Velammal College of Engineering and Technology

Viraganoor, Madurai-625 009

Department of Computer Science and Engineering

Academic Year: 2021-22 (ODD Semester)

Lab Manual

CS8581 – NETWORKS LABORATORY

Prepared by,

Dr. G VINOTH CHAKKARAVARTHY

Associate Professor/ CSE

Verified by,

Dr.P.Alli

HoD/CSE

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Vision and Mission of the Department and Institute

Institute Vision and Mission

Vision

To emerge and sustain as a Center of Excellence for Technical and Managerial Education upholding social values

Mission

Our aspirants are

- Imparted with comprehensive, innovative and value-based education.
- Exposed to technical, managerial and soft skill resources with emphasis on research, and professionalism.
- Inculcated with the need for a disciplined, happy, married and peaceful life.

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

- Heightening the knowledge of the faculty in recent trends through continuous development programmes.
- Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.
- Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Department Vision and Mission
Vision
To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.
Mission
•Heightening the knowledge of the faculty in recent trends through continuous development programs.
•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.
•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Program Educational Outcome

The PEO statements up to Regulation 2013 are

PEO 1	Graduates work productively as a successful employee with problem solving skills, core computing skills, professional ethics and soft skills with social awareness.
PEO 2	Graduates participate in lifelong learning through successful completion of higher education with Research and Development.
PEO 3	Graduates become successful entrepreneurs with determination, development, self-reliance, leadership, ethic & moral values to exploit employability.

Program Specific Outcomes

The PSOs statements up to Regulation 2013 are

PSO1. Apply Engineering knowledge to analyze, design and develop computerized solutions by selecting appropriate technology for solving complex problems**.**

PSO2. Inculcate the knowledge of Engineering and Management principles among the team members to manage projects effectively**.**

Program Outcomes for R2013

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

modern engineering and IT tools, including prediction and modelling to complex engineering activities, with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with t h e society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Table of Contents

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY
Department Of Computer Science and Engineering
Dos and Don'ts

1. Wearing T-shirts and in formals are strictly prohibited.
2. Keep all the belongings, such as bags, lunch box and food materials outside the lab.
3. On any circumstances private Laptop, Pen drive, CD, Notebooks are not allowed inside the lab without prior permission of HoD-CSE.
4. Wearing ID Cards is compulsory. Entry will not be permitted, in case of not having the ID cards.
5. On any account usage of a single PC by two or more students is prohibited.
6. Maintain a high degree of discipline inside the lab, by not roaming and making unwanted noises.
7. In case of any difficulty in using the systems, contact only the Lab Instructor.
8. Chatting and viewing any video content pages are strictly prohibited. In case of violation, serious action will be taken.
9. Remove the footwear and keep it outside only on shoe-rack.
10. Register the usage timing along with signature in the Login Register.
11. Conversation between boys and girls should be strictly avoided. Students violating will be punished seriously.
12. Co-operate with the lab instructor in maintaining the rules and regulations, and pave the way for effective utilization of lab.

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Motivation of the course

Motivation of this course is to provide basic knowledge in conception, implementation and functioning of the computer networks. This course provides students with hands on training regarding the design, troubleshooting, modeling and evaluation of computer networks. In this course, students are going to experiment in a real test-bed networking environment, and learn about network design and troubleshooting topics and tools such as: network addressing, Address Resolution Protocol (ARP), basic troubleshooting tools (e.g. ping, ICMP), IP routing (e,g, RIP), route discovery (e.g. traceroute), TCP and UDP, IP fragmentation and many others. Student will also be introduced to the network modeling and simulation, and they will have the opportunity to build some simple networking models using the tool and perform simulations that will help them evaluate their design approaches and expected network performance.

In this line, this course has been introduced by the Anna University, Chennai to the Third year Computer Science engineering undergraduate students.

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Objective of the course

- To learn and use network commands.
- To learn socket programming.
- To implement and analyze various network protocols.
- To learn and use simulation tools.
- To use simulation tools to analyze the performance of various network protocols.

Outcome of the course

Upon successful completion of this course, students will

- Implement various protocols using TCP and UDP.
- Compare the performance of different transport layer protocols.
- Use simulation tools to analyze the performance of various network protocols.
- Analyze various routing algorithms.
- Implement error correction codes.

Pre-requisite of the course

- Fundamentals in Hardware and Software aspects
- Basic knowledge in computer programming skills in C/C++/JAVA/Python

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Syllabi of the course

CS8581	NETWORKS LABORATORY	L T P C 0 0 4 2

LIST OF EXPERIMENTS

1. Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and traceroute PDUs using a network protocol analyzer and examine.
2. Write a HTTP web client program to download a web page using TCP sockets.
3. Applications using TCP sockets like:
    - Echo client and echo server
    - Chat
    - File Transfer
4. Simulation of DNS using UDP sockets.
5. Write a code simulating ARP /RARP protocols.
6. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS.
7. Study of TCP/UDP performance using Simulation tool.
8. Simulation of Distance Vector/ Link State Routing algorithm.
9. Performance evaluation of Routing protocols using Simulation tool.
10. Simulation of error correction code (like CRC

TOTAL: 60 PERIODS

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

Standalone desktops

REQUIREMENTS

1. C / C++ / Java / Python / Equivalent Compiler
2. Network simulator like NS2/Glomosim/OPNET/ Packet Tracer / Equivalent

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

Observation Note Book Evaluation Rubrics

(Applicable for change depends on the course in charge)

Velammal College of Engineering Technology *Department of CSE* *Lab Program Evaluation*	
Logic (5)	
Coding (5)	
Efficiency (5)	
Viva (5)	
Total(20)	
Course In Charge	

Record Note Book Evaluation Rubrics

(Applicable for change depends on the course in charge)

Velammal College of Engineering Technology *Department of CSE* *Evaluation of Lab Exercise*	
Observation(20)	
Record(5)	
Total(25)	
Course In Charge	

Department Vision and Mission

Vision

To become a Center of Excellence in the field of Computer Science and Engineering upholding social values.

Mission

•Heightening the knowledge of the faculty in recent trends through continuous development programs.

•Transforming the students into globally competent and technically well-equipped Computer Professionals with strong theoretical and practical knowledge.

•Cultivating the spirit of social and ethical values for the cause of development of our Nation.

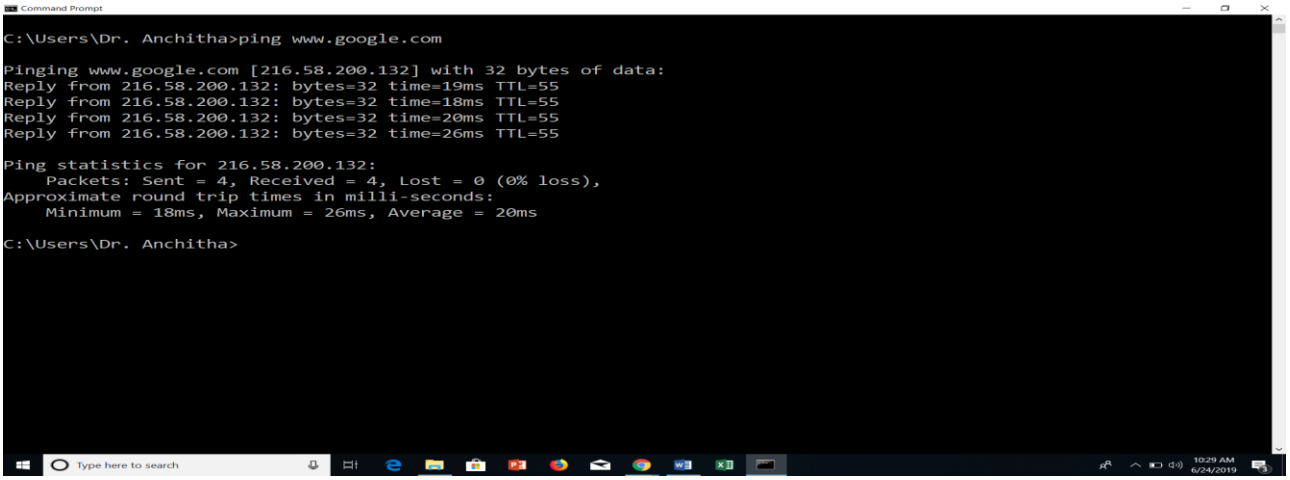
Experiment 1
Implementing the basic Commands
1. Ping Command

The ping command is one of the most often used networking utilities for detecting devices on a network and for troubleshooting network problems.

When you ping a device you send that device a short message, which it then sends back **(the echo**).

The general format is **ping hostname** or **ping IPaddress**.

Example **ping www.google.com** or **ping 216.58.208.68**



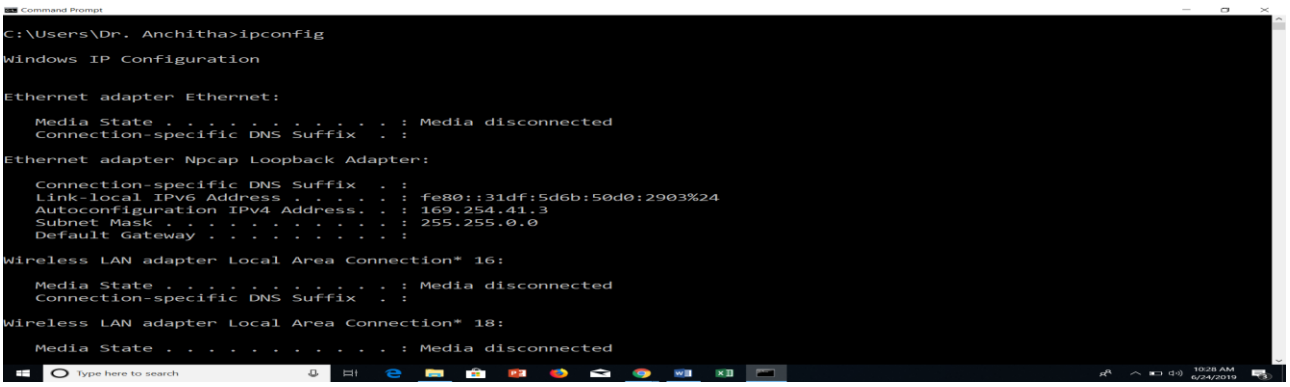
2. ipconfig Command

Another indispensable and frequently used utility that is used for finding network information about your local machine like IP addresses, DNS addresses etc

Basic Use: Finding Your IP Address and Default Gateway

Type the command ipconfig at the prompt.

The following is displayed



Ip config has a number of switches the most common are**:**

ipconfig /all – displays more information about the network setup on your systems including the MAC address.

ipconfig /release – release the current IP address

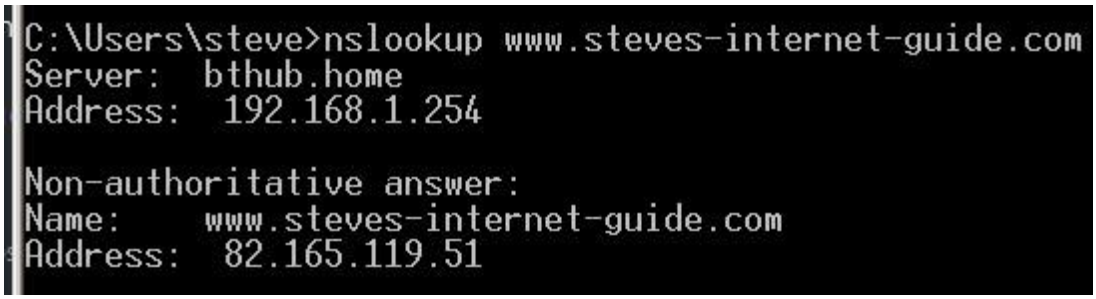
ipconfig /renew – renew IP address

ipconfig /? -shows help

3**. NSlookup**

- Used for checking DNS record entries.
- nslookup is a network administration command-line tool available for many computer operating systems.
- It is used for querying the Domain Name System (DNS) to obtain domain name or IP address mapping information
- The main use of **nslookup** is for troubleshooting DNS related problems.
- Nslookup can be use in **interactive** and **non-interactive** mode.
- To use in interactive mode **type nslookup** at the command line and hit return.
- You should get an **nslookup command prompt.**

- To use in **non-interactive** mode type **nslookup options** at the command prompt.



Using Nslookup

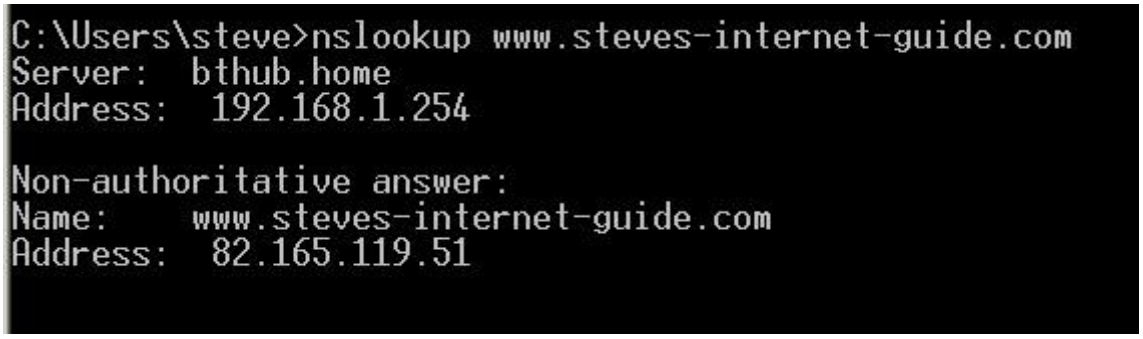
- To illustrate the use of nslookup we are going to use it to:
    - Find the IP address of a host.
    - Find the domain name of an IP address.
    - Find mail servers for a domain.

Finding The IP Address of an Host-

To find the ip address of a host e.g. www.steves-internet-guide.com type:

nslookup www.steves-internet-guide.com

at a command prompt.



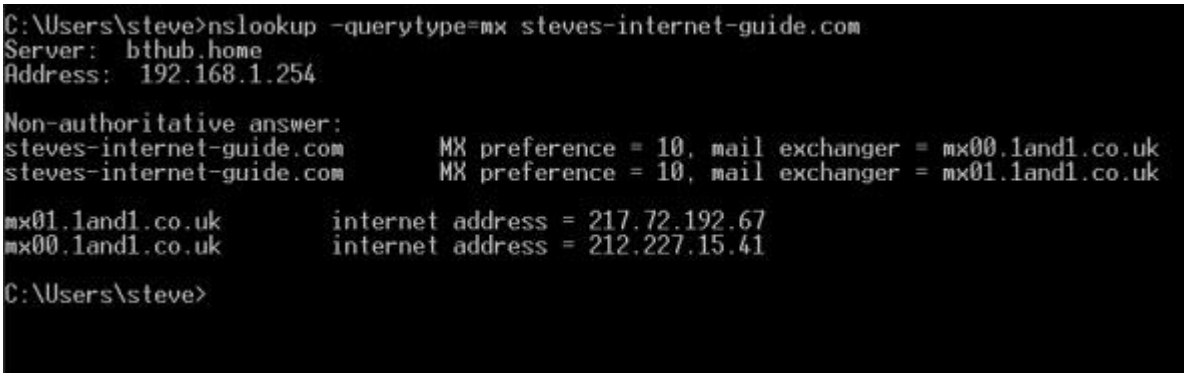
for an interactive lookup:

Reverse Lookup IP address to domain name

Type **nslookup** IP address

Find Mail Servers for a Domain

Type **nslookup -querytype=mx domain name**



4.Netstat Command

Used for displaying information about tcp and udp connections and ports

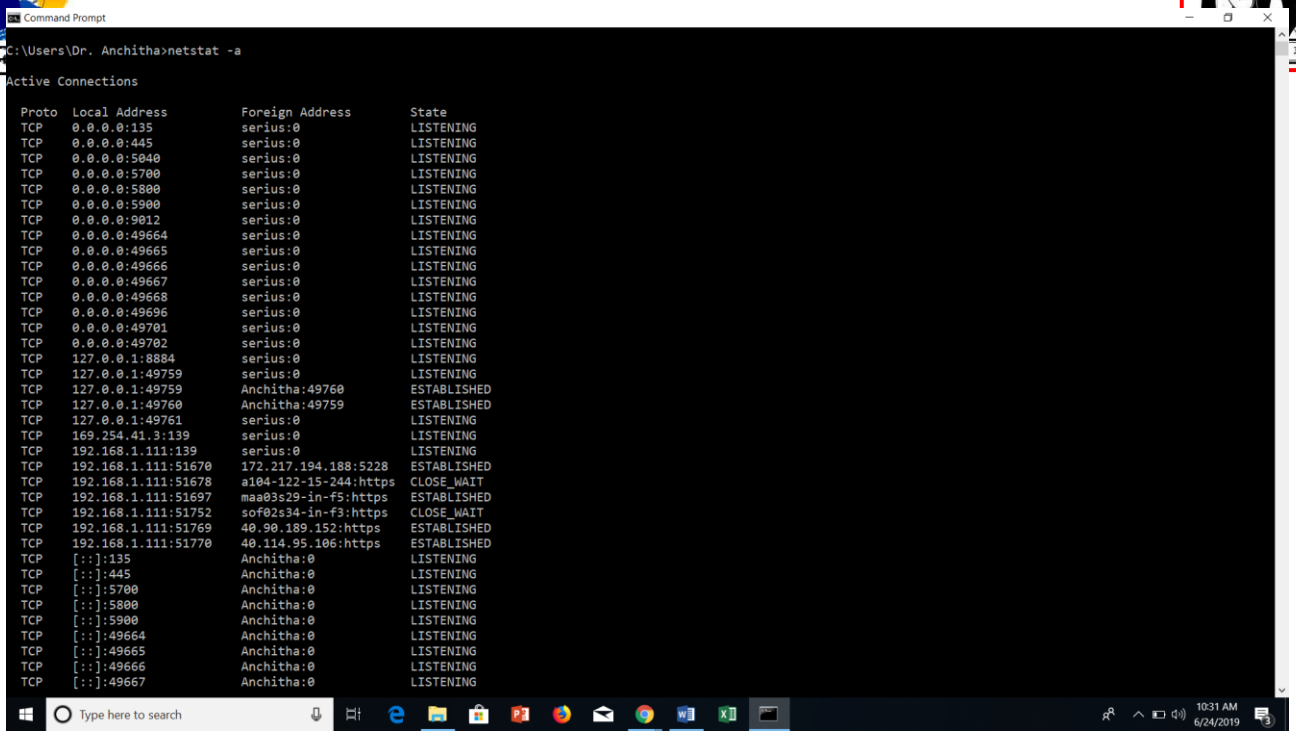
Netstat Command Syntax

netstat [**-a**] [**-b**] [**-e**] [**-f**] [**-n**] [**-o**] [**-p** *protocol*] [**-r**] [**-s**] [**-t**] [**-x**] [**-y**] [*time_interval*] [**/?**]

Netstat Command List	
Option	**Explanation**
netstat	Execute the netstat command alone to show a relatively simple list of all active TCP connections which, for each one, will show the local IP address (your computer), the foreign IP address (the other computer or network device), along with their respective port numbers, as well as the TCP state.
-a	This switch displays active TCP connections, TCP connections with the listening state, as well as UDP ports that are being listened to.
-b	This netstat switch is very similar to the **-o** switch listed below, but instead of displaying the PID, will display the process's actual file name. Using **-b** over **-o** might seem like it's

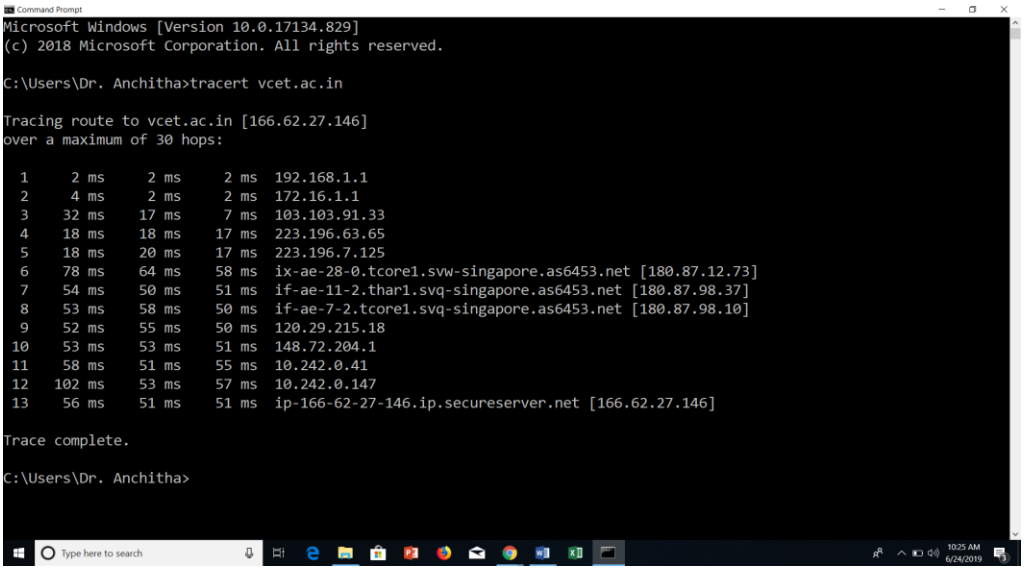
	saving you a step or two but using it can sometimes greatly extend the time it takes netstat to fully execute.
-e	Use this switch with the netstat command to show statistics about your network connection. This data includes bytes, unicast packets, non-unicast packets, discards, errors, and unknown protocols received and sent since the connection was established.
-f	The **-f** switch will force the netstat command to display the Fully Qualified Domain Name (FQDN) for each foreign IP addresses when possible.
-n	Use the **-n** switch to prevent netstat from attempting to determine host names for foreign IP addresses. Depending on your current network connections, using this switch could considerably reduce the time it takes for netstat to fully execute.
-o	A handy option for many troubleshooting tasks, the **-o** switch displays the process identifier (PID) associated with each displayed connection. See the example below for more about using **netstat -o**.
-p	Use the **-p** switch to show connections or statistics only for a particular *protocol*. You can not define more than one *protocol* at once, nor can you execute netstat with **-p** without defining a *protocol*.
protocol	When specifying a *protocol* with the **-p** option, you can use **tcp**, **udp**, **tcpv6**, or **udpv6**. If you use **-s** with **-p** to view statistics by protocol, you can use **icmp**, **ip**, **icmpv6**, or **ipv6** in addition to the first four I mentioned.
-r	Execute netstat with **-r** to show the IP routing table. This is the same as using the route command to execute **route print**.

-s	The **-s** option can be used with the netstat command to show detailed statistics by protocol. You can limit the statistics shown to a particular protocol by using the **-s**option and specifying that *protocol*, but be sure to use **-s** before **-p** *protocol* when using the switches together.
-t	Use the **-t** switch to show the current TCP chimney offload state in place of the typically displayed TCP state.
-x	Use the **-x** option to show all NetworkDirect listeners, connections, and shared endpoints.
-y	The **-y** switch can be used to show the TCP connection template for all connection. You cannot use **-y** with any other netstat option.
time_interval	This is the time, in seconds, that you'd like the netstat command to re-execute automatically, stopping only when you use Ctrl-C to end the loop.
/?	Use the help switch to show details about the netstat command's several options.



5. tracert Command

Traceroute is a command which can show you the path a packet of information takes from your computer to one you specify. It will list all the routers it passes through until it reaches its destination, or fails to and is discarded. In addition to this, it will tell you how long each 'hop' from router to router takes



TCP Sockets

- A socket is an endpoint of a two-way communication link between two programs running on the network. Socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent. User-level process/services generally use port number value > 1024. TCP provides a reliable, point-to-point communication channel that client-server application on the Internet use to communicate with each other. Examples are FTP and Telnet.

- To communicate over TCP, a client program and a server program establish a connection to one another. Each program binds a socket to its end of the connection. A server runs on a specific computer and has a socket that is bound to a specific port number. The server waits, listening to the socket for a connection request from the client.

- On the client-side, the client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to make contact with the server on the server's machine and port. The client also

needs to identify itself to the server so it binds to a local port number that it will use during this connection.

- If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.

- On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server. The client and server can now communicate by writing to or reading through I/O streams from their sockets and eventually close it.

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

Experiment 2

HTTP WEB CLIENT PROGRAM TO DOWNLOAD A WEB PAGE USING TCP SOCKETS

AIM

To download a webpage using TCP sockets

ALGORITHM:

CLIENT SIDE:

1) Start the program.
2) Create a socket which binds the Ip address of server and the port address to acquire service.
3) After establishing connection send the url to server.
4) Open a file and store the received data into the file.
5) Close the socket.
6) End the program.

SERVER SIDE:

1) Start the program.
2) Create a server socket to activate the port address.
3) Create a socket for the server socket which accepts the connection.
4) After establishing connection receive url from client.
5) Download the content of the url received and send the data to client.
6) Close the socket.
7) End the program.

Viva Questions

1. Define socket
2. What protocols fall under the TCP/IP Internet Layer?
3. What does A Socket Consists Of?
4. How do you Open A Socket?
5. Explain Data Transfer Over Connected Sockets - Send() And Recv()?

Experiment 3A

ECHO CLIENT AND ECHO SERVER USING TCP SOCKETS

AIM

To write a socket program for implementation of echo

ALGORITHM

CLIENT SIDE

1. Start the program.
2. Create a socket which binds the Ip address of server and the port address to acquire service.
3. after establishing connection send a data to server.
4. Receive and print the same data from server.
5. Close the socket. 6. End the program.

SERVER SIDE

1. Start the program.
2. Create a server socket to activate the port address.
3. Create a socket for the server socket which accepts the connection.
4. after establishing connection receive the data from client.
5. Print and send the same data to client. 6. Close the socket.
7. End the program.

Viva Questions

1. What does a Socket consists of?
2. What is the concept of Echo?
3. Explain Ping?
4. Mention some advantages of Java Sockets.
5. How do you open a Socket?

Experiment 3B

CHAT APPLICATION USING TCP SOCKETS

AIM

To write a client-server application for chat using TCP SOCKETS

ALGORITHM

CLIENT

1. Start the program
2. Include necessary package in java
3. To create a socket in client to server.
4. The client establishes a connection to the server.
5. The client accept the connection and to send the data from client to server.
6. The client communicates the server to send the end of the message
7. Stop the program .

SERVER

1. Start the program
2. Include necessary package in java
3. To create a socket in server to client
4. The server establishes a connection to the client.
5. The server accept the connection and to send the data from server to client and vice versa
6. The server communicates the client to send the end of the message.
7. Stop the program .

Viva Questions:

1. Define Socket
2. What is socket programming?
3. What is the function of command bind?
4. What is the syntax for connecting Client and Server
5. What is the command to assign port number to client and server?.

Experiment 3C

FILE TRANSFER IN CLIENT & SERVER USING TCP SOCKETS

AIM

To Perform File Transfer in Client & Server Using TCP SOCKETS

ALGORITHM

CLIENT SIDE

1. Start.
2. Establish a connection between the Client and Server.
3. Socketss=new Socket(InetAddress.getLocalHost(),1100);
4. Implement a client that can send two requests.
    i) To get a file from the server.
    ii) To put or send a file to the server.
5. After getting approval from the server, the clients either get file from the server or send file to the server.

SERVER SIDE

1. Start.
2. Implement a server socket that listens to a particular port number.
3. Server reads the filename and sends the data stored in the file for the‘get’ request.
4. It reads the data from the input stream and writes it to a file in the server for the ‘put’ instruction.
5. Exit upon client’s request.
6. Stop.

Viva Questions:

1. What does the java.net.InetAddress class represent?
2. FTP is built on which architecture?
3. How many parallel TCP connections are used by FTP to transfer a file?
4. What is the Mode of data transfer in FTP?
5. The password is sent to the server using which command?

Experiment 4

DNS USING UDP SOCKET

AIM:

To simulate DNS using UDP transport protocol using java

ALGORITHM:

1. Start the program.
2. Create a class for server.
3. Create an object for the Datagram class and declare an array which acts as buffer.
4. In server()
    - 4.1 Intialise position.
    - 4.2 Send character to client using packet if it is not a return or null
    - 4.3 Increment position
5. In main()
    - 5.1 Get the data.
    - 5.2 Create object for Datagram socket.
    - 5.3 Execute my server.
6. Create a class for client
7. Create object for Datagram socket and declare an array that acts as buffer.
8. In client()
    - 8.1 Create an object for Datagram packet.
    - 8.2 Receiver and print character from server.
9. In main()
    - 9.1 create array for string address.
    - 9.2 Setup a for loop to print address.
    - 9.3 Create object for Datagram socket.
    - 9.4 Execute myclient.
10. Stop the program.

Viva Questions:

1. What is the Return value of the UDP port “Chargen” ?
2. Beyond IP, What additional services UDP Provides?
3. What are the advantages for using UDP sockets over TCP sockets?
4. Mention the Port number used by Network Time Protocol(NTP) with UDP.
5. Which classes are used for connection-less socket programming?

Experiment 5A

ADDRESS RESOLUTION PROTOCOL

AIM

To simulate Address Resolution Protocol.

ALGORITHM

CLIENT SIDE

1. Establish a connection between the Client and Server.

Socket ss=new Socket(InetAddress.getLocalHost(),1100);

2. Create instance output stream writer

PrintWriter ps=new PrintWriter(s.getOutputStream(),true);

3. Get the IP Address to resolve its physical address.
4. Send the IPAddress to its output Stream.ps.println(ip);
5. Print the Physical Address received from the server.

SERVER SIDE

1. Accept the connection request by the client.

ServerSocket ss=new ServerSocket(2000);

Socket s=ss.accept();

2. Get the IPaddress from its inputstream.

BufferedReader br1=new BufferedReader(newInputStreamReader(s.getInputStream()));

ip=br1.readLine();

3. During runtime execute the processRuntime r=Runtime.getRuntime();

Process p=r.exec("arp -a "+ip);

4. Send the Physical Address to the client.

Viva Questions:

1. What is ARQ?
2. What is stop and wait protocol?
3. What is stop and wait ARQ?
4. What is usage of sequence number in reliable transmission?
5. What is sliding window?

Experiment 5B

REVERSE ADDRESS RESOLUTION PROTOCOL (RARP)

AIM:
To write a java program for simulating RARP protocols using UDP

ALGORITHM:
CLIENT SIDE

1. Start the program
2. Using datagram sockets UDP function is established.
3. Get the MAC address to be converted into IP address.
4. Send this MAC address to server.
5. Server returns the IP address to client.
6. Stop the program

SERVER SIDE

1. Start the program.
2. Server maintains the table in which IP and corresponding MAC addresses are stored.
3. Read the MAC address which is send by the client.
4. Map the IP address with its MAC address and return the IP address to client.
5. Stop the program

Viva Questions:

Experiment 6A

STUDY OF NETWORK SIMULATOR (NS2)

INTRODUCTION

The network simulator is discrete event packet level simulator. The network simulator covers a very large number of applications of different kind of protocols of different network types consisting of different network elements and traffic models. Network simulator is a package of tools that simulates behavior of networks such as creating network topologies, log events that happen under any load, analyze the events and understand the network. Well the main aim of our first experiment is to learn how to use network simulator and to get acquainted with the simulated objects and understand the operations of network simulation and we also need to analyze the behavior of the simulation object using network simulation.

Platform required to run network simulator

- Unix and Unix like systems
- Linux (Use Fedora or Ubuntu versions)
- Free BSD
- SunOS/Solaris
- Windows 95/98/NT/2000/XP

Backend Environment of Network Simulator

Network Simulator is mainly based on two languages.They are C++ and OTcl. OTcl is the object oriented version of Tool Command language.The network simulator is a bank of of different network and protocol objects. C++ helps in the following way:

- It helps to increase the efficiency of simulation.
- Its is used to provide details of the protocols and their operation.
- It is used to reduce packet and event processing time.

OTcl helps in the following way:

- With the help of OTcl we can describe different network topologies
- It helps us to specify the protocols and their applications
- It allows fast development
- Tcl is compatible with many platforms and it is flexible for integration

Tcl is very easy to use and it is available in free

Basics of Tcl Programming (w.r.t. ns2)

Before we get into the program we should consider the following things:

1. Initialization and termination aspects of network simulator.
2. Defining the network nodes,links,queues and topology as well.
3. Defining the agents and their applications
4. Network Animator(NAM)
5. Tracing

Initialization

To start a new simulator we write

1. set ns [new Simulator]

From the above command we get that a variable ns is being initialized by using the set command. Here the code [new Simulator] is a instantiation of the class Simulator which uses the reserved word 'new'. So we can call all the methods present inside the class simulator by using the variable ns.

Creating the output files

```
#To create the trace files we write
set tracefile1 [open out.tr w]
$ns trace-all $tracefile1
#To create the nam files we write
set namfile1 [open out.nam w]
$ns namtrace-all $namfile
```

In the above we create a output trace file out.tr and a nam visualization file out.nam. But in the Tcl script they are not called by their names declared, while they are called by the pointers initialized for them such as tracefile1 and namfile1 respectively. The line which starts with '#' are commented. The next line opens the file 'out.tr' which is used for writing is declared 'w'. The next line uses a simulator method trace-all by which we will trace all the events in a particular format.
The termination program is done by using a 'finish' procedure

```
01    # Defining the 'finish' procedure'
02
```

```
03 proc finish {} {
04     global ns tracefile1 namfile1
05     $ns flush-trace
06     close $tracefile
07     close $namfile
08     exec nam out.nam &
09     exit 0
10    }
```

In the above the word 'proc' is used to declare a procedure called 'finish'. The word 'global' is used to tell what variables are being used outside the procedure. 'flush-trace' is a simulator method that dumps the traces on the respective files. The command 'close' is used to close the trace files and the command 'exec' is used to execute the nam visualization. The command 'exit' closes the application and returns 0 as zero(0) is default for clean exit.
In ns we end the program by calling the 'finish' procedure

```
1  #end the program
2  $ns at 125.0 "finish"
```

Thus the entire operation ends at 125 seconds. To begin the simulation we will use the command

```
1  #start the the simulation process
2  $ns run
```

Defining nodes,links,queues and topology

Way to create a node:
View source print?

```
1 set n0 [ns node]
```

In the above we created a node that is pointed by a variable n0.While referring the node in the script we use $n0. Similarly we create another node n2.Now we will set a link between the two nodes.

```
1  $ns duplex-link $n0 $n2 10Mb 10ms DropTail
```

So we are creating a bi-directional link between n0 and n2 with a capacity of 10Mb/sec and a propagation delay of 10ms.

In NS an output queue of a node is implemented as a part of a link whose input is that node to handle the overflow at the queue.But if the buffer capacity of the output queue is exceeded then the last packet arrived is dropped and here we will use a 'DropTail' option. Many other options such as RED(Random Early Discard) mechanism, FQ(Fair Queuing), DRR(Deficit Round Robin), SFQ(Stochastic Fair Queuing) are available.

So now we will define the buffer capacity of the queue related to the above link

```
1   #Set queue size of the link

2   $ns queue-limit $n0 $n2 20
```

so, if we summarize the above three things we get

```
01  #create nodes
02
03  set n0 [$ns node]
04  set n1 [$ns node]
05  set n2 [$ns node]
06  set n3 [$ns node]
07  set n4 [$ns node]
08  set n5 [$ns node]
09
10  #create links between the nodes
11
12  $ns duplex-link $n0 $n2 10Mb 10ms DropTail
13  $ns duplex-link $n1 $n2 10Mb 10ms DropTail
14  $ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
15  $ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
16  $ns duplex-link $n0 $n2 0.5Mb 40ms DropTail
17  $ns duplex-link $n0 $n2 0.5Mb 40ms DropTail
18
19  #set queue-size of the link (n2-n3) to 20
20  $ns queue-limit $n2 $n3 20
```

TCP

TCP is a dynamic reliable congestion protocol which is used to provide reliable transport of packets from one host to another host by sending acknowledgements on proper transfer or loss of packets. Thus TCP requires bi-directional links in order for acknowledgements to return to the source.

Now we will show how to set up tcp connection between two nodes

```
1  #setting a tcp connection
2
3  set tcp [new Agent/TCP]
4  $ns attach-agent $n0 $tcp
5  set sink [new Agent/TCPSink]
6  $ns attach-agent $n4 $sink
7  $ns connect $tcp $sink
8  $tcp set fid_1
9  $tcp set packetSize_552
```

The command 'set tcp [new Agent/TCP]' gives a pointer called 'tcp' which indicates the tcp agent which is a object of ns. Then the command '$ns attach-agent $n0 $tcp' defines the source node of tcp connection. Next the command 'set sink [new Agent/TCPSink]' defines the destination of tcp by a pointer called sink. The next command '$ns attach-agent $n4 $sink' defines the destination node as n4.Next, the command '$ns connect $tcp $sink' makes the TCP connection between the source and the destination. i.e n0 and n4.When we have several flows such as TCP, UDP etc in a network. So, to identify these flows we mark these flows by using the command '$tcp set fid_1'. In the last line we set the packet size of tcp as 552 while the default packet size of tcp is 1000.

FTP over TCP

File Transfer Protocol (FTP) is a standard mechanism provided by the Internet for transferring files from one host to another. Well this is the most common task expected from a networking or a inter networking. FTP differs from other client server applications in that it establishes between the client and the server. One connection is used for data transfer and other one is used for providing

control information. FTP uses the services of the TCP. It needs two connections. The well Known port 21 is used

for control connections and the other port 20 is used for data transfer. Well here we will learn in how to run a FTP connection over a TCP

```
#Initiating FTP over TCP
set ftp [new Application/FTP]
 $ftp attach-agent $tcp
```

In above, the command 'set ftp [new Application/FTP]' gives a pointer called 'ftp' which indicates the FTP application. Next, we attach the ftp application with tcp agent as FTP uses the services of TCP.

UDP

The User datagram Protocol is one of the main protocols of the Internet protocol suite.UDP helps the host to send messages in the form of datagrams to another host which is present in a Internet protocol network without any kind of requirement for channel transmission setup. UDP provides a unreliable service and the datagrams may arrive out of order, appear duplicated, or go missing without notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system.

Now we will learn how to create a UDP connection in network simulator.

```
1  # setup a UDP connection
2  set udp [new Agent/UDP]
3  $ns attach-agent $n1 $udp
4  $set null [new Agent/Null]
5  $ns attach-agent $n5 $null
6  $ns connect $udp $null
7  $udp set fid_2
```

Similarly, the command 'set udp [new Agent/UDP]' gives a pointer called 'udp' which indicates the udp agent which is a object of ns. Then the command '$ns attach-agent $n1 $udp' defines the source node of udp connection. Next the command 'set null [new Agent/Null]' defines the destination of udp by a pointer called null. The next command '$ns attach-agent $n5 $null' defines the destination node as n5.Next, the command '$ns connect $udp $null' makes the UDP connection between the

source and the destination. i.e n1 and n5.When we have several flows such as TCP,UDP etc in a network. So, to identify these flows we mark these flows by using the command '$udp set fid_2

Constant Bit Rate(CBR)

Constant Bit Rate (CBR) is a term used in telecommunications, relating to the quality of service. When referring to codecs, constant bit rate encoding means that the rate at which a codec's output data should be consumed is constant. CBR is useful for streaming multimedia content on limited capacity channels since it is the maximum bit rate that matters, not the average, so CBR would be used to take advantage of all of the capacity. CBR would not be the optimal choice for storage as it would not allocate enough data for complex sections (resulting in degraded quality) while wasting data on simple sections.

CBR over UDP Connection

```
1  #setup cbr over udp
2
3  set cbr [new Application/Traffic/CBR]
4  $cbr attach-agent $udp
5  $cbr set packetSize_1000
6  $cbr set rate_0.01Mb
7  $cbr set random _false
```

In the above we define a CBR connection over a UDP one. Well we have already defined the UDP source and UDP agent as same as TCP. Instead of defining the rate we define the time interval between the transmission of packets in the command '$cbr set rate_0.01Mb'. Next, with the help of the command '$cbr set random _false' we can set random noise in cbr traffic.we can keep the noise by setting it to 'false' or we can set the noise on by the command '$cbr set random _1'. We can set by packet size by using the command '$cbr set packetSize_(packetsize).We can set the packet size up to sum value in bytes.

Scheduling Events

In ns the tcl script defines how to schedule the events or in other words at what time which event will occur and stop. This can be done using the command

$ns at .

So here in our program we will schedule the ftp and cbr.

```
# scheduling the events

$ns at 0.1 "cbr start"


$ns at 1.0 "ftp start"
$ns at 124.0 "ftp stop"
$ns at 124.5 "cbr stop"
```

Network Animator (NAM)

When we will run the above program in ns then we can can visualize the network in the NAM. But instead of giving random positions to the nodes, we can give suitable initial positions to the nodes and can form a suitable topology. So, in our program we can give positions to the nodes in NAM in the following way

```
#Give position to the nodes in NAM

$ns duplex-link-op $n0 $n2 orient-right-down
$ns duplex-link-op $n1 $n2 orient-right-up
$ns simplex-link-op $n2 $n3 orient-right
$ns simplex-link-op $n3 $n2 orient-left
$ns duplex-link-op $n3 $n4 orient-right-up
$ns duplex-link-op $n3 $n5 orient-right-down
```

We can also define the color of cbr and tcp packets for identification in NAM.For this we use the following command

```
#Marking the flows
$ns color1 Blue
$ns color2 Red
```

To view the network animator we need to type the command: nam

Tracing

Tracing Objects

NS simulation can produce visualization trace as well as ASCII file corresponding to the events that are registered at the network. While tracing ns inserts four objects: EnqT,DeqT,RecvT & DrpT.

EnqT registers information regarding the arrival of packet and is queued at the input queue of the link. When overflow of a packet occurs, then the information of thye dropped packet is registered in DrpT.DeqT holds the information abut the packet that is dequeued instantly.RecvT hold the information about the

packet that has been received instantly.

Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id

Structure of Trace files

1. The first field is event.It gives you four possible symbols '+' '-' 'r' 'd'.These four symbols correspond respectively to enqueued, dequeued, received and dropped.
2. The second field gives the time at which the event occurs
3. The third field gives you the input node of the link at which the event occurs
4. The fourth field gives you the the output node at which the event occurs
5. The fifth field shows the information about the packet type.i.e whether the packet is UDP or TCP
6. The sixth field gives the packet size
7. The seventh field give information about some flags
8. The eight field is the flow id(fid) for IPv6 that a user can set for each flow in a tcl script.It is also used for specifying the color of flow in NAM display
9. The ninth field is the source address
10. The tenth field is the destination address
11. The eleventh field is the network layer protocol's packet sequence number
12. The last field shows the unique id of packet

Following are trace of two events:

r 1.84471 2 1 cbr 210 ------- 1 3.0 1.0 195 600
r 1.84566 2 0 ack 40 ------- 2 3.2 0.1 82 602

The trace file can be viewed with the cat command:
cat out.tr

Experiment 6B

SIMULATION OF CONGESTION CONTROL ALGORITHM

AIM:

To simulate the link failure and observe the congestion control algorithm using NS2.

ALGORITHM:

1. Create a simulation object
2. Set routing protocol to routing
3. Trace packets and all links onto NAM trace and to trace file
4. Create right nodes
5. Describe their layout topology as octagon
6. Add a sink agent to node
7. Connect source and sink.

Experiment 8A

IMPLEMENTATION OF DISTANCE VECTOR ROUTING ALGORITHM

AIM:

To simulate and observe traffic route of a network using distance vector routing protocol.

ALGORITHM:

1. Create a simulator object
2. Set routing protocol to Distance vector routing
3. Trace packets on all links on to NAM trace and text trace file.
4. Define finish procedure to close files, flash tracing and run NAM
5. Create 5 nodes
6. Specify the link characteristics between the nodes
7. Describer their layout topology as a octagon
8. Add UDP agent for node n0
9. Create CBR traffic on the top of UDP and set traffic parameters
10. Add NULL agent to node n3
11. Connect source and sink
12. Schedule as follows
    - Start traffic flow at 1.0
    - Down the link n1 – n2 at 15.0
    - Up the link n1 – n2 at 25.0
    - Call finish procedure at 35.0
13. Start the scheduler
14. Observe the traffic route when the link is up and down
15. View the simulated events and trace file analyze it
16. Stop.

Viva Questions:

1. Compare connection oriented and connection less protocols.
2. What is MTU?
3. Explain the working of Distance vector routing.
4. Differentiate Proactive and Reactive routing Protocols.
5. What are the different attributes for calculating the cost of a path?

Experiment 8B

IMPLEMENTATION OF LINK STATE ROUTING ALGORITHM

AIM:

To simulate and observe traffic route of a network using distance vector routing protocol.

ALGORITHM:

1. Create a simulator object
2. Set routing protocol to Distance vector routing
3. Trace packets on all links on to NAM trace and text trace file.
4. Define finish procedure to close files, flash tracing and run NAM
5. Create 5 nodes
6. Specify the link characteristics between the nodes
7. Describer their layout topology as a octagon
8. Add UDP agent for node n0
9. Create CBR traffic on the top of UDP and set traffic parameters
10. Add NULL agent to node n3
11. Connect source and sink
12. Schedule as follows

- Start traffic flow at 1.0
- Down the link n1 – n2 at 15.0
- Up the link n1 – n2 at 25.0
- Call finish procedure at 35.0

13. Start the scheduler
14. Observe the traffic route when the link is up and down
15. View the simulated events and trace file analyze it
16. Stop

Viva Questions:

1. What is Routing?
2. What is dynamic routing?
3. What are the two steps in link state routing?
4. Compare link state and Distance Vector routing
5. What are all the route metric used in Link state routing?

Experiment 10

IMPLEMENTATION OF ERROR DETECTION AND CORRECTION TECHNIQUE

AIM:

To write a 'C' program to implement Error Detection and Correction Techniques.

ALGORITHM:

1. Get four bit data
2. Generate the generator matrix
3. Encode the data
4. Create parity check matrix
5. Enter the error data
6. Calculate syndrome using $e.H^T$
7. Decode the error data.

Viva Questions:

1. What are the types of errors?
2. What is Error Detection?
3. What are its methods?
4. What is Redundancy?
5. What is CRC?
6. What is Checksum?