Xavier Institute of Engineering

Mahim, Mumbai 400016

Department of Information Technology

(Affiliated to University of Mumbai)

Certificate

This is to certify that **Soham Desai** of **Second Year Semester IV** of Information Technology Engineering having

Roll No.**11** has performed the experiments during academic year 2021-22.

Practical In-charge

Head of the department

Examiner

Principal

Date:05/05/2022

Xavier Institute of Engineering

Mahim, Mumbai 400016

Department of Information Technology

(Affiliated to University of Mumbai)

Index of Experiments for **Network Lab** for the Academic Year **2021-22**

Sr. No.	Title of the Experiment	Date	Remarks
1	Execute and analyze basic networking commands	13/01/2022	
2	Installation and configuring of NS-2 simulator and introduction to Tcl using Hello program	20/01/2022	
3	Write TCL scripts to create topologies. Create and run traffics and analyse the result using NS2	27/01/2022	
4	Write TCL scripts for topology with Graphical simulation of traffic consideration (TCP, UDP) using NAM and plot the graph	03/02/2022	
5	Implement distance vector and link state routing protocols in NS2.	10/02/2022	
6	Study and Implement Socket Programming using TCP.	03/03/2022	
7	Study and Implement Socket Programming using UDP.	10/03/2022	
8	Study various network protocol analyser tools and analyse the network traffics using one of the network protocol analyser tools.	17/03/2022	

9	Perform remote login using Telnet Server	24/03/2022	
10	Case Study: (Group Activity):Design a network for an organization using the concepts of Addressing (IP Address Assignment), Naming (DNS) and Routing.	31/03/2022	
	Assignments		
1	NL Assignment-1	25/03/2022	
2	NL Assignment-2	08/04/2022	

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

EXPERIMENT 1

Aim: Execute and analyze basic networking commands

LO 1: Execute and evaluate network administration commands and demonstrate their use in different network scenarios

Theory: A computer network is defined as a group of 2 or more computers that are connected and can electronically communicate with each other. The computers are identified using their hostnames, IP, and mac addresses. A simple home or office network is referred to as a LAN, short for Local Area Network. A LAN covers a small area such as a home, office, or restaurant network. In contrast, a WAN (Wide Area Network) spans a large geographical region.Some Basic Networking commands are:

1. ifconfig Command:

The ifconfig command lists the network interfaces attached to the PC along with other statistics such as the IP addresses associated with each interface, subnet mask, and MTU to mention just a few.

2. ping Command:

Short for packet internet groper, the ping command is used to check connectivity between 2 systems or servers. It sends out an ICMP echo request to a remote host and waits for a reply. If the host is up, the echo request bounces off the remote host and is sent back to the source informing the user that the host is up or available.

3. traceroute Command:

The traceroute command displays the route that an ICMP ping packet takes from your device to the destination host or server. It displays the IP addresses of devices that the packet hops through before getting to the remote destination. Traceroute command is a cool diagnostic command that you can use to troubleshoot the network where the ping command gives you failed results. It shows the device at which the packets are being dropped.

4. nslookup Command:

The nslookup utility is yet another command-line tool that is used for making DNS lookups in a bid to retrieve domain names and A records.

5. netstat Command:

The netstat command prints out the network interface statistics. It can display the routing table, ports that various services are listening on, TCP and UDP connections, PID, and UID.

6. dig Command:

The dig utility (short for Domain Information Groper) is a command-line tool for probing DNS nameservers. It takes a domain name as the argument and displays information such as the host address, A record, MX (mail exchanges) record, nameservers, etc.

Name: Soham Desai
XIE ID: 202003021

Roll no:11
Batch: B

Output:

```
ipconfig
ping
ping 142.250.77.68
ping
ping -c 142.250.77.68
netstat -tl
netstat -ntl
netstat -nul
netstat -nulp
netstat -ntlp
netstat -a
netstat -tl
netstat -r
netstat -rn
nslookup
doskey /history
```

```
281  ifconfig
282  ping 172.20.208.144
283  ping 172.20.208.146
284  ping 172.20.208.146
285  ping 172.20.208.144
286  ping 192.168.163.128
287  sudi ifconfig
288  sudo ifconfig
289  nslookup www.google.com
290  netstat
291  netstat -ntl
292  netstat -ttl
293  netstat -nulp
294  netstat -ntlp
295  netstat -a
296  netstat -tl
297  netstat -nul
298  netstat -r
299  netstat -rn
300  nslookup
301  history
```

Conclusion: From this experiment we have learned some of the basic commands for networking in Windows and LINUX.

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

EXPERIMENT 2

Aim: Installation and configuring of NS-2 simulator and introduction to Tcl using Hello program

LO 2: Demonstrate the installation and configuration of network simulator.

Theory:

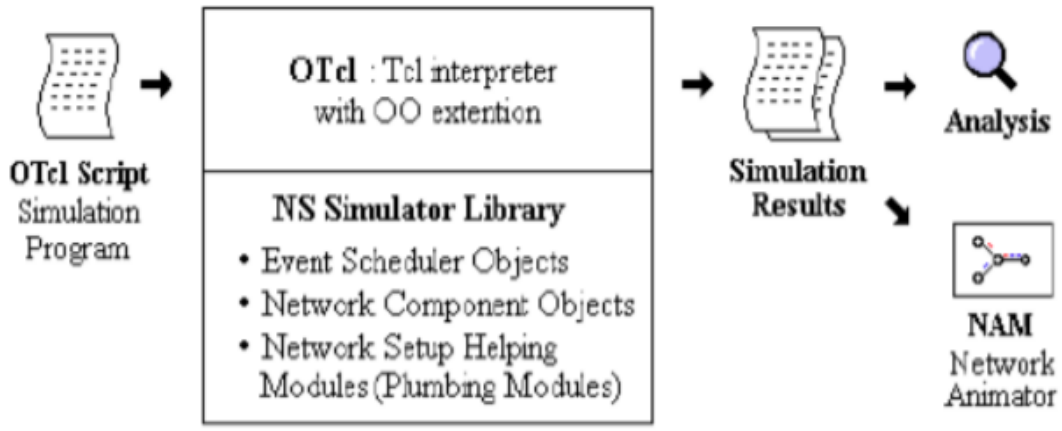
Introduction to NS2 -

NS2 is an open-source simulation tool that runs on Linux. It is a discrete event simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP and SRM over wired and wireless (local and satellite) networks.

Widely known as NS2, is simply an event driven simulation tool.Useful in studying the dynamic nature of communication networks.Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2.

In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors.

Basic Architecture of NS2



Introduction to TCL script -

TCL – Tool Command Language

Tcl is a very simple programming language.

Basic syntax

Tcl scripts are made up of commands separated by newlines or semicolons.

Commands all have the same basic form illustrated by the following example:

expr 20 + 10

This command computes the sum of 20 and 10 and returns the result, 30.

Name: Soham Desai
XIE ID: 202003021

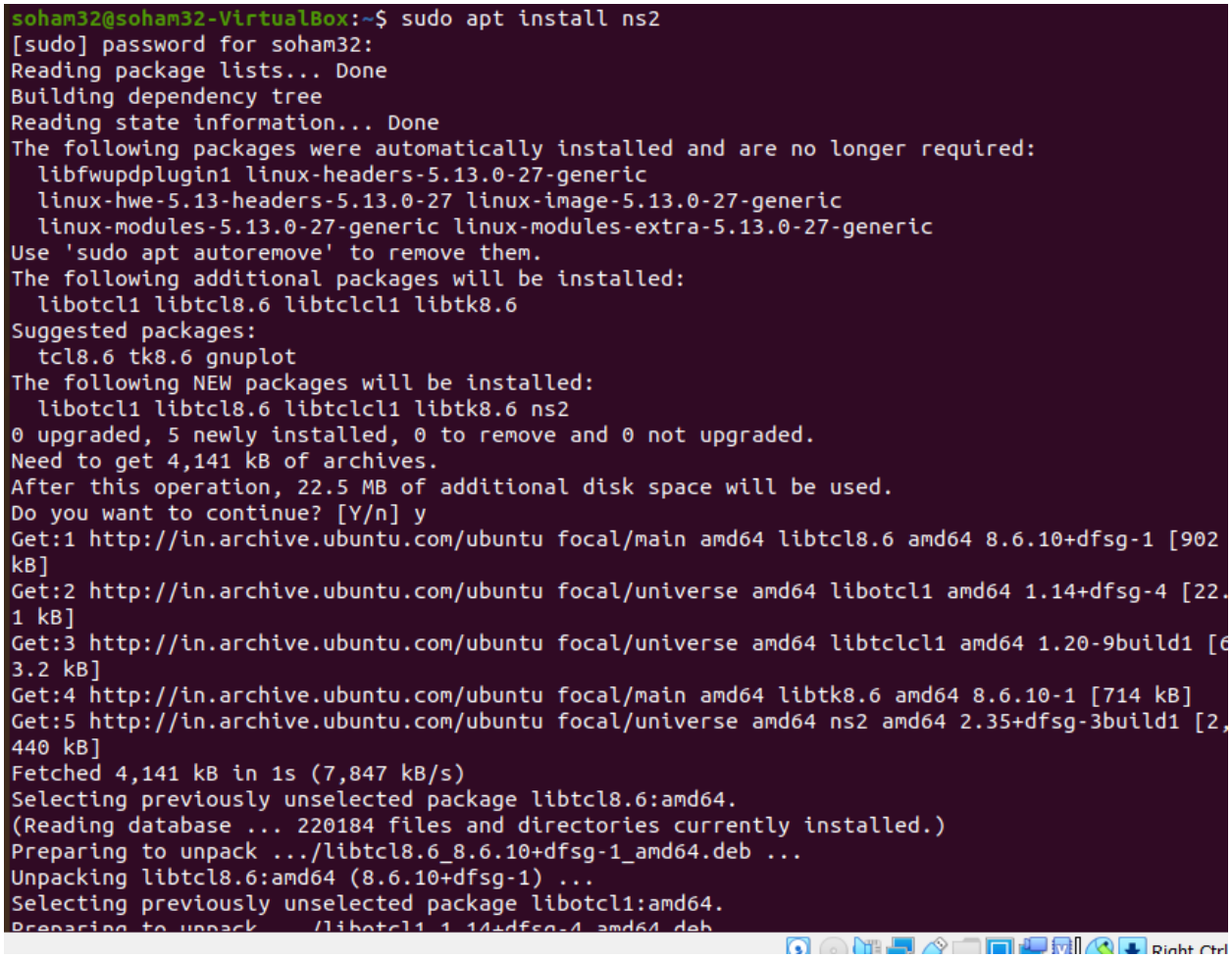
Roll no:11
Batch: B

Each Tcl command consists of one or more words separated by spaces. In this example there are four words: expr, 20, +, and 10. The first word is the name of a command and the other words are arguments to that command.

All Tcl commands consist of words, but different commands treat their arguments differently. The expr command treats all of its arguments together as an arithmetic expression, computes the result of that expression, and returns the result as a string.

However, for most commands the word structure is important, with each word used for a distinct purpose. All Tcl commands return results.

Output:



Name: Soham Desai
XIE ID: 202003021

Roll no:11
Batch: B

```
soham32@soham32-VirtualBox:~$ sudo apt install nam
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 linux-headers-5.13.0-27-generic
  linux-hwe-5.13-headers-5.13.0-27 linux-image-5.13.0-27-generic
  linux-modules-5.13.0-27-generic linux-modules-extra-5.13.0-27-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  nam
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 196 kB of archives.
After this operation, 695 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 nam amd64 1.15-5build1 [196 kB]
Fetched 196 kB in 0s (899 kB/s)
Selecting previously unselected package nam.
(Reading database ... 220555 files and directories currently installed.)
Preparing to unpack .../nam_1.15-5build1_amd64.deb ...
Unpacking nam (1.15-5build1) ...
Setting up nam (1.15-5build1) ...
Processing triggers for man-db (2.9.1-1) ...
```

```
1 puts "Hello World"
```

```
soham32@soham32-VirtualBox:~$ ns hello.tcl
Hello World
```

Conclusion: From the experiment we learn to install NS2 and how to write code in TCL scripts.

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Experiment No: 3

Aim: Write TCL scripts to create topologies. Create and run traffics and analyze the result using NS2

LO no: 3,5

LO statement: Demonstrate and measure different network scenarios and their performance behavior. Analyze the traffic flow of different protocols

Theory:

Simulation is the process of *learning by doing*. Whenever there is something new in the world, we try to analyze it first by examining it and in the process get to learn a lot of things. This entire course is called Simulation.

To create a node we can simply use the simulator method node. The following two lines create two nodes and assign them to the handles 'n0' and 'n1'.

```
set n0 [$ns node] set n1
                [$ns node]
```

We can then either use the simulator method simplex-link or the method duplex-link to connect the nodes with a link:

```
$ns simplex-link $n0 $n1 1Mb 10ms DropTail $ns duplex-link
$n0 $n1 1Mb 10ms DropTail
```

The first line creates a unidirectional link between n0 and n1 with bandwidth 1Mbps, a propagation delay of 10ms and a DropTail queue. The second line creates a bidirectional link with the same parameters.

Traffic generation in ns is based on objects of two classes, the Agent and the Application. Agents represent endpoints where network-layer packets are constructed or consumed. Every node in the network that needs to send or receive traffic must have an agent attached to it. These agents can be thought of as the implementation of the transport protocol. On top of that an agent runs an application. The application determines the kind of traffic source that is simulated (e.g. ftp or telnet). Applications represent the application layer in an ns-simulation.

A. Creating Agents

Corresponding to the two most popular transport protocols used in the Internet there are also two types of agents in ns: UDP agents and TCP agents. The following code shows an example of attaching a UDP agent to nodes n0 and n1:

```
set udp0 [new Agent/UDP]
```

Name: Soham Desai Roll no:11
XIE ID: 202003021 Batch: B

$ns attach-agent $n0 $udp0 set null0
[new Agent/Null] $ns attach-agent $n1
$null0 ns connect udp0 null0
This code first creates a UDP agent and attaches it to n0 using the attach-agent procedure. It then creates a Null agent which will act as a traffic sink and attaches it to n1. Finally, the two agents are connected using the simulator method connect. In the next section the UDP agent will be used by an application to send data. (c) Read the ns manual to find out how to write similar code to set up a TCP connection.

B. Creating Applications -
In the previous section we have set up the agents implementing the transport layer. We will now create applications that we attach to the transport agents and that will actually generate traffic. In ns there are two basic types of applications: simulated applications and traffic generators. Traffic generators generate On/Off traffic: during On-periods, packets are generated at a constant burst rate and during Off-periods no packets are generated. ns provides three different classes of traffic generators which differ in how the lengths of the On and Off-periods are modeled:
1. A traffic generator of the type Application/Traffic/Exponential takes the length of the On and Off periods from an Exponential distribution.
2. A Application/Traffic/Pareto source generates the lengths of these periods from a Pareto distribution.
3. Finally, the class Application/Traffic/CBR has no off periods and generates packets at a constant bit rate. The following code generates one traffic generator of each class.

```
set exp [new set  Application/Traffic/Exponential]
        par [new  Application/Traffic/Pareto]
set cbr0 [new Application/Traffic/CBR]
```

See the ns-manual for how to configure these traffic generators. All traffic generators run on top of a UDP agent. Therefore, we have to attach a traffic generator to a UDP agent before we can use it to send data. The following example illustrates the use of the CBR traffic generator that we created above.

```
$cbr0 set packetSize 500 $cbr0 set
interval 0.005 $cbr0 attach-agent
$udp0 $ns at 1.0 ‘‘$cbr start’’
```

The simulated applications currently implemented in ns are Application/FTP and Application/Telnet. These try to simulate the corresponding applications in the real world: FTP and Telnet. Like the real applications the ns applications can run only on TCP. They therefore have to be attached to a TCP agent.

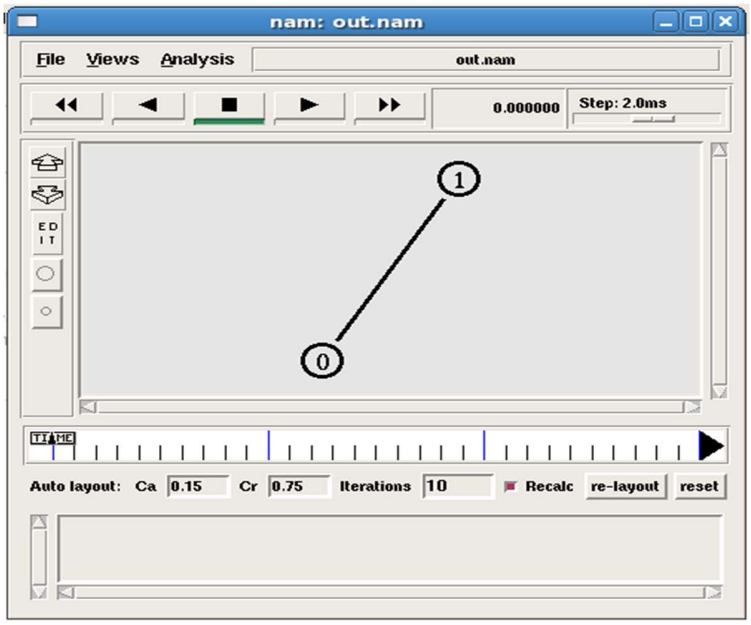
Name: Soham Desai
XIE ID: 202003021

Roll no:11
Batch: B

Code:

```
set ns [new Simulator]
set f [open out.tr w]
$ns trace-all $f
set nr [open out.nam w]
$ns namtrace-all $nr
proc finish {} {
global ns f nr
$ns flush-trace
close $f
close $nr
exec nam out.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns at 5.0 "finish"
$ns run
```

Output:



Conclusion: From the experiment we have learned to write TCL scripts and we can create a network topology and traffic and we can analyze the result using NS2.

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Experiment 4

Aim: Write TCL scripts for topology with Graphical simulation of traffic consideration (TCP, UDP) using NAM and plot the graph.

LO No: 3,5

LO statement : Demonstrate and measure different network scenarios and their performance behavior.
Analyze the traffic flow of different protocols.

Theory:

Creating topology
● Two nodes connected by a link
● Creating nodes
set n0 [$ns node]
set n1 [$ns node]
● Creating link between nodes
$ns <link_type> $n0 $n1 <bandwidth> <delay><queue-type>
$ns duplex-link$n0 $n1 1Mb 10ms DropTail

Traffic on top of TCP
● FTP
set ftp [new Application/FTP]
$ftp attach-agent$tcp0
● Telnet
set telnet [new Application/Telnet]
$telnet attach-agent$tcp0
PROCEDURE
STEP 1: Start
STEP 2: Create the simulator object ns for designing the given simulation
STEP 3: Open the trace file and nam file in the write mode
STEP 4: Create the nodes of the simulation using the ‘set’ command
STEP 5: Create links to the appropriate nodes using $ns duplex-link command
STEP 6: Set the orientation for the nodes in the simulation using ‘orient’ command
STEP 7: Create TCP agent for the nodes and attach these agents to the nodes
STEP 8: The traffic generator used is FTP for both node0 and node1

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

STEP 9: Configure node1 as the sink and attach i
STEP10: Connect node0 and node1 using ‘connect’ command
STEP 11: Setting color for the nodes
STEP 12: Schedule the events for FTP agent 10 sec
STEP 13: Schedule the simulation for 5 minutes

Code 1:

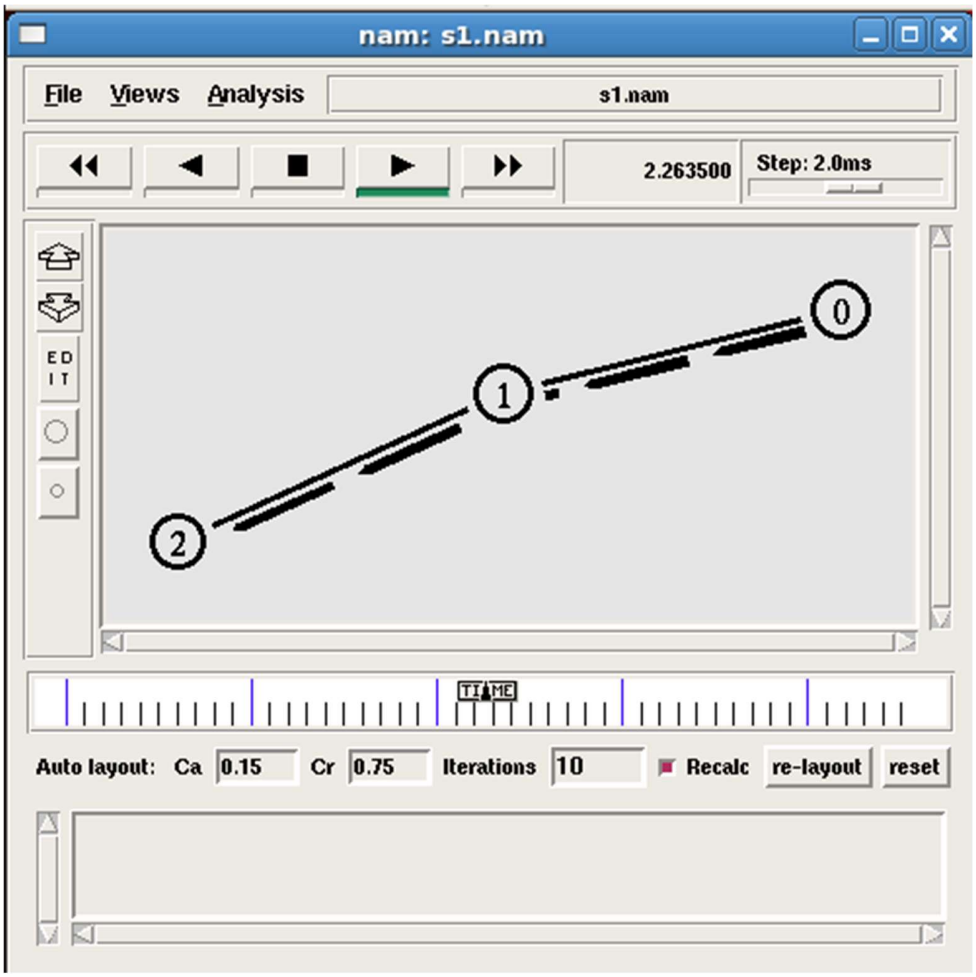
```
set ns [new Simulator]
set nf [open s1.nam w]
$ns namtrace-all $nf
set nf1 [open s1.tr w]
$ns trace-all $nf1
proc finish {}
{
global ns nf nf1
$ns flush-trace
close $nf
close $nf1
exec nam s1.nam &amp;amp;
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
 $cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n2 $null0
set null1 [new Agent/Null]
$ns attach-agent $n2 $null
```

Name: Soham Desai
XIE ID: 202003021

Roll no:11
Batch: B

```
$ns connect $udp0 $null0
$ns connect $null0 $null1
$ns at 0.5 &amp;quot;$cbr0 start&amp;quot;
$ns at 2.5 &amp;quot;$cbr0 stop&amp;quot;
$ns at 2.7 &amp;quot;$cbr1 start&amp;quot;
$ns at 4.5 &amp;quot;$cbr1 stop&amp;quot;
$ns at 5.0 &amp;quot;finish&amp;quot;
$ns run
```

Output



Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Code 2:

```
#Create a simulator object
set ns [new Simulator]
#Open trace files
set f [open out.tr w]
$ns trace-all $f
#open nam file
set nf [open out.nam w]
$ns namtrace-all $nf
#Define a &amp;#39;finish&amp;#39; procedure
proc finish {} {
global ns f nf
$ns flush-trace
close $f
close $nf
exec nam out.nam &amp;amp;
exit 0
}
#Create five nodes
set s1 [$ns node]
set s2 [$ns node]
set s3 [$ns node]
set G [$ns node]
set r [$ns node]
#Create links between the nodes
$ns duplex-link $s1 $G 1Mb 10ms DropTail
$ns duplex-link $s2 $G 1Mb 10ms DropTail
$ns duplex-link $s3 $G 1Mb 10ms DropTail
$ns duplex-link $G $r 1Mb 10ms DropTail
#Create a TCP agent and attach it to node s1
set tcp1 [new Agent/TCP/Reno]
$ns attach-agent $s1 $tcp1
$tcp1 set window_ 8
$tcp1 set fid_ 1
#Create a TCP agent and attach it to node s2
set tcp2 [new Agent/TCP/Reno]
$ns attach-agent $s2 $tcp2
$tcp2 set window_ 8
$tcp2 set fid_ 2
#Create a TCP agent and attach it to node s3
```

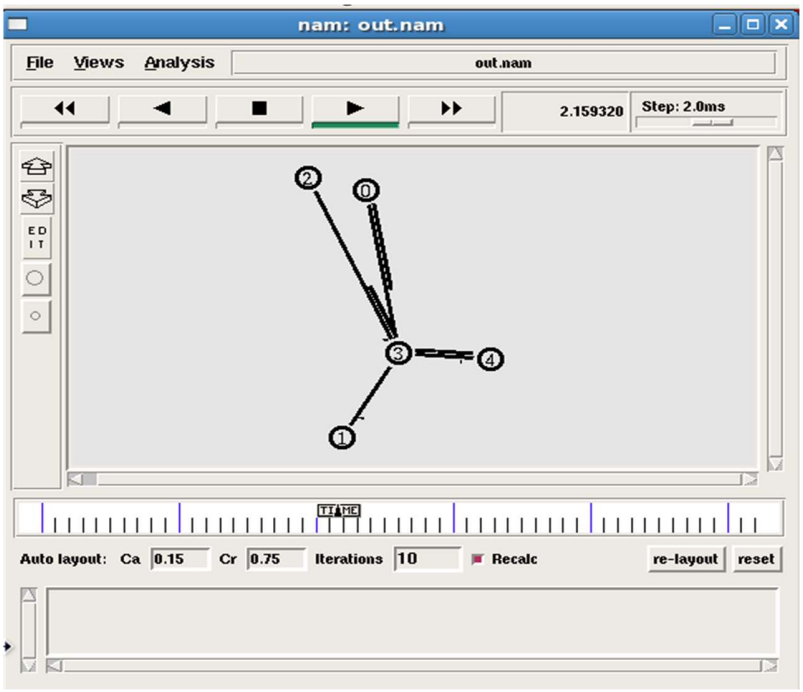
Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

```
set tcp3 [new Agent/TCP/Reno]
$ns attach-agent $s3 $tcp3
$tcp3 set window_ 4
$tcp3 set fid_ 3
#Create TCP sink agents and attach them to node r
set sink1 [new Agent/TCPSink]
set sink2 [new Agent/TCPSink]

set sink3 [new Agent/TCPSink]
$ns attach-agent $r $sink1
$ns attach-agent $r $sink2
$ns attach-agent $r $sink3
#Connect the traffic sources with the traffic sinks
$ns connect $tcp1 $sink1
$ns connect $tcp2 $sink2
$ns connect $tcp3 $sink3
# You cannot connect two TCP sources to the same TCP sink, You can do that
for UDP traffic
#Create FTP applications and attach them to agents
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3
$ns at 0.1 &quot;$ftp1 start&quot;
$ns at 0.1 &quot;$ftp2 start&quot;
$ns at 0.1 &quot;$ftp3 start&quot;
$ns at 5.0 &quot;$ftp1 stop&quot;
$ns at 5.0 &quot;$ftp2 stop&quot;
$ns at 5.0 &quot;$ftp3 stop&quot;
$ns at 5.25 &quot;finish&quot;
$ns run
```

Name: Soham Desai
XIE ID: 202003021

Roll no:11
Batch: B

Output 2:



Conclusion: From this experiment we can conclude that we can use TCL scripts for creating topology with Graphical simulation for traffic consideration (TCP, UDP) using NAM.

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Experiment 5

Aim: Implement distance vector and link state routing protocols in NS2

LO NO: 3,5

LO statement: Demonstrate and measure different network scenarios and their performance behavior.
Analyze the traffic flow of different protocols.

Theory:

Distance vector algorithm:

- The Distance vector algorithm is iterative, asynchronous and distributed.
- Distributed: It is distributed in that each node receives information from one or more of its directly attached neighbors, performs calculation and then distributes the result back to its neighbors.
- Iterative: It is iterative in that its process continues until no more information is available to be exchanged between neighbors.
- Asynchronous: It does not require that all of its nodes operate in the lock step with each other.
- The Distance vector algorithm is a dynamic algorithm.
- It is mainly used in ARPANET, and RIP.
- Each router maintains a distance table known as Vector.
- Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

The three keys to understand the Link State Routing algorithm:

- Knowledge about the neighborhood: Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.
- Flooding: Each router sends the information to every other router on the internetwork except its neighbors. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.
- Information sharing: A router sends the information to every other router only when the change occurs in the information.
- The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.

- The Dijkstra's algorithm is an iterative, and it has the property that after kth iteration of the algorithm, the least cost paths are well known for k destination nodes.

Code 1:

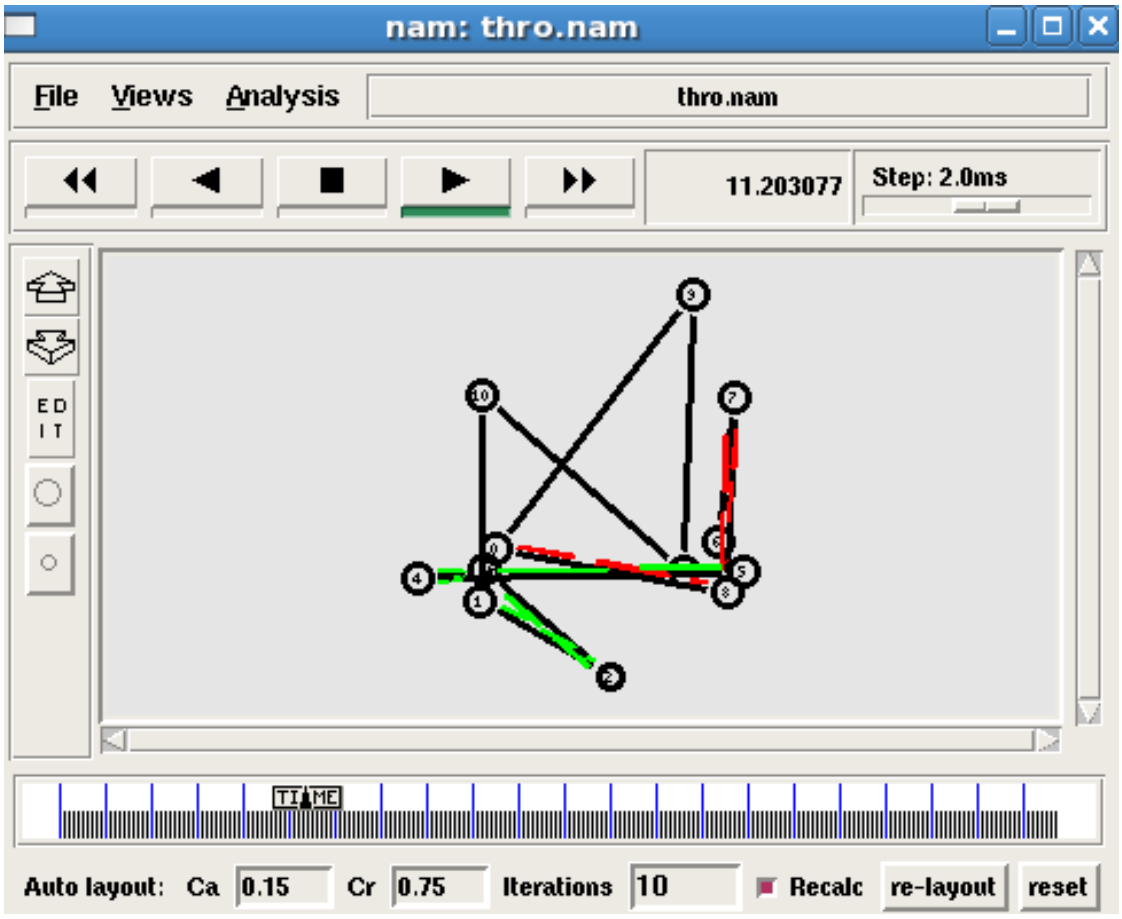
Code for distance vector routing protocol simulation:

```
set ns [new Simulator]
set nr [open thro.tr w]
$ns trace-all $nr
set nf [open thro.nam w]
$ns namtrace-all $nf
proc finish {}
{ global ns nr
$ns flush-trace close $nf close $nr
exec nam thro.nam &amp;amp; exit 0
}
for { set i 0 } { $i &amp;lt; 12} { incr i 1 } { set n($i) [$ns node]}
for {set i 0} {$i &amp;lt; 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0 set null0 [new Agent/Null]

$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0 set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1 set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0
```

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

```
$ns rtproto DV
$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)
$udp0 set fid_ 1
$udp1 set fid_ 2
$ns color 1 Red
$ns color 2 Green
$ns at 1.0 &amp;quot;$cbr0 start&amp;quot;
$ns at 2.0 &amp;quot;$cbr1 start&amp;quot;
$ns at 45 &amp;quot;finish&amp;quot;
$ns run
```

Output:



Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Code 2:

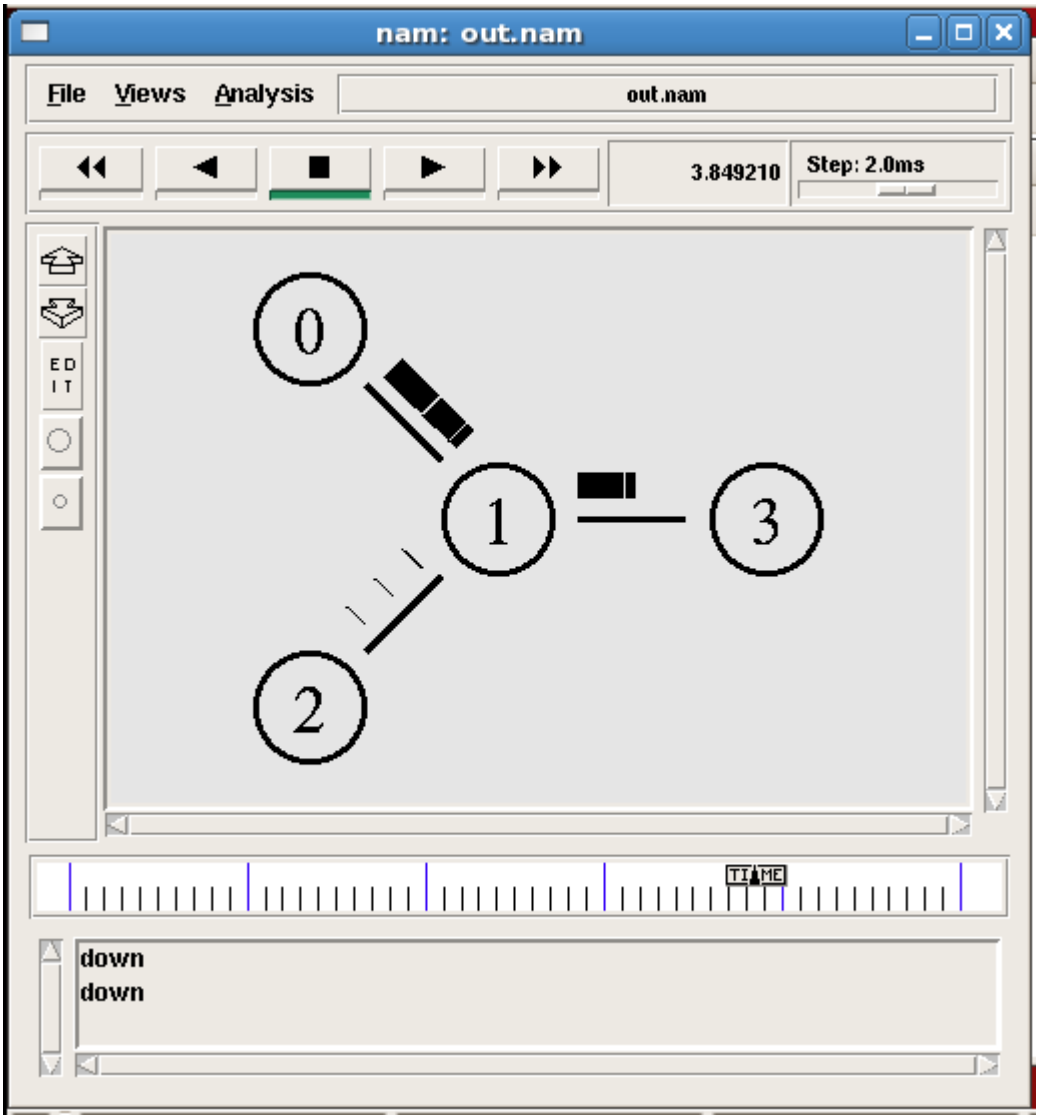
Code for Link state routing protocol simulation:

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set tr [open out.tr w]
$ns trace-all $tr
proc finish {} {
global nf ns tr
$ns flush-trace
close $tr
exec nam out.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n3 10Mb 10ms DropTail
$ns duplex-link $n2 $n1 10Mb 10ms DropTail
$ns duplex-link-op $n0 $n1 orient right-down
$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n2 $n1 orient right-up
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set ftp [new Application/FTP]
$ftp attach-agent $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
set udp [new Agent/UDP]
$ns attach-agent $n2 $udp
set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $tcp $sink
```

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

```
$ns connect $udp $null
$ns rtmodel-at 1.0 down $n1 $n3
$ns rtmodel-at 2.0 up $n1 $n3
$ns rtproto LS
$ns at 0.0 "$ftp start"
$ns at 0.0 "$cbr start"
$ns at 5.0 "finish"
$ns run
```

Output 2:



Conclusion: From this experiment we can conclude that we can implement distance vector and link state routing protocol in NS2.

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Experiment No: 6

Aim: Study and Implement Socket Programming using TCP

LO 4: Implement the socket programming for client server architecture.

Theory:

A socket programming interface provides the routines required for interprocess communication between applications, either on the local system or spread in a distributed, TCP/IP based network environment. Once a peer-to-peer connection is established, a socket descriptor is used to uniquely identify the connection. The socket descriptor itself is a task specific numerical value.

One end of a peer-to-peer connection of a TCP/IP based distributed network application described by a socket is uniquely defined by

- Internet address
  for example 127.0.0.1 (in an IPv4 network) or FF01::101 (in an IPv6 network).

- Communication protocol
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
- Port
  A numerical value, identifying an application. We distinguish between
  - "well known" ports, for example port 23 for Telnet
  - user defined ports

Socket applications were usually C or C++ applications using a variation of the socket API originally defined by the Berkeley Software Distribution (BSD). The JAVA language also provides a socket API. JAVA based Client/Server applications exploit those socket services.

Socket programming interfaces have been standardized for ease of portability by The Open Group for example.

Besides TCP/IP based sockets, UNIX systems provide socket interfaces for interprocess communication (IPC) within the local UNIX host itself. Those UNIX sockets use the local file system for interprocess communication.

z/VSE provides TCP/IP based socket services. They can be used for IPC too, although they are primarily aimed for network communication only.

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Code :

1. **Client :**

```
import java.net.*;
import java.io.*;
public class Client {
   private Socket socket = null;
   private DataInputStream input = null;
   private DataOutputStream out = null;
   public Client(String address, int port) {
      try {
         socket = new Socket(address, port);
         System.out.println("Connected");
         input = new DataInputStream(System.in);
         out = new DataOutputStream(socket.getOutputStream());
      } catch (IOException u) {
         System.out.println(u);
      }
      String line = "";
      while (!line.equals("Over")) {
         try {
            line = input.readLine();
            out.writeUTF(line);
         } catch (IOException i) {
            System.out.println(i);
         }
      }
      try {
         input.close();
         out.close();
         socket.close();
      } catch (IOException i) {
         System.out.println(i);
      }
   }
   public static void main(String[] args) {
      Client client = new Client("127.0.0.1", 5000);
   }
    }
```

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

2. Server:

```
import java.net.*;
import java.io.*;
public class Server {
   private Socket socket = null;
   private ServerSocket server = null;
   private DataInputStream in = null;
   public Server(int port) {
      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");
         in = new DataInputStream(
               new BufferedInputStream(socket.getInputStream()));
         String line = "";
         while (!line.equals("Over")) {
            try {
               line = in.readUTF();
               System.out.println(line);
            } catch (IOException i) {
               System.out.println(i);
            }
         }
         System.out.println("Closing connection");
         socket.close();
         in.close();
      } catch (IOException i) {
         System.out.println(i);
      }
   }
   public static void main(String[] args) {
      Server server = new Server(5000);
   }
}
```

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Output:

1. **Client :**

```
Connected
hello world
This is socket programming
thank you for using!!
Over
```

2. **Server:**

```
Server started
Waiting for a client ...
Client accepted
hello world
This is socket programming
thank you for using!!
Over
Closing connection
```

Conclusion: From this experiment we have learned to do socket programming using TCP in Java and also how it actually works.

Name: Soham Desai
XIE ID: 202003021

Roll no:11
Batch: B

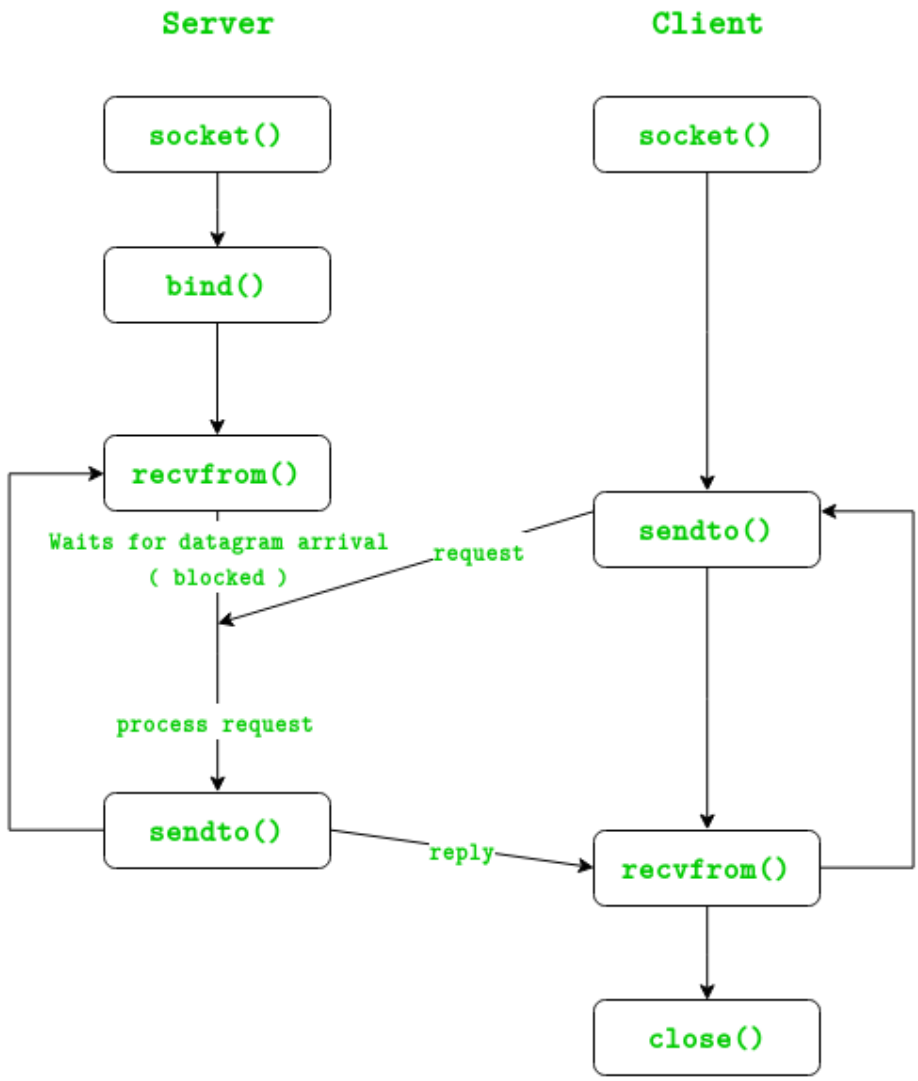
Experiment 7

Aim: Study and Implement Socket Programming using UDP

LO 4: Implement the socket programming for client server architecture.

Theory:

In UDP, the client does not form a connection with the server like in TCP and instead just sends a datagram. Similarly, the server need not accept a connection and just waits for datagrams to arrive. Datagrams upon arrival contain the address of the sender which the server uses to send data to the correct client.



The entire process can be broken down into the following steps :

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

UDP Server :

1. Create a UDP socket.
2. Bind the socket to the server address.
3. Wait until the datagram packet arrives from the client.
4. Process the datagram packet and send a reply to the client.
5. Go back to Step 3.

UDP Client :

1. Create a UDP socket.
2. Send a message to the server.
3. Wait until response from the server is received.
4. Process reply and go back to step 2, if necessary.
5. Close socket descriptor and exit.

Code :

1. Client :

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class UDP_client {
 public static void main(String args[]) throws IOException {
Scanner sc = new Scanner(System.in);
 DatagramSocket ds = new DatagramSocket();
 InetAddress ip = InetAddress.getLocalHost();
 byte buf[] = null;
 while (true) {
    String inp = sc.nextLine();
    buf = inp.getBytes();
    DatagramPacket DpSend =
          new DatagramPacket(buf, buf.length, ip, 1234);
    ds.send(DpSend);
    if (inp.equals("over"))
             break;
                 }
              }
}
```

2. Server:

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class UDP_server {
   public static void main(String[] args) throws IOException {
      DatagramSocket ds = new DatagramSocket(1234);
      byte[] receive = new byte[65535];
      DatagramPacket DpReceive = null;
      while (true) {
         DpReceive = new DatagramPacket(receive, receive.length);
         ds.receive(DpReceive);
         System.out.println("Client:-" + data(receive));
         if (data(receive).toString().equals("over")) {
            System.out.println("Closing connection.....EXITING");
            break;
         }
         receive = new byte[65535];
      }
   }
   public static StringBuilder data(byte[] a) {
      if (a == null)
         return null;
      StringBuilder ret = new StringBuilder();
      int i = 0;
      while (a[i] != 0) {
         ret.append((char) a[i]);
         i++;
      }
      return ret;
   }
}
```

Name: Soham Desai
XIE ID: 202003021
Roll no:11
Batch: B

Output:

```
hello
this is socket programming using UDP
Thank you for using!!
over
```

```
Client:-hello
Client:-this is socket programming using UDP
Client:-Thank you for using!!
Client:-over
Closing connection.....EXITING
```

Conclusion: From this experiment we have learned to do socket programming using UDP in Javaand also saw how it works.

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

EXPERIMENT NO. 8

AIM: Study various network protocol analyser tools and analyse the network traffics using one of the network protocol analyser tools.

LO5: Analyse the traffic flow of different protocols.

THEORY:
The main purpose of Cisco Packet Tracer is to help students learn the principles of networking with hands-on experience as well as develop Cisco technology-specific skills. Since the protocols are implemented in a software-only method, this tool cannot replace the hardware Routers or Switches. Interestingly, this tool does not only include Cisco products but also many more networking devices.

Using this tool is widely encouraged as it is part of the curriculum like CCNA, and CCENT where Faculties use Packet Trace to demonstrate technical concepts and networking systems. Students complete assignments using this tool, working on their own or in teams.

Engineers prefer to test any protocols on Cisco Packet Tracer before implementing them. Also, Engineers who would like to deploy any change in the production network prefer to use Cisco Packet Tracer to first test the required changes and proceed to deploy if and only if everything is working as expected.

This makes the job easier for Engineers allowing them to add or remove simulated network devices, with a Command-line interface and a drag and drop user interface.

Workspace :

Logical – A logical workspace shows the logical network topology of the network the user has built. It represents the placing, connecting, and clustering of virtual network devices.

Physical – The physical workspace shows the graphical physical dimension of the logical network. It depicts the scale and placement in how network devices such as routers, switches, and hosts would look in a real environment. It also provides a geographical representation of networks, including multiple buildings, cities, and wiring closets.

Key Features:

- Unlimited devices
- E-learning
- Customize single/multi-user activities

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

- Interactive Environment
- Visualizing Networks
- Real-time mode and Simulation mode
- Self-paced
- Supports majority of networking protocols
- International language support
- Cross platform compatibility

Configuration of the Network Devices:

Step 1: Configure the wireless router

Create the wireless network on the wireless router

Click on the Wireless Router icon on the Packet Tracer Logical workspace to open the device configuration window.

In the wireless router configuration window, click on the GUI tab to view configuration options for the wireless router.

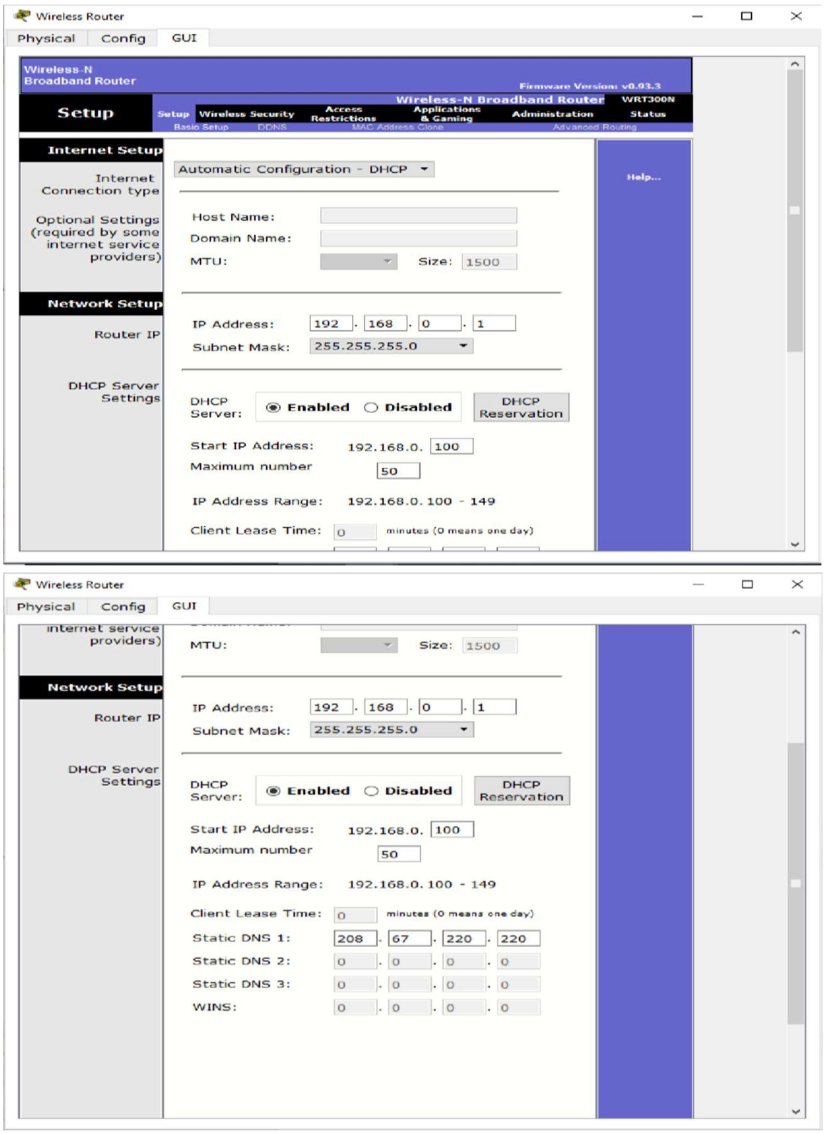
Next, click on the Wireless tab in the GUI to view the wireless settings. The only setting that needs to be changed from the defaults is the Network Name (SSID). Here, type the name "HomeNetwork"

Configure the Internet connection on the wireless router

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

Click on the Setup tab in the wireless router GUI.
In the DHCP Server settings verify that the Enabled button is selected and configure the static IP address of the DNS server as 208.67.220.220 as shown in the figure.

Click on the Save Settings tab.



Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

Step 2: Configure the laptop

Configure the Laptop to access the wireless network

Click on the Laptop icon on the Packet Tracer Logical workspace and in the laptop configuration windows select the Physical tab.

In the Physical tab, you will need to remove the Ethernet copper module and replace it with the Wireless WPC300N module.

To do this, you first power the Laptop off by clicking the power button on the side of the laptop. Then remove the currently installed Ethernet copper module by clicking on the module on the side of the laptop and dragging it to the MODULES pane on the left of the laptop window. Then install the Wireless WPC300N module by clicking on it in the MODULES pane and dragging it to the empty module port on the side of the laptop. Power the laptop back on by clicking on the Laptop power button again.

With the wireless module installed, the next task is to connect the laptop to the wireless network.

Click on the Desktop tab at the top of the Laptop configuration window and select the PC Wireless icon.

Once the Wireless-N Notebook Adapter settings are visible, select the Connect tab. The wireless network “HomeNetwork” should be visible in the list of wireless networks.

Select the network, and click on the Connect tab found below the Site Information pan

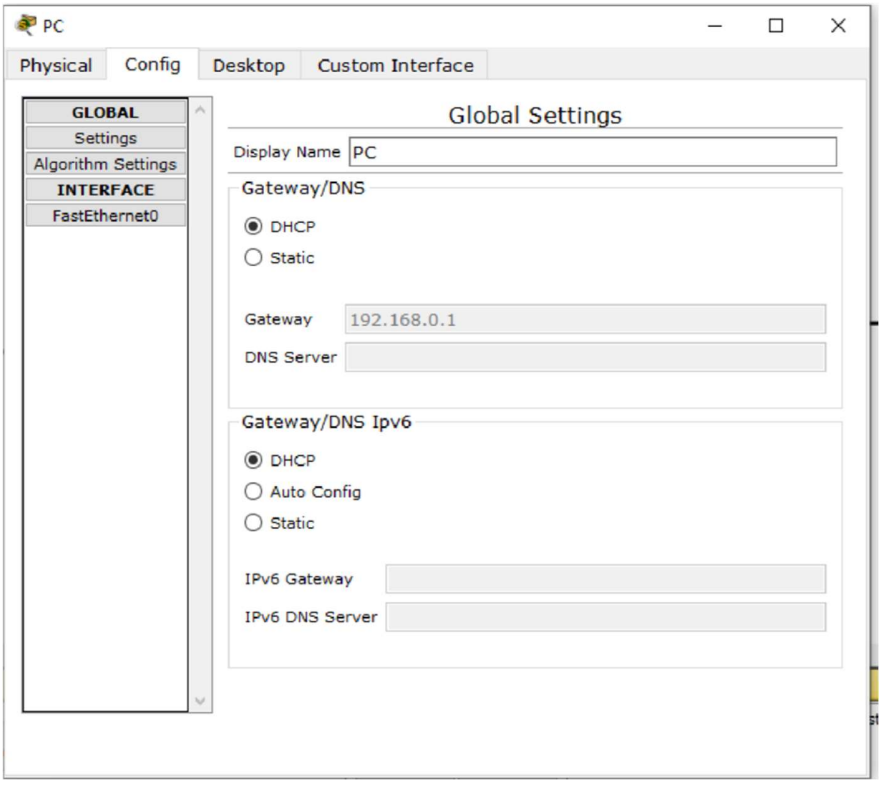
Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

Step 3: Configure the PC

Configure the PC for the wired network

Click on the PC icon on the Packet Tracer Logical workspace and select the Desktop tab and then the IP Configuration icon.

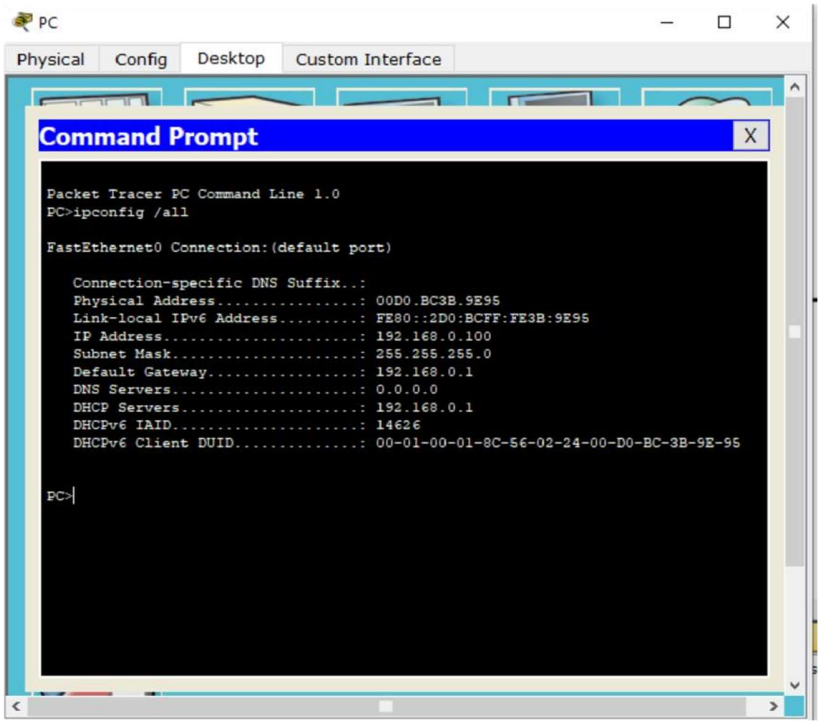
In the IP Configuration window, select the DCHP radio button as shown in the figure so that the PC will use DCHP to receive an IPv4 address from the wireless router. Close the IP Configuration window.



Click on the Command Prompt icon. Verify that the PC has received an IPv4 address by issuing the i**pconfig /all** command from the command prompt as shown in the figure. The PC should receive an IPv4 address in the 192.168.0.x range.

Name: Soham Desai **Roll No : 11**

Class: SEIT **Batch: B**



Step 4: Configure the Internet cloud

Install network modules if necessary

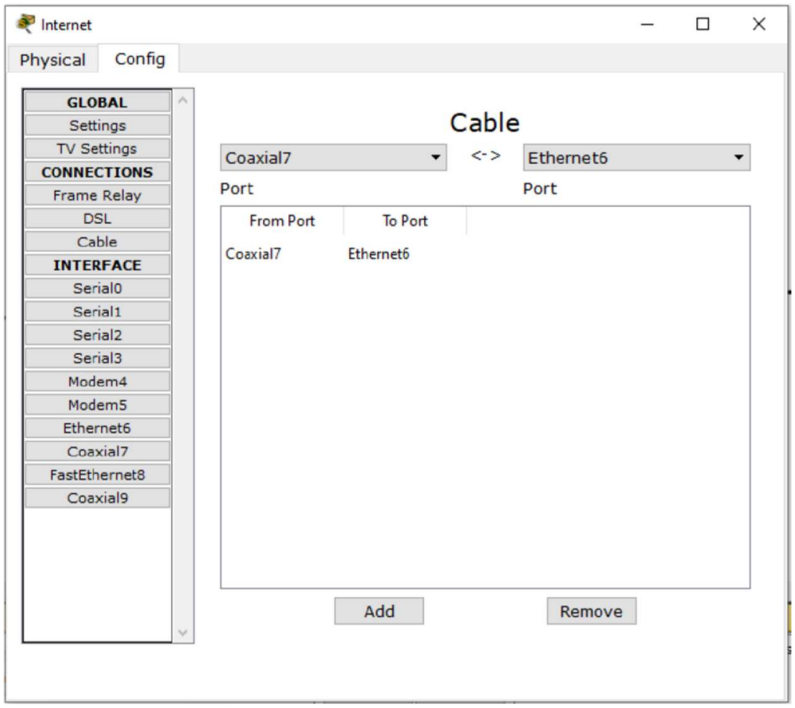
Click on the Internet Cloud icon on the Packet Tracer Logical workspace and then click on the Physical tab. The cloud device will need two modules if they are not already installed. The PT-CLOUD-NM-1CX which is for the cable modem service connection and the PT-CLOUD-NM-1CFE which is for a copper Ethernet cable connection. If these modules are missing, power off the physical cloud devices by clicking on the power button and drag each module to an empty module port on the device and then power the device back on.

Identify the From and To Ports

Click on the Config tab in the Cloud device window. In the left pane click on Cable under CONNECTIONS. In the first drop-down box choose Coaxial and in the second drop-down box choose Ethernet then click the Add button to add these as the From Port and To Port.

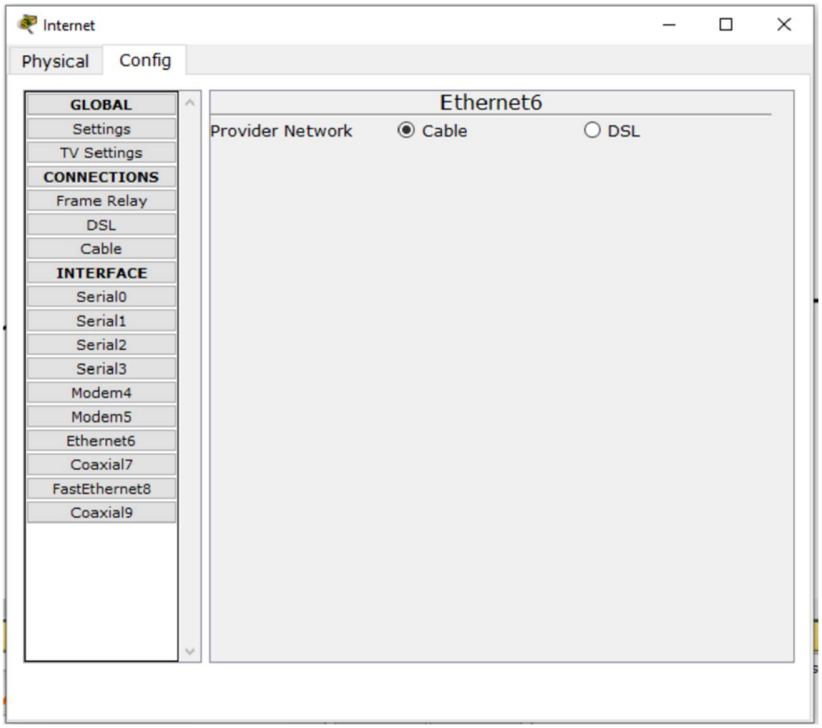
Name: Soham Desai **Roll No : 11**

Class: SEIT **Batch: B**



Identify the type of provider

While still in the Config tab click Ethernet under INTERFACE in the left pane. In the Ethernet configuration window select Cable as the Provider Network.



Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

Step 5: Configure the Cisco.com server

Configure the Cisco.com server as a DHCP server

Click on the Cisco.com server icon on the Packet Tracer Logical workspace and select the Services tab.

Select DHCP from the SERVICES list in the left pane.

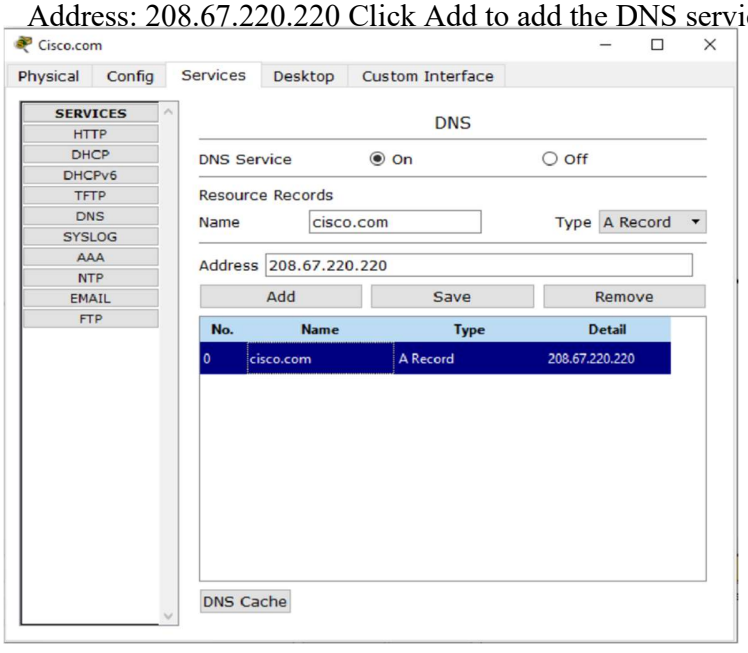
In the DHCP configuration window, configure a DHCP as shown in the figure with the following settings.

- Click On to turn the DCHP service on
- Pool name: DHCPpool
- Default Gateway: 208.67.220.220
- DNS Server: 208.67.220.220
- Starting IP Address: 208.67.220.1
- Subnet Mask 255.255.255.0
- Maximum number of Users: 50

Click Add to add the pool

Configure the Cisco.com server as a DNS server to provide domain name to IPv4 address resolution.
While still in the Services tab, select DNS from the SERVICES listed in the left pane.
Configure the DNS service using the following settings as shown in the figure.

- Click On to turn the DNS service on
- Name: Cisco.com
- Type: A Record
- Address: 208.67.220.220 Click Add to add the DNS service settings



Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

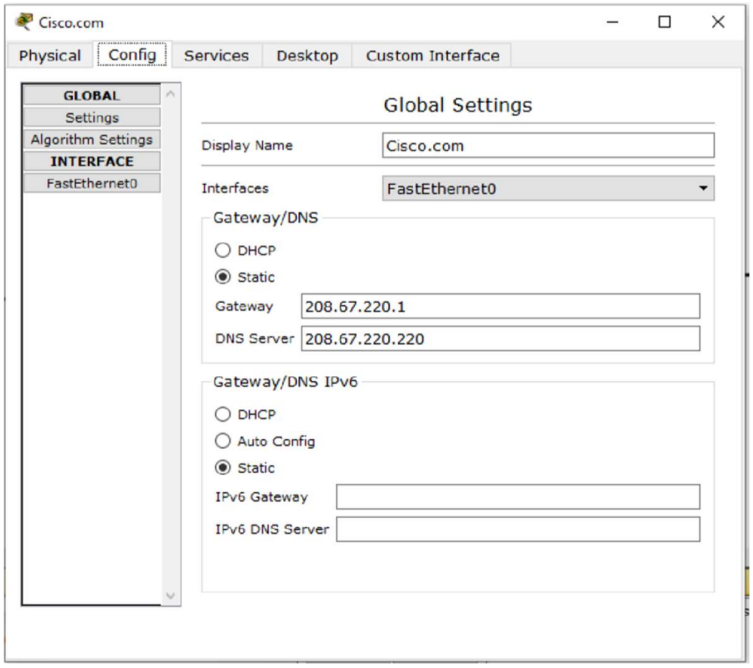
Configure the Cisco.com server Global settings.

Select the Config tab.

Click on Settings in left pane.

Configure the Global settings of the server as follows:

- Select Static
- Gateway: 208.67.220.1



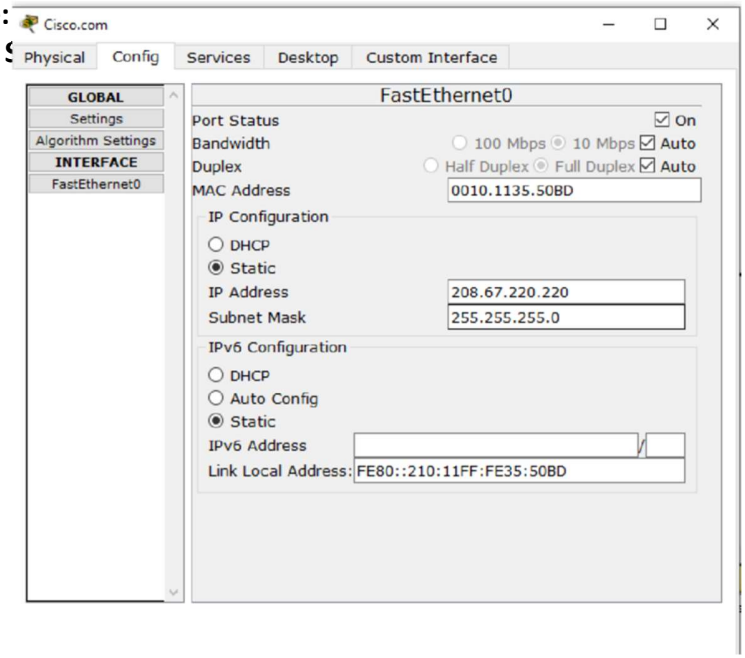
Configure the Cisco.com server FastEthernet0 Interface settings.

Click on FastEthernet in left pane of the Config tab

Configure the FastEthernet Interface settings of the server as follows:

- Select Static under IP Configuration
- IP Address: 208.67.220.220
- Subnet Mask: 255.255.255.0
- DNS Server: 208.67.220.220

Name: **Roll No : 11**
Class: S **Batch: B**



Verify the Connectivity:

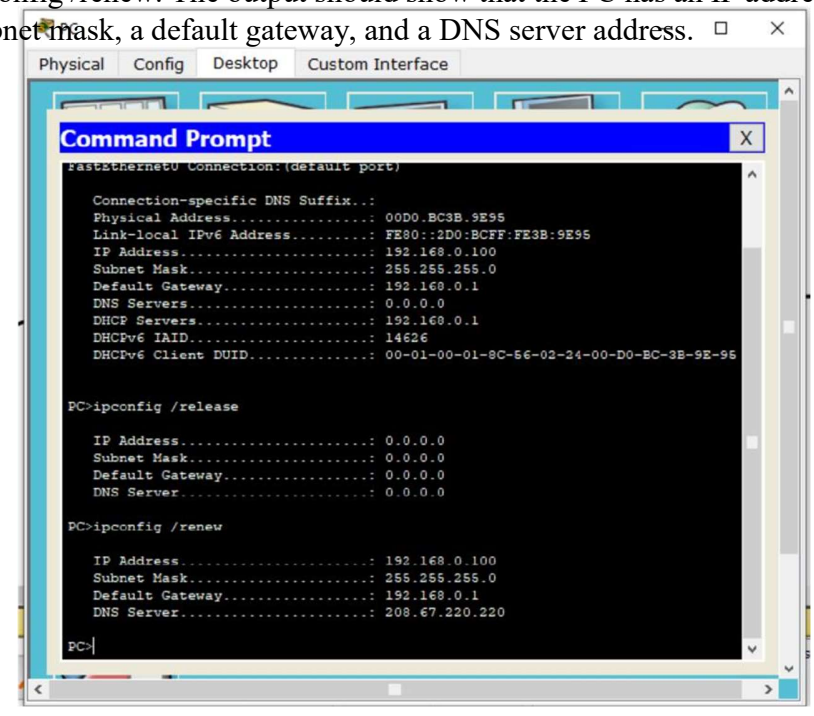
Step 1: Refresh the IPv4 settings on the PC

Verify that the PC is receiving IPv4 configuration information from DHCP.

Click on the PC on the Packet Tracer Logical workspace and then select the Desktop tab of the PC configuration window.

Click on the Command Prompt icon

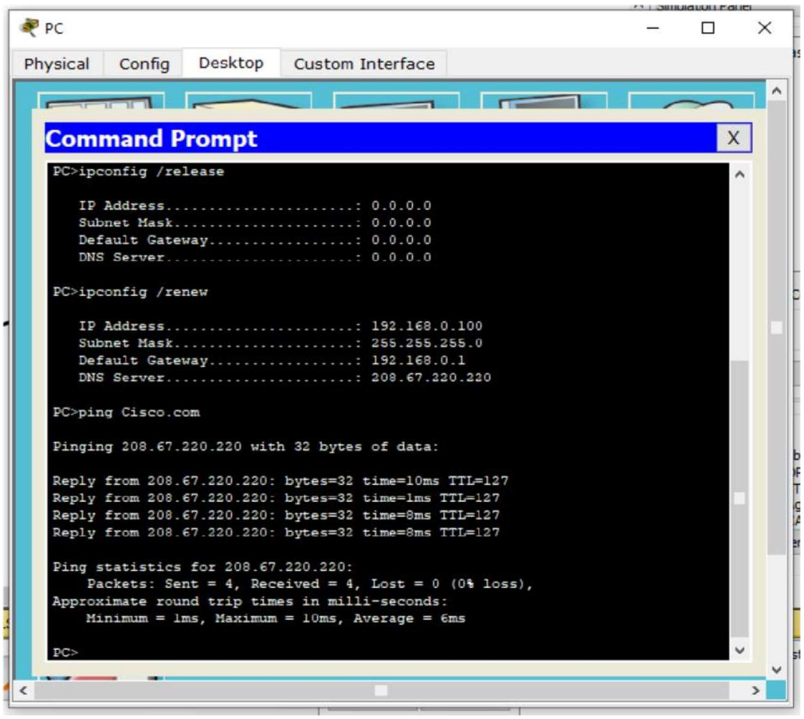
In the command prompt refresh the IP settings by issuing the commands ipconfig /release and then ipconfig /renew. The output should show that the PC has an IP address in the 192.168.0.x range, a subnet mask, a default gateway, and a DNS server address.



Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

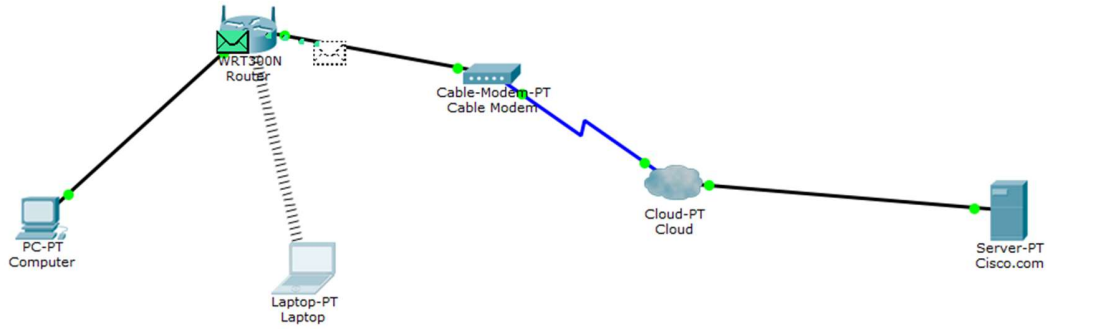
Test connectivity to the Cisco.com server from the PC

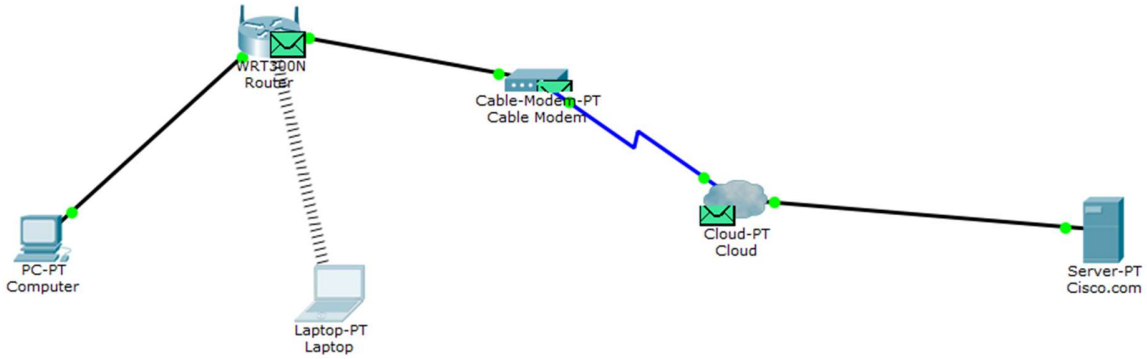
From the command prompt, issue the command ping Cisco.com. It may take a few seconds for the ping to return. Four replies should be received.



Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

OUTPUT:





CONCLUSION:

From this experiment, it is concluded that we have successfully understood the various network protocol analyser tools and analyse the network traffics using one of the network protocol analyser tools ie the CISCO PACKET TRACER. And hence, with this experiment, we have achieved Lab Outcome 5 (LO5).

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

EXPERIMENT NO. 9

AIM: Perform remote login using Telnet Server

LO6: Design a network for an organization using a network design tool

THEORY:

- The main task of the internet is to provide services to users. For example, users want to run different application programs at the remote site and transfers a result to the local site. This requires a client-server program such as FTP, SMTP. But this would not allow us to create a specific program for each demand.
- The better solution is to provide a general client-server program that lets the user access any application program on a remote computer. Therefore, a program that allows a user to log on to a remote computer. A popular client-server program Telnet is used to meet such demands. Telnet is an abbreviation for **Terminal Network**.
- Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side.

There are two types of login:

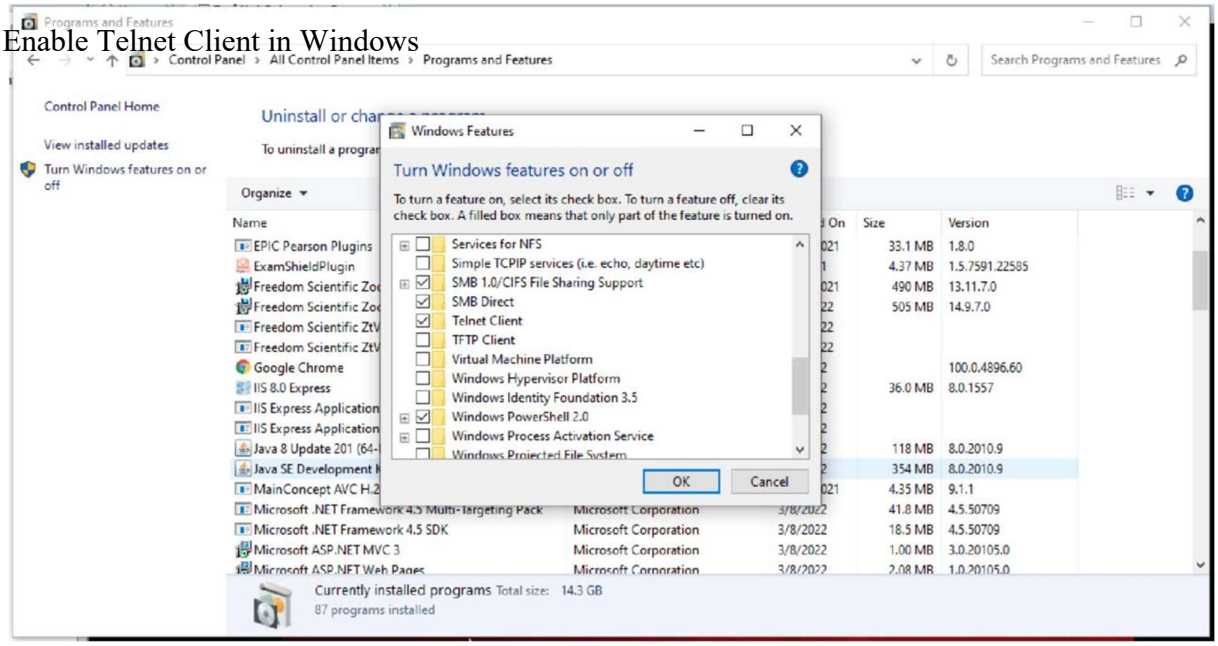
- Local Login
  - When a user logs into a local computer, then it is known as local login.
  - When the workstation running terminal emulator, the keystrokes entered by the user are accepted by the terminal driver. The terminal driver then passes these characters to the operating system which in turn, invokes the desired application program.
  - However, the operating system has special meaning to special characters. For example, in UNIX some combination of characters have special meanings such as control character with "z" means suspend. Such situations do not create any problem as the terminal driver knows the meaning of such characters. But, it can cause the problems in remote login.
- Remote login
  - When the user wants to access an application program on a remote computer, then the user must perform remote login.
  - How remote login occurs

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

At the local site
The user sends the keystrokes to the terminal driver, the characters are then sent to the TELNET client. The TELNET client which in turn, transforms the characters to a universal character set known as network virtual terminal characters and delivers them to the local TCP/IP stack

At the remote site
The commands in NVT forms are transmitted to the TCP/IP at the remote machine. Here, the characters are delivered to the operating system and then pass to the TELNET server. The TELNET server transforms the characters which can be understandable by a remote computer. However, the characters cannot be directly passed to the operating system as a remote operating system does not receive the characters from the TELNET server. Therefore it requires some piece of software that can accept the characters from the TELNET server. The operating system then passes these characters to the appropriate application program.

Enable Telnet Client in Windows

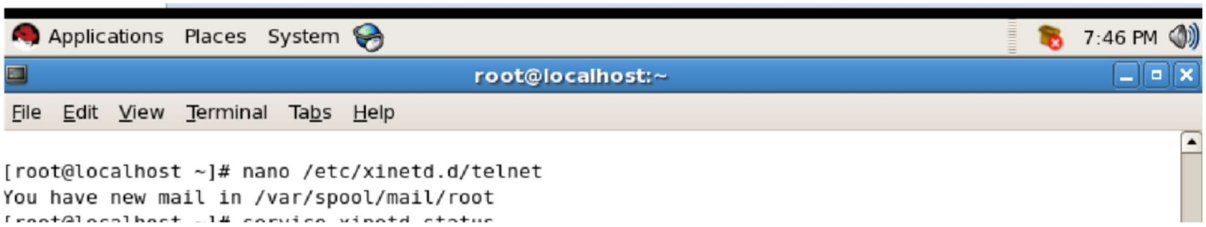


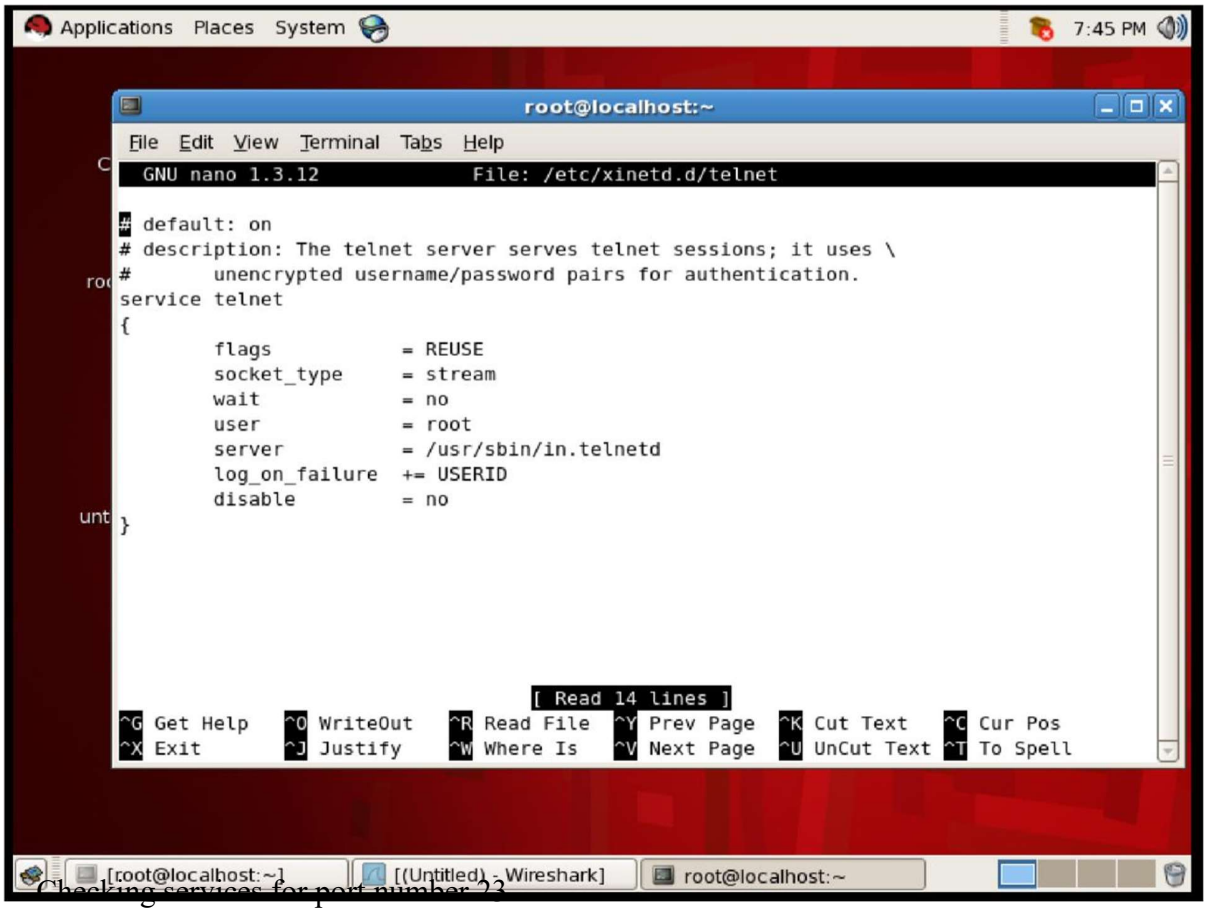
Creating user

```
New UNIX password:
BAD PASSWORD: it is too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# wireshark
```

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

Enable Telnet Server in Redhat





Checking services for port number 23

```
[root@localhost ~]# netstat -an|grep :23
```

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

Checking service status

```
[root@localhost ~]# service xinetd start
Starting xinetd:
[root@localhost ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
[root@localhost ~]# netstat -an|grep :23
tcp        0      0 0.0.0.0:23                  0.0.0.0:*
EN
[root@localhost ~]# who
root     pts/1        2022-04-07 16:38 (:0.0)
root     pts/2        2022-04-07 16:47 (:0.0)
root     pts/3        2022-04-07 16:57 (:0.0)
dhruv    pts/4        2022-04-07 17:01 (172.20.208.111)
```

Restarting the service

```
[root@localhost ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
```

Ifconfig in Redhat for IP address

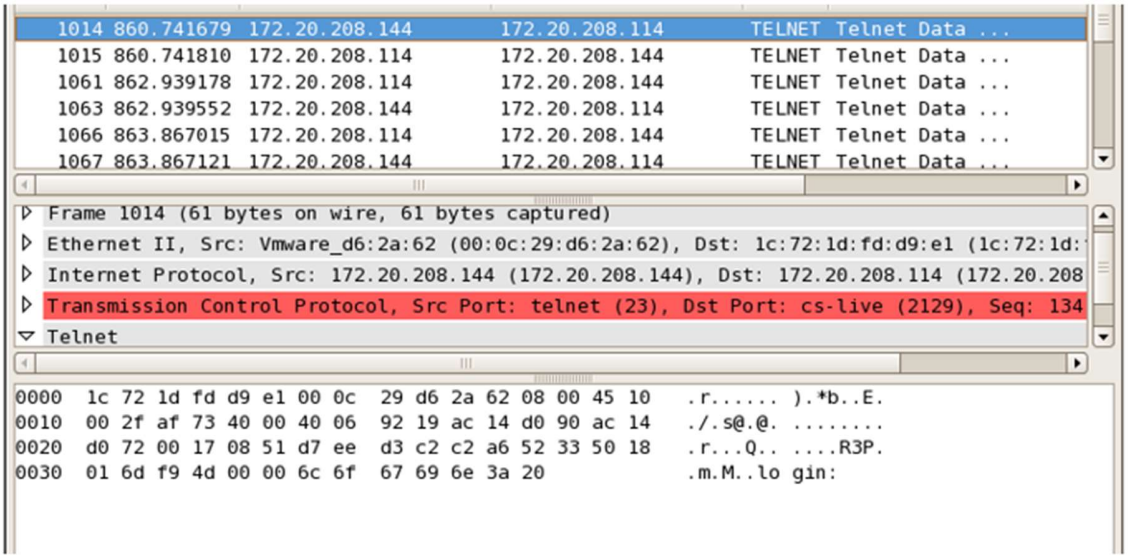
```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:A6:C1:6E
          inet addr:172.20.208.146  Bcast:172.20.208.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea6:c16e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9359 errors:0 dropped:0 overruns:0 frame:0
          TX packets:271 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1263836 (1.2 MiB)  TX bytes:23445 (22.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1280 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1280 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2035724 (1.9 MiB)  TX bytes:2035724 (1.9 MiB)
```

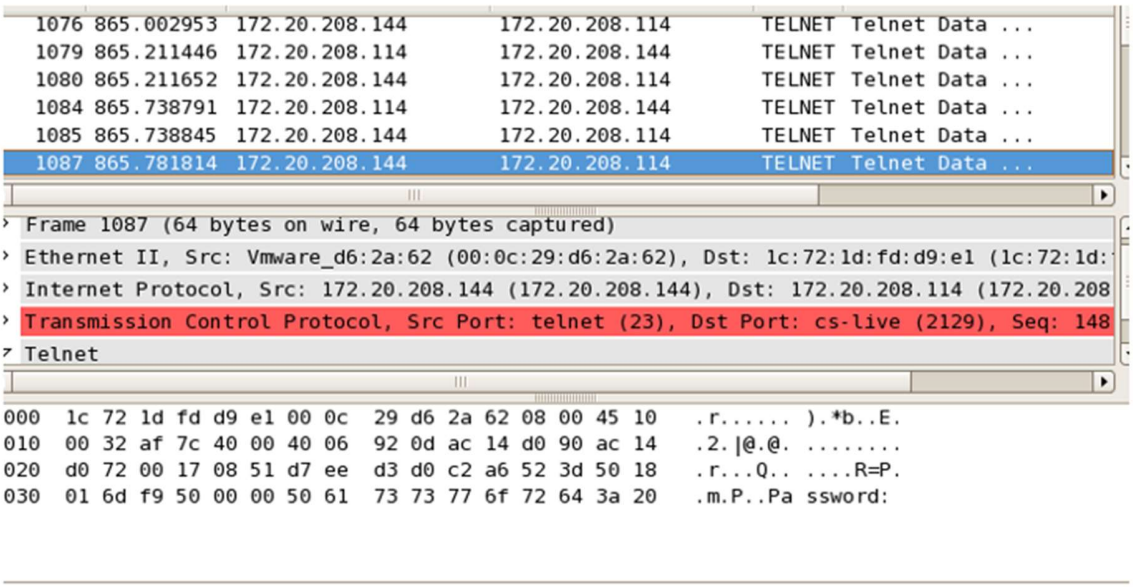
Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

Performing Remote login using Telnet Server

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5xen on an i686
login: Soham
Password:
[Soham@localhost ~]$ ls
[Soham@localhost ~]$ dir
[Soham@localhost ~]$ ls -s
total 0
[Soham@localhost ~]$ gedit soham
cannot open display:
Run 'gedit --help' to see a full list of available command line options.
[Soham@localhost ~]$ who
root     pts/1        2022-04-07 16:34 (:0.0)
root     pts/2        2022-04-07 16:36 (:0.0)
root     pts/3        2022-04-07 16:47 (:0.0)
Soham    pts/4        2022-04-07 17:01 (172.20.208.114)
Soham    pts/5        2022-04-07 17:03 (172.20.208.106)
Soham    pts/7        2022-04-07 17:03 (172.20.208.127)
Soham    pts/6        2022-04-07 17:03 (172.20.208.121)
[Soham@localhost ~]$ ls
[Soham@localhost ~]$ ls
sarvesh  tanvi
[Soham@localhost ~]$ ls
anusha  sarvesh  tanvi
[Soham@localhost ~]$ ls
anusha  sarvesh  tanvi
[Soham@localhost ~]$ mkdir Soham
[Soham@localhost ~]$ ls
anusha  sarvesh  Soham  tanvi
[Soham@localhost ~]$ dir
anusha  sarvesh  Soham  tanvi  unix
[Soham@localhost ~]$ ls
anusha  avinash  sanket  Soham  sohamwassup  unix
[Soham@localhost ~]$ rmdir sohamwassup/
[Soham@localhost ~]$ ls
anusha  avinash  sanket  Soham  unix
[Soham@localhost ~]$
```

Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**





Name: Soham Desai **Roll No : 11**
Class: SEIT **Batch: B**

After logging out:

```
[root@localhost ~]# service xinetd stop
Stopping xinetd:
```

```
Connection to host lost.
```

CONCLUSION: From this experiment, it is concluded that we have successfully performed remote login using the Telnet server. And hence, with this experiment, we have achieved Lab Outcome 6 (LO6).

Name:Soham Desai
Batch B

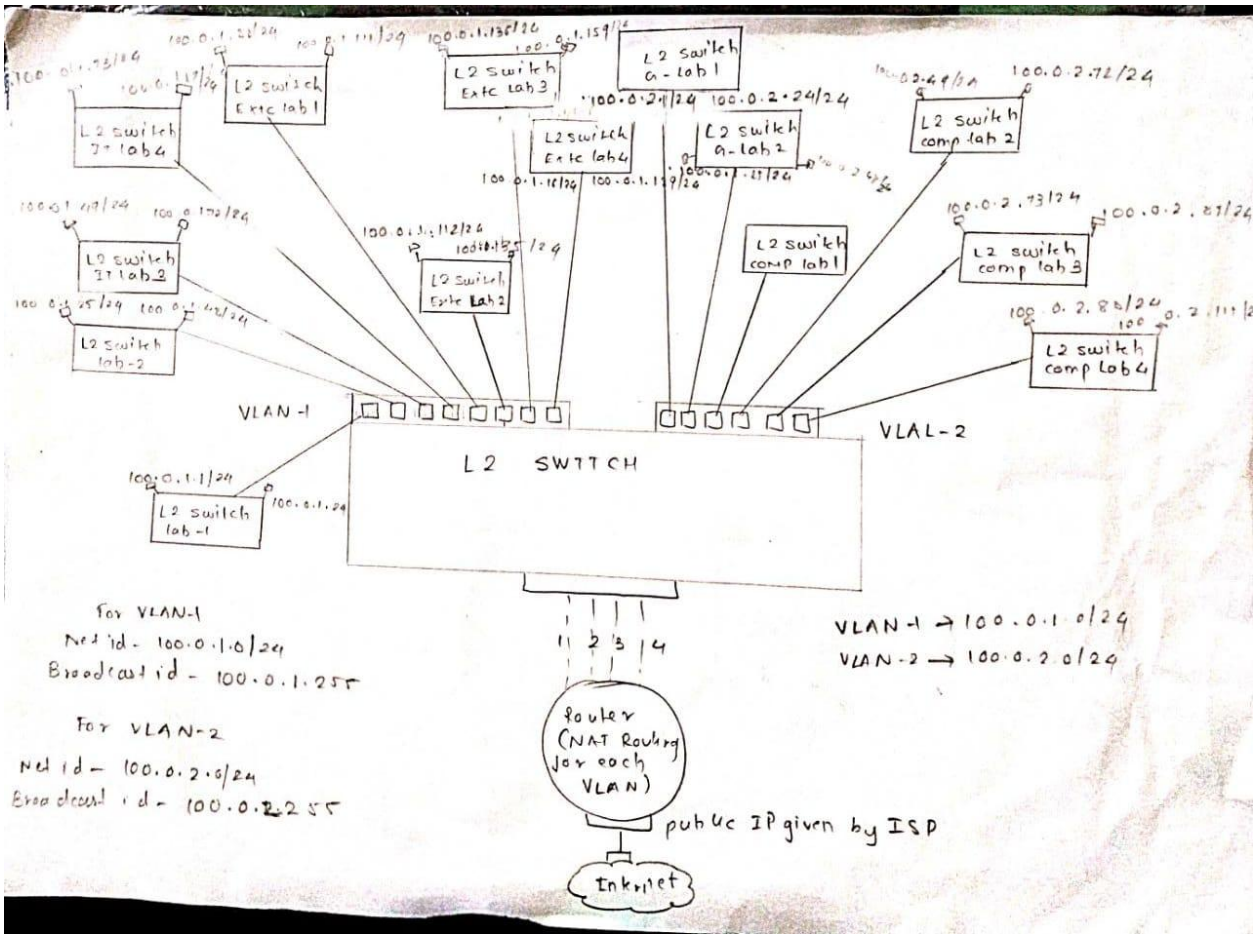
Roll no : 11
Xavier ID: 202003021

Experiment 10

Aim : Case Study: (Group Activity) Design a network for an organization using the concepts of Addressing (IP Address Assignment), Naming (DNS) and Routing.

LO 6 : Design a network for an organization using a network tool

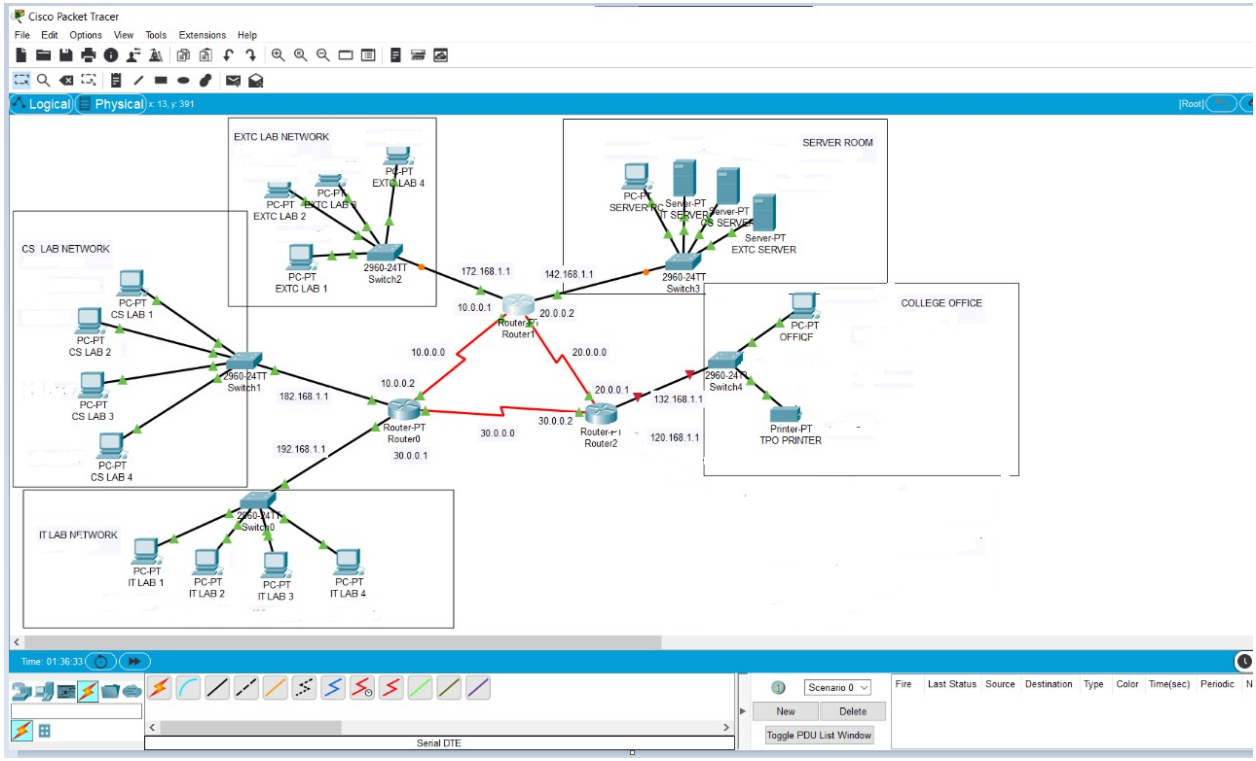
Network Design:



In our Network Design of the college we have taken two subnetworks VLAN-1 and VLAN-2 for the ground and the first floor and for second and third floor respectively

Name:Soham Desai
Batch B

Roll no : 11
Xavier ID: 202003021



This is the design of the network made and being run on the Cisco Packet Tracer and we can see that we have different switches and hubs for the respective departments.

Name:Soham Desai
Batch B
Roll no : 11
Xavier ID: 202003021

Theory:

A VLAN (virtual LAN) is a subnetwork which can group together collections of devices on separate physical local area networks (LANs). A LAN is a group of computers and devices that share a communications line or wireless link to a server within the same geographical area.

VLANs make it easy for network administrators to partition a single switched network to match the functional and security requirements of their systems without having to run new cables or make major changes in their current network infrastructure. VLANs are often set up by larger businesses to re-partition devices for better traffic management.

VLANs are also important because they can help improve the overall performance of a network by grouping together devices that communicate most frequently. VLANs also provide security on larger networks by allowing a higher degree of control over which devices have access to each other. VLANs tend to be flexible because they are based on logical connections, rather than physical.

Ports (interfaces) on switches can be assigned to one or more VLANs, enabling systems to be divided into logical groups -- based on which department they are associated with -- and establish rules about how systems in the separate groups are allowed to communicate with each other. These groups can range from the simple and practical (computers in one VLAN can see the printer on that VLAN, but computers outside that VLAN cannot), to the complex and legal (for example, computers in the retail banking departments cannot interact with computers in the trading departments).

Each VLAN provides data link access to all hosts connected to switch ports configured with the same VLAN ID. The VLAN tag is a 12-bit field in the Ethernet header that provides support for

up to 4,096 VLANs per switching domain. VLAN tagging is standardized in IEEE (Institute of Electrical and Electronics Engineers) 802.1Q and is often called *Dot1Q*.

When an untagged frame is received from an attached host, the VLAN ID tag configured on that interface is added to the data link frame header, using the 802.1Q format. The 802.1Q frame is then forwarded toward the destination. Each switch uses the tag to keep each VLAN's traffic separate from other VLANs, forwarding it only where the VLAN is configured. Trunk links between switches handle multiple VLANs, using the tag to keep them segregated. When the frame reaches the destination switch port, the VLAN tag is removed before the frame is to be transmitted to the destination device.

Multiple VLANs can be configured on a single port using a *trunk* configuration in which each frame sent via the port is tagged with the VLAN ID, as described above. The neighboring device's interface, which may be on another switch or on a host that supports 802.1Q tagging, will need to support trunk mode configuration to transmit and receive tagged frames. Any untagged Ethernet frames are assigned to a default VLAN, which can be designated in the switch configuration.

When a VLAN-enabled switch receives an untagged Ethernet frame from an attached host, it adds the VLAN tag assigned to the ingress interface. The frame is forwarded to the port of the host with the destination MAC address(media access control address). Broadcast, unknown unicast and multicast (BUM traffic) is forwarded to all ports in the VLAN. When a previously unknown host replies to an unknown unicast frame, the switches learn the location of this host and do not flood subsequent frames addressed to that host.

The switch-forwarding tables are kept up to date by two mechanisms. First, old forwarding entries are removed from the forwarding tables periodically, often a configurable timer. Second, any topology change causes the forwarding table refresh timer to be reduced, triggering a refresh.

Name:Soham Desai
Batch B
Roll no : 11
Xavier ID: 202003021

Without VLANs, a broadcast sent from a host can easily reach all network devices. Each and every device will process broadcast received frames. It can increase the CPU overhead on each device and reduce the overall network security.
In case if you place interfaces on both switches into separate VLAN, a broadcast from host A can reach only devices available inside the same VLAN. Hosts of VLANs will not even be aware that the communication took place.

Advantages and Disadvantages of VLAN

Advantages to VLAN include reduced broadcast traffic, security, ease of administration and broadcast domain confinement.

However, a disadvantage of VLANs includes the limitation of 4,096 VLANs per switching domain creates problems for large hosting providers, which often need to allocate tens or hundreds of VLANs for each customer. To address this limitation, other protocols, like VXLAN(Virtual Extensible LAN), NVGRE (Network Virtualization using Generic Routing Encapsulation) and Geneve, support larger tags and the ability to tunnel Layer 2 frames within Layer 3 (network) packets.

Application/Purpose of VLAN

Here are the important uses of VLAN:

- VLAN is used when you have 200+ devices on your LAN.
- It is helpful when you have a lot of traffic on a LAN.
- VLAN is ideal when a group of users need more security or being slow down by many broadcasts.
- It is used when users are not on one broadcast domain.
- Make a single switch into multiple switches.

Name:Soham Desai
Batch B
Roll no : 11
Xavier ID: 202003021

Internet Protocol (IP) is a set of governing rules for data packets, data format, or datagram sent through a local network or the internet. It is a connectionless and datagram-oriented protocol as it works on a dynamic computer network. An IP works without a centralized monitor or directory and never relies on a node or link. Hence, each data packet must have the source and destination's IP address and other key information to get delivered successfully.

An IP address is an address used in order to uniquely identify a device on an IP network. The address is made up of 32 binary bits, which can be divisible into a network portion and host portion with the help of a subnet mask. The 32 binary bits are broken into four octets (1 octet = 8 bits). Each octet is converted to decimal and separated by a period (dot). For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.81.100). The value in each octet ranges from 0 to 255 decimal, or 00000000 - 11111111 binary.

An IP address is the identifier that enables your device to send or receive data packets across the internet. It holds information related to your location and therefore making devices available for two-way communication. The internet requires a process to distinguish between different networks, routers, and websites. Therefore, IP addresses provide the mechanism of doing so, and it forms an indispensable part in the working of the internet. You will notice that most of the IP addresses are essentially numerical. Still, as the world is witnessing a colossal growth of network users, the network developers had to add letters and some addresses as internet usage grows.

An IP address is represented by a series of numbers segregated by periods(.). They are expressed in the form of four pairs - an example address might be 255.255.255.255 wherein each set can range from 0 to 255.

IP addresses are not produced randomly. They are generated mathematically and are further assigned by the IANA (Internet Assigned Numbers Authority), a department of the ICANN.

Name:Soham Desai Roll no : 11
Batch B Xavier ID: 202003021

ICANN stands for Internet Corporation for Assigned Names and Numbers. It is a non-profit corporation founded in the US back in 1998 with an aim to manage Internet security and enable it to be available by all.

How do IP addresses work?

Sometimes your device doesn't connect to your network the way you expect it to be, or you wish to troubleshoot why your network is not operating correctly. To answer the above questions, it is vital to learn the process with which IP addresses work.

Internet Protocol or IP runs the same manner as other languages, i.e., applying the set guidelines to communicate the information. All devices obtain, send, and pass information with other associated devices with the help of this protocol only. By using the same language, the computers placed anywhere can communicate with one another.

The process of IP address works in the following way:

1. Your computer, smartphone, or any other Wi-Fi-enabled device firstly connects to a network that is further connected to the internet. The network is responsible for giving your device access to the internet.
2. While working from home, your device would be probably using that network provided by your Internet Service Provider (ISP). In a professional environment, your device uses your company network.
3. Your ISP is responsible to generate the IP address for your device.
4. Your internet request penetrates through the ISP, and they place the requested data back to your device using your IP address. Since they provide you access to the internet, ISP's are responsible for allocating an IP address to your computer or respective device.
5. Your IP address is never consistent and can change if there occurs any changes in its internal environment. For instance, if you turn your modem or router on or off, it will change your IP address. Or the user can also connect the ISP to change their IP address.

6. When you are out of your home or office, mainly if you travel and carry your device with you, your computer won't be accessing your home IP address anymore. This is because you will be accessing the different networks (your phone hotspot, Wi-Fi at a cafe, resort, or airport, etc.) to connect the device with the internet. Therefore, your device will be allocated a different (temporary) IP address by the ISP of the hotel or cafe.

Types of IP addresses

There are various classifications of IP addresses, and each category further contains some types.

Consumer IP addresses

Every individual or firm with an active internet service system pursues two types of IP addresses, i.e., Private IP (Internet Protocol) addresses and public IP (Internet Protocol) addresses. The public and private correlate to the network area. Therefore, a private IP address is practiced inside a network, whereas the other (public IP address) is practiced outside a network.

1. Private IP addresses

All the devices that are linked with your internet network are allocated a private IP address. It holds computers, desktops, laptops, smartphones, tablets, or even Wi-Fi-enabled gadgets such as speakers, printers, or smart Televisions. With the expansion of IoT (internet of things), the demand for private IP addresses at individual homes is also seemingly growing. However, the router requires a method to identify these things distinctly. Therefore, your router produces unique private IP addresses that act as an identifier for every device using your internet network. Thus, differentiating them from one another on the network.

2. Public IP addresses

A public IP address or primary address represents the whole network of devices associated with it. Every device included within with your primary address contains their own private IP address.

Name:Soham Desai Roll no : 11
Batch B Xavier ID: 202003021

ISP is responsible to provide your public IP address to your router. Typically, ISPs contains the bulk stock of IP addresses that they dispense to their clients. Your public IP address is practiced by every device to identify your network that is residing outside your internet network.

Public IP addresses are further classified into two categories- dynamic and static.

- **Dynamic IP addresses**

  As the name suggests, Dynamic IP addresses change automatically and frequently. With this types of IP address, ISPs already purchase a bulk stock of IP addresses and allocate them in some order to their customers. Periodically, they re-allocate the IP addresses and place the used ones back into the IP addresses pool so they can be used later for another client. The foundation for this method is to make cost savings profits for the ISP.

- **Static IP addresses**

  In comparison to dynamic IP addresses, static addresses are constant in nature. The network assigns the IP address to the device only once and, it remains consistent. Though most firms or individuals do not prefer to have a static IP address, it is essential to have a static IP address for an organization that wants to host its network server. It protects websites and email addresses linked with it with a constant IP address.

In reality, other versions were defined, from versions 1 to 9, but only versions 4 and 6 found widespread use. Version 1 and 2 were TCP protocol names in 1974 and ’77 to separate the IP specification at that time. Moreover, version 3 was introduced in 1978, where v3.1 was the first ever version in which TCP got separated from IP. Next, version 5 that surfaced in 1979 was the experimental protocol – Internet Stream Protocol.

IPv6 is a combination of various versions – v6, v7, v8, and v9.

Name:Soham Desai Roll no : 11
Batch B Xavier ID: 202003021

The Domain Name System (DNS) underpins the web we use every day. It works transparently in the background, converting human-readable website names into computer-readable numerical IP addresses. DNS does this by looking up that information on a system of linked DNS servers across the Internet. However, different DNS servers can behave differently in terms of speed and security. So, let's take a look at how DNS works and what you can do to make sure it's working its best for you.

Domain Names and IP Addresses

Domain names are the human-readable website addresses we use every day. For example, Google's domain name is google.com. If you want to visit Google, you just need to enter google.com into your web browser's address bar.

DNS Servers

DNS servers match domain names to their associated IP addresses. When you type a domain name into your browser, your computer contacts your current DNS server and asks what IP address is associated with the domain name. Your computer then connects to the IP address and retrieves the right web page for you.

The DNS servers you use are likely provided by your Internet service provider (ISP). If you're behind a router, your computer may be using the router itself as its DNS server, but the router is forwarding requests to your ISP's DNS servers.

Some viruses and other malware programs can change your default DNS server to a DNS server run by a malicious organization or scammer. This malicious DNS server can then point popular websites to different IP addresses, which could be run by scammers.

Name:Soham Desai Roll no : 11
Batch B Xavier ID: 202003021

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.

In packet switching networks, routing is the higher-level decision making that directs network packets from their source toward their destination through intermediate network nodes by specific packet forwarding mechanisms. Packet forwarding is the transit of network packets from one network interface to another. Intermediate nodes are typically network hardware devices such as routers, gateways, firewalls, or switches. General-purpose computers also forward packets and perform routing, although they have no specially optimized hardware for the task.

The routing process usually directs forwarding on the basis of routing tables. Routing tables maintain a record of the routes to various network destinations. Routing tables may be specified by an administrator, learned by observing network traffic or built with the assistance of routing protocols.

Routing, in a narrower sense of the term, often refers to IP routing and is contrasted with bridging. IP routing assumes that network addresses are structured and that similar addresses imply proximity within the network. Structured addresses allow a single routing table entry to represent the route to a group of devices. In large networks, structured addressing (routing, in the narrow sense) outperforms unstructured addressing (bridging). Routing has become the dominant form of addressing on the Internet. Bridging is still widely used within local area networks.

Name:Soham Desai Roll no : 11
Batch B Xavier ID: 202003021

The following protocols help data packets find their way across the Internet:

IP: The Internet Protocol (IP) specifies the origin and destination for each data packet. Routers inspect each packet's IP header to identify where to send them.

BGP: The Border Gateway Protocol (BGP) routing protocol is used to announce which networks control which IP addresses, and which networks connect to each other. (The large networks that make these BGP announcements are called autonomous systems.) BGP is a dynamic routing protocol.

The below protocols route packets within an AS:

OSPF: The Open Shortest Path First (OSPF) protocol is commonly used by network routers to dynamically identify the fastest and shortest available routes for sending packets to their destination.

RIP: The Routing Information Protocol (RIP) uses "hop count" to find the shortest path from one network to another, where "hop count" means number of routers a packet must pass through on the way. (When a packet goes from one network to another, this is known as a "hop.")

Other interior routing protocols include EIGRP (the Enhanced Interior Gateway Routing Protocol, mainly for use with Cisco routers) and IS-IS (Intermediate System to Intermediate System).

Name : Soham Desai
Batch B

Roll No : 11
XIE ID:202003021

Report for Lab 3-1: UDP

Name: Soham Desai	**Student ID:** 202003021	**Date:** 21/03/2022

1

a. Source port number: **63985**

b. Destination port number:**hostman(5355)**

c. Total length of user diagram **41**

d. Length of data **33**

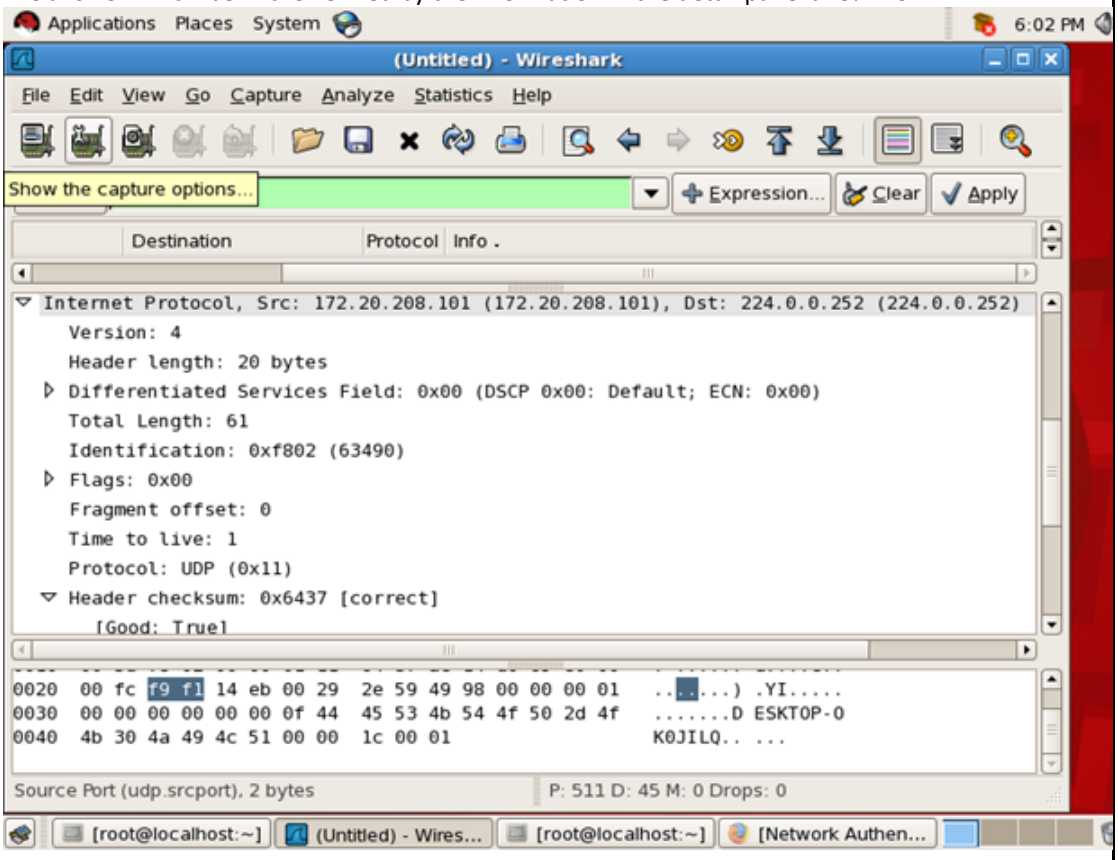
e. Is the packet from client or server? **Both**

f. Application layer protocol: **UDP**

g. Is checksum calculated? **Yes**

2

Are answer in number 1 are verified by the information in the detail pane lane? **YES**



3

Source and destination IP addresses in the query message:
fe80::ad7f:681f:79ac:cf3f(source)
ff02::1:3(destination)

Source and destination IP addresses in the response) message:
63985(source)

	hostman(5355) (destination) Relation between IP addresses: **Source IP address: the IP packet field containing the IP address of the workstation from which it came.** **Destination IP address: the IP packet field containing the IP address of the workstation to which it is addressed.**
4	Source and destination port number in the query message: **63985 (source)** **hostman(5355) (destination)** Source and destination port number in the response message: **fe80::ad7f:681f:79ac:cf3f(source)** **ff02::1:3(destination)** Relation between port numbers: **Source port number: the port number field containing the port number of the workstation from which packet is sent.** **Destination port number: the port number field containing the port number of the workstation to which it is a packet is received.** Which port number is well-known? **0 - 1023**
5	The length of the first UDP packet: **76** How many bytes of payload are carried by the first UDP packet? **total length - header length = 76-20 = 56**
6	Number of bytes in the DNS message:**4** Does the count agree with the answer to question 5? **No**
7	Is the checksum calculated for the first UDP packet? **Yes** Value of the checksum: **0xc43e**

Name : Soham Desai
Batch B
Roll No : 11
XIE ID:202003021

Report for LAB 3-2: TCP

Name: Soham Desai	**Student ID:** 202003021	**Date:** 21/03/2022

	Part I
1	Socket addresses: **Source: 142.168.195.128** **Destination: 142.250.66.4**
2	Set flags: **PSH and ACK**
3	Sequence number and acknowledgement number: **2805 and 893**
4	Window size: **5840**

	Part II
1	Set flag in HTTP GET message: **ACK, PSH**
2	Number of bytes transmitted by the HTTP GET message: **235**
3	Acknowledgement frequency: **22.831829570 s - 22.826527150 = 0.00530242 s = 5.30242 ms** Corresponding rule: **An acknowledgement be sent for at least every other full-size segment, and that no more than 500ms expire before any segment is acknowledged.**
4	Number of bytes transmitted by each packet: **60** Relation to sequence and acknowledgement Number: **1:1**
5	Original window sizes: **5840** Are these numbers expected? **Yes** How window sizes change? **The window size keeps increasing as long as the receiver sends acknowledgments for all our segments or when the window size hits a certain maximum limit. When the receiver doesn't send an acknowledgment within a certain time period (called the round-trip time) then the window size will be reduced.**

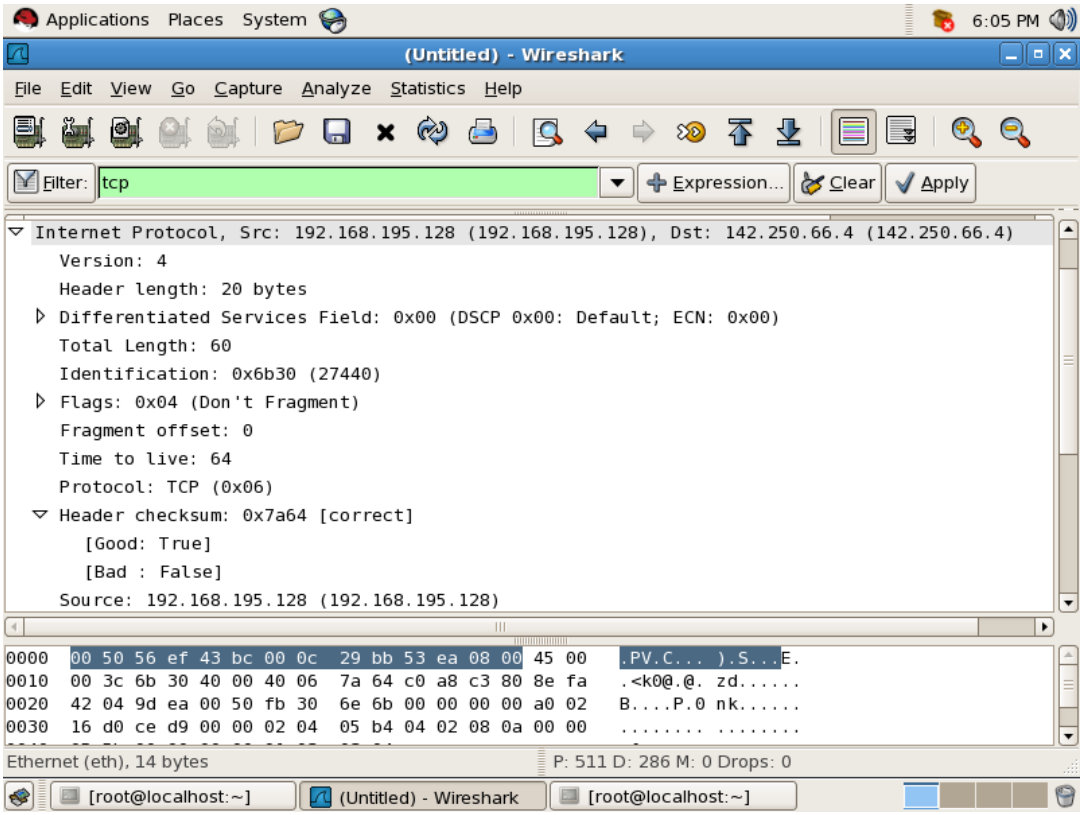
6	How the window size is used in flow control? **Flow control is accomplished by the receiver sending back a window to the sender. The size of this window, called the receive window, tells the sender how much data to send. Often, when the client is saturated, it might not be able to send back a receive window to the sender to signal it to slow down transmission.**
7	Purpose of the HTTP OK message: **The request has succeeded. The meaning of the success depends on the HTTP method:** **GET: The resource has been fetched and is transmitted in the message body.** **HEAD: The entity headers are in the message body.** **PUT or POST: The resource describing the result of the action is transmitted in the message body.** **TRACE: The message body contains the request message as received by the server.**

Part III	
1	Number of TCP segments exchanged for connection termination: **N/A**
2	Which end point started the connection termination phase? **N/A**
3	Flags sets in each of the segments used for connection termination: **N/A**

Name : Soham Desai
Batch B

Roll No : 11
XIE ID:202003021

Part IV

1	a. Source port number: **40426**	b. Destination port number: **http (80)**
	c. Sequence number: **2805**	d. Acknowledgment number: **897**
	e. Header length: **40 Bytes**	f. Set flags: **ACK and PSH**
	g. Window size: **5840**	h. Urgent pointer: **0**
2	Are answer in the question number 1 verified by the information in the detail pane lane? **Yes** 
3	Does any of the TCP packet headers carry options? **Yes** Explain **TCP options fields are at the end of the header and they are multiple of 8 bits. If any of bits remains from options, they can be filled by padding, with zeros. This options are MSS, Window Scaling, Selective Acknowledgements, Timestamps and Nop. The size of this field is changable according tot the used options. Generally these options are used during the 3-way handshake but others can be use during normal TCP session.**	

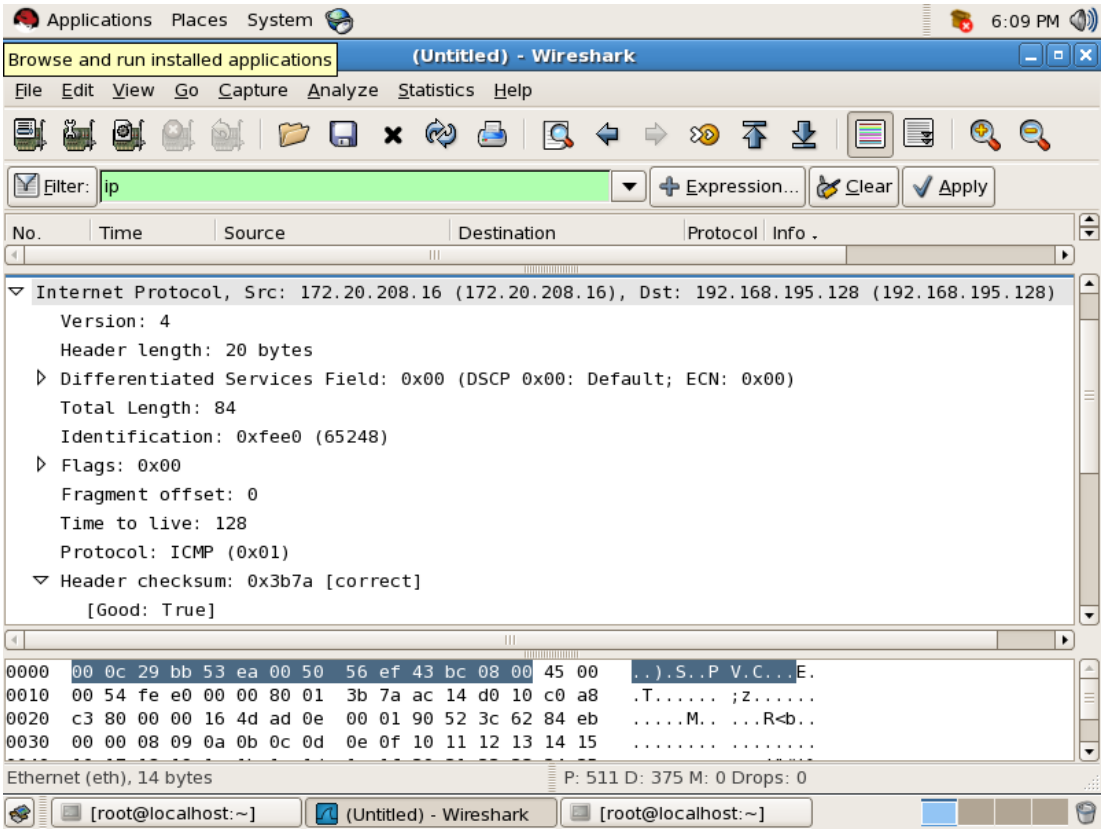
4	Size of a TCP packet with no option: **157** Size of a TCP packet with options: **169**
5	Is window size in any of the TCP packet zero? Explain: **When a client (or server – but it is usually the client) advertises a zero value for its window size, this indicates that the TCP receive buffer is full and it cannot receive any more data. It may have a stuck processor or be busy with some other task, which can cause the TCP to receive buffer to fill. Zero Windows can also be caused by a problem within the application, where the TCP buffer is not being retrieved.**

Name : Soham Desai
Batch B

Roll No : 11
XIE ID:202003021

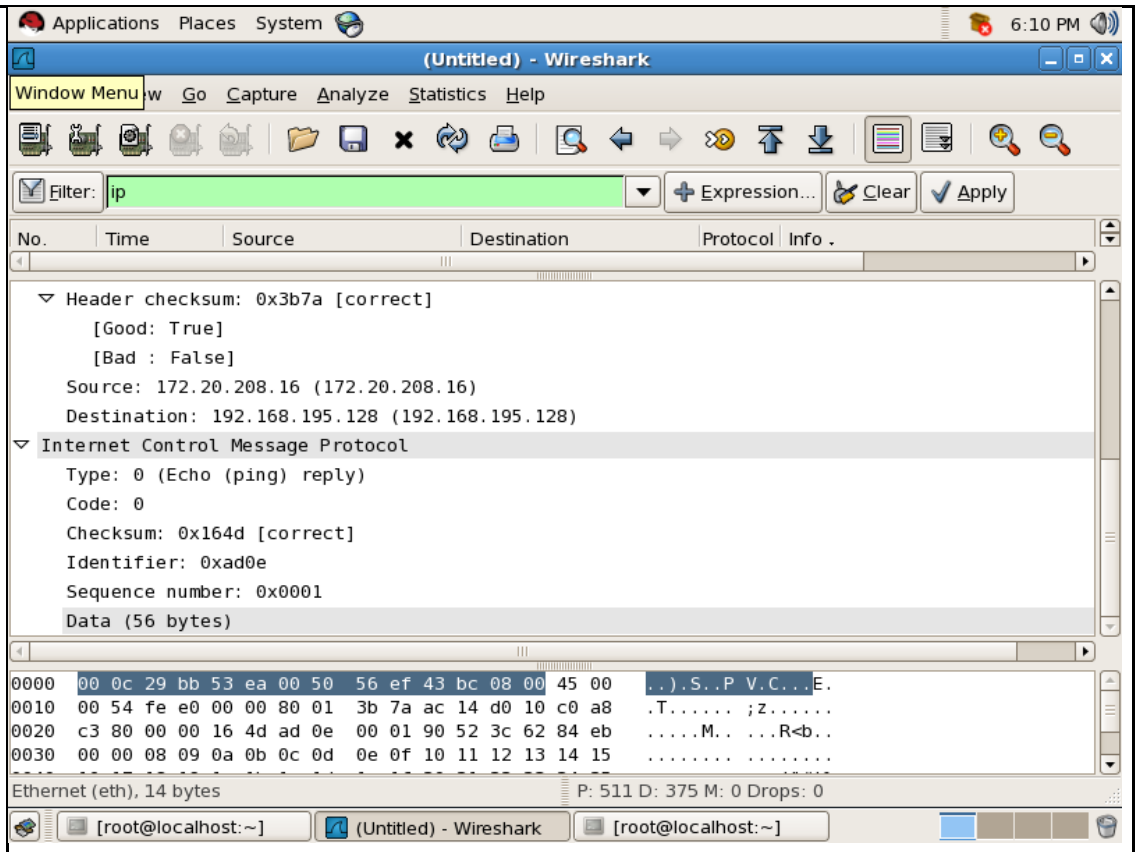
Report for Lab 4-1: IP

Name: Soham Desai	**Student ID:** 202003021	**Date:** 21/03/2022

1	a. IP version:**4**	b. Header length:**20 bytes** Number of bytes in the header:**20**
	c. Service type: **Differentiated**	d. Total length:**84**
	e. Identification: **0xfee0 (65248)**	f. Flags:**0x00**
	g. Fragmentation offset: **0**	h. TTL:**128**
	i. Upper layer protocol: **ICMP(0x11)**	j. Checksum:**0x3b7a [correct]**
	k. Source IP address:**172.20.208.16**	l. Destination IP address:**192.168.195.128**
2	Are answers to question 1 verified by the information in the packet detail pane? **Yes** 	

Name : Soham Desai
Batch B

Roll No : 11
XIE ID:202003021

	
3	If the checksum in the packet detail pane is marked correct, can we conclude that the IP payload is not corrupted? Explain. **YES, IPv4 has a header checksum. It only covers the header, so any corruption of the IP payload is not detected. Due to each router having to update the TTL field, the checksum has to be recomputed at each hop.**
4	Is the datagram fragmented? Explain. **NO, Since the don't fragment flag is set, the datagram is not fragmented.**
5	Does source or destination address belong to one of the special addresses? If yes which one? **YES, they are ipv4 addresses and are also called the loopback addresses**
6	Number of bytes of data in the IP payload: **total length - header length = 84-20 = 64**

NAME Soham Desai ROLL NO-11
BATCH-B XIE ID - 202003021

Xavier Institute of Engineering

Department of Information Technology

Subject: CNND (SE SEM IV)

AY- 2021-22

Lab Assignment 2

Q. 1. What Is Ping Utility? What is Ping of Death attack? How is it performed? Discuss methods to mitigate this attack. (LO1)

Ans:

- A ping (Packet Internet or Inter-Network Groper) is a basic Internet program that allows a user to test and verify if a particular destination IP address exists and can accept requests in computer network administration. The acronym was contrived to match the submariners' term for the sound of a returned sonar pulse.
- A Ping of death (PoD) attack is a denial-of-service (DoS) attack, in which the attacker aims to disrupt a targeted machine by sending a packet larger than the maximum allowable size, causing the target machine to freeze or crash.
- A correct Internet Protocol version 4 (IPv4) packet is formed of 65,535 bytes, and most legacy computers cannot handle larger packets. Sending a ping larger than this violates the IP, so attackers send packets in fragments which, when the targeted system attempts to reassemble, results in an oversized packet that can cause the system to crash, freeze, or reboot.
- To mitigate the losses caused due to Ping Of Death Attack, various websites are strengthening their inbuilt security parameters to block the ICMP ping messages.
- An alternative method can be blocking the fragmented pings from your device. This would prevent the size of packet recombination from exceeding the permissible levels. It means that the pings can pass your system in an unhindered manner.
- Increasing the memory buffer can be another way to stay protected from the Ping Of Death Attacks.

NAME Soham Desai ROLL NO-11
BATCH-B XIE ID - 202003021

Q.2. What is Network Simulation? Discuss various types of simulators available. Compare network simulator and network emulation. (LO2)

Ans:

A network simulator is a software program that can predict the performance of a computer network. Since communication networks have become too complex for traditional analytical methods to provide an accurate understanding of system behavior, network simulators are used. In simulators, the computer network is modeled with devices, links, applications, etc., and the network performance is reported. Simulators come with support for the most popular technologies and networks in use today such as 5G, Internet of Things (IoT), Wireless LANs, mobile ad hoc networks, wireless sensor networks, vehicular ad hoc networks, cognitive radio networks, LTE etc.

The different types of network simulators/ network simulation tools are open source and commercial

- Network Simulator version 2 (NS-2)
- Ns3
- Netkit
- Marionnet
- JSIM (Java-based Simulation)
- OPNET
- QualNet
- The open-source simulators are Marrionet, Netkit, NS2, JSIM
- The commercial simulators are OPNET and QualNet

Network simulators:
On a basic level, a network simulator uses mathematical formulas to create a theoretical and entirely virtual model of a network. Simulators are software solutions and different types are available for different applications. While used primarily for research and educational purposes, they can also act as crucial testing tools in the design and development of a network.

Simulators, such as ns-3, are used to simulate networking and routing protocols. OPNET, which was acquired by Riverbed in 2012 and applied to their SteelCentral product line, also provided a standalone simulation environment.

Both of these network simulators use *discrete event simulation* which chronologically queues and processes events like data flow. This allows a network architect or engineer to build and evaluate an experimental model of a network, including its topology and application flow. Since a variety of theoretical scenarios can be introduced to a network where anything can be built and applied, performance can be hypothesized before the network itself has even been implemented within the real-world.

Network emulators:
A network emulator, also referred to as a *WAN emulator*, is used to test the performance of a real network. These devices can also be used for such purposes as quality assurance, proof of concept, or troubleshooting. Available as hardware or software solutions, a network emulator allows network architects, engineers, and developers to accurately gauge an application's responsiveness, throughput, and quality of end-user experience prior to applying making changes or additions to a system.

By physically placing it between two LAN segments, a network emulator can accurately replicate a client/server WAN connection without the need for a router, modem, or even live traffic. It can then be configured to manipulate bandwidth constraints and apply impairments, such as packet loss, delay, and jitter, to the mirrored network. Latency can be specified to emulate the transfer of data over large distances and applications behave and respond as if they're actually physically separated. Application performance and end-user experience can then be observed, tested, and validated under such conditions in real-time.

Software solutions, such as NetEm, which comes prepackaged within the Linux kernel, are ideal for testing at low data rates, but are limited by the testing machines on which they're run.

Q.3. What is network performance? Discuss various ways to measure network performance. Also list tools available to measure network performance. (LO3)

Ans:

- Network performance is the analysis and review of collective network statistics, to define the quality of services offered by the underlying computer network.
- It is a qualitative and quantitative process that measures and defines the performance level of a given network. It guides a network administrator in the review, measure and improvement of network services.
- When optimizing network performance there are important metrics that must be measured. Some common metrics used to measure network traffic performance include latency, packet loss indicators, jitter, bandwidth, and throughput.
- Top ten network monitoring tools:

  While network monitoring tools focus on aspects like performance monitoring, fault monitoring, and account monitoring, they're also used to examine components such as applications, email servers, and more. While there are several network monitoring tools available in the market, choosing the right device with in-depth research and tracking capability is challenging.

  - SolarWinds Network Performance Monitor
  - Nagios
  - Zabbix
  - Spiceworks
  - Icinga
  - PRTG Network Monitor

- Site 24×7
- Atera
- ManageEngine OpManager
- Zenoss Cloud

Q.4. What is Socket programming? Discuss in detail. (LO4)

Ans:

- A socket is a communications connection point (endpoint) that you can name and address in a network. Socket programming shows how to use socket APIs to establish communication links between remote and local processes.
- The processes that use a socket can reside on the same system or different systems on different networks. Sockets are useful for both stand-alone and network applications. Sockets allow you to exchange information between processes on the same machine or across a network, distribute work to the most efficient machine, and they easily allow access to centralized data. Socket application program interfaces (APIs) are the network standard for TCP/IP. A wide range of operating systems support socket APIs. i5/OS sockets support multiple transport and networking protocols. Socket system functions and the socket network functions are threadsafe.
- Programmers who use Integrated Language Environment® (ILE) C can refer to this topic collection to develop socket applications. You can also code to the sockets API from other ILE languages, such as RPG.
- The Java™ language also supports a socket programming interface.

Q.5. Why is network design important? Discuss various tools available for network design. (LO6)

Ans:

Networks that are part of a designed plan rather than those who are just pieced together, perform typically better due to taking into consideration the needs of the network. A good network design will be more robust and allow for better performance overall. Good networks work quickly and efficiently, providing the best platform for all the applications that you wish to use. If there is lag or delayed response time, then any work which is trying to be done by employees will be affected, this is one of the reasons why it's important to consider the network design first before implementing any changes or when starting the initial installation.

- **Strong Networks Mean Scalability**

If you have a strong network, then when you need to scale you won't have to think about a re-design, rather you can increase the numbers of users and computers more easily from the original design and also minimise any potential downtime. We will verify what structure you need for the data cabling and provide quality feedback as to what installation is best. Regulated hardware and

software that is used bi the network will also ensure that you have networks that need less maintenance or updates. If you are allowing for future growth, then we can build different options into the design.

- **Improved Service Levels**

By having a fresh design and installation of cabling for your business's network, you will find that you always enjoy prompt service and performance. The range of reaction times to tasks will be much more effective and avoid in instances like loss within a client's session. Users like to login and expect the business's systems to work, so to that effect having a well-planned network design will minimise security issues as well as an increase in speed and performance.

- **Cost-effective LAN Design**

There is a chance to considerably save on costs when it comes to doing a completely fresh network design and installation, compared to working on patches of cabling. The cost of design and install could mean that you invariably save money on having to buy new hardware. The ultimate goal of cost-effective network installation from AEL Systems is that you have the type of controlled network that you need, one that provides security against risk and improves usability at the same time. We will evaluate any potential risks before proceeding with the design and installation. We will also try to look at where we can maximise performance of your currently installed data cabling system.

- **Security of Networks**

We should always consider the security of the networks, with the goal being to avoid any attacks being repeated throughout the business. A disaster recovery plan is also part of any network design and there should be provisions for backup power where needed. On a side note, and pertaining to the safety of business's data, data should also be backed up daily to minimise risk of loss should anything happen to the systems due to fire or other security breach.

1. **SolarWinds Network Topology Mapper**
2. **CADE**
3. **Dia Diagram Editor**
4. **Microsoft Visio**