Dr. D. Y. Patil Technical Campus
Dr. D. Y. Patil School of Engineering
Dr. D. Y. Patil Knowledge City, Charholi (Bk), Lohegaon, Pune – 412 105

Department of Computer Engineering

Computer Networks and Security Laboratory
(T. E. Computer) 2019 Course

LABORATORY MANUAL
(Version 1, w. e. f. Jan 2022)

Prepared By
Prof. Renuka Gavli
Prof. Amruta Chitari
Prof. Swati Bagade
(Asst. Professor)

Approved By
Dr. P. M. Agarkar (H. O. D.)

Sr. No.	Expt. No.	Name of Experiment	Date of Performance	Date of Submission	Remark/ Sign
1.	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.	2	Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool.			
3.	4	Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.			
4.	5	Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.			
5.	6	Write a program to demonstrate Sub-netting and find subnet masks.			
6.	7	Write a program to implement link state /Distance vector routing protocol to find suitable pathfrtransmission.			
7.	8	Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF/BGP.			
8.	9	Write a program using TCP socket for wired network for following a. Say Hello to Each other b. File transfer c. Calculator			
9.	10	Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines.			
10.	11	Write a program for DNS lookup. Given an IP address as input, it should return URL and vice- versa.			
11.	12	Installing and configure DHCP server and write a program to install the software on remote machine.			
12.	14	Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer tool.			
13.	15	To study the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website (banking, e-commerce etc.).			
14.	16	Illustrate the steps for implementation of S/MIME email security through Microsoft® Office Outlook.			

Assignment 1

Title: Setup a wired LAN using switch

Problem Statement:
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.

Objectives:

1. To understand the structure and working of various networks including the interconnecting devices used in them.
2. To get hands on experience of making and testing cables.

Outcomes:
Develop and demonstrate a wired LAN for four computers.

Tools Required:

Hardware: Computer, LAN Cards, RJ-45 Connectors, Switch, CAT-5 Cable, Cable tester, Crimping tool, etc.
Software: Open source O.S.and wireshark

Theory:

LAN - Local Area Network

A LAN connects network devices over a relatively short distance. A networked office building, school, or home usually contains a single LAN, though sometimes one building will contain a few small LANs (perhaps one per room), and occasionally a LAN will span a group of nearby buildings.

MAN-Metropolitan Area Network

A network spanning a physical area larger than a LAN but smaller than a WAN, such as a city. A MAN is typically owned and operated by a single entity such as a government body or large corporation.

WAN:

A **wide area network** (**WAN**) is a telecommunications network or computer network that extends over a large geographical distance. Wide area networks are often established with leased telecommunication circuits. Business, education and government entities use wide area networks to relay data to staff, students, clients, buyers, and suppliers from various locations across the world. In essence, this mode of telecommunication allows a business to effectively carry out its daily function regardless of location. The Internet may be considered a WAN

What is Network Cabling?

Cable is the medium through which information usually moves from one network device to another. There are several types of cable which are commonly used with LANs. In some cases, a network will utilize only one type of cable, other networks will use a variety of cable types. The type of cable chosen for a network is related to the network's topology, protocol, and size. Understanding the characteristics of different types of cable and how they relate to other aspects of a network is necessary for the development of a successful network. The following sections discuss the types of cables used in networks and other related topics.

- Unshielded Twisted Pair (UTP) Cable
- Shielded Twisted Pair (STP) Cable
- Coaxial Cable
- Fiber Optic Cable
- Cable Installation Guides
- Wireless LANs
- Unshielded Twisted Pair (UTP) Cable

Twisted pair cabling comes in two varieties: shielded and unshielded. Unshielded twisted pair (UTP) is the most popular and is generally the best option for school networks (See fig. 1).

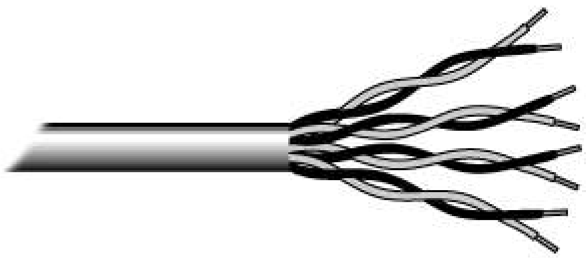


Fig.1. Unshielded twisted pair

The quality of UTP may vary from telephone-grade wire to extremely high-speed cable. The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other electrical devices. The tighter the twisting, the higher the supported transmission rate and the greater the cost per foot. The EIA/TIA (Electronic Industry Association/Telecommunication Industry Association) has established standards of UTP and rated six categories of wire (additional categories are emerging).

Categories of Unshielded Twisted Pair

Category	Speed	Use
1	1 Mbps	Voice Only (Telephone Wire)
2	4 Mbps	LocalTalk & Telephone (Rarely used)
3	16 Mbps	10BaseT Ethernet
4	20 Mbps	Token Ring (Rarely used)
5	100Mbps(2 pair)	100BaseT Ethernet
6	10,000 Mbps	Gigabit Ethernet

Unshielded Twisted Pair Connector

The standard connector for unshielded twisted pair cabling is an RJ-45 connector. This is a plastic connector that looks like a large telephone-style connector (See fig. 2). A slot allows the RJ-45 to be inserted only one way. RJ stands for Registered Jack, implying that the connector follows a standard borrowed from the telephone industry. This standard designates which wire goes with each pin inside the connector.

Fig. 2. RJ-45 connector

Shielded Twisted Pair (STP) Cable

Although UTP cable is the least expensive cable, it may be susceptible to radio and electrical frequency interference (it should not be too close to electric motors, fluorescent lights, etc.). If you must place cable in environments with lots of potential interference, or if you must place cable in extremely sensitive environments that may be susceptible to the electrical current in the UTP, shielded twisted pair may be the solution. Shielded cables can also help to extend the maximum distance of the cables.

Shielded twisted pair cable is available in three different configurations:

1. Each pair of wires is individually shielded with foil.

2. There is a foil or braid shield inside the jacket covering all wires (as a group).
3. There is a shield around each individual pair, as well as around the entire group of wires (referred to as double shield twisted pair).

Coaxial Cable

Coaxial cabling has a single copper conductor at its center. A plastic layer provides insulation between the center conductor and a braided metal shield (See fig. 3). The metal shield helps to block any outside interference from fluorescent lights, motors, and other computers.

Fig. 3. Coaxial cable

Although coaxial cabling is difficult to install, it is highly resistant to signal interference.
In addition, it can support greater cable lengths between network devices than twisted pair cable. The two types of coaxial cabling are thick coaxial and thin coaxial. Thin coaxial cable is also referred to as thinnet. 10Base2 refers to the specifications for thin coaxial cable carrying Ethernet signals. The 2 refers to the approximate maximum segment length being 200 meters. In actual fact the maximum segment length is 185 meters. Thin coaxial cable has been popular in school networks, especially linear bus networks.
Thick coaxial cable is also referred to as thicknet. 10Base5 refers to the specifications for thick coaxial cable carrying Ethernet signals. The 5 refers to the maximum segment length being 500 meters. Thick coaxial cable has an extra protective plastic cover that helps keep moisture away from the center conductor. This makes thick coaxial a great choice when running longer lengths in a linear bus network. One disadvantage of thick coaxial is that it does not bend easily and is difficult to install.

Coaxial Cable Connectors

The most common type of connector used with coaxial cables is the Bayone-Neill-Concelman (BNC) connector (See fig. 4). Different types of adapters are available for BNC connectors, including a T-connector, barrel connector, and terminator. Connectors on the cable are the weakest points in any network. To help avoid problems with your network, always use the BNC connectors that crimp, rather screw, onto the cable.

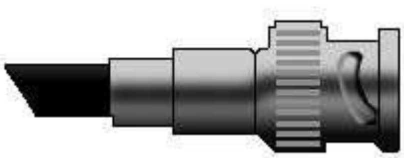


Fig:4 BNC Connector

Fiber Optic Cable

Fiber optic cabling consists of a center glass core surrounded by several layers of protective materials (See fig. 5). It transmits light rather than electronic signals eliminating the problem of electrical interference. This makes it ideal for certain environments that contain a large amount of electrical interference. It has also made it the standard for connecting networks between buildings, due to its immunity to the effects of moisture and lighting.

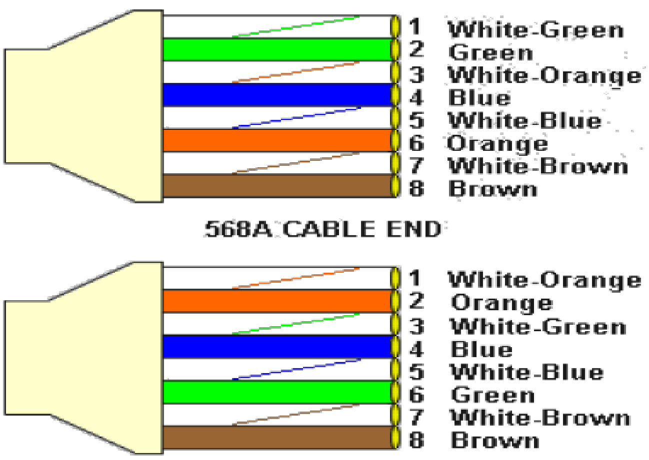
Fiber optic cable has the ability to transmit signals over much longer distances than coaxial and twisted pair. It also has the capability to carry information at vastly greater speeds. This capacity broadens communication possibilities to include services such as video conferencing and interactive services. The cost of fiber optic cabling is comparable to copper cabling; however, it is more difficult to install and modify. 10BaseF refers to the specifications for fiber optic cable carrying Ethernet signals.

The center core of fiber cables is made from glass or plastic fibers (see fig 5). A plastic coating then cushions the fiber center, and kevlar fibers help to strengthen the cables and prevent breakage. The outer insulating jacket made of teflon or PVC.

Fig. 5: Fiber optic cable

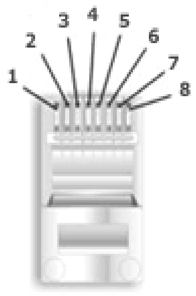
<u>Paring Rules and Color Code:-</u>

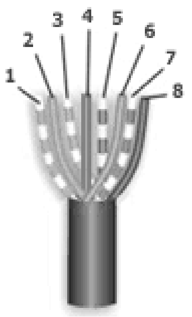
Colour Code



The CAT 6 Cable consist of 8 wires which comes pares of White/Blue, Blue, White/Orange, Orange, White/Green, Green, White/Brown, Brown and they are coded for **Straight** and **Cross** combinations respectively.

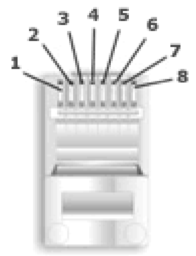
Straight:-

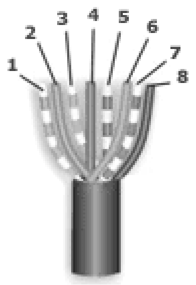




Pair #	Wire	Pin #
1-White/Blue	White/Blue	5
	Blue	4
2-Wht./Orange	White/Orange	1
	Orange	2
3-White/Green	White/Green	3
	Green	6
4-White/Brown	White/Brown	7
	Brown	8

<u>Cross</u>





Pair #	Wire	Pin #
1-White/Blue	White/Blue	5
	Blue	4
2-White/Green	White/Green	1
	Green	2
3-White/Orange	White/Orange	3
	Orange	6
4-White/Brown	White/Brown	7
	Brown	8

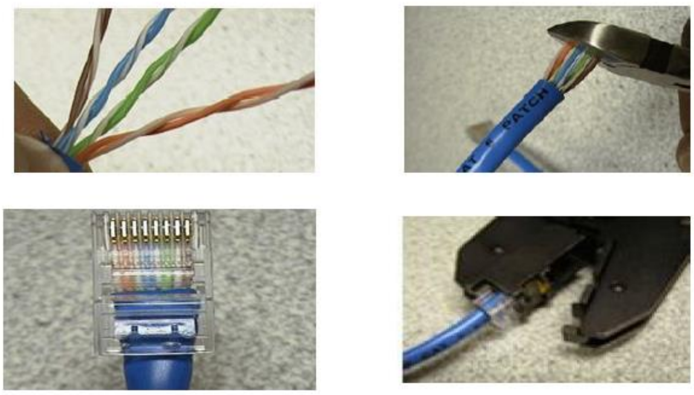
Connections among devices:-

Node to Node -Straight -Cross,
Switch to Node -Straight – Straight,
Switch to Switch - Straight

How to Crimp a Cat 5 cable with RJ 45 Connector:-

1. Skin off the cable jacket approximately 1" or slightly more.
2. Un-twist each pair, and straighten each wire between the fingers.
3. Place the wires in the order of one of the two diagrams shown above .Bring all of the wires together, until they touch.
4. At this point, recheck the wiring sequence with the diagram.
5. Optional: Make a mark on the wires at 1/2" from the end of the cable jacket.
6. Hold the grouped (and sorted) wires together tightly, between the thumb, and the forefinger.
7. Cut all of the wires at a perfect 90 degree angle from the cable at 1/2" from the end of the cable jacket. This is a very critical step. If the wires are not cut straight, they may not all make contact. We suggest using a pair of scissors for this purpose.
8. Conductors should be at a straight 90 degree angle, and be 1/2" long, prior to insertion into the connector.
9. Insert the wires into the connector (pins facing up).
10. Push moderately hard to assure that all of the wires have reached the end of the connector. Be sure that the cable jacket goes into the back of the connector by about 3/16".
11. Place the connector into a crimp tool, and squeeze hard so that the handle reaches its full swing.
12. Repeat the process on the other end. For a straight through cable, use the same wiring.
13. Use a cable tester to test for proper continuity.

Cable Preparation:

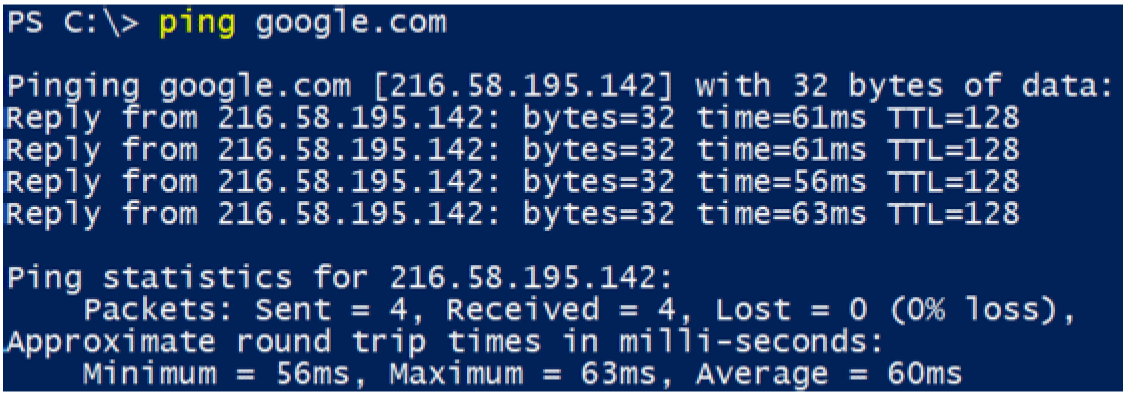


Orange – 1 & 2 Green – 3 & 6
Blue- 4 & 5 Brown- 7 & 8

PING Command:

A ping (Packet Internet or Inter-Network Groper) is a basic Internet program that allows a user to test and verify if a particular destination IP address exists and can accept requests in computer network administration.

Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network. It measures the round-trip time for messages sent from the originating host to a destination computer that are echoed back to the source .Ping operates by sending Internet Control Message Protocol (ICMP/ICMP6) Echo Request packets to the target host and waiting for an ICMP Echo Reply.
The program reports errors, packet loss, and a statistical summary of the results, typically including the minimum, maximum, the mean round-trip times, and standard deviation of the mean. The command-line options of the ping utility and its output vary between the numerous implementations.



```
PS C:\> ping google.com

Pinging google.com [216.58.195.142] with 32 bytes of data:
Reply from 216.58.195.142: bytes=32 time=61ms TTL=128
Reply from 216.58.195.142: bytes=32 time=61ms TTL=128
Reply from 216.58.195.142: bytes=32 time=56ms TTL=128
Reply from 216.58.195.142: bytes=32 time=63ms TTL=128

Ping statistics for 216.58.195.142:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 56ms, Maximum = 63ms, Average = 60ms
```

Conclusion: Thus we have studied wired LAN setup and connection.

Assignment 2

Title: Types of technologies and types of transmission media.

Problem Statement:

Demonstrate the different types of technologies and types of transmission media using a packet tracer tool.

Objectives:

1. To understand the working of different types of topologies.
2. To understand the transmission media.
3. To understand the working of Cisco packet tracer tool.

Outcomes:

Demonstrate types of technologies and types of transmission media using Cisco Packet tracer tool.

Tools Required:

Software: Cisco Packet Tracer

Procedure:

- Open the CISCO Packet tracer software
- Drag and drop 4 pcs using End Device Icons on the left corner
- Select 8 port switch from switch icon list in the left bottom corner
- Make the connections using Straight through Ethernet cables
- Give IP address of the PCs as per table, ping between PCs and observe the transfer of data packets in real and simulation mode.

Theory:

Cisco Packet Tracer:

Cisco Packet Tracer (CPT) is multi-tasking network simulation software to perform and analyze various network activities such as implementation of different topologies, select optimum path based on various routing algorithms, create DNS and DHCP server, sub netting, analyze various network configuration and troubleshooting commands. In order to start communication between end user devices and to design a network, we need to select appropriate networking devices like routers, switches, hubs and make physical Connection by connection cables to serial and fast Ethernet ports from the component list of packet tracer. Networking devices are costly so it is better to perform first on packet tracer to understand the concept and behavior of networking.

Types of Topologies:

Bus Topology

In local area network, it is a single network cable runs in the building or campus and all nodes are connected along with this communication line with two endpoints called the bus or backbone. In other words, it is a multipoint data communication circuit that is easily control data flow between the computers because this configuration allows all stations to receive every transmission over the network. For bus topology we build network using three generic pc which are serially connected with three switches using copper straight through cable and switches are interconnected using copper cross over cable.

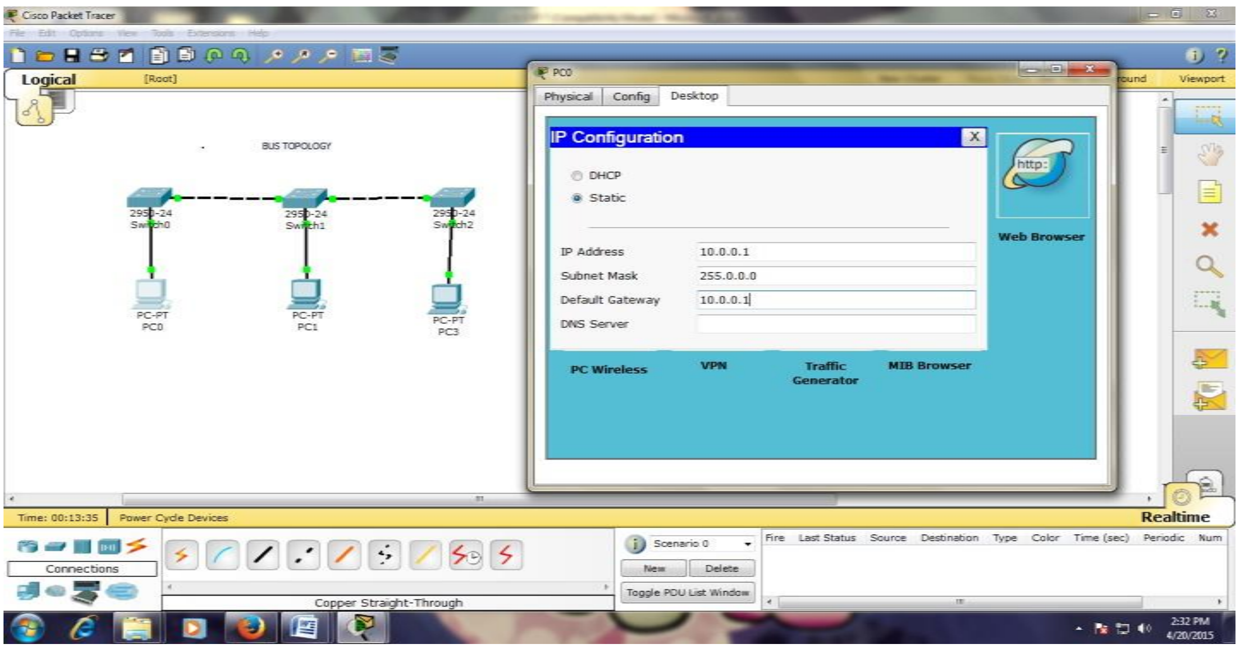


Fig -1: Design of bus topology

Star Topology:

In star topology, all the cables run from the computers to a central location where they are all connected by a device called a hub. It is a concentrated network, where the end points are directly reachable from a central location when network is expanded .Ethernet 10 base T is a popular network based on the star topology. For star topology we build network using five generic pc which are centrally connected to single switch 2950 -24 using copper straight through cable.

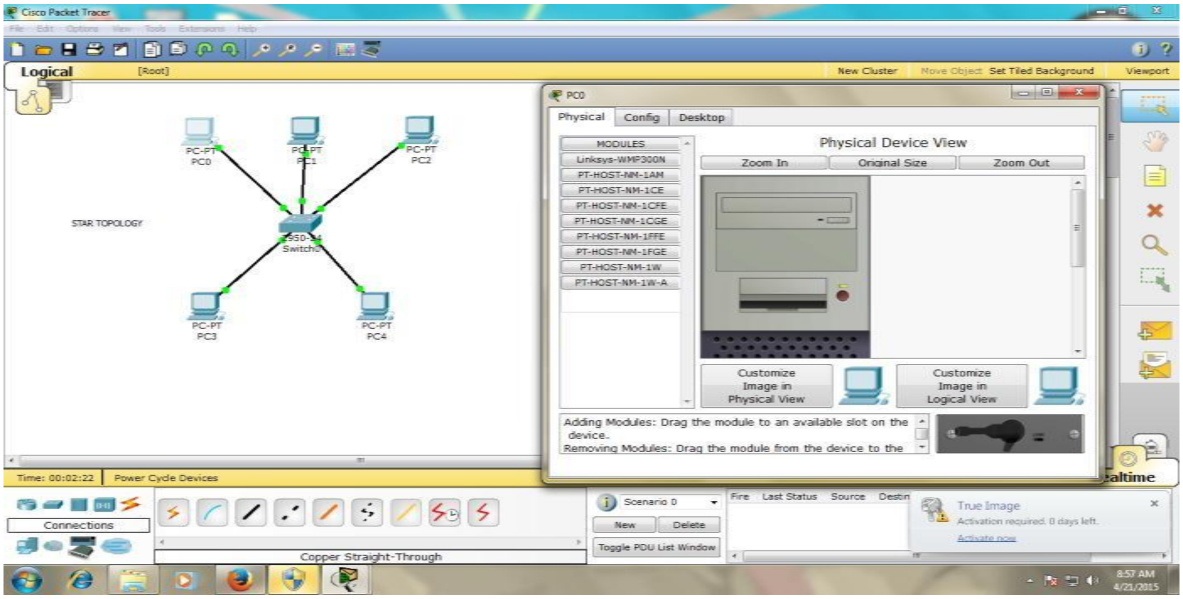


Fig -2: Design of star topology

Mesh Topology:

In mesh topology every device has a dedicated point to point link to every other device. The term dedicated stand for link carries traffic only between two devices it connects. It is a well-connected topology; in this every node has a connection to every other node in the network. The cable requirements are high and it can include multiple topologies. Failure in one of the computers does not cause the network to break down, as they have alternative paths to other computers star topology, all the cables run from the computers to a central location For mesh topology we build network using five 1841 router. To design four serial port router click on router ->turn off->drag the WIC2T module two t times.->power on To establish connection between router to router using DCE cables.

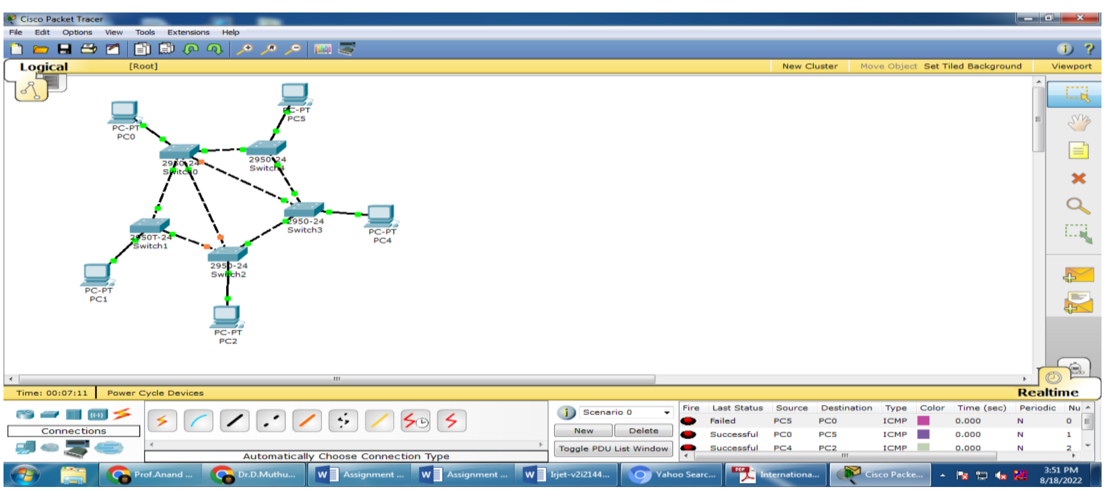


Fig -3: Design of mesh topology

Configuration of component:

Bus topology: To configure the IP address of an interface, we configure all PC one by one click on pc, open DESKTOP window, fill IP Address, Fill subnet mask and default gateway. After that, simulate the network using simulation

Star topology: To configure the IP address of an interface, we configure all PC one by one click on pc, open DESKTOP window, fill IP Address, Fill subnet mask and default gateway. After that, simulate the network using simulation.

Mesh topology: To configure the IP address of an interface, we configure all routers one by one. Click on router, open config window, and fill IP Address of serial port which are connected to router. Fill subnet mask, set clock rate and port status is ON. After that, simulate the network using simulation mode.

Simulation of network topology:

1. **Simulation of bus topology:**

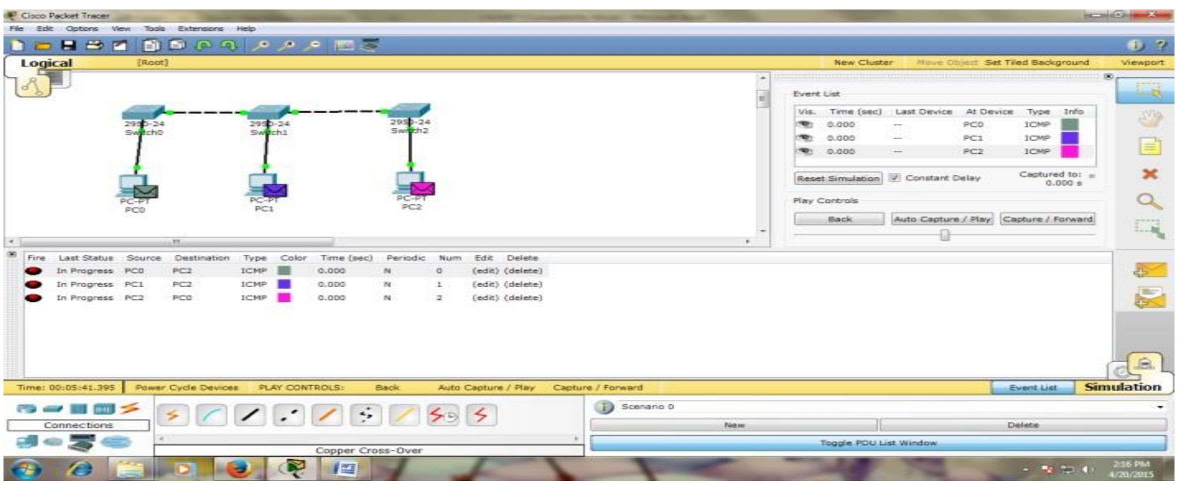


Fig.4 Simulation of Bus Topology

2. **Simulation of star topology:**

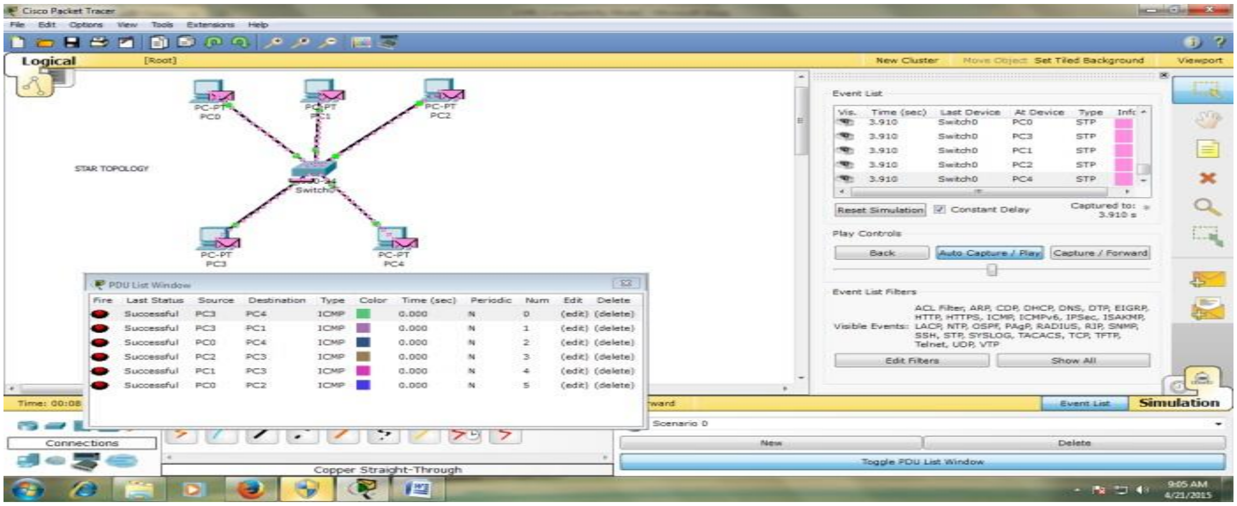


Fig.5 Simulation of Star Topology

3. Simulation of Mesh topology:

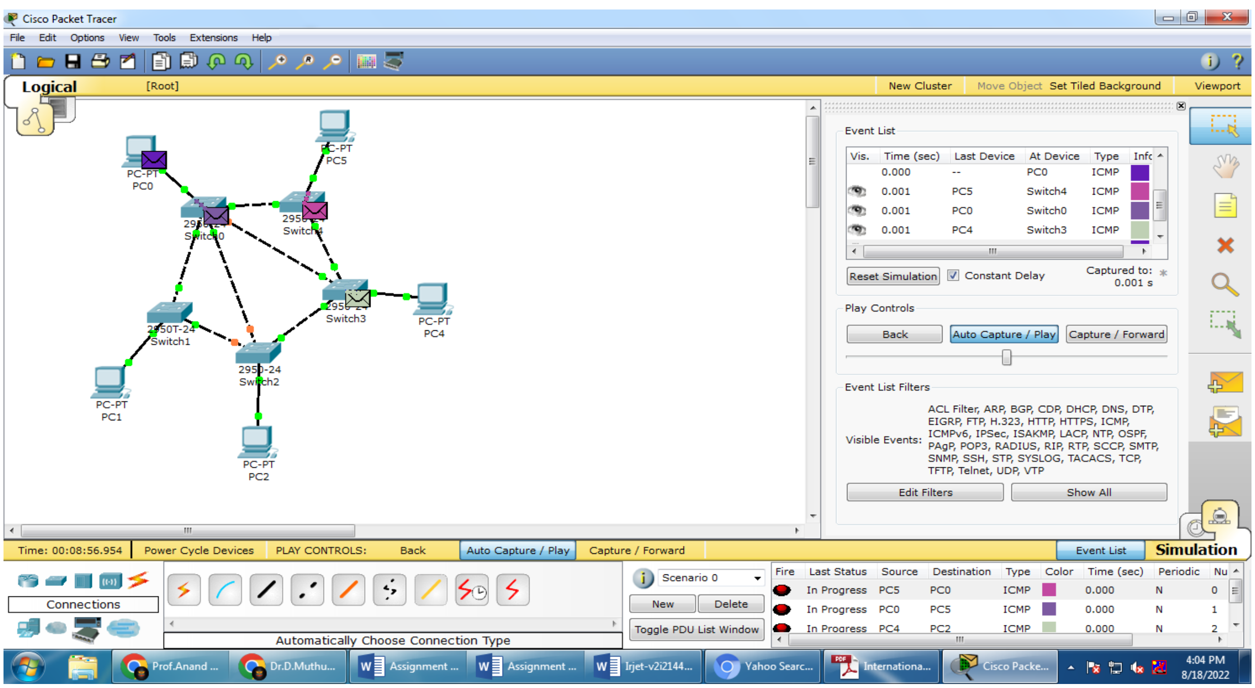


Fig.6 Simulation of Mesh Topology

Conclusion: Thus we have implemented various topologies in a single network using Cisco Packet Tracer. We have used switch configuration and send packet data from one device to another.

Assignment 4

Problem Definition:

Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.

Problem Definition:

Write a Program for Error Detection & correction for 7 Bit Ascii using Hamming Codes.

3.1 Prerequisite:

1. Data Communication
2. Basic Logical Operations.

3.2 Learning Objectives:

1. Understand the concept of Error Analysis.
2. Detection of Error at Reciever Side.

3.3 New Concepts:

1. Even Parity
2. Odd Parity

3.4 Theory

3.4.1 Introduction

Hamming code

Hamming codes are a family of linear error-correcting codes that generalize the Hamming(7,4)-code

- Invented by Richard Hamming in 1950

Hamming codes can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors.

General 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. 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.

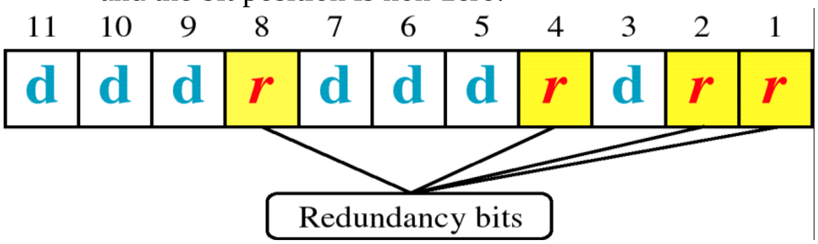
• Parity bit 1 covers all bit positions which have the least significant bit set: bit 1 (the parity bit itself), 3, 5, 7, 9, etc.

• 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.

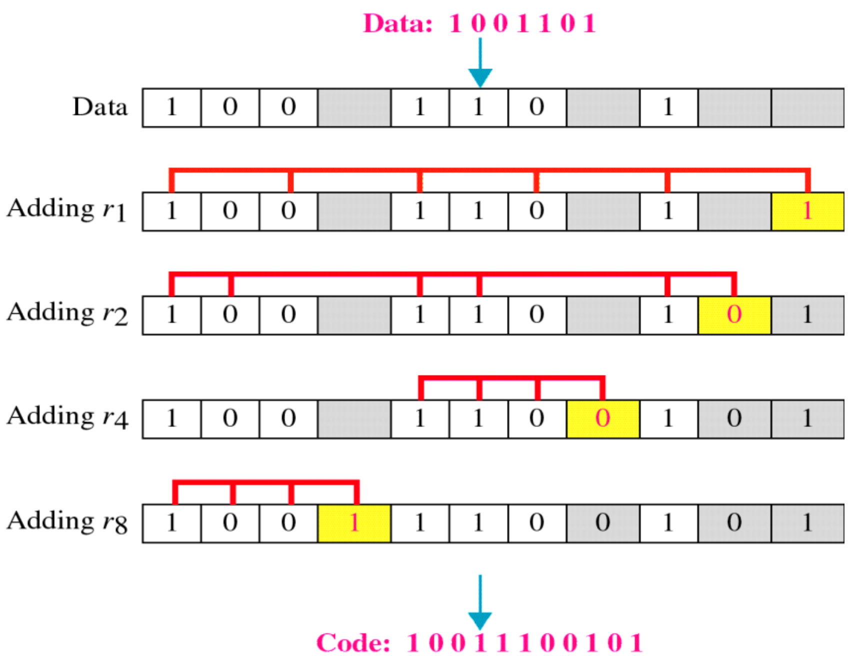
• Parity bit 4 covers all bit positions which have the third least significant bit set: bits 4–7, 12–15, 20–23, etc.

• Parity bit 8 covers all bit positions which have the fourth least significant bit set: bits 8–15, 24–31, 40–47, etc.

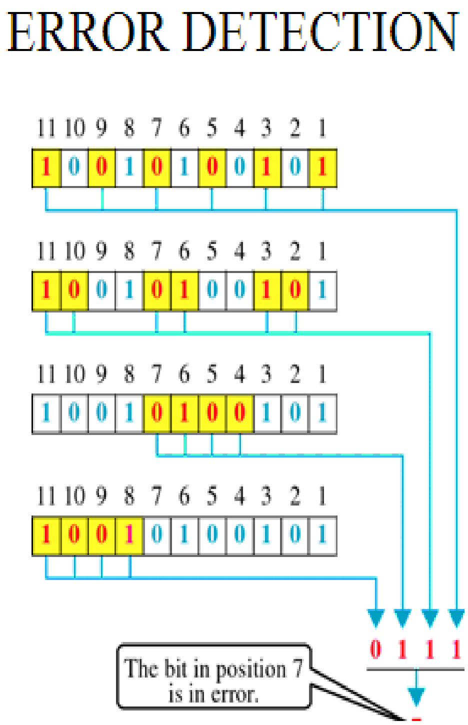
• In general each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.



Example
Error detection



Error correction

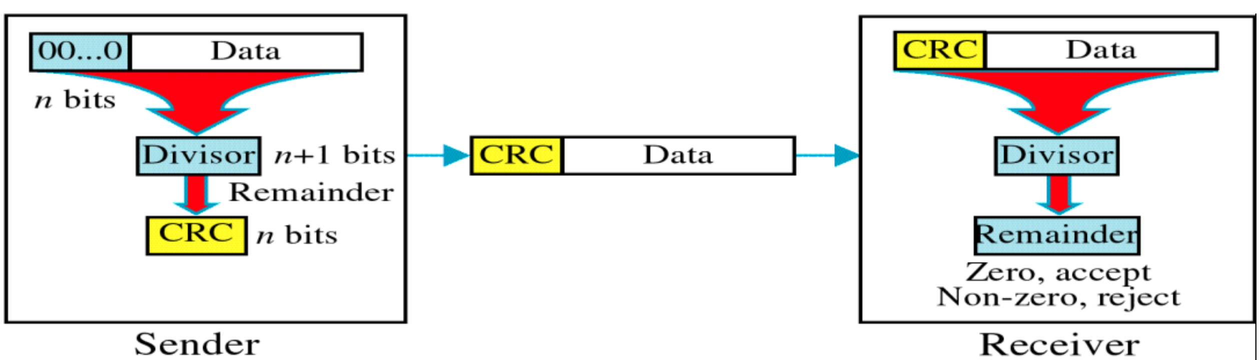


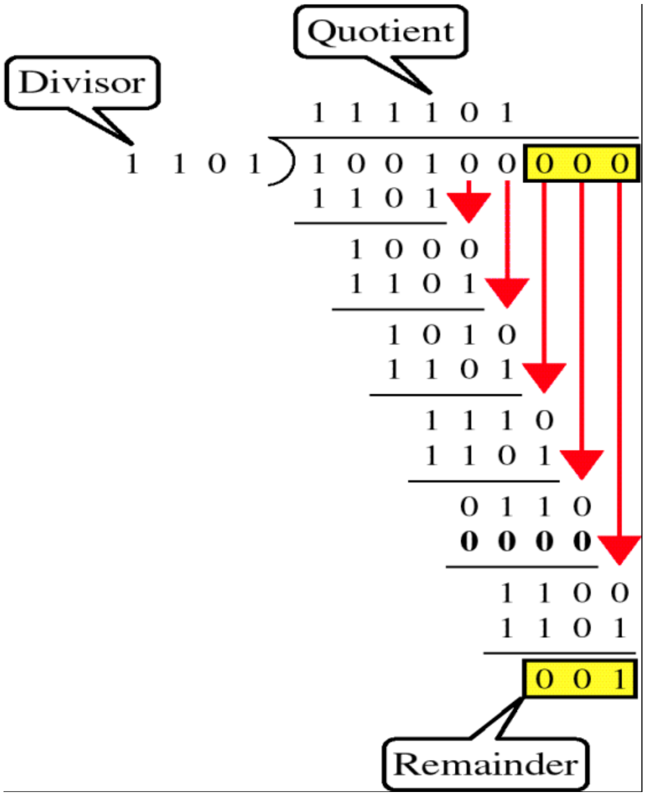
3.5 CRC

Cyclic Redundancy Check: CRC

• Given a k-bit frame or message, the transmitter generates an n-bit sequence, known as a *frame check sequence (FCS),* so that the resulting frame, consisting of (k+n) bits, is exactly divisible by some predetermined number.

• The receiver then divides the incoming frame by the same number and, if there is no remainder, assumes that there was no error





3.6 Assignment Questions:

1. What is the importance of Hamming Code?
2. What is the Difference between Even & Odd parity?
3. Write down steps to detect and correct error with example ?

Conclusion:

Here we conclude that the Message can be detected & corrected using Hamming Code.

Assignment 5: Group A (Unit I & II)

Problem Definition:

Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.

Problem Definition:

Implementation of sliding window protocol Java.

1.1 Prerequisite:

Data Link layer working

1.2 Learning Objective:

- To understand how data is transferred reliable and in-order.

1.3 Theory:

1.3.1 Introduction

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 Control Protocol (TCP). Conceptually, each portion of the transmission is assigned a unique consecutive sequence number, and the receiver uses the numbers to place received packets in the correct order, discarding duplicate packets and identifying missing ones.

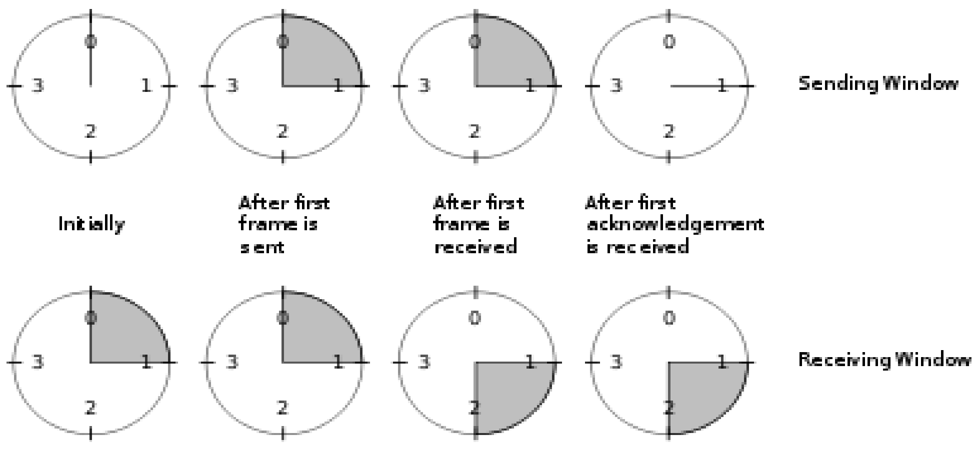
Sliding window is used by most connection oriented network protocol, among others, the Point-to-Point protocol (PPP) which many people use to establish their home PC as temporary Internet node via a phone-line connection to an existing node.

1.3.2 Working of Sliding window Protocol

Sliding Window Protocols assumes two-way communication (full duplex). It uses two types of frames:

1. Data
2. Ack (sequence number of last correctly received frame)

The basic idea of sliding window protocol is that both sender and receiver keep a ``window'' of acknowledgment. The sender keeps the value of expected acknowledgment; while the receiver keeps the value of expected receiving frame. When it receives an acknowledgment from the receiver, the sender advances the window.



A sliding window with a 2-bit sequence, of size 1

In sliding window method, multiple frames are sent by sender at a time before needing an acknowledgment. The term "window" on the transmitter side represents the logical boundary of the total number of packets yet to be acknowledged by the receiver. The receiver informs the transmitter in each acknowledgment packet the current maximum receiver buffer size (window boundary). The TCP header uses a 16 bit field to report the receive window size to the sender. Therefore, the largest window that can be used is 2^{16} = 64 kilobytes.

Sliding windows usually start out with a given size, however, more sophisticated protocols will dynamically adapt the window size, trying to find an agreed-upon size between sender and receiver.

The **characteristics of sliding windows u**sed at the sender and receiver usually involve

a. error correction (by retransmission),
b. flow control and
c. message ordering by sender (FIFO).

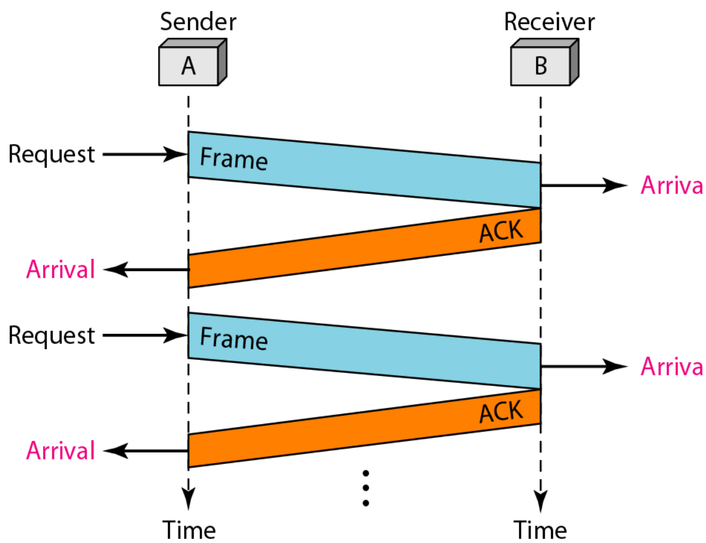
1.3.3 Examples

Stop-And-Wait

One Bit Sliding Window Protocol

One bit sliding window protocol is also called Stop-And-Wait protocol. In this protocol, the sender sends out one frame, waits for acknowledgment before sending next frame, thus the name Stop-And-Wait.

Problem with Stop-And-Wait protocol is that it is very inefficient. At any one moment, only in frame is in transition. The sender will have to wait at least one round trip time before sending next. The waiting can be long for a slow network such as satellite link.



Stop-And-Wait ARQ

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires. Lost frames are more difficult to handle than corrupted ones. The received frame could be the correct one, or a duplicate, or a frame out of order.

The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated. The completed and lost frames need to be present in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field.

Sequence Number

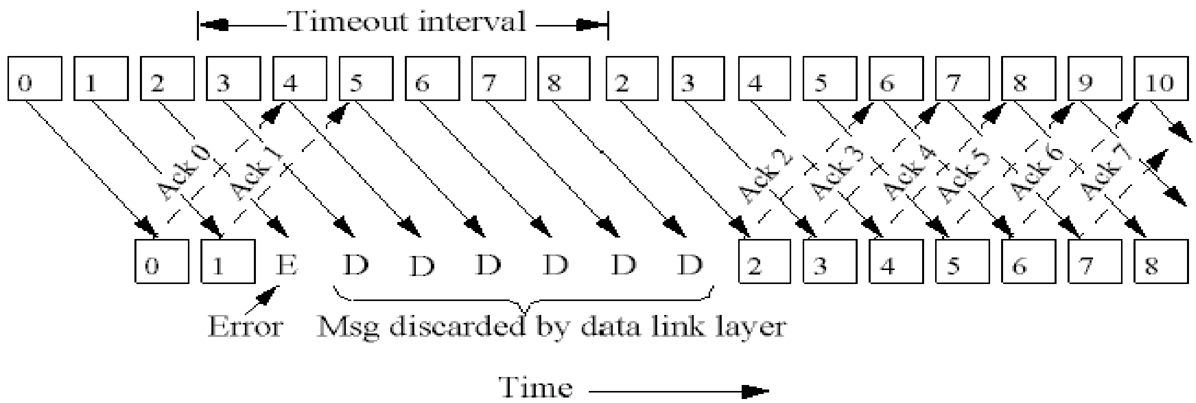
- A field is added to the data frame to hold the sequence number of that frame.
- The sequence numbers of course can wrap around. For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to 2^m - 1, and then are repeated.
- Let us reason out the range of sequence numbers we need. Assume we have used x as a sequence number; we only need to use x + 1 after that. There is no need for x + 2

Acknowledgement Number

- The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver.
- For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next).
- If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

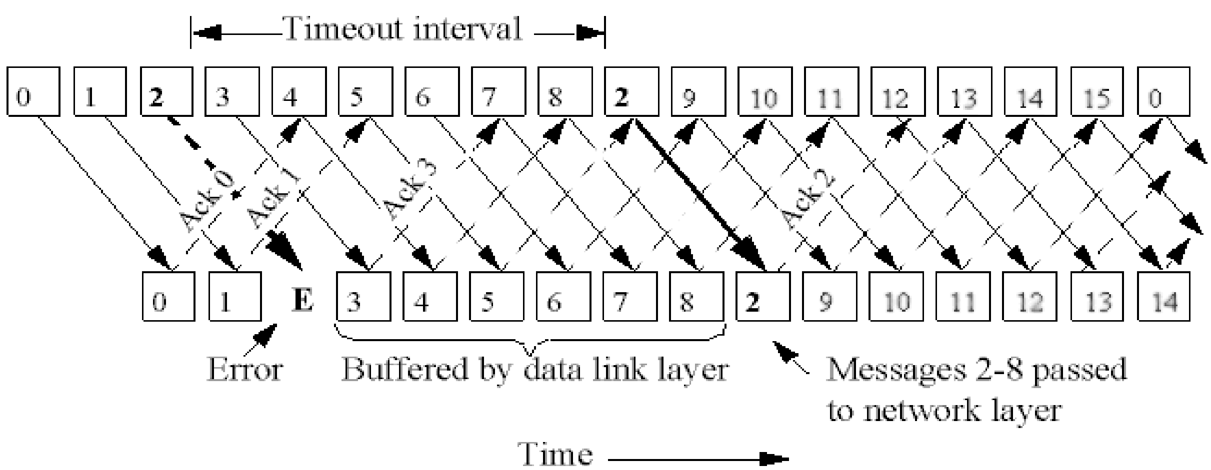
Go-Back-n

If there is one frame k missing, the receiver simply discard all subsequent frames k+1, k+2, ..., sending no acknowledgments. So the sender will retransmit frames from k onwards. Figure 3-15(a) on page 208. This effectively sets the receiver window size to be 1. This can be a waste of bandwidth.



Selective Repeat

Another strategy is to re-send only the ones that are actually lost or damaged. The receiver buffers all the frames after the lost one. When the sender finally noticed the problem (e.g. no ack for the lost frame is received within time-out limit), the sender retransmits the frame in question.



1.5 Assignment Question:

1. What are the functions of Data Link Layer?
2. Explain in detail Sliding window protocol?
3. Explain Selective Repeat Protocol & Go Back N ARQ Protocol?
4. Compare Sliding Wondow, Selective Repeat & Go Back N ARQ Protocol?

Conclusion:

Hence we Studied that how Sliding window protocol works in Data link layer.

Assignment 6 : Group B (Unit III & IV)

Problem Definition:
Write a program to demonstrate Sub-netting and find subnet masks.

PROBLEM STATEMENT:

Sub-netting and find subnet masks.

6.1 Prerequisite:

1. IP Address Classes
2. Classless & Classful IP Addressing

6.2 LearningObjectives:

1. Understand the concept Subnetting.
2. Understand the Concept of Supernet.

6.3 Theory

6.4.1 Introduction

IPv4 or Internet Protocol version 4, address is a 32-bit string of numbers separated by periods. It uniquely identifies a network interface in a device. IP is a part of the TCP/IP (Transmission Control Protocol/Internet Protocol) suite, where IP is the principal set of rules for communication on the Internet.

IPv4 Address Format

IPv4 addresses are expressed as a set of four numbers in decimal format, and each set is separated by a dot. Thus, the term 'dotted decimal format.' Each set is called an 'octet' because a set is composed of 8 bits. The figure below shows the binary format of each octet in the 192.168.10.100 IP address:

Format	1^{st} Octet	2^{nd} Octet	3^{rd} Octet	4^{th} Octet
Dotted Decimal	192	168	10	100
Binary	1100 0000	1010 1000	0000 1010	0110 0100

A number in an octet can range from 0 to 255. Therefore, the full IPv4 address space goes from 0.0.0.0 to 255.255.255.255. The IPv4 address has two parts, the network part and the host part. A subnet mask is used to identify these parts.

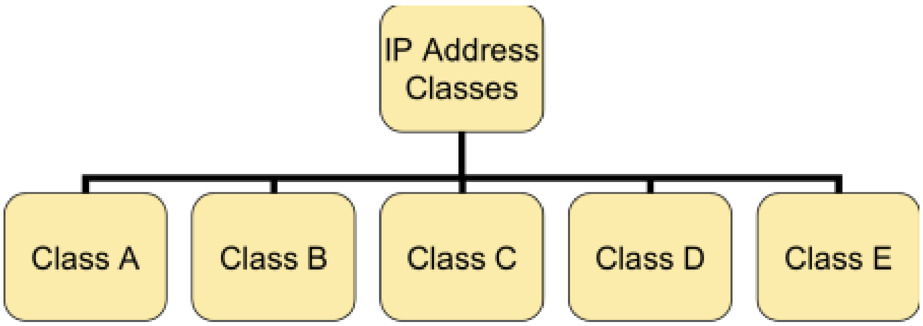
IPv4 Address Allocation

The Internet Protocol address can be allocated to hosts or interfaces either manually or dynamically.

- **Static** – static IP address is set manually on the device. It is best practice to set static IP addresses on network devices, such as routers and switches, and on servers as well.
- **Dynamic** – dynamic IP address can be automatically allocated to a device via Dynamic Host Configuration Protocol (DHCP). Dynamic IP addresses are best to be used on end devices, such as PCs.

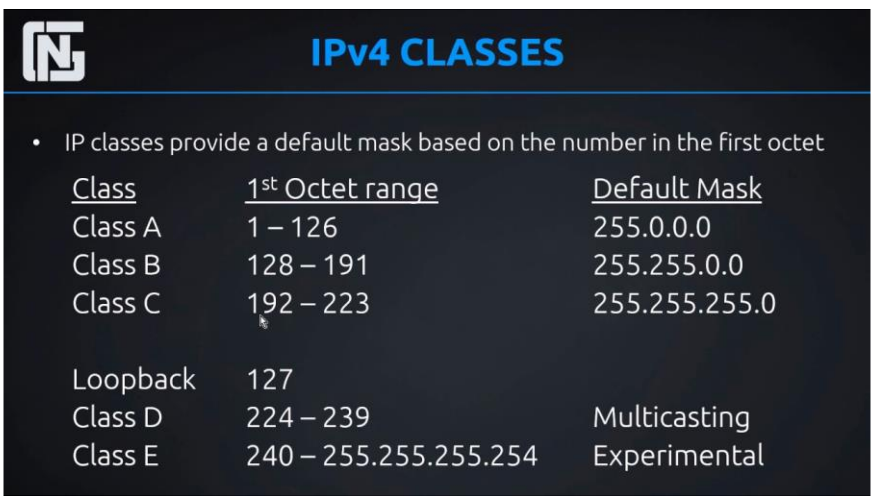
6.4.2Classes of Address

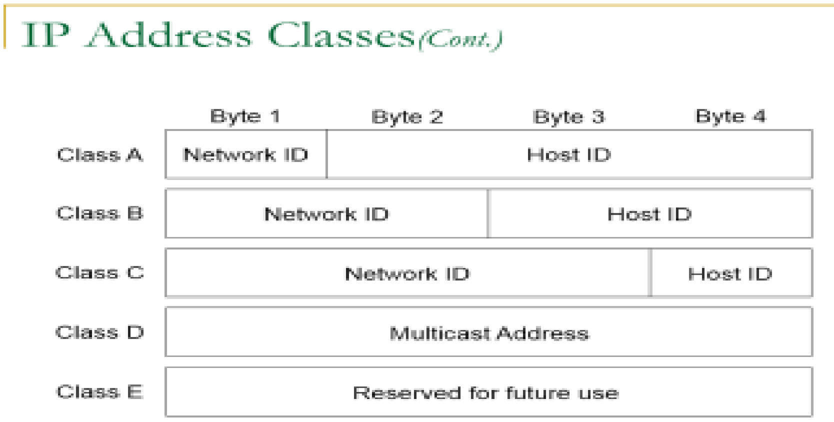
- **IP address structure consists of two addresses, Network and Host**
- **IP address is divided into five classes**



IP Address Classes

The 5 IP classes are split up based on the value in the 1st octet:





Examples of IP Address

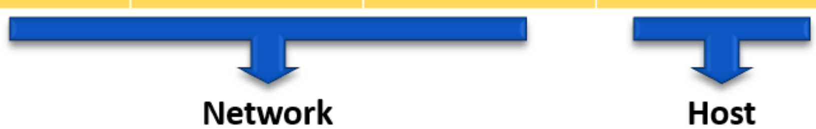
- 14.23.120.8 - The first byte of the address represents 14 which lies between 0 and 127, hence Class A address.
- 134.11.78.56 - The first byte of address is 134 which lies between 128 and 191 hence the address belongs to Class B.
- 193.14.56.22 - As first byte is 193 which is between 192 and 223, hence the address belongs to Class C.

Subnet Mask

An IP address has 2 parts:

- The Network identification.
- The Host identification.

Format	1st Octet	2nd Octet	3rd Octet	4th Octet
Dotted Decimal	192	168	10	100
Binary	1100 0000	1010 1000	0000 1010	0110 0100



- IP address is divided into two parts: network and host parts.
  For example, an IP class A address consists of 8 bits identifying the network and 24 bits identifying the host. This is because the default subnet mask for a class A IP address is 8.

Computers use it to determine the network part and the host part of an address. The 1s in the subnet mask represent a network part, the 0s a host part.

- Computers works only with bits. The math used to determine a network range is binary AND.

INPUT 1	INPUT 2	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

Let’s say that we have the IP address of 10.0.0.1 with the default subnet mask of 8 bits 255.0.0.0).
First, we need to convert the IP address to binary:

IP address: 10.0.0.1 = 00001010.00000000.00000000.00000001
Subnet mask 255.0.0.0 = 11111111.00000000.00000000.0000000
Computers then use the AND operation to determine the network number:

00001010.00000000.00000000.00000001 = 10.0.0.1

11111111.00000000.00000000.00000000 = 255.0.0.0

00001010.00000000.00000000.00000000 = 10.0.0.0

The computer can then determine the size of the network. Only IP addresses that begin with 10 will be in the same network. So, in this case, the range of addresses in this network is 10.0.0.0 – 10.255.255.255.

Slash Notation

Aside from the dotted decimal format, we can also write the subnet mask in slash notation. It is a slash ‘/’ then followed by the subnet mask bits. To determine the slash notation of the subnet mask, convert the dotted decimal format into binary, count the series of 1s, and add a slash on the start.

For example, we have the dotted decimal subnet mask of 255.0.0.0. In binary, it is 11111111.00000000.00000000.0000000. The number of succeeding 1s are 8, therefore the slash notation of 255.0.0.0 is /8.

Frequently, the Network & Host portions of the address need to be separately extracted.

- In most cases, if you know the address class, it’s easy to separate the 2 portions.

- Specifies part of IP address used to identify a subnetwork.
- Subnet mask when logically ANDed with IP address provides 32-bit network address

Default Mask:

- Has predetermined number of 1s
- Class A, B and C contains 1s in network ID fields for default subnet mask.

Computer Network laboratory (2015) Pattern TE Computer

-Address Class	**Default Mask (in Binary)**
Class A	**11111111.00000000.00000000.00000000**
Class B	**11111111.11111111.00000000.00000000**
Class C	**11111111.11111111.11111111.00000000**

Default Standard Subnet Masks

There are default standard subnet masks for Class A, B and C addresses:

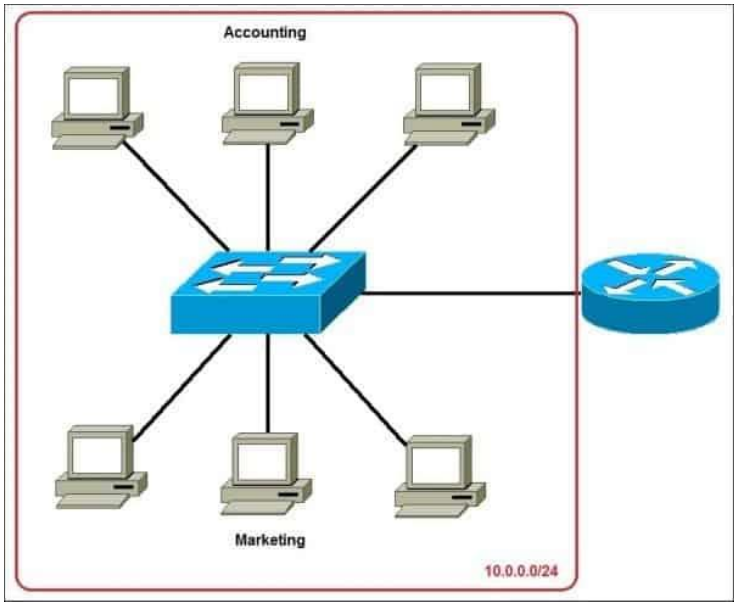
Default Subnet Masks	
Address Class	Subnet Mask
Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

IP SUBNETTING

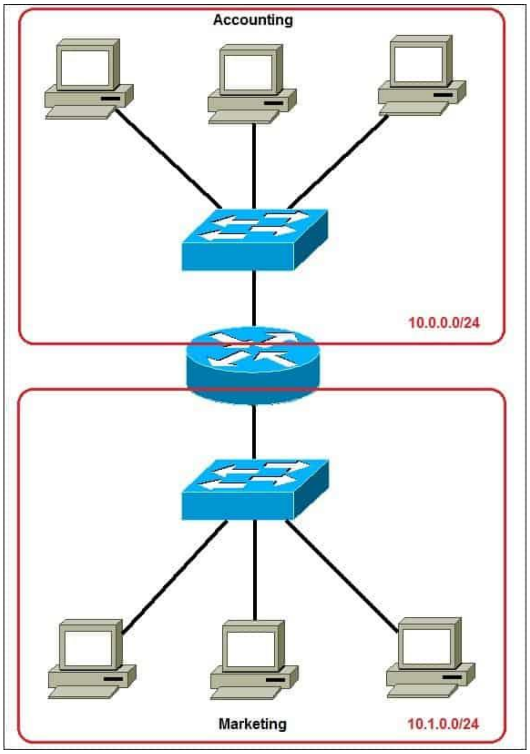
Subnetting is the practice of dividing a network into two or more smaller networks. It increases routing efficiency, enhances the security of the network and reduces the size of the broadcast domain.

IP Subnetting:

- Allows you to divide a network into smaller sub-networks
- Each subnet has its own sub-network address
- Subnet can be created within Class A, B, or C based networks



In the picture above we have one huge network: **10.0.0.0/24**.



Now, two subnets were created for different departments: **10.0.0.0/24** for Accounting and **10.1.0.0/24** for Marketing. Devices in each subnet are now in a different broadcast domain. This will reduce the amount of traffic flowing on the network and allow us to implement packet filtering on the router.

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 subnets
- 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

Subnetting Principle:

- 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 form one node to the other .

Create subnets

There are a couple of ways to create subnets. In this article we will subnet a class C address **192.168.0.0** that, by default, has **24** subnet bits and **8** host bits.

Before we start subnetting, we have to ask ourselves these two questions:

1. **How many subnets do we need?**

   2^x = number of subnets. **x** is the number of 1s in the subnet mask. With 1 subnet bit, we can have 2^1 or 2 subnets. With 2 bits, 2^2 or 4 subnets, with 3 bits, 2^3 or 8 subnets, etc.

2. **How many hosts per subnet do we need?**

   $2^y - 2$ = number of hosts per subnet. **y** is the number of 0s in the subnet mask.

■ Although the 2 formulas look identical, the key is to remember the number you're trying to calculate, hosts or subnets.

■ Eg., suppose you are asked to determine the number of subnets available & the number of hosts available on each subnet on the network 192.168.1.0

■ Using the subnet & hosts formulas, the answers are easily calculated. Of course, you must know your powers of 2 to calculate the answers

Example:

■ Host IP Address: 192.168.10.0/25

■ Subnet Mask: 255.255.255.128 (or /26)

Given the following find the number of Network,find the number of IP Address on each Network, findthe number of Host of each network,what is the broadcast address of each subnet & valid, What are the valid subnets?

Step 1: How to find the number of Network?

2^x = number of subnets. **x** is the number of 1s in the subnet mask

$2^1 = 2$

We can create only 2 subnet network

Step 2: How to find the number of IP address on each Network?

2^y = number of subnets. **y** indicates total number of host bits.

$2^7 = 128$

On each network you will have 128 IP address

Step 3: How to find the number of Host on each Network?

$2^y - 2$= number of subnets. **y** indicates total number of host bits.

$2^7 - 2 = 126$

You will have total 126 Host IP address on each network.

Step 3: What is the broadcast address of each subnet & Network Address?

Network1

192.168.10.0-→ Network Address
192.168.10.1
192.168.10.2
to
192.168.10.126
192.168.10.127--→ Broadcast Address

Network2

192.168.10.128-→ Network Address
192.168.10.1
192.168.10.2
to
192.168.10.126
192.168.10.255--→ Broadcast Address

AssignmentQuestions:

1.What is subnetting?
2. What is the importance of subnetting?.
3.How to find the first and last address of subnet?
4.What is the difference between supernetting and subnetting?
5.What is classful and classless ip address?
6. What is subnet mask? How to find subnet mask?

Conclusion:

Hence we have studied Subnetting and the importance of subnetting.

Title: Write a program to implement link state /Distance vector routing protocol to find suitable path for transmission.

Objective: To understand working of Distance vector routing protocol.

Prerequisite:

1. Shortest path finding
2. Classification of routing Algorithm

Learning Objectives:

1. Understand the concept Distance vector routing
2. Understand the Concept of Routing Algorithms

Theory:

Introduction:

A distance-vector routing (DVR) protocol requires that a router inform its neighbors of topology changes periodically. Historically known as the old ARPANET routing algorithm (or known as Bellman-Ford algorithm).

Bellman Ford Basics – Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances, based on a chosen metric, are computed using information from the neighbors' distance vectors.

Information kept by DV router -

- Each router has an ID

Associated with each link connected to a router,

- There is a link cost (static or dynamic).
- Intermediate hops

Distance Vector Table Initialization -

- Distance to itself = 0
- Distance to ALL other routers = infinity number.

Distance vector Algorithm:

1. A router transmits its distance vector to each of its neighbors in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
   a. It receives a distance vector from a neighbor containing different information than before.
   b. It discovers that a link to a neighbor has gone down.

The DV calculation is based on minimizing the cost to each destination

$D_x(y)$ = Estimate of least cost from x to y

$C(x,v)$ = Node x knows cost to each neighbor v

$D_x = [D_x(y): y \in N]$ = Node x maintains distance vector

Node x also maintains its neighbors' distance vectors
– For each neighbor v, x maintains $D_v = [D_v(y): y \in N]$

Distance Vector Routing:

- It is a dynamic routing algorithm in which each router computes distance between itself and each possible destination i.e. its immediate neighbors.
- The router share its knowledge about the whole network to its neighbors and accordingly updates table based on its neighbors.
- The sharing of information with the neighbors takes place at regular intervals.
- It makes use of Bellman Ford Algorithm for making routing tables.
- Problems – Count to infinity problem which can be solved by splitting horizon.
  – Good news spread fast and bad news spread slowly.
  – Persistent looping problem i.e. loop will be there forever.

Link State Routing:

- It is a dynamic routing algorithm in which each router shares knowledge of its neighbors with every other router in the network.
- A router sends its information about its neighbors only to all the routers through flooding.
- Information sharing takes place only whenever there is a change.
- It makes use of Dijkastra's Algorithm for making routing tables.
- Problems – Heavy traffic due to flooding of packets.
  – Flooding can result in infinite looping which can be solved by using Time to live (TTL) field.

Conclusion: Hence we have studied distance vector algorithm to find suitable path for transmission.

Assignment 8 : Group B (Unit III & IV)

Problem Definition:

- **Problem Definition :Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF/BGP**

- **Prerequisite**

1. Routing Protocols.
2. Basics of Packet Tracer.

- **Learning Objectives**:

1. To Understand Simulation Tool.
2. Should Able to Configure Routing Protocols

1.3 Theory:

1.3.1 Introduction

ROUTING INFORMATION PROTOCOL:

The **Routing Information Protocol** (**RIP**) is one of the oldest distance-vector routing protocols which employ the hop count as a routing metric. RIP prevents routing loops by implementing a limit on the number of hops allowed in a path from source to destination. The maximum number of hops allowed for RIP is 15, which limits the size of networks that RIP can support. A hop count of 16 is considered an infinite distance and the route is considered unreachable. RIP implements the split horizon, route poisoning and holddown mechanisms to prevent incorrect routing information from being propagated.

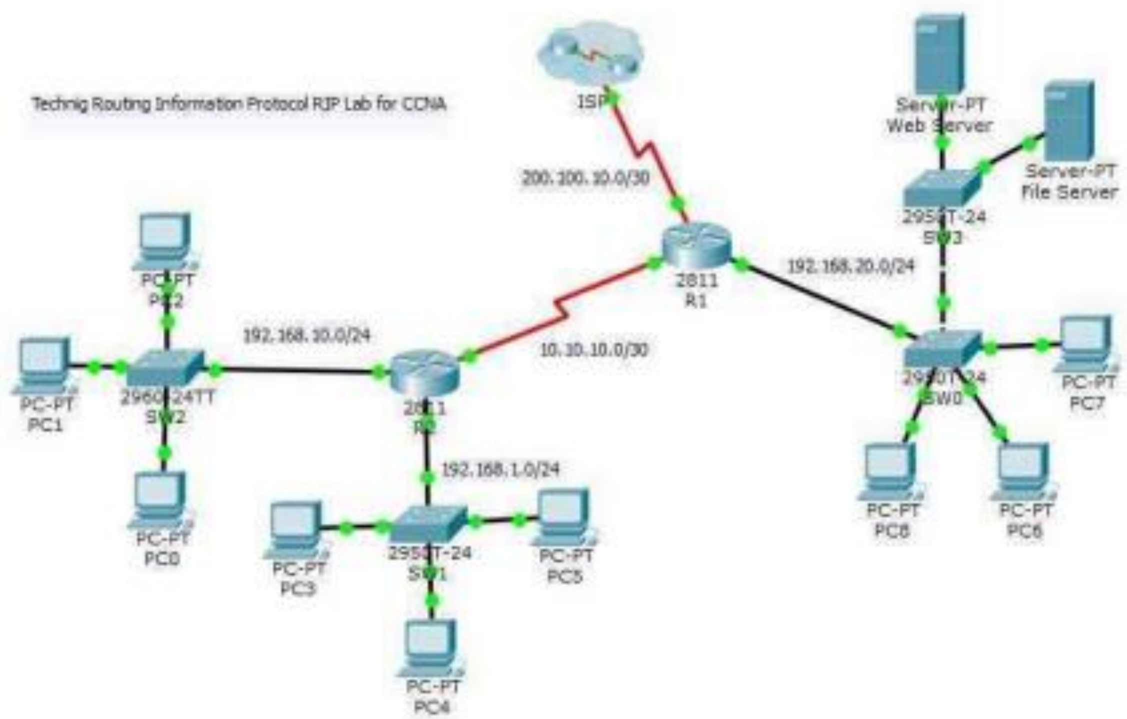
- Originally, each RIP router transmitted full updates every 30 seconds. In the early deployments, routing tables were small enough that the traffic was not significant. As networks grew in size, however, it became evident there could be a massive traffic burst every 30 seconds, even if the routers had been initialized at random times. It was thought, as a result of random initialization, the routing updates would spread out in time, but this was not true in practice. Sally Floyd and Van Jacobson showed in 1994 that, without slight randomization of the update timer, the timers synchronized over time.

- **OPEN SHORTEST PATH FIRST: (OSPF)**

- OSPF is an interior gateway protocol (IGP) for routing Internet Protocol (IP) packets solely within a single routing domain, such as an autonomous system. It gathers link state information from available routers and constructs a topology map of the network. The topology is presented as a routing table to the Internet layer which routes packets based solely on their destination IP address.

- Open Shortest Path First (OSPF) is a routing protocol for Internet Protocol (IP) networks. It uses a link state routing (LSR) algorithm and falls into the group of interior gateway protocols (IGPs), operating within a single autonomous system (AS).

CONFIGURE ROUTING INFORMATION PROTOCOL (RIP)



Open the router 1 (**R1**) which is the main router connected to ISP router. Do the following command for RIP Routing.

R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 200.100.10.0

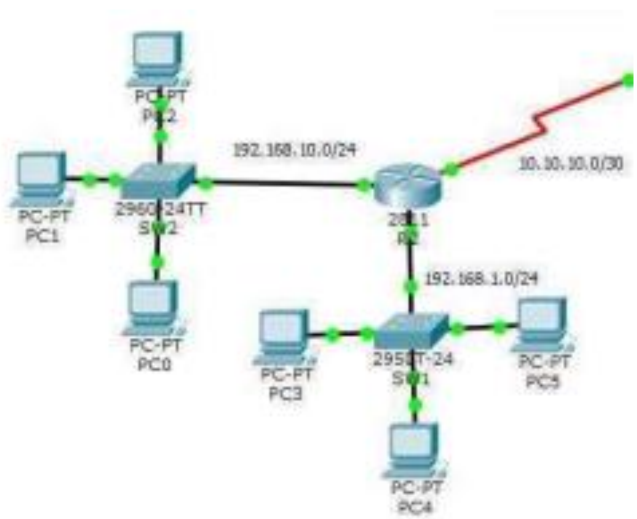
DHCP message types:

R1(config-router)#network 192.168.20.0
R1(config-router)#network 10.10.10.0
R1(config-router)#

After enabling router with enable command then go to privileged mode with configure terminal command. Now with router rip command, enable routing for all routers. The version 2 Command, configure routing information protocol with version two. And next set all network id like the above network command. I have set all three network which connect directly to R1.

Now go to router R2 and configure routing protocol the same as router R1. On router 2 you must assign the network ids of all connected network the R2.

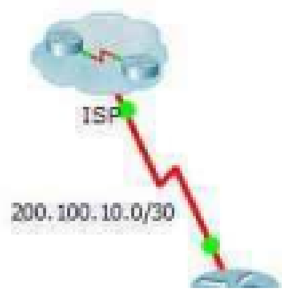
```
R2>enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 10.10.10.0
R2(config-router)#network 192.168.10.0
R2(config-router)#network 192.168.1.0
R2(config-router)#
```



For ISP router, just enter the network id 200.100.10.0, because only one network connected to ISP router.

```
ISP>enable
ISP#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ISP(config)#router rip
```

ISP(config-router)#version 2
ISP(config-router)#network 200.100.10.0
ISP(config-router)#



- **CONFIGURE OSPF ROUTING PROTOCOL**

In the router R1 configure OSFP routing with Router ospf command.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#network 20.10.10.0 0.0.0.3 area 0
R1(config-router)#network 10.10.10.0 0.0.0.3 area 0
R1(config-router)#network 10.10.10.4 0.0.0.3 area 0
R1(config-router)#
```

The router OSPF command is enable OSPF routing on the router, and the 1 before OSFP is the process ID of the OSFP Protocol. You can set different process id from “1-65535” for each router.

The network command with network ID “network 20.10.10.0” is the network identifier, and the “ 0.0.0.3″ is the wildcard mask of 20.10.10.0 network. Wildcard mask determine which interfaces to advertise, because OSPF advertise interfaces, not networks.

Now go to Router R3 and configure with the following commands.

*R3>*enable

*R3#*configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

*R3(config)#*router ospf 1

*R3(config-router)#*network 192.168.1.0 0.0.0.255 area 0

*R3(config-router)#*network 10.10.10.0 0.0.0.3 area 0

Don? So do the following for router R2.

*R2>*enable

*R2#*configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

*R2(config)#*router ospf 1

*R2(config-router)#*network 192.168.10.0 0.0.0.255 area 0

*R2(config-router)#*network 10.10.10.4 0.0.0.3 area 0

OK, OSPF routing configuration has been finished successfully, now test your network whether they can ping with each other or not.

Assignment Questions:

1. Explain Routing Protocols in Details?
2. Draw Classification Diagram of Routing Protocols?
3. Explain Packet Tracer Routing Commands?
4. Difference between RIP & OSPF?
5. Key differences between RIPv1 and RIPv2?

Conclusion:

Hence we have studied Packet Tracer Properly.

Assignment 9 : Group B (Unit III & IV)

Problem Definition:

Problem Definition:Write a program using TCP socket for wired network for following a. Say Hello to Each other b. File transfer c. Calculator

1.1Prerequisite:

a) Socket Header b) Network Programming c) Ports

LearningObjectives:

1. To understand Work of Socket
2. Different methods associated with Client & Server Socket

NewConcepts:

1. Client Server Communication
2. Port Address

1.3 Theory:

1.3.1 Introduction

TCP:

The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol. It provides host-to-host connectivity at the Transport Layer of the Internet model.

The client server model

Most inter-process communication uses the client server model. These terms refer to the two processes which will be communicating with each other. One of the two processes, the client, connects to the other process, the server, typically to make a request for information. A socket is one end of an inter-process communication channel. The two processes each establish their own socket.

The steps involved in establishing a socket on the client side are as follows:

1. Create a socket with the socket( ) system call

2. Connect the socket to the address of the server using the connect( ) system call

3. Send and receive data. There are a number of ways to do this, but the simplest is to use the read ( ) and write ( ) system calls.

The steps involved in establishing a socket on the server side are as follows:

1. Create a socket with the socket ( ) system call
2. Bind the socket to an address using the bind ( ) system call. For a server socket on the Internet, an address consists of a port number on the host machine.
3. Listen for connections with the listen ( ) system call.
4. Accept a connection with the accept ( ) system call. This call typically blocks until a client connects with the server.
5. Send and receive data

Algorithm: Server Program

1. Open the Server Socket: ServerSocket server = new ServerSocket(PORT);
2. Wait for the Client Request: Socket client = server.accept();
3. Create I/O streams for communicating to the client
DataInputStream is = new DataInputStream(client.getInputStream()); DataOutputStreamos = new DataOutputStream(client.getOutputStream());
4. Perform communication with client Receive from client: String line = is.readLine();
5. Send to client: os.writeBytes("Hello\n")
6. Close socket: client.close();

Algorithm: Client Program

1. Create a Socket Object: Socket client = new Socket(server, port_id);

2. Create I/O streams for communicating with the server. is = new DataInputStream(client.getInputStream()); os = new DataOutputStream(client.getOutputStream());

3. Perform I/O or communication with the server: Receive data from the server: String line = is.readLine(); Send data to the server: os.writeBytes("Hello\n");

4. Close the socket when done client.close();

TYPES OF SOCKETS

Socket Types

There are four types of sockets available to the users. The first two are most commonly used and the last two are rarely used.

Processes are presumed to communicate only between sockets of the same type but there is no restriction that prevents communication between sockets of different types.

- Stream Sockets − Delivery in a networked environment is guaranteed. If you send through the stream socket three items "A, B, C", they will arrive in the same order − "A, B, C". These sockets use TCP (Transmission Control Protocol) for data transmission. If delivery is impossible, the sender receives an error indicator. Data records do not have any boundaries.

- Datagram Sockets − Delivery in a networked environment is not guaranteed. They're connectionless because you don't need to have an open connection as in Stream Sockets − you build a packet with the destination information and send it out. They use UDP (User Datagram Protocol).

- Raw Sockets − These provide users access to the underlying communication protocols, which support socket abstractions. These sockets are normally datagram oriented, though their exact characteristics are dependent on the interface provided by the protocol. Raw sockets are not intended for the general user; they have been provided mainly for those interested in developing new communication protocols, or for gaining access to some of the more cryptic facilities of an existing protocol.

Here is the description of the parameters −

- **socket_family −** This is either AF_UNIX or AF_INET, as explained earlier.
- **socket_type −** This is either SOCK_STREAM or SOCK_DGRAM.
- **protocol −** This is usually left out, defaulting to 0.

Once you have socket object, then you can use required functions to create your client or server program. Following is the list of functions required –

SERVER SOCKET METHODS

Sr.No.	Method & Description
1	**s.bind()**This method binds address (hostname, port number pair) to socket.
2	**s.listen()**This method sets up and start TCP listener.
3	**s.accept()**This passively accept TCP client connection, waiting until connection arrives (blocking).

CLIENT SOCKET METHODS

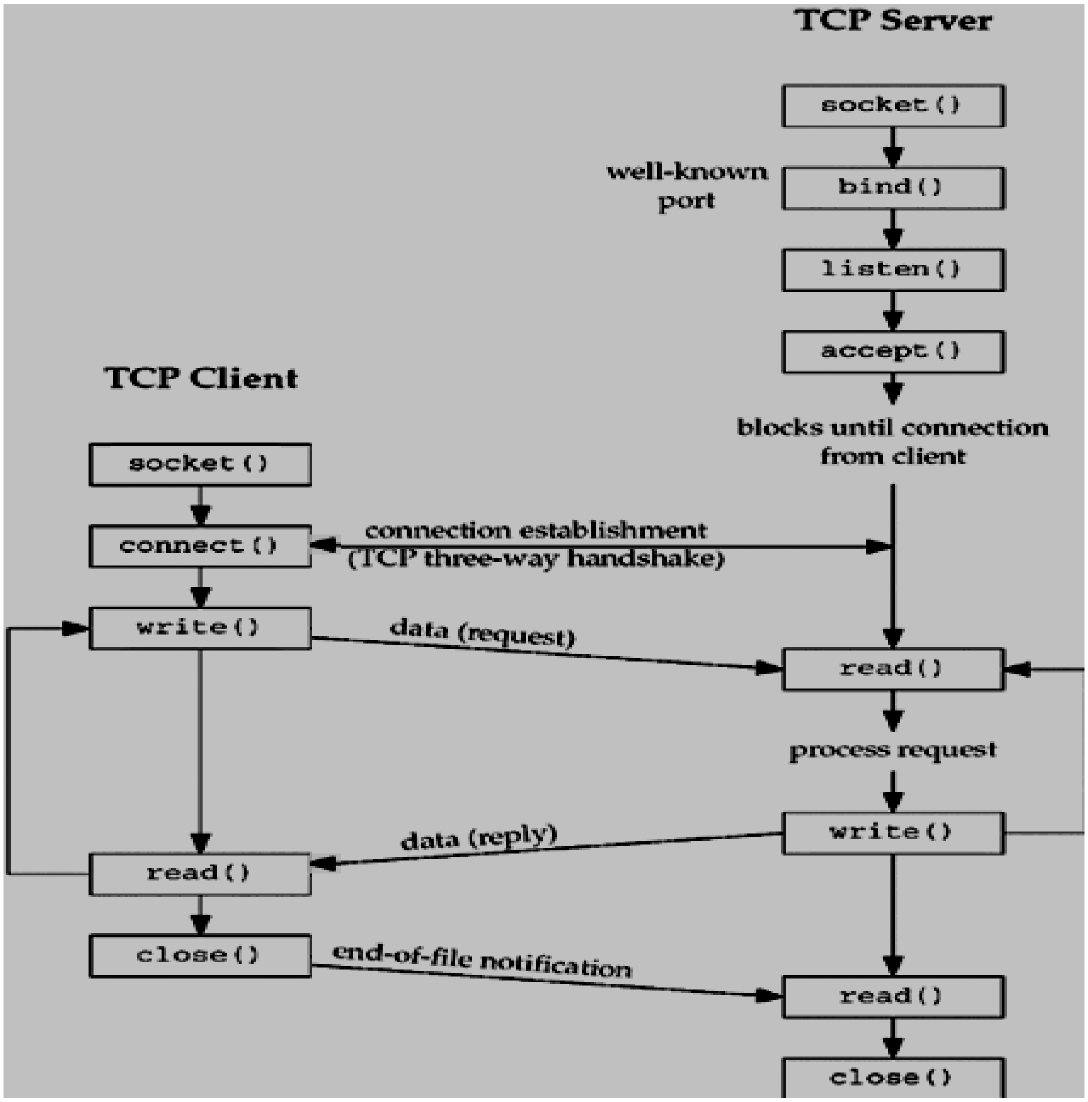
Sr.No.	Method & Description

1	**s.connect()**This method actively initiates TCP server connection.

GENERAL SOCKET METHODS

Sr.No.	Method & Description
1	**s.recv()**This method receives TCP message
2	**s.send()**This method transmits TCP message
3	**s.recvfrom()**This method receives UDP message
4	**s.sendto()**This method transmits UDP message
5	**s.close()**This method closes socket
6	**socket.gethostname()**Returns the hostname.

Methods Associated with Socket:The following diagram shows the complete Client and Server interaction −



Questions:

1. What is a socket? Explain different types of Sockets
2. What is the difference between a connection-less and a connection-oriented communication system? How are they implemented?

3. Why is the port number required?

4. How is the socket programming in linux different from that in windows?

5.What are the Methods Associate with Server Socket ?

6. What are the methods associated with Client Socket ?

CONCLUSION: Thus we have successfully implemented the socket programming for TCP

Assignment 10 : Group B (Unit III & IV)

Problem Definition:

Problem Definition:Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines

1.1Prerequisite:

a) Socket Header b) Network Programming c) Ports

LearningObjectives:

1. To understand Work of Socket
2. Different methods associated with Client & Server Socket

NewConcepts:

1. Client Server Communication
2. Port Address

1.3 Theory:

1.3.1 Introduction

What is UDP?

UDP is a connectionless and unreliable transport protocol.The two ports serve to identify the end points within the source and destination machines. User Datagram Protocol is used, in place of TCP, when a reliable delivery is not required.However, UDP is never used to send important data such as web-pages, database information, etc. Streaming media such as video,audio and others use UDP because it offers speed.

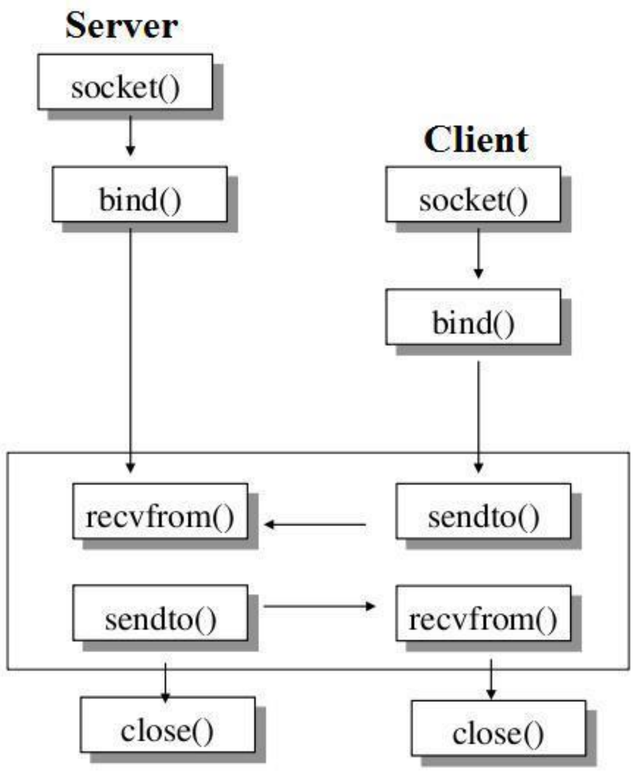
Why UDP is faster than TCP?

The reason UDP is faster than TCP is because there is no form of flow control. No error checking,error correction, or acknowledgment is done by UDP.UDP is only concerned with speed. So when, the data sent over the Internet is affected by collisions, and errors will be present. UDP packet's called as user datagrams with 8 bytes header. A format of user datagrams is shown in figur 3. In the user datagrams first 8 bytes contains header information and the remaining bytes contains data.

LINUX SOCKET PROGRAMMING:

The Berkeley socket interface, an API, allows communications between hosts or between processes on one computer, using the concept of a socket. It can work with many different I/O devices and drivers, although support for these depends on the operating-system implementation. This interface implementation is implicit for TCP/IP, and it is therefore one of the fundamental technologies underlying the Internet. It was first developed at the University of California, Berkeley for use on Unix systems. All modern operating systems now have some implementation of the Berkeley socket interface, as it has become the standard interface for connecting to the Internet. Programmers can make the socket interfaces accessible at three different levels, most powerfully and fundamentally at the RAW socket level. Very few applications need the degree of control over outgoing communications that this provides, so RAW sockets support was intended to be available only on computers used for developing Internet-related technologies. In recent years, most operating systems have implemented support for it anyway, including Windows XP. The header files: The Berkeley socket development library has many associated header files. They include: Definitions for the most basic of socket structures with the BSD socket API Basic data types associated with structures within the BSD socket API Definitions for the socketaddr_in{} and other base data structures.

Connectionless Protocol



The header files:

The Berkeley socket development library has many associated header files.

They include: ***<sys/socket.h>***

Definitions for the most basic of socket structures with the BSD

socket API *<sys/socket.h>*

Basic data types associated with structures within the BSD socket API <sys/types.h>

Socket API<sys/types.h>

Definitions for the socketaddr_in{} and other base data structures

<sys/un.h>

Definitions and data type declarations for SOCK_UNIX streams

UDP: UDP consists of a connectionless protocol with no guarantee of delivery. UDP packets may arrive out of order, become duplicated and arrive more than once, or even not arrive at all. Due to the minimal guarantees involved, UDP has considerably less overhead than TCP. Being connectionless means that there is no concept of a stream or connection between two hosts, instead, data arrives in datagrams. UDP address space, the space of UDP port numbers (in ISO terminology, the TSAPs), is completely disjoint from that of TCP ports. Server: Code may set up a UDP server on port 7654 as follows:

```
sock = socket(PF_INET,SOCK_DGRAM,0);
sa.sin_addr.s_addr = INADDR_ANY;
 sa.sin_port = htons(7654);
bound = bind(sock,(struct sockaddr *)&sa, sizeof(struct sockaddr));
 if (bound < 0) fprintf(stderr, "bind(): %s\n",strerror(errno)); listen(sock,3);
```

bind() binds the socket to an address/port pair. listen() sets the length of the new connections queue.

```
 while (1)
 {
 printf ("recv test....\n");
recsize = recvfrom(sock, (void *)hz, 100, 0, (struct sockaddr *)&sa, fromlen);
printf ("recsize: %d\n ",recsize);
 if (recsize < 0)
fprintf(stderr, "%s\n", strerror(errno));
sleep(1);
printf("datagram: %s\n",hz);
 }
```

This infinite loop receives any UDP datagrams to port 7654 using recvfrom(). It uses the parameters: l socket l pointer to buffer for data l size of buffer l flags (same as in read or other receive socket function)

Client: A simple demo to send an UDP packet containing "Hello World!" to address 127.0.0.1, port 7654 might look like this:

```
#include #include #include #include #include
#include int main(int argc, char *argv[])
{
int sock; struct sockaddr_in sa;
int bytes_sent, buffer_length;
char buffer[200];
sprintf(buffer, "Hello World!");
buffer_length = strlen(buffer) + 1;
 sock = socket(PF_INET, SOCK_DGRAM, 0);
 sa.sin_family = AF_INET;
 sa.sin_addr.s_addr = htonl(0x7F000001);
sa.sin_port = htons(7654); bytes_sent = sendto(sock, buffer, buffer_length, 0, &sa,
sizeof(struct sockaddr_in) );
 if(bytes_sent < 0) printf("Error sending packet: %s\n", strerror(errno) );
 return 0;
 }
```

In this code, buffer provides a pointer to the data to send, and buffer_length specifies the size of the buffer contents. Typical UDP client code

- Create UDP socket to contact server (with a given hostname and service port number)
- Create UDP packet.
- Call send(packet), sending request to the server.
- Possibly call receive(packet) (if we need a reply).

Typical UDP Server code

- Create UDP socket listening to a well known port number.
- Create UDP packet buffer Call receive(packet) to get a request, noting the address of the client.
- Process request and send reply back with send(packet).

APPLICATION :

Socket programming is essential in developing any application over a network.

Conclusion: Thus we have studied Working of UDP Socket.

Questions:

1. Can multiple clients connect to same UDP socket?
2. Which socket is used in UDP?
3. How do you specify socket type for UDP?

CONCLUSION Thus we have successfully implemented the socket programming for TCP

Assignment - 11

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

Objective:

- To get the host name and IP address.
- Map the host name with IP address and Vice-versa

Learning Objectives:

- Understand what is Domain Name System and DNS lookup working.
- Understand what is DNS Structure and Hierarchy.

New Concepts:

- Name Server and Domain Name System.
- DNS lookup, Zone

Theory:

Need for DNS:

To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name. When the Internet was small, mapping was done using a host file. The host file had only two columns: name and address. Every host could store the host file on its disk and update it periodically from a master host file. When a program or a user wanted to map a name to an Address, the host consulted the host file and found the mapping.

Today, however, it is impossible to have one single host file to relate every address with a name and vice versa. The host file would be too large to store in every host. In addition, it would be impossible to update all the host files every time there is a change. One solution would be to store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping. But we know that this would create a huge amount of traffic on the Internet. Another solution, the one used today, is to divide this huge amount of information into smaller parts and store each part on a different computer. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS).

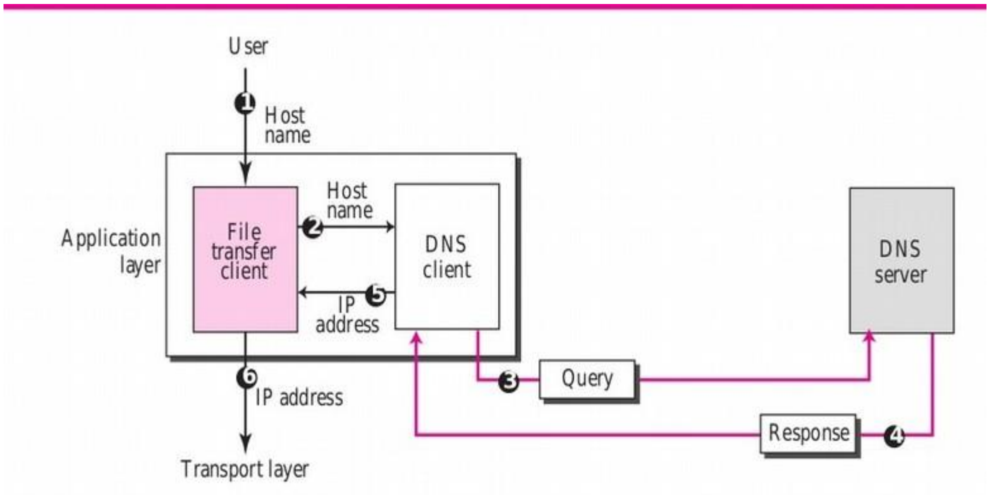


Figure 1: How TCP/IP uses a DNS client and a DNS server to map a name to an address; the reverse mapping is similar.

In Figure 1, a user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as forouzan.com. However, the TCP/IP suite needs the IP address of the file transfer server to makethe connection. The following six steps map the host name to an IP address.

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. We know that each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS client passes the IP address to the file transfer server.
6. The file transfer client now uses the received IP address to access the file transfer server.

NAME SPACE:

To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses. In other words, the names must be unique because the addresses are unique.

A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

Flat Name Space:

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section; if they do, it has no meaning. The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space;
In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on. In this case, the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.

Domain Name Space:
To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127 (see Figure 2).

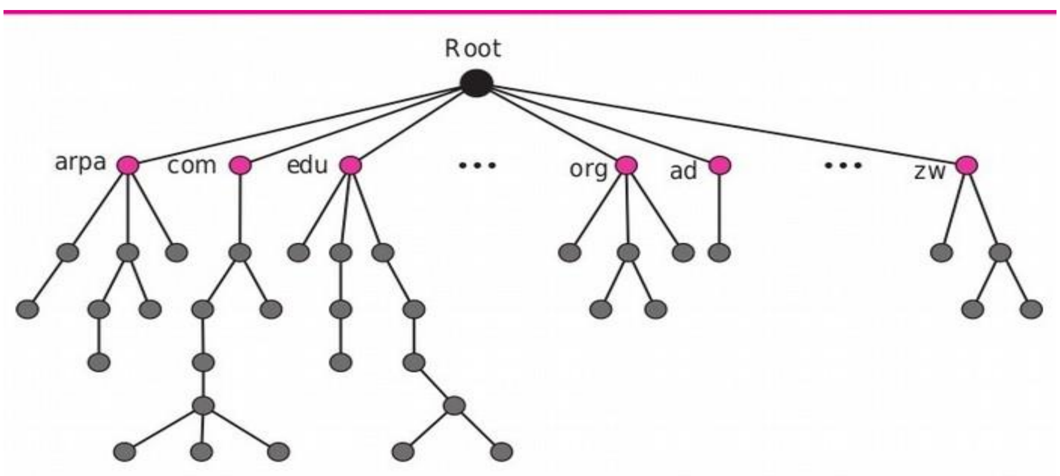


Figure 2: Domain Name Space

Label:
Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name:
Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the

label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

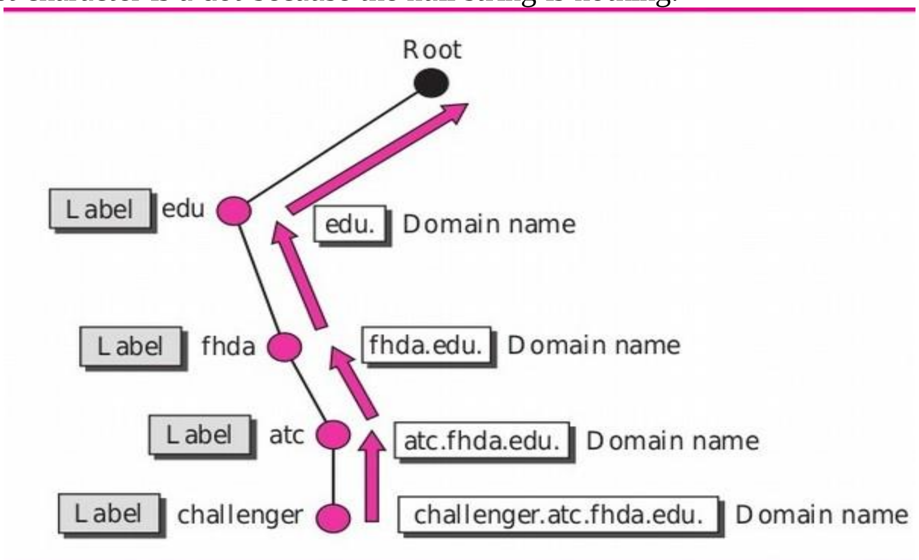


Figure 3 shows some domain names.

Domain:

A domain is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree. Figure 4 shows some domains. Note that a domain may itself be divided into domains (or subdomains as they are sometimes called).

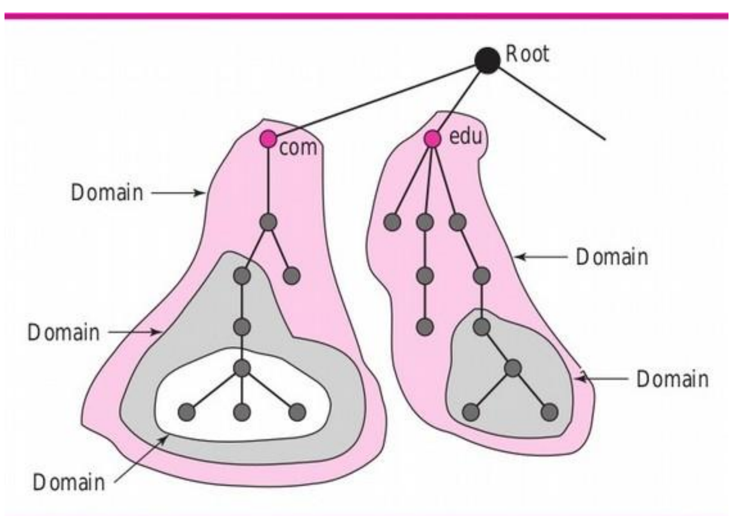


Figure 4 Domain

RESOLUTION:
Mapping a name to an address or an address to a name is called name-address resolution.

Resolver:
DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the Information. After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.

Mapping Names to Addresses:
Most of the time, the resolver gives a domain name to the server and asks for the corresponding address. In this case, the server checks the generic domains or the country domains to find the mapping. If the domain name is from the generic domains section, the resolver receives a domain name such as “chal.atc.fhda.edu.” The query is sent by the resolver to the local

DNS server for resolution:
If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly. If the domain name is from the country domains section, the resolver receives a domain name such as “ch.fhda.cu.ca.us.” The procedure is the same. Mapping Addresses to Names a client can send an IP address to a server to be mapped to a domain name. As mentioned before, this is called a PTR query. To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address is reversed and two labels, in-addr and arpa, are appended to create a domain acceptable by the inverse domain section. For example, if the resolver receives the IP address 132.34.45.121, the resolver first inverts the address and then adds the two labels before sending. The domain name sent is “121.45.34.132.in-addr.arpa.”, which is received by the local DNS and resolved.

Recursive Resolution:
The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the authority, it

responds; otherwise, it sends the query to yet another server. When the query is finally resolved, the response travels back until it finally reaches the requesting client.

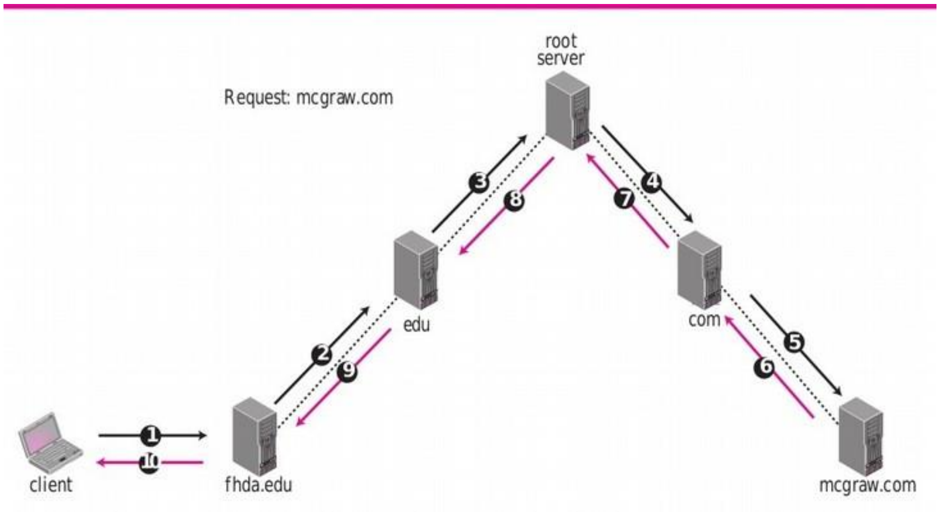


Figure 5: Recursive resolution.

Algorithm:

1. **Give the input as website address e.g.www.google.com.**
2. Get address, if first character is a digit then assume is an address
3. Convert address from string representation to byte array
4. Get Internet Address of this host address
5. Get Internet Address of this host name
6. Show the Internet Address as name/address
7. Print Host Name and Host Address
8. Get the default initial Directory Context
9. Get the DNS records for Address
10. Get an enumeration of the attributes and print them out

```
#  javac DNSLookup.java
 #java DNSLookup google.com
google.com/64.233.167.99
-- DNS INFORMATION --
A: 64.233.187.99, 72.14.207.99, 64.233.167.99
NS: ns4.google.com., ns1.google.com., ns2.google.com., ns3.google.com.
MX: 10 smtp4.google.com., 10 smtp1.google.com., 10 smtp2.google.com., 10
```

Conclusion:

Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses. Thus we have implemented DNS lookup

Assignment 12

Problem Definition:
Installing and configure DHCP server and write a program to install the software on remote machine.

Prerequisite:

1. Knowledge about IP and Subnets.
2. Linux basic commands.

Learning Objectives:

1. Understand the concept of DHCP.
2. Configuring DHCP and installation of software.

New Concepts:

1. Crimping
2. Access Point Configuration

Theory

Introduction

- DHCP (Dynamic Host Configuration Protocol) is a protocol that lets network administrators manage centrally and automate the assignment of IP (Internet Protocol) configurations on a computer network.
- When using the Internet's set of protocols (TCP/IP), in order for a computer system to communicate to another computer system it needs a unique IP address.
- Without DHCP, the IP address must be entered manually at each computer system. DHCP lets a network administrator supervise and distribute IP addresses from a central point.
- The purpose of DHCP is to provide the automatic (dynamic) allocation of IP client configurations for a specific time period (called a lease period) and to eliminate the work necessary to administer a large IP network.
- When connected to a network, every computer must be assigned a unique address.
- However, when adding a machine to a network, the assignment and configuration of network (IP) addresses has required human action.
- The computer user had to request an address, and then the administrator would manually configure the machine. Mistakes in the configuration process are easy for novices to make, and can cause difficulties for both the administrator making the error

as well as neighbors on the network. Also, when mobile computer users travel between sites, they have had to relive this process for each different site from which they connected to a network.

- In order to simplify the process of adding machines to a network and assigning unique IP addresses manually, there is a need to automate the task.
- The introduction of DHCP alleviated the problems associated with manually assigning TCP/IP client addresses. Network administrators have quickly appreciated the importance, flexibility and ease-of-use offered in DHCP.

Advantages of DHCP:-

DHCP has several major advantages over manual configurations.

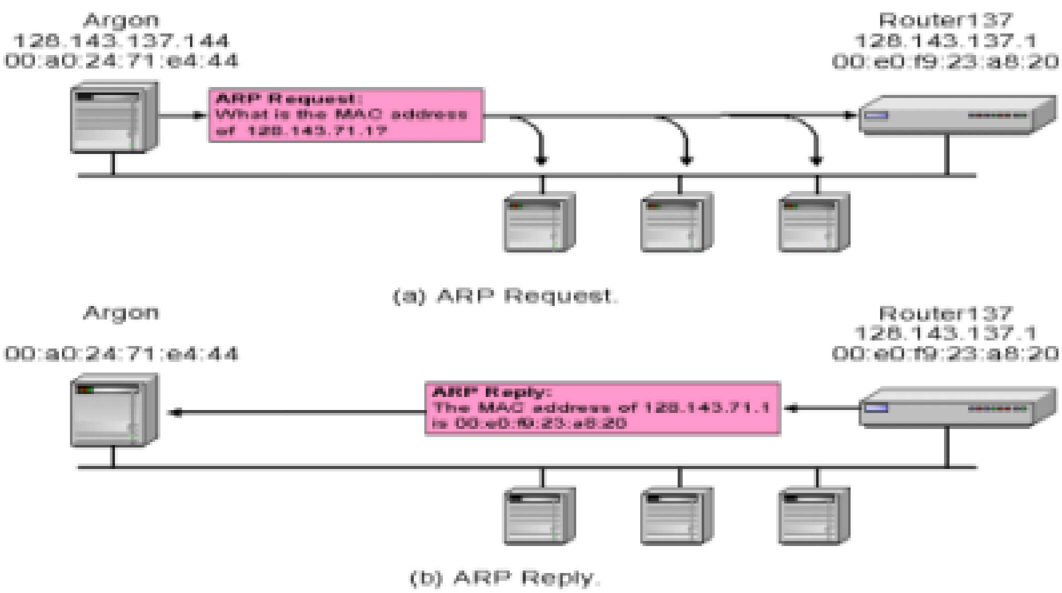
- Each computer gets its configuration from a "pool" of available numbers automatically for a specific time period (called a leasing period), meaning no wasted numbers.
- When a computer has finished with the address, it is released for another computer to use. Configuration information can be administered from a single point.
- Major network resource changes (e.g. a router changing address), requires only the DHCP server be updated with the new information, rather than every system.

DHCP message types:

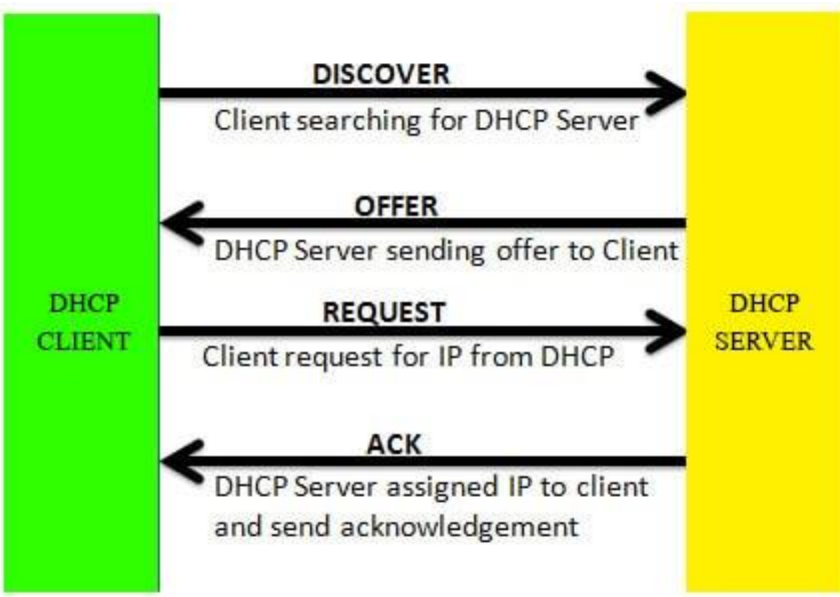
Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM

DHCP Operations:-

1. DHCP Discover



2. DHCP Offer



3. DHCP Discover: At this time, the DHCP client can start to use the IP address
4. DHCP Release: At this time, the DHCP client has released the IP address

5.4.4 Installing DHCP in Ubuntu:

Open terminal and type following commands:-

1. sudo apt-get install isc-dhcp-server
2. sudo gedit /etc/dhcp/dhcpd.conf then make changes in file....

default-lease-time 600; max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 10.1.32.255;
subnet 192.168.1.0 netmask 255.255.255.0
Range 10.1.32.10 10.1.32.20;}

3.

default-lease-time 600; max-lease-time 7200;

option subnet-mask 255.255.255.0;

option broadcast-address 10.1.32.255;

subnet 192.168.1.0 netmask 255.255.255.0

range 10.1.32.10 10.1.32.20; }

3. save file and close
4. again on terminal give following commands....

   sudo service isc-dhcp-server restart

   sudo service isc-dhcp-server start

5. On another PC in Internet properties change to Obtain IP address automatically and then check the IP address.

Conclusion:

Hence we Installed and Configured DHCP and studied Installation of Software on remote Machine

Assignment 14: Group C (Unit V & VI)

Problem Definition: Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer tool.

Prerequisite:
Learning Objectives:

1. To Understand the concept of HTTP,HTTPS and FTP Protocol.

Theory

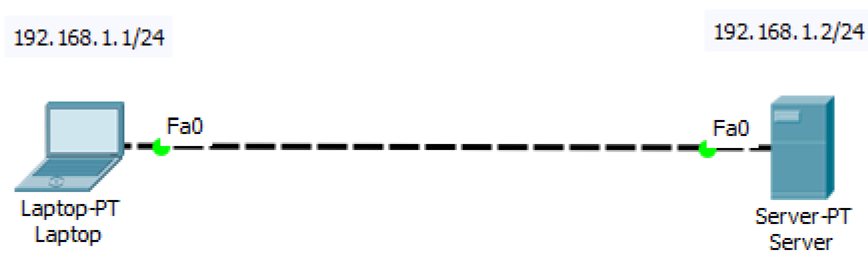
The File Transfer Protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP employs a client-server architecture whereby the client machine has an FTP client installed and establishes a connection to an FTP server running on a remote machine. After the connection has been established and the user is successfully authenticated, the data transfer phase can begin.

Worth noting: Although FTP does support user authentication, all data is sent in clear text, including usernames and passwords. 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).

Let's now do FTP configuration in Packet Tracer:

1.Build the network topology.



FTP topology.PNG

2. Configure static IP addresses on the Laptop and the server.

Laptop: IP address: 192.168.1.1 Subnet Mask: 255.255.255.0

Server: IP address: 192.168.1.2 Subnet Mask: 255.255.255.0

3. Now try using an FTP client built in the Laptop to send files to an FTP server configured in the Server.

From the Laptop's command prompt, FTP the server using the server IP address by typing: ftp 192.168.1.2

Provide the username(cisco) and password(cisco) [which are the defaults] for ftp login.

ftp from laptop.PNG

You are now in the FTP prompt .

PC0 has an FTP client which can be used to read, write, delete and rename files present in the FTP server.

The FTP server can be used to read and write configuration files as well as IOS images. Additionally, the FTP server also supports file operations such rename, delete and listing directory.

With that in mind, we can do something extra. So let's do this:

4. Create a file in the Laptop then upload it to the server using FTP.

To do this, open the Text Editor in the Laptop, create a file and give it your name of choice.

Type any text in the editor then save your file. e.g. myFile.txt.

5. Now upload the file from the Laptop to the server using FTP. (An FTP connection has to be started first. But this is what we've done in step 3)

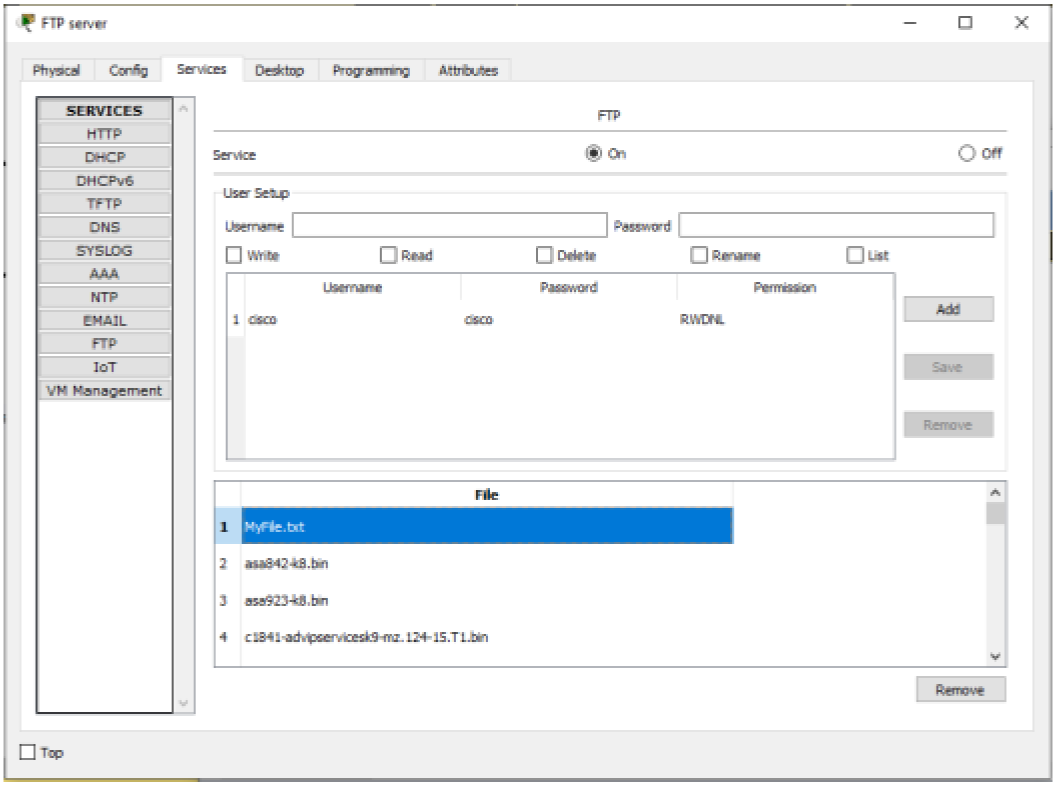
So to do an FTP upload, we'll type:

put MyFile.txt

```
ftp>
ftp>put MyFile.txt

Writing file MyFile.txt to 192.168.1.2:
File transfer in progress...

[Transfer complete - 47 bytes]

47 bytes copied in 0.023 secs (2043 bytes/sec)
ftp>
```

put MyFile to FTP directory.PNG

6. Once file upload is successful, go to the Server FTP directory to verify if the file sent has been received . To do this, go to Server-> Services->FTP. Here look for MyFile.txt sent from the laptop.



MyFile.txt really send to sever.PNG

Something extra: To check other FTP commands supported by the FTP client running on the Laptop(or PC), you can use a question mark (?) on the Laptop's command prompt as shown below:

All FTP commands supported

You can see the put command that we used to upload our file to the FTP server. Other commands listed include:

get-used to get(download) a file from the server.

For example: get MyFile.txt

delete– to delete a file in the FTP directory with the server

For example: delete MyFile.txt

Rename– used to Rename a file

cd – used to change directory.

For example, we can open an HTTP directory in the server by typing: cd /http. This will change the current directory from FTP directory to HTTP directory

Once the http directory is open, you can upload a file to the HTTP server. You're now uploading a file to an HTTP folder(directory) using FTP.

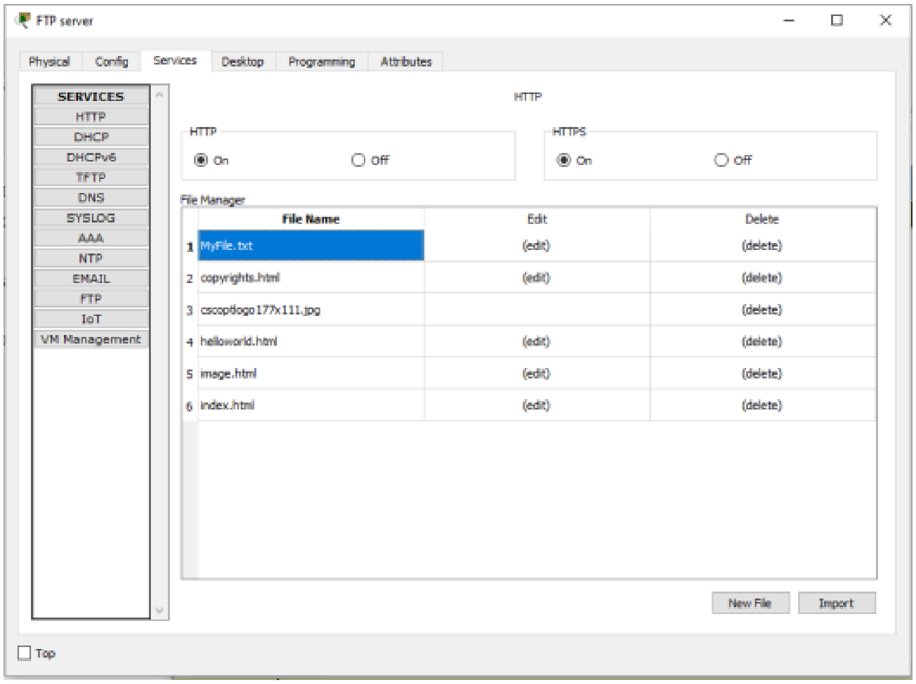
For example: put MyFile.txt

```
ftp>cd /http
ftp>
Working directory changed to /http successfully
ftp>put MyFile.txt

Writing file MyFile.txt to 192.168.1.2:
File transfer in progress...

[Transfer complete - 47 bytes]

47 bytes copied in 0.01 secs (4700 bytes/sec)
```

To see this working, let's open an HTTP directory and upload(put) a file to it using FTP:

changing directory then put files to HTTP directory using FTP

You can now check up in the HTTP directory in the server and verify that the file uploaded from the Laptop(MyFile.txt) is well received:



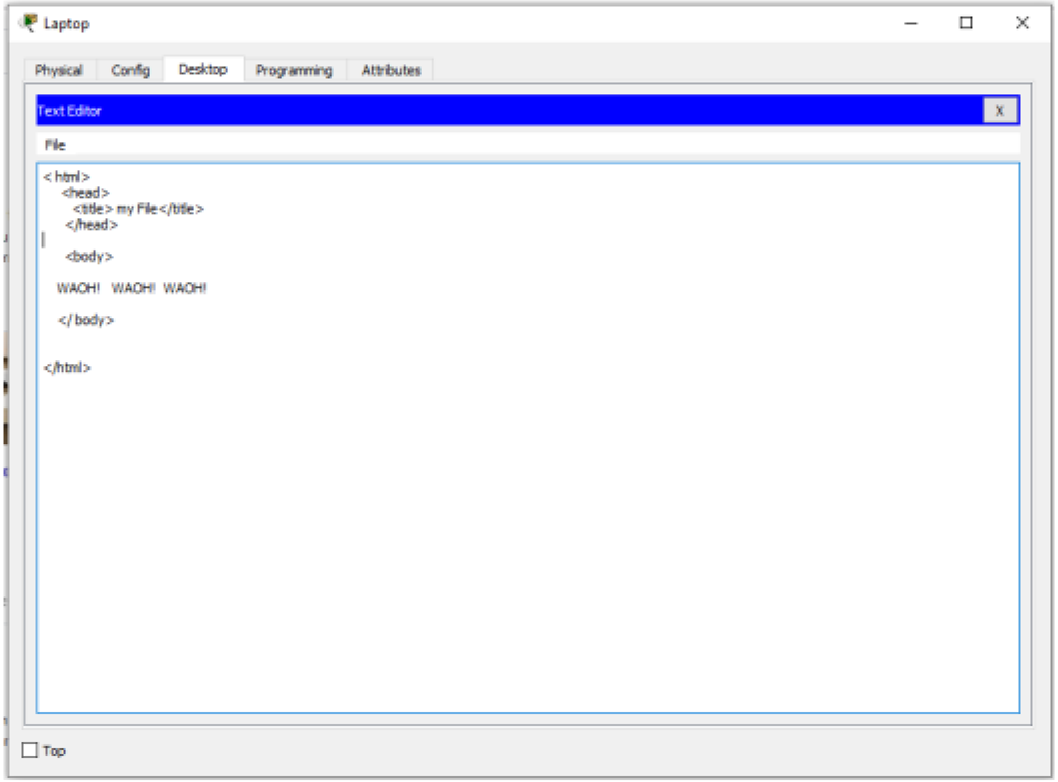
MyFile.txt really send to HTTP server

Notice that we are uploading files to an HTTP Server directory using File Transfer Protocol.(FTP). This is what actually happens when you use an FTP client such as FileZilla client to upload files to a website. In our case here, we are using an FTP client built-in the Laptop.

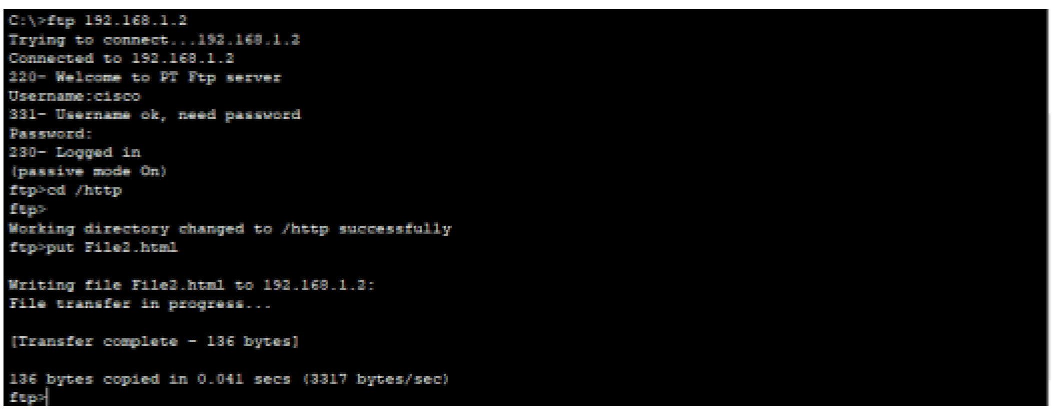
This may interest you: The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows and Linux operating systems. (Actually this is what we have been using this far). Many FTP clients(e.g. FileZilla) and automation utilities have since been developed for desktops, servers, mobile devices, and hardware. FTP has also been incorporated into productivity applications, such as HTML editors.

We'll create an html file in our Laptop, upload it to HTTP server directory using FTP, then try to access the file from the Laptop's browser.

On the Laptop, open the text editor, then type some markup(html) and save the file with the extension .html. See all this below:

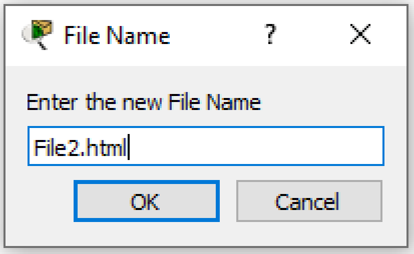


File2 HTML code



```
C:\>ftp 192.168.1.2
Trying to connect...192.168.1.2
Connected to 192.168.1.2
220- Welcome to PT Ftp server
Username:cisco
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>cd /http
ftp>
Working directory changed to /http successfully
ftp>put File2.html

Writing file File2.html to 192.168.1.2:
File transfer in progress...

[Transfer complete - 136 bytes]

136 bytes copied in 0.041 secs (3317 bytes/sec)
ftp>
```

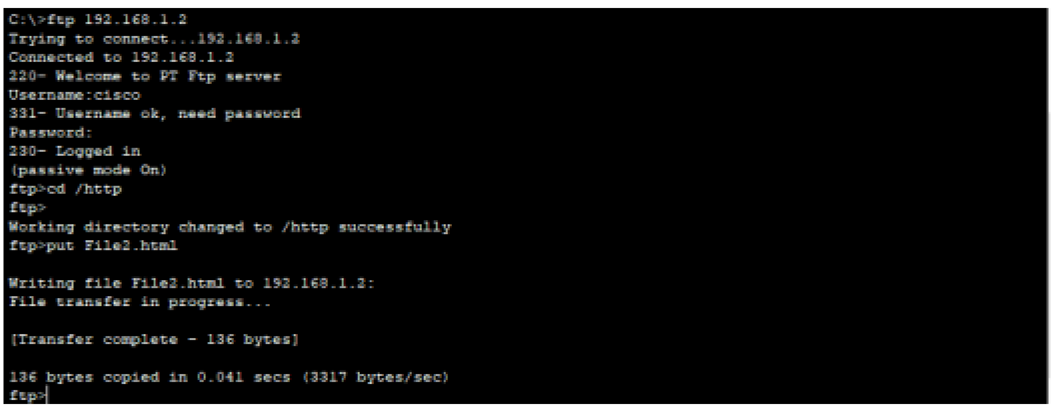
Save your file as an html file like this:



File2 html.PNG

Now upload the file(File2.html) to the HTTP server using FTP. This is easy. We've already done it previously!

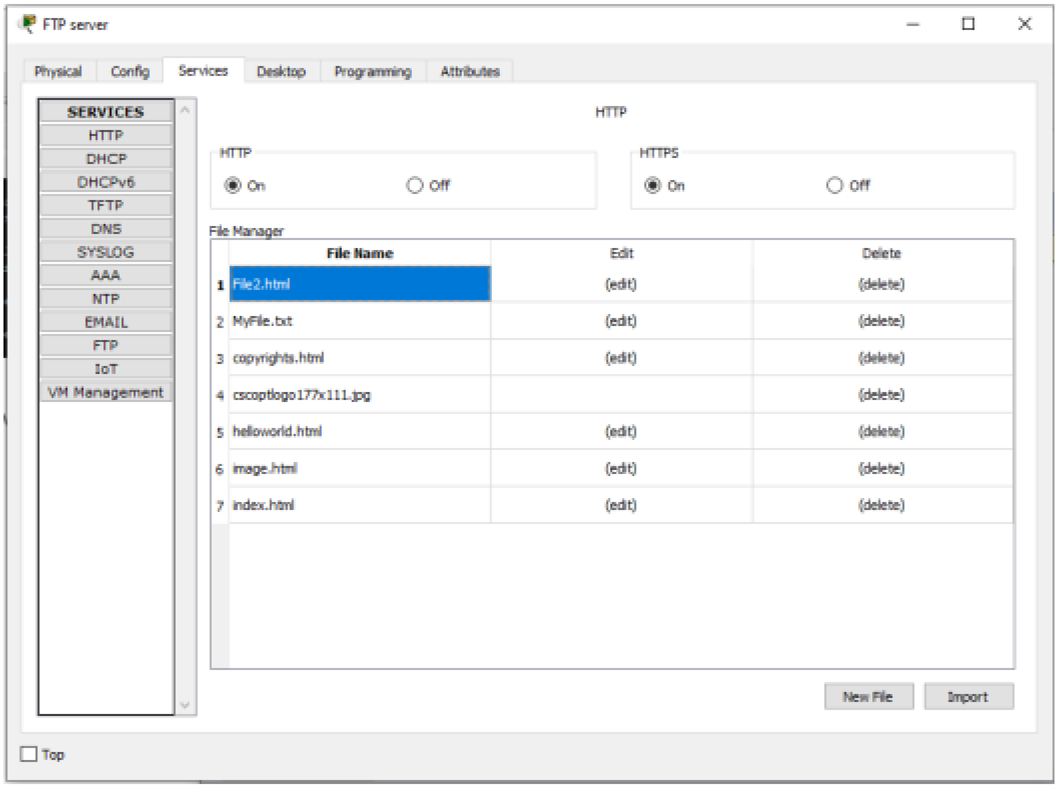
If you're already in the HTTP directory, you just need to type: put File2.html. If no, first ftp the server(ftp 192.168.1.2), provide the login username(cisco) and password(cisco); change the current directory to HTTP(cd /http) , and finally upload the html file onto the HTTP directory(put File2.html)



Sending File2. html to HTTP directory.PNG

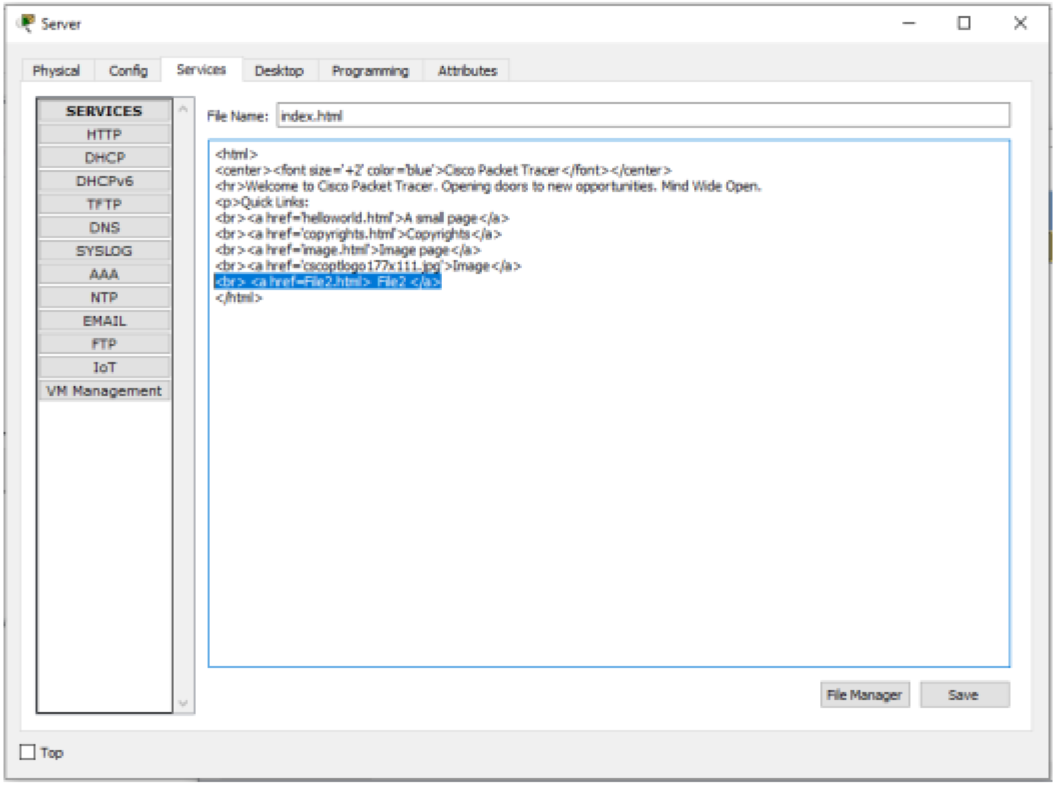
Check whether the html file uploaded has been received in the HTTP directory:

Go to Server->Services-> HTTP. Then look up for the file in the File Manager.



File2 HTML really uploaded into HTTP directory.PNG

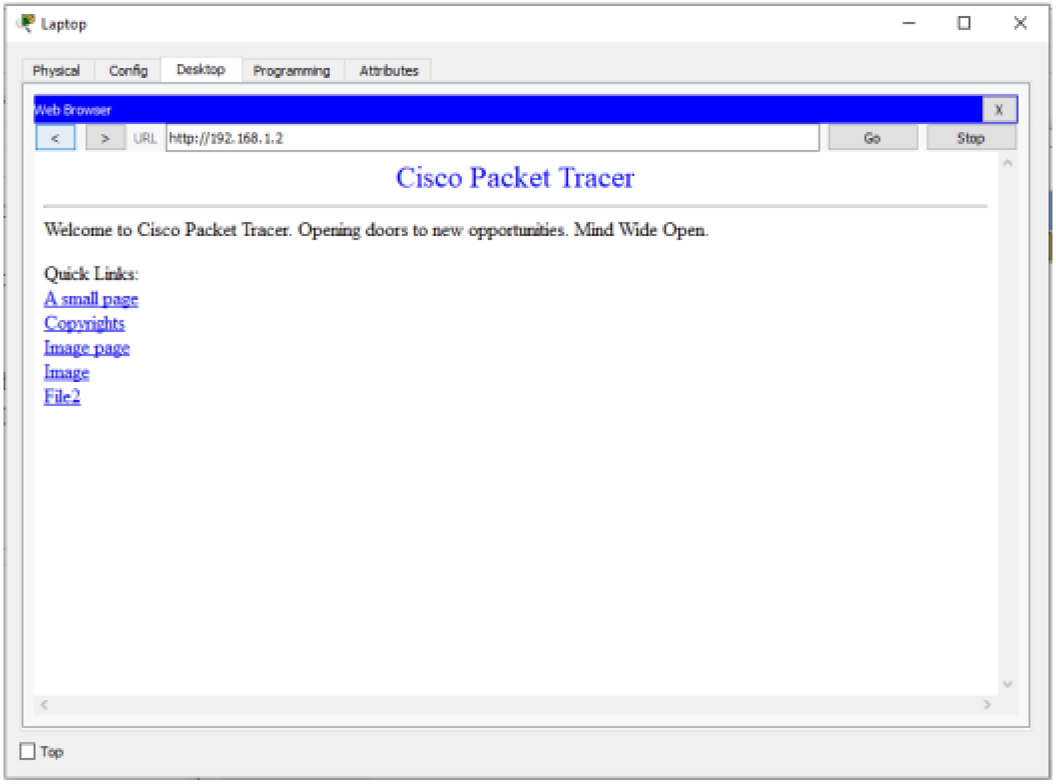
Now edit index.html file in the HTTP directory so as to include a link to File2 that we've just uploaded. This will make File2 accessible from the Laptop's browser. To do this, locate index.html then click edit. Proceed to edit it as shown below. Then save and accept overwrite.Index.html editing to include File2 html.PNG



Finally, try to access the newly uploaded file from the Laptop's browser.

So go to the Laptop's browser and access the server using the server's IP address. By doing this, the browser is making an http request to the server. The server will respond to the Laptop with the index.html file containing a link to File2 which we've uploaded from the Laptop using FTP.

Http response with File2.PNG



Click File2 link to view the contents of the file in the browser.

Conclusion:

We Studied and analyze the performance of HTTP,HTTPS and FTP protocol using Packet tracer tool.

Assignment 15

Problem Statement: To study the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website (banking, e-commerce etc.)

Outcomes: Retrieve SSL protocol by capturing the packets using Wireshark.

Theory:

SSL, or Secure Sockets Layer, is an encryption-based Internet security protocol. It was first developed by Netscape in 1995 for the purpose of ensuring privacy, authentication, and data integrity in Internet communications. SSL is the predecessor to the modern TLS encryption used today.

How does SSL/TLS work?

In order to provide a high degree of privacy, SSL encrypts data that is transmitted across the web. This means that anyone who tries to intercept this data will only see a garbled mix of characters that is nearly impossible to decrypt. SSL initiates an authentication process called a handshake between two communicating devices to ensure that both devices are really who they claim to be. SSL also digitally signs data in order to provide data integrity, verifying that the data is not tampered with before reaching its intended recipient. There have been several iterations of SSL, each more secure than the last. In 1999 SSL was updated to become TLS.

Why is SSL/TLS important?

Originally, data on the Web was transmitted in plaintext that anyone could read if they intercepted the message. For example, if a consumer visited a shopping website, placed an order, and entered their credit card number on the website, that credit card number would travel across the Internet unconcealed.

SSL was created to correct this problem and protect user privacy. By encrypting any data that goes between a user and a web server, SSL ensures that anyone who intercepts the data can only see a scrambled mess of characters. The consumer's credit card number is now safe, only visible to the shopping website where they entered it. Study.narendradwivedi.org SSL also stops certain kinds of cyber-attacks: It authenticates web servers, which is important because attackers will often try to set up fake websites to trick users and steal data. It also prevents attackers from tampering with data in transit, like a tamper-proof seal on a medicine container.

Are SSL and TLS the same thing?

SSL is the direct predecessor of another protocol called TLS (Transport Layer Security). In 1999 the Internet Engineering Task Force (IETF) proposed an update to SSL. Since this update was being developed by the IETF and Netscape was no longer involved, the name was changed to TLS. The differences between the final version of SSL (3.0) and the first version of TLS are not drastic; the name change was applied to signify the change in ownership. Since they are so closely related, the two terms are often used interchangeably and confused. Some people still use SSL to refer to TLS, others use the term "SSL/TLS encryption" because SSL still has so much name recognition.

Conclusion:

We studied the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website (banking, e-commerce etc.)

Assignment 16: Group C (Unit V & VI)

Problem Definition: Illustrate the steps for implementation of S/MIME email security through Microsoft® Office Outlook.

Prerequisite:

Learning Objectives:

1. Understand the concept and working of Encrypted mails

Theory

MIME (Secure/Multipurpose Internet Mail Extensions)

S/MIME allows users to send encrypted and digitally signed emails. This protocol allows recipients of the email to be certain the email they receive is the exact message that began with the sender. It also helps ensure that a message going to an outbound recipient is from a specific sender and not someone assuming a false identity.

How does S/MIME work?

S/MIME provides cryptographic-based security services like authentication, message integrity, and digital signatures. All these elements work together to enhance privacy and security for both the sender and recipient of an email.

S/MIME also works with other technologies such as Transport Layer Security (TLS) which encrypts the path between two email servers. The protocol is also compatible with Secure Sockets Layer (SSL) which masks the connection between email messages and Office 365 (a common email service) servers.

In addition, BitLocker works in conjunction with S/MIME protocol, which encrypts data on a hard drive in a data center so if a hacker gets access, he or she won't be able to interpret the information.

Benefits of encrypted email

1. Safeguards sensitive data

If you're sending information like your Social Security number over email, it's important that it's not easily stolen by hackers.

2. Economical

Instead of purchasing security equipment, you can simply rely on email encryption that's integrated directly on the server.

3. Timesaving

Instead of wasting time using several programs to make sure a connection is secure, you can rely on email encryption to do most of the work for you.

4. Regulation compliance

If you work in the healthcare industry, for example, and you haven't taken the right steps to secure medical data, you could be in violation of HIPAA laws [6]. Encryption helps you avoid those missteps.

5. Protects against malware

Malicious emails sometimes contain viruses masked as innocent email attachments. If you or someone else send an attachment using encrypted email, the email has a digital signature to prove its authenticity.

How does email encryption work?

If you don't want anyone but the receiver to see the contents of a message, encryption is vital. To the outsider, an encrypted email will have a bunch of random letters, digits, or symbols instead of readable text. The person with the private key to decrypt it, typically the receiver, will be able to read the email as usual.

There are generally three encryption types available:

- S/MIME encryption works as long as both the sender and recipient have mailboxes that support it. Windows Outlook is the most popular version that works with this method. Gmail uses it as well.
- Office 365 Message Encryption is best for users with valid Microsoft Office licenses who can use this tool to encrypt the information and files sent via email. It's also a top choice for Outlook users
- PGP/MIME is a more affordable and popular option that other email clients may prefer to use. It's reliable and integrated into many of the apps we use today

Other email products may have their own brand of encryption, but the science behind it is the same. Only senders and recipients who have exchanged keys or digital signatures can communicate within the encrypted network.

How to send encrypted email in Outlook

Encrypting email may sound complicated, but it's not. Microsoft has a reputation for providing its users with simple ways to encrypt data, from files to folders to emails, too. It makes sense that they would include built-in tools for Outlook, their proprietary email system. You don't need a separate software tool or plug-in to start sending secure messages. Just follow these steps to begin.

1. Create a digital certificate

For Outlook users, encrypting a single email is simple. First, you must have a digital signature. To create a digital signature:

1. Start in your Outlook window and click on the File tab

2. Select Options, then Trust Center, then Trust Center Settings
3. Select Email Security, Get a Digital ID
4. You'll be asked to choose a certification authority. This is entirely up to you as most are rated the same
5. You'll receive an email with your digital certificate/ID included
6. Go back into Outlook and select Options and the Security tab
7. In the Security Settings Name field, type in a name of your choosing
8. Ensure that S/MIME is selected from the Secure Message Format box and that Default Security Settings is checked as well
9. Go to Certificates and Algorithms, select Signing Certificate, and click Choose
10. Make sure the box is checked next to Secure Email Certificate, and check the box next to "Send These Certificates with Signed Messages"
11. Click OK to save your settings and start using Outlook

2. Use your digital signature

Now that you have a digital ID, you need to start using it:

1. Open a new message to access the Tools tab
2. Click that, then Customize, and finally the Commands tab
3. From Categories, select Standard
4. From Categories, select Digitally Sign Message

3. Encrypt Outlook messages

You can now send encrypted messages to a recipient with the next steps.

1. Open the window to compose a new message and select the Options tab, then More Options
2. Click the dialog box (triangle with arrow pointing down) in the lower-right corner
3. Choose Security Settings and check the box next to Encrypt message contents and attachments
4. Write your message as normal and send

After you've sent and received a message that you've both signed and encrypted, you don't have to sign it again. Outlook will remember your signature.

4. Encrypt all Outlook messages

You can encrypt each one, or you can use the steps below to encrypt all outgoing messages in Outlook:

Open the File tab in Outlook
Select Options, then Trust Center, and Trust Center Settings
From the Email Security tab, select Encrypted email
Check the box next to Encrypt content and attachments for outgoing messages
Use Settings to customize additional options, including certificates

How to Send Encrypted Email

Have you ever wondered about the security of your private email conversations? Whether at work, school, or home, sending emails comes with a bit of a risk. There's one thing you can do to discourage data breaches and attacks on your sensitive data, however. Use encrypted email. Learn how to practice this common-sense method for communicating in our step-by-step guide. But first, let’s look at why you should embrace encryption for your email correspondence.

How to Encrypt Email and Send Secure Messages

Emails sent over an open network can be intercepted and malicious actors can see email contents, attachments, or even take over your account.

To drive home the importance of email security, take a look at some alarming statistics that show the widespread cybersecurity issues that may have affected you in the past and still pose a threat today.

Conclusion: Thus we have studied the steps for implementation of S/MIME email security through Microsoft® Office Outlook.