Security**

Periodically, sensitive data may be broadcast on a network. In such cases, placing only those users who can have access to that data on a VLAN can reduce the chances of an outsider gaining access to the data.

## **Types of VLAN's**

- 1) Layer 1 VLAN: Membership by Port
- 2) Layer 2 VLAN: Membership by MAC Address
- 3) Layer 2 VLAN: Membership by Protocol Type
- 4) Layer 3 VLAN: Membership by IP Subnet Address

## **Types of Connections**

### **1) Trunk Link**

All the devices connected to a trunk link, including workstations, must be VLAN-aware. All frames on a trunk link must have a special header attached. These special frames are called tagged frames

### **2) Access Link**

An access link connects a VLAN-unaware device to the port of a VLAN-aware bridge. All frames on access links must be implicitly tagged (untagged). The VLAN-unaware device can be a LAN segment with VLAN unaware workstations or it can be a number of LAN segments containing VLAN-unaware devices

## **CONFIGURATION:**

If we go to the switch and enter the command —show vlan 1. It shows the following.

As we can see in the figure above all the interfaces are being displayed and they are all the part of the default vlan 1. Now let us apply vlans on the switch. We are going to create three vlans as follows.

Now, that we have created the vlans. Lets see if they are visible to us.

In the above figure, vlans are visible. Now, we are going to assign interfaces to vlans. They are two ways to do this. i. We can select an interface and assign that interface to a specific vlan ii. We can select multiple interfaces (range of interfaces) at once and assign those interfaces to vlan. In the figure below, we have done both of these.

MITSoE S.Y CSE

Now, when we write `—show vlan —command`, we will realize that interfaces have been assigned to desired vlans respectively

Let us assign IP addresses to PCs. Open the PC.

MITSoE S.Y CSE

After assigning IP addresses, when we try to communicate between two PCs belonging to two different vlans, it will fail. Thus, we have achieved our purpose.

**CONCLUSIONS:**

We have successfully configured VLAN.

**AIM:** To configure dynamic routing protocols

**OBJECTIVES:**

To design a small network and configure OSPF & BGP protocols.

**SOFTWARES USED:**

Cisco Packet Tracer

**THEORY:**

**Open Shortest Path First (OSPF)**

Open Shortest Path First (OSPF) is a link-state routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS).

OSPF is an interior gateway protocol that routes Internet Protocol (IP) packets solely within a single routing domain (autonomous system). It gathers link state information from available routers and constructs a topology map of the network. The topology determines the routing table presented to the Internet Layer which makes routing decisions based solely on the destination IP address found in IP packets. OSPF was designed to support variable-length subnet masking (VLSM) or Classless Inter-Domain Routing (CIDR) addressing models.

OSPF detects changes in the topology, such as link failures, and converges on a new loop-free routing structure within seconds. It computes the shortest path tree for each route using a method based on Dijkstra's algorithm, a shortest path first algorithm.

**CONFIGURATION:**



**1841 Series Router0 (R1)**

FastEthernet0/0 Serial0/0/0

IP address 10.0.0.1 20.0.0.1

Connected With Pc0 R2 on Serial 0/0 **2811 Series Router0 (R4)**

FastEthernet0/0 Serial0/0/0

IP address 50.0.0.1 40.0.0.2

Connected With Pc1 R3 on Serial 0/0

**2621XM Series Router0 (R3)**

FastEthernet0/0 Serial0/0/0

IP address 30.0.0.2 40.0.0.1

Connected With FastEthernet0/0 R4 on Serial 0/0/0 **2620XM Series Router1 (R2)**

FastEthernet0/0 Serial0/0

IP address 30.0.0.1 20.0.0.2

Connected With R3 on FastEthernet0/0 R1 on Serial 0/0/0

**PC-PT PC0**

FastEthernet0

10.0.0.2

R1 on FastEthernet0/0

Default Gateway

IP address 10.0.0.1 Connected With

**PC-PT PC1**

FastEthernet0

50.0.0.2

R4 on FastEthernet0/0

Default Gateway

IP address 50.0.0.1 Connected With

Configuring OSPF is slightly different from configuring RIP. When configuring OSPF, use the following syntax:

```
Router(config)# router ospf process_ID
Router(config-router)# network IP_address wildcard_mask area area_#
```

**The process\_ID** is locally significant and is used to differentiate between OSPF processes running on the same router. Your router might be a boundary router between two OSPF autonomous systems, and to differentiate them on your router, you will give them unique process IDs. Note that these numbers do not need to match between different routers so they have nothing to do with autonomous system numbers.

To configure and enable ospf routing on R1 follow these commands

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up %LINEPROTO-5-
UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
R1(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
R1(config)#router ospf 1
R1(config-router)#network 10.0.0.0 0.255.255.255 area 0
R1(config-router)#network 20.0.0.0 0.255.255.255 area 0
R1(config-router)#exit
R1(config)#
```

To configure and enable ospf routing on R2 follow these commands

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R2
```

```

R2(config)#interface serial 0/0
R2(config-if)#ip address 20.0.0.2 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R2(config-if)#exit
R2(config)#interface fastethernet 0/0
R2(config-if)#ip address 30.0.0.1 255.0.0.0
R2(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config)#router ospf 2
R2(config-router)#network 20.0.0.0 0.255.255.255 area 0
R2(config-router)#network 3
00:03:10: %OSPF-5-ADJCHG: Process 2, Nbr 20.0.0.1 on Serial0/0
    from LOADING to FULL, Loading Done
R2(config-router)#network 30.0.0.0 0.255.255.255 area 0
R2(config-router)#exit
R2(config)#

```

**To configure and enable ospf routing on R3 follow these commands**

49

MITS&E S.Y CSE

```

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R3
R3(config)#interface fastethernet 0/0
R3(config-if)#ip address 30.0.0.2 255.0.0.0
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config-if)#interface serial 0/0
R3(config-if)#ip address 40.0.0.1 255.0.0.0
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial0/0, changed state to down
R3(config-if)#exit
%LINK-5-CHANGED: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R3(config)#router ospf 3
R3(config-router)#network 40.0.0.0 0.255.255.255 area 0
R3(config-router)#network 30.0.0.0 0.255.255.255 area 0
00:04:53: %OSPF-5-ADJCHG: Process 3, Nbr 30.0.0.1 on FastEthernet0/0
    from LOADING to FULL, Loading D
R3(config-router)#exit
R3(config)#
%SYS-5-CONFIG_I: Configured from console by console
R3#

```

**To configure and enable ospf routing on R4 follow these commands**

```

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shutdown

```

```
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to
up Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%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
R4(config)#router ospf 4
R4(config-router)#network 50.0.0.0 0.255.255.255 area 0
R4(config-router)#network 40.0.0.0 0.255.255.255 area 0
R4(config-router)#
00:06:32: %OSPF-5-ADJCHG: Process 4, Nbr 40.0.0.1 on Serial0/0/0
from LOADING to FULL, Loading Done
R4(config-router)#exit
R4(config)#
```

50

MITSoE S.Y CSE

You can verify that ospf is running successfully via **show ip protocols** command in privilege mode. R4#show ip protocols

```
Routing Protocol is "ospf 4"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 50.0.0.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    50.0.0.0 0.255.255.255 area 0
    40.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
    Gateway Distance Last Update
    40.0.0.1 110 00:01:26
  Distance: (default is 110)
```

R4#

## **Border Gateway Protocol (BGP)**

Routing involves two basic activities: determination of optimal routing paths and the transport of information groups (typically called packets) through an internetwork. The transport of packets through an internetwork is relatively straightforward. Path determination, on the other hand, can be very complex. One protocol that addresses the task of path determination in today's networks is the *Border Gateway Protocol* (BGP). This chapter summarizes the basic operations of BGP and provides a description of its protocol components.

BGP performs interdomain routing in Transmission-Control Protocol/Internet Protocol (TCP/IP) networks. BGP is an exterior gateway protocol (EGP), which means that it performs routing between multiple autonomous systems or domains and exchanges routing and reachability information with other BGP systems.

BGP was developed to replace its predecessor, the now obsolete *Exterior Gateway Protocol* (EGP), as the standard exterior gateway-routing protocol used in the global Internet. BGP solves serious problems with EGP and scales to Internet growth more efficiently. Note EGP is a particular instance of an exterior gateway protocol (also EGP)—the two should not be confused.

Core routers can use BGP to route traffic between autonomous systems. The figure given below illustrates core routers using BGP to route traffic between autonomous systems.

BGP is specified in several *Request for Comments* (RFCs):

- RFC 1771 – Describes BGP4, the current version of BGP
- RFC 1654 – describes the first BGP4 specification
- RFC 1105, RFC 1163, and RFC 1267 – Describes versions of BGP prior to BGP4

## **BGP Operation**

BGP performs three types of routing: *interautonomous system routing*, *intra-autonomous system routing*, and *pass-through autonomous system routing*.

*Interautonomous system routing* occurs between two or more BGP routers in different autonomous systems. Peer routers in these systems use BGP to maintain a consistent view of the internetwork topology. BGP neighbors communicating between autonomous systems must reside on the same physical network. The Internet serves as an example of an entity that uses this type of routing because it is comprised of autonomous systems or administrative domains. Many of these domains represent the various institutions, corporations, and entities that make up the Internet. BGP is frequently used to provide path determination to provide optimal routing within the Internet.

*Intra-autonomous system routing* occurs between two or more BGP routers located within the same autonomous system. Peer routers within the same autonomous system use BGP to maintain a consistent view of the system topology. BGP also is used to determine which router will serve as the connection point for specific external autonomous systems. Once again, the Internet provides an example of inter autonomous system routing. An organization, such as a university, could make use of BGP to provide optimal routing within its own administrative domain or autonomous system. The BGP protocol can provide both inter- and intra-autonomous system routing

services.

In pass-through autonomous system routing, BGP pairs with another intra-autonomous system-routing protocol.

*Pass-through autonomous system* routing occurs between two or more BGP peer routers that exchange traffic across an autonomous system that does not run BGP. In a pass-through autonomous system environment, the BGP traffic did not originate within the autonomous system in question and is not destined for a node in the autonomous system. BGP must interact with whatever intra-autonomous system routing protocol is being used to successfully transport BGP traffic through that autonomous system. Figure 2 illustrates a pass-through autonomous system environment.

### BGP Routing

As with any routing protocol, BGP maintains routing tables, transmits routing updates, and bases routing decisions on routing metrics. The primary function of a BGP system is to exchange network-reachability information, including information about the list of autonomous system paths, with other BGP systems. This information can be used to construct a graph of autonomous system connectivity from which routing loops can be pruned and with which autonomous system-level policy decisions can be enforced.

Each BGP router maintains a routing table that lists all feasible paths to a particular network. The router does not refresh the routing table, however. Instead, routing information received from peer routers is retained until an incremental update is received. BGP devices exchange routing information upon initial data exchange and after

incremental updates. When a router first connects to the network, BGP routers exchange their entire BGP routing tables. Similarly, when the routing table changes, routers send the portion of their routing table that has changed. BGP routers do not send regularly scheduled routing updates, and BGP routing updates advertise only the optimal path to a network.

BGP uses a single routing metric to determine the best path to a given network. This metric consists of an arbitrary unit number that specifies the degree of preference of a particular link. The BGP metric typically is

53

MITSoE S.Y CSE

assigned to each link by the network administrator. The value assigned to a link can be based on any number of criteria, including the number of autonomous systems through which the path passes, stability, speed, delay, or cost.

## **CONFIGURATION**

Only external BGP is supported at this moment. Packet Tracer does not support internal BGP in this version. Only external neighbors are supported.

```
Router0(config)#router bgp 1
Router0(config-router)#neighbor 20.0.0.2 remote-as 2
Router0(config-router)#network 10.0.0.0 mask 255.0.0.0
```

Configure all routers accordingly.

How to configure BGP? For short:

1. Declare your own AS number by "router bgp *as-number*".
2. Define neighbors with "neighbor *Address* remote-as *as-number*".
3. Define the networks you own by "network *Address* mask *Mask*".

## **CONCLUSIONS:**

We have successfully configured dynamic routing protocols.

**Experiment No.: 9. Date: Title: Design & configure small network by using Sub net.**

**AIM:** Network design and implementation for small network using subnetting with IP address scheme

**OBJECTIVES:**

To design a small network and implementation of that small network using subnetting

**SOFTWARES USED:**

Cisco Packet Tracer

**THEORY:****Network Devices****Hubs**

At the bottom of the networking food chain, so to speak, are hubs. Hubs are used in networks that use twisted pair cabling to connect devices. Hubs can also be joined together to create larger networks. Hubs are simple devices that direct data packets to all devices connected to the hub, regardless of whether the data package is destined for the device. This makes them inefficient devices and can create a performance bottleneck on busy networks.

In its most basic form, a hub does nothing except provide a pathway for the electrical signals to travel along. Such a device is called a passive hub. Far more common nowadays is an active hub, which, as well as providing a path for the data signals, regenerates the signal before it forwards it to all of the connected devices. A hub does not perform any processing on the data that it forwards, nor does it perform any error checking.

**Passive hub:**

It does nothing except provide a pathway for the electrical signals to travel along

**Active hub:**

As well as providing a path for the data signals, regenerates the signal before it forwards it to all of the connected devices.

**Intelligent hub:**

It is enabled for remote monitoring and management through Simple Network Management Protocol (SNMP). An intelligent hub contains Management Information Base (MIB) information that specifies which conditions can be monitored and which functions can be managed

**Switches**

Like hubs, switches are the connectivity points of an Ethernet network. Devices connect to switches via twisted pair cabling, one cable for each device. The difference between hubs and switches is in how the devices deal with the data that they receive. Whereas a hub forwards the data it receives to all of the ports on the device, a switch forwards it only to the port that connects to the destination device. It does this by learning the MAC address of the devices attached to it, and then by matching the destination MAC address in the data it receives.

- **Cut-through:** In a cut-through switching environment, the packet begins to be forwarded as soon as it is received. This method is very fast, but creates the possibility of errors being propagated through the network, as there is no error checking.



- **Store-and-forward:** Unlike cut-through, in a store-and-forward switching environment, the entire packet is received and error checked before being forwarded. The upside of this method is that errors are not propagated through the network. The downside is that the error checking process takes a relatively long time, and store-and-forward switching is considerably slower as a result.
- **FragmentFree:** To take advantage of the error checking of store-and-forward switching, but still offer performance levels nearing that of cutthrough switching, FragmentFree switching can be used. In a FragmentFree-switching environment, enough of the packet is read so that the switch can determine whether the packet has been involved in a collision. As soon as the collision status has been determined, the packet is forwarded.

## **Bridges**

Bridges are used to divide larger networks into smaller sections. They do this by sitting between two physical network segments and managing the flow of data between the two. By looking at the MAC address of the devices connected to each segment, bridges can elect to forward the data (if they believe that the destination address is on another interface), or block it from crossing (if they can verify that it is on the interface from which it came).

## **Routers**

In a common configuration, routers are used to create larger networks by joining two network segments. A router can be a dedicated hardware device or a computer system with more than one network interface and the appropriate routing software. All modern network operating systems include the functionality to act as a router. A router derives its name from the fact that it can route data it receives from one network onto another. When a router receives a packet of data, it reads the header of the packet to determine the destination address. Once it has determined the address, it looks in its routing table to determine whether it knows how to reach the destination and, if it does, it forwards the packet to the next hop on the route. The next hop might be the final destination, or it might be another router.

## **Gateways**

Any device that translates one data format to another is called a gateway. Some examples of gateways include a router that translates data from one network protocol to another, a bridge that converts between two networking systems, and a software application that converts between two dissimilar formats. The key point about a gateway is that only the data format is translated, not the data itself. In many cases, the gateway functionality is incorporated into another device.

## **Firewalls**

A firewall is a networking device, either hardware or software based, that controls access to your organization's network. This controlled access is designed to protect data and resources from an outside threat. To do this, firewalls are typically placed at entry/exit points of a network—for example, placing a firewall between an internal network and the Internet. Once there, it can control access in and out of that point. Although firewalls typically protect internal networks from public networks, they are also used to control access between specific network segments within a network—for example, placing a firewall between the Accounts and the Sales departments.

## **Networking Cables**

Cable is the medium through which information usually moves from one network device to another. There are several types of cable which are commonly used with LANs. In some cases, a network will utilize only one type of cable, other networks will use a variety of cable types. The type of cable chosen for a network is related to

the network's topology, protocol, and size. Understanding the characteristics of different types of cable and how they relate to other aspects of a network is necessary for the development of a successful network.

The following sections discuss the types of cables used in networks and other related topics.

- Unshielded Twisted Pair (UTP) Cable
- Shielded Twisted Pair (STP) Cable
- Coaxial Cable
- Fiber Optic Cable

- Cable Installation Guides
- Wireless LANs
- Unshielded Twisted Pair (UTP) Cable

Twisted pair cabling comes in two varieties: shielded and unshielded. Unshielded twisted pair (UTP) is the most popular and is generally the best option for school networks. The quality of UTP may vary from telephone grade wire to extremely high-speed cable. The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other electrical devices.

Fig. Unshielded twisted pair

### **Categories of Unshielded Twisted Pair**

#### **Category Speed Use**

1	1 Mbps Voice Only (Telephone Wire)
2	4 Mbps LocalTalk & Telephone (Rarely used)
3	16 Mbps 10BaseT Ethernet
4	20 Mbps Token Ring (Rarely used)
5	100 Mbps (2 pair) 100BaseT Ethernet
	1000 Mbps (4 pair) Gigabit Ethernet
5e	1,000 Mbps Gigabit Ethernet
6	10,000 Mbps Gigabit Ethernet

### **Unshielded Twisted Pair Connector**

The standard connector for unshielded twisted pair cabling is an RJ-45 connector. This is a plastic connector that looks like a large telephone-style connector. A slot allows the RJ-45 to be inserted only one way. RJ stands for Registered Jack, implying that the connector follows a standard borrowed from the telephone industry. This standard designates which wire goes with each pin inside the connector.

Fig. RJ-45 connector

### **Shielded Twisted Pair (STP) Cable**

Although UTP cable is the least expensive cable, it may be susceptible to radio and electrical frequency interference (it should not be too close to electric motors, fluorescent lights, etc.). If you must place cable in environments with lots of potential interference, or if you must place cable in extremely sensitive environments that may be susceptible to the electrical current in the UTP, shielded twisted pair may be the solution. Shielded cables can also help to extend the maximum distance of the cables.

Shielded twisted pair cable is available in three different configurations:

1. Each pair of wires is individually shielded with foil.
2. There is a foil or braid shield inside the jacket covering all wires (as a group).
3. There is a shield around each individual pair, as well as around the entire group of wires (referred to as double shield twisted pair).

## Coaxial Cable

Coaxial cabling has a single copper conductor at its center. A plastic layer provides insulation between the center conductor and a braided metal shield. The metal shield helps to block any outside interference from fluorescent lights, motors, and other computers.

Fig. Coaxial cable

Although coaxial cabling is difficult to install, it is highly resistant to signal interference. In addition, it can support greater cable lengths between network devices than twisted pair cable. The two types of coaxial cabling are thick coaxial and thin coaxial. Thin coaxial cable is also referred to as thinnet. 10Base2 refers to the specifications for thin coaxial cable carrying Ethernet signals. The 2 refers to the approximate maximum segment length being 200 meters. In actual fact the maximum segment length is 185 meters. Thin coaxial cable has been popular in school networks, especially linear bus networks. Thick coaxial cable is also referred to as thicknet. 10Base5 refers to the specifications for thick coaxial cable carrying Ethernet signals. The 5 refers to the maximum segment length being 500 meters. Thick coaxial cable has an extra protective plastic cover that helps keep moisture away from the center conductor. This makes thick coaxial a great choice when running longer lengths in a linear bus network. One disadvantage of thick coaxial is that it does not bend easily and is difficult to install.

## Coaxial Cable Connectors

The most common type of connector used with coaxial cables is the Bayonet-Neill-Concelman (BNC) connector. Different types of adapters are available for BNC connectors, including a T-connector, barrel

connector, and terminator. Connectors on the cable are the weakest points in any network. To help avoid problems with your network, always use the BNC connectors that crimp, rather screw, onto the cable.

Fig. BNC connector

## Fiber Optic Cable

Fiber optic cabling consists of a center glass core surrounded by several layers of protective materials. It transmits light rather than electronic signals eliminating the problem of electrical interference. This makes it ideal for certain environments that contain a large amount of electrical interference. It has also made it the standard for connecting networks between buildings, due to its immunity to the effects of moisture and lighting. Fiber optic cable has the ability to transmit signals over much longer distances than coaxial and twisted pair. It also has the capability to carry information at vastly greater speeds. This capacity broadens communication possibilities to include services such as video conferencing and interactive services. The cost of fiber optic cabling is comparable to copper

cabling; however, it is more difficult to install and modify. 10BaseF refers to the specifications for fiber optic cable carrying Ethernet signals.

The center core of fiber cables is made from glass or plastic fibers. A plastic coating then cushions the fiber center, and kevlar fibers help to strengthen the cables and prevent breakage. The outer insulating jacket made of Teflon or PVC.

Fig. Fiber optic cable

There are two common types of fiber cables -- single mode and multimode. Multimode cable has a larger diameter; however, both cables provide high bandwidth at high speeds. Single mode can provide more distance, but it is more expensive.

### **IP Address Classes**

### **Special Purpose Addresses**

### **CONFIGURATION**

We have to make four blocks like IT, COMP, FE and ETC.  
Let's we create for block of address from 192.168.10.0

Given subnet mask = 255.255.255.0  
= 11111111. 11111111. 11111111. 11000000  
= 255.255.255.192

Subnets =  $2^2 = 4$

Hosts =  $2^6 = 64$

Subnet 1 192.168.10.1 - 192.168.10.62  
Subnet 2 192.168.10.64 - 192.168.10.127  
Subnet 3 192.168.10.128 - 192.168.10.191  
Subnet 4 192.168.10.192 - 192.168.10.254

60

MITSoE S.Y CSE

Configure Router

Router>en

Router#conf t

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

Router(config)#int fa 0/0

Router(config-if)#no shut

Create subinterface for IT.

Router(config-if)#int fa0/0.10

Router(config-subif)#encapsulation dot1Q 10

Router(config-subif)#ip add 192.168.10.1 255.255.255.192

Router(config-subif)#no shut

Create subinterface for COMP.

Router(config-subif)#int fa 0/0.20

Router(config-subif)#encapsulation dot1Q 20

Router(config-subif)#ip address 192.168.10.70 255.255.255.192

Router(config-subif)#no shut

Create subinterface for FE.

```
Router(config-subif)#int fa0/0.30
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 192.168.10.130 255.255.255.192
Router(config-subif)#no shut
```

Create subinterface for ETC.

```
Router(config-subif)#int fa0/0.40
Router(config-subif)#encapsulation dot1Q 40
Router(config-subif)#ip address 192.168.10.200 255.255.255.192
Router(config-subif)#no shut
```

### **CONCLUSIONS:**

We have designed a small network and implemented it using subnetting with IP address scheme.

### **Experiment No.: 10 & 11. Date: Title: Socket Programming**

**AIM:** Implement sockets using any socket APIs

- a. TCP Sockets
- b. UDP Sockets

### **OBJECTIVES:**

To implement client server for arithmetic operations using TCP & UDP sockets.

### **SOFTWARES USED:**

GCC Compiler

### **THEORY:**

Sockets provide an interface for programming networks at the transport layer. Network communication using Sockets is very much similar to performing file I/O. In fact, socket handle is treated like file handle. The streams used in file I/O operation are also applicable to socket-based I/O. Socket-based communication is independent of a programming language used for implementing it. That means, a socket program written in Java language can communicate to a program written in non-Java (say C or C++) socket program. A server (program) runs on a specific computer and has a socket that is bound to a specific port. The server listens to the socket for a client to make a connection request. If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to a different port. It needs a new socket (consequently a different port number) so that it can continue to listen to the original socket for connection requests while serving the connected client.

## **TCP/IP SOCKET PROGRAMMING**

The two key classes from the java.net package used in creation of server and client programs are:

- ServerSocket
- Socket

62

MITSoE S.Y CSE

A server program creates a specific type of socket that is used to listen for client requests (server socket). In the case of a connection request, the program creates a new socket through which it will exchange data with the client using input and output streams. The socket abstraction is very similar to the file concept: developers have to open a socket, perform I/O, and close it. Below figure illustrates key steps involved in creating socket-based server and client programs.

### **A simple Server Program in Java**

#### **1. Open the Server Socket:**

```
ServerSocket server = new ServerSocket( PORT );
```

#### **2. Wait for the Client Request:**

```
Socket client = server.accept();
```

#### **3. Create I/O streams for communicating to the client**

```
    DataInputStream is = new DataInputStream(client.getInputStream());
    DataOutputStream os = new DataOutputStream(client.getOutputStream());
```

**4. Perform communication with client Receive from client:**

```
    String line = is.readLine(); Send to client: os.writeBytes(—Hello\n\n);
```

**5. Close socket:**

```
        client.close();
```

### **A simple Client Program in Java**

**1. Create a Socket Object:**

```
        Socket client = new Socket(server, port_id);
```

**2. Create I/O streams for communicating with the server.**

```
        is = new DataInputStream(client.getInputStream());
        os = new DataOutputStream(client.getOutputStream());
```

**3. Perform I/O or communication with the server:**

```
        Receive data from the server: String line = is.readLine();
        Send data to the server: os.writeBytes(—Hello\n\n);
```

**4. Close the socket when done:**

```
        client.close();
```

## **UDP SOCKET PROGRAMMING**

Datagram packets are used to implement a connectionless packet delivery service supported by the UDP protocol. Each message is transferred from source machine to destination based on information contained within that packet. That means, each packet needs to have destination address and each packet might be routed differently, and might arrive in any order. Packet delivery is not guaranteed. The format of datagram packet is:

Java supports datagram communication through the following classes:

- DatagramPacket
- DatagramSocket

The class DatagramPacket contains several constructors that can be used for creating packet object. One of them is:

```
        DatagramPacket(byte[] buf, int length, InetAddress address, int port);
```

This constructor is used for creating a datagram packet for sending packets of length *length* to the specified port number on the specified host. The message to be transmitted is indicated in the first argument. The key methods of DatagramPacket class are:

```
byte[] getData()
```

Returns the data buffer.

```
int getLength()
```

Returns the length of the data to be sent or the length of the data received.

```
void setData(byte[] buf)
```

Sets the data buffer for this packet.

```
void setLength(int length)
```

Sets the length for this packet.

The class *DatagramSocket* supports various methods that can be used for transmitting or receiving data a datagram over the network.

The two key methods are:

```
void send(DatagramPacket p)
```



Sends a datagram packet from this socket.

*void receive(DatagramPacket p)*

Receives a datagram packet from this socket.

### **CONCLUSIONS:**

We have successfully implemented TCP & UDP socket programming.