**Meenakshi Sundararajan Engineering College**

Approved by AICTE and Affiliated to Anna University

Accredited by NBA for programs applied

363, Arcot Road, Kodambakkam, Chennai – 24

**DEPARTMENTOFARTIFICIAL INTELLIGENCE AND DATASCIENCE**

**LAB MANUAL**



**CS3951-COMPUTER NETWORKS LABORATORY**

**Regulation2021**

**Year/Semester:II/IV**



|  |  |
| --- | --- |
| **EXNO: 1** | **Learntousecommandsliketcpdump,netstat,ifconfig,nslookupandtraceroute.CapturepingandtraceroutePDUsusinganetworkprotocolanalyzerandexamine.** |
| **DATE:** |

## AIM:

Tostudythebasicnetworkingcommands.

**MAPPING:**

This experiment is not mapped with any of the CO’s. This is basic commands students should know to proceed with further coding.

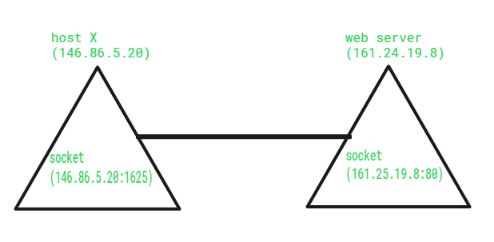
It will be thought first half an hour in lab session.

**Therory**

Socket:

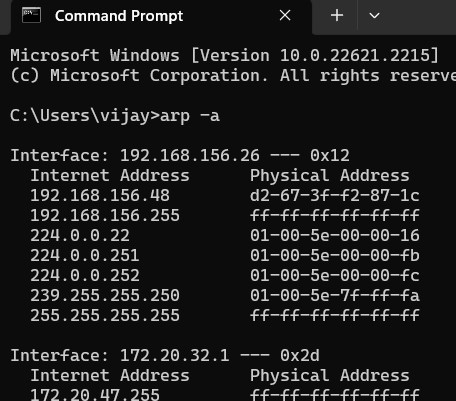
A **socket** is one endpoint of a **two way** communication link between two programs running on the network. The socket mechanism provides a means of inter-process communication (IPC) by establishing named contact points between which the communication take place

Socket are generally employed in client server applications. The server creates a socket, attaches it to a network port addresses then waits for the client to contact it. The client creates a socket and then attempts to connect to the server socket. When the connection is established, transfer of data takes place.



## COMMANDS:

C:\>arp –a: ARP is short form of address resolution protocol, It will show the IP address ofyourcomputeralong with theIP address and MACaddress of your router.



C:\>hostname: This is the simplest of all TCP/IP commands. It simply displays the name ofyourcomputer.



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



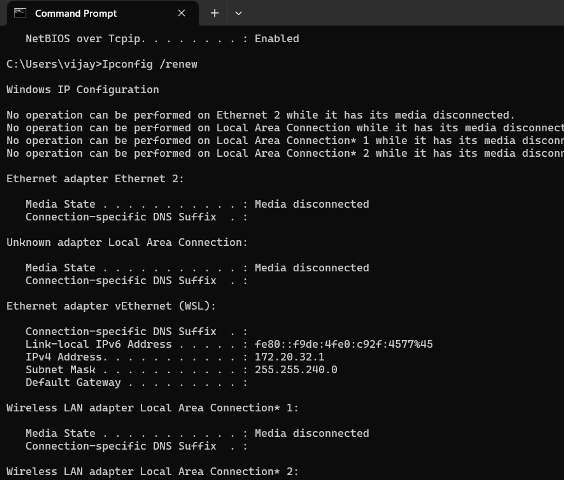


C:\>ipconfig /all: This command displays detailed configuration information about yourTCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernetadapterin your system.

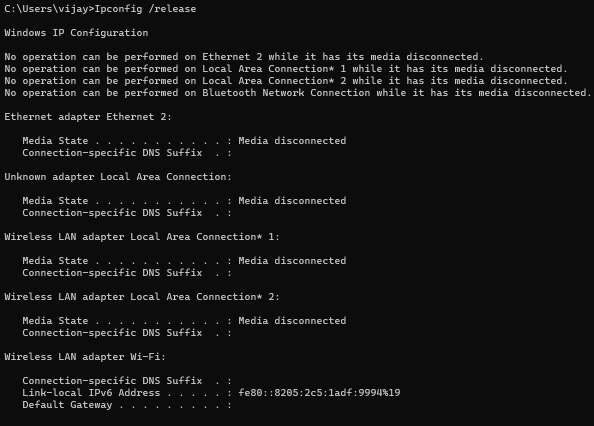


C:\>Ipconfig /renew: Using this command will renew all your IP addresses that you arecurrently (leasing) borrowing from the DHCP server. This command is a quick problemsolver if you are having connection issues, but does not work if you have been configuredwitha staticIPaddress.

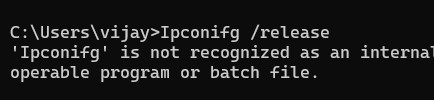




C:\>Ipconifg /release: This command allows you to drop the IP lease from the DHCPserver.

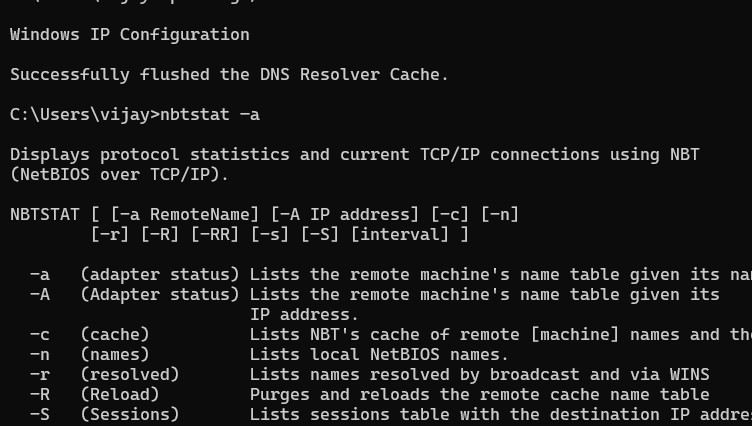


C:\>ipconfig /flushdns: This command is only needed if you’re having trouble with yournetworks DNS configuration. The best time to use this command is after networkconfigurationfrustrationsetsin,andyoureallyneedthecomputertoreplywithflushed.



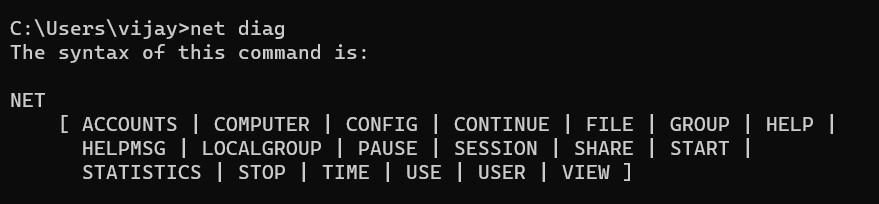
C:\>nbtstat –a: This command helps solve problems with NetBIOS name resolution.(Nbtstands for NetBIOS overTCP/IP)



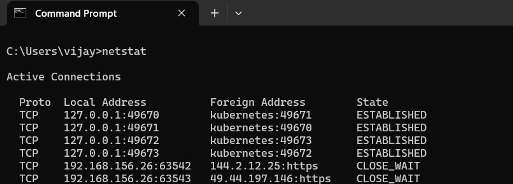


C:\>netdiag: Netdiag is a network testing utility that performs a variety of network diagnostictests, allowing you to pinpoint problems in your network. Netdiag isn’t installed by default,but can be installed from the Windows XP CD after saying no to the install. Navigate to theCDROM driveletterandopen thesupport\tools folderon theXP

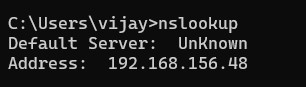
CDandclick thesetup.exeiconinthesupport\toolsfolder.



C:\>netstat: Netstat displays a variety of statistics about a computers active TCP/IPconnections. This tool is most useful when you’re having trouble with TCP/IP applicationssuchasHTTP, andFTP.

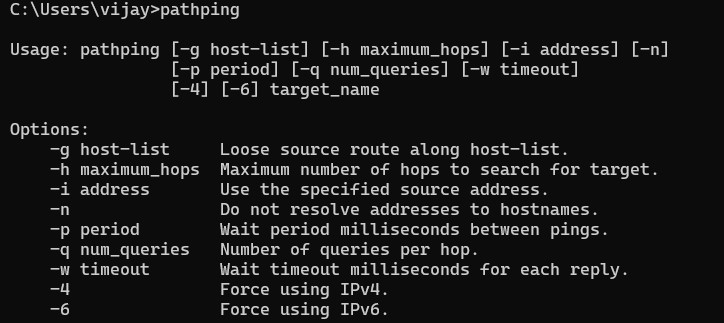


C:\>nslookup: Nslookup is used for diagnosing DNS problems. If you can access aresourcebyspecifying anIP addressbutnotit’s DNSyou havea DNSproblem.



C:\>pathping: Pathping is unique to Window’s,and isbasically a combination of the PingandTracertcommands.Pathpingtracestheroutetothedestinationaddressthenlaunchesa25 second test of each router along the way, gathering statistics on the rate of data loss alongeachhop.



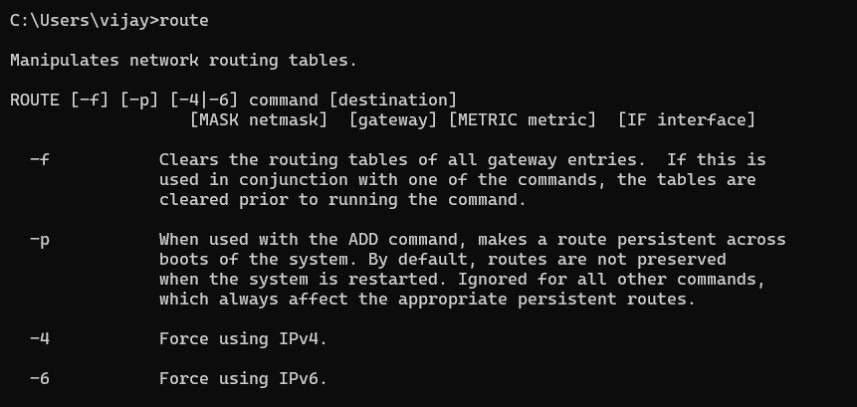


C:\>ping:PingisthemostbasicTCP/IPcommand,andit’sthesameasplacingaphonecallto your best friend. You pick up your telephone and dial a number, expecting your best friendto reply with “Hello” on the other end. Computers make phone calls to each other over anetwork by using a Ping command. The Ping commands main purposeis to place a phonecall to another computer on the network, and request an answer. Ping has 2 options it can usetoplaceaphonecalltoanothercomputeronthenetwork.ItcanusethecomputersnameorIPaddress.



C:\>route: The route command displays the computers routing table. A typicalcomputer, with a single network interface, connected to a LAN, with a router is fairlysimpleand generally doesn’tpose any networkproblems. But ifyou’rehaving trouble

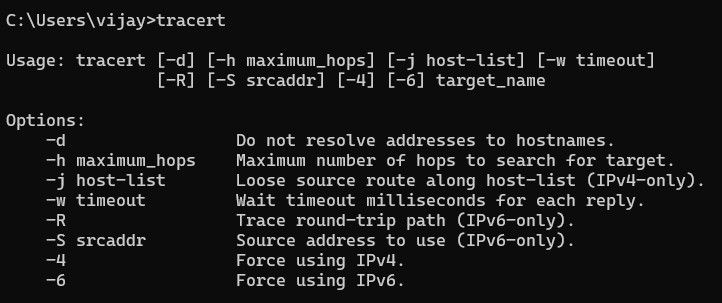
accessingothercomputersonyournetwork,youcanuse theroutecommandtomake suretheentriesintherouting tablearecorrect.



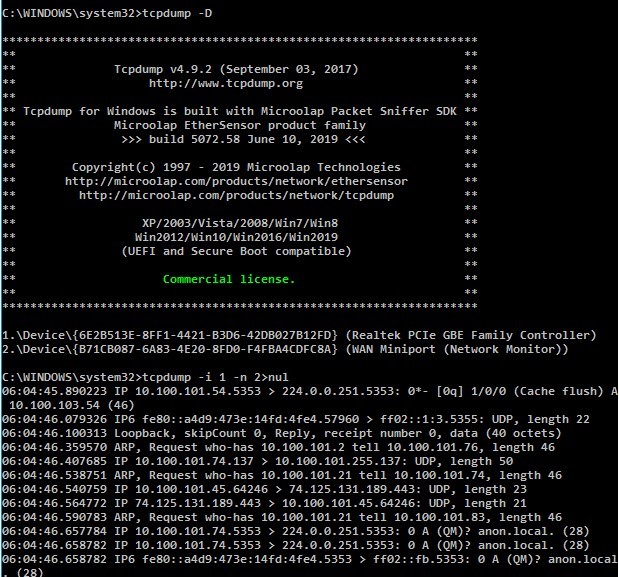


C:\>tracert:Thetracertcommanddisplaysalistofalltheroutersthatapackethastogothroughto get fromthecomputerwheretracertisrun to anyother computeron the

internet.



c:\>tcpdump: TCPDUMP is a clone of TCPDUMP, the most used network sniffer/analyzer forUNIX, compiled with the original tcpdump code (tcpdump.org), and our own packet capturetechnologyMicroolap PacketSniffer SDK(nolibpcap/WinPcap/npcap).



**Where do we apply these basic commands?**

Basic commands in networking are essential for managing and troubleshooting network

devices and connections. These commands are often used in command-line interfaces (CLI) of networking devices such as routers, switches, and firewalls



|  |  |
| --- | --- |
| **Ex.No:2** | **HTTPweb clientprogramtodownloada webpageusingTCP sockets** |
| **Date:** |

## AIM

TodownloadawebpageusingTCP sockets

**MAPPING**

CO 2: Understand the basics of how data flows from one node to another

The theory concept of this experiment is taken in Unit 1

**Theory**

# 1. Overview:

# The Hypertext Transfer Protocol (HTTP) is the foundation of data communication on the World Wide Web. It operates on a client-server model, where clients (web browsers or other applications) make requests to servers for resources like web pages.

# 2. Client-Server Communication:

# In this model, a web client initiates communication by sending a request to a web server. The server responds with the requested resource or an error message.

# 3. TCP Sockets:

# Transmission Control Protocol (TCP) provides reliable, connection-oriented communication between two devices. Sockets facilitate this communication, allowing data to be sent and received in a stream.

# 4. Steps in an HTTP Web Client Program:

# a. Create a TCP Socket:

# - The client creates a TCP socket to establish a connection with the web server. The socket is identified by the combination of the client's IP address and a unique port number.

# b. Specify Server Address and Port:

# - The client specifies the server's IP address and port number to which it wants to connect. Commonly, HTTP uses port 80 for communication.

# c. Establish a Connection:

# - The client uses the connect method to establish a connection to the server. A three-way handshake occurs to ensure a reliable connection.

# d. HTTP GET Request:

# - The client sends an HTTP GET request to the server to request a specific resource, like a web page. The request includes the method (GET), the resource's URI, and the HTTP version.

# e. Server Response:

# - The server responds with an HTTP response, including a status line, headers, and the content of the requested resource. The status line indicates the success or failure of the request.

# f. Handle Redirects:

# - The client checks the status code in the response. If it's a redirect (e.g., 301 or 302), it follows the new location specified in the response headers and sends another request.

# g. Process Content:

# - The client processes the content received from the server. This may involve saving the content to a file or displaying it to the user.

# h. Close Connection:

# - After completing the communication, the client closes the TCP connection using the close method.

# ALGORITHM

1. Starttheprogram
2. Importthesocketmodule
3. DefinetheHostaddressandtheporttobeconnected
4. MakeasocketbybindingHostandportwith"connect"function
5. Requesttheclientwiththe"sendall"function
6. Recievethepacketswithrecvanddefineittoaresponsevariable
7. Decodetheresponse
8. Printtheresponse
9. Close the socket
10. Endtheprogram

Purpose of this experiment:

The purpose of the provided client-side program is to establish a TCP connection to a server, send a URL to the server, receive data from the server, and store that data into a file on the client side.



|  |  |
| --- | --- |
| **EXNO:3A** | **ApplicationsusingTCPsockets-Echoclientandechoserver** |
| **DATE:** |

**AIM**

Towriteasocketprogramforimplementationofechoclientandechoserver

**MAPPING**

CO 2: Understand the basics of how data flows from one node to another

The theory concepts of this experiment is taken in unit 2

**Theory**

Implementing an echo server and client using TCP sockets is a common exercise to understand network programming. In this scenario, the server will listen for incoming connections, and the client will connect to the server. Whatever the client sends to the server, the server echoes back to the client.

**Echo server:**

The echo server's primary purpose is to echo back any data it receives from a client. It listens for incoming connections on a specified port, receives data from connected clients, and sends the received data back to the clients.

Echo client:

The echo client connects to the echo server, sends a message, and waits for the server to echo the same message back. It's a simple way to test the bidirectional communication between a client and a server.

**ALGORITHM**

**CLIENTSIDE**

1. Starttheprogram.
2. CreateasocketwhichbindstheIpaddressofserverandtheportaddresstoacquireservice.
3. Afterestablishingconnectionsendadatatoserver.
4. Receiveandprintthesamedatafromserver.
5. Closethesocket.
6. Endtheprogram.

**SERVERSIDE**

1. Starttheprogram.
2. Createaserversockettoactivatetheportaddress.
3. Createasocketfortheserversocketwhichacceptstheconnection.
4. Afterestablishingconnectionreceivethedatafromclient.
5. Printandsendthesamedatato client.

6.Closethesocket.

1. Endtheprogram.

**Purpose**

It is a simply way to test the bidirectional communication between a client and a server

**RESULT**

Thustheprogramforsimulationofechoserverwaswritten&executed



|  |  |
| --- | --- |
| **EXNO:3B** | **ApplicationsusingTCPsockets–Chatapplication** |
| **DATE:** |

# AIM

Towriteasocketprogramforimplementationofclient-serverchatapplicationusingTCP

**Mapping**

CO 2: Understand the basics of how data flows from one node to another

The theory concept of this experiment is taken in Unit II

**Theory**

Developing a chat application using TCP sockets involves creating a system where multiple clients can connect to a server, send messages to each other, and receive messages in real-time.

# ALGORITHM

1. Start the program.
2. Define the HOST and the PORT.
3. Start Listening for connection.
4. When a connection is establishes display the message as ‘connection established’.
5. After connecting, prompt a message to input text.
6. Encode the message and send it to the other party.
7. Wait for the other party to send their message.
8. End the program when the connection is lost.

**RESULT**

Thustheaboveprogramaclient-serverapplicationforchatusingTCP/IPwasexecutedandsuccessfully.



|  |  |
| --- | --- |
| **EXNO:4** | **CAPTURINGAND EXAMINING THE PACKETS USING WIRESHARK TOOL** |
| **DATE:** |

**AIM:**

# To capture and examine the packets using wireshark tool

# 

# MAPPING

CO 2: Understand the basics of how data flows from one node to another.

This concept is taken in UnitII

**THEORY**

Wireshark is an open-source network protocol analyzer that allows users to capture and inspect the data traveling back and forth on a network in real-time. It supports various protocols and provides a detailed view of network traffic.

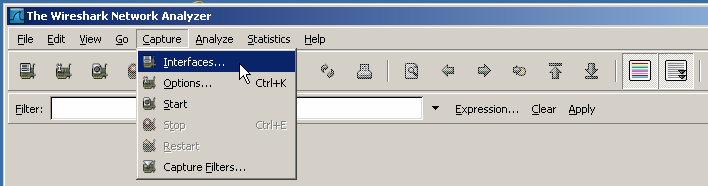
Packet capturing involves intercepting and logging data packets as they traverse a network. Wireshark captures these packets by putting the network interface into promiscuous mode, allowing it to see all network traffic, not just traffic specifically destined for the capturing machine.

Wireshark provides powerful filtering capabilities to focus on specific types of traffic. Filters can be based on protocols, IP addresses, port numbers, and various other criteria. This helps in isolating and analyzing relevant packets.

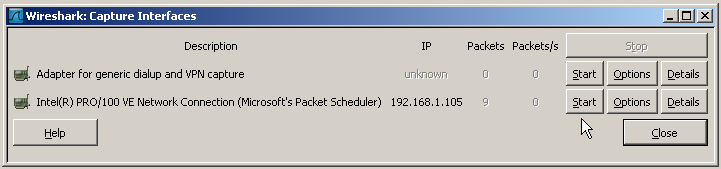
## PROCEDURE:

## Step1:StartWireshark. Double-clicktheWiresharkicon,whichislocatedonthedesktop.

## Step2:Selectaninterfacetouseforcapturingpackets. FromtheCapturemenu,choose Interfaces.

****

Step3:Startanetworkcapture.

1. ChoosethelocalnetworkEthernetinterfaceadapterforcapturingnetworktraffic.ClicktheStart button of the chosen interface.
2. WritedowntheIPaddressassociatedwiththeselectedEthernetadapter,becausethatisthesource IP address to look for when examining captured packets.

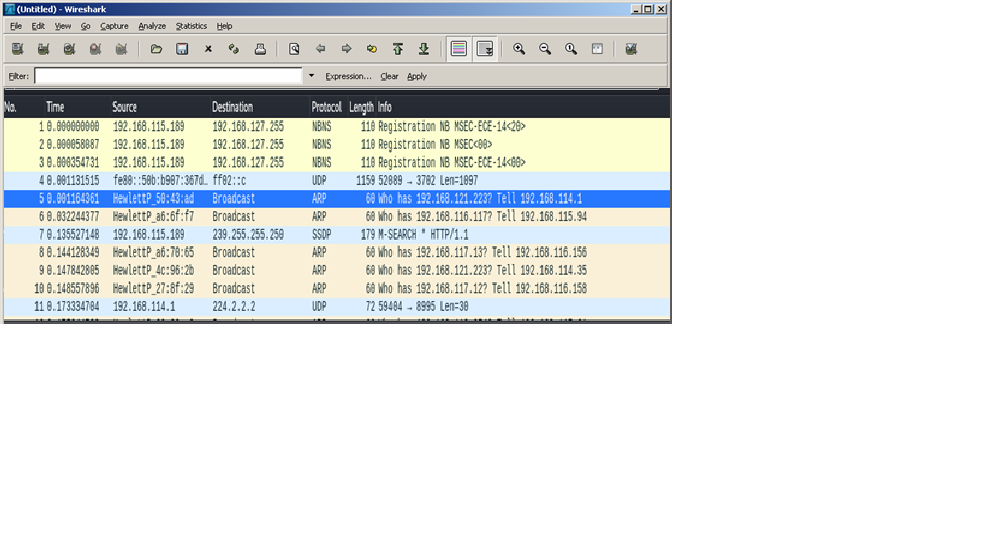
# GenerateandAnalyzeCapturedPackets

## Step1:Openabrowserandaccessawebsite.

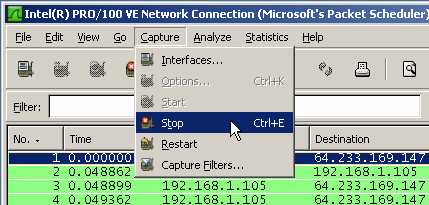
1. Goto[www.google.com.](http://www.google.com/)MinimizetheGooglewindow,andreturntoWireshark.Youshouldsee captured traffic similar to that shown below.

Note:Yourinstructormayprovideyouwithadifferentwebsite.Ifso,enterthewebsitenameor address here:

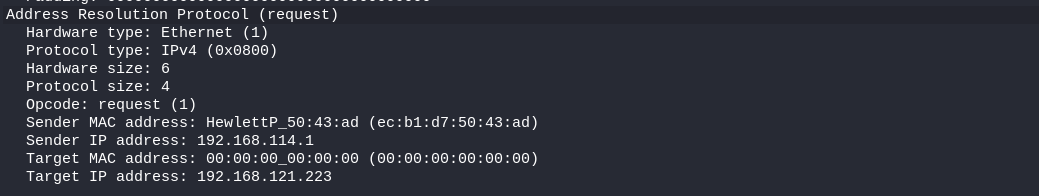
1. Thecapturewindowsarenowactive.LocatetheSource,Destination,andProtocolcolumnsonthe Wireshark display screen. The HTTP data that carries web page text and graphics uses TCP for reliability.

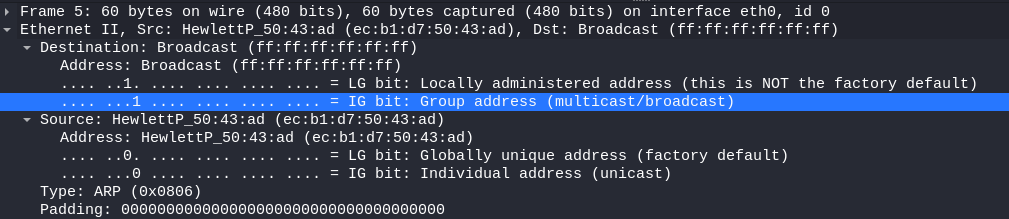


## Step2:Stopthecapture. FromtheWiresharkCapturemenu,chooseStop.



## Step3:Analyzethecapturedoutput.





|  |  |
| --- | --- |
| **EXNO:5** | Simulation of DNS using UDP sockets. |
| **DATE:** |

Aim

To Simulate DNS using UDP sockets

MAPPING

CO 2: Understand the basics of how data flows from one node to another.

This concept is taken in Unit2

Theory

Simulating DNS (Domain Name System) using UDP (User Datagram Protocol) sockets in Python involves creating a simple DNS server that listens for DNS queries over UDP and responds with the corresponding IP address.

DNS

DNS, or Domain Name System, is a decentralized hierarchical system that translates human-readable domain names into IP addresses, which are used by computers to identify each other on a network. Instead of users needing to remember numerical IP addresses (like 192.168.1.1), they can use domain names (like www.example.com) to access websites, services, and other resources on the internet.

Coding

UDP

It provides a simple, connectionless communication service that operates on top of the Internet Protocol.

UDP is commonly used in scenarios where low latency and quick data transmission are crucial, such as real-time multimedia applications (voice and video streaming), online gaming, and DNS (Domain Name System) queries. These applications can tolerate occasional packet loss and prefer the speed and simplicity offered by UDP.

Result:

Thus the program is executed and verified

|  |  |
| --- | --- |
| **EXNO:6** | Write a code simulating ARP /RARP protocols.. |
| **DATE:** |

Aim

To Write a code simulating ARP /RARP protocols..

**MAPPING**

CO 3: Analyze routing algorithms.

This concept is taken in Unit 3.

**THEORY**

ARP, or Address Resolution Protocol, is a network protocol used **to map an IP address (Internet Protocol address) to the corresponding hardware or MAC (Media Access Control) address in a local network**. ARP is essential for communication between devices within the same network, particularly in Ethernet-based networks.

**step-by-step explanation of how ARP works:**

**Need for ARP:**

Devices in a network communicate using IP addresses at the network layer (Layer 3 of the OSI model). However, data link layer protocols, such as Ethernet, use MAC addresses for communication within a local network. ARP bridges the gap between these two layers.

**ARP Request:**

When a device in a local network wants to communicate with another device but only knows its IP address, it broadcasts an ARP request message to the entire network. The ARP request contains the IP address for which the device is seeking the MAC address.

**ARP Response:**

The device with the corresponding IP address replies to the ARP request with its MAC address. This response is unicast (directed to the requesting device).

**ARP Table**:

The requesting device updates its ARP table, associating the IP address with the MAC address it received in the ARP response. The ARP table is a local cache that stores mappings between IP addresses and MAC addresses for quick reference.

**Address Resolution:**

With the MAC address known, the device can now construct the data link layer frame and send the data packet to the intended recipient using the MAC address.

**ARP Caching:**

To optimize network performance, ARP results are cached in memory. Cached entries have a time-to-live (TTL) value, and when this time expires, the device may send another ARP request to refresh the entry.

ARP is a stateless protocol, meaning devices do not maintain a continuous connection. Instead, they update their ARP tables dynamically as needed. While ARP is efficient for small local networks, it does not scale well to larger networks or the internet. In such cases, routers and switches play a role in forwarding packets to their destination.

ARP is a fundamental component of local network communication, ensuring that devices can discover each other's MAC addresses dynamically, making IP-based communication possible within the local network.

**Real time application**

**Local Network Communication:**

Scenario: When devices within the same local network need to communicate.

Role of ARP: ARP resolves the MAC addresses corresponding to IP addresses within the local network, enabling direct communication between devices at the data link layer.

**Ethernet Communication:**

Scenario: In Ethernet-based networks.

Role of ARP: ARP is commonly used in Ethernet networks to map IP addresses to MAC addresses, allowing devices to communicate using Ethernet frames.

**IP Address Configuration:**

Scenario: When a device joins a network and needs to obtain its IP address.

Role of ARP: ARP is involved in the DHCP (Dynamic Host Configuration Protocol) process, where a device requests an IP address assignment. ARP helps devices discover the MAC address of the DHCP server.

**Dynamic Host Configuration:**

Scenario: Dynamic assignment of IP addresses to devices on a network.

Role of ARP: ARP plays a role in obtaining the MAC address of the DHCP server, allowing devices to configure their IP addresses dynamically.

**Network Troubleshooting:**

Scenario: Identifying and resolving network connectivity issues.

Role of ARP: Network administrators use ARP commands to troubleshoot connectivity problems by inspecting and manipulating ARP tables on devices. ARP information helps diagnose issues related to IP and MAC address mappings.

**Virtual Local Area Networks (VLANs):**

Scenario: Communication within segmented or virtualized local networks.

Role of ARP: ARP facilitates communication within VLANs by resolving IP addresses to MAC addresses, even in segmented network environments.

**Router Operation:**

Scenario: Communication between devices in different subnets.

Role of ARP: ARP is used to discover the MAC address of the router or gateway when devices need to communicate across different subnets. The router's MAC address is used to encapsulate and forward packets between subnets.

**Wireless Networks (Wi-Fi):**

Scenario: Communication within wireless local networks.

Role of ARP: ARP is used in Wi-Fi networks to map IP addresses to MAC addresses, enabling communication among devices connected to the same wireless network.

**IPv6 Networks:**

Scenario: Communication in IPv6 networks.

Role of ARP: In IPv6, the Neighbor Discovery Protocol (NDP) replaces ARP. NDP serves similar purposes, including address resolution, neighbor unreachability detection, and router discovery.

ARP is fundamental to the efficient operation of local networks, ensuring that devices can dynamically discover and communicate with each other using their MAC addresses. Its real-time applications are pervasive in everyday network operations.

Result

Thus the program is executed and output is verified.

|  |  |
| --- | --- |
| **EXNO:7** | Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS |
| **DATE:** |

Aim:

To Study Network simulator (NS) and Simulation of Congestion Control Algorithms using NS

MAPPING

CO 2: Understand the basics of how data flows from one node to another.

This concept is explained in unit 2

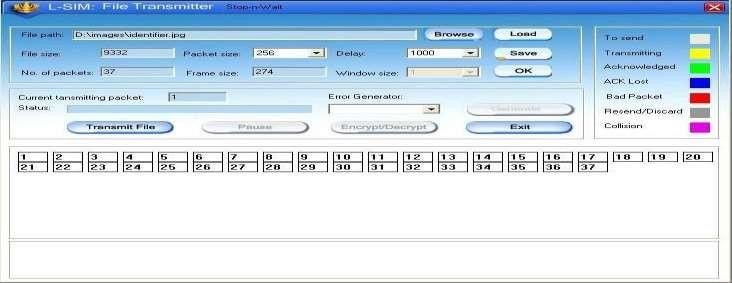
Theory

network simulator:

A network simulator is a software tool or platform that allows the simulation of computer networks for research, testing, and educational purposes. It provides a virtual environment where users can model, configure, and analyze the behavior of computer networks without the need for physical hardware. Network simulators are valuable tools for understanding, testing, and optimizing networking protocols, algorithms, and applications.

**L-SIM LAN Protocol Simulator & Analyzer Software:**

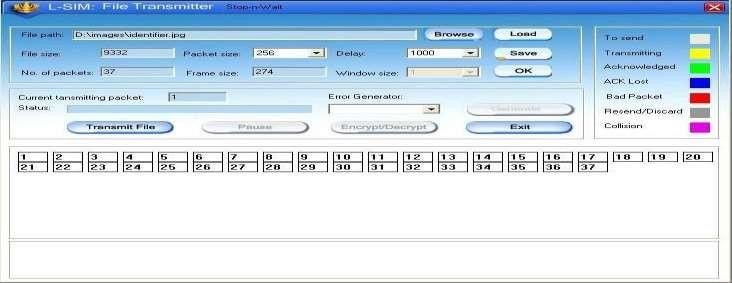
It is designed to teach the basic concepts, topologies & various protocols involved in networking. The software is provided with analysis of protocols, different layers, network and measurement of error rate and throughput.Software is supported with neat operating instruction manual and online help.

**MODEL WINDOW DIAGRAM FOR L-SIM**

**N-S IM Network simulation software:**

It is developed to provide basic understanding and implementation of various advanced concepts in networking. The software provides an opportunity to understand network fundamentals through animations & simulations.

The simulation provides for network experimentation with various LAN and WAN protocols, network devices, routers, encryption, decryption, file transfer, error insertion and analysis of error rate and throughput etc.

**MODEL WINDOW DIAGRAM FOR N-SIM**

**RESULT:**

Thus the study of network simulator (ns) and simulation of congestion control algorithms using ns is executed and verified.

|  |  |
| --- | --- |
| **EXNO:8** | Study of TCP/UDP performance using Simulation tool |
| **DATE:** |

Aim:

To Study of TCP/UDP performance using Simulation tool

MAPPING

CO 2: Understand the basics of how data flows from one node to another.

This concept is taught in second unit

THEORY

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are both transport layer protocols in the Internet Protocol (IP) suite

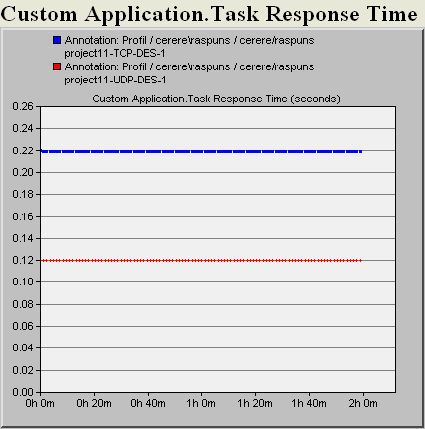
UDP

UDP is a lightweight, fast, and connectionless protocol that is well-suited for applications where low latency and quick data transmission are more critical than reliability.

|  |  |  |
| --- | --- | --- |
| **Service** | **TCP** | **UDP** |
| Guaranteedmessagedelivery | Returnsacknowledgments. | UDPdoesnotreturnACKs |
| Congestioncontrols | NetworkdevicescantakeadvantageofTCPACKtocontrolthe  behaviorofsender. | IfACK,aremissing,thenetwork  cannotsignalcongestiontothesender. |

**SIMULATIONRESULTS:**

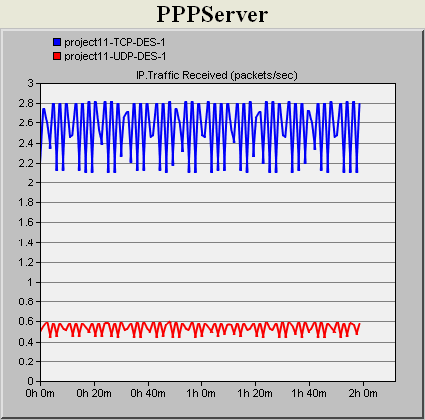
The simulation time is set for two hours data transfer between LAN network and the serverwith no packet latency and packet discard ratio of 0% while packets traverse thru the WAN.The task response time, in seconds, Fig. 1, shows how long the application need to becompleted.Thetimewhenusing TCP to completethetask isgreaterthattheoneusing UDP.



TCP

UDP

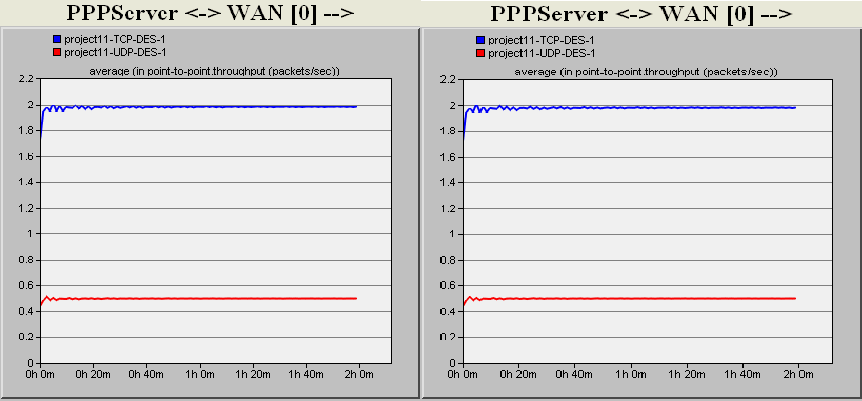
ResponsetimeforTCPandUDP



TCP

UDP

Trafficreceived(packets/sec)fortheserver



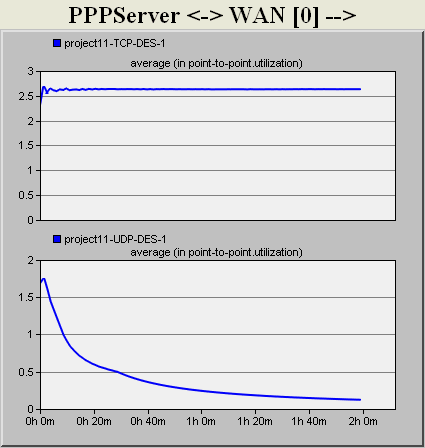
TCP

TCP

UDP

UDP

Traffic/LinkutilizationfromtheWANtotheserver



TCP

UDP

Linkutilizationwitha0.5%packetsdiscardratio

ThemaindifferencebetweenthesetwoprotocolsisthatTCPprovidesreliabilityandcongestioncontrolservices,whileUDPisorientatedto improveperformance.

The most important and common thing that TCP and UDP are using is the ability to set ahost-to-hostcommunicationchannel,sothepacketswillbedeliveredbetweenprocessesrunningontwodifferentcomputers.

# RESULT:

ThustheTCP/UDP performancehas beensimulated successfullyusingOPNET.