Lab Manual
Computer Network
[CP213P]

Department of Computer Science and Engineering
School of Technology
Pandit Deendayal Petroleum University
Raysan, Gandhinagar

Group No: H2G2
Jay Bhagiya 18BIT040
Jehil Thakkar 18BIT043

Index

Experiment - 1

Aim: To study and prepare LAN cable.

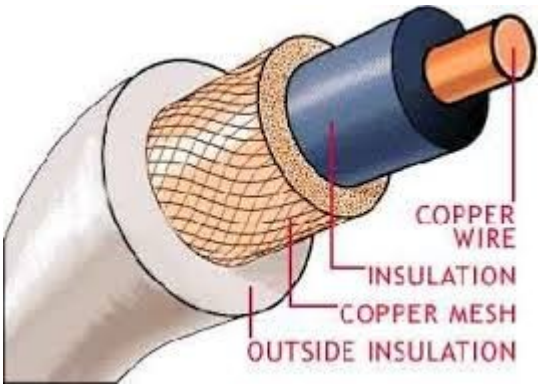
Apparatus/devices: Twisted pair wires, RJ-45 connector, Wire cutter.

Introduction :

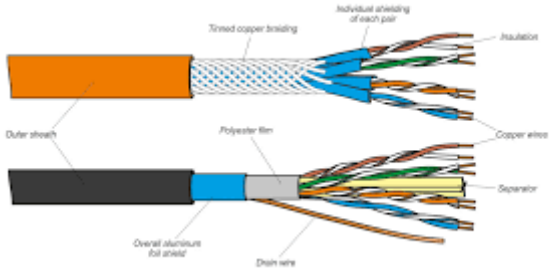
->Types of wires used for transmission purpose:

- Co-axial cables
- Twisted pair cables
- Fiber optics

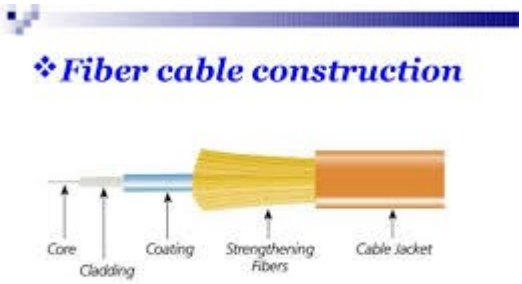
➢**Co-axial cables :** Coaxial cable is very common & widely used commutation media. It got its name because it contains two conductors that are parallel to each other. The center conductor in the cable is usually copper. The copper can be either a solid wire or stranded material. Outside this central Conductor is a non-conductive material. It is usually white, plastic material used to separate the inner Conductor form the outer Conductor. The other Conductor is a fine mesh made from Copper. It is used to help shield the cable form EMI. Outside the copper mesh is the final protective cover.



➢ **Twisted pair cables:** The most popular network cabling is twisted pair. It light weight, easy to install, inexpensive and support many different types of network. It also supports the speed of 100 mbps. Twisted pair cabling is made of pairs of solid or stranded copper twisted along each other. The twists are done to reduce vulnerably to EMI and cross talk. The number of pairs in the cable depends on the type. Twisted pair cabling comes in two varieties: shielded and unshielded.



➢ **Fiber optics:** 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. 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 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.

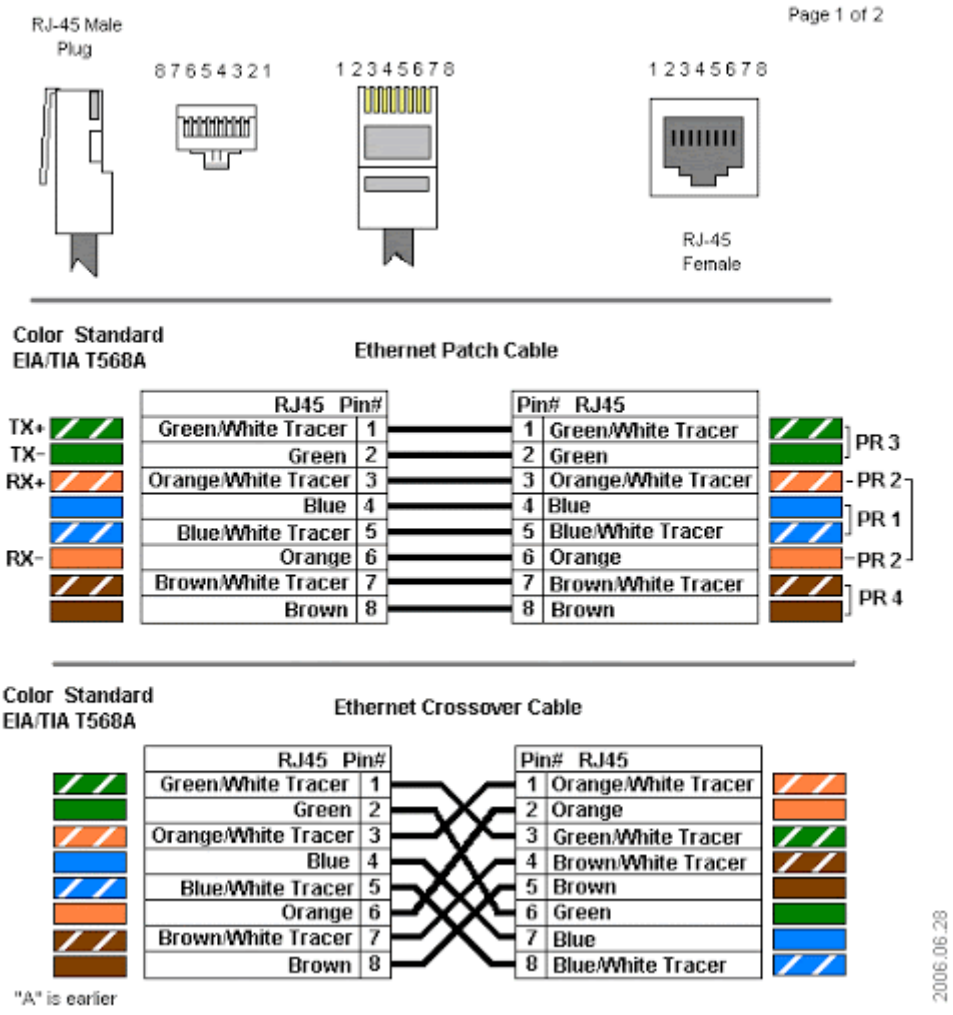


connectors: RJ45 plugs feature eight pins to which the wire strands of a cable interface electrically. Each plug has eight locations spaced about 1 mm apart into which individual wires are inserted using special cable crimping tools. The industry calls this type of connector 8P8C, shorthand for eight position, eight contact.
Ethernet cables and 8P8C connectors must be crimped into the RJ45 wiring pattern to function properly. Technically, 8P8C can be used with other types of connections besides Ethernet; it is also used with RS-232 serial cables, for example. However, because RJ45 is the predominant usage of 8P8C, industry professionals often use the two terms interchangeably.

Traditional dial-up and broadband modems use a variation of RJ45 called RJ45s, which features two contacts in 8P2C configuration instead of eight. The close physical similarity of RJ45 and RJ45s makes it difficult for an untrained eye to tell the two apart. However, they are not interchangeable.

Color codes for straight and crossover cables:

The standards say that Ethernet connectors should be cabled with specific colors on specific pins. There are two standard layouts - if a cable has the same layout on both ends it's a straight through cable. If a cable has one layout on one end and the other layout on the other end then it's a crossover cable. Whilst not universal, the color codes shown below are generally used on professional cables.



Implementation :

- Strip your cable. Use your cable strippers at about 1-2 inches from the end of the cable to remove the outer jacket.
- Untwist the twisted pair wires all the way back to the jacket. This can be done just like a regular twist-tie on a loaf of bread, but with four of them of different colors.
- Align the untwisted wires in the order necessary for your needs. For this scenario, you'll be making a straight-through cable, which has both ends of the cable with the same alignment of wires, so it's easy enough to do. Since this is your first cable, we'll consult the cheat sheet to know what order we're aligning in!
- Cut the extra wire. Once you've untwisted the wires, you'll have a superfluous amount of copper wiring left; we don't need this much, but it's good to have it in the previous step to help in aligning the colors properly. Use the wire-cutting scissors to cut these off.
- Push the remaining wires into the RJ45 head. Be careful not to bend the wires while pushing them in or you run the risk of creating a bad cable. You also don't want too little or too much wire left in the head; there's no definite length necessary, but it's pretty obvious to tell if there's too much cable or not enough. A short length of the jacket should be up the RJ45 head; use this knowledge as a reference.
- Double-check that the wires are all the way up into the gold pins of the head and made it up in the proper order.
- Push the head into the open space of the crimping tool and squeeze it closed, hard. If you don't crimp the cable all the way, the head may come off.
- Open the crimping tool and remove your newly-crimped Ethernet connector.
- Repeat the crimping process on the other side of the cable if you're making a completely new cable. If you're repairing one end, this won't apply to you, so move on.
- Plug one end of the cable into the tan, two-port end of the cable tester, and the other end into the other part of the tester with the graphic display window. Turn it on and listen for the beep. If it beeps once, you successfully made an Ethernet cable; if it beeps twice, some part of the cable is messed up and needs repairing. Depending on the error, the cable may or may not still be usable.

Warning/precaution: There's a reason you need eye-protection: when cutting the copper wires, they can go flying and potentially get your eye! Be smart, be safe!

Learning Outcome : We have learn how to prepare LAN cable.

Experiment - 2

Aim: To configure LAN and perform Static Routing.

Apparatus/Devices: LAN cables, Switch, System (DELL / HP)

etc.

Theory:

Internet working

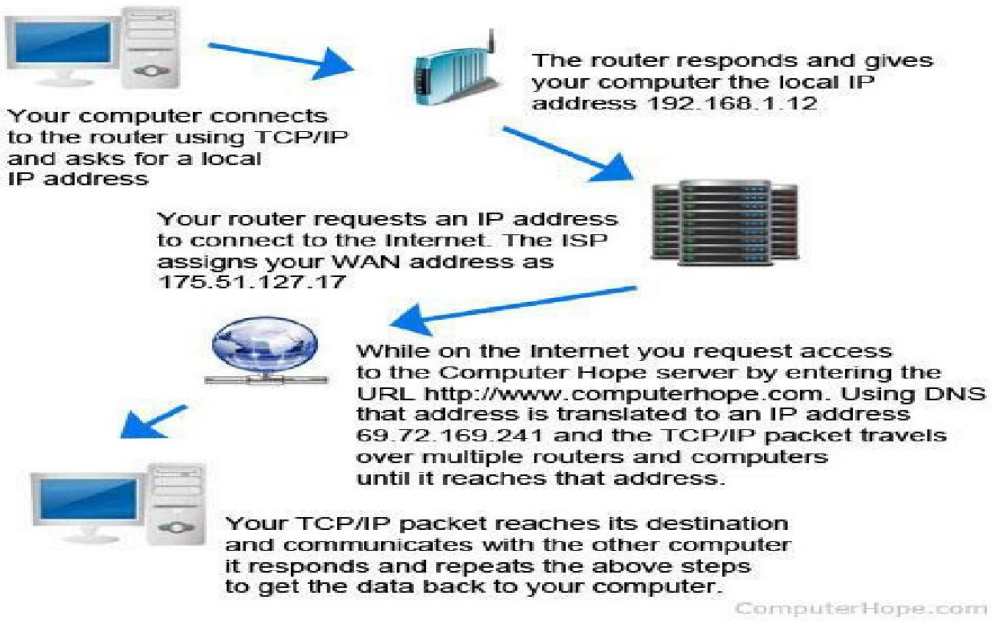
devices:

1. NIC (Network Interface Card): A network card, network adapter, or NIC (network interface card) is a piece of computer hardware designed to allow computers to communicate over a computer network. It provides physical access to a networking medium and often provides a low-level addressing system through the use of MAC addresses.

2. Hub: A network hub contains multiple ports. When a packet arrives at one port, it is copied unmodified to all ports of the hub for transmission. The destination address in the frame is not changed to a broadcast address. It works on the Physical Layer of the OSI model.

3. Switch: A network switch is a device that forwards and filters OSI layer 2 datagrams (chunk of data communication) between ports (connected cables) based on the MAC addresses in the packets, is distinct from a hub in that it only forwards the frames to the ports involved in the communication rather than all ports connected. A switch breaks the collision domain but represents itself as a broadcast domain. Switches make forwarding decisions of frames on the basis of MAC addresses. A switch normally has numerous ports, facilitating a star topology for devices, and cascading additional switches.

4. Router: A router is a 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. A router is a networking device whose software and hardware are customized to the tasks of routing and forwarding information. A router has two or more network interfaces, which may be to different physical types of network (such as copper cables, fiber,or wireless) or different network standards. Each network interface is a small convert electric signals from one form to another.

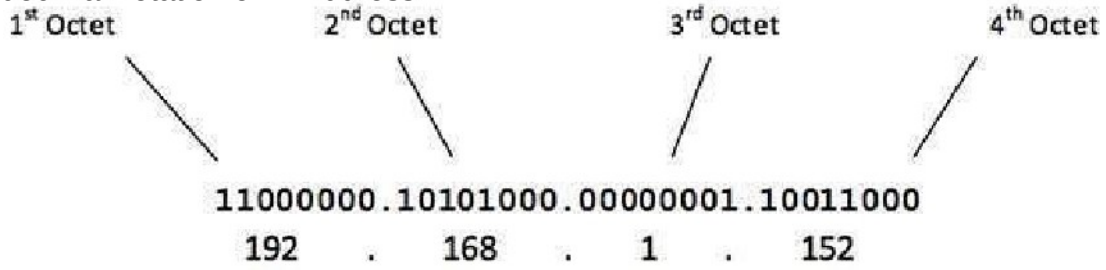

Your computer connects to the router using TCP/IP and asks for a local IP address
The router responds and gives your computer the local IP address 192.168.1.12
Your router requests an IP address to connect to the Internet. The ISP assigns your WAN address as 175.51.127.17
While on the Internet you request access to the Computer Hope server by entering the URL http://www.computerhope.com. Using DNS that address is translated to an IP address 69.72.169.241 and the TCP/IP packet travels over multiple routers and computers until it reaches that address.
Your TCP/IP packet reaches its destination and communicates with the other computer it responds and repeats the above steps to get the data back to your computer.
ComputerHope.com

IP Address and Hostname:

Internet Protocol hierarchy contains several classes of IP Addresses to be used efficiently in various situations as per the requirement of hosts per network. Broadly, the IPv4 Addressing system is divided into five classes of IP Addresses. All the five classes are identified by the first octet of IP Address.

Internet Corporation for Assigned Names and Numbers is responsible for assigning IP addresses.

The first octet referred here is the left most of all. The octets numbered as follows depicting dotted decimal notation of IP Address -



The number of networks and the number of hosts per class can be derived by this formula -

```
Number of networks          = 2^network_bits

Number of Hosts/Network     = 2^host_bits - 2
```

When calculating hosts' IP addresses, 2 IP addresses are decreased because they cannot be assigned to hosts, i.e. the first IP of a network is network number and the last IP is reserved for Broadcast IP.

Class A Address

The first bit of the first octet is always set to 0 (zero). Thus the first octet ranges from 1 - 127, i.e.

```
00000001 - 01111111
       1 - 127
```

Class A addresses only include IP starting from 1.x.x.x to 126.x.x.x only. The IP range 127.x.x.x is reserved for loopback IP addresses.

The default subnet mask for Class A IP address is 255.0.0.0 which implies that Class A addressing can have 126 networks (2_7-2) and 16777214 hosts (2_{24}-2).

ClassA IP address format is thus: **0NNNNNNN**.HHHHHHHH.HHHHHHHH.HHHHHHHH

Class B Address

An IP address which belongs to class B has the first two bits in the first octet set to 10, i.e.

```
10000000 - 10111111
     128 - 191
```

Class B IP Addresses range from 128.0.x.x to 191.255.x.x. The default subnet mask for Class B is 255.255.x.x.

Class B has 16384 (2_{14}) Network addresses and 65534 (2_{16}-2) Host addresses.

Class B IP address format is: **10NNNNNN.NNNNNNNN**.HHHHHHHH.HHHHHHHH

Class C Address

The first octet of Class C IP address has its first 3 bits set to 110, that is -

```
11000000 - 11011111
     192 - 223
```

Class C IP addresses range from 192.0.0.x to 223.255.255.x. The default subnet mask for Class C is 255.255.255.x.

Class C gives 2097152 (2_{21}) Network addresses and 254 (2_{8}-2) Host addresses.

Class C IP address format is: **110NNNNN.NNNNNNNN.NNNNNNNN**.HHHHHHHH

Class D Address

Very first four bits of the first octet in Class D IP addresses are set to 1110, giving a range of -

```
11100000 - 11101111
     224 - 239
```

Class D has IP address range from 224.0.0.0 to 239.255.255.255. Class D is reserved for Multicasting. In multicasting data is not destined for a particular host, that is why there is no need to extract host address from the IP address, and Class D does not have any subnet mask.

Class E Address

This IP Class is reserved for experimental purposes only for R&D or Study. IP addresses in this class ranges from 240.0.0.0 to 255.255.255.254. Like Class D, this class too is not equipped with any subnet mask.

Steps:

1. Go to Ethernet setting and Add new network. Give proper name with individual node.
2. In individual node give network id, IP address and gateway if needed.
3. Now check for given three Scenario and verify observation table. 4. To Enable IP Forwarding in machine with two NIC Card.

Step 1. Login as a admin or root or cnl, Open Terminal
Step 2. Type "sudo gedit /etc/sysctl.conf"
Step 3. Go to the following lines:
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
Step 4. To enable IP Forwarding, Uncomment above line and to stop IP Forwarding, comment the above one.
Step 5. To apply the changes to sysctl.conf, either restart the machine or type "sudo sysctl -p".

5. In Scenario 5 we have to add route for transferring data one node to another node. So following steps are for adding new route.

"route" command : To see the routing table.
"sudo route add -net net-address netmask sub-netmask gw gateway-ip": To add the route in routing table.
Ex:
1. sudo route add -net 10.0.0.0 netmask 255.0.0.0 gw 172.16.0.1
2. sudo route add -net 192.168.16.0 netmask 255.255.255.0 gw 172.16.1.1

Observation Table:

Observation	NODE 1 [WITH SINGLE NIC]	NODE 2 [WITH 2 NIC]	NODE 3 [WITH SINGLE NIC]
Scenario 1	**Net. ID: 10.100.0.0** **IP Add. : 10.100.0.2 [A]** **NO GATEWAY**	**Net. ID: 10.0.0.0 [NIC 1]** **IP Add. 10.100.0.1 [B-1] [NIC 1]** **Net. ID: 172.16.0.0 [NIC 2]** **IP Add: 172.16.0.1 [B-2] [NIC 2]** **NO GATEWAY** **IP Forwarding Enable: No**	**Net. ID: 172.16.0.0** **IP Add. 172.16.0.2 [C]** **NO GATEWAY**
Observation 1	Ping: [A] ---> [B-1]	Observation : PING SUCCESS	
	Ping: [C] ---> [B-2]	Observation : PING SUCCESS	
	Ping: [A] ---> [B-2]	Observation : PING FAIL [N/W UNREACHABLE]	
	Ping: [C] ---> [B-1]	Observation : PING FAIL [N/W UNREACHABLE]	
	Ping: [A] ---> [C]	Observation : PING FAIL [N/W UNREACHABLE]	
Scenario 2	**Net. ID: 10.100.0.0** **IP Add. : 10.100.0.2 [A]** **Gateway : 10.100.0.1**	**Net. ID: 10.0.0.0 [NIC 1]** **IP Add. 10.100.0.1 [B-1] [NIC 1]** **Net. ID: 172.16.0.0 [NIC 2]** **IP Add: 172.16.0.1 [B-2] [NIC 2]** **No Gateway** **IP Forwarding Enable: No**	**Net. ID: 172.16.0.0** **IP Add. 172.16.0.2 [C]** **Gateway : 172.16.0.1**
Observation 2	Ping: [A] ---> [B-1]	Observation : PING SUCCESS	
	Ping: [C] ---> [B-2]	Observation : PING SUCCESS	
	Ping: [A] ---> [B-2]	Observation : PING SUCCESS	

	Ping: [C] ---> [B-1]	Observation : PING SUCCESS	
	Ping: [A] ---> [C]	Observation : PING FAIL [DESTINATION HOST UNREACHABLE]	
Scenario 3	**Net. ID: 10.100.0.0 IP Add. : 10.100.0.2 [A] Gateway : 10.100.0.1**	**Net. ID: 10.0.0.0 [NIC 1] IP Add. 10.100.0.1 [B-1] [NIC 1] Net. ID: 172.16.0.0 [NIC 2] IP Add: 172.16.0.1 [B-2] [NIC 2] No Gateway IP Forwarding Enable: Yes**	**Net. ID: 172.16.0.0 IP Add. 172.16.0.2 [C] Gateway : 172.16.0.1**
Observation 3	Ping: [A] ---> [B-1]	Observation : PING SUCCESS	
	Ping: [C] ---> [B-2]	Observation : PING SUCCESS	
	Ping: [A] ---> [B-2]	Observation : PING SUCCESS	
	Ping: [C] ---> [B-1]	Observation : PING SUCCESS	
	Ping: [A] ---> [C]	Observation : PING SUCCESS	

Scenario 4	**Net. ID: 10.100.0.0 IP Add. : 10.100.1.2 [A] Gateway : 10.100.1.1**	**Net. ID: 10.0.0.0 [NIC 1] IP Add. 10.100.1.1 [B-1] [NIC 1] Net. ID: 172.16.0.0 [NIC 2] IP Add: 172.16.1.1 [B-2] [NIC 2] Net. ID: 192.168.16.0 [NIC 2] IP Add: 192.168.16.1 [B-3] [NIC 3] No Gateway IP Forwarding Enable: Yes**	**Net. ID: 172.16.0.0 IP Add. 172.16.1.2 [C-1] Gateway : 172.16.1.1 Net. ID: 192.168.16.0 IP Add. 192.168.16.2 [C-1] Gateway : 192.168.16.1**

Observation 4	Ping: [A] ---> [B-1]	Observation : PING SUCCESS
	Ping: [A] ---> [B-2]	Observation : PING SUCCESS
	Ping: [A] ---> [B-3]	Observation : PING SUCCESS
	Ping: [C-1] ---> [B-1]	Observation : PING SUCCESS
	Ping: [C-1] ---> [B-2]	Observation : PING SUCCESS
	Ping: [C-1] ---> [B-3]	Observation : PING SUCCESS
	Ping: [C-2] ---> [B-1]	Observation : PING SUCCESS
	Ping: [C-2] ---> [B-2]	Observation : PING SUCCESS
	Ping: [C-2] ---> [B-3]	Observation : PING SUCCESS
	Ping: [C-1] ---> [A]	Observation : PING SUCCESS
	Ping: [C-2] ---> [A]	Observation : PING SUCCESS
	Ping: [A] ---> [C-1]	Observation : PING SUCCESS
	Ping: [A] ---> [C-2]	Observation : PING SUCCESS
NOW, DISCONNECT THE NETWORK OF 192.168.16.0 From Middle Device. Means, Down the link **[B-3] and perform above all ping again.**		
Observation 5	Ping: [A] ---> [B-1]	Observation : PING SUCCESS
	Ping: [A] ---> [B-2]	Observation : PING SUCCESS
	Ping: [A] ---> [B-3]	Observation : PING SUCCESS
	Ping: [C-1] ---> [B-1]	Observation : PING SUCCESS
	Ping: [C-1] ---> [B-2]	Observation : PING SUCCESS
	Ping: [C-1] ---> [B-3]	Observation : PING SUCCESS
	Ping: [C-2] ---> [B-1]	Observation : PING SUCCESS
	Ping: [C-2] ---> [B-2]	Observation : PING SUCCESS
	Ping: [C-2] ---> [B-3]	Observation : PING SUCCESS
	Ping: [C-1] ---> [A]	Observation : PING SUCCESS
	Ping: [C-2] ---> [A]	Observation : PING SUCCESS
	Ping: [A] ---> [C-1]	Observation : PING SUCCESS
	Ping: [A] ---> [C-2]	Observation : PING SUCCESS

Experiment - 3

Aim – Configuring a dhcp server.

Theory - The Dynamic Host Configuration Protocol (DHCP) client TCP/IP software is not configured with a static IP address and it is configured to obtain an IP address dynamically from a Dynamic Host Configuration Protocol (DHCP) Server. When a DHCP client device boots up, it not capable send and receive network traffic, because TCP/IP is not configured. But it can participate in broadcast traffic. DHCP Clients and DHCP Servers uses broadcast messages to communicate with each other. The scope of a broadcast message is only within the local broadcast domain. Broadcast messages will never cross the router to reach another network, because Routers drop Limited Broadcast IP Address.

*The process of leasing TCP/IP configuration from the Dynamic Host Configuration Protocol (DHCP) server involves four steps as listed below.

1. **DHCPDISCOVER**: The Dynamic Host Configuration Protocol (DHCP) client broadcasts a DHCP discover message on the network containing its MAC address destined for UDP port number 68 (used by BOOTP and Dynamic Host Configuration Protocol (DHCP) servers). This first datagram is known as a DHCPDISCOVER message, which is a request to any DHCP Server that receives the datagram for configuration information. As the name implies, the purpose of DHCPDISCOVER message is to discover a DHCP server.

2. **DHCPOFFER**: DHCPDISCOVER Message was delivered to every connected computers in the Broadcast Domain. Every DHCP Server in the Broadcast Domain which received the DHCPDISCOVER message responds with a DHCPOFFER message. Other computers simply drop the DHCPDISCOVER Message.

DHCPOFFER Message contains the offered TCP/IP Configuration values like IPv4 address and subnet mask. If the DHCP client device received multiple DHCPOFFER, the DHCP client accepts the first DHCPOFFER Message that arrives.

3. **DHCPREQUEST**: The Dynamic Host Configuration Protocol (DHCP) client accepts an offer and broadcasts a DHCPREQUEST datagram. The DHCPREQUEST datagram contains the IP address of the server that issued the offer and the physical address (MAC Address) of the DHCP client. DHCPREQUEST message requests the selected DHCP server to assign the DHCP client an IP address and other TCP/IP configuration values.

DHCPREQUEST message also notifies all other DHCP servers that their offers were not accepted by the DHCP client.

4. **DHCPACK**: When the DHCP server from which the offer was selected receives the DHCPREQUEST datagram, it constructs a DHCPACK datagram. This datagram is known as a DHCPACK (DHCP ACKNOWLEDGEMENT). The DHCPACK includes an IP address and subnet mask for the DHCP client. It may include other TCP/IP configuration information like IP address of the default gateway, IP addresses of DNS servers, IP addresses of WINS servers etc.

List of commands:

COMMAND	USE
sudo apt-get update	To update the system
sudo apt-get install isc-dhcp-server -y	To install DHCP Server
sudo gedit /etc/dhcp/dhcpd.conf	To open dhcp configuration file
sudo systemctl status isc-dhcpserver.service	To check the status of DHCP Server
sudo systemctl start isc-dhcpserver.service	To start the DHCP Server
sudo systemctl restart isc-dhcp- server.service	To restart the DHCP Server
sudo systemctl stop isc-dhcpserver.service	To stop the DHCP Server
Ifconfig	To know the interface name
ip a	To know the ip address of DHCP Server

Setup DHCP Server:

Steps:

- Update the system
- Install DHCP server in Ubuntu
- Create network with **DHCP-SERVER** name.
- Go to IPv4 setting, Select manual option. Add new and Allocate IP Address : 10.0.0.2, Subnet mask: 8. o No Gateway o Save it.
- Get the name of interface on which **DHCP-SERVER** network is running. Ex: **etho0**
- Open the DHCP configuration file.Go to following line:

  #option definitions common to all supported networks.. option domain-name "example.org";
  option domain-name-servers 8.8.8.8;

  default-lease-time 600; max-lease-time 7200;
  INTERFACES="eth o0";

- Go to following line:

  # This is a very basic subnet

  declaration. subnet 10.0.0.0 netmask

  255.0.0.0 { range 10.0.0.2 10.0.0.20;
  option routers 10.0.0.1;
  option domain-name-servers 8.8.8.8; }

- Go to following line:

  #network, the authoritative directive should be uncommented. authoritative;

- Save the configuration file and restart the DHCP service. Check the status of DHCP Service.

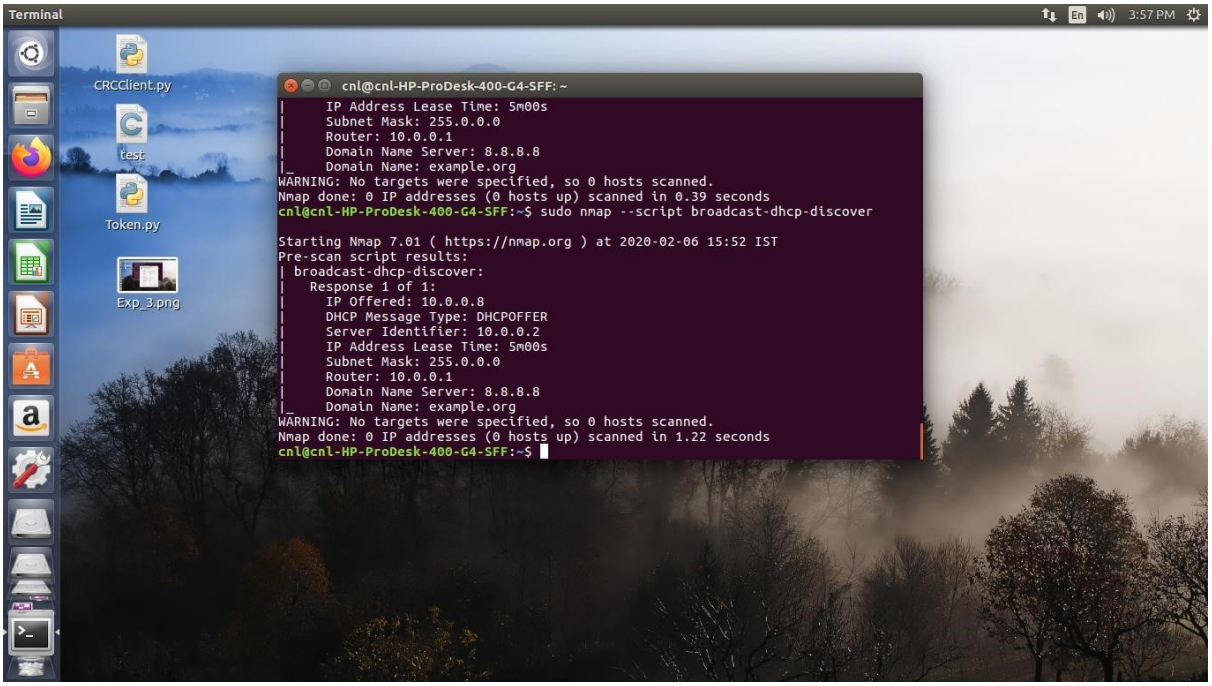
Setup DHCP Client:

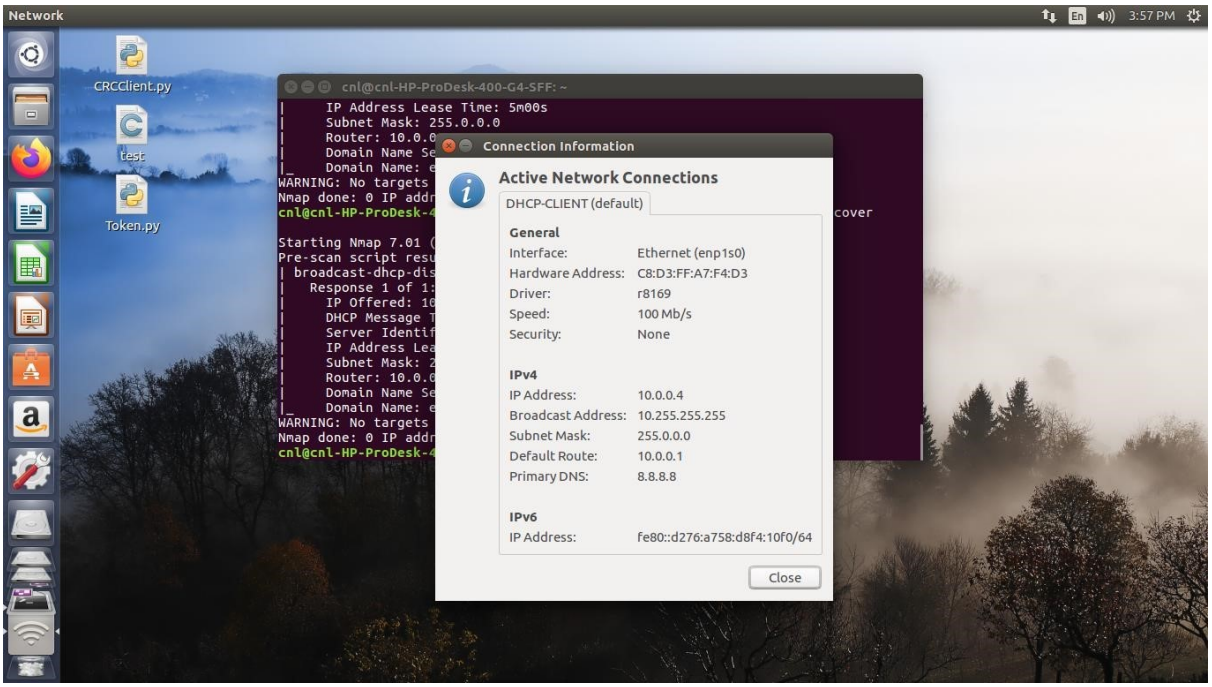
- Go to edit connection and create new Ethernet connection as follow.
- Connection name: DHCP-CLIENT, choose Automatic (DHCP) option and save it.
- Select DHCP-CLIENT and wait for ip-allocation.

Steps to install nmap:

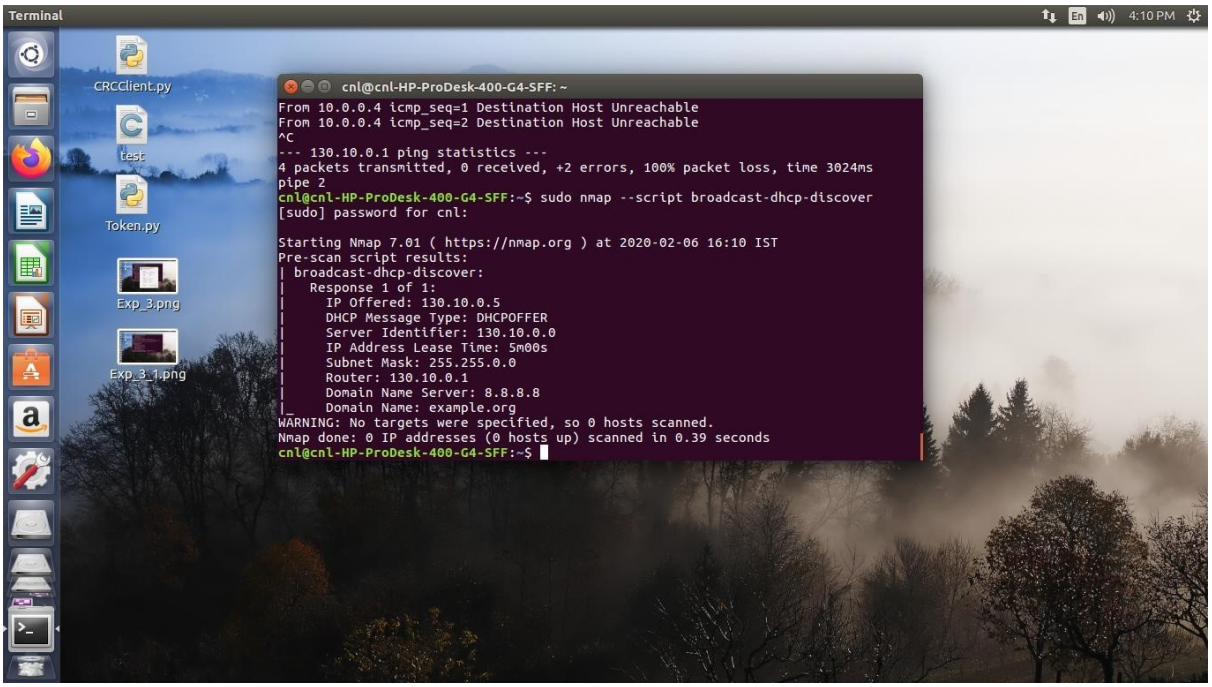
- Install nmap using command *sudo apt install nmap.*
- For the message communication use the command sudo nmap --script broadcast-dhcp-discover.

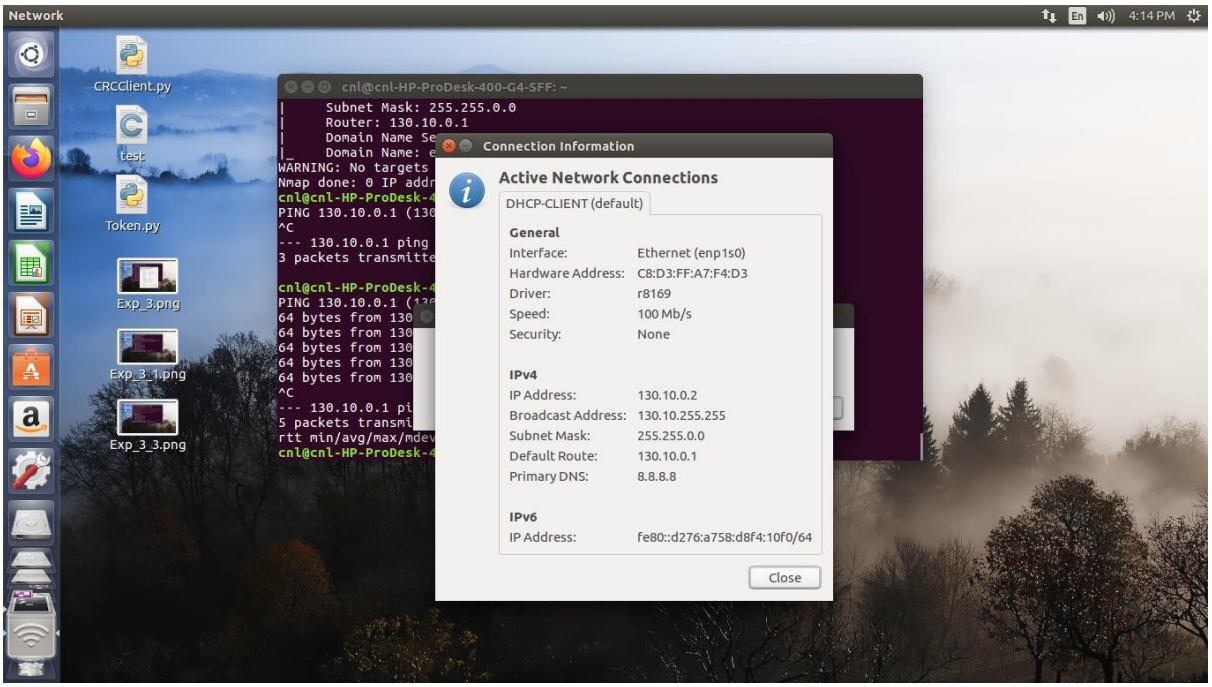
Observation 1:



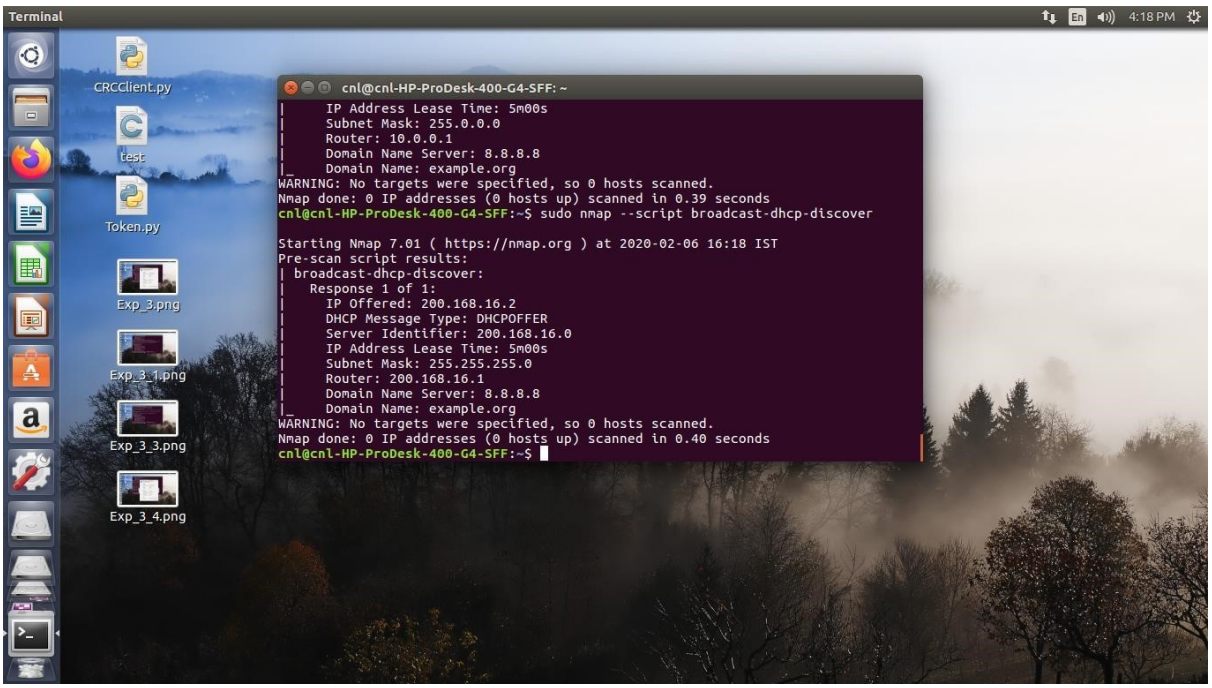


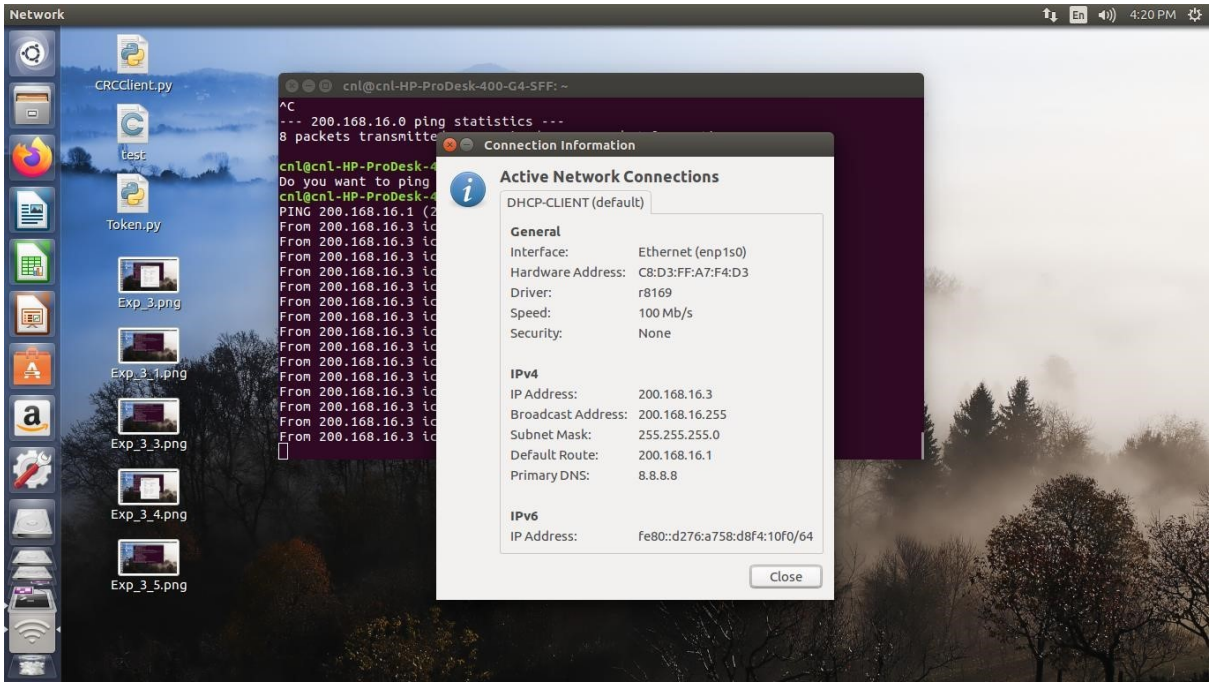
Observation 2:





Observation 3:





Learning Outcome : We have learn how to setup DHCP server in ubuntu.

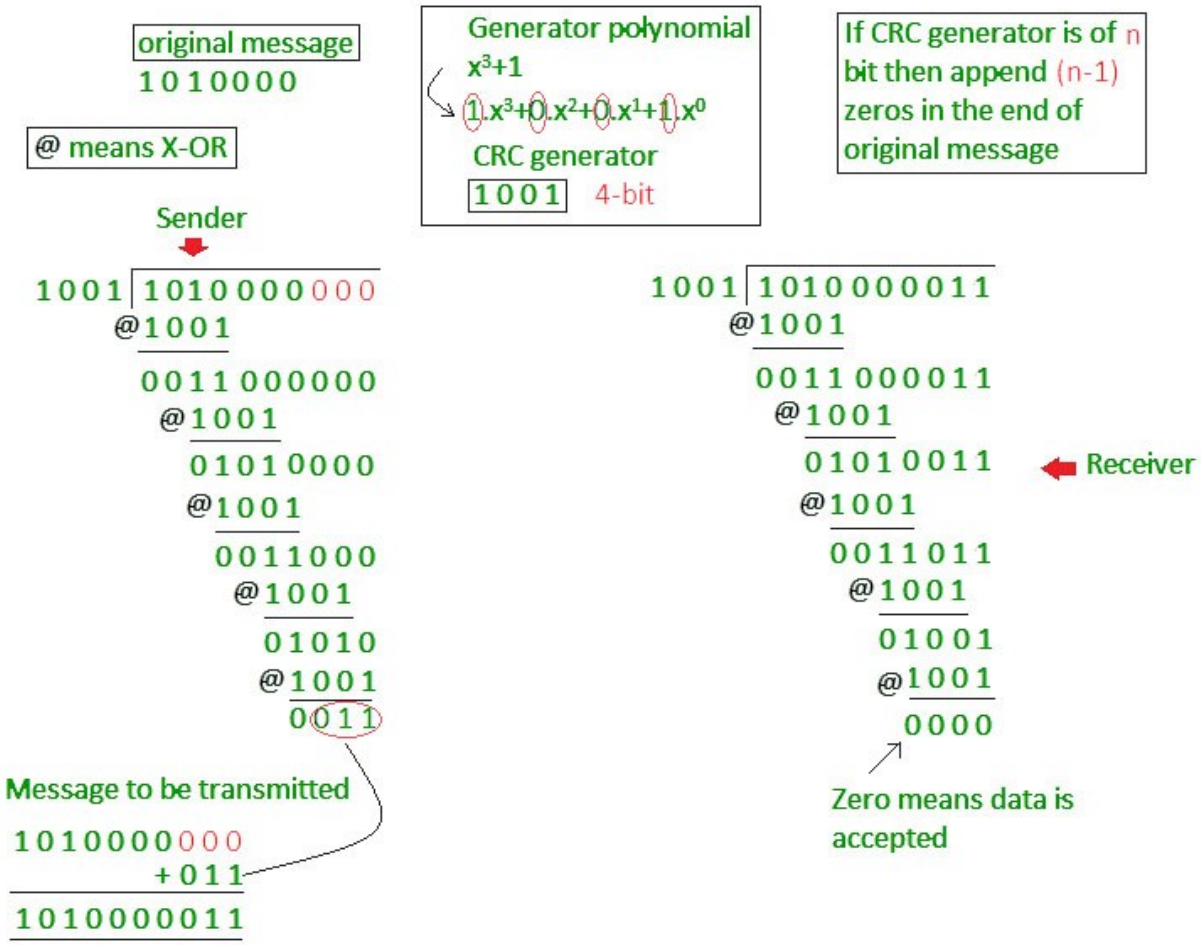
Experiment - 4

Aim : Implementation of error detectation methods like checksum,CRC (Cyclic Redundancy Check) using python programming.

Introduction :

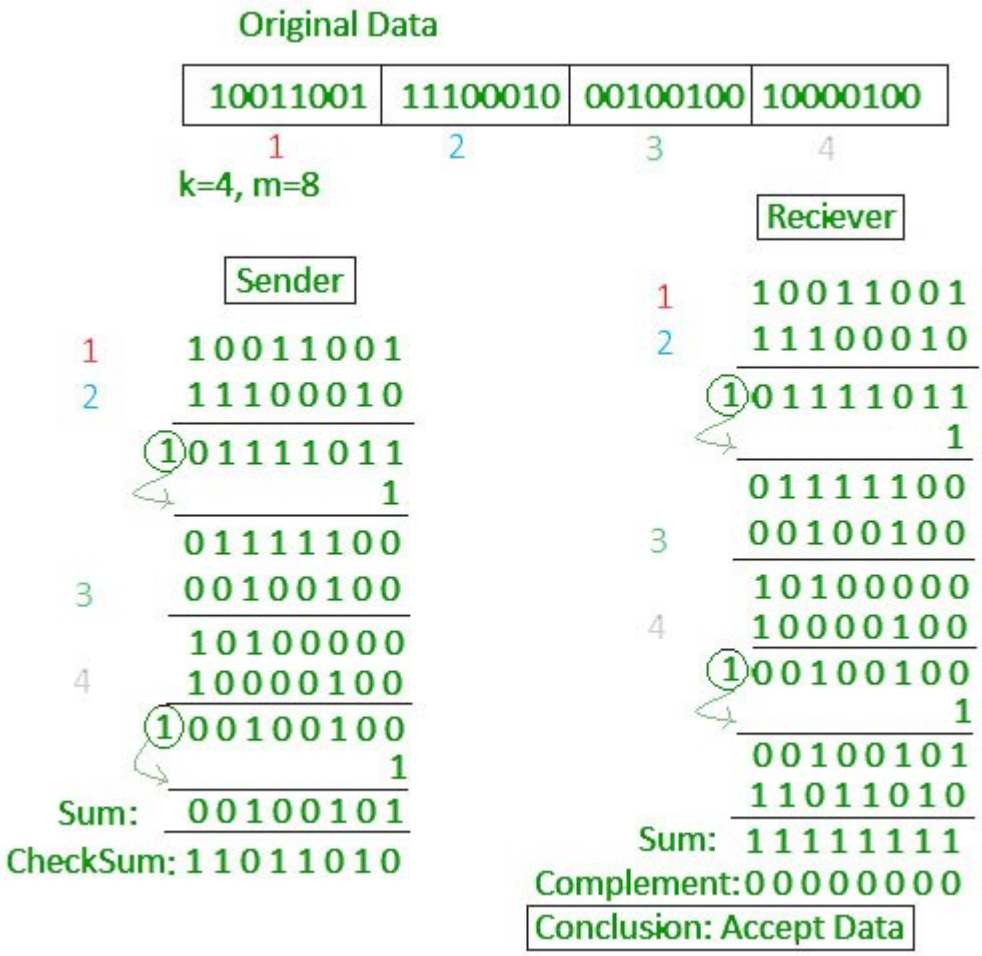
1. CRC (Cyclic Redundancy Check) :

   CRC or Cyclic Redundancy Check is a method of detecting accidental changes/errors in the communication channel. CRC uses Generator Polynomial which is available on both sender and receiver sides. An example generator polynomial is of the form like x^3 + 1. This generator polynomial represents key 1001. Another example is x^2 + x. that represents key 110.



2. Checksum :

Checksum is an error-detecting technique that can be applied to message of any length. It is used mostly at the network and transport layers of the TCP/IP protocol suite.
Here, we have considered decimal data that is being sent by the sender to the receiver using socket programming. The number of segments that the data is divided into here depends on the length of data being sent. If the length of data being sent is ‘x’, then the number of segments is also ‘x’, implying that each segment has single data. Here, we basically deal with decimal data. The concept will be consistent for string data as well because each character of the string can be represented by its equivalent ASCII code, hence again leaving us with decimal data.



Implementation :

1. CRC Codes :

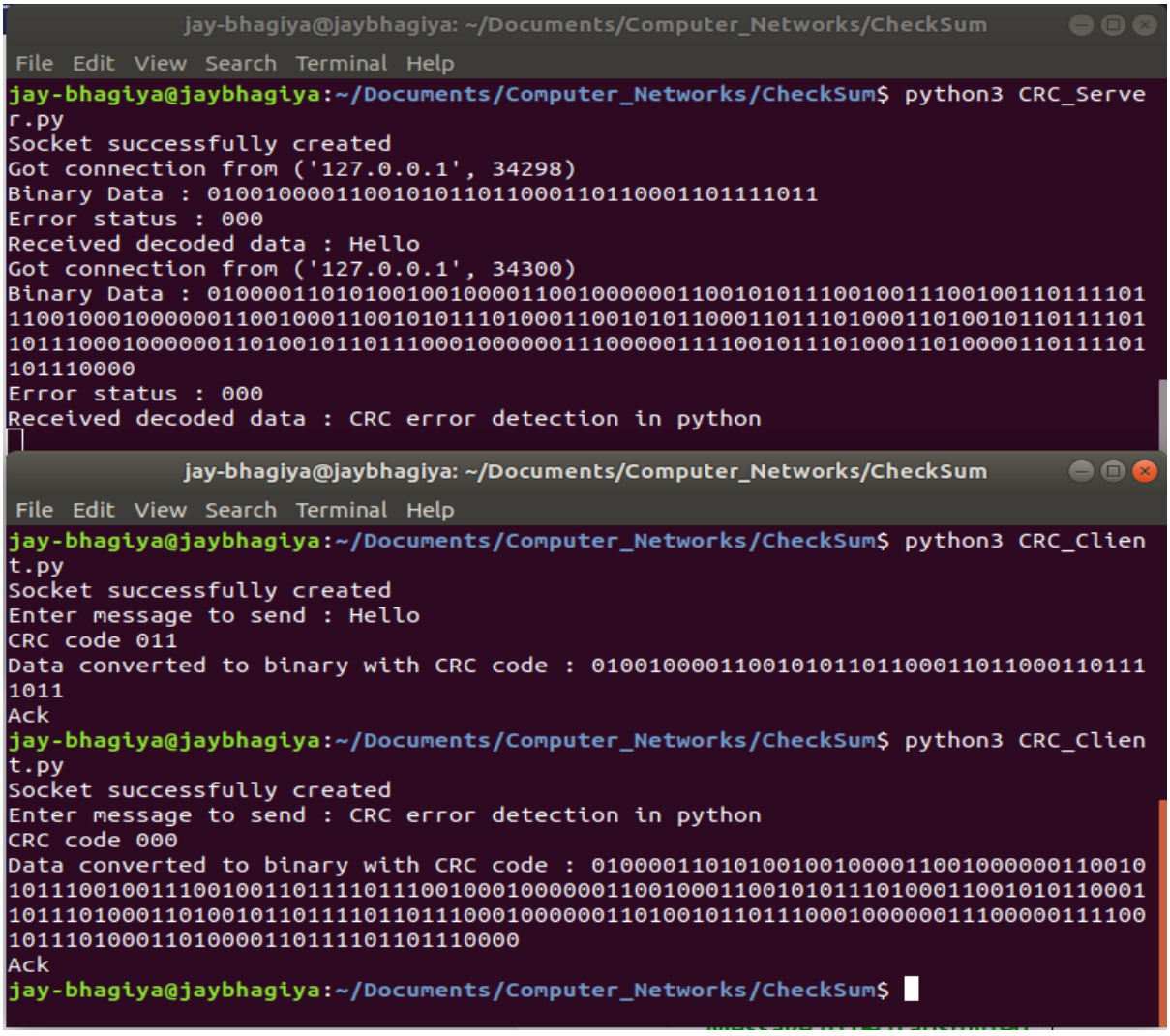
   CRC-Client.py
   CRC-Server.py

2. Checksum Codes :

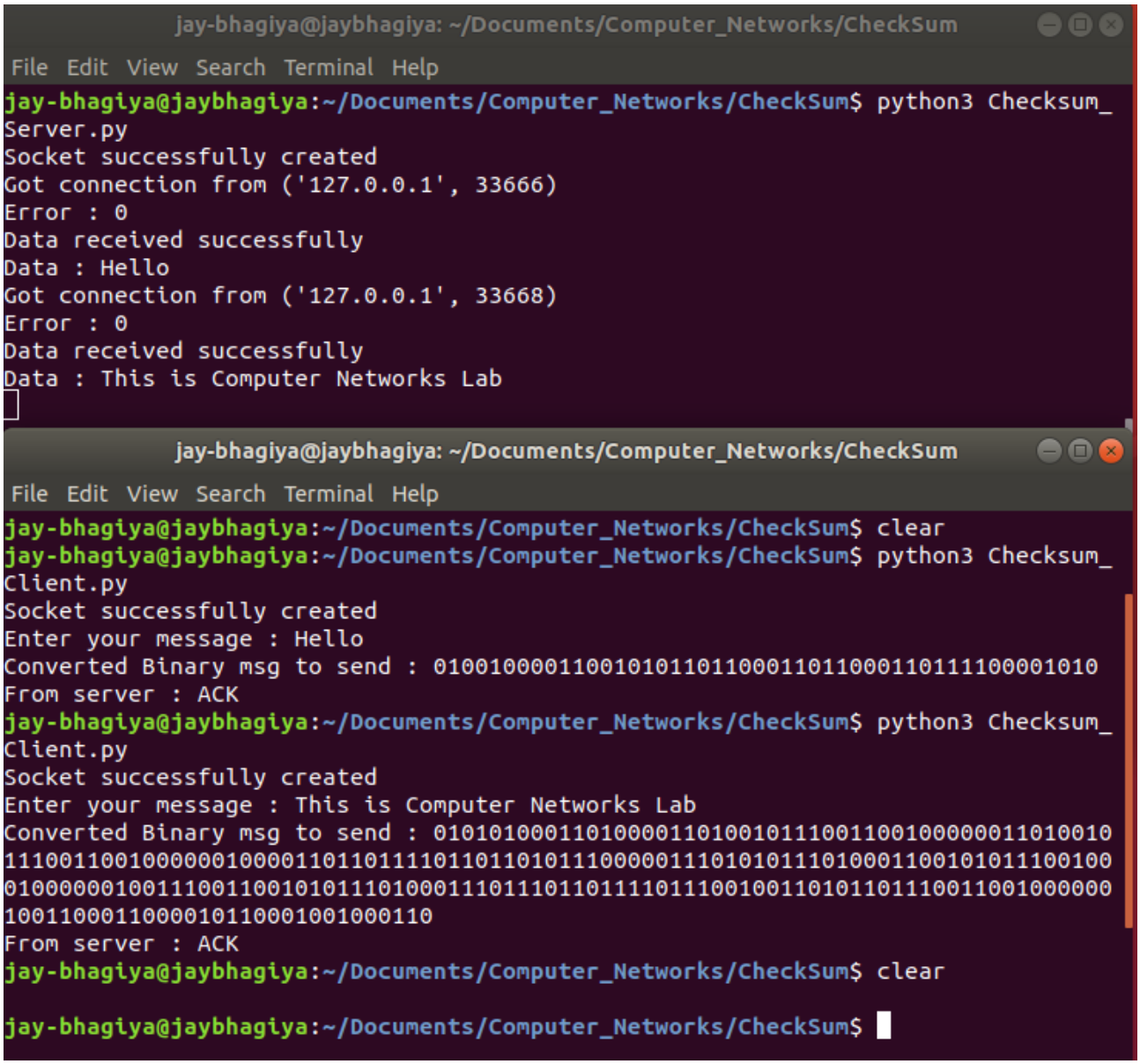
   Checksum-Client.py
   Checksum-Server.py

Output :

CRC Output :



Checksum Output :



Learning Outcome : We have learn how to implement error detection methods like CRC and Checkum in python.

Experiment – 5

Aim : Study socket programming and perform TCP socket using python/C/C++/Java

Introduction :

There are two types of Internet Protocol (IP) traffic. They are **TCP** or **Transmission Control Protocol** and **UDP** or **User Datagram Protocol**. TCP is connection oriented – once a connection is established, data can be sent bidirectional.

A TCP connection is established via three way handshake, which is a process of initiating and acknowledging a connection. Once the connection is established data transfer can begin. After transmission, the connection is terminated by closing of all established virtual circuits.

Data Transfer Features

TCP ensures a reliable and ordered delivery of a stream of bytes from user to server or vice versa.

Reliability

TCP is more reliable since it manages message acknowledgment and retransmissions in case of lost parts. Thus there is absolutely no missing data.

Ordering

TCP transmissions are sent in a sequence and they are received in the same sequence. In the event of data segments arriving in wrong order, TCP reorders and delivers application.

Connection

TCP is a heavy weight connection requiring three packets for a socket connection and handles congestion control and reliability.

Method of transfer

TCP reads data as a byte stream and message is transmitted to segment boundaries.

Error Detection

TCP uses both error detection and error recovery. Errors are detected via checksum and if a packet is erroneous, it is not acknowledged by the receiver, which triggers a retransmission by the sender. This operating mechanism is called Positive Acknowledgement with Retransmission (PAR).

Usage

TCP is suited for applications that require high reliability, and transmission time is relatively less critical.

Implementation :

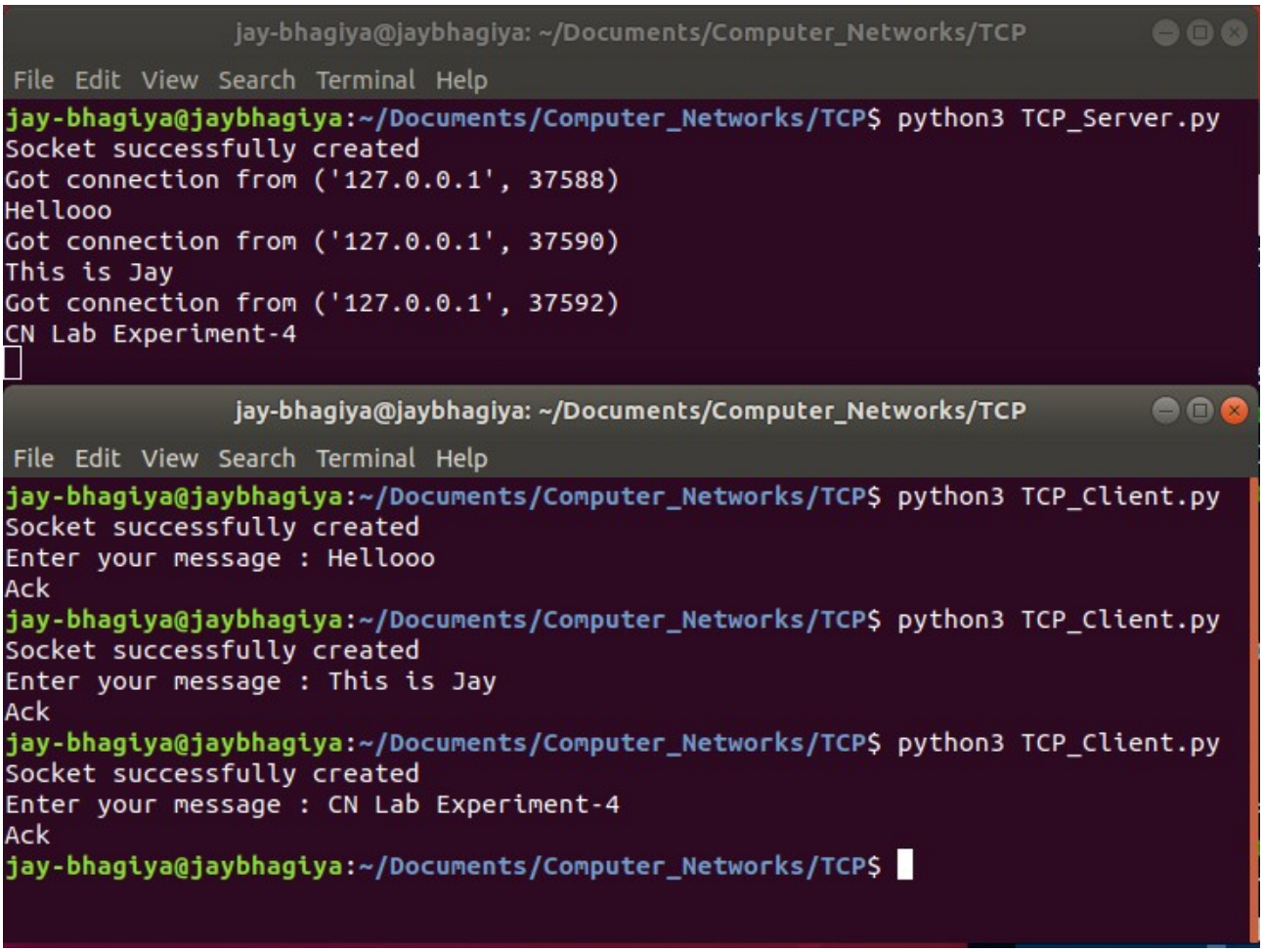
Code for TCP and UDP :

TCP-Client.py

TCP-Server.py

Output :

TCP Client-Server :



Learning Outcome :

We have learn about TCP and implemented using socket programming in python.

Experiment – 6

Aim : Study socket programming and perform UDP socket using python/C/C++/Java

Introduction :

UDP uses a simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagrams may arrive out of order, appear duplicated, or go missing without notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Unlike TCP, UDP is compatible with packet broadcasts (sending to all on local network) and multicasting (send to all subscribers).

Data Transfer Features

UDP is not dedicated to end to end connections and communication does not check readiness of receiver.

Reliability

UDP does not ensure that communication has reached receiver since concepts of acknowledgment, time out and retransmission are not present.

Ordering

UDP, sent message sequence may not be maintained when it reaches receiving application. There is absolutely no way of predicting the order in which message will be received.

Connection

UDP is a lightweight transport layer designed atop an IP. There are no tracking connections or ordering of messages.

Method of transfer

UDP messages are packets which are sent individually and on arrival are checked for their integrity. Packets have defined boundaries while data stream has none.

Error Detection

UDP works on a "best-effort" basis. The protocol supports error detection via checksum but when an error is detected, the packet is discarded. Retransmission of the packet for recovery from that error is not attempted. This is because UDP is usually for time-sensitive applications like gaming or voice transmission. Recovery from the error would be pointless because by the time the retransmitted packet is received, it won't be of any use.

Usage

UDP is suitable for applications that need fast, efficient transmission, such as games. UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients.

Implementation :

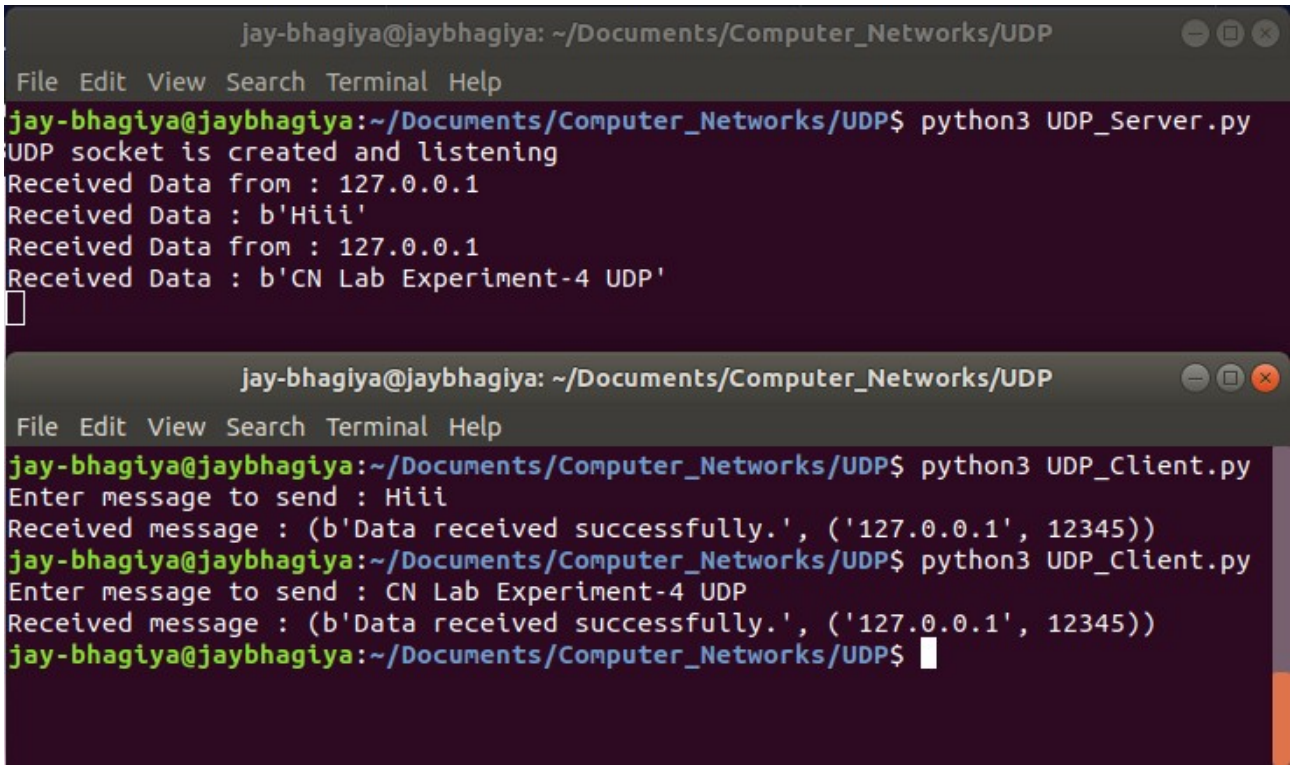
Code for UDP :

UDP-Client.py

UDP-Server.py

Output :

TCP Client-Server :



Learning Outcome :

We have learn about UDP and implemented using socket programming in python.

Experiment - 7

Aim : Implement stop-and-wait ARQ, Go-back-N ARQ and Selective-Repeat using python.

Introduction :

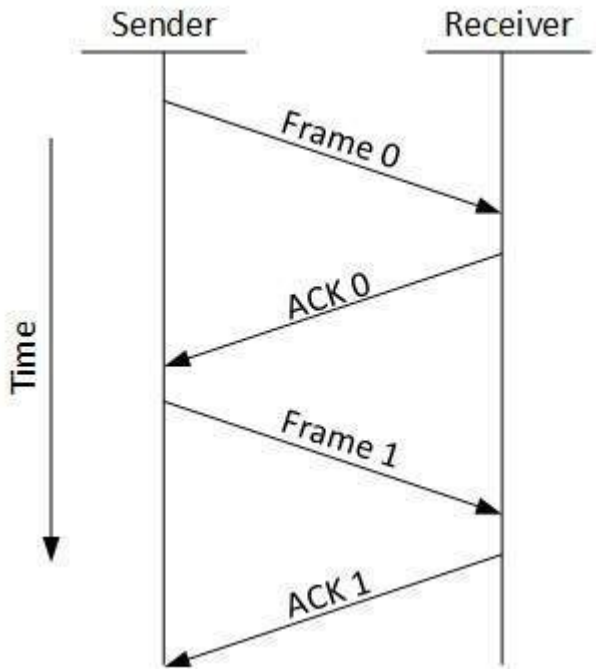
Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Two types of mechanisms can be deployed to control the flow:

Stop and Wait



This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.

Sliding Window

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

Error detection - The sender and receiver, either both or any, must ascertain that there is some error in the transit.

Positive ACK - When the receiver receives a correct frame, it should acknowledge it.

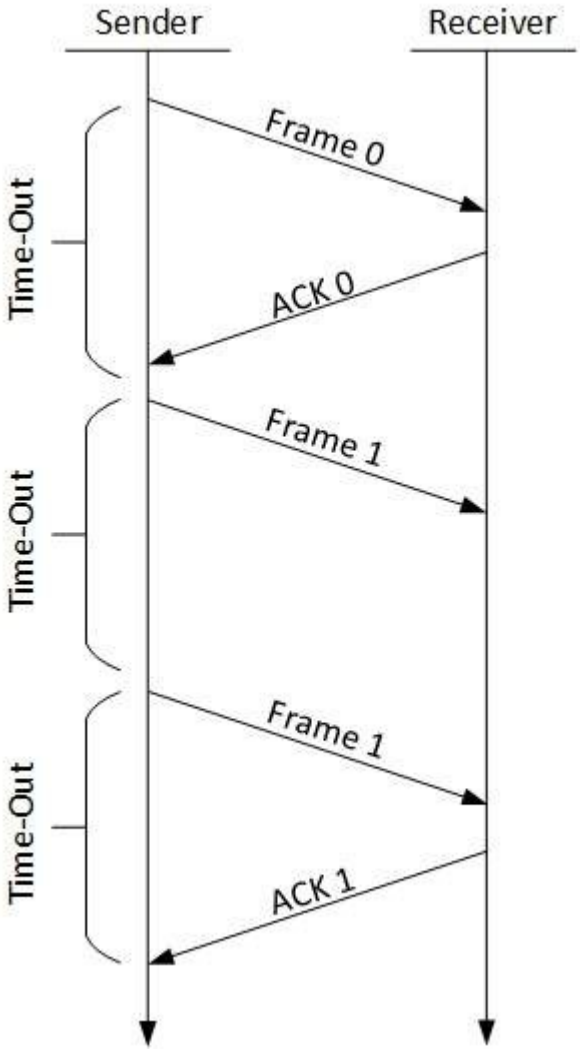
Negative ACK - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.

Retransmission: The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it’s acknowledgement is lost in transit.

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

1. Stop-and-wait ARQ
2. Go-back-N ARQ
3. Selective Repeat ARQ

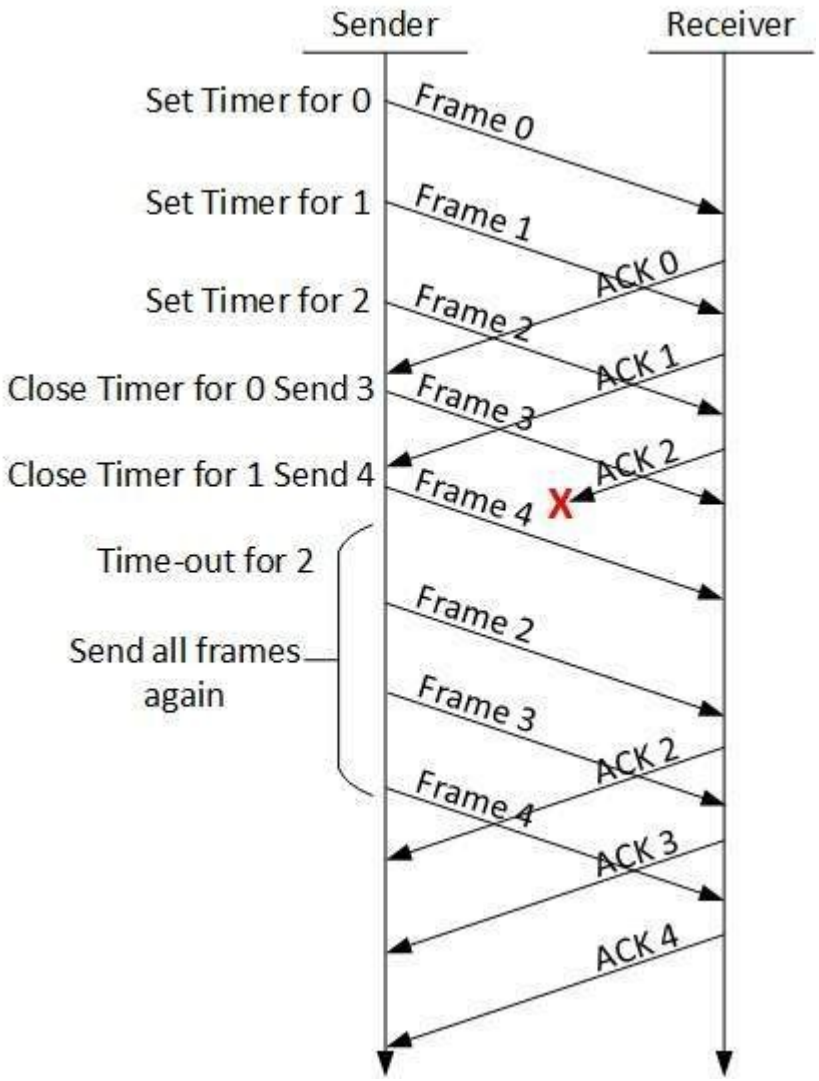
Stop-and-wait ARQ



- The following transition may occur in Stop-and-Wait ARQ:
- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.

Go-Back-N ARQ

Stop and wait ARQ mechanism does not utilize the resources at their best.When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window.



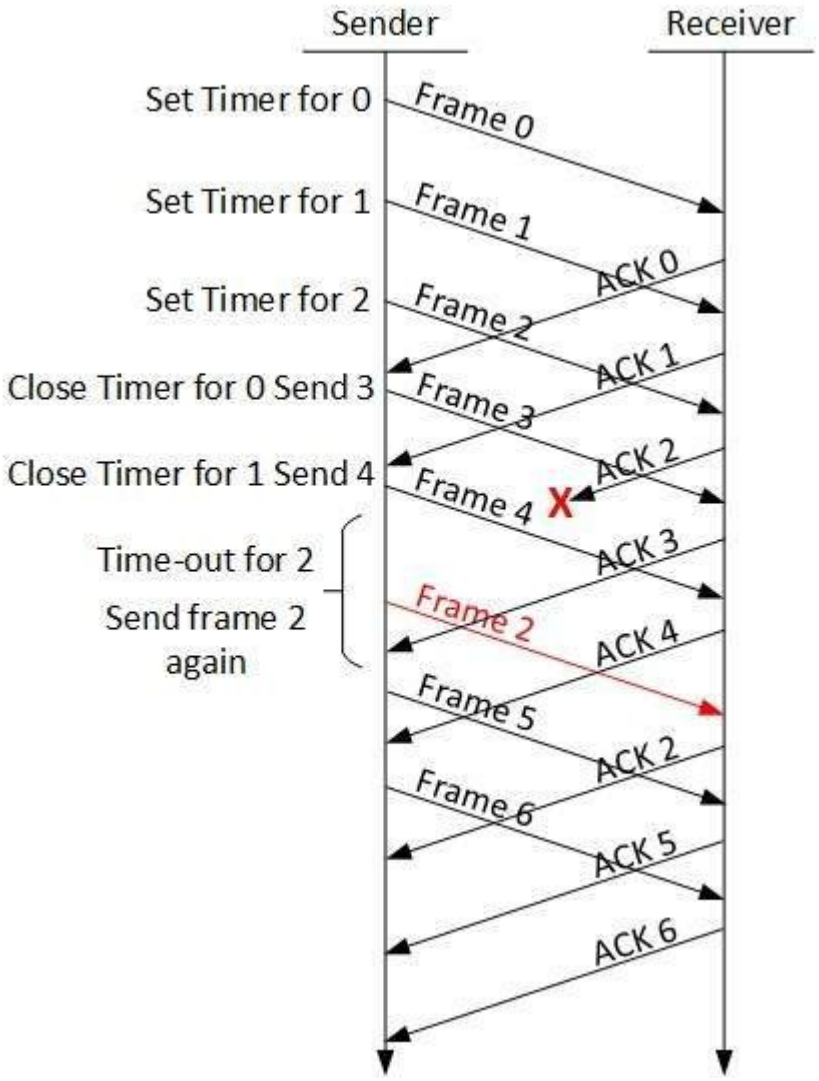
The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a

particular frame, it retransmits all the frames after which it does not receive any positive ACK.

Selective Repeat ARQ

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.



In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.

Implementation :

How to run :

For Client, follow this command :

python <source file> <port num> <protocol type> [<window size>]

i.e. GBN => python 12345 GBN

SR => python 12345 SR 5

For Server, follow this command :

python <source file> <config filename> <port num> <num of packets>

i.e. GBN => python GBN.txt 12345 5

SR => python SR.txt 12345 5

Go-Back-N and Selective Repeat using python :

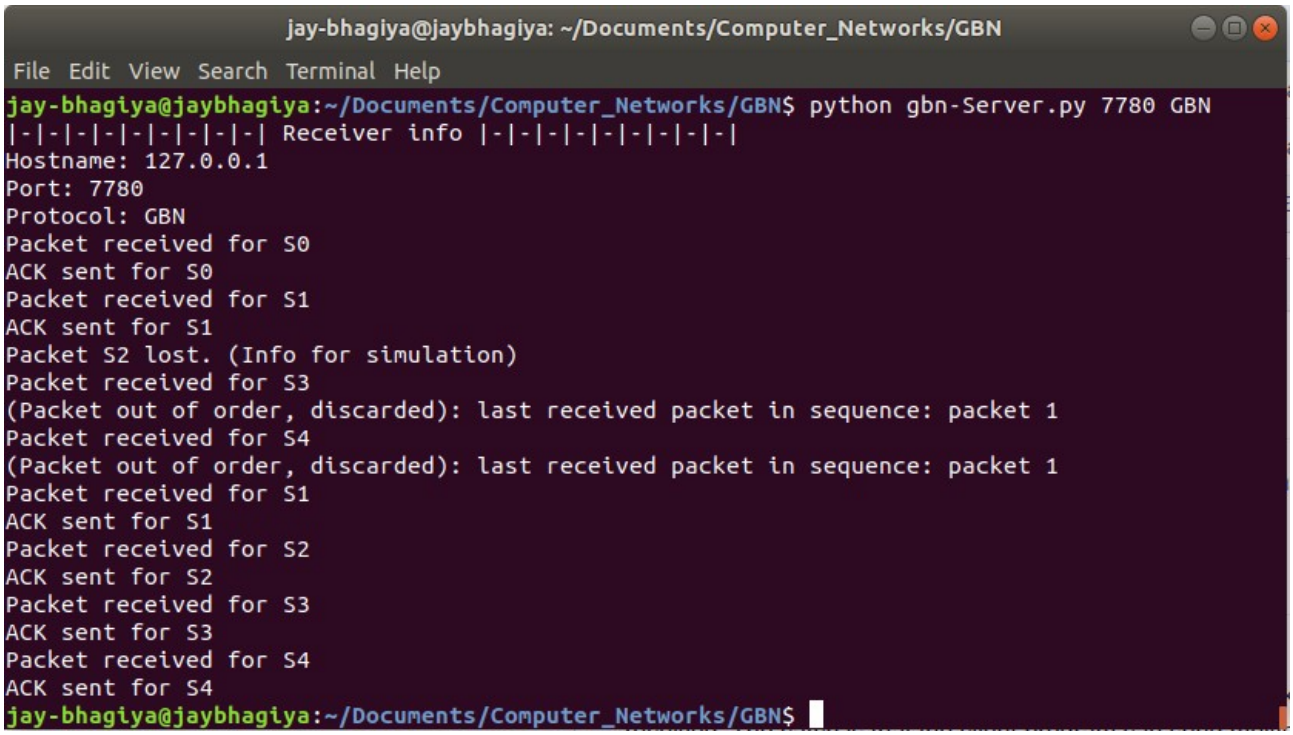
- GBN-SR-Client.py
- GBN-SR-Server.py

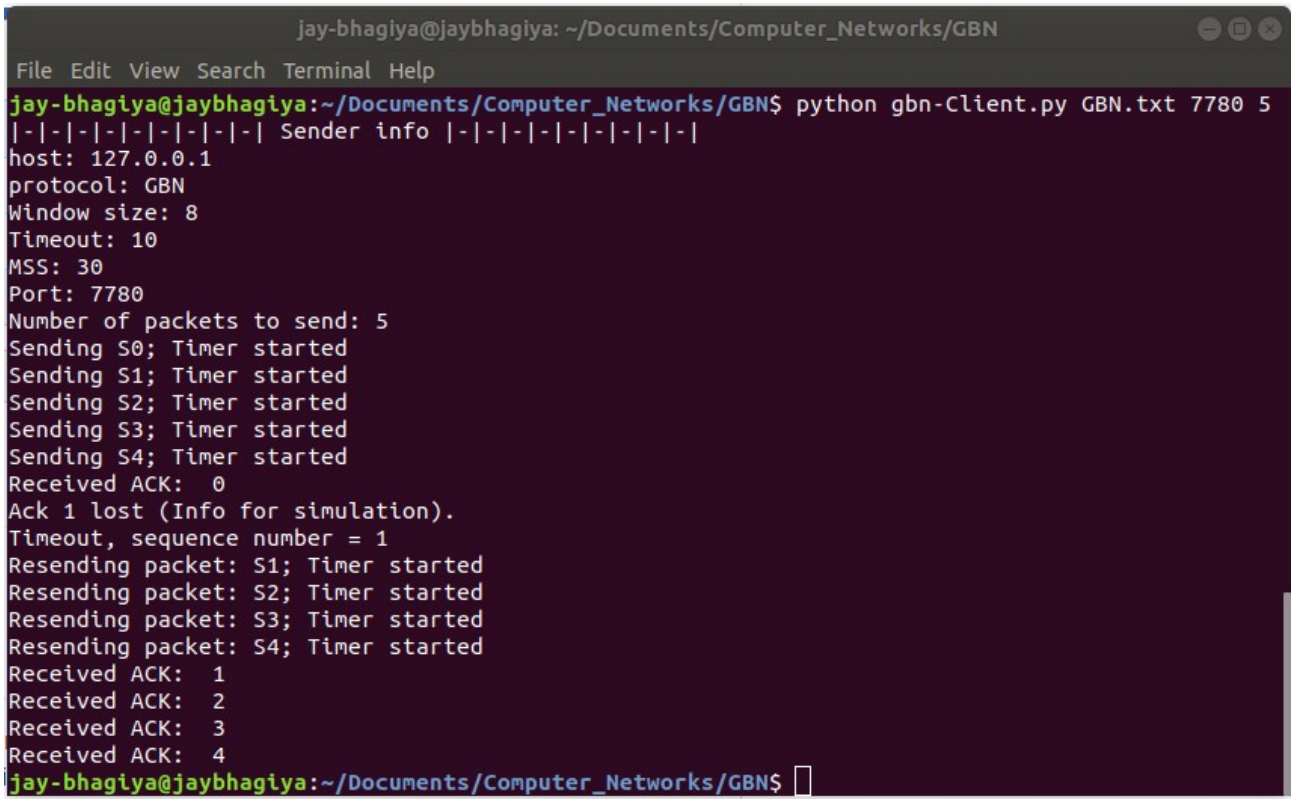
Configuration file :

- GBN.txt
- SR.txt

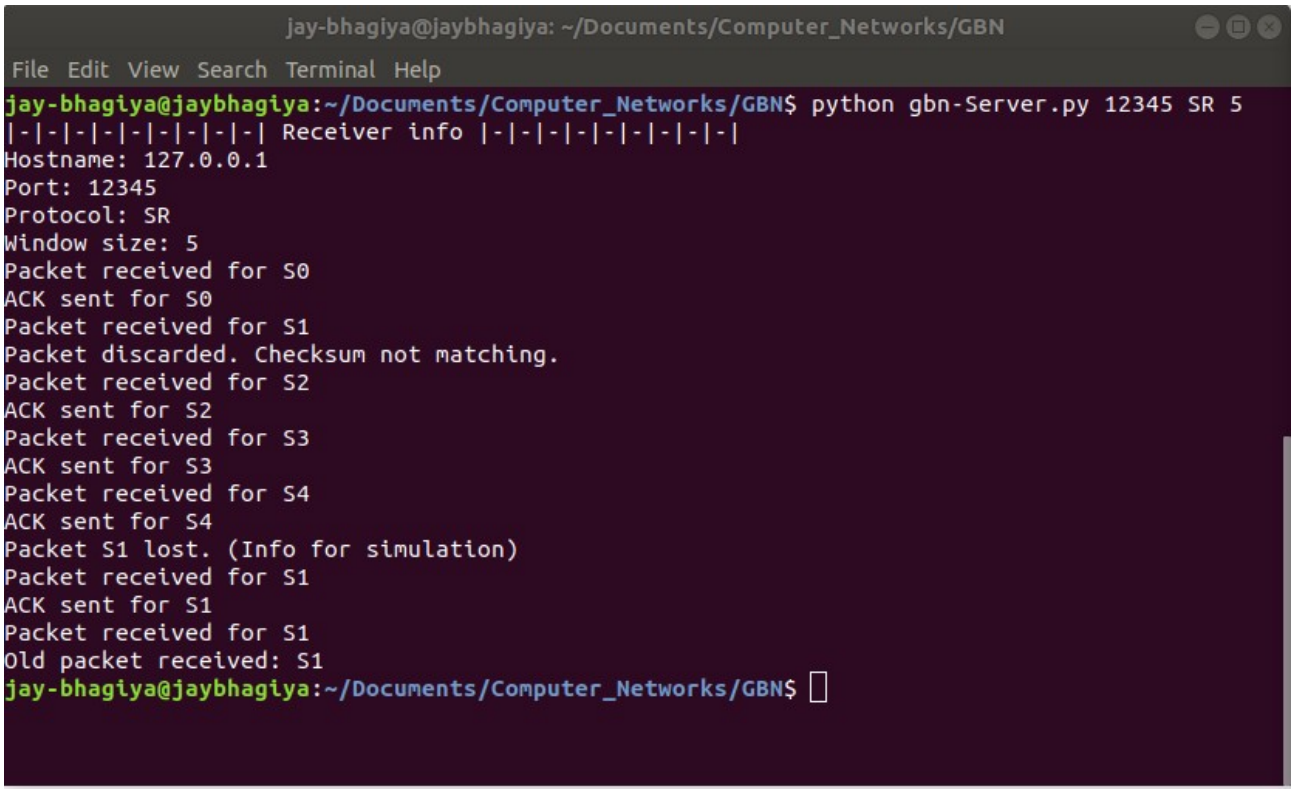
Output :

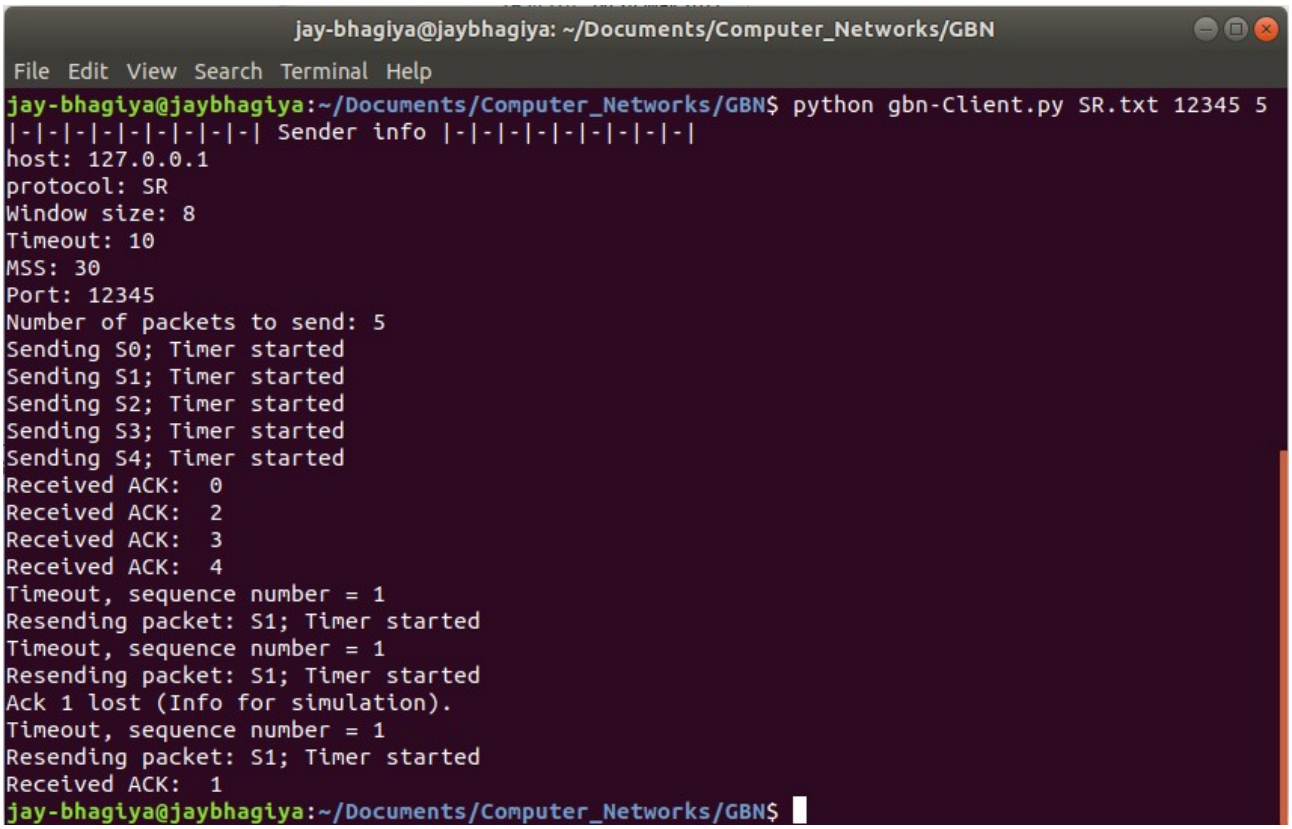
GBN Output :





SR Output :





Learning Outcome : We have learn types of flow control and implemented them using pyhton.

Experiment - 8

Aim : Study cisco packet tracer simulation and simulate DHCP / DNS / EMAIL / HTTP / FTP server.

Introduction: Packet Tracer is a cross-platform visual simulation tool that allows users to create network topologies and imitate modern computer networks.Packet Tracer provides simulation, visualization, authoring , assessment, and collaboration capabilities and facilitates the teaching and learning of complex technology concepts. Packet Tracer supports an array of simulated Application Layer protocols, as well as basic routing with RIP, OSPF, EIGRP, BGP, to the extents required by the current CCNA curriculum. Packet Tracer also supports the Border Gateway Protocol. The simulation-based learning environment helps students develop 21st century skills such as decision making, creative and critical thinking, and problem solving.

Implementation :

1. **Open your Network Topology** - Once you've opened your Network Topology on Cisco Packet Tracer, access your network and identify the components of your network, for example; Servers, Routers, End Devices, etc.
2. **Complete the cabling -** Access the cables section and connect completely and correctly the cables between the network in order to ensure connectivity between the devices in the network using the connections table given.
3. **Configure the IP addresses on the end devices -** Using the address table still, correctly and completely configure the IP addresses on all end devices. This can be done by accessing the desktop platform on each device and locating the IP configuration section. The reason for doing this is to enable the devices be on the right network.
4. **Configure the IP addresses on your routers and switches -** After configuring the right IP addresses on the end devices, you will have to do the same on the routers and switches also, using the address table. But this time in a different way because there's no desktop platform on the routers and switches. You will have to access the configuration panel on both devices and this can be done in two ways:
    - Click on the device and open the Command Line Interface (CLI) and then type in the right commands to configure the right addresses for the router using the addressing table.
    - Use a console cable from an end device and connect it to the device you wish to configure and access the terminal platform on the end device and it will take you to the device's Command Line Interface and then you type in the commands in other to configure the right addresses.

5. **Configure your default gateway -** After configuring the IP addresses, you will need to configure the default gateway also. The reason for this is so the end devices would know what network they are operating on. You can find the default gateway either in the addressing table (if given) or in the network topology.
6. **Test connectivity -** After configuring the addresses, you will have to test connectivity by opening a command prompt window on the end devices and try pinging the address which the network operates on. If it gives you a reply, it means your network was configured correctly.

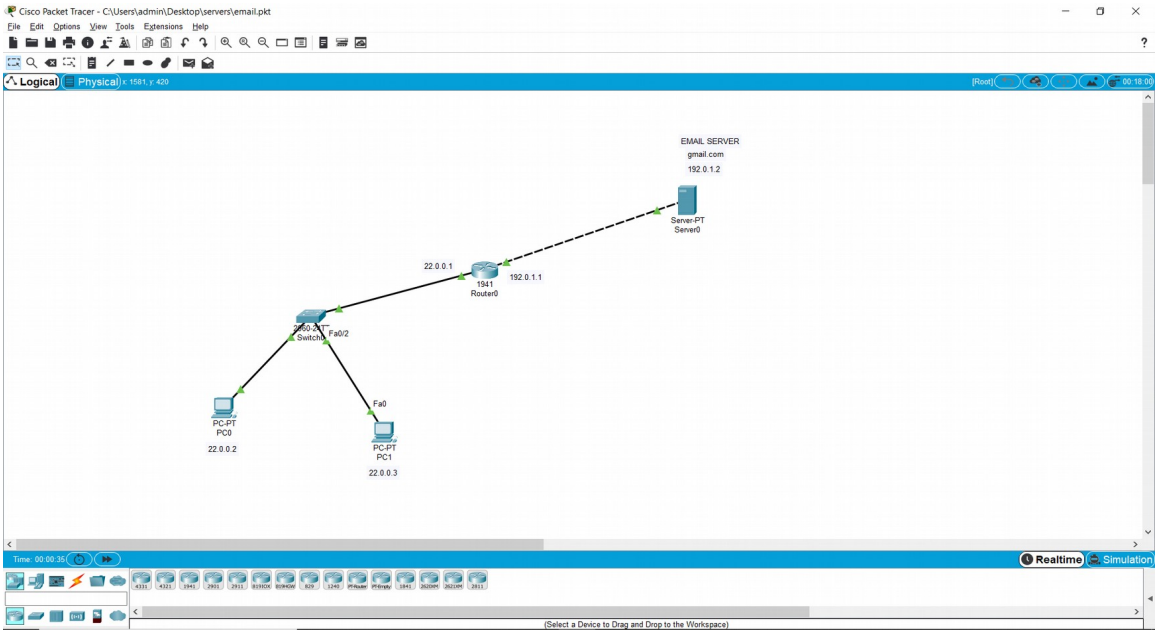
Files :

CPT pkt file can found here also :

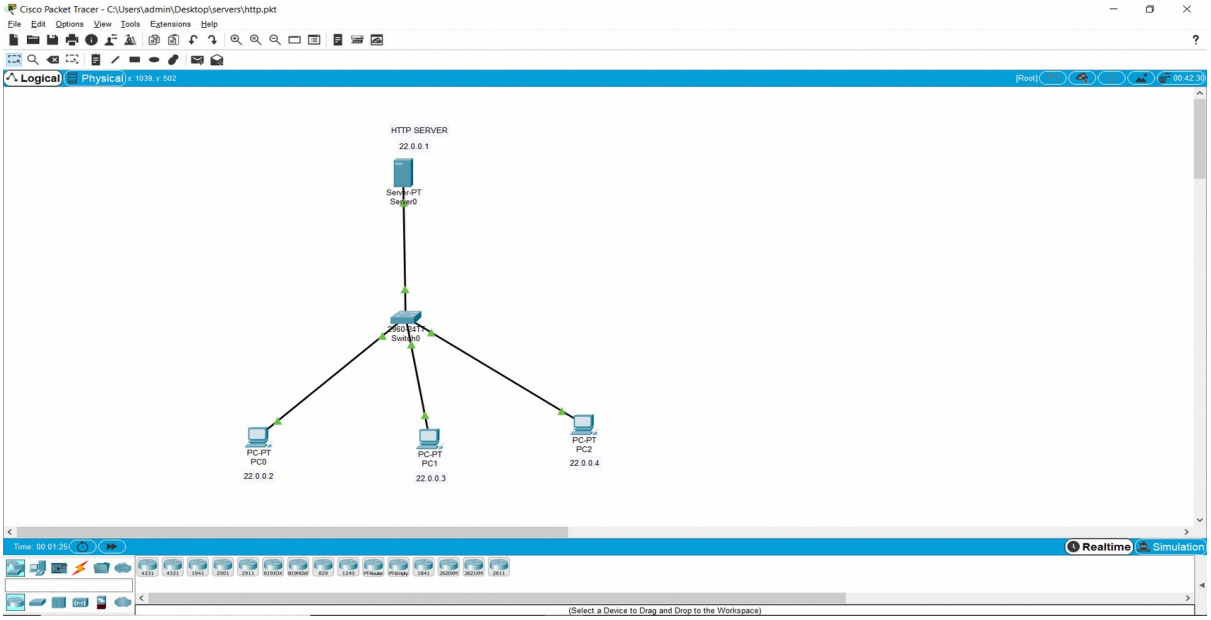
CPT Files

Output :

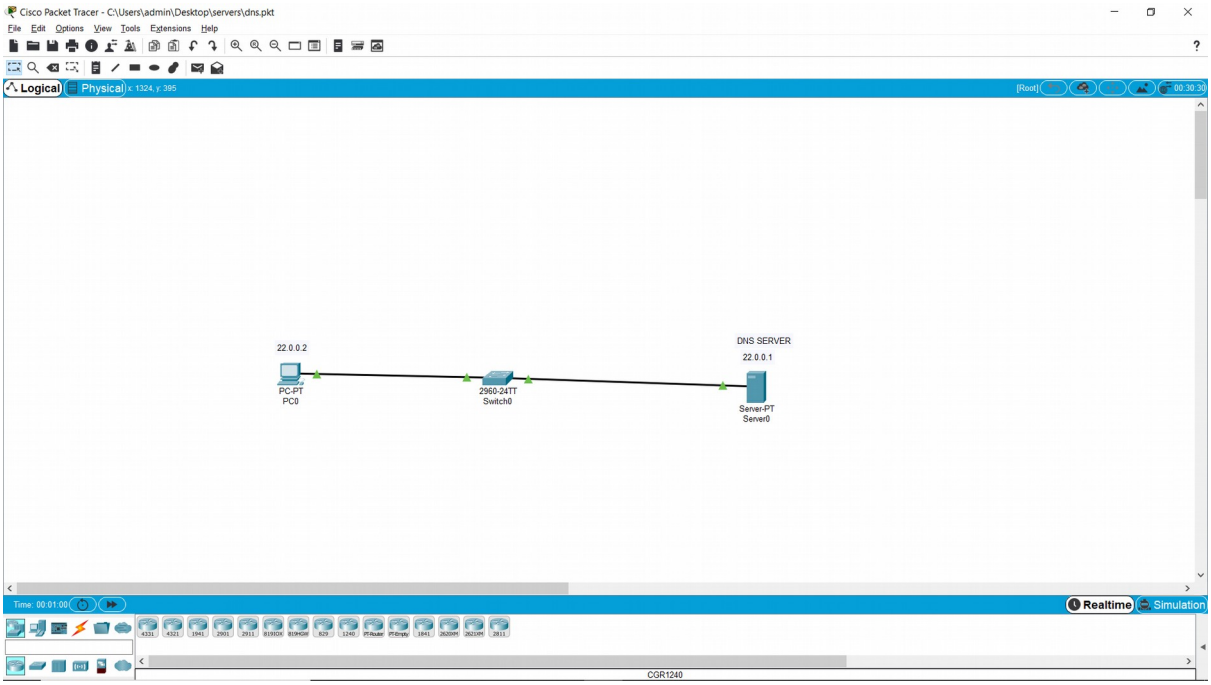
1. EMAIL:



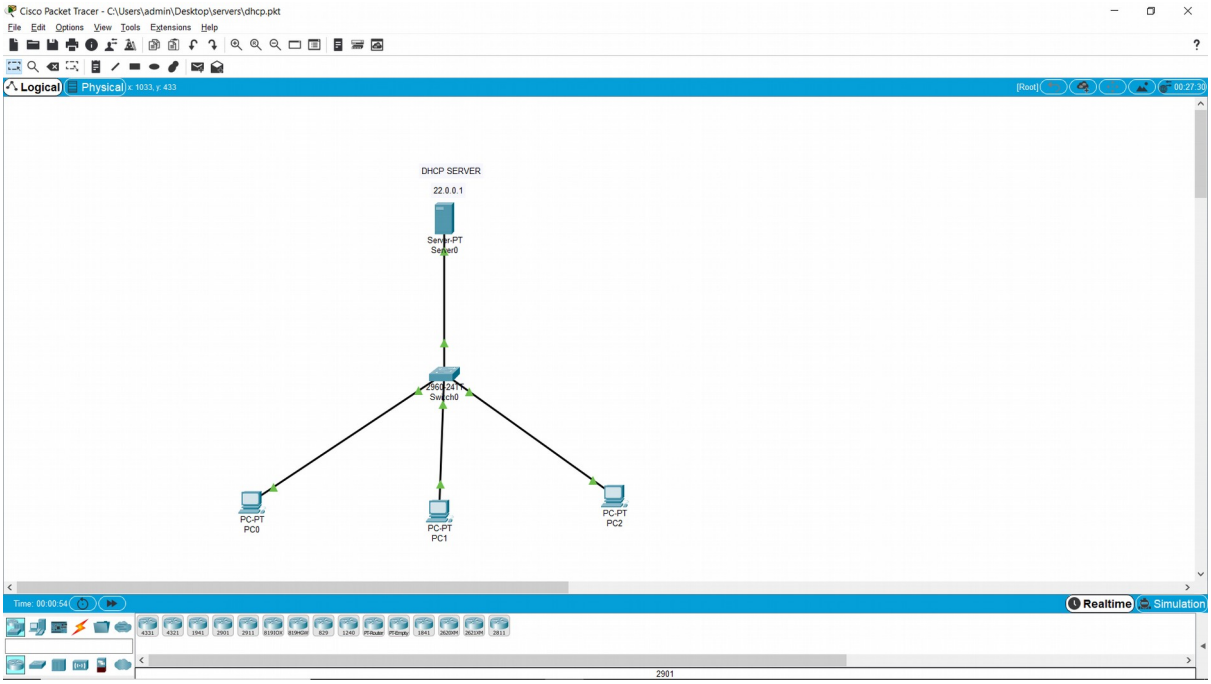
2. HTTP:


Cisco Packet Tracer - C:\Users\admin\Desktop\servers\http.pkt
File Edit Options View Tools Extensions Help
Logical Physical
HTTP SERVER
22.0.0.1
Server-PT
Server0
2960-24TT
Switch0
PC-PT
PC0
22.0.0.2
PC-PT
PC1
22.0.0.3
PC-PT
PC2
22.0.0.4
Realtime Simulation
(Select a Device to Drag and Drop to the Workspace)

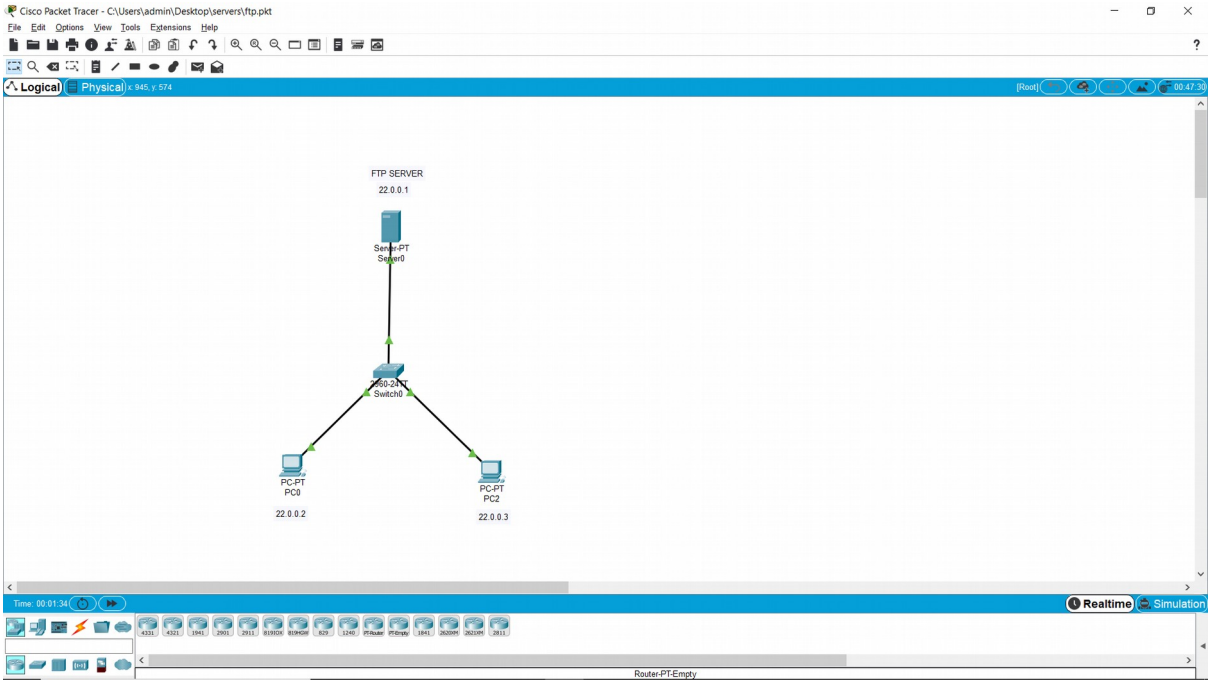
3. DNS:


Cisco Packet Tracer - C:\Users\admin\Desktop\servers\dns.pkt
File Edit Options View Tools Extensions Help
Logical Physical
22.0.0.2
PC-PT
PC0
2960-24TT
Switch0
DNS SERVER
22.0.0.1
Server-PT
Server0
Realtime Simulation
CGR1240

4. DHCP:



5. FTP:



Learning Outcome :

We have learn about cisco packet tracer and simulation of DHCP/DNS/EMAIL/HTTP/FTP server.

Experiment - 9

Aim : study cisco packet tracer simulation and simulate static and dynamic routing

Introduction :

Routing is the process of selecting a path for traffic in a network or between or across multiple networks. Broadly, routing is performed in many types of networks, including circuit-switched networks, such as the public switched telephone network (PSTN), and computer networks, such as the Internet. To route IP packets, a host or a router has a routing table with entries for each destination or a combination of destinations. A static routing table contains information, which is entered manually. The administrator enters the route for each destination into the table. Dynamic routing table is updated periodically by using dynamic protocols like RIP, OSEP or BGP. The main function of the network is to route the packets from source to destination. More than one route is possible in every network , however the shortest route should be selected. The shortest route means, a route which passes through the least number of nodes to reach the destination. The routing algorithm is designed to find the shortest root and it is part of a network software.

Implementation :

1. Test connectivity between the PCs and the default gateway.
2. Ping between routers to test connectivity.
3. Viewing the routing tables.
4. Configure default routes on the BranchOffice and PartnerNet routers.
5. Configure static and dynamic routes at Main Office.
6. Test connectivity.

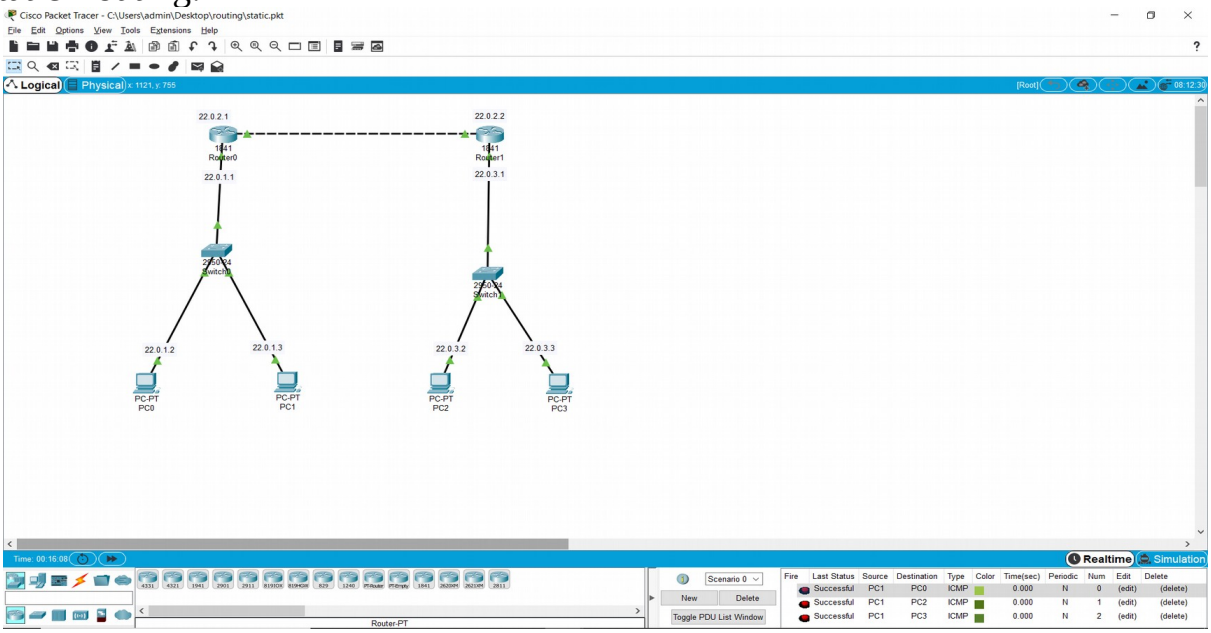
Files :

CPT Routing pkt files can be found here also

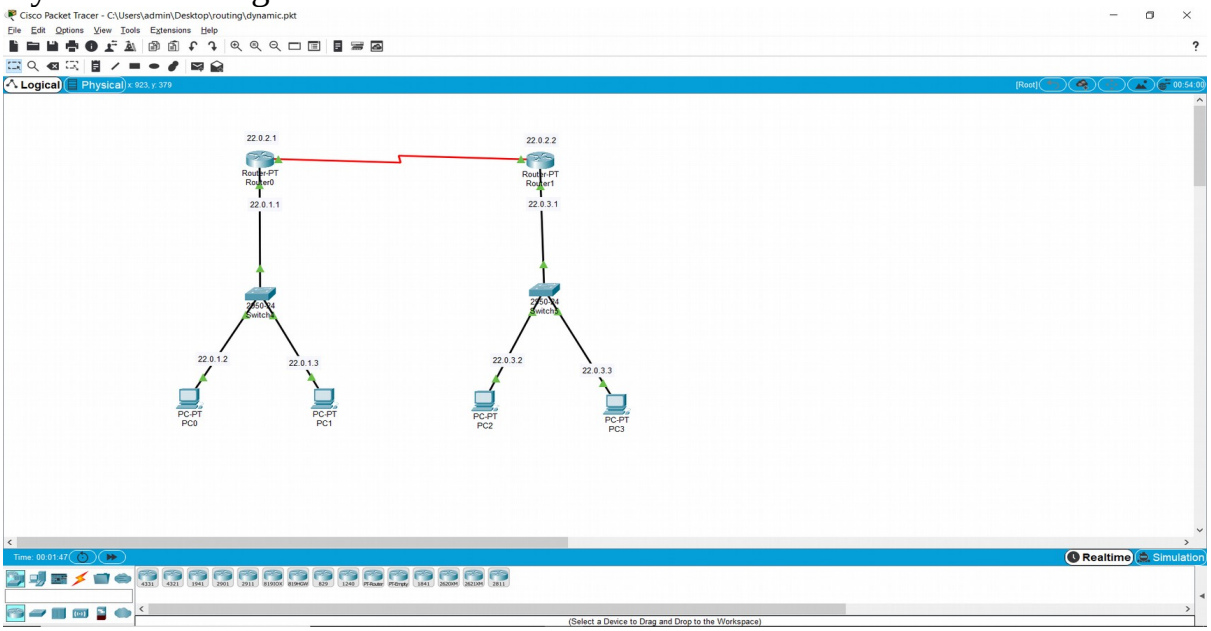
CPT Routing File

Output :

1. Static Routing:



2. Dynamic Routing:



Learning Outcome :

We have learn about cisco packet tracer simulation and simulation of static and dynamic routing.

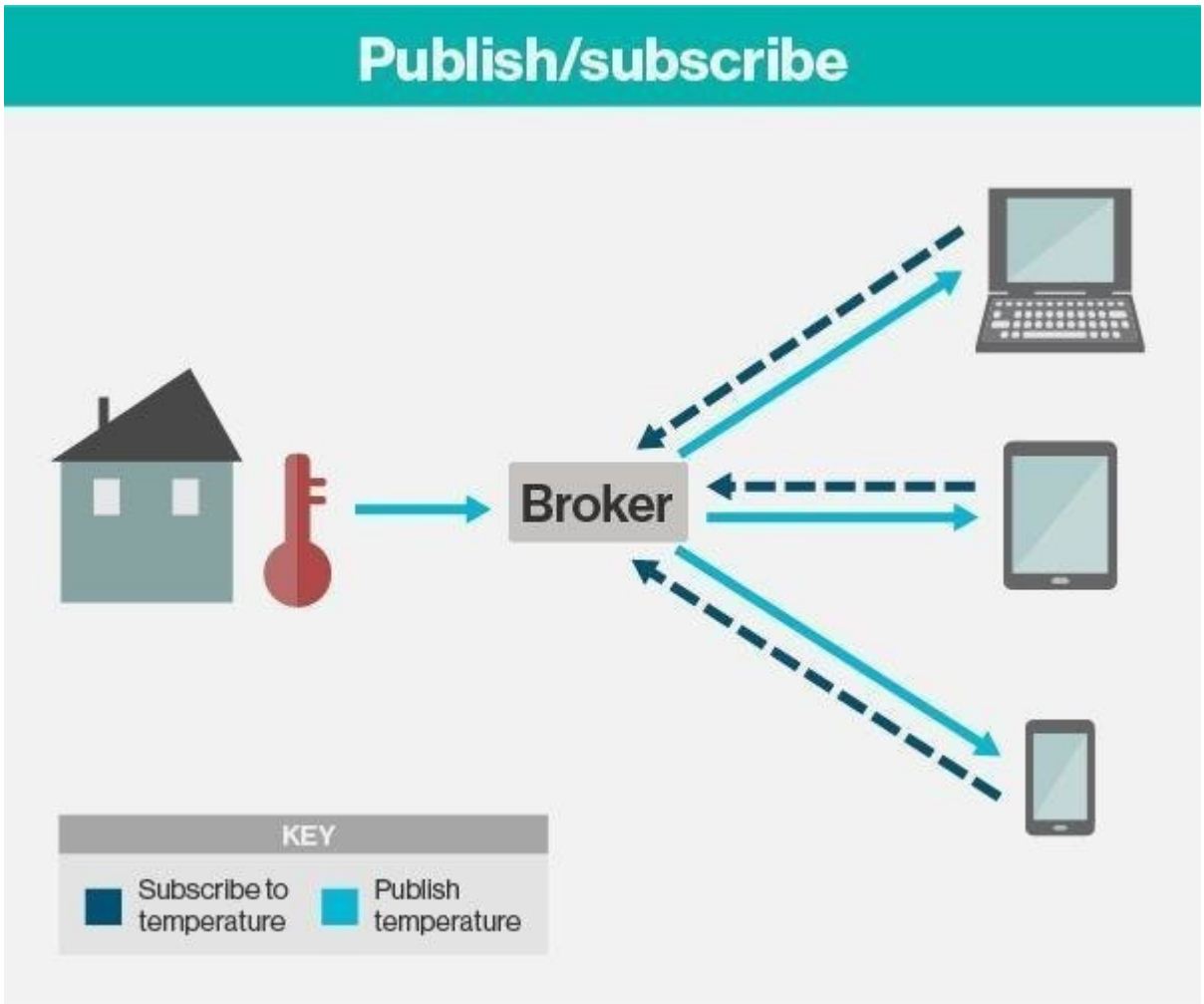
Experiment - 10

Aim : To learn about MQTT (Message Queuing Telemetry Transport) protocol.

Introduction :

MQTT (MQ Telemetry Transport)is a lightweight messaging protocol that provides resource-constrained network - clients with a simple way to distribute(telemetry)information. The protocol, which uses a publish/subscribe communication pattern, is used for machine-to-machine (M2M) communication and plays an important role in the internet of things (IoT).

The MQTT protocol surrounds two subjects: a client and a broker. An MQTT broker is a server, while the clients are the connected devices. When a device -- or client -- wants to send data to a server -- or broker-- it is called a publish. When the operation is reversed, it is called a subscribe.

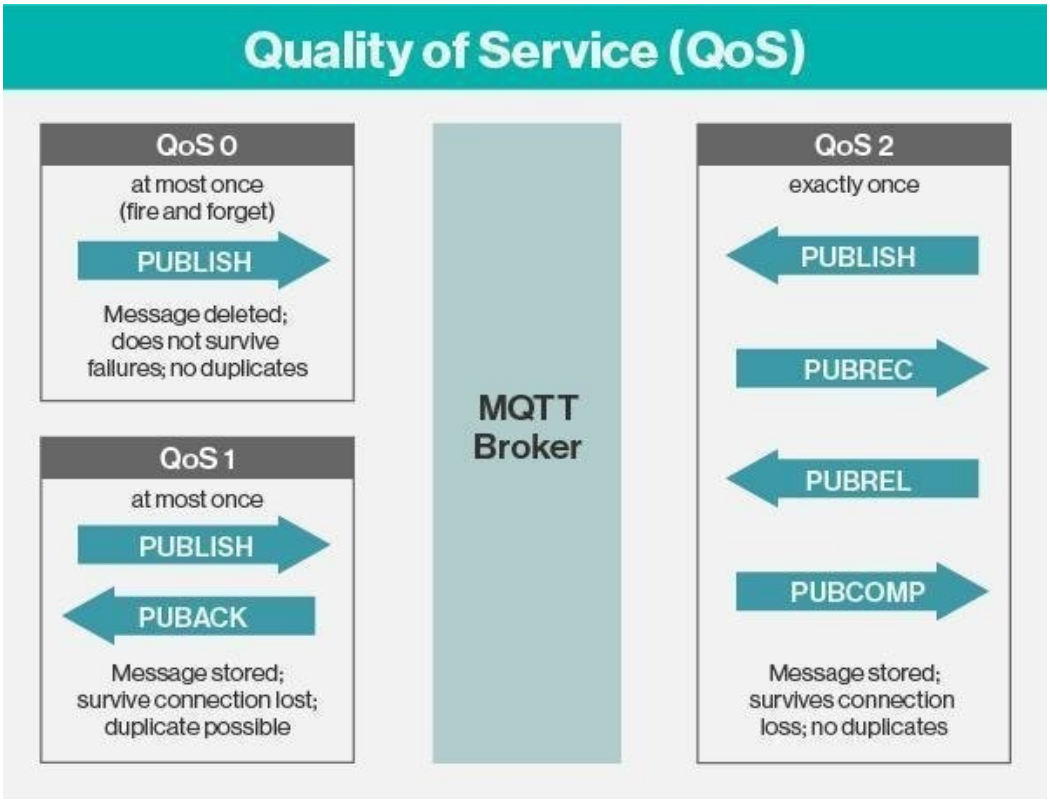


If the connection from a subscribing client to a broker is broken, then the broker will buffer messages and push them out to the subscriber when it is back online. If the connection from the publishing client to the broker is disconnected without notice, then the broker can close the connection and send subscribers a cached message with instructions from the publisher.

MQTT Message	Description
CONNECT	Client request to connect to server
CONNACK	Connect acknowledgement
PUBLISH	Publish message
PUBACK	Publish acknowledgement
PUBREC	Publish received
PUBREL	Publish release
PUBCOMP	Publish complete
SUBSCRIBE	Client subscribe request
SUBACK	Subscribe acknowledgement
UNSUBSCRIBE	Unsubscribe request
UNSUBACK	Unsubscribe acknowledgement
PINGREQ	PING request
PINGRESP	PING response
DISCONNECT	Client is disconnecting

Quality of service levels :

QoS refers to an agreement between the sender of a message and the message's recipient. QoS will define the guarantee of delivery in referring to a specific message. QoS acts as a key feature in MQTT, giving the client the ability to choose between three levels of service.



Implementation :

- Subscriber Python script.

```
import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc):
        print("Connect with result code "+str(rc))
        client.subscribe("test")

def on_message(client, userdata, msg):
        print(msg.topic+" "+str(msg.payload))

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect("192.168.43.160",1883,60)

client.loop_forever()
```

- Publisher Python script.

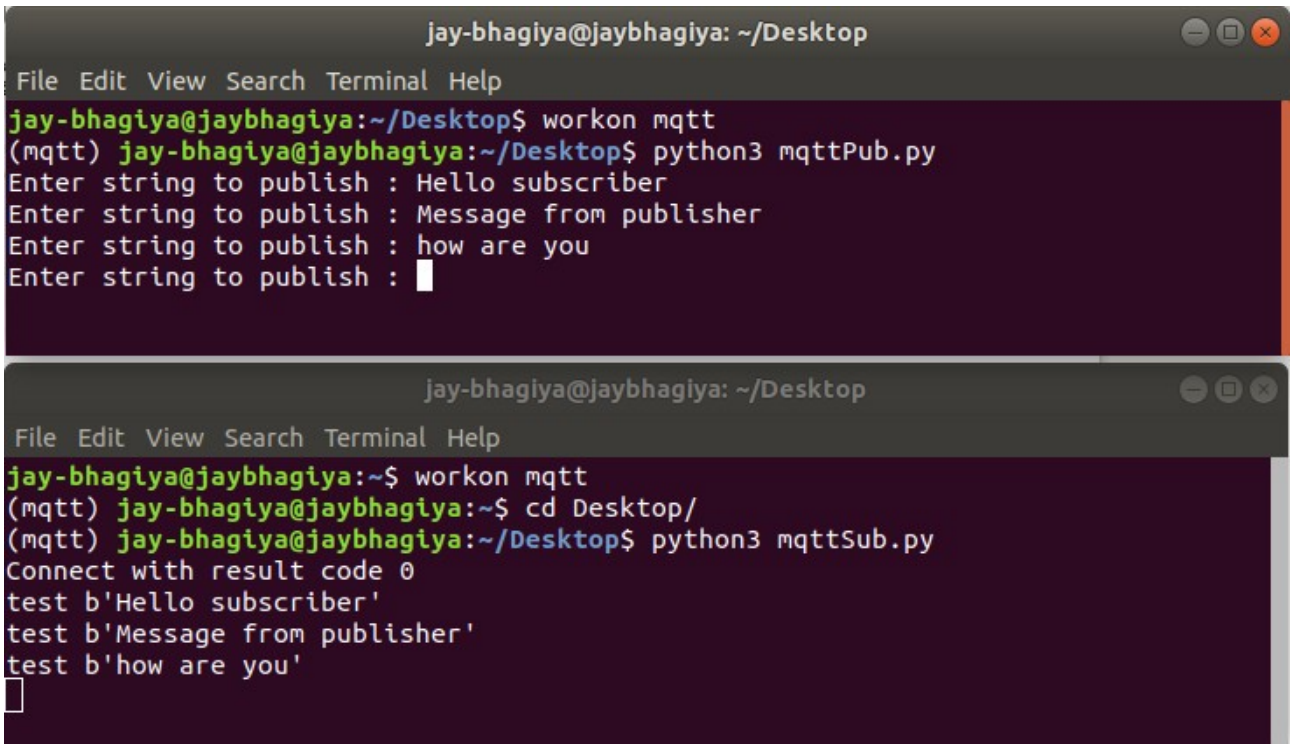
```
import paho.mqtt.client as mqtt

client = mqtt.Client()

client.connect("192.168.43.160",1883,60)

while True:
        x = input("Enter string to publish : ")
        client.publish("test", x, qos=0, retain=False)
```

Output :



Learning outcome :

we have learn about lightweight protocol named MQTT which is widely used in IoT.

Experiment - 11

Aim : Learn packet sniffing and perform packet sniffing using wireshark tool

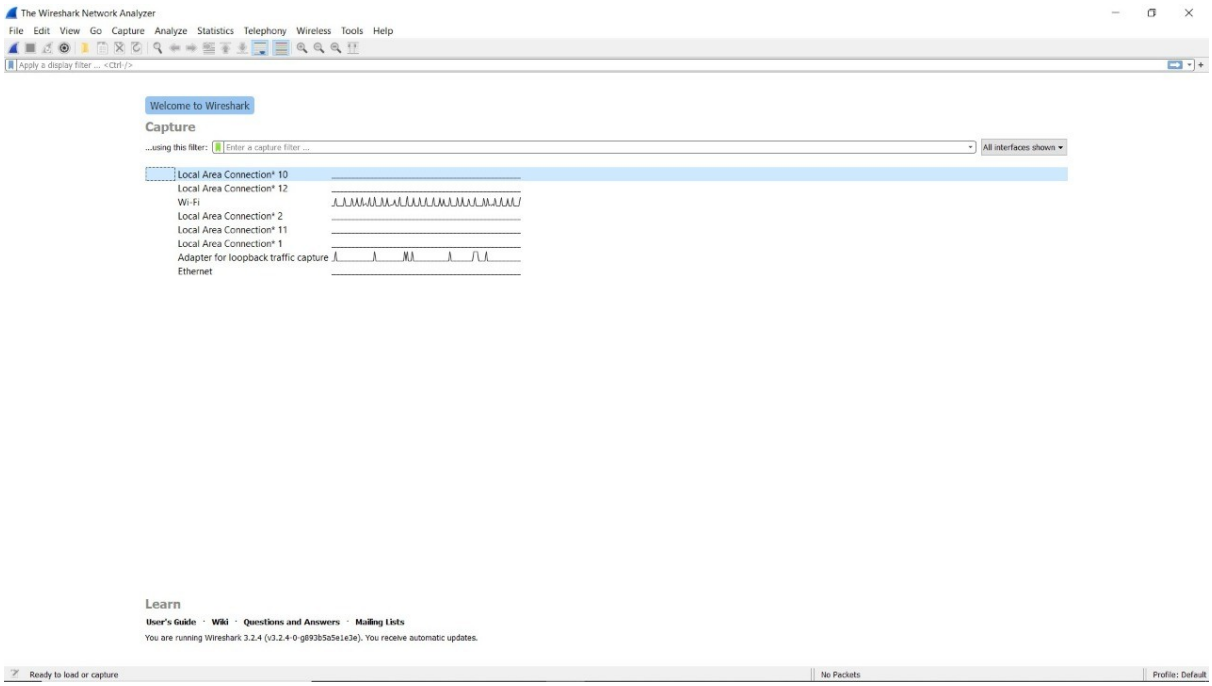
Introduction:

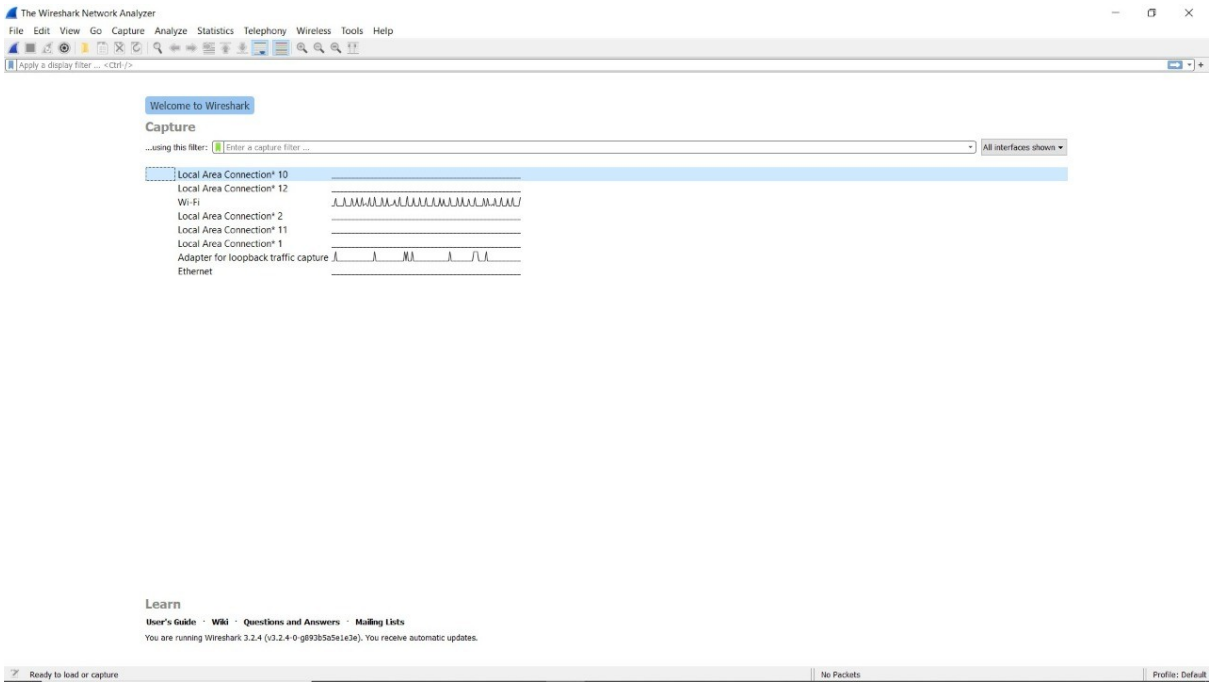
The basic tool for observing the messages exchanged between executing protocol entities is called a packet sniffer. As the name suggests, a packet sniffer captures (“sniffs”) messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/received from/by application and protocols executing on your machine.

Implementation:

- Installing and using wireshark:
- Wireshark is open source software which is available for all the platforms.
- Install Wireshark, and then open the application.
- In the top menu, select Capture > Interfaces.
- Click Start for the interface that is connected to your network.
- After the transmission has finished, navigate back in Wireshark to Capture > Stop.
- Select File > Save As and save the file.

Output :





Learning outcome: We learn about performing packet sniffing through wireshark tool.

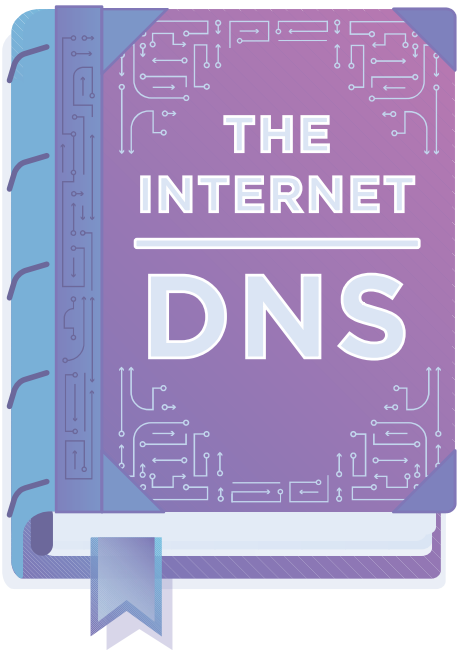
Expriment – 12

Aim : Setup DNS (Domain Name System) in ubuntu.

Introduction :

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).



The 8 steps in a DNS lookup:

1.A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.

2.The resolver then queries a DNS root nameserver (.).

3.The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information

for its domains. When searching for example.com, our request is pointed toward the .com TLD.

4.The resolver then makes a request to the .com TLD.

5.The TLD server then responds with the IP address of the domain's nameserver, example.com.

6.Lastly, the recursive resolver sends a query to the domain's nameserver.

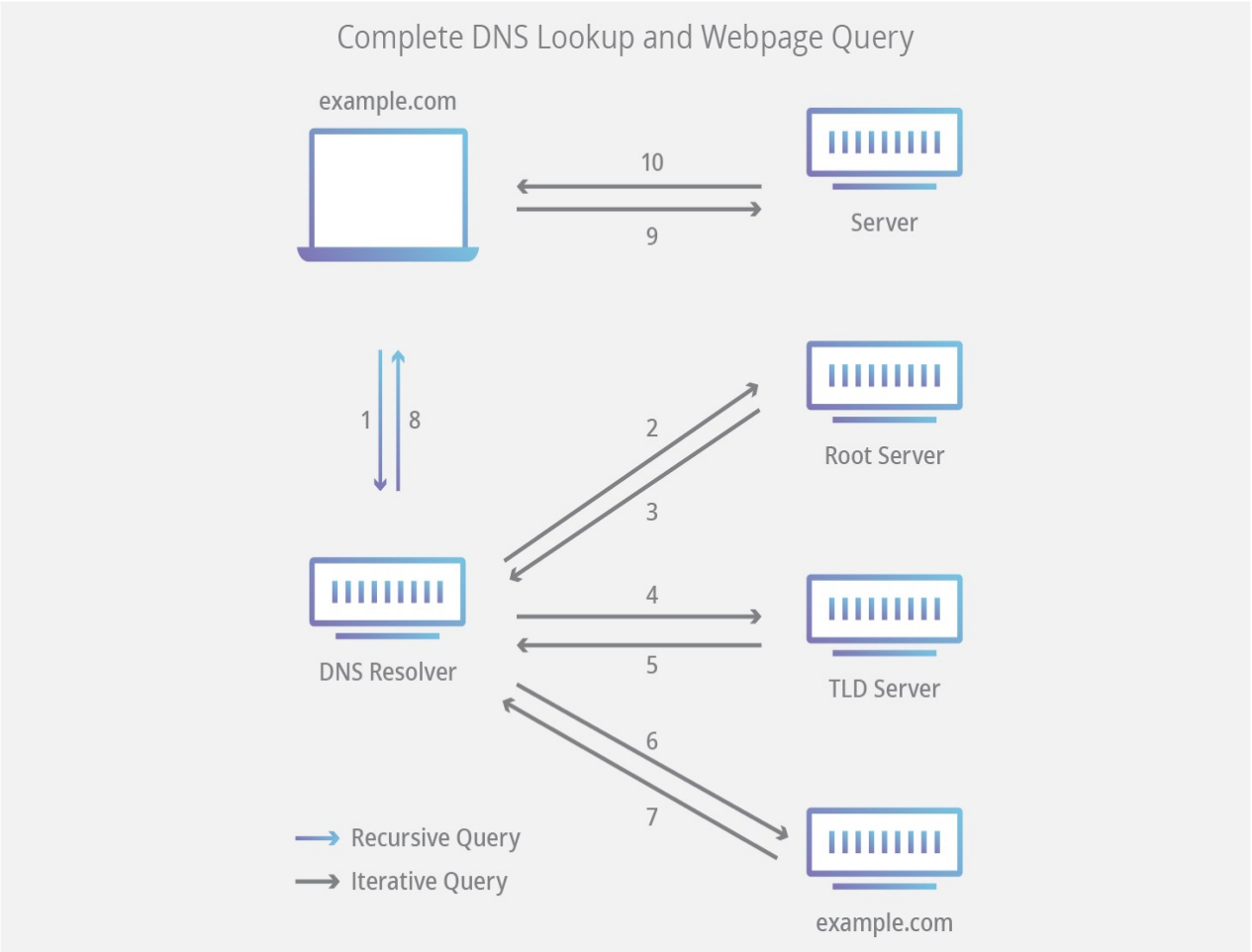
7.The IP address for example.com is then returned to the resolver from the nameserver.

8.The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:

9.The browser makes a HTTP request to the IP address.

10.The server at that IP returns the webpage to be rendered in the browser (step 10).



Implemetation :

configure Caching-only name server

Install BIND9

After updating the system, run the following command to install BIND9 packages which are used to setup DNS server.

```
sudo apt-get install bind9 bind9utils bind9-doc
```

Configuring Caching name server

Caching name server saves the DNS query results locally for a particular period of time. It reduces the DNS server's traffic by saving the queries locally, therefore it improves the performance and efficiency of the DNS server.
To configure Caching name server, edit **/etc/bind/named.conf.options** file:

```
sudo nano /etc/bind/named.conf.options
```

Uncomment the following lines. And then, add your ISP or Google public DNS server IP addresses.

```
forwarders {
 8.8.8.8;
 };
```

And then restart bind9 service to take effect the changes.

```
sudo systemctl restart bind9
```

We have successfully installed the caching name server.
Testing Caching name server

Now let us check if it is working or not using command:

```
dig -x 127.0.0.1
```

Primary DNS Server

Let us edit bind9 configuration file

Edit ‘**/etc/bind/named.conf’** using any editor of your choice:

```
sudo nano /etc/bind/named.conf
```

This file should have the following lines in it. If the lines are not there, just add them.

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Save the changes and exit the file.

We need to define the forward and reverse zone files.
To do so, edit **named.conf.local** file:

```
sudo nano /etc/bind/named.conf.local
```

Define the forward and reverse files as shown below.

```
zone "pdpu.lan" {
        type master;
        file "/etc/bind/for.pdpu.lan";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/rev.pdpu.lan";
};
```

Let us now create the zone files which we defined in the previous step.

First let us create forward zone file as shown below.

```
sudo nano /etc/bind/for.pdpu.lan
```

Add the following lines:

```
$TTL 86400
@   IN  SOA     pri.pdpu.lan. root.pdpu.lan. (
        2011071001  ;Serial
        3600 ;Refresh
        1800  ;Retry
        604800 ;Expire
        86400 ;Minimum TTL
)
@       IN  NS          pri.pdpu.lan.
@       IN  A           192.168.1.200
@       IN  A           192.168.1.201
pri     IN  A           192.168.1.200
client  IN  A           192.168.1.202
```

Similarly, you can add the other client records as defined in the above file.

Save and close the file. Next create reverse zone.

```
sudo nano /etc/bind/rev.pdpu.lan
```

Add the following lines:

```
$TTL 86400
@  IN SOA   pri.pdpu.lan. root.pdpu.lan. (
        2011071002 ;Serial
        3600            ;Refresh
        1800            ;Retry
        604800          ;Expire
        86400           ;Minimum TTL
)
@       IN NS           pri.pdpu.lan.
@       IN PTR          pdpu.lan.
pri     IN A            192.168.1.200
sec     IN  A           192.168.1.201
client IN  A            192.168.1.202
200     IN  PTR         pri.pdpu.lan.
202     IN  PTR         client.pdpu.lan.
```

Set the proper permissions and ownership to the bind9 directory.

```
sudo chmod -R 755 /etc/bind

sudo chown -R bind:bind /etc/bind
```

Next, we need to verify the DNS configuration files and zone files.
Check the DNS configuration files with commands:

```
sudo named-checkconf /etc/bind/named.conf

sudo named-checkconf /etc/bind/named.conf.local
```

If the above commands returns nothing, it means DNS configuration is valid.
Next, check the zone files using commands:

```
sudo named-checkzone pdpu.lan /etc/bind/for.pdpu.lan
```

Sample output:

```
zone pdpu.lan/IN: loaded serial 2011071001
OK
```

Check the reverse zone file:

```
sudo named-checkzone pdpu.lan /etc/bind/rev.pdpu.lan
```

Sample output:

```
 zone pdpu.lan/IN: loaded serial 2011071002
OK
```

Finally, restart Bind9 service.

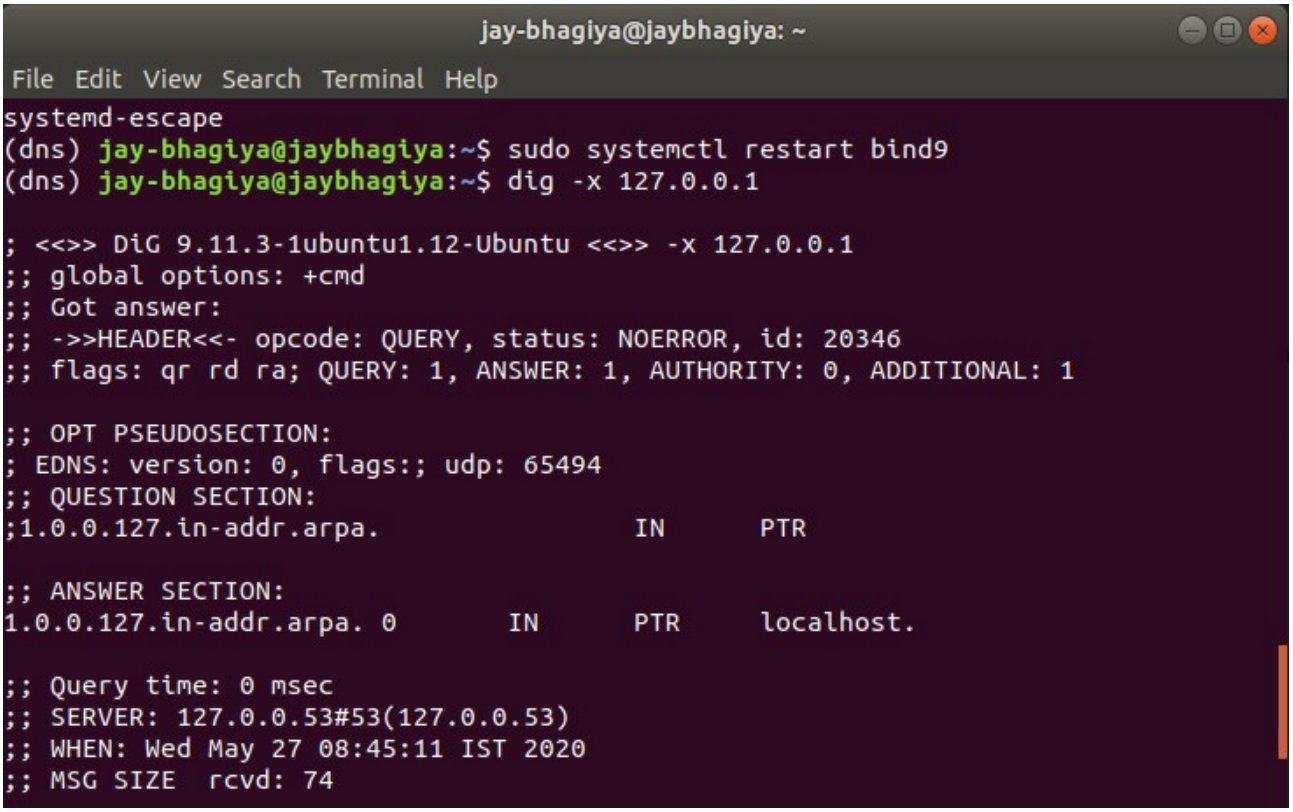
```
sudo systemctl restart bind9
```

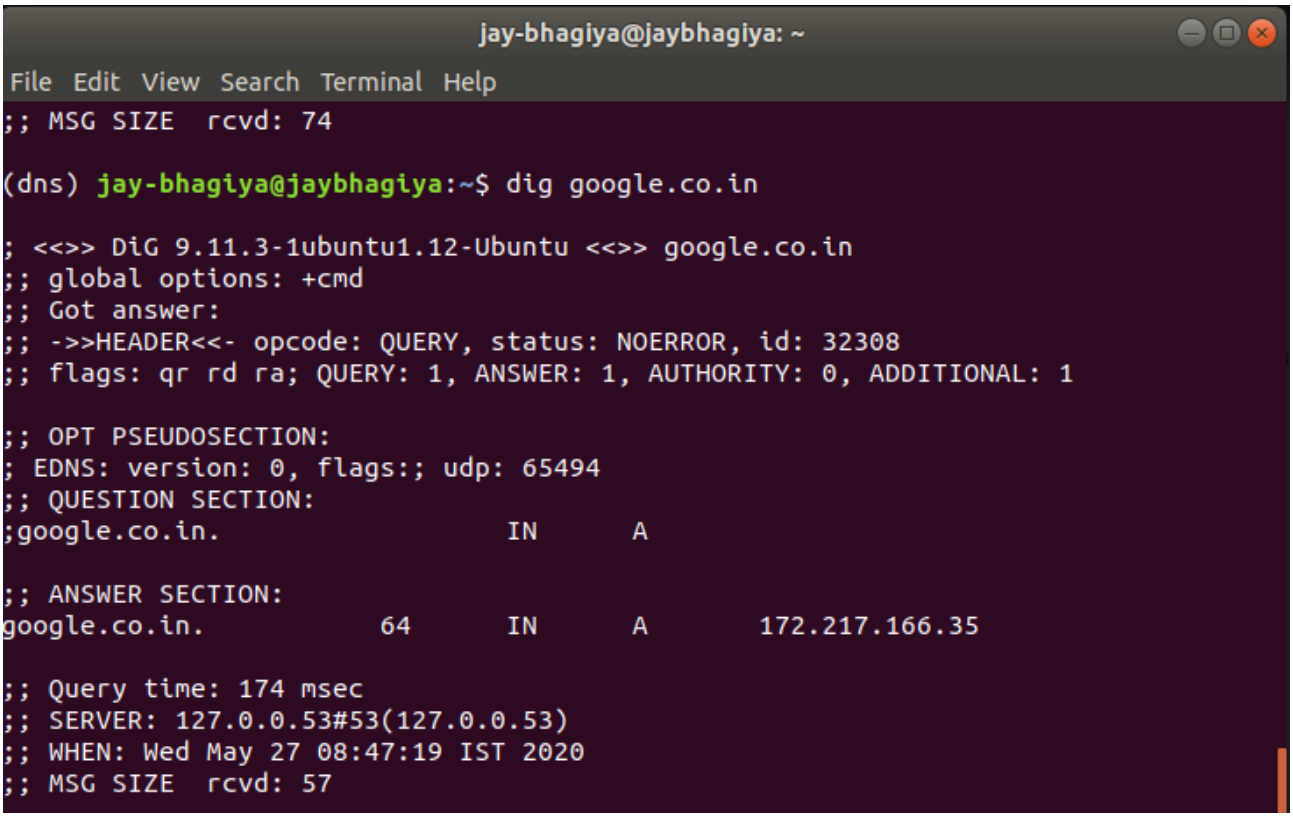
Verify DNS server using dig or nslookup commands.

```
dig pri.pdpu.lan
```

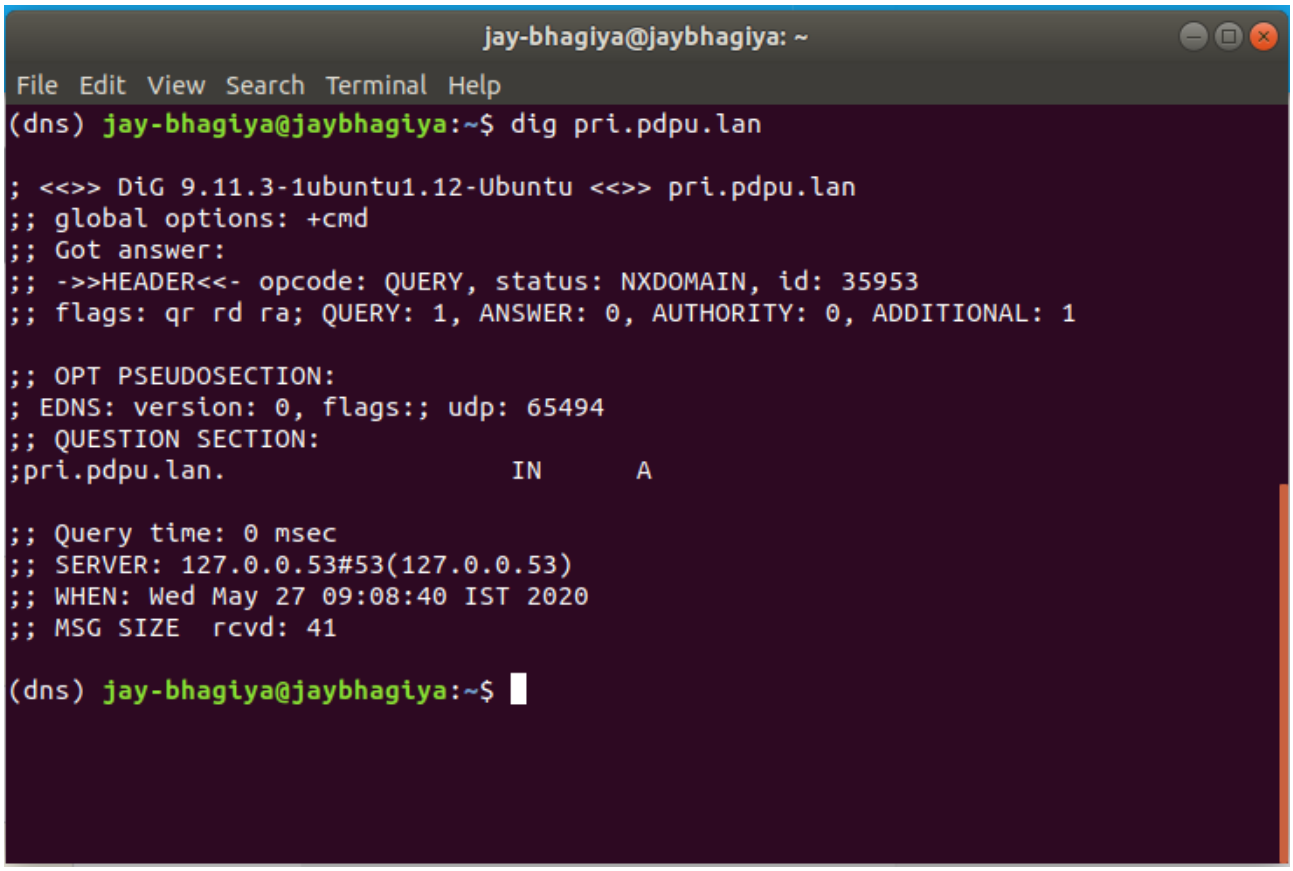
Output :

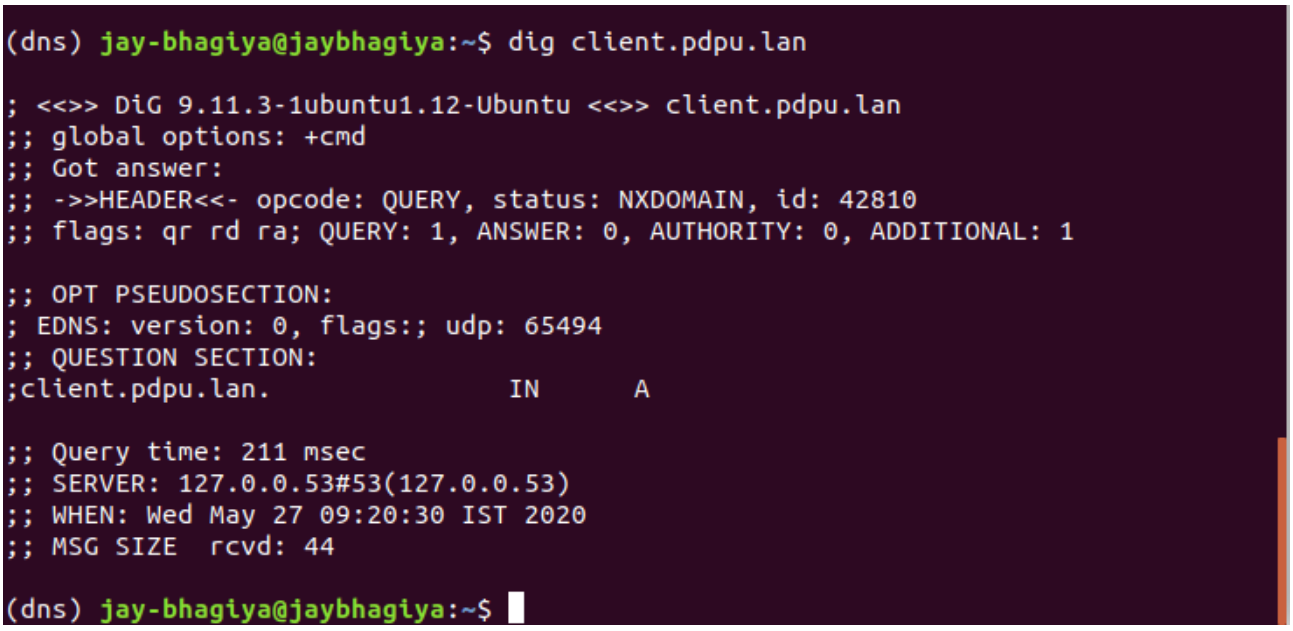
- Test of Catching name server.





- Test of primary DNS Server





Learning Outcome :

We have learn how to setup DNS server in Ubuntu.