**EX NO:1**

**DATE:**

**Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and traceroute PDUs using wireshark and examine.**

## AIM:

To study the basic networking commands.

## NETWORKING COMMANDS:

## C:\>tcpdump: Tcpdump is a command line utility that allows you to capture and analyze network traffic going through your system. It is often used to help troubleshoot network issues, as well as a security tool.

C:\>netstat: Netstat displays a variety of statistics about a computers active TCP/IP connections. This tool is most useful when you’re having trouble with TCP/IP applications such as HTTP, and FTP.

C:\>nbtstat –a: This command helps solve problems with NetBIOS name resolution. (Nbt stands for NetBIOS over TCP/IP)

C:\>ifconfig: The command ifconfig stands for interface configurator. This command enables us to initialize an interface, assign IP address, enable or disable an interface. It display route and network interface.

C:\>ipconfig: The ipconfig command displays information about the host (the computer your sitting at)computer TCP/IP configuration.

C:\>nslookup: Nslookup is used for diagnosing DNS problems. If you can access a resource by specifying an IP address but not it’s DNS you have a DNS problem.

C:\>route: The route command displays the computers routing table. A typical computer, with a single network interface, connected to a LAN, with a router is fairly simple and generally doesn’t pose any network problems. But if you’re having trouble accessing other computers on your network, you can use the route command to make sure the entries in the routing table are correct.

C:\>tracert: The tracert command displays a list of all the routers that a packet has to go through to get from the computer where tracert is run to any other computer on the internet.

C:\>pathping: Pathping is unique to Window’s, and is basically a combination of the Ping and Tracert commands. Pathping traces the route to the destination address then launches a 25 second test of each router along the way, gathering statistics on the rate of data loss along each hop.

C:\>ping: Ping is the most basic TCP/IP command, and it’s the same as placing a phone call to your best friend. You pick up your telephone and dial a number, expecting your best friend to reply with “Hello” on the other end. Computers make phone calls to each other over a network by using a Ping command. The Ping commands main purpose is to place a phone call to another computer on the network, and request an answer. Ping has 2 options it can use to place a phone call to another computer on the network. It can use the computers name or IP address.

## RESULT:

Thus the above list of primitive has been studied.

EX NO: 2

DATE:

# Write a HTTP web client program to download a

**web page using TCP sockets**

## AIM:

To Write a HTTP web client program to download a web page 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.

## RESULT:

The webpage is successfully downloaded and the contents are displayed and verified.

EX NO : 3 A

DATE:

**SOCKET PROGRAM FOR ECHO**

## 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.

## RESULT:

Thus the program for simulation of echo server was written & executed.

EX NO : 3 B

DATE:

**CLIENT- SERVER APPLICATION FOR CHAT**

## AIM:

To write a client-server application for chat using TCP

## 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
6. vice versa
7. The server communicate the client to send the end of the message.
8. Stop the program.

## RESULT:

Thus the above program a client-server application for chat using TCP / IP was executed and successfully.

EX NO : 3 C

DATE:

**FILE TRANSFER IN CLIENT & SERVER**

## AIM:

To Perform File Transfer in Client & Server Using TCP/IP.

## ALGORITHM:

**CLIENT SIDE**

* 1. Start.
  2. Establish a connection between the Client and Server.
  3. Socket ss=new Socket(InetAddress.getLocalHost(),1100);
  4. Implement a client that can send two requests.
     1. To get a file from the server.
     2. To put or send a file to the server.
  5. After getting approval from the server ,the client either get file from the server or send
  6. 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 theserver for the ‘put’ instruction.
5. Exit upon client’s request.
6. Stop.

## RESULT:

Thus the File transfer Operation is done & executed successfully.

EX NO : 4

DATE:

**Simulation of DNS using UDP sockets.**

## AIM:

To write a program to Simulation of DNS using UDP sockets..

## ALGORITHM:

1. Start the program.
2. Get the frame size from the user
3. To create the frame based on the user request. 4.To send frames to server from the client side.
4. If your frames reach the server it will send ACK signal to client otherwise it will send NACK signal to client.
5. Stop the program.

## RESULT:

Thus the above program for Simulation of DNS using UDP sockets was executed and successfully

EX NO : 5 A

DATE:

**Write a code simulating ARP protocols.**

## AIM

To implement 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);

* 1. Get the IP Address to resolve its physical address.
  2. Send the IPAddress to its output Stream.ps.println(ip);
  3. 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();

1. Get the IPaddress from its inputstream.

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

1. During runtime execute the processRuntime r=Runtime.getRuntime(); Process p=r.exec("arp -a "+ip);
2. Send the Physical Address to the client.

## RESULT

Thus the implementation of ARP is done & executed successfully.

EX NO : 5 B

DATE:

**Write a code simulating RARP protocols.**

## AIM:

To write a java program for simulating RARP protocols.

**ALGORITHM:**

**CLIENT**

* 1. Start the program
  2. using datagram sockets UDP function is established. 2.Get the MAC address to be converted into IP address. 3.Send this MAC address to server.

4.Server returns the IP address to client.

**SERVER**

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.

## RESULT:

Thus the implementation of RARP is done & executed successfully.

EX NO : 6

DATE:

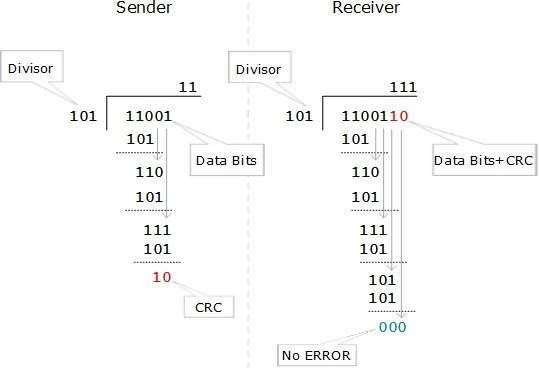
**Simulation of error correction code (like CRC)**

**AIM:**

To implement and check the error detection/error correction techniques in networks using a c program.

## Cyclic Redundancy Check (CRC)

CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as code words.



At the other end, the receiver performs division operation on codewords using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

**PROCEDURE:**

* Start the process.
* Give the data which is the message.
* Compile and run the program.
* Enter the received hamming code.
* The error is corrected codeword.

**RESULT:**

Thus the error detection/error correction techniques were implemented successfully.