



**CHANDIGARH
UNIVERSITY**
Discover. Learn. Empower.

Apex Institute of Technology
Program Name: B.E.-AIT-CSE (All)
LAB MANUAL

Semester	: 5th
Course Name	: Computer Networks Lab
Course Code	: 22CSH-335
Course Coordinator	: Dr. Ratish Kumar

SN	22CSH-335	Course Name: Computer Networks	L	T	P	S	C	CH	Course Type
4		Course Coordinator: Dr. Ratish Kumar	3	0	2	0	4	5	Core
PREREQUISITE		Basic Knowledge of Networking concepts, C or Java programming is required.							
CO-REQUISITE		Nil							
ANTI-REQUISITE		Nil							

a. Course Objectives:

1. To bring together several key of Computer network design and architecture
2. To familiarize the student with the basic taxonomy and terminology of the computer networking area.
3. To allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks

b. Course Outcomes

CO1	To develop an understanding of basic networking concepts
CO2	To implement the functionality of different Algorithm and Protocols.
CO3	To learn about different connection establishments techniques
CO4	To understand TCP and UDP model and communication and connection establishments techniques
CO5	To learn conjunction control techniques

Lab Experiments with CO Mapping

S.NO.	Experiment	Mapped CO
Unit 1:- Basic Structure of Networking		
1	Understand the working of following (i) IP Address. (ii) Cisco IOS. (iii) Straight Cable & Cross Cable, RJ45 (iv) Layer 2 Switch. (v) Router.	CO1
2	Study the basic network command and Network configuration commands like ping, variations of ip config, tracert, nslookup, netstat, arp, rarp, hostname, pathping and basic networking commands.	CO1
3	Configure and understand working of network devices hub, switch and router.	CO1
Unit 2:- Basics of Routing		
4	Implementation of Static Routing using n routers.	CO2
5	Implement Dynamic Routing using RIP (Routing Information Protocol)	CO2
6	Implement VLAN and VLAN Trunking protocols.	CO4
7	Implement Router as DHCP server that can serve multiple VLAN's.	CO4
Unit 3 :- Advanced Theories of Networking		
8	Using Socket programming implement the Connection oriented using standard Ports in any programming language (Java/Python etc).	CO3
9	Using Socket programming implement the Connectionless oriented using standard Ports in any programming language (Java/Python etc).	CO3
10	Capture and analyse network packets using network packet analyzer.	CO4

MODE OF EVALUATION: The performance of students is evaluated as follows:

	Practical	
Components	Continuous Internal Assessment (CAE)	Semester End Examination (SEE)
Marks	60	40
Total Marks	100	

Course Outcome	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O 1	PS O 2	PSO 3
CO1	1	1	2	2	1	-	-	-	-	-	-	-	2	1	2
CO2	-	1	2	-	2	-	-	-	-	-	-	-	-	1	-
CO3	-	2	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	2	-	-	3	-	-	-	-	-	-	-	-	1	-	2
CO5	2	1	2	2	-	-	-	-	-	-	-	1	2	-	2

EXPERIMENT 1.1

Mapped Course Outcomes- CO1

CO1: Understand the working of routers, switches and hubs

AIM: Understand the working of following (i) IP Address. (ii) Cisco IOS. (iii) Straight Cable & Cross Cable, RJ45 (iv) Layer 2 Switch. (v) Router.

Apparatus required:

Cisco Packet Tracer

Objective: - Students will understand concepts of

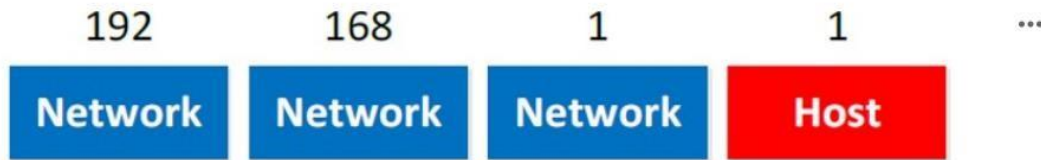
1. IP Address
2. Cisco IOS
3. Straight Cable & Cross Cable, RJ45
4. Layer2 Switch
5. Routers

S/W Requirement: - NA

a) IP Address.

A core function of IP address is to provide logical addressing for hosts. An IP address both uniquely identify a **host** and **network** of that host. An IP address is a 32-bit number that uniquely identifies a host (computer or other device, such as a printer or router) on a TCP/IP network. IP addresses are normally expressed in dotteddecimal format, with four numbers separated by periods, such as 192.168.123.132. To understand how subnet masks are used to distinguish between hosts, networks, and subnetworks, examine an IP address in binary notation. For example, the dotted-decimal IP address 192.168.123.132 is (in binary notation) the 32 bit number 110000000101000111101110000100. This number may be hard to make sense of, so divide it into four parts of eight binary digits. These eight bit sections are known as octets.

The second item, which is required for TCP/IP to work, is the subnet mask. The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network. In TCP/IP, the parts of the IP address that are used as the network and host addresses are not fixed, so the network and host addresses above cannot be determined unless you have more information. This information is supplied in another 32-bit number called a subnet mask.



IP Address Classes.

Internet addresses are allocated by the InterNIC, the organization that administers the Internet. These IP addresses are divided into classes. The most common of these are classes A, B, and C. Classes D and E exist, but are not generally used by end users. Each of the address classes has a different default subnet mask. You can identify the class of an IP address by looking at its first octet. Following are the ranges of Class A, B, and C Internet addresses, each with an example address.

- Class A networks use a default subnet mask of 255.0.0.0 and have 0-127 as their first octet. The address 10.52.36.11 is a class A address. Its first octet is 10, which is between 1 and 126, inclusive.
- Class B networks use a default subnet mask of 255.255.0.0 and have 128-191 as their first octet. The address 172.16.52.63 is a class B address. Its first octet is 172, which is between 128 and 191, inclusive.
- Class C networks use a default subnet mask of 255.255.255.0 and have 192-223 as their first octet. The address 192.168.123.132 is a class C address. Its first octet is 192, which is between 192 and 223, inclusive.

b) Cisco IOS.

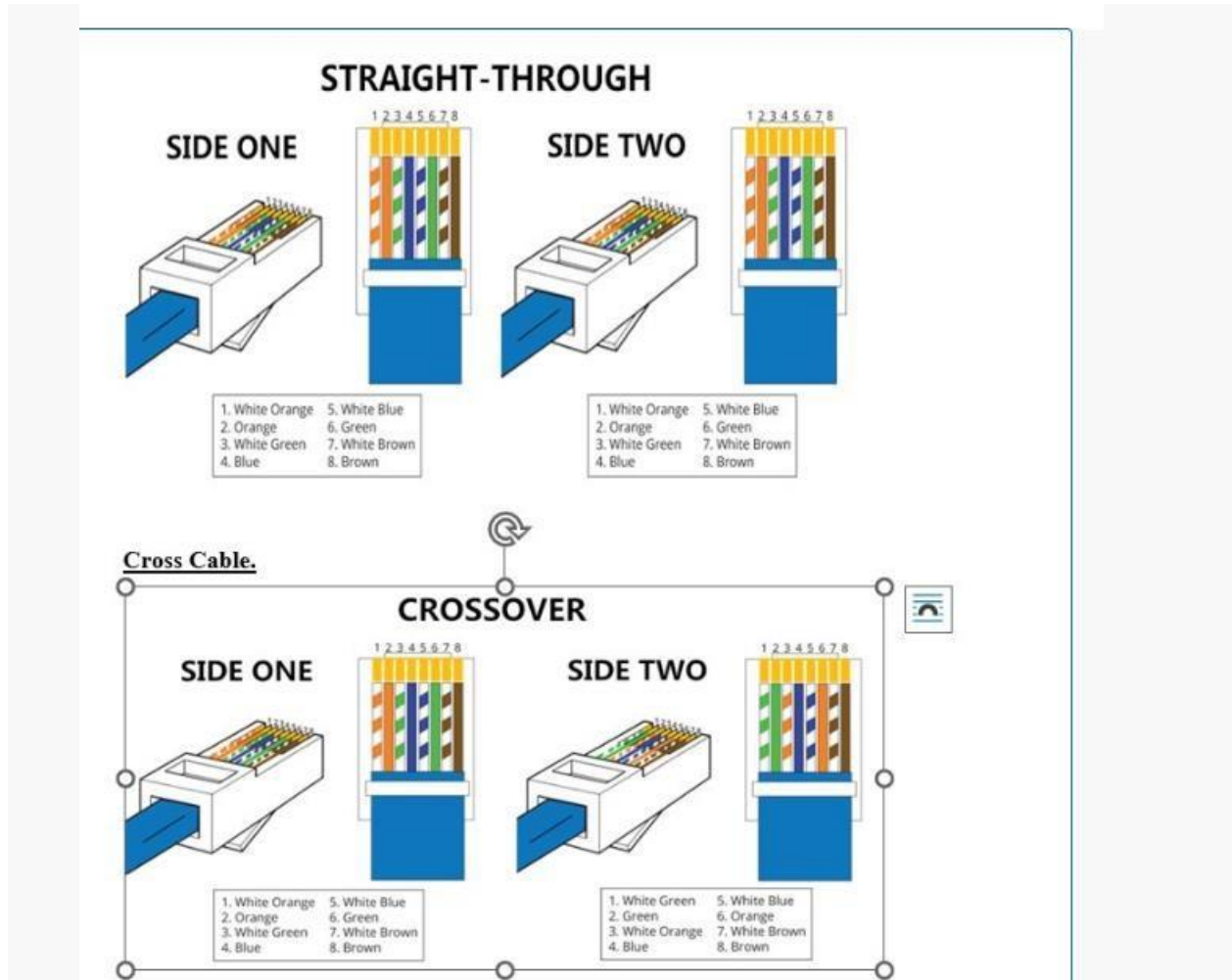
The Cisco IOS is a command-line interface used by nearly all current Cisco Routers & Switches. Cisco IOS, formally the Internetwork Operating System, is a family of network operating systems used on many Cisco Systems routers and current Cisco network switches. Earlier, Cisco switches ran CatOS. IOS is a package of routing, switching, internetworking and telecommunications functions integrated into a multitasking operating system. Although the IOS code base includes a cooperative multitasking kernel, most IOS features have been ported to other kernels such as QNX and Linux for use in Cisco products.

c) Straight Cable & Cross Cable, RJ45.

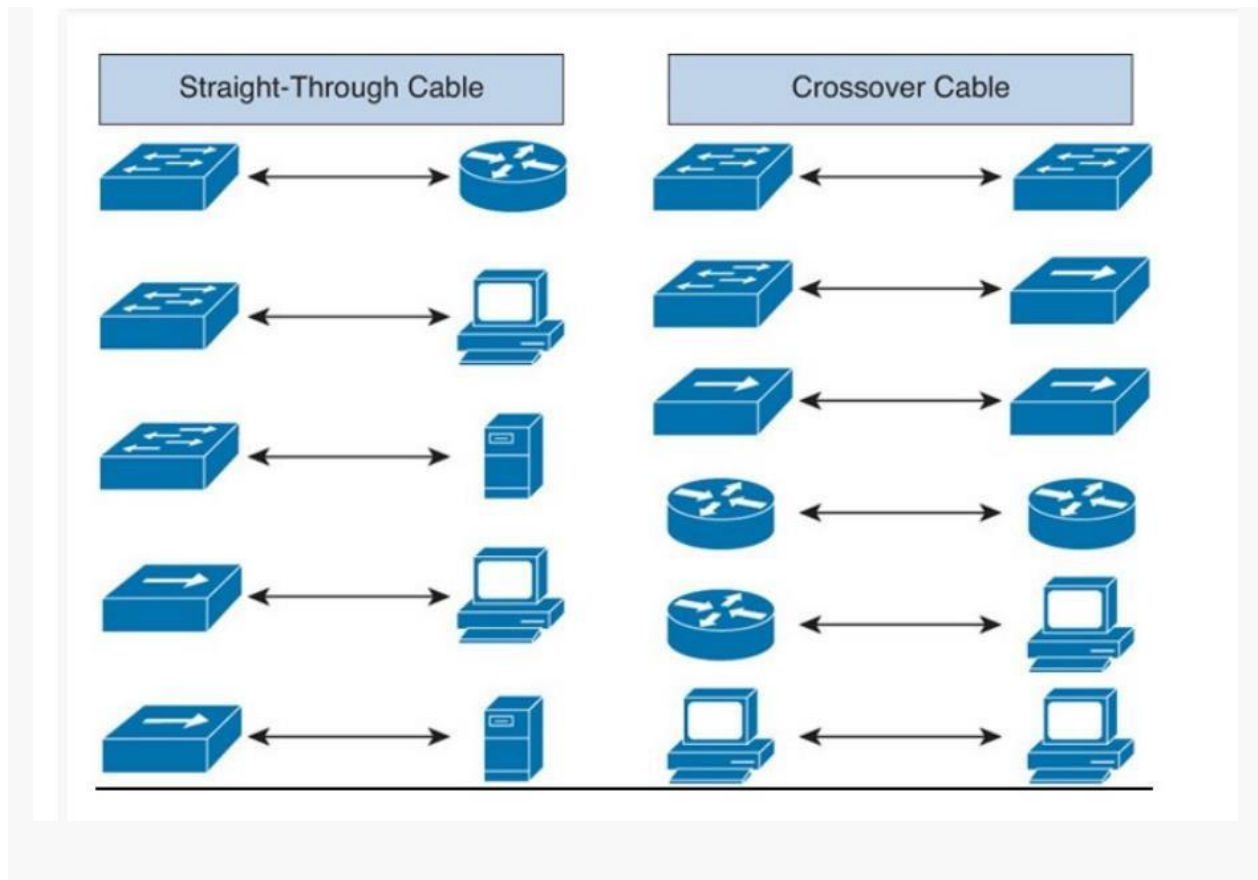
The cable can be categorized as Cat 5, Cat 5e, and Cat 6 UTP cable. Cat 5 UTP cable can support 10/100 Mbps Ethernet network, whereas Cat 5e and Cat 6 UTP cable can support Ethernet network running at 10/100/1000 Mbps. You might hear about Cat 3 UTP cable, it's not popular anymore since it can only support 10 Mbps

Ethernet network. Straight and crossover cable can be Cat3, Cat 5, Cat 5e or Cat 6 UTP cable, the only difference is each type will have different wire arrangement in the cable for serving different purposes.

Straight Cable.



Cross Cable.



RJ45 Connector.

Registered Jack 45 (RJ45) is a standard type of physical connector for network cables. RJ45 connectors are most commonly seen with Ethernet cables and networks. Modern Ethernet cables feature small plastic plugs on each end that are inserted into the RJ45 jacks of Ethernet devices.



d) Layer-2 Switch.

d) Layer-2 Switch.

A layer 2 switch is a type of network switch or device that works on the data link layer (OSI Layer 2) and utilizes MAC Address to determine the path through where the frames are to be forwarded. It uses hardware based switching techniques to connect and transmit data in a local area network (LAN).



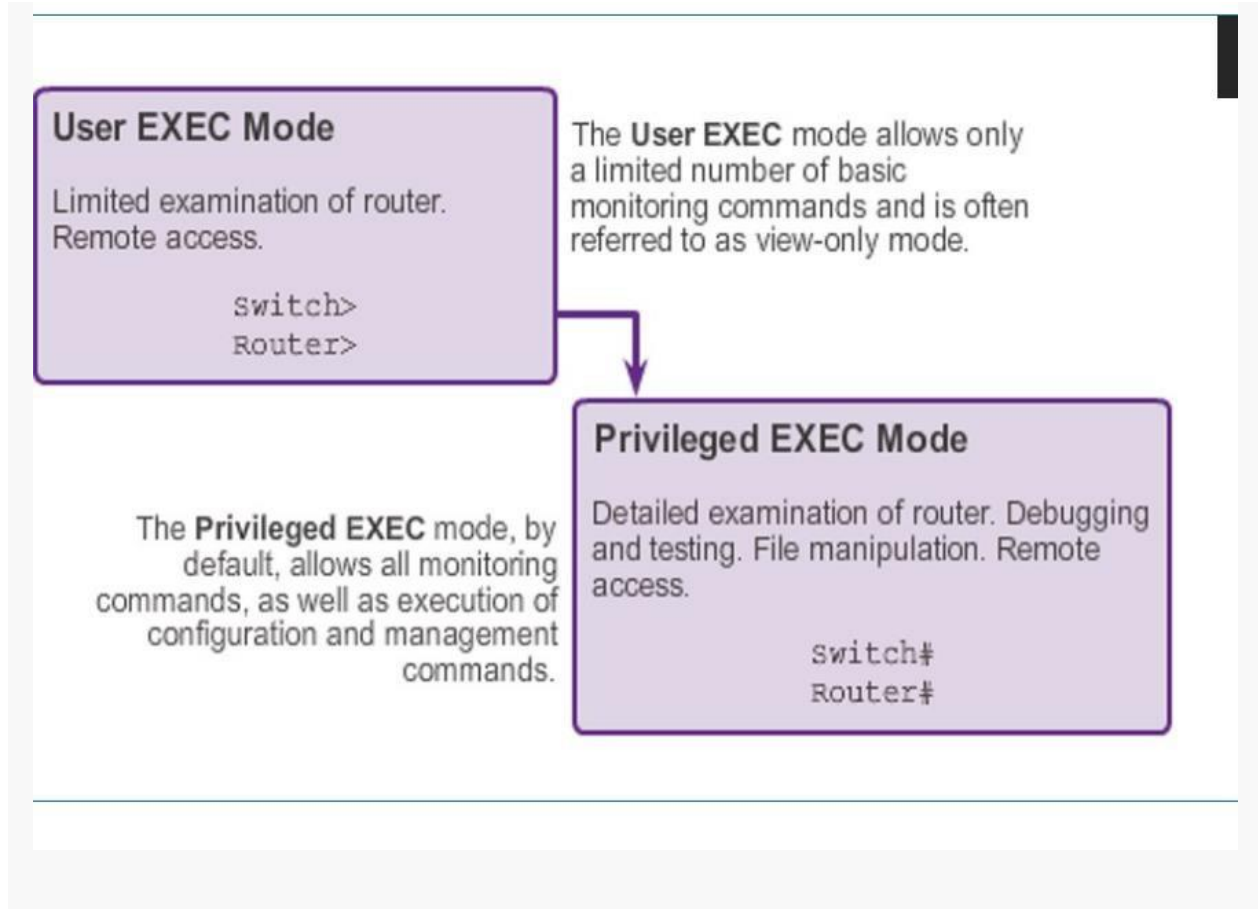
d) Routers.

A router is a networking device that forwards data packets between computer networks. Routers perform the traffic directing functions on the Internet. Data sent through the internet, such as a web page or email, is in the form of data packets. A packet is typically forwarded from one router to another router through the networks that constitute an internetwork (e.g. the Internet) until it reaches its destination node.

Router Memory Components.

Memory	Volatile / Non-Volatile	Stores
RAM	Volatile	<ul style="list-style-type: none"> Running IOS Running configuration file IP routing and ARP tables Packet buffer
ROM	Non-Volatile	<ul style="list-style-type: none"> Bootup instructions Basic diagnostic software Limited IOS
NVRAM	Non-Volatile	<ul style="list-style-type: none"> Startup configuration file
Flash	Non-Volatile	<ul style="list-style-type: none"> IOS Other system files

Router Modes.



Further Reading

1. <https://geekflare.com/understanding-ip-address/>
2. <https://docs.oracle.com/cd/E19504-01/802-5753/planning3-18471/index.html>
3. <https://www.ibm.com/cloud/learn/networking-a-complete-guide>

Research Articles:

1. <https://www.researchgate.net/publication/323511648> **INTRODUCTION TO NET WORKING**
2. <https://www.tandfonline.com/doi/full/10.1080/21642850.2018.1521283>

EXPERIMENT 1.2

Mapped Course Outcomes-CO1

CO1: Understand the working of routers, switches and hubs.

Aim: Study the basic network command and Network configuration commands like ping, variations of ip config, traceroute, nslookup, netstat, arp, rarp, hostname, pathping and basic networking commands.

Networking Command

These commands are as follow-

1. Ping

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

For Example: If an Internet connection is not in the office, for instance, the ping utility is used to determine if the problem exists in the office or the Internet provider's network. The following shows an image of how ping tools to obtain the locally connected router's connectivity status.

There are various options a user can use with the Ping command.

Options are as follows-

Options	Description
target	This is the destination IP address or a hostname user want to ping.
-a	This option resolves the hostname of an IP address target.
-t	This ping command option will ping the target until you stop it by pressing Ctrl-C.
-n count	This option is used to set the number of ICMP Echo Requests to send, from 1 to 4294967295. If -n is not specified, the ping command will return 4 by default.
-l size	This option is used to set the size, in bytes, of the echo-request packet from 32 to 65,527. If the -l option is not specified, the ping command will send a 32-byte echo request.
-s count	This option is used to report the time in the Internet Timestamp format that each echo request is received and an echo reply is sent. The maximum count value is 4, i.e. only the first four hops can be time stamped.
-r count	This command uses the ping command option to specify the number of hops between the source computer and the target computer. The maximum count value is 9; the Tracert command can also be used if the user wants to view all the hops between two devices.
-i TTL	This ping command option sets the Time to Live (TTL) value; the maximum value is 255.

-f	Use this ping command option to prevent ICMP Echo Requests from being fragmented by routers between the source and the target. The -f option is often used to troubleshoot Path Maximum Transmission Unit (PMTU) issues.
-w timeout	A timeout value must be specified while executing this ping command. It adjusts the amount of time in milliseconds. If the -w option is not specified, then the default timeout value of 4000 is set, which is 4 seconds.
-p	To ping a Hyper-V Network Virtualization provider address.
-S srcaddr	This option is used to specify the source address.

2. NetStat

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

There are various options a user can use with the Netstat command.

Options are as follows-

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

3. Ip Config

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

4. Hostname

To communicate with each and other, the computer needs a unique address. A hostname can be alphabetic or alphanumeric and contain specific symbols used specifically to define a specific node or device in the network. For example, a hostname should have a domain name (TLD) of the top-level and a distance between one and 63 characters when used in a domain name system (DNS) or on the Internet.

Steps to Determine Your Computer's Name

Open a terminal window and type the command given below.

hostname

It will provide the name of your computer. The first part of the result is the name of a computer and the second part is the name of the domain.

To get only the computer name, run the following

command: `hostname -s`

The output will be localhost.

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

`hostname -d`

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

command.” `hostname -i`

User can find out all the aliases for the computer by using the command given below.

`hostname -a`

5. Tracert

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

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

operating systems. The syntax for Tracert Command `tracert [-d] [-h MaxHops] [-`

`w Timeout] target`

There are various options the user can use with tracert command.

Options for tracert Command are as follows-

- **target:** This is the destination, either an IP address or hostname.
- **-d:** This option prevents Tracert from resolving IP addresses to hostnames to get faster results.
- **-h MaxHops:** This Tracert option specifies the maximum number of hops in the search for the target. If the MaxHops option is not specified the target has not been found by 30 hops, then the tracert command will stop looking.

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

6. Nslookup

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

The syntax for Nslookup is as follows.

Nslookup

or

Nslookup [domain_name]

7. Route

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

8. ARP

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

9. Path Ping

We discussed the Ping command and the Tracert command. There are similarities between these commands. The pathping command which provides a combination of the best aspects of Tracert and Ping.

This command takes 300 seconds to gather statistics and then returns reports on latency and packet loss statistics at intermediate hops between the source and the target in more detail than those reports provided by Ping or Tracert commands.

The syntax for path ping is as follows:

```
path ping [-n] [-h] [-g <Hostlist>] [-p <Period>] [-q <NumQueries>] [-w <timeout>] [-i
<IPaddress>] [-4
<IPv4>] [-6 <IPv6>][<TargetName>]
```

- **N:** Prevents path ping functioning from attempting to resolve routers' IP addresses to their names.

- **-h MaxHops:** This tracert option specifies the maximum number of hops in the search for the target. If the MaxHops option is not specified the target has not been found by 30 hops then the tracert command will stop looking.
 - **-w timeout:** A timeout value must be specified while executing this ping command. It adjusts the amount of time in milliseconds.
 - **-ip <IPaddress>:** Indicates the source address.
 - **target:** This is the destination IP address or a hostname user want to ping.
-

Further Reading

4. <https://geekflare.com/understanding-ip-address/>
5. <https://docs.oracle.com/cd/E19504-01/802-5753/planning3-18471/index.html>
6. <https://www.ibm.com/cloud/learn/networking-a-complete-guide>

Research Articles:

3. <https://www.researchgate.net/publication/323511648> **INTRODUCTION TO NETWORKING**
4. <https://www.tandfonline.com/doi/full/10.1080/21642850.2018.1521283>

EXPERIMENT 1.3

Mapped Course Outcomes-CO1

CO1: Understand the working of routers, switches and hubs

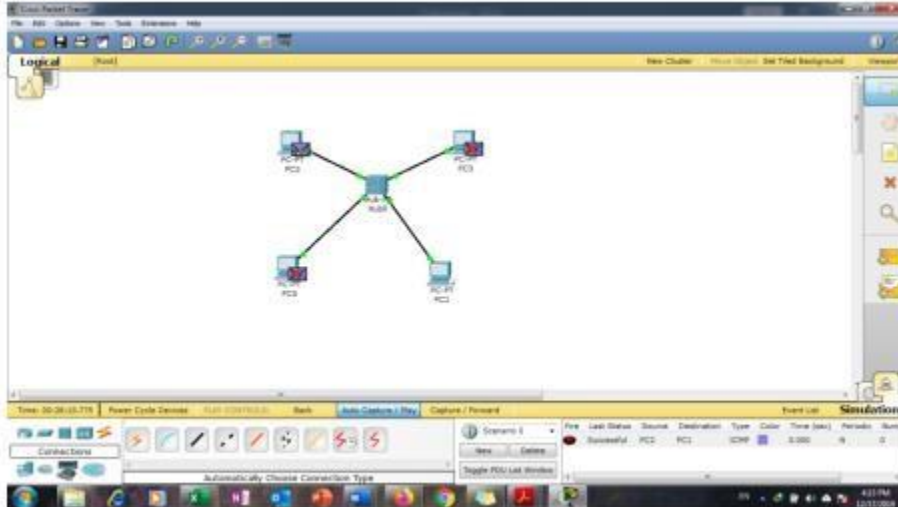
Aim-

Configure and understand working of network devices hub, switch and router..

Procedure-

1. Attach required devices (Hub/Switch/Router) in the packet tracer software.
2. Assign IP address to devices.
3. Select source and destination and drop packet from source to destination.
4. Go to Simulation mode and click capture/Play.
5. Simulation will start and packet will only be accepted by destination

SAMPLE OUTPUT:



Procedure: Following should be done to understand this practical.

1. Repeater: Functioning at Physical Layer. A repeater is an electronic device that receives a signal and retransmits it at a higher level and/or higher power, or onto

the other side of an obstruction, so that the signal can cover longer distances.

Repeater have two ports ,so cannot be use to connect for more than two devices

2. Hub: An Ethernet hub, active hub, network hub, repeater hub, hub or concentrator is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and making them act as a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. The device is a form of multiport repeater. Repeater hubs also participate in collision detection, forwarding a jam signal to all ports if it detects a collision.

3. Switch: A network switch or switching hub is a computer networking device that connects network segments. The term commonly refers to a network bridge that processes and routes data at the data link layer (layer 2) of the OSI model. Switches that additionally process data at the network layer (layer 3 and above) are often referred to as Layer 3 switches or multilayer switches.

4. Bridge: A network bridge connects multiple network segments at the data link layer (Layer 2) of the OSI model. In Ethernet networks, the term bridge formally means a device that behaves according to the IEEE 802.1D standard. A bridge and switch are very much alike; a switch being a bridge with numerous ports. Switch or Layer 2 switch is often used interchangeably with bridge. Bridges can analyze incoming data packets to determine if the bridge is able to send the given packet to another segment of the network.

5. Router: A router is an electronic device that interconnects two or more computer networks, and selectively interchanges packets of data between them. Each data packet contains address information that a router can use to determine if the source and destination are on the same network, or if the data packet must be transferred from one network to another. Where multiple routers are used in a large

collection of interconnected networks, the routers exchange information about target system addresses, so that each router can build up a table showing the preferred paths between any two systems on the interconnected networks.

6. Gate Way: In a communications network, a network node equipped for interfacing with another network that uses different protocols.

- A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It also requires the establishment of mutually acceptable administrative procedures between both networks.
- A protocol translation/mapping gateway interconnects networks with different network protocol technologies by performing the required protocol conversions

Further Reading

7. <https://geekflare.com/understanding-ip-address/>
8. <https://docs.oracle.com/cd/E19504-01/802-5753/planning3-18471/index.html>
9. <https://www.ibm.com/cloud/learn/networking-a-complete-guide>

Research Articles:

5. [**https://www.researchgate.net/publication/323511648 INTRODUCTION TO NET WORKING**](https://www.researchgate.net/publication/323511648)
6. [**https://www.tandfonline.com/doi/full/10.1080/21642850.2018.1521283**](https://www.tandfonline.com/doi/full/10.1080/21642850.2018.1521283)

EXPERIMENT 1.4

Mapped Course Outcomes-CO2

CO2: Analyze the different routing techniques.

Aim:- Implementation of Static Routing using n routers.

Objective: How static routing is created , maintained and updated by a network administrator using static route.

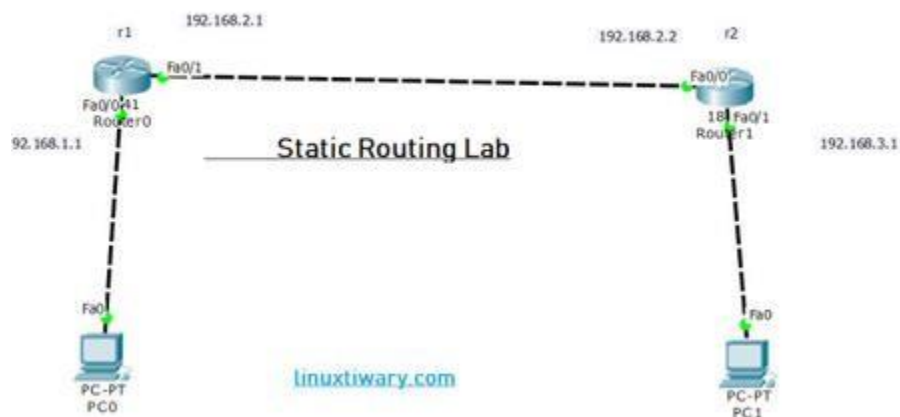
Prerequisites:-

1. An access to CISCO PACKET TRACER .

A static routing table is created, maintained, and updated by a network administrator, manually. A static route to every network must be configured on every router for full connectivity. This provides a granular level of control over routing but quickly becomes impractical on large networks. Routers will not share static routes with each other, thus reducing CPU/RAM overhead and saving bandwidth. However, static routing is not fault-tolerant, as any change to the routing infrastructure (such as a link going down, or a new network added) requires manual intervention. Routers operating in a purely static environment cannot seamlessly choose a better route if a link becomes unavailable. Static routes have an Administrative Distance (AD) of 1, and thus are always preferred over dynamic routes, unless the default AD is changed. A static route with an adjusted AD is called a floating static route.[1]

Step by Step Procedure:

Topology.



Configuration.

Router0 Configuration.

Router#conf terminal

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

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip address 10.0.0.1 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#interface fastEthernet 1/0

Router(config-if)#ip address 192.168.1.1 255.255.255.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#ip route 20.0.0.0 255.0.0.0 192.168.1.1

Router(config)#exit

Router1 Configuration.

Router#conf terminal

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

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip address 20.0.0.1 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#interface fastEthernet 1/0

Router(config-if)#ip address 192.168.1.2 255.255.255.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#ip route 10.0.0.0 255.0.0.0 192.168.1.2

Router(config)#exit

Further Reading:-

1. <https://www.techopedia.com/definition/26161/static-routing>

Research Articles

1. Dowdell, J. & Benamar, Nabil. (2015). Static Routing for DTN. Gardner, W.R. 1987. Water content: an overview. International Conference on Measurement of Soil and Plant Water Status. Centennial of Utah State Univ., pp. 7-9.
2. Saini, Himanshi and Pondwal, Venu, An Analysis of Static and Dynamic Routing Techniques Based on Various Network Performance Parameters (May 2017). The IUP Journal of Telecommunications, Vol. IX, No. 2, May 2017, pp. 20-31, Available at SSRN: <https://ssrn.com/abstract=3212546>

Experiment 1.5

Mapped Course Outcomes-CO2

CO2: Analyze the different routing techniques.

Aim:- Implement Dynamic Routing using RIP (Routing Information Protocol)

Prerequisites:-

1. An access to CISCO PACKET TRACER .

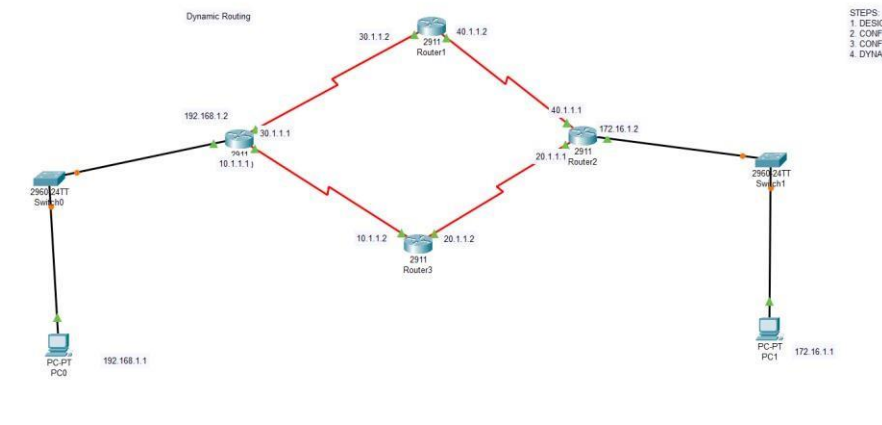
RIP is a standardized Distance Vector protocol, designed for use on smaller networks. RIP was one of the first true Distance Vector routing protocols and is supported on a wide variety of systems. RIP has two versions, Version 1 (RIPv1) and Version 2 (RIPv2). RIPv1 (RFC 1058) is classful and thus does not include the subnet mask with its routing table updates. Because of this, RIPv1 does not support Variable Length Subnet Masks (VLSMs). When using RIPv1, networks must be contiguous, and subnets of a major network must be configured with identical subnet masks. Otherwise, route table inconsistencies (or worse) will occur. RIPv1 sends updates as broadcasts to address 255.255.255.255. RIPv2 (RFC 2543) is classless and thus does include the subnet mask with its routing table updates. RIPv2 fully supports VLSMs, allowing discontinuous networks and varying subnet masks to exist. [1]

RIP Characteristics.

- RIP sends out periodic routing updates (every 30 seconds)
- RIP sends out the full routing table every periodic update
- RIP uses a form of distance as its metric (in this case, hop count)
- RIP uses the Bellman-Ford Distance Vector algorithm to determine the best “path” to a particular destination.
- RIP supports IP and IPX routing.
- RIP utilizes UDP port 520.
- RIP routes have an administrative distance of 120.

- RIP has a maximum hop count of 15 hops.
- Any network that is 16 hops away or more is considered unreachable to RIP, thus the maximum diameter of the network is 15 hops. A metric of 16 hops in RIP is considered a poisoned route or infinity metric.
- If multiple paths exist to a particular destination, RIP will load balance between those paths (by default, up to 4) only if the metric (hop count) is equal. RIP uses an around-robin system of load-balancing between equal metric routes.

Topology.



Configuration.

Router0 Configuration.

Router#conf terminal

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

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip address 192.168.1.1 255.255.255.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)# interface serial 2/0

Router(config-if)#ip address 10.0.0.1 255.0.0.0

```
Router(config-if)#no shut
Router(config-if)#exit
Router(config)# interface serial 3/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#network 192.168.1.0
Router(config)#exit
```

Router1 Configuration.

```
Router#conf terminal

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

Router(config)# interface serial 2/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)# interface serial 3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#exit
```



```
Router(config)#router rip
```

```
Router(config-router)#network 20.0.0.0 Router(config-
```

```
router)#network 30.0.0.0
```

```
Router(config)#exit
```

Router2 Configuration.

```
Router#conf terminal
```

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

```
Router(config)# interface serial 2/0
```

```
Router(config-if)#ip address 40.0.0.1 255.0.0.0
```

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

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

```
Router(config)# interface serial 3/0
```

```
Router(config-if)#ip address 20.0.0.2 255.0.0.0
```

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

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

```
Router(config)#router rip
```

```
Router(config-router)#network 20.0.0.0 Router(config-
```

```
router)#network 40.0.0.0
```

```
Router(config)#exit
```

Router3 Configuration.

```
Router#conf terminal
```

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

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

Router(config-if)#ip address 172.16.10.1 255.255.0.0 Router(config-if)#no shut

Router(config-if)#exit

Router(config)# interface serial 2/0

Router(config-if)#ip address 40.0.0.2 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)# interface serial 3/0

Router(config-if)#ip address 30.0.0.2 255.0.0.0

Router(config-if)#no shut

Router(config-if)#exit

Router(config)#router rip

Router(config-router)#network 40.0.0.0

Router(config-router)#network 30.0.0.0

Router(config-router)#network 172.16.0.0

Router(config)#exit

Further Reading:-

2. <https://www.techopedia.com/definition/26161/static-routing>

Research Articles

3. Dowdell, J. & Benamar, Nabil. (2015). Static Routing for DTN. Gardner, W.R. 1987. Water content: an overview. International Conference on Measurement of Soil and Plant Water Status. Centennial of Utah State Univ., pp. 7-9.
4. Saini, Himanshi and Pondwal, Venu, An Analysis of Static and Dynamic Routing Techniques Based on Various Network Performance Parameters (May 2017). The IUP Journal of Telecommunications, Vol. IX, No. 2, May 2017, pp. 20-31, Available at SSRN: <https://ssrn.com/abstract=3212546>

Experiment 1.6

Mapped Course Outcomes-CO4

CO4: Implement VLAN and VLAN Trunking protocols

Aim- Implement VLAN and VLAN Trunking Protocol.

What will you learn:

How VLAN is created using simple connection or creation of VLAN with the help of trunking protocol.

Prerequisites:-

1. A web browser with access to CISCO PACKET TRACER .

Virtual Local Area Network: Virtual LAN (VLAN) is a concept in which we can divide the devices logically on data Link Layer i.e.(Layer2). Generally, Network Layer(layer 3) devices divides broadcast domain and each broadcast domain can be divided by switches using the concept of VLAN.A broadcast domain is a network segment in which if a device broadcast a packet then all the devices in the same network will receive it.However, due to limitations of switches packets don't send outside the broadcast network.To forward out the packets to different VLAN (from one VLAN to another) or broadcast domain, inter VLAN routing is needed. Through VLAN, different small size sub networks are created which are comparatively easy to handle.

Topology

Topology Diagram

Configuration of VLAN:

VLAN can be created by VLAN ID and VLAN name.

Syntax:

```
#switch1(config)#vlan 2 // Vlan ID
```

```
#switch1(config-vlan)#vlan accounts // Vlan Name
```

```
Switch(config)#int range fa0/0-2
```

```
Switch(config-if) #switchport access Vlan 2
```

VLAN Trunking Protocol [1]

VLAN Trunking Protocol (VTP) is also a Cisco proprietary protocol that propagates the definition of Virtual Local Area Networks (VLAN) on the whole local area network to synchronize the VLAN information in same VTP domain. VTP allows you to add, delete and rename VLANs which is then propagated to other switches in the VTP domain. VTP advertisements can be sent over 802.1Q, and ISL trunk. To do this, VTP carries VLAN information to all the switches in a VTP domain.

Requirements of VTP:

There are some requirements for VTP to communicate VLAN information between switches. These are:

1. The VTP version must be same on the switches user wants to configure
2. VTP domain name must be same on the switches
3. One of the switches must be a server
4. Authentication should match if applied.

Topology for Trunking Protocol

Configuration of Switches.

```
SwitchA> enable
```

```
SwitchA# configure terminal
```

```
SwitchA(config)# interface fa 0/24
```

```
SwitchA(config-if)#switchport trunk encapsulation dot1q
```

```
SwitchA(config-if)# switchport mode trunk
```

```
SwitchA(config-if)# end
```

Conclusion & Discussion.

The trunk is activated. For switch port mode trunk command it is not necessary to use the same command at the other side of the link, without this job it is automatically activated.

Further Readings:

1. <https://www.practicalnetworking.net/stand-alone/vlans/>

Experiment 1.7

Mapped Course Outcomes-CO4

CO4: Deploy the concept to handle traffic and control on it

AIM:- Implement Router as DHCP server that can serve multiple VLAN's.

What will you learn:

How DHCP is used to assign IP addresses dynamically in the network .DHCP network creation with VLAN Protocol.

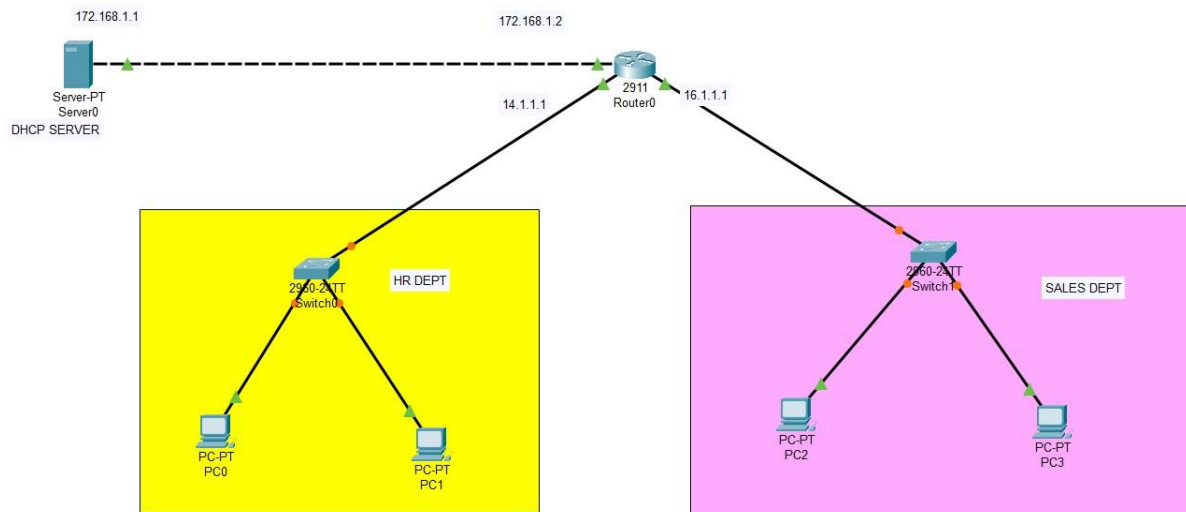
Prerequisites:-

1. An access to CISCO PACKET TRACER .

Dynamic Host Configuration Protocol (DHCP) is an application layer protocol used to distribute various network configuration parameters to devices on a TCP/IP network. – IP addresses, subnet masks, default gateways, DNS servers, etc. DHCP employs a client-server architecture; a DHCP client is configured to request network parameters from a DHCP server on the network. A DHCP server is configured with a pool of available IP addresses and assigns one of them to the DHCP client.[1]

A Cisco router can be configured as a DHCP server.

Topology:



Step by Step Procedure:

Here are the steps:

1. Exclude IP addresses from being assigned by DHCP by using the *ip dhcp excluded-address FIRST_IP LAST_IP*
2. Create a new DHCP pool with the *ip dhcp pool NAME* command.
3. Define a subnet that will be used to assign IP addresses to hosts with the *network SUBNET SUBNET_MASK* command.
4. Define the default gateway with the *default-router IP* command.
5. Define the DNS server with the *dns-server IP* address command.
6. (Optional) Define the DNS domain name by using the *ip domain-name NAME* command.
7. (Optional) Define the lease duration by using the *lease DAYS HOURS MINUTES* command.

If you don't specify this argument, the default lease time of 24 hours will be used.

Example Configuration.

```
Floor1(config)#ip dhcp excluded-address 192.168.0.1 192.168.0.50
```

```
Floor1(config)#ip dhcp pool Floor1DHCP
```

```
Floor1(dhcp-config)#network 192.168.0.0 255.255.255.0
```

```
Floor1(dhcp-config)#default-router 192.168.0.1
```

```
Floor1(dhcp-config)#dns-server 192.168.0.1
```

In the example above you can see that I've configured the DHCP server with the following parameters:

- the IP addresses from the **192.168.0.1 – 192.168.0.50** range will not be assigned to hosts.
- the DHCP pool was created and named **Floor1DHCP**
- the IP addresses assigned to the hosts will be from the **192.168.0.0/24** range
- the default gateway's IP address is **192.168.0.1**
- the DNS server's IP address is **192.168.0.1**

Conclusion & Discussion.

In the output above you can see that there is a single DHCP client that was assigned the IP address of **192.168.0.51**. Since we've excluded the IP addresses from the **192.168.0.1 – 192.168.0.50** range, the device got the first address available – **192.168.0.51**. To display information about the configured DHCP pools, you can use the *show ip dhcp pool* command

Further Reading:-

1. <https://www.geeksforgeeks.org/dynamic-host-configuration-protocol-dhcp/>

Experiment 1.8

Mapped Course Outcomes-CO3

CO3: Identify the connection establishment techniques and features

AIM:- Using Socket Programming implement the connection oriented service using standard ports in any programming language (C,C++, JAVA,Python etc).

Apparatus Required:-Pycharm

What will you learn:

In this lab, we will discuss the socket API and support for TCP communications between end hosts. Socket programming is the key API for programming distributed applications on the Internet.

We plan to learn the following:

- What is a socket?
- The client-server model
- Byte order
- TCP socket API
- Concurrent server design
- Example of echo client and iterative server
- Example of echo client and concurrent server

Prerequisites:-

1. A web browser with access to Pycharm

The basic Concepts

Program: A program is an executable file residing on a disk in a directory. A program is read into memory and is executed by the kernel as a result of an `exec()` function. The `exec()` has six variants, but we only consider the simplest one (`exec()`) in this course.

Process.: An executing instance of a program is called a process. Sometimes, task is used instead of process with the same meaning. UNIX guarantees that every process

has a unique identifier called the process ID. The process ID is always a non-negative integer.

File descriptors: File descriptors are normally small non-negative integers that the kernel uses to identify the files being accessed by a particular process. Whenever it opens an existing file or creates a new file, the kernel returns a file descriptor that is used to read or write the file. As we will see in this course, sockets are based on a very similar mechanism (socket descriptors).

The client-server model

- The client-server model is one of the most used communication paradigms in networked systems.
- Clients normally communicates with one server at a time.
- From a server's perspective, at any point in time, it is not unusual for a server to be communicating with multiple clients. Client need to know of the existence of and the address of the server, but the server does not need to know the address of (or even the existence of) the client prior to the connection being established
- Client and servers communicate by means of multiple layers of network protocols.

Client Code.

```
// A Java program
```

```
for a Client import
```

```
java.net.*; import
```

```
java.io.*;
```

```
public class Client
```

```
{
```

```
    // initialize socket and input
```

```
    output streams    private Socket
```

```
    socket           = null;    private
```

```

DataInputStream input = null;

private DataOutputStream out =

null;

    // constructor to put ip address and
port    public Client(String address,
int port)

    {
        // establish a connection

        try
        {
            socket = new Socket(address, port);

            System.out.println("Connected");

            // takes input from terminal

            input = new

            DataInputStream(System.in);

            // sends output to the socket        out = new

            DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException u)

        {

```

```

        System.out.println(u);
    }
    catch(IOException i)
    {
        System.out.println(i);
    }

    // string to read message from input
    String line = "";

    // keep reading until "Over" is
    input    while
    (!line.equals("Over"))

    {

    t
    r
    y
        {
    line =

```

```

input.readLine();
out.writeUTF(line);
    }
    catch(IOException i)
    {
        System.out.println(i);
    }
}
// close the connection
try
{
input.close
();
out.close();
socket.clos
e();
}
catch(IOException i)
{
    System.out.println(i);
}
}

public static void main(String args[])

```

```

    {
        Client client = new Client("127.0.0.1", 5000);
    }
}

```

Server Code. // A

Java program for a

Server import

java.net.*; import

java.io.*; public

class Server

```

{
    //initialize socket and input
    stream      private Socket
    socket  = null;      private
    ServerSocket  server = null;
    private DataInputStream in
    = null;

```

// constructor with

port public

Server(int port)

```

{
    // starts server and waits for a
    connection

```

```

try
{
    server = new ServerSocket(port);

    System.out.println("Server started");

    System.out.println("Waiting for a client ...");

    socket = server.accept();

    System.out.println("Client accepted");

    // takes input from the client socket
    in = new DataInputStream(        new
    BufferedInputStream(socket.getInputStream()));

    String line = "";

    // reads message from client until "Over" is
    sent        while (!line.equals("Over"))

    {
        tr
        y
        {
            line = in.readUTF();

            System.out.println(line);

        }
    }
}

```

```

        catch(IOException i)
        {
            System.out.println(i);
        }
    }

    System.out.println("Closing connection");

    // close
connection
socket.close();
in.close();
    }

    catch(IOException i)
    {
        System.out.println(i);
    }
}

public static void main(String args[])
{
    Server server = new Server(5000);
}
}

```

Related links and references

1. <https://realpython.com/python-sockets/>

2.<https://docs.python.org/3/howto/sockets.html>

Experiment 1.9

Mapped Course Outcomes-CO3

CO3: Identify the connection establishment techniques and features

AIM:- Using Socket Programming implement the connectionless oriented service using standard ports in any programming language (C,C++, JAVA,Python etc).

Apparatus Required:-Pycharm

The socket() Function

The first step is to call the socket function, specifying the type of communication protocol (TCP based on IPv4, TCP based on IPv6, UDP).

We plan to learn the following:

- What is a socket?
- The client-server model
- Byte order
- UDP socket API
- Concurrent server design
- Example of echo client and iterative server
- Example of echo client and concurrent server.

Prerequisites:-

1. A web browser with access to C,C++, JAVA,Python etc

The function is defined as follows:

```
#include <sys/socket.h> int  
socket (int family, int type, int  
protocol);
```

where family specifies the protocol family (AF_INET for the IPv4 protocols), type is a constant described the type of socket (SOCK_STREAM for stream sockets and SOCK_DGRAM for datagram sockets).

The function returns a non-negative integer number, similar to a file descriptor, that we define socket descriptor or -1 on error.

The connect() Function

The connect() function is used by a TCP client to establish a connection with a TCP server/

The function is defined as follows:

```
#include <sys/socket.h> int connect (int sockfd, const struct  
sockaddr *servaddr, socklen_t addrlen); where sockfd is the  
socket descriptor returned by the socket function.
```

The function returns 0 if the it succeeds in establishing a connection (i.e., successful TCP threeway handshake, -1 otherwise.

The client does not have to call bind() in Section before calling this function: the kernel will choose both an ephemeral port and the source IP if necessary.

The send() Function

Since a socket endpoint is represented as a file descriptor, we can use read and write to communicate with a socket as long as it is connected. However, if we want to specify options we need another set of functions.

For example, send() is similar to write() but allows to specify some options. send() is defined as follows:

```
#include <sys/socket.h> ssize_t send(int sockfd, const  
void *buf, size_t nbytes, int flags);
```

where buf and nbytes have the same meaning as they have with write. The additional argument flags is used to specify how we want the data to be transmitted. We will not consider the possible options in this course. We will assume it equal to 0.

The function returns the number of bytes if it succeeds, -1 on error.

The receive() Function

The `recv()` function is similar to `read()`, but allows to specify some options to control how the data are received. We will not consider the possible options in this course. We will assume it is equal to 0.

receive is defined as follows: `#include`

```
<sys/socket.h> ssize_t recv(int sockfd, void  
*buf, size_t nbytes, int flags);
```

The function returns the length of the message in bytes, 0 if no messages are available and peer had done an orderly shutdown, or -1 on error.

The close() Function

The normal `close()` function is used to close a socket and terminate a TCP socket. It returns 0 if it succeeds, -1 on error. It is defined as follows:

```
#include  
  
<unistd.h>  
  
int close(int  
sockfd);
```

Coding Implementation: Exchange of hello message between server and client is shown to demonstrate the connection-less model.

// Server side implementation of UDP client-server model

```
#include <stdio.h>  
  
#include <stdlib.h>  
  
#include <unistd.h>  
  
#include <string.h>  
  
#include <sys/types.h>  
  
#include <sys/socket.h>  
  
#include <arpa/inet.h>
```

```

#include <netinet/in.h>

#define PORT    8080

#define MAXLINE 1024

//

Driver

code

int

main(

) {

intsoc

kfd;

char

buffer

[MA

XLIN

E];

char

*hello

=

"Hell

o

from

server

```

```

",
structs
ockad
dr_ins
ervad
dr,
cliadd
r;

        // Creating socket file descriptor        if (
(sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {

        perror("socket creation failed");
exit(EXIT_FAILURE);

    }

    memset(&servaddr, 0, sizeof(servaddr));
memset(&cliaddr, 0, sizeof(cliaddr));

    // Filling server information
servaddr.sin_family = AF_INET; // IPv4
servaddr.sin_addr.s_addr = INADDR_ANY;
servaddr.sin_port = htons(PORT);


    // Bind the socket with the server
address        if ( bind(sockfd,
(conststructsockaddr *)&servaddr,

        sizeof(servaddr)) < 0 )

```

```

    {
perror("bind failed");
exit(EXIT_FAILURE);
    }

intlen,

n;

    n = recvfrom(sockfd, (char *)buffer, MAXLINE,

                                MSG_WAITALL, ( structsockaddr *) &cliaddr,

                                &len);

buffer[n] = '\0';          printf("Client :
%s\n", buffer);          sendto(sockfd, (const
char *)hello, strlen(hello),

                                MSG_CONFIRM, (conststructsockaddr *) &cliaddr,

                                len);

    printf("Hello message sent.\n");

return

0;

}

```

// Client side implementation of UDP client-server model

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>


#define PORT    8080

#define MAXLINE 1024


//

Driver

code

int

main(

) {

    int sockfd;

    char buffer[MAXLINE];

    char *hello = "Hello from

    client";

    struct sockaddr_in servaddr;


    // Creating socket file descriptor    if (

(sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {

```



```

        perror("socket creation failed");
exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));

    // Filling server information
servaddr.sin_family = AF_INET;
servaddr.sin_port = htons(PORT);
servaddr.sin_addr.s_addr = INADDR_ANY;
int n, len;

    sendto(sockfd, (const char *)hello, strlen(hello),
        MSG_CONFIRM, (const struct sockaddr *) &servaddr,
        sizeof(servaddr));

    printf("Hello message sent.\n");

    n = recvfrom(sockfd, (char *)buffer, MAXLINE,
        MSG_WAITALL, (struct sockaddr *)
        &servaddr,
        &len);
    buffer[n] = '\0';
    printf("Server : %s\n", buffer);

```

```
close(sockfd);  
  
return 0;  
  
}
```

Further Reading-

1. <https://realpython.com/python-sockets/>
2. <https://docs.python.org/3/howto/sockets.html>

Experiment 1.10

CO MAPPED- CO4 CO4: Deploy the concept to handle traffic and control on it.

Aim- Capturing & Analyzing network packets using Wireshark.

Purpose:

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

Prerequisites:-

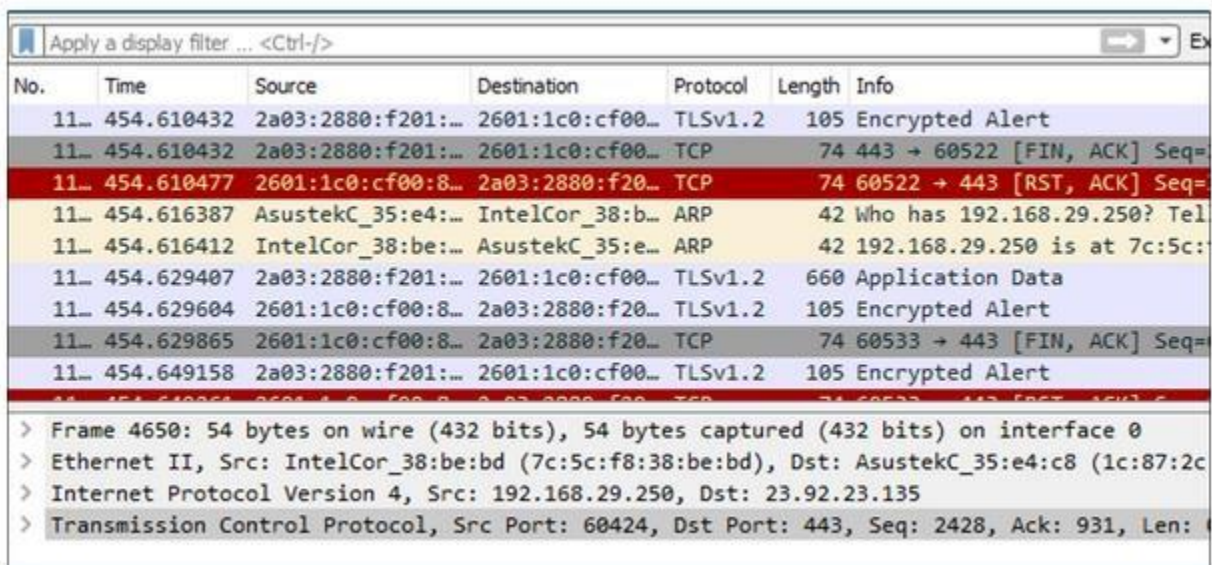
1. A web browser with access to CISCO PACKET TRACER .

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course). In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analyzers available today.[1]

Following are the major benefits of Wireshark.

- Network administrators use it to *troubleshoot network problems*.
- Network security engineers use it to *examine security problems*.
- QA engineers use it to *verify network applications*.
- Developers use it to *debug protocol implementations*.
- People use it to *learn network protocol internals*.

Step by Step Procedure:



The image shows the Wireshark network protocol analyzer interface. At the top, there is a display filter bar with the text "Apply a display filter ... <Ctrl-/>". Below this is a table of captured packets. The table has columns for "No.", "Time", "Source", "Destination", "Protocol", "Length", and "Info". Several packets are listed, including TLSv1.2 Encrypted Alerts, TCP FIN/ACK packets, and ARP requests. One packet (No. 11, Time 454.610477) is highlighted in red. Below the packet list, the "Packet Details" pane shows the structure of the selected packet: Frame 4650, Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
11	454.610432	2a03:2880:f201:...	2601:1c0:cf00:...	TLSv1.2	105	Encrypted Alert
11	454.610432	2a03:2880:f201:...	2601:1c0:cf00:...	TCP	74	443 → 60522 [FIN, ACK] Seq=
11	454.610477	2601:1c0:cf00:8...	2a03:2880:f20...	TCP	74	60522 → 443 [RST, ACK] Seq=
11	454.616387	AsustekC_35:e4:...	IntelCor_38:b...	ARP	42	Who has 192.168.29.250? Tel
11	454.616412	IntelCor_38:be:...	AsustekC_35:e...	ARP	42	192.168.29.250 is at 7c:5c:
11	454.629407	2a03:2880:f201:...	2601:1c0:cf00:...	TLSv1.2	660	Application Data
11	454.629604	2601:1c0:cf00:8...	2a03:2880:f20...	TLSv1.2	105	Encrypted Alert
11	454.629865	2601:1c0:cf00:8...	2a03:2880:f20...	TCP	74	60533 → 443 [FIN, ACK] Seq=
11	454.649158	2a03:2880:f201:...	2601:1c0:cf00:...	TLSv1.2	105	Encrypted Alert

> Frame 4650: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: IntelCor_38:be:bd (7c:5c:f8:38:be:bd), Dst: AsustekC_35:e4:c8 (1c:87:2c:
> Internet Protocol Version 4, Src: 192.168.29.250, Dst: 23.92.23.135
> Transmission Control Protocol, Src Port: 60424, Dst Port: 443, Seq: 2428, Ack: 931, Len: 4

Wireshark Interface Diagram

Select the network interface you want to sniff. Note for this demonstration, we are using a wireless network connection. If you are on a local area network, then you should select the local area network interface.

Further Reading-

1. <https://pages.cpsc.ucalgary.ca/~carey/CPSC441/archive/W2018/tutorials/Wireshark.pdf>
2. <https://www.comptia.org/content/articles/what-is-wireshark-and-how-to-use-it>