Dr. D. Y.Patil Unitech Society

Dr. D. Y. Patil Institute of Technology

Sant Tukaram Nagar, Pimpri, Pune-411 018.

DEPARTMENT OF COMPUTER ENGINEERING

LABORATORY MANUAL

Third Year

310248: Computer Networks and Security Lab

Course Incharge

Dr. Vinod Kimbahune

Dr. Kapil Vhatkar

Prof. Sukhes Kothari

Academic Year 2022-23

SYLLABUS STRUCTURE

Savitribai Phule Pune University

Third Year of Computer Engineering (2019 Course)

310248: Computer Networks and Security Laboratory

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Practical: 02 Hours/Week	**01**	**Term Work: 25 Marks** **Oral:25 Marks**

Companion Course : Computer Network and Security (310244)

Course Objectives:

- To learn computer network hardware and software components
- To learn computer network topologies and types of network
- To develop an understanding of various protocols, modern technologies and applications
- To learn modern tools for network traffic analysis
- To learn network programming

Course Outcomes:

On completion of the course, learners will be able to

CO1: Analyze the requirements of network types, topology and transmission media

CO2: Demonstrate error control, flow control techniques and protocols and analyze them

CO3: Demonstrate the subnet formation with IP allocation mechanism and apply various routing algorithms

CO4: Develop Client-Server architectures and prototypes

CO5: Implement web applications and services using application layer protocols

CO6: Use network security services and mechanisms

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, punctuality.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Operating System recommended: -64-bit Open-source Linux or its derivative

Programming tools recommended: - Open-Source /C/C++/JAVA

Programming tool like G++/GCC, Wireshark/Ethereal and Packet Tracer.

Virtual Laboratory: http://vlabs.iitb.ac.in/vlab/

	Suggested List of Laboratory Experiments/Assignments
Sr. No.	**Group A (Unit I and II): Attempt any two assignments from Sr. No. 1 to 3. Assignments 4 and 5 are compulsory.**
1	Setup a wired LAN using Layer 2 Switch. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and demonstrating the PING packets captured traces using Wireshark Packet Analyzer Tool.
2	Demonstrate the different types of topologies and types of transmission media by using a packer tracer tool.
3	Setup a WAN which contains wired as well as wireless LAN by using a packet tracer tool. Demonstrate transfer of a packet from LAN 1 (wired LAN) to LAN2 (Wireless LAN).
4	Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.
5	Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.
Group B (Unit III and IV)	
6	Write a program to demonstrate Sub-netting and find subnet masks.
7	Write a program to implement link state /Distance vector routing protocol to find suitable path for transmission.
8	Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF/BGP
9	Write a program using TCP socket for wired network for following a. Say Hello to Each other b. File transfer c. Calculator
10	Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines.

	Group C (Unit V and VI): Assignment Sr. No. 11 is Compulsory and attempts any four from Assignments Sr. No 12 to 17.
11	Write a program for DNS lookup. Given an IP address as input, it should return URL and vice-versa.
13	Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in the code.
14	Write an X86/64 ALP password program that operates as follows: a. Do not display what is actually typed instead display asterisk (“*”). If the password is correct display, “access is granted” else display “Access not Granted”
15	Study Assignment: Motherboards are complex. Break them down, component by component, and Understand how they work. Choosing a motherboard is a hugely important part of building a PC. Study- Block diagram, Processor Socket, Expansion Slots, SATA, RAM, Form Factor, BIOS, Internal Connectors, External Ports, Peripherals and Data Transfer, Display, Audio, Networking, Overclocking, and Cooling. 4. https://www.intel.in/content/www/in/en/support/articles/000006014/boards-and-kits/desktop-boards.html

@The CO-PO Mapping Matrix

CO\ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	-	2	2	-	1	1	-	-	1	-	1
CO2	-	3	-	1	1	-	-	1	-	-	-	-
CO3	3	2	1	1	-	-	-	1	-	-	1	1
CO4	-	1	2	1	1	1	-	-	-	-	-	1
CO5	2	3	-	-	1	-	-	-	1	-	-	-
CO6	-	1	3	1	1	-	1	-	2	-	-	1

Sr. No	Group	Asg No.	TITLE: of Assignment	CO	PO
1.		**1.**	**Lab Assignment on Unit I:** **(Mandatory Assignment)** **Part A:** Setup a wired LAN using Layer 2 Switch and then IP switch of minimum four computers. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and demonstrate the PING packets captured traces using Wireshark Packet Analyzer Tool. **Part B:** Extend the same Assignment for Wireless using Access Point	CO1,CO3	PO1,PO2, PO3,PO5
2.		**2.**	**L**ab Assignment on Unit II: (Use C/C++) Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.	CO2	PO1,PO2, PO3
3.		**3.**	**Lab Assignment on Unit IV:** **(Use JAVA/PYTHON)** Write a program to demonstrate subnetting and find the subnet masks.	CO2	PO1,PO2, PO3
4.		**4.**	**Lab Assignment on Unit VI:** **(Use JAVA/PYTHON)** Write a program for DNS lookup. Given an IP address input, it should return URL and vice- versa.	CO2,CO4	PO1,PO2, PO3
5.		**5.**	**Lab Assignment on Unit V:** **(Mandatory Assignment) (Use C/C++)** Write a program using TCP socket for wired network for following a. Say Hello to Each other (For all students) b. File transfer (For all students) c. Calculator (Arithmetic) (50% students) d. Calculator (Trigonometry) (50% students) packet Analyzer Tool for peer to peer mode.	CO2,CO4	PO1,PO2, PO3,PO5

6.		**6.**	**Lab Assignment on Unit V:** **(Mandatory Assignment) (Use C/C++)** Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.	CO2,CO4	PO1,PO2, PO3,PO5
7.		**7.**	**Lab Assignment on Unit V:** **(Mandatory Assignment) (Use C/C++)** Write a program to analyze following packet formats captured through Wireshark for wired network. 1. Ethernet 2. IP 3.TCP 4. UDP	CO2,CO4	PO1,PO2, PO3,PO5
8.		**8.**	**Lab Assignment on Unit II:** **(Use JAVA/PYTHON)** Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.	CO2	PO1
9.	**B** **(Any 6)**	**9.**	**Lab Assignment on Unit VI:** Installing and configure DHCP server and write a program to install the software on remote machine.	CO1,CO3, CO4	PO1,PO2, PO3,PO5
10.		**10.**	**Lab Assignment on Unit V:** **(Use JAVA/PYTHON)** Write a program using TCP sockets for wired network to implement a. Peer to Peer Chat b. Multiuser Chat Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.	CO2,CO4	PO1,PO2, PO3,PO5
11.		**11.**	**Lab Assignment on Unit V:** **(Use JAVA/PYTHON)** Write a program using UDP sockets for wired network to implement a. Peer to Peer Chat b. Multiuser Chat Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.	CO2,CO4	PO1,PO2, PO3,PO5

12.		**12.**	**Lab Assignment on Unit IV and Unit V: (Mandatory Assignment)** Use network simulator NS2 to implement: a. Monitoring traffic for the given topology b. Analysis of CSMA and Ethernet protocols Network Routing: Shortest path routing, AODV. c. Analysis of congestion control (TCP and UDP).	CO2,CO4	PO1,PO, PO5,PO12
13.		**13.**	**Lab Assignment on Unit IV: (Mandatory Assignment)** Configure RIP/OSPF/BGP using packet Tracer.	CO2,CO3, CO4	PO1,PO2, PO5,PO12
14.	**Content Beyond Syllabus**	**14.**	**Content Beyond Syllabus:** Demonstration of DOS attack by using ping of death command	CO2,CO4	PO1,PO2, PO5
15.	**VLAB**	**15.**	**Packet Send and Receive in Unicast and Broadcast manner in Wireless Networking.**	CO2,CO4	PO1,PO2, PO5

Group A

Assignment No. 1

AIM : Setup a wired LAN using Layer 2 Switch and wireless LAN using Access point.

PROBLEM STATEMENT: Part A: Setup a wired LAN using Layer 2 Switch and then IP switch of minimum four computers. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and demonstrate the PING packets captured traces using Wireshark Packet Analyzer Tool.

Part B: Extend the same Assignment for Wireless using Access Point

PREREQUISITES: Knowledge of Wired and wireless media, networking devices and protocols**.**

COURSE OBJECTIVES: To understand and setup a wired and wireless LAN and configuring the nodes.

COURSE OUTCOMES:

CO1: Demonstrate LAN and WAN protocol behavior using Modern

Tools CO3: Demonstrate basic configuration of switches and routers

COURSE OUTCOME MAPPED : CO1 **:**Analyze the requirements of network types, topology and transmission media

PROGRAM OUTCOME MAPPED : PO1,PO3,PO5,PO12

PROGRAM SPECIFIC OUTCOME MAPPED : PSO1

THEORY:

To implement the cross-wired cable and straight through cable using crimping tool.

Components:

RJ-45 connector, Crimping Tool, Twisted pair Cable

Procedure:

To do these practical following steps should be done:

1. Start by stripping off about 2 inches of the plastic jacket off the end of the cable. Be very careful at this point, as to not nick or cut into the wires, which are inside. Doing so could alter the characteristics of wire cable, or even worse render is useless. Check the wires, one more time for nicks or cuts. If there are any, just whack the whole end off, and start over.

2. Spread the wires apart, but be sure to hold onto the base of the jacket with wire in the other hand. You do not want the wires to become untwisted down inside the jacket. Category 5 cable must only have 1/2 of an inch of 'untwisted' wire at the end; otherwise it will be 'out of spec'. At this point, we obviously have more than 1/2 of an inch of un-twisted wire.

3. We have 2 end jacks, which must be installed on our cable. If we are using a premade cable, with one of the ends whacked off, we only have one end to install - the crossed over end. Below are two diagrams, which show how we need to arrange the cables for each type of cable end. Decide at this point which end we are making and examine the associated picture below.

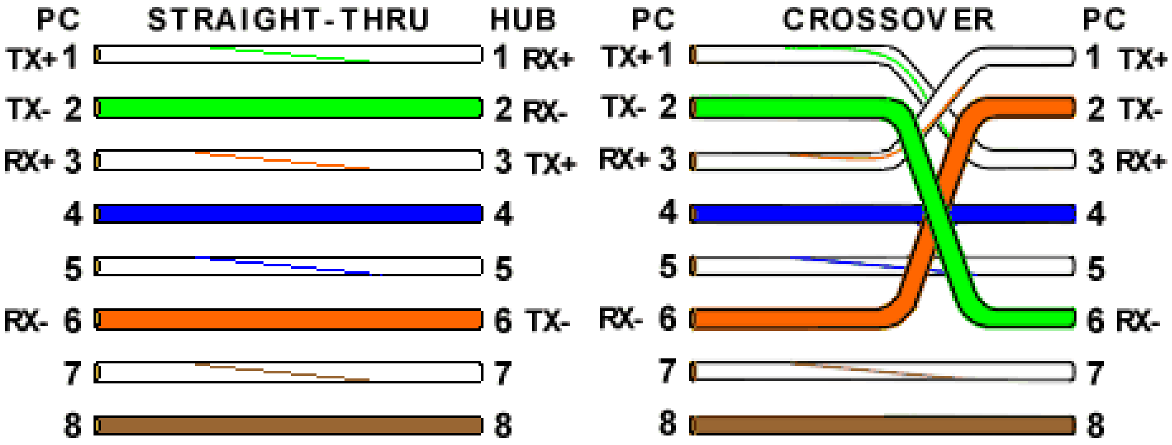


Figure: Cross wired and straight connection.

Network Hardware Devices:

Switch:

A switch or switching hub is a computer networking device that connects multiple nodes together. Switches that additionally process data at the network layer (layer 3 and above) are often referred to as Layer 3 switches or multilayer switches. The basic function that any switch is supposed to perform is to receive information from any source connected to it and dispatch that information to the appropriate destination only.

Access point

Wireless access points (APs or WAPs) are special-purpose communication devices on Wireless Access LAN (WLAN). Access points act as a central transmitter and receiver of wireless radio signals.

Configuring Access Point:

- Enable/Disable: Enables or disables the device’s wireless access point functions.
- SSID: The Service Set Identifier used to identify the network. Most access points have well-known defaults.
- Allow broadcast SSID to associate? Disables the access point’s periodic broadcast of the SSID. Normally, the access point regularly broadcasts its SSID so that wireless devices that come within range can detect the network and join in. For a more secure network, we can disable this function. Then, a wireless client must already know the network’s SSID in order to join the network.Channel: Lets we select one of 11 channels on which to broadcast. All the access points and computers in the wireless network should use the same channel. We may be experiencing interference from a cordless phone or other wireless device operating on the same channel.
- WEP — Mandatory or Disable: Lets you use a security protocol called ***wired equivalent privacy.***
- Take a wireless device for connecting multiple nodes (Access point or wifi router).
- Access the AP from any remote machine.
- Configure access point by giving suitable password and network name.
- Add multiple node’s MAC address (48 bit) in Wireless MAC Filter provided in Wireless Field.
- Save the settings.

Required Equipment for LAN

1. Two or Three PCs (with Linux OS).
2. PC’s should be equipped with Network Interface Cards.
3. One switch/Access point.
4. Cables.

Set-up a physical LAN

a. First, determine which cables are necessary for the available hardware
b. Insert one end of the cable into the Switch and the other end of the cable into the host computer.
c. Connect all host computers to the Switch and configure the IP address to every host.
192.168.1.1   192.168.1.2   192.168.1.3

Host set-up (e.g. IP address and netmask) using ifconfig

a. Boot each host machine and log on with your user-name.

b. open Network connections and assign IP address, mask address for all the hosts

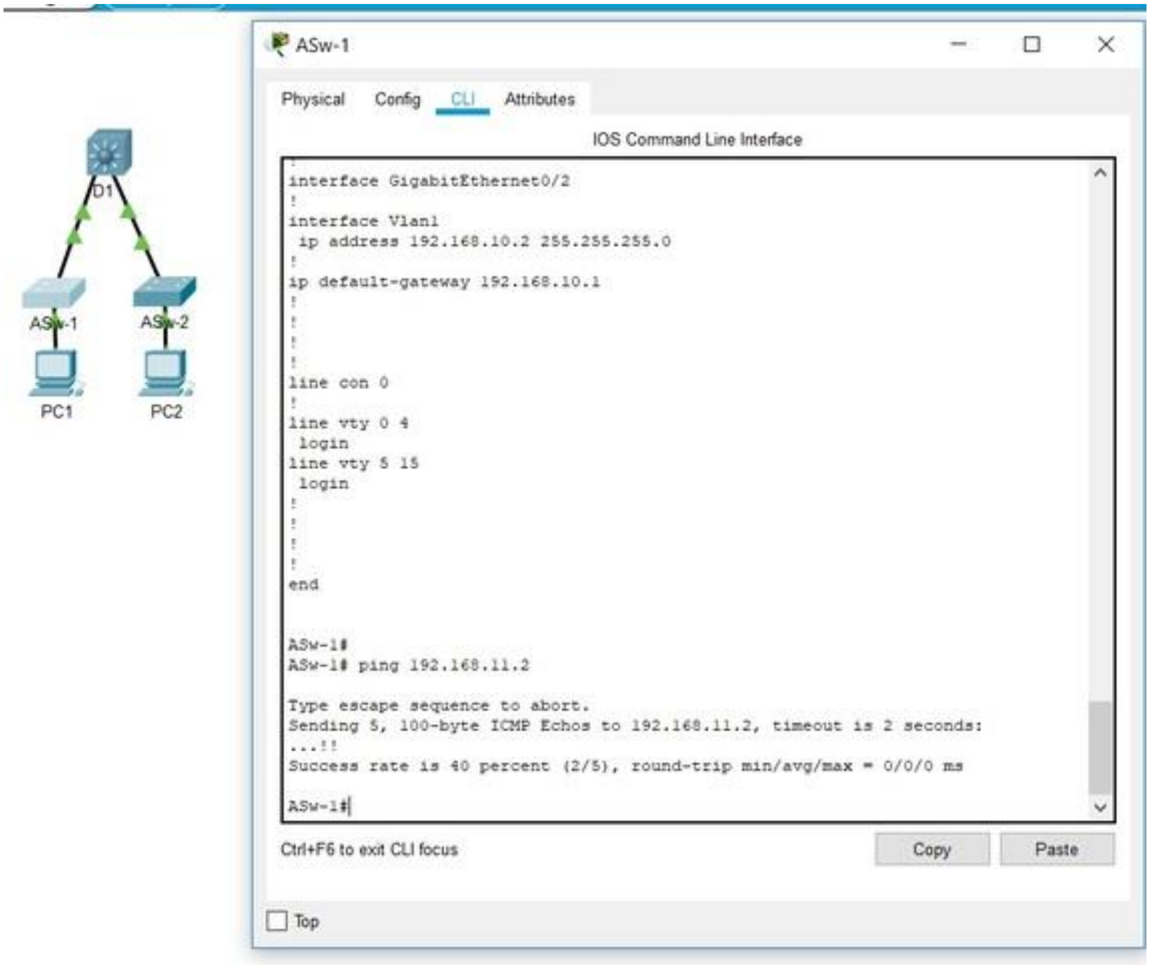
c. Check network interface settings using **ifconfig**

# ifconfig –a

d. Verify that eth0 exists on each host by checking the output from above

e. Verify communication among the machines by using the **ping** command.

OUTPUT :



```
interface GigabitEthernet0/2
!
interface Vlan1
 ip address 192.168.10.2 255.255.255.0
!
ip default-gateway 192.168.10.1
!
!
!
!
line con 0
!
line vty 0 4
 login
line vty 5 15
 login
!
!
!
!
end


ASw-1#
ASw-1# ping 192.168.11.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.11.2, timeout is 2 seconds:
...!!
Success rate is 40 percent (2/5), round-trip min/avg/max = 0/0/0 ms

ASw-1#
```

# ping 192.168.1.2

Install Wireshark Tool Using:

```
#sudo apt-get install wireshark
```

Run the wireshark tool and monitor the packets in the network.

CONCLUSION :

The Outcome of the assignment is :

1. Students are able to set up a LAN(with wired and Wireless techniques)

FAQs:

1.What are the characteristics of Switch?

Ans: A switch is a layer 2 connecting device or networking device, it perform store and forward approach.

2. Compare switch and Router

Sr. No	**Switch**	**Router**
Function	Directs data in a network. Passes data between home computers, and between computers and the modem.	Allow to connect multiple networks.
Layer	Network	Data Link Layer
Transmission Type	At Initial Level Broadcast then Unicast & Multicast	First broadcast; then unicast & multicast as needed.
Used for	Connecting two or more networks	Connecting two or more nodes in the same network or different network

3. What is the difference between straight and cross cable?

Ans: Straight Cable Connect different types of devices together e.g.a computer to a switch/hub's normal port. Crossover Cable Connect similar types of devices together e.g. 2 computers directly.

4. What is the role of NIC?

Ans: The function of Network Interface Card is to allow computers to connect to networks. One Network card can handle a number of Ethernet-connections by attaching a switch or router to it.

5. What is the purpose of Wireshark?

Ans: Wireshark is a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.

Group: A

Assignment No.2

AIM : Write a program to implement Hamming Code and CRC.

PROBLEM STATEMENT: Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

PREREQUISITES: OSI layers and Functionalities

COURSE OBJECTIVES: To study and implement Hamming code and CRC

COURSE OUTCOMES:

CO2: Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

COURSE OUTCOME Mapped:

CO2:Demonstrate error control, flow control techniques and protocols and analyze them

Program OUTCOME Mapped: PO1,PO2,PO4,PO5,PO12

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering specialization to the solution of complex engineering problems.

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

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

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

Program Specific OUTCOME Mapped: PSO1:

PSO1: Professional Skills- The ability to understand and develop the software systems by apply the concepts and techniques in the areas related to data structures, algorithms, system software, networking, multimedia, web design and data science for efficient computer-based solutions.

THEORY:

Hamming Code:

The most common types of error-correcting codes used in RAM are based on the codes devised by R. W. Hamming. In the Hamming code, k parity bits are added to an n-bit data word, forming a new word of n Ƨ k bits. The bit positions are numbered in sequence from 1 to n Ƨ k. Those positions numbered with powers of two are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length.

Before giving the general characteristics of the Hamming code, we will illustrate its operation with a data word of eight bits. Consider, for example, the 8-bit data word 11000100. We include four parity bits with this word and arrange the 12 bits as follows:

Bit position

1	2	3	4	5	6	7	8	9	10	11	12
P1	P2	1	P4	1	0	0	P8	0	1	0	0

The 4 parity bits P 1 through P 8 are in positions 1, 2, 4, and 8, respectively. The 8 bits of the data word are in the remaining positions. Each parity bit is calculated as follows:

$P\ 1 = \text{XOR of bits } (3, 5, 7, 9, 11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$
$P\ 2 = \text{XOR of bits } (3, 6, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$
$P\ 4 = \text{XOR of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$
$P\ 8 = \text{XOR of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$

Recall that the exclusive-OR operation performs the odd function. It is equal to 1 for an odd number of 1’s among the variables and to 0 for an even number of 1’s.

Thus, each parity bit is set so that the total number of 1’s in the checked positions, including the parity bit, is always even.

The 8-bit data word is written into the memory together with the 4 parity bits as a 12-bit composite word. Substituting the 4 parity bits in their proper positions, we obtain the 12-bit composite word written into memory:

Bit position

1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	1	1	0	0	1	0	1	0	0

When the 12 bits are read from memory, they are checked again for errors. The parity of the word is checked over the same groups of bits, including their parity bits. The four check bits are evaluated as follows:

C 1 = XOR of bits (1, 3, 5, 7, 9, 11)

C 2 = XOR of bits (2, 3, 6, 7, 10,11)

C 4 = XOR of bits (4, 5, 6, 7, 12)

C 8 = XOR of bits (8, 9, 10, 11, 12)

A 0 check bit designates an even parity over the checked bits, and a 1 designates an odd parity. Since the bits were written with even parity, the result, C = C8 C4 C2 C1 = 0000, indicates that no error has occurred. However, if $C \neq 0$, the 4-bit binary number formed by the check bits gives the position of the erroneous bit if only a single bit is in error.

Thus, for no error, we have C = 0000; with an error in bit 1, we obtain C = 0001;and with an error in bit 5, we get C = 0101. Hence, when C is not equal to 0, the decimal value of C gives the position of the bit in error. The error can then be corrected by complementing the corresponding bit. Note that an error can occur in the data or in one of the parity bits.

Algorithm

The following general algorithm generates a single-error correcting (SEC) code for any number of bits.

- Number the bits starting from 1: bit 1, 2, 3, 4, 5, etc.
- Write the bit numbers in binary: 1, 10, 11,100, 101, etc.
- All bit positions that are powers of two (have only one 1 bit in the binary form of their position) are parity bits: 1, 2, 4, 8, etc. (1, 10, 100, 1000)
- All other bit positions, with two or more 1 bits in the binary form of their position, are data bits.

- Each data bit is included in a unique set of 2 or more parity bits, as determined by the binary form of its bit position.Parity bit 1 covers all bit positions which have the least significant bit set: bit 1 (the parity bit itself), 3, 5, 7, 9, etc.
- Parity bit 2 covers all bit positions which have the second least significant bit set: bit 2 (the parity bit itself), 3, 6, 7, 10, 11, etc.
- Parity bit 4 covers all bit positions which have the third least significant bit set: bits 4–7, 12–15, 20–23, etc.
- Parity bit 8 covers all bit positions which have the fourth least significant bit set: bits 8–15, 24–31, 40–47, etc.
- In general each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.

The form of the parity is irrelevant. Even parity is simpler from the perspective of theoretical mathematics, but there is no difference in practice.

Cyclic redundancy check(CRC) :

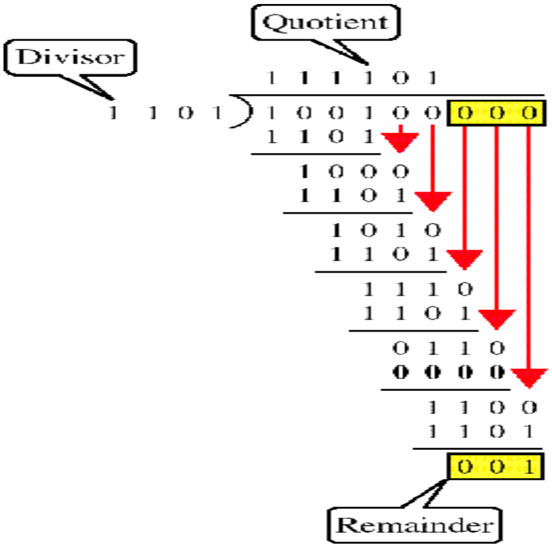
A cyclic redundancy check (CRC) is an error detecting codes commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short *check value* attached, based on the remainder of a polynomial division of their contents. On retrieval, the calculation is repeated and, in the event the check values do not match, corrective action can be taken against data corruption.

Application

A CRC-enabled device calculates a short, fixed-length binary sequence, known as the check value or CRC, for each block of data to be sent or stored and appends it to the data, forming a codeword. When a codeword is received or read, the device either compares its check value with one freshly calculated from the data block, or equivalently, performs a CRC on the whole codeword and compares the resulting check value with an expected *residue* constant.

If the CRC check values do not match, then the block contains a data error.

The device may take corrective action, such as rereading the block or requesting that it be sent again. Otherwise, the data is assumed to be error-free (though, with some small probability, it may contain undetected errors; this is the fundamental nature of error-checking). Compute an *n*-bit binary CRC, line the bits representing the input in a row, and position the (*n* + 1)-bit pattern representing the CRC's divisor (called a "polynomial") underneath the left-hand end of the row.



In this example, we shall encode 14 bits of message with a 3-bit CRC, with a polynomial $x^3 + x + 1$. The polynomial is written in binary as the coefficients; a 3rd-order polynomial has 4 coefficients ($1x^3 + 0x^2 + 1x + 1$). In this case, the coefficients are 1, 0, 1 and 1. The result of the calculation is 3 bits long.

Mathematical Model for CRC

S: {I, O, F, DD, NDD, shared

memory} No shared memory

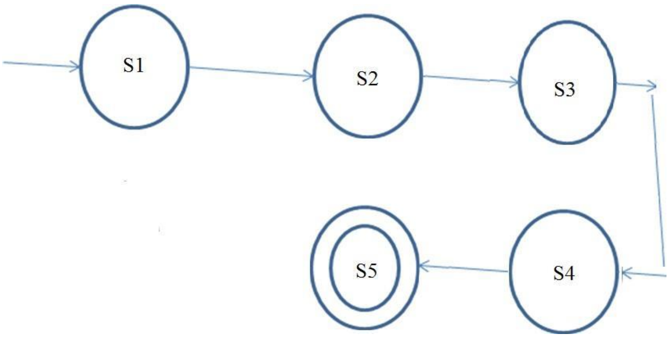
I= {Dataword,Generator/Divisor}

Where O= Codeword F: Functions used in Program

parity_calculate(),CRC(),XOR();

DD Deterministic Data: set of input i.e. dataword , divisor

NDD Non Deterministic Data: codeword



Where,

S1: Start state

S2: Accept Dataword,Assign

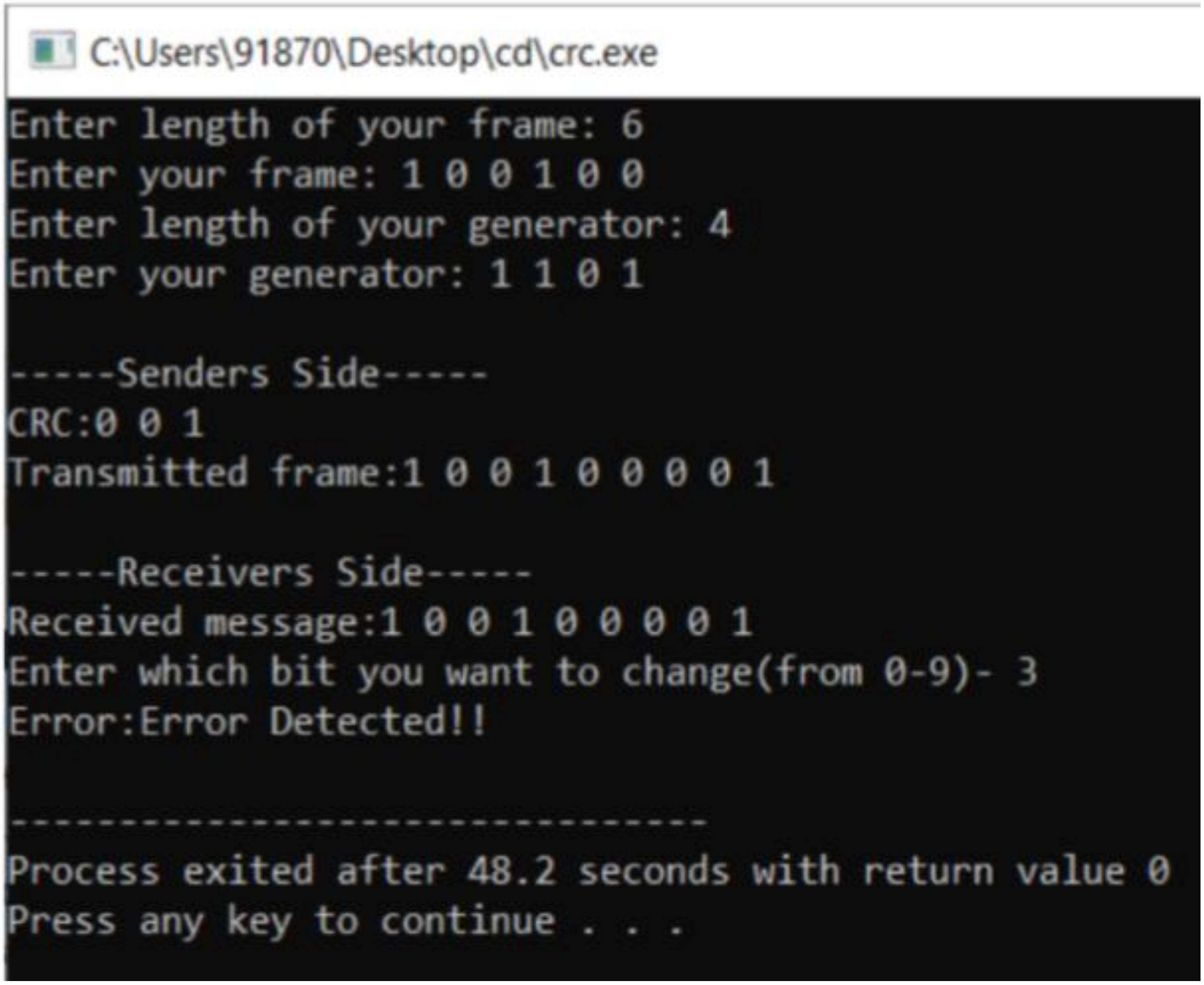
Polynomial/generator S3: XOR dataword with

polynomial/generator

S4: get CRC remainder add it in dataword

S5: End state

OUTPUT:



```
Enter length of your frame: 6
Enter your frame: 1 0 0 1 0 0
Enter length of your generator: 4
Enter your generator: 1 1 0 1

-----Senders Side-----
CRC:0 0 1
Transmitted frame:1 0 0 1 0 0 0 0 1

-----Receivers Side-----
Received message:1 0 0 1 0 0 0 0 1
Enter which bit you want to change(from 0-9)- 3
Error:Error Detected!!

--------------------------------
Process exited after 48.2 seconds with return value 0
Press any key to continue . . .
```

CONCLUSION:

Outcome of the experiment is

1. Students are able to implement Hamming codes and CRC

FAQs:

1. What is Hamming Distance?

Ans: The Hamming Distance is a number used to denote the difference between two binary strings. Hamming's formulas allow computers to detect and correct error on their own.

2. What is even and odd parity?

Ans: Odd parity: The number of 1-bit in the data word must add up to an odd number Even parity: the number of 1-bit in the data word must add up to an even number.

3. Give Even parity Example.

Ans: A single bit is appended to each data chunk makes the number of 1 bits even/oddample: even parity 1000000(1).

4. What is the minimum hamming distance?

Ans: It is the min distance between all possible pairs of codewords. The number of errors that can be detected will always be one less than the min hamming distance.

5. What is error control?

Ans: Error control involves retransmission of the lost, damaged, or corrupted frame.

Group:A

Assignment No: 3

AIM: Write a program to demonstrate subnetting and find the subnet masks.

PROBLEM STATEMENT: Lab Assignment on Unit IV: (Use JAVA/PYTHON).Write a program to demonstrate subletting and find the subnet masks.

PREREQUISITES: Routing IP Addressing & Subnetting Concept.

COURSE OBJECTIVES: To understand subnetting and pinging the host.

COURSE OUTCOMES:CO2 : Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols.

COURSE OUTCOME MAPPED: CO3: Demonstrate the subnet formation with IP allocation mechanism and apply various routing algorithms

PROGRAM OUTCOME MAPPED: PO1,PO2,PO3,PO4,PO12

PROGRAM SPECIFIC OUTCOME MAPPED: PSO1,PSO2

THEORY

IP Address

- A Unique, 32-bit address used by computers to communicate over a computer networkIP address structure consists of two addresses, Network and Host Figure shows IP address classes

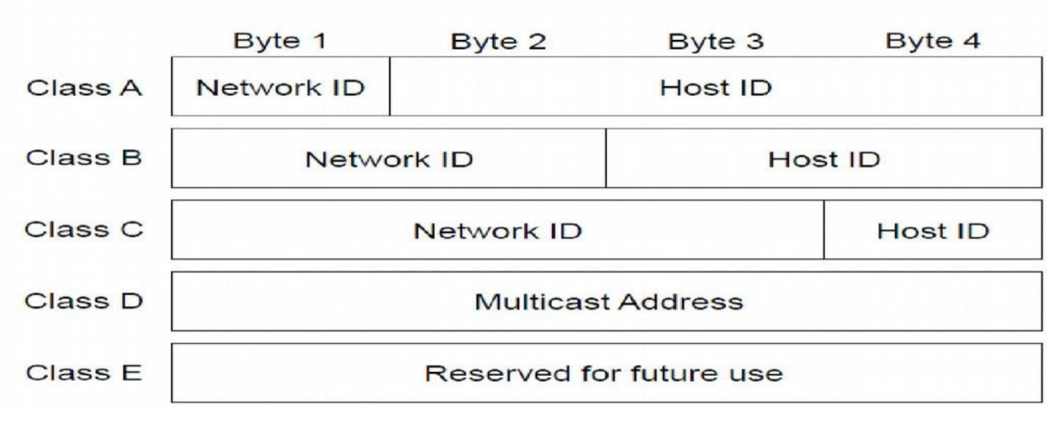


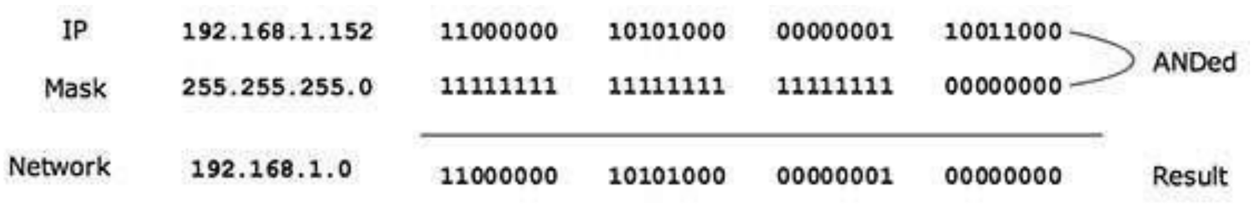
Figure: IP address class

Subnet Mask

The 32-bit IP address contains information about the host and its network. It is very necessary to distinguish both. For this, routers use Subnet Mask, which is as long as the size of the network address in the IP address. Subnet Mask is also 32 bits long. If the IP address in binary is ANDed with its Subnet Mask, the result yields the Network address.

For example, say the IP Address is

192.168.1.152 and the Subnet Mask is 255.255.255.0 then:



IP	192.168.1.152	11000000	10101000	00000001	10011000	ANDed
Mask	255.255.255.0	11111111	11111111	11111111	00000000	
Network	192.168.1.0	11000000	10101000	00000001	00000000	Result

Subnetting

- Division of a network into subnets
    - For example, division of a Class B address into several Class C addresses
- Some of the host IDs are used for creating subnet IDs

Need for Subnetting

- Classes A and B have a large number of hosts corresponding to each network ID
- It may be desirable to subdivide the hosts in Class C subnetting.
- Often, there is a limitation on the number of hosts that could be hosted on a single
- network segment.
- The limitation may be imposed by concerns related to the management of hardware.
- Smaller broadcast domains are more efficient and easy to manage.
- Use parts of the host IDs for subnetting purpose
- A subnet mask is used to facilitate the flow of traffic between the different subnets and the outside network (hops).
- A hop is the distance a data packet travels from one node to the other

Knowing How to Calculate Subnets

- To determine the number of subnets & hosts per subnet available for any of the available subnet masks, 2 simple formulas to calculate these numbers:
- Number of Subnets=(2^n)
- Number of Host per Subnets=(2^{h-2})

Example: Consider the network id 192.168.8.0 or such relevant IP and create five subnets namely A, B, C, D. Assign the subnet mask. Determine the number of bits used for Subnet and host.

Given network id ->192.168.4.0

The IP address 192.168.4.0 belongs to Class C

Network ID-192.168.4-(first 3 bytes of dotted decimal Number)

Host ID-0 (last one byte of dotted decimal Number)

Default subnet mask of class C-255.255.255.0

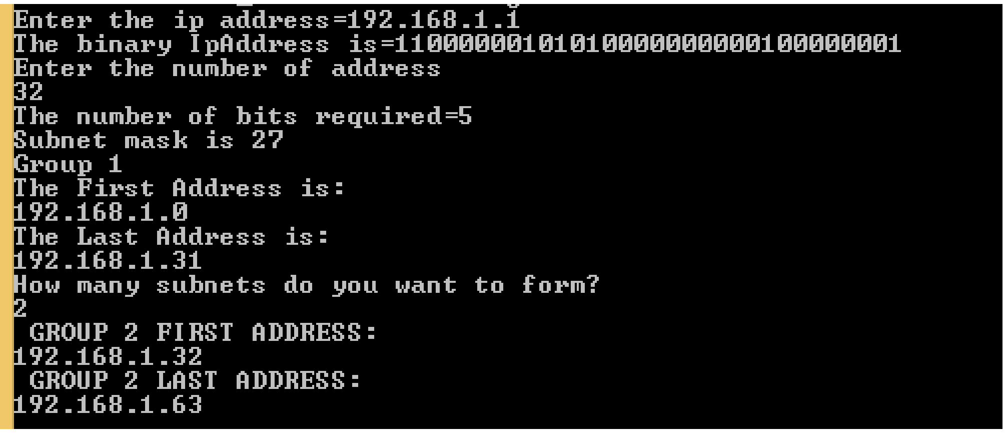
To create four subnets namely A, B, C, D we consider Host ID bits. For four subnets we require 2 bits out of of 8 host id bits.

0 0 0 0 0 0 0 0

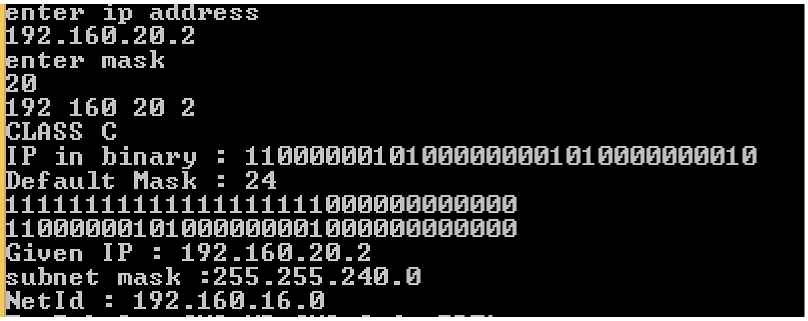
So with 2 bits of Subnet ID 2^2=4 Subnets can be formed A,B,C &D

- Each subnet holding 2^6 -2=62 hosts
- Mask address -255.255.255.192(11111111.11111111.11111111.11000000)

OUTPUT:



```
Enter the ip address=192.168.1.1
The binary IpAddress is=11000000101010000000000100000001
Enter the number of address
32
The number of bits required=5
Subnet mask is 27
Group 1
The First Address is:
192.168.1.0
The Last Address is:
192.168.1.31
How many subnets do you want to form?
2
 GROUP 2 FIRST ADDRESS:
192.168.1.32
 GROUP 2 LAST ADDRESS:
192.168.1.63
```



```
enter ip address
192.160.20.2
enter mask
20
192 160 20 2
CLASS C
IP in binary : 11000000101000000001010000000010
Default Mask : 24
11111111111111111111000000000000
11000000101000000001000000000000
Given IP : 192.160.20.2
subnet mask :255.255.240.0
NetId : 192.160.16.0
```

CONCLUSION: Outcome of the experiment is students are able to divide the network using implement subnetting.

FAQs:

1.What is the default mask address of class A, B ,C

Ans: Class A-255.0.0.0,

Class B- 255.255.0.0,ClassC-255.255.255.0

2.What is a Mask Address?

Ans: It is called a subnet mask because it is used to identify the network address of an IP address by performing a bitwise AND operation on the netmask.

3.Write the range of addresses for classful addressing Scheme

Ans:

Class A	1.0.0.1 to 126.255.255.254
Class B	128.1.0.1 to 191.255.255.254
Class C	192.0.1.1 to 223.255.254.254
Class D	224.0.0.0 to 239.255.255.255

4.What is its use of subnetting?

Ans: It helps in achieving security, reduced routing table length and hence quicker processing of table entries.

5.What are special IP addresses?

Ans: A network address is an address where all host bits in the IP address are set to zero (0).

A broadcast address is an address where all host bits in the IP address are set to one (1).

The 127.0.0.0 class 'A' subnet is used for special local addresses, most commonly the loopback address 127.0.0.1

Group: A

Assignment No. 4

AIM: Write a program for DNS lookup. Given an IP address input, it should return URL and vice-versa.

PROBLEM STATEMENT: Lab Assignment on Unit IV: (Use JAVA/PYTHON)
Write a program for DNS lookup.

PREREQUISITES: Network Protocols, Layered Architecture.

COURSE OBJECTIVES: To study and implement DNS lookup in a network.

COURSE OUTCOME MAPPED:

CO4: Develop Client-Server architectures and prototypes

CO5: Implement web applications and services using application layer protocols

CO6: Use network security services and mechanisms

PROGRAM SPECIFIC OUTCOME MAPPED: PSO1, PSO2

PROGRAM OUTCOME MAPPED: PO1, PO3, PO4, PO5

COURSE OUTCOMES:

CO2 : Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

THEORY:

DNS: Domain name system

A domain name is a meaningful name that identifies an internet address. DNS is a system where these domain names are located.

DNS services

- hostname to IP address translation v host aliasing
- canonical, alias names v mail server aliasing v load distribution
- replicated Web servers: many IP addresses correspond to one name

DNS: a distributed, hierarchical database DNS: Root name servers

contacted by local name server that can not resolve name root name server: a. contacts authoritative name server if name mapping not known

a. gets mapping

b. returns mapping to local name server

Top-level domain (TLD) servers:

§ responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp

§

Network Solutions maintains servers for .com TLD § Educause for .edu TLD

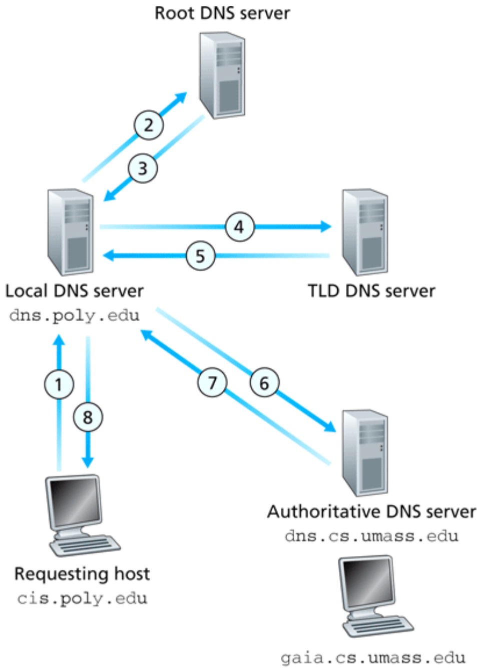


Figure 2.21 ♦ Interaction of the various DNS servers

A**uthoritative DNS servers:** § organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts § can be maintained by organization or service provider

Local DNS name server v does not strictly belong to hierarchy v each ISP (residential ISP, company, university) has one § also called "default name server" v when host makes DNS query, query is sent to its local DNS server § has local cache of recent name-to-address translation pairs (but may be out of date!) § acts as proxy, forwards query into hierarchy

DNS name resolution example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

- iterated query: v contacted server replies with name of server to contact v "I don't know this name, but ask this server

- recursive query: v puts burden of name resolution on contacted name server v heavy load at upper levels of hierarchy?

DNS: caching, **updating records**

once (any) name server learns mapping, it caches mapping

- cache entries timeout (disappear) after some time (TTL)
- TLD servers typically cached in local name servers • thus root name servers not often visited
- Cached entries may be out-of-date (best effort name-to-address translation!)
- If name host changes IP address, may not be known Internet-wide until all TTLs expire

DNS records

DNS: distributed db storing resource records (RR)

RR format: (name, value,
type, ttl) type=NS

§ name is domain (e.g., foo.com)
§ value is hostname of authoritative name server for this domain

type=A

§ name is hostname
§ value is IP address

type=CNAME

§ name is alias name for some “canonical” (the real) name
§ www.ibm.com is really servereast.backup2.ibm.com
§ value is canonical name type=MX
§ value is name of mail server associated with name

ALGORITHM/ PROGRAM:

```
import java.net.*;
import java.util.*;

public class IPDemo {
        public static void main(String[] args) {
                String host;
                Scanner ch = new Scanner(System.in);
                System.out.print("1.Enter Host Name \n2.Enter IP address \nChoice=");
                int choice = ch.nextInt();
                if(choice==1) {
                        Scanner input = new Scanner(System.in);
                        System.out.print("\n Enter host name: ");
                        host = input.nextLine();
                        try {
```

```
                InetAddress address = InetAddress.getByName(host);
                System.out.println("IP address: " + address.getHostAddress());
                System.out.println("Hostname : " + address.getHostName());
                System.out.println("Hostname and IP address: " +
                address.toString());
            }
            catch (UnknownHostException ex) {
                System.out.println("Could not find " + host);
            }
        }
        else {
            Scanner input = new Scanner(System.in);
            System.out.print("\n Enter IP address: ");
            host = input.nextLine();
            try {
                InetAddress address = InetAddress.getByName(host);
                System.out.println("Host name : " + address.getHostName());
                System.out.println("IP address: " + address.getHostAddress());
                System.out.println("Hostname and IP address: " +
            address.toString());

            }
            catch (UnknownHostException ex) {
             System.out.println("Could not find " + host);
            }
        }

    }
}
```

OUTPUT:

```
1.Enter Host Name
2.Enter IP address
Choice=1

Enter host name: www.google.com
IP address: 172.217.160.196
Host name : www.google.com
Hostname and IP address: www.google.com/172.217.160.196

1.Enter Host Name
2.Enter IP address
Choice=2

 Enter IP address: 8.8.8.8
```

Host name : dns.google
IP address: 8.8.8.8
Hostname and IP address: dns.google/8.8.8.8

CONCLUSION:

The outcome of the assignment is the students will be able to understand and implement DNS lookup

FAQ’s:

1. **What is the port no of dns ?**
   Ans: 53.
2. **What is a Resource Record?**
   Ans: It is a record that provides information about the resources available in the N/W infrastructure.
3. **What is a Zone?**
   Ans: Zone is a subtree of DNS databases.
4. WhatareaForwardandReverseLookup?

   Ans: **Forward Lookup:** When a name query is send to the DNS server against IP addresses, it is generally said to be a forward lookup.
   Reverse Lookup: DNS also provides a reverse lookup process, enabling clients to use a known IP address during a name query and look up a computer name based on its address.

Group: A

Assignment No. 5

AIM: To study TCP socket programming in C/C++.

PROBLEM STATEMENT: Lab Assignment on Unit V: (Mandatory Assignment) (Use C/C++) Write a program using TCP socket for wired network for following

a. Say Hello to Each other (For all students)

b. File transfer (For all students) c. Calculator (Arithmetic) d. Calculator (Trigonometry) Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode

PREREQUISITES:

1. Knowledge of layered architecture and its protocols
2. Knowledge of socket programming

COURSE OBJECTIVES:

CO2 : Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

COURSE OUTCOMES:

Ability to perform client-server programming using TCP socket.

COURSE OUTCOME MAPPED:

CO4: Develop Client-Server architectures and prototypes

PROGRAM OUTCOME MAPPED: PO1, PO2, PO3, PO5, PO12

PROGRAM SPECIFIC OUTCOME MAPPED: PSO1, PSO2

THEORY:

Introduction to Sockets Programming :

A socket is the mechanism that most popular operating systems provide to give programs access to the network. It allows messages to be sent and received between applications (unrelated processes) on different networked machines.

Figure: **Socket Primitives**

There are a few steps involved in using sockets:

1. Create the socket
2. Identify the socket
3. On the server, wait for an incoming connection
4. On the client, connect to the server's socket
5. Send and receive messages
6. Close the socket

Mathematical Model:

S: {I, O, F, DD, NDD, shared

memory} No shared memory

I= {Data/msg } Where O ={Data/msg} F: Functions used in Program

read(sockid, recvBuf, bufLen, flags)

write(sockid, msg, msgLen, flags)

DD Deterministic Data: generated packet

NDD Non Deterministic Data: delivered packets

PROGRAM:

```
#include <arpa/inet.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <unistd.h>
#define PORT 8080

int main(int argc, char const* argv[])
{
     int sock = 0, valread, client_fd;
    struct sockaddr_in serv_addr;
    char* hello = "Hello from client";
    char buffer[1024] = { 0 };
     if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
           cout<<("\n Socket creation error \n");
            return -1;
```

```
       }

      serv_addr.sin_family = AF_INET;
      serv_addr.sin_port = htons(PORT);

       // Convert IPv4 and IPv6 addresses from text to binary
     // form
       if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr)
              <= 0) {
             cout<<(
                      "\nInvalid address/ Address not supported \n");
              return -1;
     }

      if ((client_fd
             = connect(sock, (struct sockaddr*)&serv_addr,
                              sizeof(serv_addr)))
              < 0) {
               cout<<("\nConnection Failed \n");
              return -1;
      }
       send(sock, hello, strlen(hello), 0);
      cout<<("Hello message sent\n");
      valread = read(sock, buffer, 1024);
      cout<<("%s\n", buffer);

    // closing the connected socket
     close(client_fd);
      return 0;
}
```

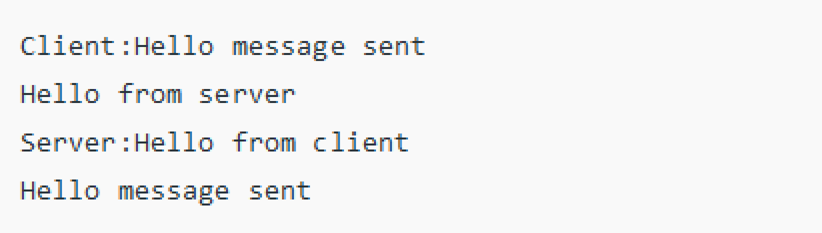
Code for server:

```
// Server side C/C++ program to demonstrate Socket
// programming
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <unistd.h>
#define PORT 8080
int main(int argc, char const* argv[])
{
     int server_fd, new_socket, valread;
       struct sockaddr_in address;
      int opt = 1;
```

```
 int addrlen = sizeof(address);
  char buffer[1024] = { 0 };
  char* hello = "Hello from server";

// Creating socket file descriptor
if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
          perror("socket failed");
          exit(EXIT_FAILURE);
 }

 // Forcefully attaching socket to port 8080
 if (setsockopt(server_fd, SOL_SOCKET,
                         SO_REUSEADDR | SO_REUSEPORT, &opt,
                         sizeof(opt))) {
        perror("setsockopt");
         exit(EXIT_FAILURE);
  }
  address.sin_family = AF_INET;
  address.sin_addr.s_addr = INADDR_ANY;
  address.sin_port = htons(PORT);

 // Forcefully attaching socket to port 8080
 if (bind(server_fd, (struct sockaddr*)&address,
                 sizeof(address))
         < 0) {
          perror("bind failed");
          exit(EXIT_FAILURE);
  }
  if (listen(server_fd, 3) < 0) {
         perror("listen");
          exit(EXIT_FAILURE);
 }
  if ((new_socket
         = accept(server_fd, (struct sockaddr*)&address,
                          (socklen_t*)&addrlen))
          < 0) {
         perror("accept");
         exit(EXIT_FAILURE);
 }
  valread = read(new_socket, buffer, 1024);
  cout<<("%s\n", buffer);
  send(new_socket, hello, strlen(hello), 0);
  cout<<("Hello message sent\n");

  // closing the connected socket
  close(new_socket);
```

```
        // closing the listening socket
        shutdown(server_fd, SHUT_RDWR);
        return 0;
    }
```

OUTPUT:



```
Client:Hello message sent
Hello from server
Server:Hello from client
Hello message sent
```

CONCLUSION:

Outcome of the experiment is students are able to

implement chatting application using TCP socket programming

FAQS:

1. What are the steps involved in creating & using sockets for TCP connection?

Ans: There are a few steps involved in using sockets:

- Create the socket
- Identify the socket
- On the server, wait for an incoming connection
- On the client, connect to the server's socket
- Send and receive messages
- Close the socket

2. What are the types of Socket?

Ans: Stream Socket, Datagram Socket, Raw Socket.

3. What is Port Number?

Ans: A port number is a way to identify a specific process to which an Internet or other network message is to be forwarded when it arrives at a server.

Group A

Assignment No:6

TITLE: To implement a UDP socket program in C/C++.

PROBLEM STATEMENT: Lab Assignment on Unit V: (Mandatory Assignment)
Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode. Write TFTP program using socket programming for UDP using C++.

PREREQUISITES: Layered architecture and functionalities

COURSE OBJECTIVES: To understand and implement file transfer using UDP protocol

COURSE OUTCOME:

CO2: Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

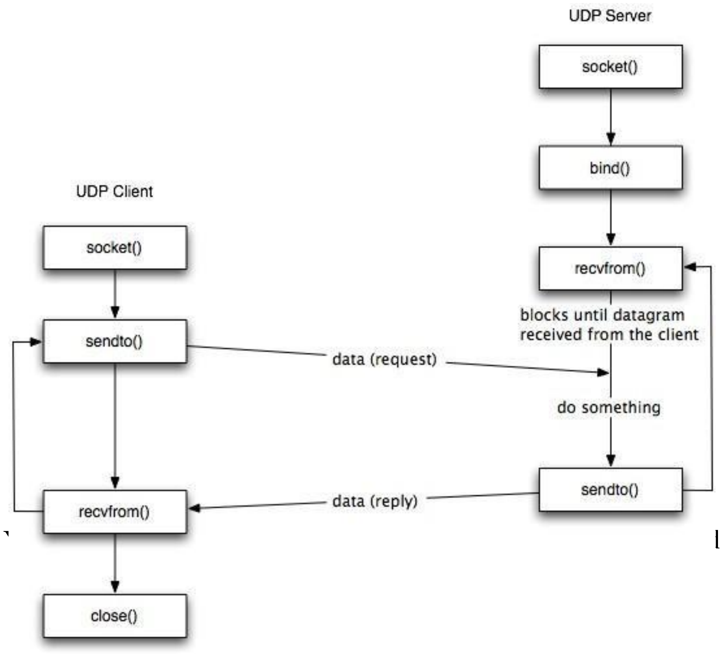
CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

COURSE OUTCOME MAPPED: CO4: Develop Client-Server architectures and prototypes

PROGRAM OUTCOME MAPPED: PO1,PO2,PO3,PO5,PO12

PROGRAM SPECIFIC OUTCOME MAPPED: PSO1,PSO2

THEORY:



Mathematical Model

S: {I, O, F, DD, NDD, shared

memory} No shared memory

I= {Data/msg }

Where O ={Data/msg} F: Functions used in Program Sendto(sockid, msg, msgLen, flags, &foreignAddr, addrlen) recvfrom(sockid, recvBuf, bufLen, flags,&clientAddr, addrlen) DD Deterministic Data: generated packet
NDD Non Deterministic Data: delivered

packets

ALGORITHM/PROGRAM:

Filename: UDPServer.c++

```
// Server side implementation of UDP client-server model
#include <bits/stdc++.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define PORT  8080
#define MAXLINE 1024

// Driver code
int main() {
        int sockfd;
        char buffer[MAXLINE];
        const char *hello = "Hello from server";
        struct sockaddr_in servaddr, cliaddr;

        // Creating socket file descriptor
        if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
                perror("socket creation failed");
                exit(EXIT_FAILURE);
        }
```

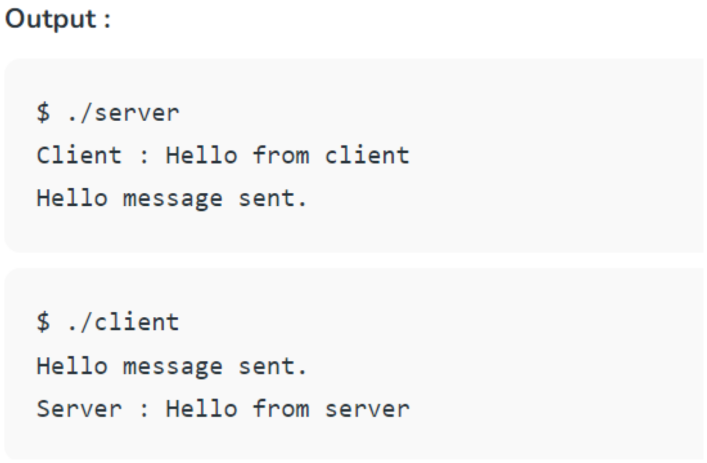
```
    memset(&servaddr, 0, sizeof(servaddr));
    memset(&cliaddr, 0, sizeof(cliaddr));

    // Filling server information
    servaddr.sin_family = AF_INET; // IPv4
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(PORT);

    // Bind the socket with the server address
    if ( bind(sockfd, (const struct sockaddr *)&servaddr,
            sizeof(servaddr)) < 0 )
    {
        perror("bind failed");
        exit(EXIT_FAILURE);
    }

    socklen_t len;
int n;

    len = sizeof(cliaddr); //len is value/result

    n = recvfrom(sockfd, (char *)buffer, MAXLINE,
                MSG_WAITALL, ( struct sockaddr *) &cliaddr,
                &len);
    buffer[n] = '\0';
    printf("Client : %s\n", buffer);
    sendto(sockfd, (const char *)hello, strlen(hello),
        MSG_CONFIRM, (const struct sockaddr *) &cliaddr,
            len);
    std::cout<<"Hello message sent."<<std::endl;

    return 0;
}
```

Filename: UDPClient.c++

```
// Client side implementation of UDP client-server model
#include <bits/stdc++.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
```

```
#include <netinet/in.h>

#define PORT     8080
#define MAXLINE 1024


// Driver code
int main() {
    int sockfd;
    char buffer[MAXLINE];
    const char *hello = "Hello from client";
    struct sockaddr_in   servaddr;

    // Creating socket file descriptor
    if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));

    // Filling server information
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = INADDR_ANY;

    int n;
    socklen_t len;

    sendto(sockfd, (const char *)hello, strlen(hello),
        MSG_CONFIRM, (const struct sockaddr *) &servaddr,
            sizeof(servaddr));
    std::cout<<"Hello message sent."<<std::endl;

    n = recvfrom(sockfd, (char *)buffer, MAXLINE,
                MSG_WAITALL, (struct sockaddr *) &servaddr,
                &len);
    buffer[n] = '\0';
    std::cout<<"Server :"<<buffer<<std::endl;
    close(sockfd);
    return 0;
}
```

OUTPUT:



Output :

```
$ ./server
Client : Hello from client
Hello message sent.
```

```
$ ./client
Hello message sent.
Server : Hello from server
```

CONCLUSION:

Outcome of the experiment is students are able to implement UDP Socket.

FAQ's

1. **What are the characteristics of UDP protocol**

Ans: UDP provides a minimal, unreliable, best-effort, message-passing transport to applications and upper-layer protocols.

2. **Which applications prefer UDP and why?**

Ans: Multimedia applications ,DNS,DHCP prefer UDP. Timely delivery is the major constraint of these applications

3. What is the size of TCP and UDP packets?
Ans: TCP: 20 bytes minimum UDP: 8 Bytes

Group A

Assignment No. : 7

AIM : Packet Analyzer

PROBLEM STATEMENT: Write a program to analyze following packet formats captured through Wireshark for wired network. 1. Ethernet 2. IP 3.TCP 4. UDP.

PREREQUISITES: Understanding of Wireshark and different protocols of Transport and network layer.

COURSE OBJECTIVES: Configure the computing node with understanding of protocols and technologies

COURSE OUTCOMES:

CO2: Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

COURSE OUTCOME MAPPED: CO4, CO5

CO4: Develop Client-Server architectures and prototypes

CO5: Implement web applications and services using application layer protocols

PROGRAM OUTCOME MAPPED: PO1, PO2, PO4, PO5, PO12

PROGRAM SPECIFIC OUTCOME MAPPED: PSO1, PSO2

THEORY:

Packet sniffer \ Packet analyzer:

A packet analyzer (also known as a network analyzer, protocol analyzer or packet sniffer or for particular types of networks, an Ethernet sniffer or wireless sniffer) is a computer program or a piece of computer hardware that can intercept and log traffic passing over a digital network or part of a network. As data streams own across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet, and analyzes its content according to the appropriate RFC or other specifications.

Different types of packet:

1. TCP:

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP. TCP provides reliable, ordered and error-checked delivery (or notification of failure to deliver) of a stream of octets between programs running on computers connected to a local area network, intranet or the public Internet. It resides at the transport

2. UDP:

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol Suite. UDP uses a simple connectionless transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. There is no guarantee of delivery, ordering, or duplicate protection

3.ICMP:The Internet Control Message Protocol (ICMP) is one of the main protocols of the Internet Protocol Suite. It is used by network devices, like routers, to send error messages indicating, for example, that a requested service is not available or that a host or router could not be reached.

4.IGMP:

The Internet Group Management Protocol (IGMP) is a communications protocol used by hosts and adjacent routers on IP networks to establish multicast group memberships. IGMP is an integral part of IP multicast. IGMP can be used for one-to-many networking applications such as online streaming video and gaming, and allows more efficient use of resources when supporting these types of applications. IGMP messages are carried in bare IP packets with IP protocol.

Mathematical Model

S: {I, O, F, DD, NDD, shared memory}

No shared memory

I= { packet log file captured from wireshark }

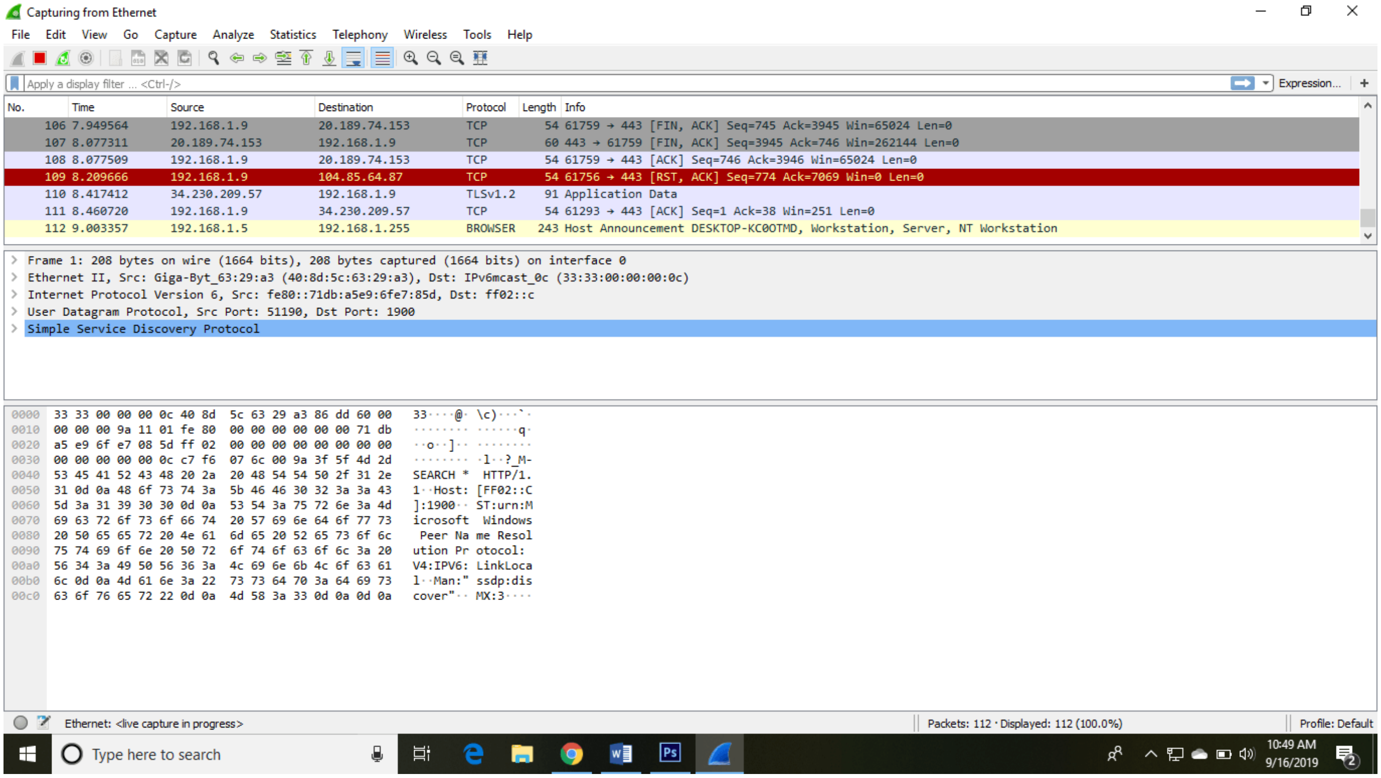
Where O ={Count of Traced packet } F: Functions used in Program

recvfrom(sock_raw , buffer , 65536 , 0 , &saddr , &saddr_size)
ProcessPacket(buffer , data_size);

DD Deterministic Data: input file

NDD Non Deterministic Data: packet count

OUTPUT:


Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Type here to search
10:49 AM
9/16/2019

```
No.   Time       Source          Destination     Protocol  Length Info
  106 7.949564   192.168.1.9     20.189.74.153   TCP        54 61759 → 443 [FIN, ACK] Seq=745 Ack=3945 Win=65024 Len=0
  107 8.077311   20.189.74.153   192.168.1.9     TCP        60 443 → 61759 [FIN, ACK] Seq=3945 Ack=746 Win=262144 Len=0
  108 8.077509   192.168.1.9     20.189.74.153   TCP        54 61759 → 443 [ACK] Seq=746 Ack=3946 Win=65024 Len=0
  109 8.209666   192.168.1.9     104.85.64.87    TCP        54 61756 → 443 [RST, ACK] Seq=774 Ack=7069 Win=0 Len=0
  110 8.417412   34.230.209.57   192.168.1.9     TLSv1.2    91 Application Data
  111 8.460720   192.168.1.9     34.230.209.57   TCP        54 61293 → 443 [ACK] Seq=1 Ack=38 Win=251 Len=0
  112 9.003357   192.168.1.5     192.168.1.255   BROWSER   243 Host Announcement DESKTOP-KC0OTMD, Workstation, Server, NT Workstation
```

```
> Frame 1: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface 0
> Ethernet II, Src: Giga-Byt_63:29:a3 (40:8d:5c:63:29:a3), Dst: IPv6mcast_0c (33:33:00:00:00:0c)
> Internet Protocol Version 6, Src: fe80::71db:a5e9:6fe7:85d, Dst: ff02::c
> User Datagram Protocol, Src Port: 51190, Dst Port: 1900
> Simple Service Discovery Protocol
```

```
0000  33 33 00 00 00 0c 40 8d  5c 63 29 a3 86 dd 60 00   33····@· \c)···`·
0010  00 00 00 9a 11 01 fe 80  00 00 00 00 00 00 71 db   ········ ······q·
0020  a5 e9 6f e7 08 5d ff 02  00 00 00 00 00 00 00 00   ··o··]·· ········
0030  00 00 00 00 00 0c c7 f6  07 6c 00 9a 3f 5f 4d 2d   ········ ·l··?_M-
0040  53 45 41 52 43 48 20 2a  20 48 54 54 50 2f 31 2e   SEARCH *  HTTP/1.
0050  31 0d 0a 48 6f 73 74 3a  5b 46 46 30 32 3a 3a 43   1··Host: [FF02::C
0060  5d 3a 31 39 30 30 0d 0a  53 54 3a 75 72 6e 3a 4d   ]:1900·· ST:urn:M
0070  69 63 72 6f 73 6f 66 74  20 57 69 6e 64 6f 77 73   icrosoft  Windows
0080  20 50 65 65 72 20 4e 61  6d 65 20 52 65 73 6f 6c    Peer Na me Resol
0090  75 74 69 6f 6e 20 50 72  6f 74 6f 63 6f 6c 3a 20   ution Pr otocol:
00a0  56 34 3a 49 50 56 36 3a  4c 69 6e 6b 4c 6f 63 61   V4:IPV6: LinkLoca
00b0  6c 0d 0a 4d 61 6e 3a 22  73 73 64 70 3a 64 69 73   l··Man:" ssdp:dis
00c0  63 6f 76 65 72 22 0d 0a  4d 58 3a 33 0d 0a 0d 0a   cover"·· MX:3····
```

Ethernet: <live capture in progress>
Packets: 112 · Displayed: 112 (100.0%)
Profile: Default

CONCLUSION:

Outcome of the experiment is students are able to
analyze formats captured through Wireshark for wired network. 1. Ethernet 2. IP 3.TCP 4. UDP

Group:A

ASSIGNMENT NO.8

TITLE: Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol

PROBLEM STATEMENT: Lab Assignment on Unit II: (Use JAVA/PYTHON)

Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode

PREREQUISITES: Protocols, layered Architecture

COURSE OBJECTIVES: To study and implement Go back N and Selective Repeat

COURSE OUTCOME MAPPED: CO2

PROGRAM OUTCOME MAPPED: PO1, PO2, PO4, PO5, PO12

PROGRAM SPECIFIC OUTCOME MAPPED: PSO1, PSO2

THEORY

The most important functions of the Data Link layer to satisfy the above requirements are error control and flow control.

Flow Control is a technique so that transmitter and receiver with different speed characteristics can communicate with each other. Flow control ensures that a transmitting station, such as a server with higher processing capability, does not overwhelm a receiving station, such as a desktop system, with lesser processing capability. This is where there is an orderly flow of transmitted data between the source and the destination.

Error Control involves both error detection and error correction. It is necessary because errors are inevitable in data communication, in spite of the use of better equipment and reliable transmission media based on the current technology

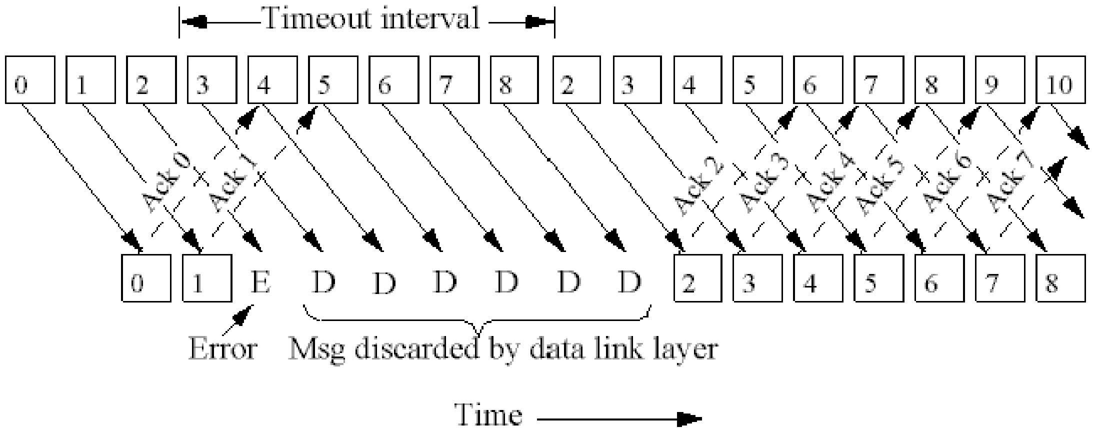
There are two methods developed for flow control namely Stop-and-wait and Sliding-window.

Go-back-N ARQ: The most popular ARQ protocol is the go-back-N ARQ, where the sender sends the frames continuously without waiting for acknowledgement. That is why it is also called as continuous ARQ. As the receiver receives the frames, it keeps on sending ACKs or a NACK, in case a frame is incorrectly received. When the sender receives a NACK, it retransmits the frame in error plus all the succeeding frames. Hence, the name of the protocol is go-back-N ARQ. If a frame is lost, the receiver sends NAK after receiving the next frame.

In case there is long delay before sending the NAK, the sender will resend the lost frame after its timer times out. If the ACK frame sent by the receiver is lost, the sender resends the frames after its timer times out. Assuming full-duplex transmission, the receiving end sends piggybacked acknowledgement by using some number in the ACK field of its data frame.

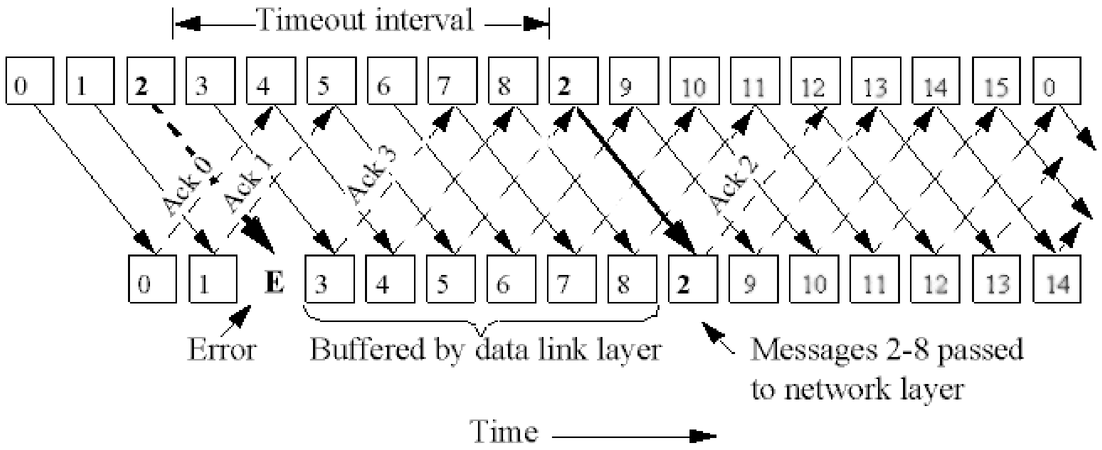
Let us assume that a 3-bit sequence number is used and suppose that a station sends frame 0 and gets back an RR1, and then sends frames 1, 2, 3, 4, 5, 6, 7, 0 and gets another RR1.This might either mean that RR1 is a cumulative ACK or all 8 frames were damaged. This ambiguity can be overcome if the maximum window size is limited to 7, i.e. for a k-bit sequence number field it is limited to 2k-1. The number N (=2k-1) specifies how many frames can be sent without receiving acknowledgement.

If no acknowledgement is received after sending N frames, the sender takes the help of a timer. After the time-out, it resumes retransmission.



Selective-Repeat ARQ

The selective-repetitive ARQ scheme retransmits only those for which NAKs are received or for which timer has expired, this is shown in the Fig.3.3.12. This is the most efficient among the ARQ schemes, but the sender must be more complex so that it can send out-of-order frames. The receiver also must have storage space to store the postNAK frames and processing power to reinsert frames in the proper sequence.



FAQs:

1. What is ARQ

Ans: Automatic Repeat reQuest (ARQ), also known as Automatic Repeat Query, is an error-control method for data transmission that uses acknowledgements (messages sent by the receiver indicating that it has correctly received a data frame or packet)

2. What is Sliding window protocol?

Ans: A sliding window protocol is a feature of packet-based data transmission protocols. Sliding window protocols are used where reliable in-order delivery of packets is required, such as in the Data Link Layer (OSI model) as well as in the Transmission ControlProtocol (TCP).

3. What is Stop and Wait Protocol?

Ans: A stop-and-wait ARQ sender sends one frame at a time; it is a special case of the general sliding window protocol with transmit and receive window sizes equal to one

4. Compare go back n and selective repeat

GO BACK AND ARQ	**SELECTIVE REPEAT ARQ**
Go Back N ARQ is inefficient for noisy link.	Selective repeat ARQ is efficient for noisy links
Go Back N ARQ is less complicated than Selective repeat ARQ.	Selective Repeat ARQ is complicated
Go Back N ARQ Sender Window Size is 2^ (m)-1 and receiver window size is 1.	In Sender and receiver Window Size is 2^ (m-1

ALGORITHM/PROGRAM:

```
 import java.lang.System;
 import java.net.*;
 import java.io.*;
   public class Client {
 static Socket connection;

            public static void main(String a[]) throws SocketException {
                        try {
                                    int v[] = new int[9];
                                    //int g[] = new int[8];
                                    int n = 0;
                                    InetAddress addr = InetAddress.getByName("Localhost");
                                    System.out.println(addr);
                                    connection = new Socket(addr, 8011);
                                    DataOutputStream out = new DataOutputStream(
                                                            connection.getOutputStream());
                                    DataInputStream in = new DataInputStream(
                                                            connection.getInputStream());
                                    int p = in.read();
                                    System.out.println("No of frame is:" + p);

                                    for (int i = 0; i < p; i++) {
                                                v[i] = in.read();
                                                System.out.println(v[i]);
                                                //g[i] = v[i];
```

```
                    }
                    v[5] = -1;
                    for (int i = 0; i < p; i++)
                     {
                          System.out.println("Received frame is: " + v[i]);
                      }
                    for (int i = 0; i < p; i++)
                            if (v[i] == -1) {
                    System.out.println("Request to retransmit packet no "
                                                    + (i+1) + " again!!");
                                    n = i;
                                    out.write(n);
                                    out.flush();
                            }
                    System.out.println();
                            v[n] = in.read();
                    System.out.println("Received frame is: " + v[n]);
                    System.out.println("quiting");
            }
                    catch (Exception e)
                    {
                            System.out.println(e);
                    }
            }
    }
```

OUTPUT:

```
No of frame is:9
30
40
50
60
70
80
90
100
110
Received frame is: 30
Received frame is: 40
Received frame is: 50
Received frame is: 60
Received frame is: 70
Received frame is: -1
Received frame is: 90
Received frame is: 100
Received frame is: 110
Request to retransmit packet no 6 again!!

Received frame is: 80
quiting
```

CONCLUSION:

The outcome of the assignment is the ability to implement go back N and selective repeat

Group B

Assignment No:9

AIM: To install and Configure DHCP server

PROBLEM STATEMENT:

Installing and configure the DHCP server and write a program (C++\Python\Java) to install the software on a remote machine.

PREREQUISITES: TCP/IP Reference model, Application Protocols.

COURSE OBJECTIVES: To configure the DHCP server and remotely install software

COURSE OUTCOMES:

CO1: Demonstrate LAN and WAN protocol behavior using Modern Tools

CO3: Demonstrate basic configuration of switches and routers

CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

COURSE OUTCOMES MAPPED:

CO4: Develop Client-Server architectures and prototypes

CO5: Implement web applications and services using application layer protocols

PROGRAM OUTCOME MAPPED: PO1,PO2,PO3,PO4,PO5

PROGRAM SPECIFIC MAPPED:

THEORY:

Dynamic Host Configuration Protocol (DHCP) The Dynamic Host Configuration Protocol (DHCP) is a network service that enables host computers to be automatically assigned settings from a server as opposed to manually configuring each network host. Computers configured to be DHCP clients have no control over the settings they receive from the DHCP server, and the configuration is transparent to the computer's user.

The most common settings provided by a DHCP server to DHCP clients include:

IP address and netmask
IP address of the default-gateway to use
IP addresses of the DNS servers to use

The advantage of using DHCP is that changes to the network, for example a change in the address of the DNS server, need only be changed at the DHCP server, and all network hosts will be reconfigured the next time their DHCP clients poll the DHCP server. As an added advantage, it is also easier to integrate new computers into the network, as there is no need to check for the availability of an IP address. Conflicts in IP address allocation are also reduced.

A DHCP server can provide configuration settings using the following

methods: Manual allocation (MAC address)

This method entails using DHCP to identify the unique hardware address of each network card connected to the network and then continually supplying a constant configuration each time the DHCP client makes a request to the DHCP server using that network device. This ensures that a particular address is assigned automatically to that network card, based on it's MAC address.

Dynamic allocation (address pool)

In this method, the DHCP server will assign an IP address from a pool of addresses (sometimes also called a range or scope) for a period of time or lease, that is configured on the server or until the client informs the server that it doesn't need the address anymore. This way, the clients will be receiving their configuration properties dynamically and on a "first come, first served" basis.

Automatic allocation

Using this method, the DHCP automatically assigns an IP address permanently to a device, selecting it from a pool of available addresses. Usually DHCP is used to assign a temporary address to a client, but a DHCP server can allow an infinite lease time.

How DHCP Works?

Before learning the process through which DHCP achieves it’s goal, we first have to understand the different messages that are used in the process.

1. DHCPDISCOVER It is a DHCP message that marks the beginning of a DHCP interaction between client and server. This message is sent by a client.
2. DHCPOFFER It is DHCP message that is sent in response to DHCPDISCOVER by a DHCP server to DHCP client. This message contains the network configuration settings for the client that sent the DHCPDISCOVER message.
3. DHCPREQUEST This DHCP message is sent in response to DHCPOFFER indicating that the client has accepted the network configuration sent in DHCPOFFER message from the server.
4. DHCPACK This message is sent by the DHCP server in response to DHCPREQUEST recieved from the client.
5. DHCPNAK This message is the exact opposite to DHCPACK described above. This message is sent by the DHCP server when it is not able to satisfy the DHCPREQUEST message from the client.
6. DHCPDECLINE This message is sent from the DHCP client to the server in case the client finds that the IP address assigned by DHCP server is already in use.
7. DHCPINFORM This message is sent from the DHCP client in case the IP address is statically configured on the client and only other network settings or configurations are desired to be dynamically acquired from DHCP server.
8. DHCPRELEASE This message is sent by the DHCP client in case it wants to terminate the lease of network address it has be provided by DHCP server.

Installation At a terminal prompt, enter the following command to install dhcpd:

sudo apt-get install isc-dhcp-server

You will probably need to change the default configuration by editing /etc/dhcp/dhcpd.conf to suit your needs and particular configuration.You also may need to edit /etc/default/isc-dhcp-server to specify the interfaces dhcpd should listen to.

Configuration

If you have two network cards in your ubuntu server you need to select which interface you want to use for DHCP server listening.By default it listens to eth0.

You can change this by editing /etc/default/dhcp3-server file

ALGORITHM:

Type following command for installation of ssh in command prompt-

–>***sudo apt-get install ssh***

1. Proceed with installation steps on Remote machine

2. After installation, for obtaining remote access, type following command-

–>***sudo ssh hostname@ipaddress***

–***For Example. >sudo ssh student@172.25.28.60***

3. Enter the password for host machine then enter the password for remote machine.

4. After login for installation of any package such as SBCL package type following command:-

–>***sudo apt-get install package_name.***

OUTPUT:

sudo apt-get install isc-dhcp-server

You will probably need to change the default configuration by editing /etc/dhcp/dhcpd.conf to suit your needs and particular configuration.You also may need to edit /etc/default/isc-dhcp-server to specify the interfaces dhcpd should listen to.

Configuration

If you have two network cards in your ubuntu server you need to select which interface you want to use for DHCP server listening.By default it listens to eth0.

You can change this by editing /etc/default/dhcp3-server file

sudo vi /etc/default/dhcp3-server

Find this line INTERFACES="eth0″

Most commonly, what you want to do is assign an IP address randomly. This can be done with settings as follows:

```
# minimal sample /etc/dhcp/dhcpd.conf

default-lease-time

 600;
```

CONCLUSION:

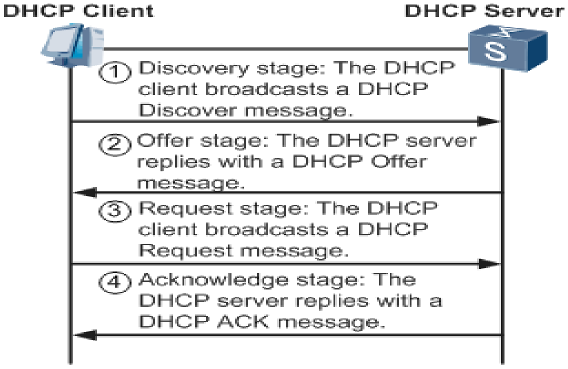
Outcome of the experiment is students are able to implement DHCP server

FAQ’s

What is Bootp? What are its drawbacks.

Ans: To automatically assign an IP address to network devices from a configuration server.Drawback: needs manual configuration by administrator

3. What messages are exchanged between a DHCP client and a DHCP server before the client receives an IP address



① Discovery stage: The DHCP client broadcasts a DHCP Discover message.

② Offer stage: The DHCP server replies with a DHCP Offer message.

③ Request stage: The DHCP client broadcasts a DHCP Request message.

④ Acknowledge stage: The DHCP server replies with a DHCP ACK message.

4. What is DHCP scope?

Ans: DHCP scopes are used to define ranges of addresses from which a DHCP server can assign IP addresses to clients.

5. Draw the DHCP 6. What is the Role of ISP?Ans: An **Internet service provider** (**ISP**) is an organization that provides services accessing and using the Internet. Internet service providers may be organized in various forms, such as commercial, community-owned, non-profit, or otherwise privately owned.

Group:B

Assignment No.10

AIM: To study several Application layer protocols

PROBLEM STATEMENT: Lab Assignment on Unit V: (Use JAVA/PYTHON)
Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer tool.

PREREQUISITE: Reference models and network architecture, Protocols

COURSE OBJECTIVES: To learn modern tools for network traffic analysis

COURSE OUTCOME MAPPED: CO2, CO5

PROGRAM OUTCOME MAPPED: PO2, PO4, PO8, PO1, PO5, PO9

PROGRAM SPECIFIC OUTCOME MAPPED: PS01, PS02

THEORY:

FTP:-

The File Transfer Program (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP is built on a client-server model architecture separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as web page editors.

HTTP:-

HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. As soon as a Web user opens their Web browser, the user is indirectly making use of HTTP. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols (the foundation protocols for the Internet).

HTTP concepts include (as the Hypertext part of the name implies) the idea that files can contain references to other files whose selection will elicit additional transfer requests. Any Web servermachine contains, in addition to the Web page files it can serve, an HTTP daemon, a program that is designed to wait for HTTP requests and handle them when they arrive. Your Web browseris an HTTP client, sending requests to server machines. When the browser user enters file requests by either "opening" a Web file (typing in a Uniform Resource Locator or URL) or clicking on a hypertext link, the browser builds an HTTP request and sends it to the Internet Protocol address (IP address) indicated by the URL. The HTTP daemon in the destination server machine receives the request and sends back the requested file or files associated with the request. (A Web page often consists of more than one file.)

The latest version of HTTP is HTTP 1.1.

Data Structures:
FTP allows three types of data structures:

1. File Structure – In file-structure there is no internal structure and the file is considered to be a continuous sequence of data bytes.
2. Record Structure – In record-structure the file is made up of sequential records.
3. Page Structure – In page-structure the file is made up of independent indexed pages.

CONCLUSION:

Outcome of the experiment is students are able to Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer tool.

FAQ's

1. What is Socket? List primitives for TCP socket communication.
   Ans: Socket is the end point of Communication. Socket(), accept(),send() ,recv()

2. **What is port address. What are reserved port addresses?**
   Ans: A **port** number is a way to identify a specific process to which an Internet or other network message is to be forwarded when it arrives at a server.
   Reserved port addresses are port addresses that are reserver for well known applications

3. **List the types of socket.**
   Ans: TCP Socket,UDP Socket,Raw Socket

4. **Compare connection oriented and connectionless services.**
   Ans: Connection oriented involves three phases a)Establish connection b) transfer data c) close Connection.
   Connection less; directly transmits data into the network.It is unreliable

Group: B

Assignment No.11

AIM : Study on different security applications.

PROBLEM STATEMENT : To study the SSL protocol by capturing the packets using wireshark tool while visiting any SSLsecured website (banking, e-commerce etc.)

PREREQUISITES : OSI,TCP/IP reference models,protocol stack

COURSE OBJECTIVES: To learn modern tools for network traffic analysis

COURSE OUTCOME MAPPED:

CO5: Implement web applications and services using application layer protocols

CO6: Use network security services and mechanisms

PROGRAM OUTCOME MAPPED: PO1, PO2, PO4, PO5, PO12

PROGRAM SPECIFIC OUTCOME MAPPED: PSO1, PSO2

THEORY :

Wireshark: This uses Wireshark to capture or examine a packet trace. A packet trace is a record of traffic at some location on the network, as if a snapshot was taken of all the bits that passed across a particular wire. The packet trace records a timestamp for each packet, along with the bits that make up the packet, from the low-layer headers to the higher-layer contents. Wireshark runs on most operating systems, including Windows, Mac and Linux. It provides a graphical UI that shows the sequence of pack- ets and the meaning of the bits when interpreted as protocol headers and data. The packets are color- coded to convey their meaning, and Wireshark includes various ways to filter and analyze them to let you investigate different aspects of behavior. It is widely used to troubleshoot networks. You can down- load Wireshark from www.wireshark.org if it is not already installed on your computer. We highly rec- ommend that you watch the short, 5 minute video <Introduction

to Wireshark= that is on the site.

wget / curl: This lab uses wget (Linux and Windows) and curl (Mac) to fetch web resources. wget and curl are command-line programs that let you fetch a URL. Unlike a web browser, which fetches and executes entire pages, wget and curl give you control over exactly which URLs you fetch and when you fetch them. Under Linux, wget can be installed via your package manager. Under Windows, wget is available as a binary; look for download information on http://www.gnu.org/software/wget/. Under Mac, curl comes installed with the OS. Both have many options (try <wget --help= or <curl --help= to see) but a URL can be fetched simply with <wget URL= or <curl URL".

Step 1: Capture a Trace

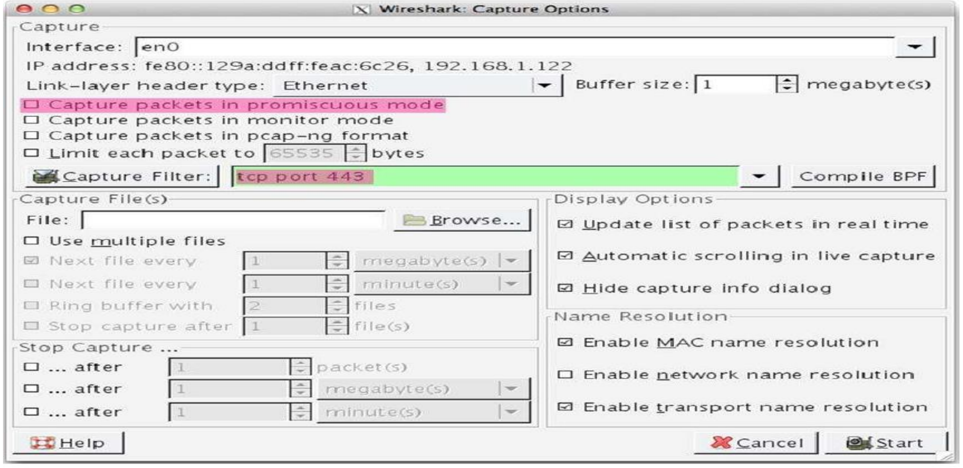
Proceed as follows to capture a trace of SSL traffic; alternatively, you may use a supplied trace. The easi- est way for us to produce SSL traffic is to fetch web pages with HTTPS. Any URL with HTTPS will do, e.g., https://www.google.com. However, web browsers have complex behaviors that can lead to a complex trace. Instead, we will use wget / curl to fetch a single HTTPS resource.

1. Close all unnecessary browser tabs and windows. Browsing web sites may generate HTTPS traf- fic. We want to minimize browser activity so that we capture only the intended DNS traffic. (You may want to redo your trace if you see unexpected HTTPS traffic due to background processes on your computer.)

2. Launch Wireshark and start a capture with a filter of <tcp port 443<. We use this filter be- cause there is no shorthand for SSL, but SSL is normally carried on port 443 in the case of secure web pages. Your capture window should be similar to the one pictured below, other than our highlighting. Select the interface from which to capture as the main wired or wireless interface used by your computer to connect to the Internet. If unsure, guess and revisit this step later if your capture is not successful. Uncheck <capture packets in promiscuous mode=. This mode is useful to overhear packets sent to/from other computers on broadcast networks. We only want to record packets sent to/from your computer. Leave other options at their default values. The capture filter, if present, is used to prevent the capture of other traffic your computer may send or receive. On Wireshark 1.8, the capture filter box is present directly on the options screen, but on Wireshark 1.9, you set a capture filter

by double-clicking on the interface.

OUTPUT:



Wireshark: Capture Options

Capture

Interface: en0

IP address: fe80::129a:ddff:feac:6c26, 192.168.1.122

Link-layer header type: Ethernet

Buffer size: 1 megabyte(s)

☐ Capture packets in promiscuous mode

☐ Capture packets in monitor mode

☐ Capture packets in pcap-ng format

☐ Limit each packet to 65535 bytes

Capture Filter: tcp port 443

Capture File(s)

File:

☐ Use multiple files

☑ Next file every 1 megabyte(s)

☐ Next file every 1 minute(s)

☐ Ring buffer with 2 files

☐ Stop capture after 1 file(s)

Stop Capture ...

☐ ... after 1 packet(s)

☐ ... after 1 megabyte(s)

☐ ... after 1 minute(s)

Display Options

☑ Update list of packets in real time

☑ Automatic scrolling in live capture

☑ Hide capture info dialog

Name Resolution

☑ Enable MAC name resolution

☐ Enable network name resolution

☑ Enable transport name resolution

CONCLUSION:

Outcome of the experiment is students are able to implement UDP socket in JAVA

Group: B

Assignment No.12

AIM: Implement a routing algorithm using given topology in NS2
PROBLEM STATEMENT: Lab Assignment on Unit IV and Unit V: (Mandatory

Assignment) Use network simulator NS2 to implement:

a. Monitoring traffic for the given topology
b. Analysis of CSMA and Ethernet protocols
c. Network Routing: Shortest path routing, AODV.

PREREQUISITES: Network protocols, routing algorithms

COURSE OBJECTIVES: To Learn and design a topology of network. To implement routing algorithm on the given network

COURSE OUTCOMES:

CO2: Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

CO3: Demonstrate basic configuration of switches and routers

CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

THEORY:

What is NS2?

- ns-2 stands for Network Simulator version 2.
- ns-2: is a discrete event simulator for networking research
- Work at packet level.
- Provide substantial support to simulate bunch of protocols like TCP, UDP, FTP, HTTP and DSR.
- Simulate wired and wireless network.
- Is primarily Unix based.
- Use TCL as its scripting language.
- ns-2 is a standard experiment environment in research community.
- otcl: Object-oriented support
- tclcl: C++ and otcl linkage

Why two languages? (Tcl & C++)

- C++: Detailed protocol simulations require systems programming language (byte manipulation, packet processing, algorithm implementation)
- – Run time speed is important
- – Turn around time (run simulation, find bug, fix bug,recompile, re-run) is slower
- Tcl: Simulation of slightly varying parameters or configurations
- – quickly exploring a number of scenarios
- – iteration time (change the model and re -run) is more important

ns-2 Directory Structure

Installation

- Get ns-2.29 all-in-one package from ns site. Get MannaSim Framework ns-2.29 patch.

http://www.mannasim.dcc.ufmg.br/download.htm

- Type the following command on the ns-allinone-2.29

folder: patch -p1 < file_name.diff

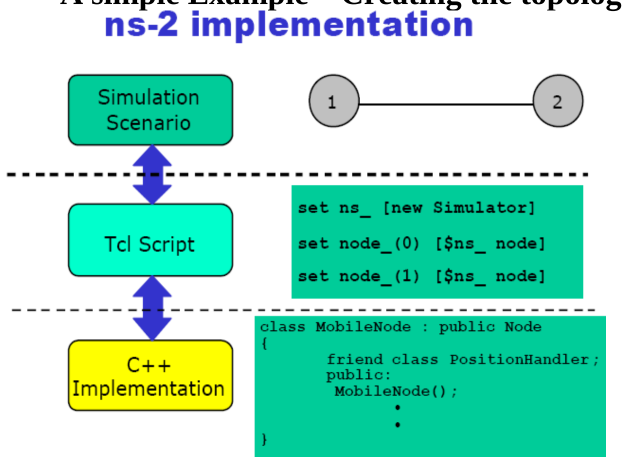
- Install ns-2.29 as usual typing ns-allinone-2.29 folder:

./install

How Do I use it?

- Creating a Simple Topology
- Getting Traces
- Using NAM

A simple Example – Creating the topology



```
set ns_ [new Simulator]
set node_(0) [$ns_ node]
set node_(1) [$ns_ node]
```

```
class MobileNode : public Node
{
        friend class PositionHandler;
        public:
         MobileNode();
                •
                •
}
```

Creating the topology

#create a new simulator object

set ns [new Simulator]

generates an NS simulator object instance, and assigns it to variable *ns*.

What this line does is the following:

- Initialize the packet format
- Create a scheduler

- Select the default address format

The "Simulator" object has member functions that do the following:

- Create compound objects such as nodes and links
- Connect network component objects created (ex. attach-agent)
- Set network component parameters
- Create connections between agents (ex. make connection between a "tcp" and "sink")
- Specify NAM display options Etc.
- **#open the nam trace file**

#Run the simulation

$ns run

To run a tcl script using NS2

- To run,

$ ns simple_network.tcl

- Internally, NS2 instantiates C++ classes based on the tcl scripts
- Output is in form of trace files or NAM file or both

CONCLUSION: Thus we have studied and implemented protocol in NS2

FAQ's:

1. What is simulation? List some simulation software's.
   Ans: Simulation software is based on the process of modeling a real phenomenon with a set of mathematical formulas .NS-2,NS-3,Opnet,neqsim etc
2. Compare TCP with UDP
   Ans:TCP is Connection oriented protocol, reliable and provides guaranteed service
   UDP is Connection less protocol, unreliable and does not provide guaranteed service
3. Give the packet formats of TCP and UDP
4. What is FTP. What are its Characteristics?
   Ans: FTP is a Connection oriented protocol. It provides two connections ,one for data transfer and other for control data transfer
5. Differentiate between Static and Dynamic Routing
   Ans: Static: routing decisions based on predetermined values of routing table
   Dynamic: routing decisions based on dynamic topology and load on network

Group:B

Assignment:13

AIM : Lab Assignment on Unit IV: (Mandatory Assignment)
Configure RIP/OSPF/BGP using packet Tracer: Simulation of WAN (RIP) using packet tracer.

PROBLEM STATEMENT: Simulation of WAN (RIP) using packet tracer.

PRE-REQUISITES: Routing protocols, topologies, networking basics

COURSE OBJECTIVES: To simulate LAN.WAN using packet tracer

COURSE OUTCOMES :

CO2: Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

CO3: Demonstrate basic configuration of switches and routers

CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

COURSE OUTCOME MAPPED:
CO1: Analyze the requirements of network types, topology and transmission media
CO3**:** Demonstrate the subnet formation with IP allocation mechanism and apply various routing algorithms.

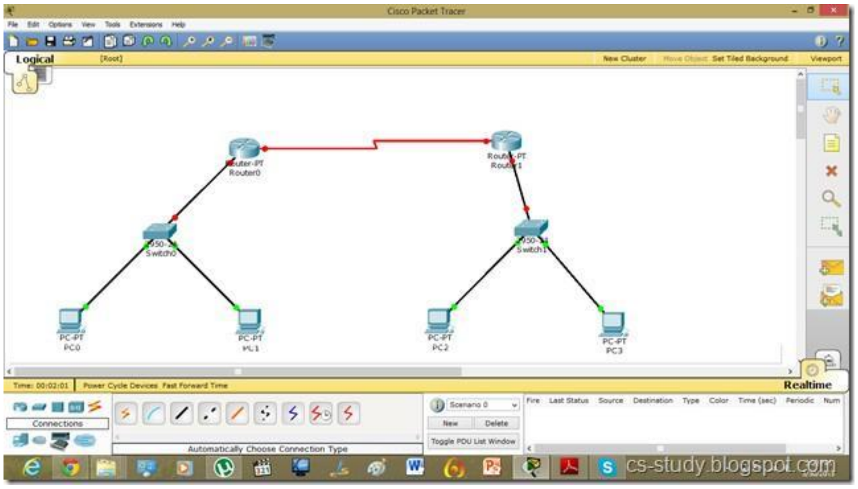
PROGRAM OUTCOME MAPPED: PO1,PO2,PO3,PO5,PO12

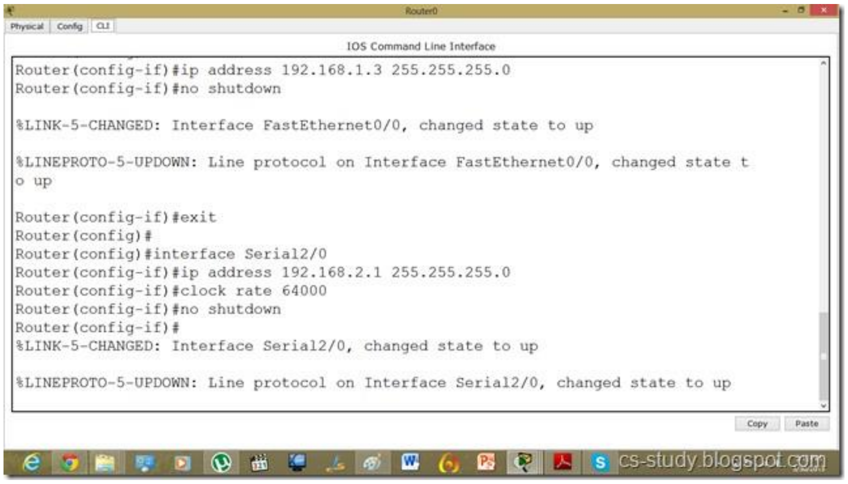
PROGRAM SPECIFIC OUTCOME MAPPED: PSO1,PSO2

THEORY:

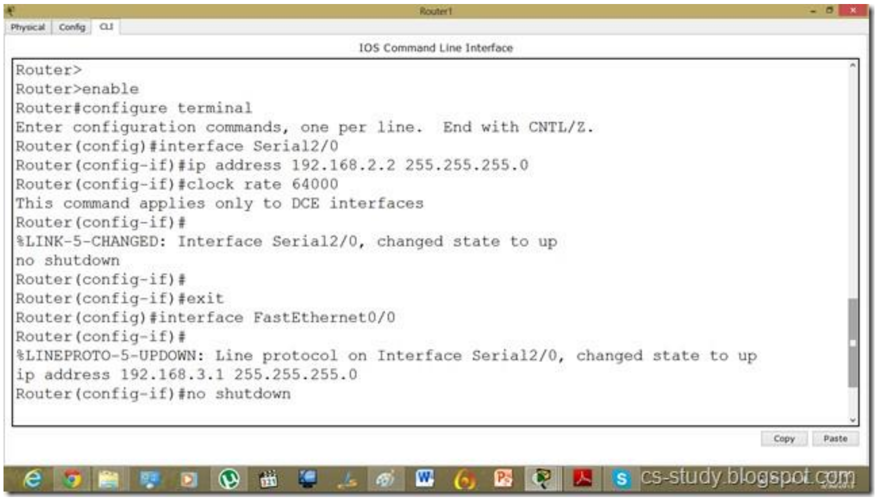
Packet Tracer (PT) is a powerful and dynamic tool that displays the various protocols used in networking, in either Real Time or Simulation mode. Routing is the process where routers exchange their network information with each other. Routing Information Protocol comes under the category of distance vector routing protocol. In RIP, destination network is calculated on the basis of distance. The metric used to calculate in RIP is hop count. The below snapshots shows the process of configuring RIP in packet tracer.

OUTPUT:



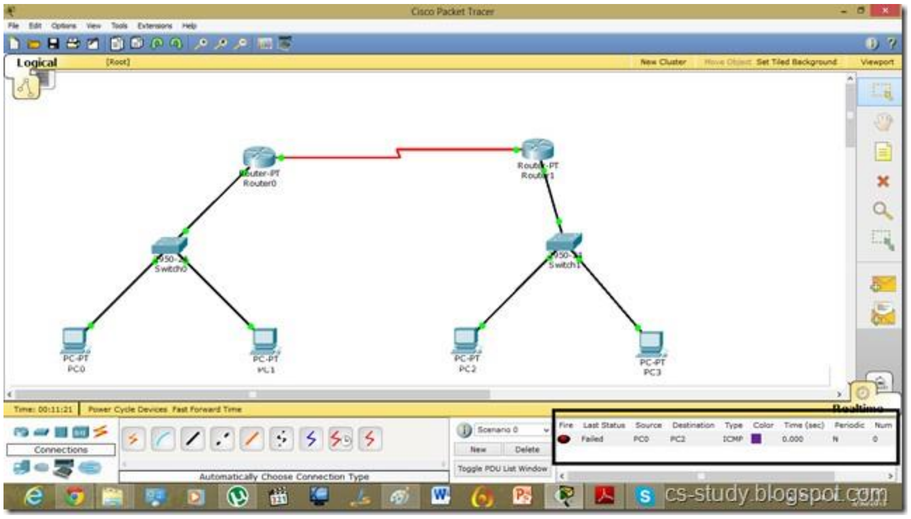


```
Router(config-if)#ip address 192.168.1.3 255.255.255.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state t
o up

Router(config-if)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
```

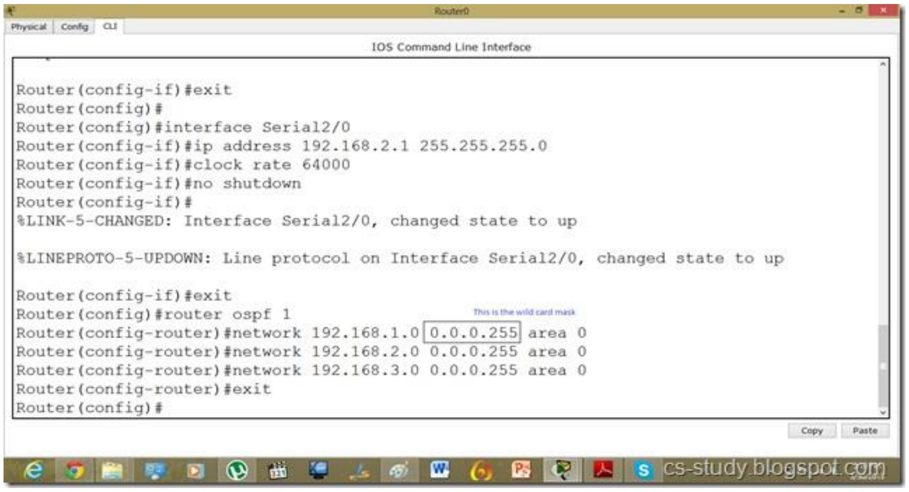


```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#ip address 192.168.2.2 255.255.255.0
Router(config-if)#clock rate 64000
This command applies only to DCE interfaces
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
no shutdown
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
ip address 192.168.3.1 255.255.255.0
Router(config-if)#no shutdown
```

Now, as we can see, interfaces are up but the communication is not enabled because we have not applied the protocol yet.



Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num
●	Failed	PC0	PC2	ICMP	🟪	0.000	N	0



```
Router(config-if)#exit
Router(config)#
Router(config)#interface Serial2/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#router ospf 1
Router(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router(config-router)#network 192.168.2.0 0.0.0.255 area 0
Router(config-router)#network 192.168.3.0 0.0.0.255 area 0
Router(config-router)#exit
Router(config)#
```

CONCLUSION: Thus we have simulated LAN/WAN using cisco packet tracer.

FAQ's

1. What is packet tracer. What is its role.
2. what are the functionalities of a router
3. Briefly explain a. Switch b. Hub
4. what is the role of bridge in networking.
5. what are the types of media. briefly explain.
6. compare cat5,cat6,cat7

Group: B
Assignment No. 14

AIM: Demonstrate DOS attack.

PROBLEM STATEMENT: Demonstration of DOS attack by using ping of death command.

PREREQUISITES:
1. Knowledge of C.
2. Knowledge of Attacks.

COURSE OBJECTIVES: To study different programming tools and methods.

COURSE OUTCOMES:
CO2: Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols.
CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

COURSE OUTCOMES MAPPED:
CO6**:** Use network security services and mechanisms

PROGRAM OUTCOMES MAPPED: PO1, PO2, PO3, PO4, PO5, PO12

PROGRAM SPECIFIC OUTCOMES MAPPED: PSO1, PSO2

THEORY:

In Computer networks, a denial-of- service (DoS) or distributed denial of service(DDoS) attack is an attempt to make a machine or network resource unavailable to its intended users.DDoS (Distributed Denial of Service) attacks are sent by two or more persons, DoS (Denial of Service) attacks are sent by one person or system. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways,and even root name servers.

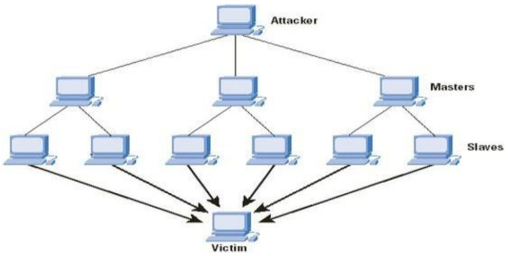


Fig a. DDOS attack

In a clear and simple way, this Cisco graphic shows the relationship of the parties in a DDOS attack.One common method of attack involves saturating the target machine with external communications requests, so much so that it cannot respond to legitimate traffic or responds so slowly as to be rendered essentially unavailable. Such attacks usually lead to a server overload.

In general terms, DOS attacks are implemented by either forcing the targeted computer(s) to reset, or consuming its resources so that it can no longer provide its intended service or obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

Denial-of- service attacks to include:

1. Unusually slow network performance (opening files or accessing web sites)
2. Unavailability of a particular web site
3. Inability to access any web site
4. Dramatic increase in the number of spam emails received (this type of DoS attack is considered an e-mail bomb)
5. Disconnection of a wireless or wired internet connection
6. Long term denial of access to the web or any internet services

Denial-of- service attacks can also lead to problems in the network &#39;branches&#39; around the actual computer being attacked.

Methods of Attack:

A **denial-of- service** attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. There are two general forms of DoS attacks: those that crash services and those that flood services

Internet Control Message Protocol (ICMP) flood a smurf attack is one particular variant of a flooding DoS attack on the public Internet. It relies on misconfigured network devices that allow packets to be sent to all computer hosts on a particular network via the broadcast address of the network, rather than a specific machine. The network then serves as a smurf amplifier. In such an attack, the perpetrators will send large numbers of IP packets with the source address faked to appear to be the address of the victim. The networks bandwidth is quickly used up, preventing legitimate packets from getting through to their destinationPing flood is based on sending the victim an overwhelming number of ping packets, usually using the ping command from Unix-like hosts (the -t flag on Windows systems is much less capable of overwhelming a target, also the -l (size) flag does not allow sent packet size greater than 65500 in Windows). It is very simple to launch, the primary requirement being access to greater bandwidth than the victim.

Ping of death is based on sending the victim a malformed ping packet, which might lead to a system crash.

(S)SYN flood

A SYN flood occurs when a host sends a flood of TCP/SYN packets, often with a forged sender address. Each of these packets is handled like a connection request, causing the server to spawn a half-open connection, by sending back a TCP/SYN-ACK packet (Acknowledge), and waiting for a packet in response from the sender address (response to the ACK Packet). However, because the sender address is forged, the response never comes. These half-open connections saturate the number of available connections the server is able to make, keeping it fromresponding to legitimate requests until after the attack ends.

Teardrop attacks

A teardrop attack involves sending mangled IP fragments with overlapping, over-sized payloads to the target machine. This can crash various operating systems because of a bug in their TCP/IP fragmentation re-assembly code. Windows 3.1x,Windows 95 and Windows NT operating systems, as well as versions of Linux prior to versions 2.0.32 and 2.1.63 are vulnerable to this attack.

Around September 2009, a vulnerability in Windows Vista was referred to as a teardrop attack.

But the attack targeted SMB2 which is a higher layer than the TCP packets that teardrop used.
Internet Control Message Protocol (ICMP) flood

This DOS attack could be carried out even through the command line, in the following manner:

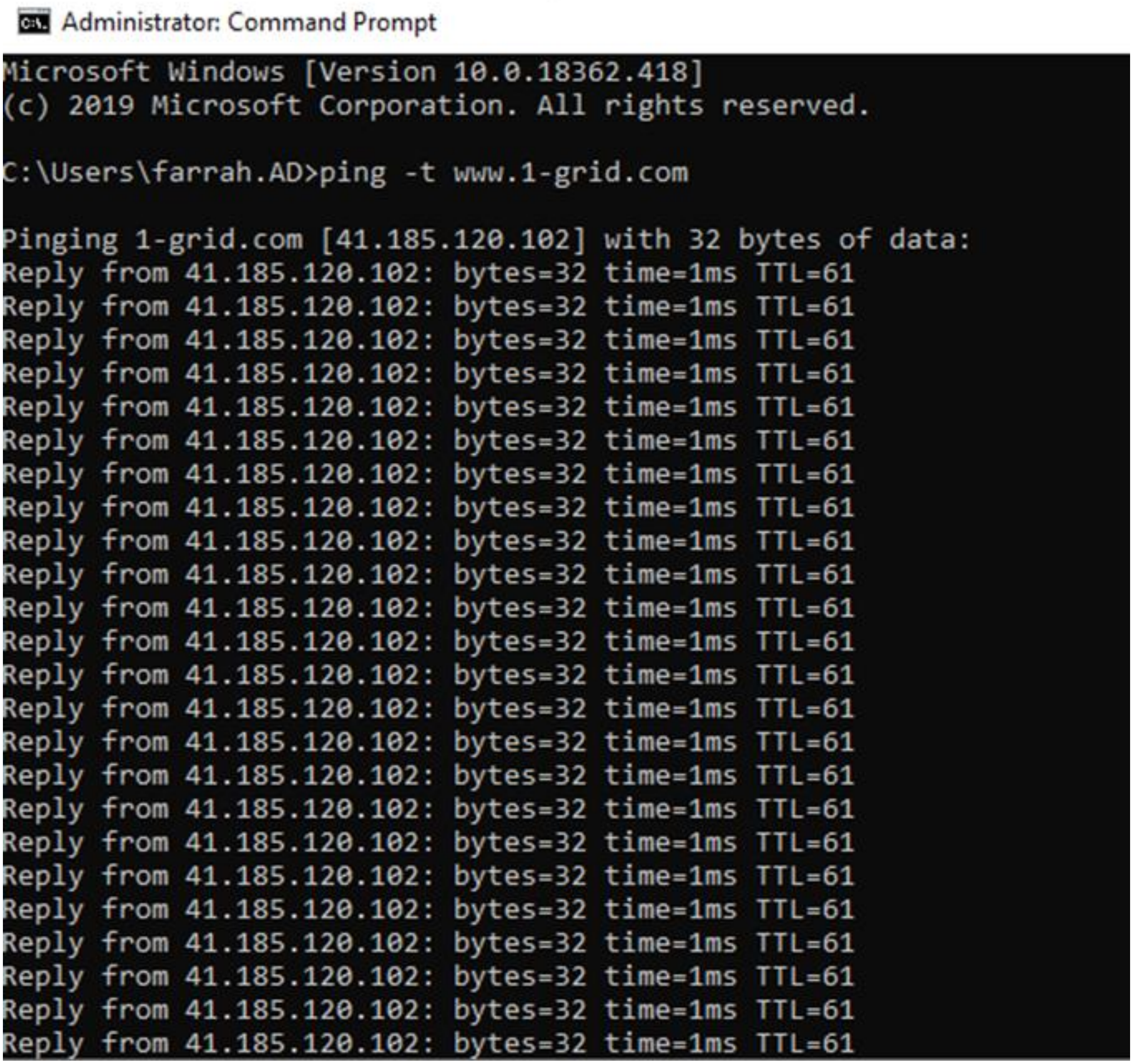
The following Ping command creates a giant packet size 65500 for Ping. It might hang the victims computer:

C:\windows: ping (ip address) –l 65500

Example: C:\windows: ping www.1-grid.com

Continues ping

OUTPUT:



```
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\farrah.AD>ping -t www.1-grid.com

Pinging 1-grid.com [41.185.120.102] with 32 bytes of data:
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
Reply from 41.185.120.102: bytes=32 time=1ms TTL=61
```

CONCLUSION: Thus we have demonstrated a DOS attack by using ping of death command.

Group: B

Assignment No.15

AIM: VLAB Assignment Demonstration

PROBLEM STATEMENT: To learn the wireless communication between the sensor nodes.

PREREQUISITES: 1.Wireless hardware devices

COURSE OBJECTIVES: To study VLAB.

COURSE OUTCOMES:

CO2: Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols

CO4: Develop Client-Server architectures and prototypes by the means of correct standards and technology.

THEORY:

Virtual Labs is a project initiated by the Ministry of Human Resource Development, Government of India, under the National Mission on Education through Information and Communication Technology. The project aims to provide remote-access to Laboratories in various disciplines of science and engineering for students at all levels from under-graduate to research

Wireless Sensor Network Remote Triggered Lab

Wireless Sensor Network is the study of wireless sensors that are distributed in a wide area for sensing the environmental parameters. Wireless Sensor Network Remote Triggered Lab is an experimental wireless sensor network deployed partly indoor and partly outdoor. This lab is envisioned to provide a practical experience of designing, deploying and implementing wireless sensor networks in both indoor and outdoor conditions

Here the required components for send and receive vlab assignment

Implement the module file

The functions and events implemented in this program

Display(): This function keeps on toggling LEDs between the pattern 101 and 010 upon successful transmission of each packet

sendPacket(): This function comprise the packet and broadcaste it over the radio

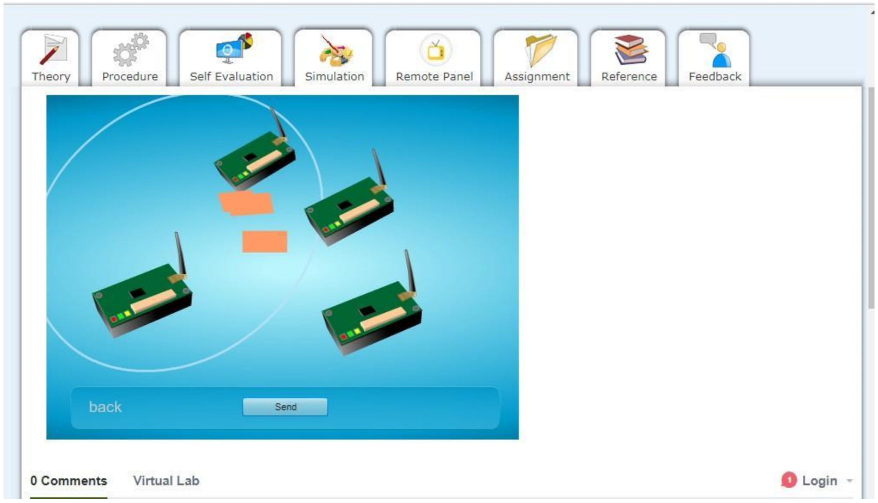
readData(): This function reads the current letter from string and pass it to the sendPacket() function. Once the last is being sent, it will reset the pointer to the beginning and so on receiveM – the module file of our application

LedsC – provides Leds interface

GenericComm – provides both SendMsg and ReceiveMsg interface

The steps included in running the regular Send and Receive experiment in remote panel

OUTPUT:



CONCLUSION: Thus we have simulated LAN/WAN using VLAB.