SINHGAD COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

SUBJECT CODE: 310247

LAB MANUAL

Computer Networks and Security Laboratory

Semester – I, Academic Year:
2022-23

Sinhgad Institutes

Sinhgad College of Engineering
Department of Computer Engineering
Academic Year 2022-23

Third Year of Computer Engineering (2019 Course)

SUBJECT NAME: Computer Networks and Security Laboratory **Code: 310247**

Teaching Scheme	Credit:01	Exam Scheme & Marks
Practical: 02 Hours/Week		Term work: 25 Marks Oral: 25 Marks

Course Objectives:

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

Course Outcomes:

On completion of the course, learners will be able to

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

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

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

CO4: Develop Client-Server architectures and prototypes

CO5: Implement web applications and services using application layer protocols

CO6: Use network security services and mechanisms

CO-PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	1	-	2	-	2	1	1	-	-	1	-	1
CO 2	-	3	-	1	1	-	-	1	-	-	-	-
CO 3	3	2	1	1	-	-	-	1	-	-	1	1
CO 4	-	1	2	1	1	1	-	-	-	-	-	1
CO 5	2	3	-	-	1	-	-	-	1	-	-	-
CO 6	-	1	3	1	1	-	1	-	2	-	-	1

Third Year of Computer Engineering (2019 Course)
310247: Computer Networks and Security Laboratory

List of Experiments

Sr. No	Gr. No	Group - C	
10)	C.11	Write a program for DNS lookup. Given an IP address as input, it should return URL and vice-versa.	53
11)	C.13	Capture packets using Wireshark, write the exact packet capture filter expressions to accomplish the following and save the output in file: 1. Capture all TCP traffic to/from Facebook, during the time when you log in to your Facebook account 2. Capture all HTTP traffic to/from Facebook, when you log in to your Facebook account 3. Write a DISPLAY filter expression to count all TCP packets (captured under item #1) that have the flags SYN, PSH, and RST set. Show the fraction of packets that had each flag set. 4. Count how many TCP packets you received from / sent to Face book, and how many of each were also HTTP packets.	56
12)	C.14	Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer tool.	58
13)	C.15	To study the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website (banking, e-commerce etc.).	65
14)	C.16	To study the IPsec (ESP and AH) protocol by capturing the packets using Wireshark tool.	70

Faculty Co-Ordinator for the Lab:	
1. Prof. N. A. Mhetre 2. Prof. R. A. Vasmatkar	Dr. M.P. Wankhade HoD, Dept of Computer Engg.

Assignment A1

Title: Wired LANs

Objective/s: To learn basics of computer network.

Problem statement: Setup a wired LAN using Layer 2 Switch. Prepare and test cable using line tester, configure machine using IP addresses, test using PING utility and demonstrate the PING packets captured using wireshark.

Software&/hardware requirements: CAT5 cables, line tester, wire-shark, cisco packet tracer.

Theory: //Here write answers to FAQ questions given below.

1. Explain following networking devices:

   Repeater, Hubs, Switch, Bridge, Router, Gateway, Access point

2. Refer question 1 and State on which layer of network model do these devices work?
3. In order to increase bandwidth per node which device will you use- A hub or a switch? Explain why?
4. What is a LAN?
5. What is a MAN?
6. What is a WAN?
7. What is a PAN?
8. Explain in brief guided/wired medium along with its types.
9. Explain in brief unguided/wireless medium along with its types.
10. What is the maximum length allowed for a UTP cable?

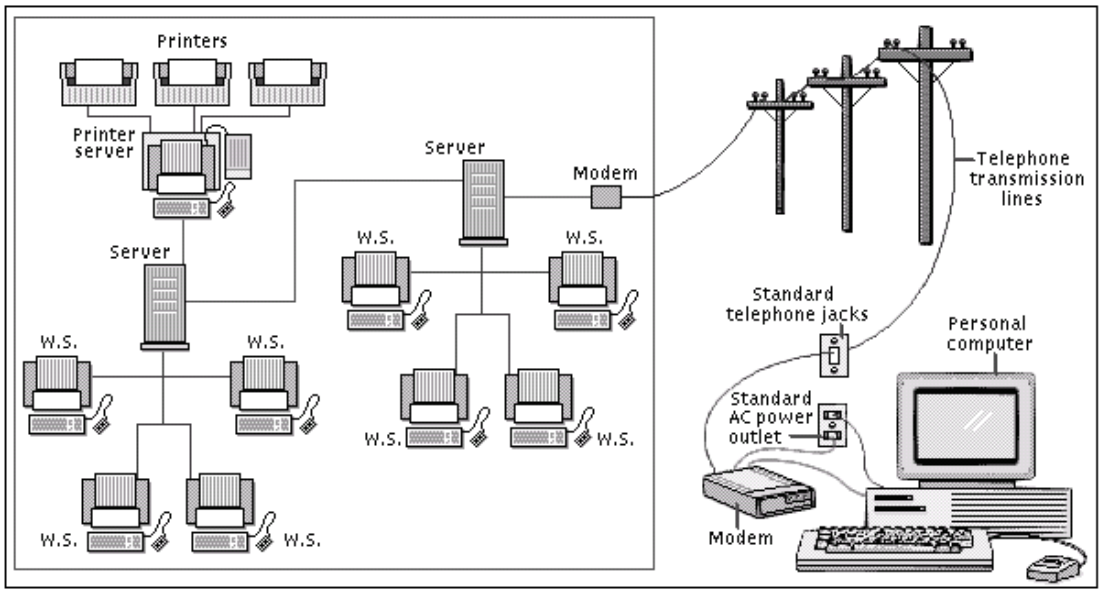
Conclusion: Studied wired LAN using Layer 2 Switch.

Theory:

Introduction:-

Computer Networks, the widespread sharing of information among groups of computers and their users, a central part of the information age. The popular adoption of the personal computer (PC) and the local area network (LAN) during the 1980s has led to the capacity to access information on a distant database; download an application from overseas; send a message to a friend in a different country; and share files with a colleague—all from a personal computer.

The networks that allow all this to be done so easily are sophisticated and complex entities. They rely for their effectiveness on many cooperating components. The design and deployment of the worldwide computer network can be viewed as one of the great technological wonders of recent decades.



Computer Network

Networks are connections between groups of computers and associated devices that allow users to transfer information electronically. The local area network shown on the left is representative of the setup used in many offices and companies. Individual computers, called work stations (WS), communicate to each other via cable or telephone line linking to servers. Servers are computers exactly like the WS, except that they have an administrative function and are devoted entirely to monitoring and controlling WS access to part or all of the network and to any shared resources (such as printers). The red line represents the larger network connection between

servers, called the backbone; the blue line shows local connections. A modem (modulator/demodulator) allows computers to transfer information across standard telephone lines. Modems convert digital signals into analogue signals and back again, making it possible for computers to communicate, or network, across thousands of miles.

Study of Network Devices:-

NIC (Network Interface Card):-

Each computer includes the File server or a Network will have a card plugged in the PCI Expunction slot or will have on-board NIC (Network Interface Card), which will provide connectivity among the workstation in the network through cables.

Type's of Card:-

1. Arc net card (2.5 mbits/sec)
2. Ethernet card (10/100 mbps)
3. Token Ring card (4-16 mbits/sec)

Hub/Switch:-

These devices are used for Re-directing traffic, i.e. in a **Star** Topology the central device is used to ECHO/Re-Direct the packets coming from one workstation/node to the Destination workstation/node.

This is done by using the devices like Hub/Switch, during the present situation **Hub's are absolute due to their disadvantages of Echoing a packet from one node to all, which leads to increasing N/W traffic and packet Collision.**

Type of Hub:-

1. **Passive Hub:-**

   It is a device which do not require any type of power supply and does not boost incoming signal, it just echo the incoming signal to all nodes.

2. **Active Hub :-**

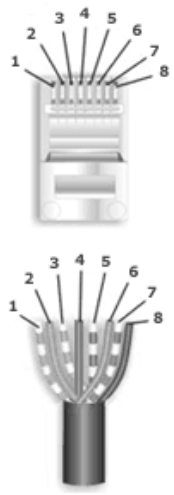
It is a device which requires power supply and boosts the incoming signal and echoes the signal to all nodes.

Hub where absolute due to use of an intelligent device called **Switch** which reads the destination adders and sends the incoming packet to it.

Paring Rules and Color Code:-

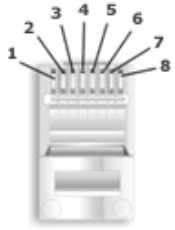
The CAT 5 Cable consist of 8 wires which comes pares of White/Blue, Blue, White/Orange, Orange, White/Green, Green, White/Brown, Brown and they are coded for **Straight** and **Cross** combinations respectively.

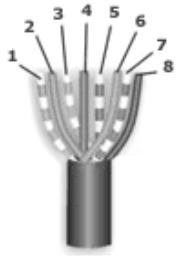
Straight:-



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

Cross:-





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

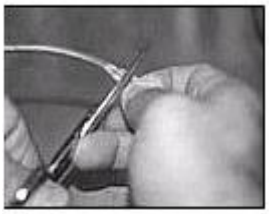
Connections among devices:-

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

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.

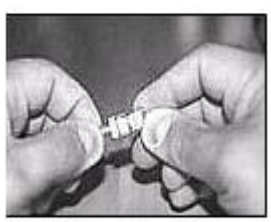
4. At this point, recheck the wiring sequence with the diagram.



5. Optional: Make a mark on the wires at 1/2" from the end of the cable jacket.

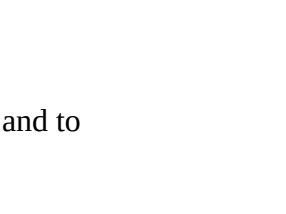
6. Hold the grouped (and sorted) wires together tightly, between the thumb, and the forefinger.

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.

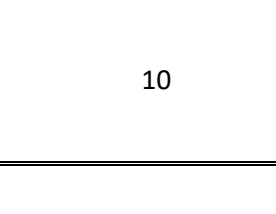


8. Conductors should be at a straight 90 degree angle, and be 1/2" long, prior to insertion into the connector.

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 Testing Tool:-

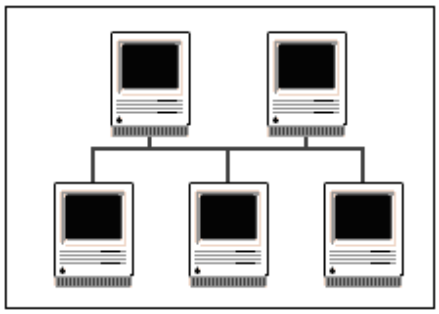
It is a tool used for testing weather there is no cut in between two terminals and to identify the type of pair crimp with.

Study of Topologies:-

What is a Topology?

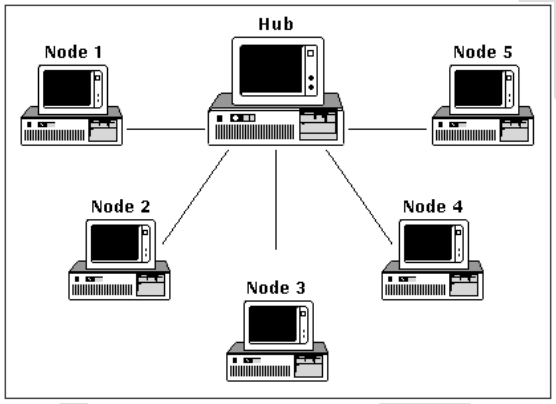
The physical topology of a network refers to the configuration of cables, computers, and other peripherals. Physical topology should not be confused with logical topology which is the method used to pass information between workstations.

1. Bus Topologies:-



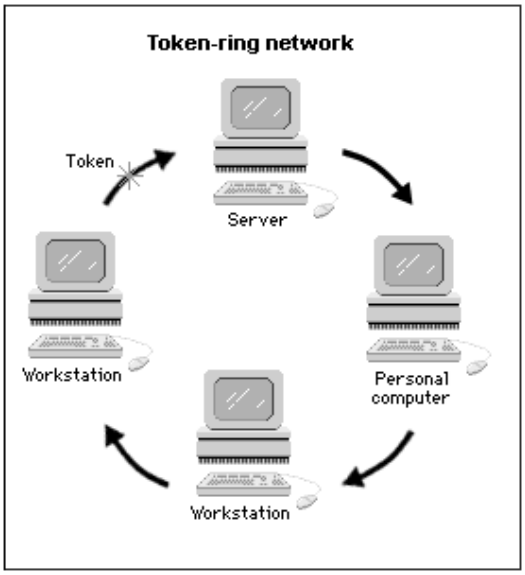
In a bus network configuration, each node is connected to one main communications line. With this arrangement, even if one of the nodes goes down, the rest of the network can continue to function normally.

2. Star Topologies:-



A star network consists of several nodes connected to a central hub/switch in a star-shaped configuration. Messages from individual nodes pass directly to the hub/switch, which determines any further routing.

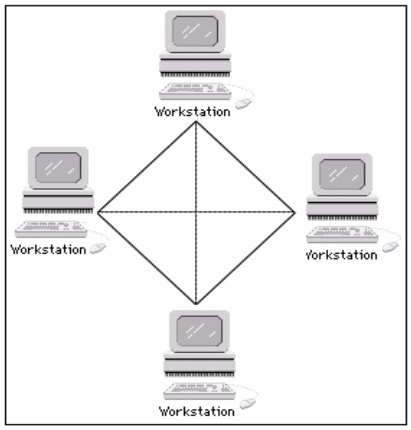
3. Ring Topology:-



Token Ring Network, in computer science, a LAN formed in a ring (closed loop) topology that uses token passing as a means of regulating traffic. On a token ring network, a token governing the right to transmit is passed from one station to the next in a physical circle. If a station has information to transmit, it "seizes" the token, marks it as being in use, and inserts the information. The "busy" token, plus

message, is then passed around the circle, copied when it arrives at its destination, and eventually returned to the sender. The sender removes the attached message and then passes the freed token to the next station in line. Token ring networks are defined in the IEEE 802.5 standards.

4. Mesh Topology:-

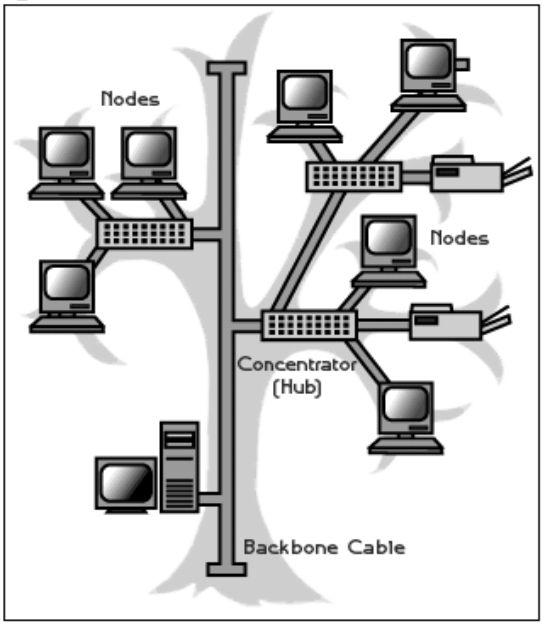


The type of network topology in which each of the nodes of the network is connected to each of the other nodes in the network with a point-to-point link – this makes it possible for data to be simultaneously transmitted from any single node to all of the other nodes.

Note: The physical fully connected mesh topology is generally too costly and complex for practical networks, although the topology is used when there are only a small number of nodes to be interconnected

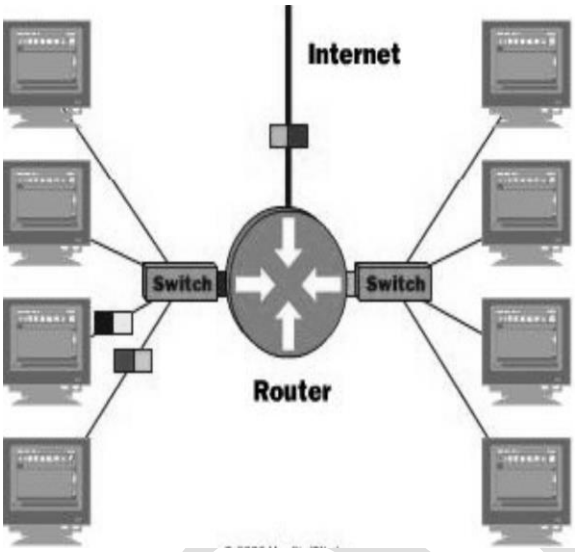
5. Hybrid/Tree Topology:-

A tree topology combines characteristics of linear bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable.



These topologies can also be mixed. For example, a bus-star network consists of a high-bandwidth bus, called the ***backbone,*** which connects a collection of slower-bandwidth star segments.

How Routers Work



Routers are the traffic cops of intranets. They make sure that all data gets sent to where it's supposed to go and that it gets sent via the most efficient route. Routers are also useful tools to make the most efficient use of the intranet. Routers are used to segment traffic and provide redundancy of routes. Routers use encapsulation to permit different protocols to be sent across otherwise incompatible networks.

Just as routers direct traffic on the Internet, sending information to its proper destination, routers on an intranet perform the same function. Routers-equipment that is a combination of hardware and software-can send the data to a computer on the same subnetwork inside the intranet, to another network on the intranet, or outside to the Internet. They do this by examining header information in IP packets, and then sending the data on its way. Typically, a router will send the packet to the next router closest to the final destination, which in turn sends it to an even closer router, and so on, until the data reaches its intended recipient.

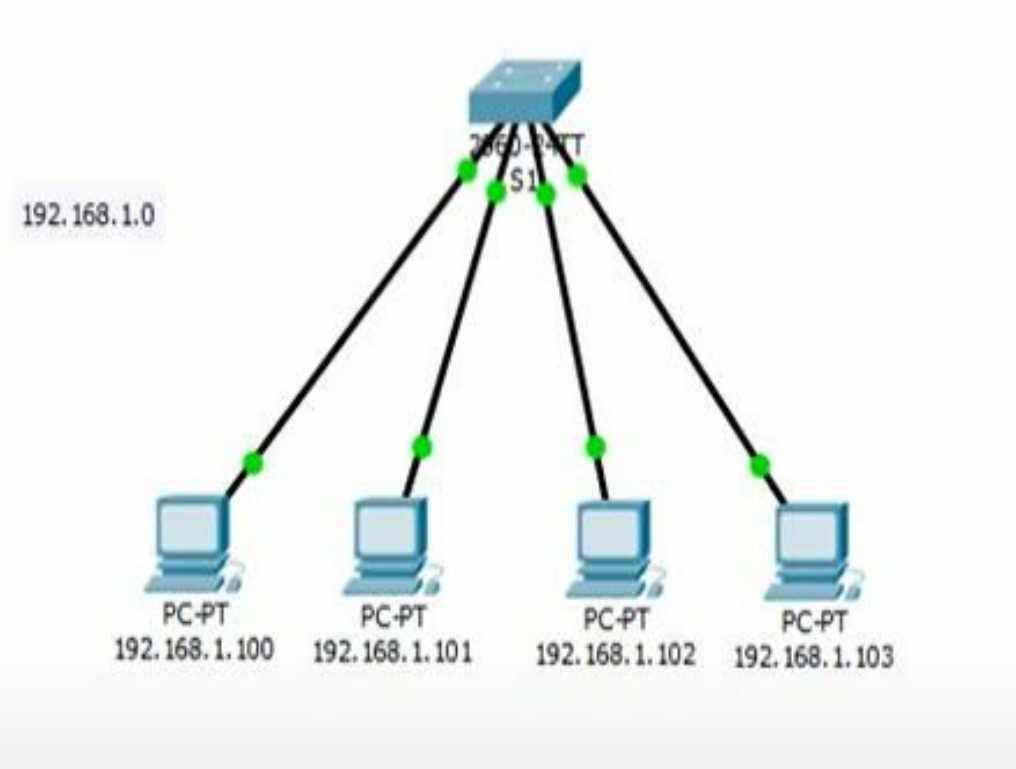
A router has input ports for receiving IP packets, and output ports for sending those packets toward their destination. When a packet comes to the input port, the router examines the packet header, and checks the destination in it against a routing table-a database that tells the router how to send packets to various destinations.

Based on the information in the routing table, the packet is sent to a particular output port, which sends the packet to the next closest router to the packet's destination.

If packets come to the input port more quickly than the router can process them, they are sent to a holding area called an input queue. The router then processes packets from the queue in the order they were received. If the number of packets received exceeds the capacity of the queue (called the length of the queue), packets may be lost.

In a simple intranet that is a single, completely self-contained network, and in which there are no connections to any other network or the intranet, only minimal routing need be done, and so the routing table in the router is exceedingly simple with very few entries, and is constructed automatically by a program called *ifconfig*

Screenshot of Wired LAN using Layer2 – Switch in Cisco Packet Tracer.



Assignment A3

Title: WAN

Objective/s: To learn WAN setup using packet tracer tool.

Problem statement: Setup a WAN which contains wired as well as wireless LAN by using a packet tracer tool. Demonstrate transfer of a packet from LAN1(wired) to LAN2(wireless).

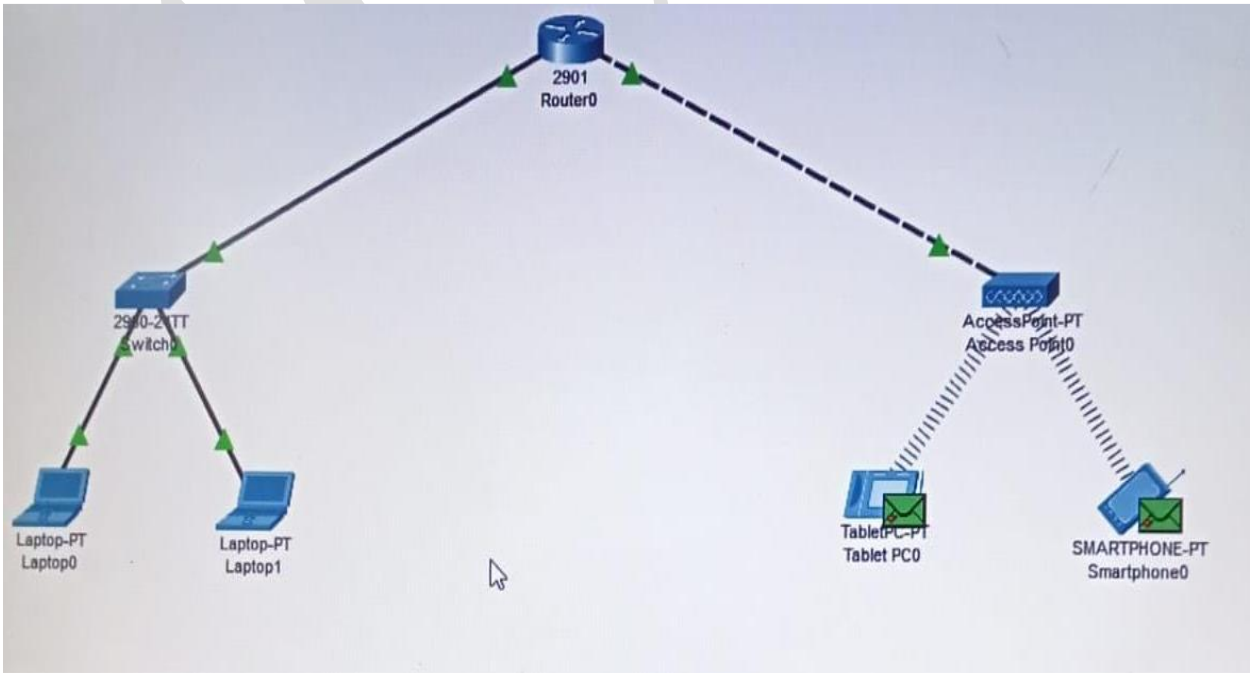
Software&/hardware requirements: cisco packet tracer tool.

Theory: //Here write answers to FAQ questions given below.

1. Explain Bus Topology.
2. Explain Star Topology.
3. Explain Ring Topology.
4. Explain Mesh Topology.
5. Explain Tree / hierarchical Topology.
6. Explain Hybrid Topology.
7. What is NIC? Explain MAC address.
8. State layers of OSI reference model and explain their functions in brief.
9. State layers of TCP/IP model and explain their functions in brief.
10. Compare and differentiate TCP/IP and OSI reference model.

Conclusion: Studied setup and configuration of WAN using packet tracer.

Screenshot of WAN which contains wired as well as wireless LAN by using a packet tracer tool.



Assignment A4

Title: Hamming code & CRC method.
Objectives: To understand the concept of Hamming code & CRC method in datalink layer

Problem Statement:

Write a program for error correction & detection for 7/8 bit ASCII codes using Hamming codes or CRC.

Outcomes:

Demonstrate Hamming code & CRC method

Software&/hardware requirements:

Hardware: PC-2

Software: gcc, JDK compiler and Wireshark.

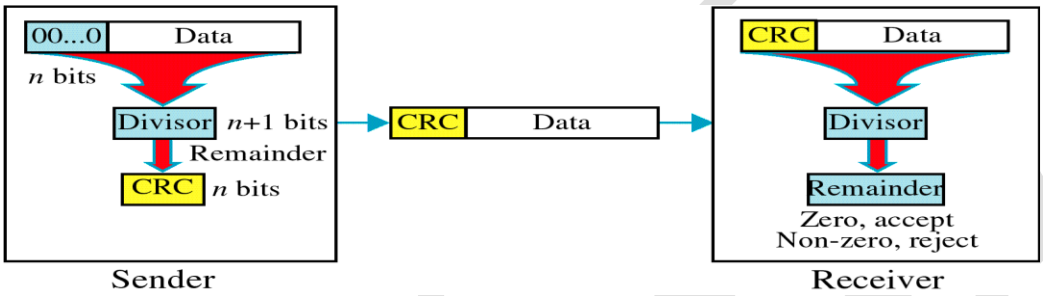
Theory: //Here write answers to FAQ questions given below.

1. What are the different types of errors?
2. What is Hamming code?
3. What is CRC method?
4. What is difference between Hamming code & CRC method?
5. Define hamming distance and calculate its value for two code words 11100 and 11011.
6. State any four desirable properties of line code.
7. Define parity check.
8. A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1.
    a. What is the actual bit string transmitted?
    b. Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?
9. What is redundancy?
10. Test if this code words are correct 1010101, assuming this is created using an even parity Hamming Code. If it is incorrect, indicate what the correct code word should have been. Also, indicate what the original data was.

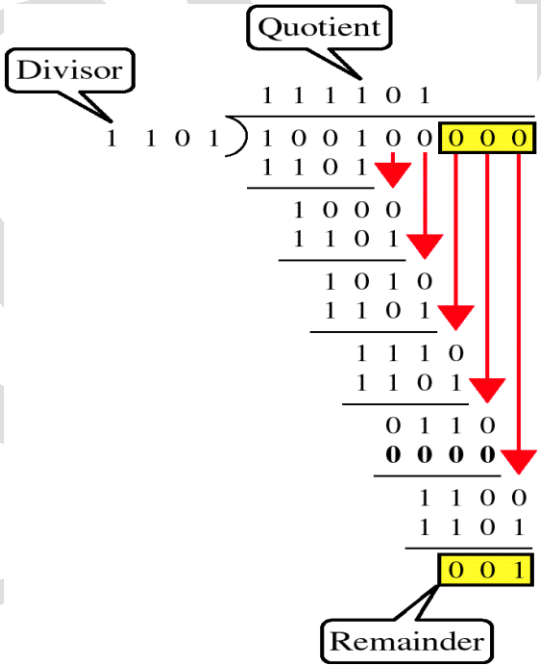
Theory:

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.



Example:



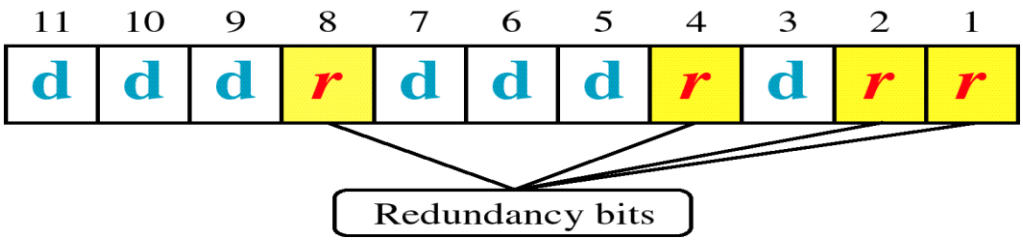
Hamming code:

- 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.

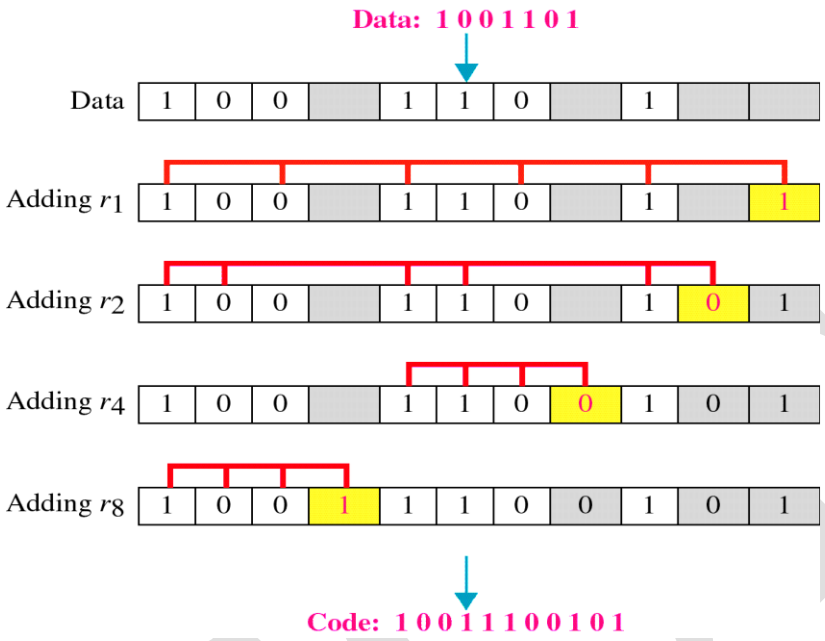
General algorithm

- 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.

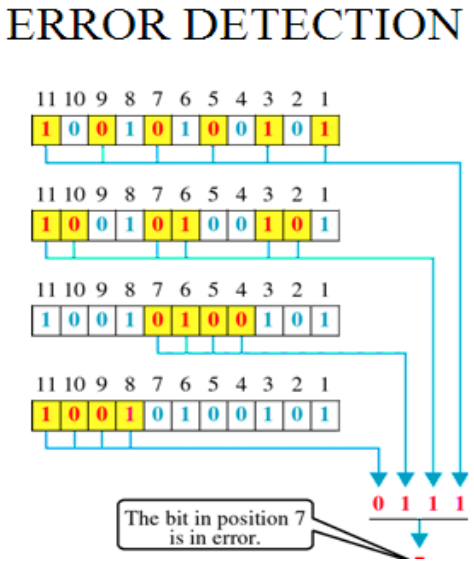


Example

Error detection



Error correction



Conclusion: Hence we have implemented CRC and Hamming code.

Assignment A5

Title: Go back N and Selective Repeat Modes of Sliding Window Protocol.

Objectives: To develop an understanding of various protocolsat datalink layer

Problem Statement:

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

Outcomes:

Demonstrate Go Back N and Selective Repeat protocols

Software&/hardware requirements:

Hardware: PC-2

Software: JDK compiler and Wireshark.

Theory: //Here write answers to FAQ questions given below.

1. What is the difference between flow control and error control?
2. What Automatic Repeat Request (ARQ)?
3. Describe briefly stop and wait ARQ.
4. Describe briefly Go-Back-N ARQ?
5. Describe briefly Selective Repeat ARQ?
6. What do you mean by pipelining, is there any pipelining in error control?
7. What is piggybacking?
8. Explain how flow control and error control is achieved using GBN and SR ARQ protocol.
9. What is sliding window protocol? Differentiate between stop-and wait ARQ and Go-Back-N protocol.
10. Complete the table :

Protocol/parameters	Pipelining No/Yes	Sender window size	Receiver window size
Stop and wait			
Go back N			
Selective repeat			

Theory:

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. When it receives the expected frame, the receiver advances the window.

In transmit flow control, sliding window is a variable-duration window that allows a sender to transmit a specified number of data units before an acknowledgement is received or before a specified event occurs.

Flow Control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data should not be allowed to overwhelm the receiver. Receiver should also be able to inform the transmitter before its limits (this limit may be amount of memory used to store the incoming data or the processing power at the receiver end) are reached and the sender must send fewer frames. Hence, Flow control refers to the set of procedures used to restrict the amount of data the transmitter can send before waiting for acknowledgment.

There are two methods developed for flow control namely Stop-and-wait and Sliding-window Sliding window algorithms, used by TCP, permit multiple data packets to be in simultaneous transit, making more efficient use of network bandwidth.

Sliding Window Protocol:

With the use of multiple frames for a single message, the stop-and-wait protocol does not perform well. Only one frame at a time can be in transit. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time. Efficiency can also be improved by making use of the full-duplex line. To keep track of the frames, sender station sends sequentially numbered frames. Since the sequence number to be used occupies a field in the frame, it should be of limited size. If the header of the frame allows k bits, the sequence numbers range from 0 to 2k – 1. Sender maintains a list of sequence numbers that it is allowed to send (sender window).

The size of the sender's window is at most 2k – 1. The sender is provided with a buffer equal to the window size. Receiver also maintains a window of size 2k – 1. The receiver acknowledges a frame by sending an ACK frame that includes the sequence number of the next frame expected.

This also explicitly announces that it is prepared to receive the next N frames, beginning with the number specified. This scheme can be used to acknowledge multiple frames. It could receive frames 2, 3, 4 but withhold ACK until frame 4 has arrived. By returning an ACK with sequence number 5, it acknowledges frames 2, 3, 4 in one go. The receiver needs a buffer of size 1.

Sliding window algorithm is a method of flow control for network data transfers. TCP, the Internet's stream transfer protocol, uses a sliding window algorithm.

A sliding window algorithm places a buffer between the application program and the network data flow. For TCP, the buffer is typically in the operating system kernel, but this is more of an implementation detail than a hard-and-fast requirement.

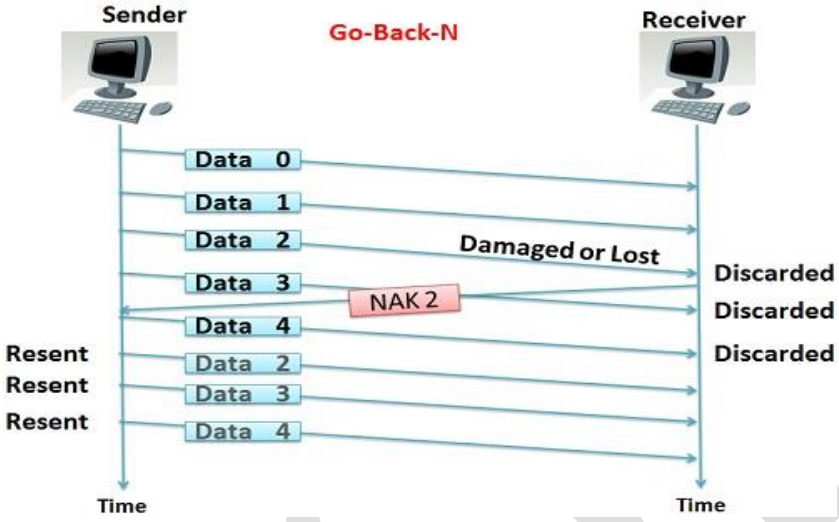
Data received from the network is stored in the buffer, from where the application can read at its own pace. As the application reads data, buffer space is freed up to accept more input from the network. The window is the amount of data that can be "read ahead" - the size of the buffer, less the amount of valid data stored in it. Window announcements are used to inform the remote host of the current window size.

An example of a sliding window in packet transmission is one in which, after the sender fails to receive an acknowledgement for the first transmitted packet, the sender "slides" the window, i.e. resets the window, and sends a second packet. This process is repeated for the specified number of times before the sender interrupts transmission. Sliding window is sometimes (loosely) called *acknowledgement delay period*.

Go-Back-N Protocol and “Selective Repeat Protocol” are the sliding window protocols. The sliding window protocol is primarily an error control protocol, i.e. it is a method of error detection and error correction. The basic difference between go-back-n protocol and selective repeat protocol is that the “go-back-n protocol” retransmits all the frames that lie after the frame which is damaged or lost. The “selective repeat protocol” retransmits only that frame which is damaged or lost.

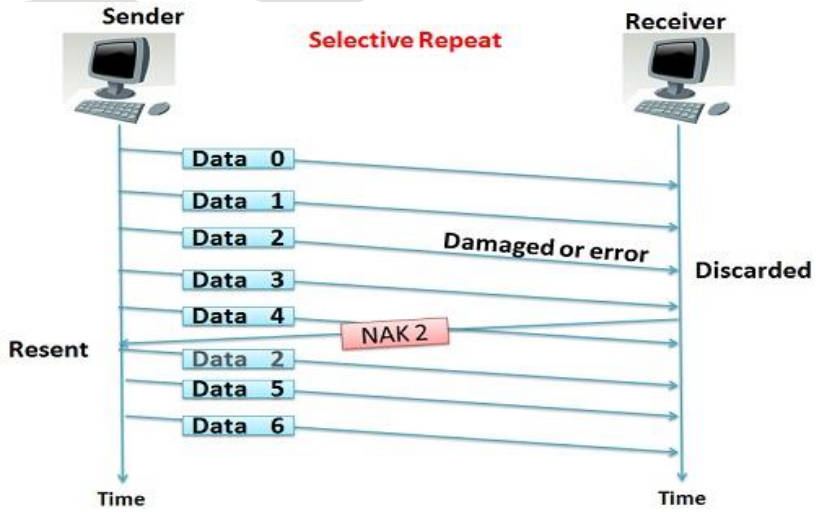
Go back N ARQ

In the Go-Back-N Protocol, the sequence numbers are modulo 1!",
where*m*is the size of the sequence number field in bits.



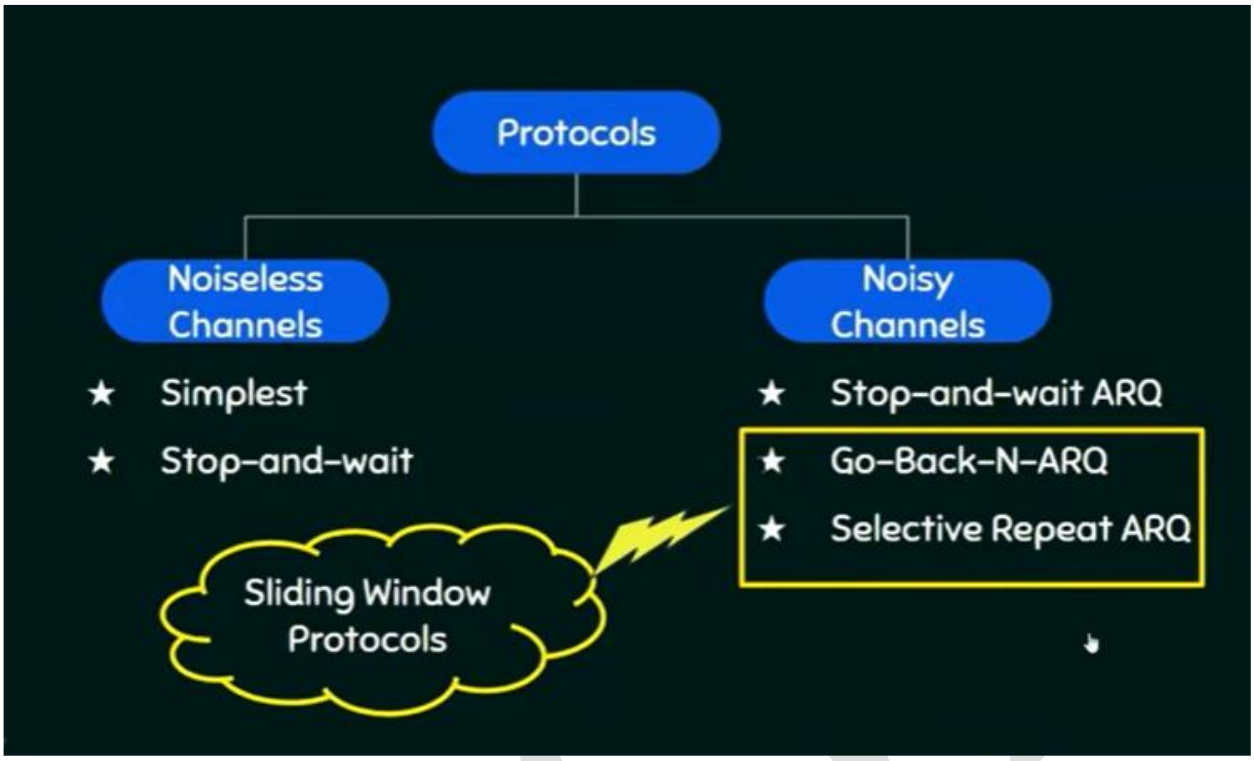
Selective Repeat ARQ

Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not resend *N* frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.



Key Differences Between Go-Back-N and Selective Repeat

1. Go-Back-N protocol is design to retransmit all the frames that are arrived after the damaged or a lost frame. On the other hand, Selective Repeat protocol retransmits only that frame that is damaged or lost.
2. If the error rate is high i.e. more frames are being damaged and then retransmitting all the frames that arrived after a damaged frame waste the lots of bandwidth. On the other hand, selective repeat protocol re-transmits only damaged frame hence, minimum bandwidth is wasted.
3. All the frames after the damaged frame are discarded and the retransmitted frames arrive in a sequence from a damaged frame onwards, so, there is less headache of sorting the frames hence it is less complex. On the other hand only damaged or suspected frame is retransmitted so, extra logic has to be applied for sorting hence, it is more complicated.
4. Go-Back-N has a window size of N-1 and selective repeat have a window size <=(N+1)/2.
5. Neither sender nor receiver need the sorting algorithm in Go-Back-N whereas, receiver must be able to sort the as it has to maintain the sequence.
6. In Go-Back-N receiver discards all the frames after the damaged frame hence, it don't need to store any frames. Selective repeat protocol does not discard the frames arrived after the damaged frame instead it stores those frames till the damaged frame arrives successfully and is sorted in a proper sequence.
7. In selective repeat NAK frame refers to the damaged frame number and in Go-Back-N, NAK frame refers to the next frame expected.
8. Generally the Go-Back-N is more is use due to its less complex nature instead of Selective Repeat protocol.



Go-Back-N ARQ

- ★ Go - Back - N ARQ uses the concept of protocol pipelining i.e. the sender can send multiple frames before receiving the acknowledgment for the first frame.
- ★ There are finite number of frames and the frames are numbered in a sequential manner.
- ★ The number of frames that can be sent depends on the window size of the sender.
- ★ If the acknowledgment of a frame is not received within an agreed upon time period, all frames in the current window are transmitted.

Go-Back-N ARQ

★ The size of the sending window determines the sequence number of the outbound frames.

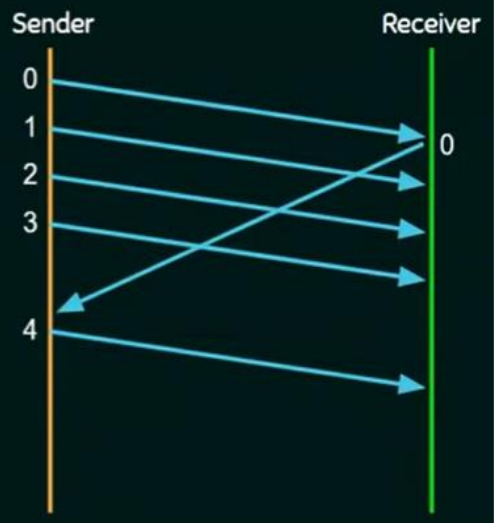
★ N – Sender's Window Size.

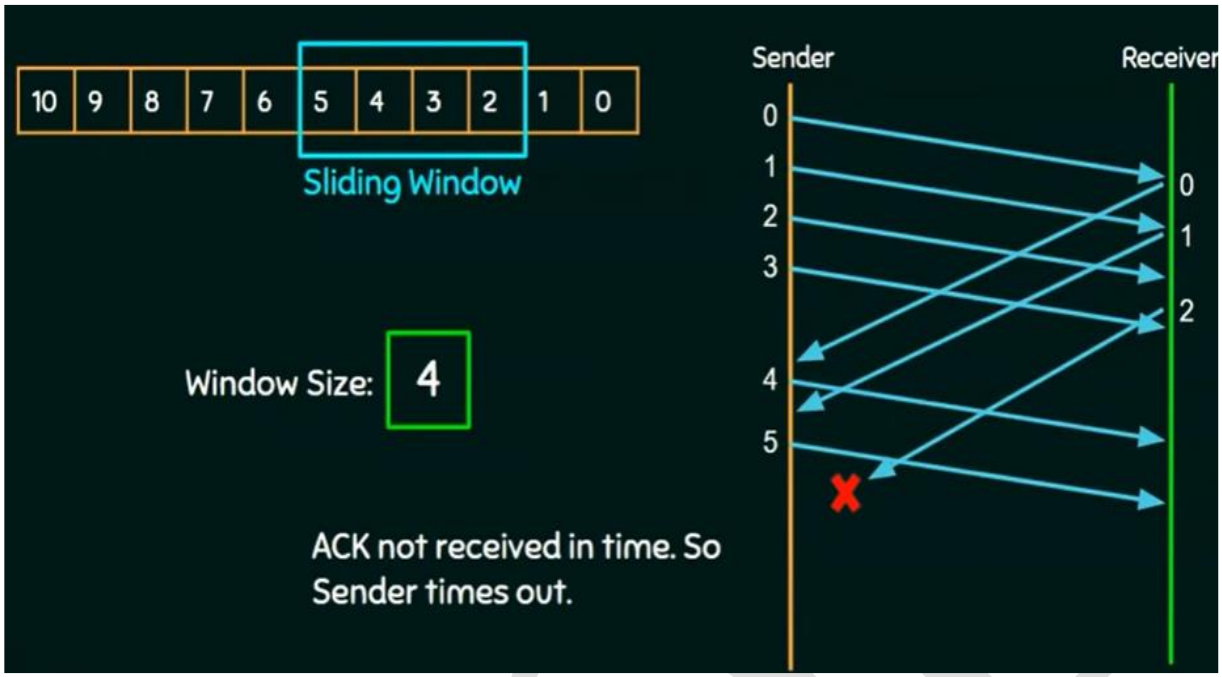
★ For example, if the sending window size is 4 (2^2), then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so on.

★ The number of bits in the sequence number is 2 to generate the binary sequence 00, 01, 10, 11.

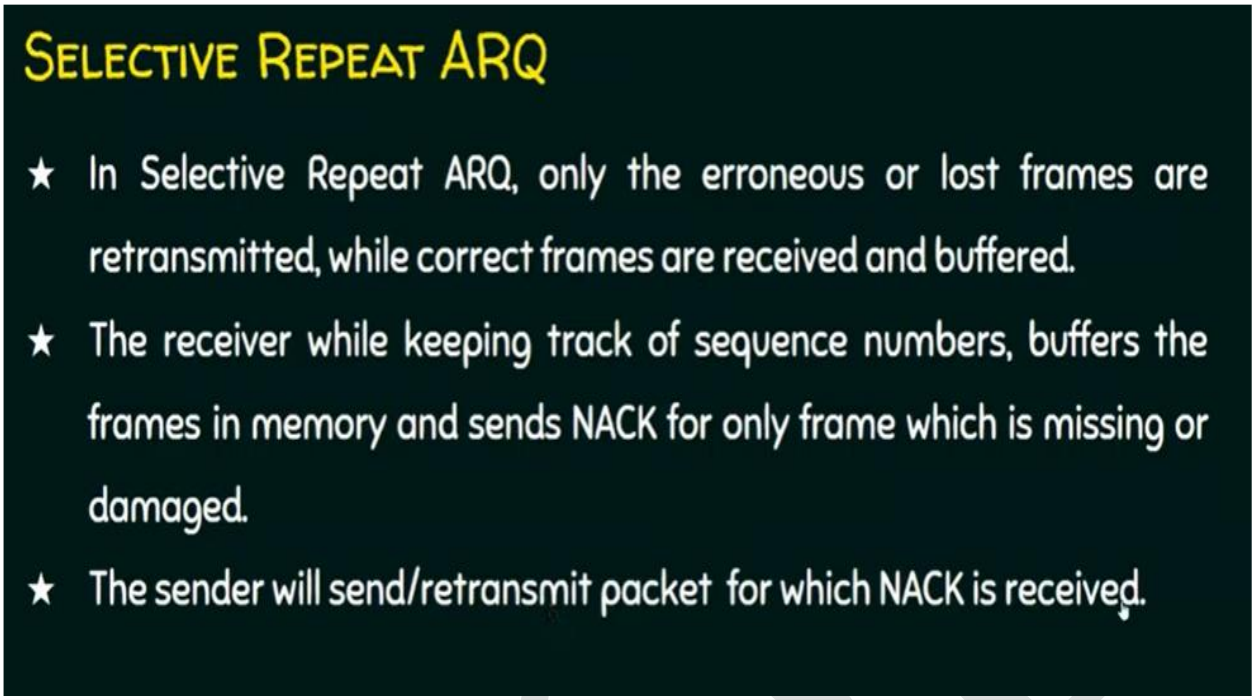
Working of Go-Back-N ARQ

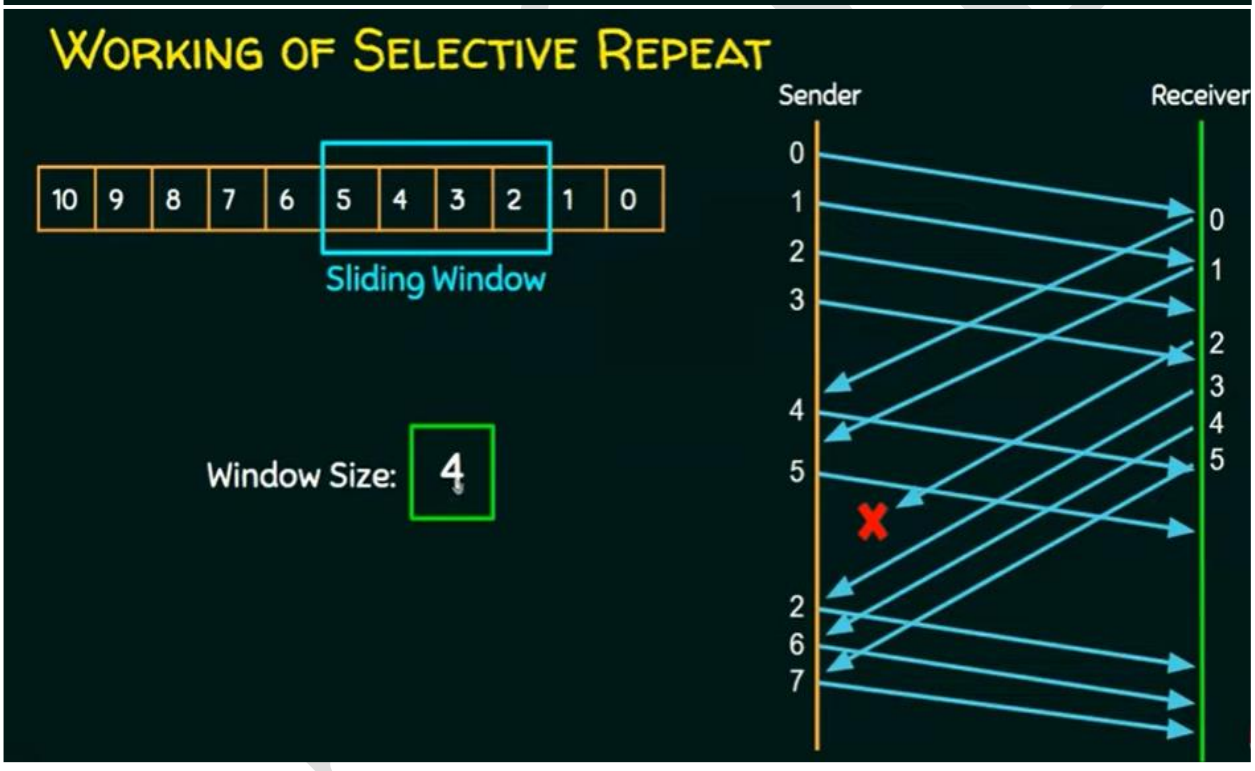
Window Size: 4




10 9 8 7 6 5 4 3 2 1 0
Sliding Window
Window Size: 4
ACK not received in time. So Sender times out.
Sender
Receiver
0 1 2 3 4 5
0 1 2


WORKING OF GO-BACK-N ARQ
10 9 8 7 6 5 4 3 2 1 0
Sliding Window
Go-Back to 2
Window Size: 4
Sender
Receive
0 1 2 3 4 5 2 3 4 5
0 1 2





Conclusion: Hence we have implemented of sliding window protocol(Go back N and Selective Repeat).

Assignment B6

Title: Subnetting & subnet mask.

Objective/s: To implement subnetting & find out subnet mask.

Problem statement Write a program to demonstrate subnetting & to find out subnet mask.

Software&/hardware requirement: Java/Pentium IV and above

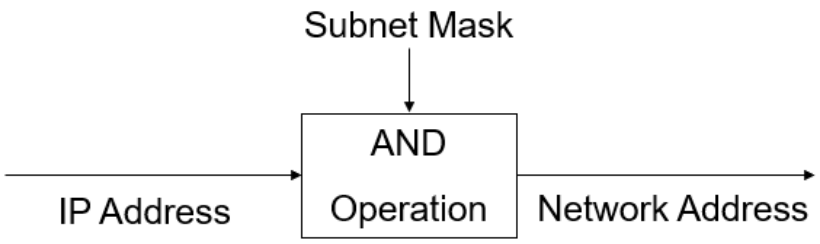
Theory: //Here write answers to FAQ questions given below.

1) What is subnetting? Why it is needed?
2) What is super-netting? Where it is used?
3) What is subnet mask.
4) How to calculate 1st address? Explain all methods with one example.
5) How to calculate last address? Explain all methods with one example.
6) How to find out total no. of IP addresses in the group? Explain all methods with one example..
7) What are the different classes of IP addresses? State its range.
8) What is CIDR concept? Explain with example.
9) What is the difference between IPV4 & IPV6 addresses?
10) What is classful & classless addressing?

Conclusion: Thus, learnt to demonstrate subnetting & understand the concept of subnet mask.

Subnet Mask:

- An IP address has 2 parts:
  - The Network identification.
  - The Host identification.
- 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



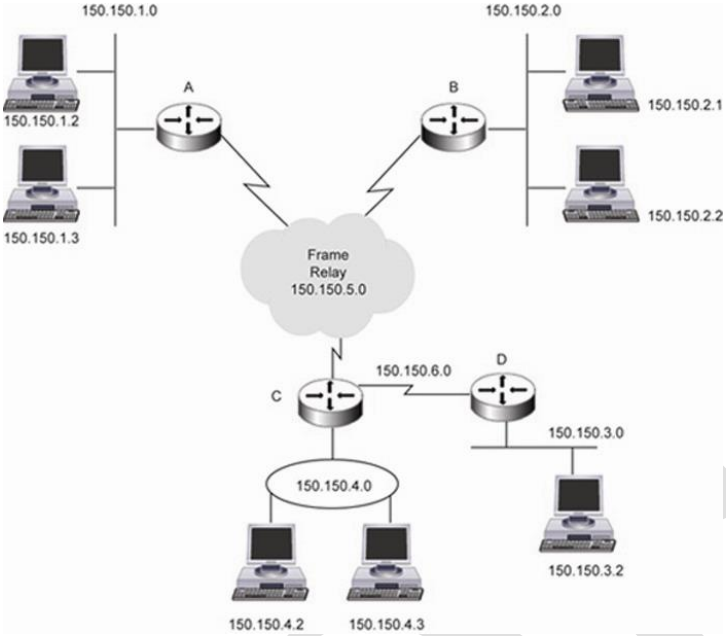
Default Mask:

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

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

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


150.150.1.0
150.150.2.0
A
B
150.150.1.2
150.150.2.1
150.150.1.3
150.150.2.2
Frame
Relay
150.150.5.0
150.150.6.0
D
C
150.150.3.0
150.150.4.0
150.150.3.2
150.150.4.2
150.150.4.3

Assignment B7

Title: Link state / Distance vector routing protocol

Objective/s: To implement Link state / distance vector routing

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

Software&/hardware requirement: Java/Pentium IV and above

Theory: //Here write answers to FAQ questions given below.

1. What Is Distance-vector Routing Protocol?
2. State the difference between Distance vector routing and Link State routing
3. Differentiate User Mode from Privileged Mode
4. What is unicast routing?
5. Explain RIP, OSPF and BGP.
6. what are the different types of metrics used in routing protocols
7. What type of metric is used by RIP?
8. What are the disadvantages of Distance Vector routing?
9. What is count to infinity problem.
10. Explain split horizon and route poisoning.

Conclusion: Thus, learnt to demonstrate distance vector routing protocol.

Theory:

Overview

The Link State Routing Algorithm is an interior protocol used by every router to share information or knowledge about the rest of the routers on the network. The link state routing algorithm is distributed by which every router computes its routing table. With the knowledge of the network topology, a router can make its routing table. The routing table created by each router is exchanged with the rest of the routers present in the network which helps in faster and more reliable delivery of data. This information exchange only occurs when there is a change in the information. Hence, the link state routing algorithm is effective.

Scope

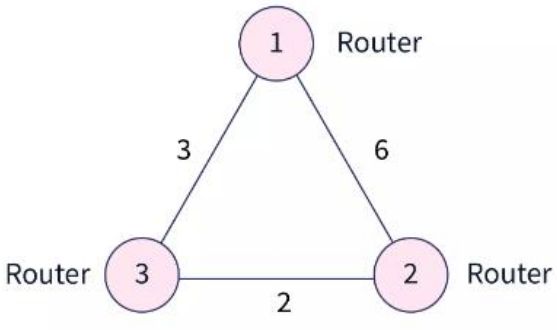
- Introduction to the Link State Routing Algorithm.
- Introduction to the Link State Routing Protocols.
- Phases and Functions of the Link State Routing Algorithm.

What is a Link State Routing Algorithm?

Before learning about the Link State Routing Algorithm, let us briefly discuss the term Routing.

Routing is a process of establishing the routes that data packets must follow to reach the destination. In this process, a routing table is created, which contains the information regarding routes that data packets follow. Now, various routing algorithms are there which are used to decide the best optimal route that the incoming data packet must be transmitted on.

The best or optimal path is the path from source to destination router, having the least connection cost. For example, refer to the routers shown in the image below.



If a packet needs to be transmitted from the Router-1 to Router-2, then it can follow two paths.

1. Directly from Router-1 to Router-2, the cost of this traveling is 6.
2. It can also go from Router-1 to Router-2, via path: Router-1 --> Router-3 --> Router-2. The cost of this traveling is (2 + 3) = 5.

So, the data packet will be sent from the second path i.e. Router-1 --> Router-3 --> Router-2.

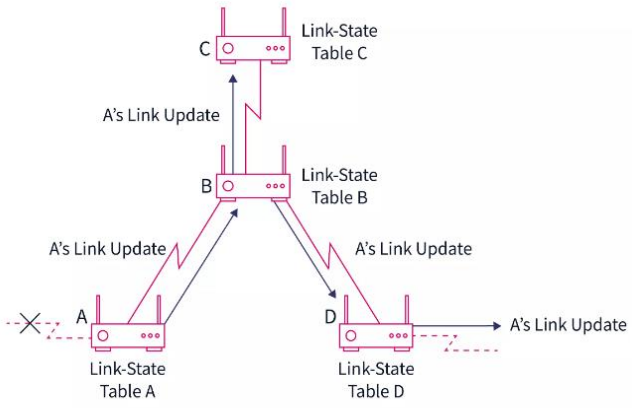
The Link State Routing Algorithm is an interior protocol used by every router to share information or knowledge about the rest of the routers on the network. The link state routing algorithm is distributed by which every router computes its routing table.

With the knowledge of the network topology, a router can make its routing table. Now, for developing the routing table, a router uses a shortest path computation algorithm like Dijkstra's algorithm along with the knowledge of the topology. The routing table created by each router is exchanged with the rest of the routers present in the network, which helps in faster and more reliable delivery of data.

A router does not send its entire routing table with the rest of the routers in the inter-network. It only sends the information of its neighbors. A router broadcasts this information and contains information about all of its directly connected routers and the connection cost.

Now, the process of transferring the information about a router's neighbors is termed flooding. A router transfers the information to all the inter-network routers except its neighbors. Every router that receives the information sends the information copies to all its neighbors. In this way, all the routers of the inter-connected network have the same copy of the information.

This information exchange only occurs when there is a change in the information. Hence, the link state routing algorithm is effective. Refer to the image below for the basic overview of the router and updation done by the link state routing algorithm.



Link State Routing Algo

Assignment B8

Title: RIP

Objective/s: To learn RIP protocol using packet tracer tool.

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

Software&/hardware requirements: cisco packet tracer tool.

Theory: //Here write answers to FAQ questions given below.

1. What is RIP?
2. What is route poisoning?
3. What is the default routing update period for RIP?
4. Which transport layer protocol does RIP use? And which port number is associated with RIP?
5. What is the major benefit of dynamic routing protocol like RIP over Static route?
6. What route entry will be assigned to dead or invalid route in case of RIP?
7. Explain the message types used in RIP?
8. Which type of routing algorithm does RIP use?
9. Which metric does RIP use?
10. Can RIP be a preferred Dynamic Routing protocol for large networks?

Conclusion: Studied setup and configuration of RIP on the network of 3 routers using packet tracer.

Theory:

Introduction:-

ROUTING PROTOCOLS: RIP [ROUTING INFORMATION PROTOCOL]

RIP is an open standard routing protocol. It is a distance vectored routing protocols. It is a class-full routing protocol where updates are exchanged through broadcast. The routing table is exchanged every 30 seconds among the routers in the inter-network. The RIP protocol uses hop count as the metric to find the shortest path but the maximum allowable hop count is 15 by default. The RIP protocols is used only for a small network and is ineffective for a large network. The Administrative Distance of RIP is 120.

Some of the of key features of RIP protocol are:

It supports maximum 15 hops in a path.

It uses hops count metric to calculate the best path from a source to a destination network.

It sends routing updates (entire routing table) after every 30 seconds and when the network changes.

It uses UDP broadcast packets to exchange routing information.

The Administrative Distance (AD) value of the RIP protocol is 120.

It has two versions: RIPv1 and RIPv2.

RIP TIMERS To manage the routing performance, RIP uses four different kinds of timers:

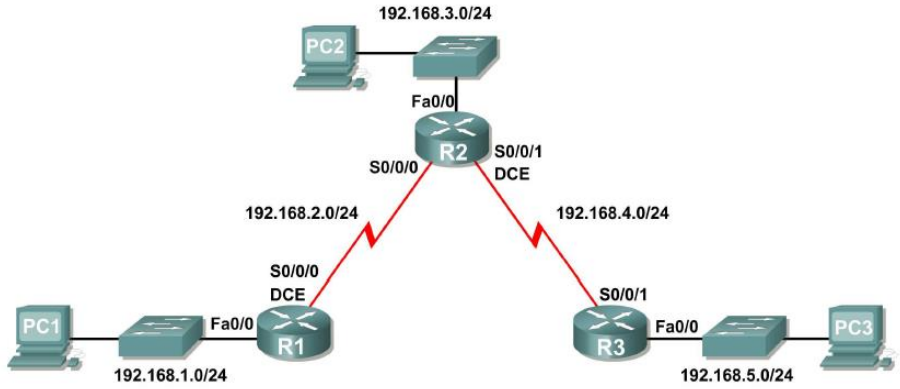
- Update timer: It is the time interval after which a router sends it's a copy of the routing table as update to the neighbor routers. The update timer is 30 sec by default.
- Invalid time: It is the time interval after which a router understands that the path to a network is invalid or becomes invalid. The invalid timer is 180 sec by default.
- Hold-down timer: It specifies the amount of time for which the information about the poorer routes are ignored. The hold-down timer is 180sec by default.
- Flush timer: It is the time before the invalid route is purged from the routing table. The flush timer is 240 sec by default.

Dis-advantages of RIP

- It uses more bandwidth as updates are exchanged every 30 seconds where each update contains the complete routing table of the router.
- It does not uses bandwidth as the metric for calculation of the shortest path.
- RIP has a very slow convergence.

- RIP implementation can lead to routing loops in the network.
- RIP is only applicable to small network and is inefficient for larger networks.

Topology Diagram



Scenario A: Running RIPv1 on Classful Networks

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Fa0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	192.168.2.1	255.255.255.0	N/A
R2	Fa0/0	192.168.3.1	255.255.255.0	N/A
	S0/0/0	192.168.2.2	255.255.255.0	N/A
	S0/0/1	192.168.4.2	255.255.255.0	N/A
R3	Fa0/0	192.168.5.1	255.255.255.0	N/A
	S0/0/1	192.168.4.1	255.255.255.0	N/A
PC1	NIC	192.168.1.10	255.255.255.0	192.168.1.1
PC2	NIC	192.168.3.10	255.255.255.0	192.168.3.1
PC3	NIC	192.168.5.10	255.255.255.0	192.168.5.1

Task 1: Prepare the Network.

Step 1: Cable a network that is similar to the one in the Topology Diagram.

You can use any current router in your lab as long as it has the required interfaces shown in the topology.

Step 2: Clear any existing configurations on the routers.

Task 2: Perform Basic Router Configurations.

Perform basic configuration of the R1, R2, and R3 routers according to the following guidelines:

1. Configure the router hostname.
2. Disable DNS lookup.
3. Configure an EXEC mode password.

4. Configure a message-of-the-day banner.
5. Configure a password for console connections.
6. Configure a password for VTY connections.

Task 3: Configure and Activate Serial and Ethernet Addresses.
Step 1: Configure interfaces on R1, R2, and R3.
Configure the interfaces on the R1, R2, and R3 routers with the IP addresses from the table under the
Topology Diagram.
Step 2: Verify IP addressing and interfaces.
Use the **show ip interface brief** command to verify that the IP addressing is correct and that the
interfaces are active.
When you have finished, be sure to save the running configuration to the NVRAM of the router.
Step 3: Configure Ethernet interfaces of PC1, PC2, and PC3.
Configure the Ethernet interfaces of PC1, PC2, and PC3 with the IP addresses and default gateways
from the table under the Topology Diagram.
Step 4: Test the PC configuration by pinging the default gateway from the PC.

Task 4: Configure RIP.
Step 1: Enable dynamic routing.
To enable a dynamic routing protocol, enter global configuration mode and use the **router** command.
Enter **router ?**at the global configuration prompt to a see a list of available routing protocols on your
router.
To enable RIP, enter the command **router rip** in global configuration mode.
R1(config)#**router rip**
R1(config-router)#
Step 2: Enter classful network addresses.
Once you are in routing configuration mode, enter the classful network address for each directly
connected network, using the **network** command.
R1(config-router)#**network 192.168.1.0**
R1(config-router)#**network 192.168.2.0**
R1(config-router)#

The **network** command:
□□Enables RIP on all interfaces that belong to this network. These interfaces will now both send and
receive RIP updates.
□□Advertises this network in RIP routing updates sent to other routers every 30 seconds.
When you are finished with the RIP configuration, return to privileged EXEC mode and save the current
configuration to NVRAM.
R1(config-router)#**end**

%SYS-5-CONFIG_I: Configured from console by console
R1#**copy run start**
Step 3: Configure RIP on the R2 router using the router rip and network commands.
R2(config)#**router rip**
R2(config-router)#**network 192.168.2.0**
R2(config-router)#**network 192.168.3.0**
R2(config-router)#**network 192.168.4.0**
R2(config-router)#**end**
%SYS-5-CONFIG_I: Configured from console by console
R2#**copy run start**
When you are finished with the RIP configuration, return to privileged EXEC mode and save the current
configuration to NVRAM.
Step 4: Configure RIP on the R3 router using the router rip and network commands.
R3(config)#**router rip**
R3(config-router)#**network 192.168.4.0**
R3(config-router)#**network 192.168.5.0**
R3(config-router)#**end**
%SYS-5-CONFIG_I: Configured from console by console
R3# **copy run start**
When you are finished with the RIP configuration, return to privileged EXEC mode and save the current
configuration to NVRAM.

Task 5: Verify RIP Routing.
Step 1: Use the show ip route command to verify that each router has all of the networks in the
topology entered in the routing table.
Routes learned through RIP are coded with an **R** in the routing table. If the tables are not converged as
shown here, troubleshoot your configuration. Did you verify that the configured interfaces are active? Did
you configure RIP correctly? Return to Task 3 and Task 4 to review the steps necessary to achieve
convergence.

R1#**show ip route**
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 192.168.1.0/24 is directly connected, FastEthernet0/0
C 192.168.2.0/24 is directly connected, Serial0/0/0
R 192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:04, Serial0/0/0

R 192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:04, Serial0/0/0
R 192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:04, Serial0/0/0
R1#
R2#**show ip route**
<Output omitted>
R 192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:22, Serial0/0/0
C 192.168.2.0/24 is directly connected, Serial0/0/0
C 192.168.3.0/24 is directly connected, FastEthernet0/0
C 192.168.4.0/24 is directly connected, Serial0/0/1
R 192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:23, Serial0/0/1
R2#
R3#**show ip route**
<Output omitted>
R 192.168.1.0/24 [120/2] via 192.168.4.2, 00:00:18, Serial0/0/1
R 192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:18, Serial0/0/1
R 192.168.3.0/24 [120/1] via 192.168.4.2, 00:00:18, Serial0/0/1
C 192.168.4.0/24 is directly connected, Serial0/0/1
C 192.168.5.0/24 is directly connected, FastEthernet0/0
R3#

Step 2: Use the show ip protocols command to view information about the routing processes.

The **show ip protocols** command can be used to view information about the routing processes that
are occurring on the router. This output can be used to verify most RIP parameters to confirm that:

- RIP routing is configured
- The correct interfaces send and receive RIP updates
- The router advertises the correct networks
- RIP neighbors are sending updates

R1#**show ip protocols**
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 16 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
Interface Send Recv Triggered RIP Key-chain
FastEthernet0/0 1 2 1
Serial0/0/0 1 2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
192.168.1.0
192.168.2.0
Passive Interface(s):
Routing Information Sources:

Gateway Distance Last Update
192.168.2.2 120
Distance: (default is 120)
R1#
R1 is indeed configured with RIP. R1 is sending and receiving RIP updates on FastEthernet0/0 and
Serial0/0/0. R1 is advertising networks 192.168.1.0 and 192.168.2.0. R1 has one routing information
source. R2 is sending R1 updates.

Step 3: Use the debug ip rip command to view the RIP messages being sent and received.

Rip updates are sent every 30 seconds so you may have to wait for debug information to be displayed.

R1#**debug ip rip**
R1#RIP: received v1 update from 192.168.2.2 on Serial0/0/0
192.168.3.0 in 1 hops
192.168.4.0 in 1 hops
192.168.5.0 in 2 hops
RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (192.168.1.1)
RIP: build update entries
network 192.168.2.0 metric 1
network 192.168.3.0 metric 2
network 192.168.4.0 metric 2
network 192.168.5.0 metric 3
RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (192.168.2.1)
RIP: build update entries
network 192.168.1.0 metric 1

The debug output shows that R1 receives an update from R2. Notice how this update includes all the
networks that R1 does not already have in its routing table. Because the FastEthernet0/0 interface
belongs to the 192.168.1.0 network configured under RIP, R1 builds an update to send out that interface.
The update includes all networks known to R1 except the network of the interface. Finally, R1 builds an
update to send to R2. Because of split horizon, R1 only includes the 192.168.1.0 network in the update.

Step 4: Discontinue the debug output with the undebug all command.

R1#**undebug all**
All possible debugging has been turned off

Assignment B9

Title: TCP Socket Programming

Objective/s: To Implement TCP sockets to transfer files.

Problem statement: Write a program using TCP Sockets for

a. Say Hello to each other
b. File transfer
c. Calculator

Software&/hardware requirements: JAVA/C (strike the one not used} /Pentium IV and above

Theory: //Here write answers to FAQ questions given below.

1. What are applications of TCP?
2. What is socket address?
3. What is mean by Socket programming?
4. What Is the Difference Between TCP And UDP?
5. What are the socket primitives? And What are socket primitives for TCP?
6. Which socket is used in TCP?
7. What is the type of data used in TCP socket?
8. What is multiprogramming?
9. Explain AF_INET Family.
10. How sockets can be used to write client server applications using a connection oriented client server technique.

Conclusion: Thus, we have successfully implemented the socket programming for TCP.

THEORY:

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

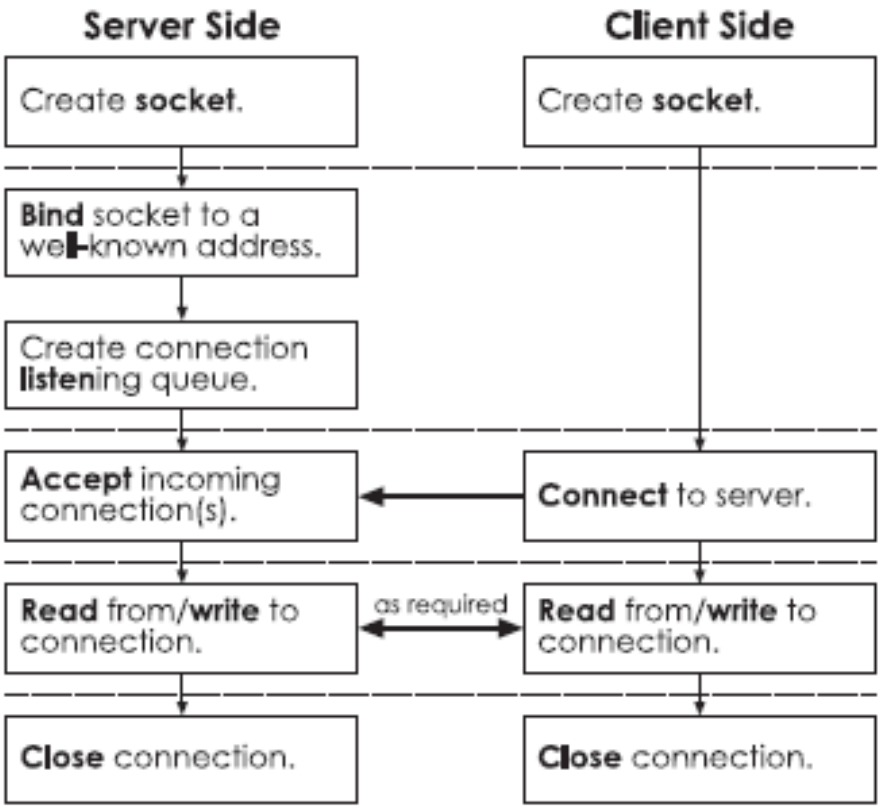
Most interprocess communication uses the client server model. These terms refer to the two processes which will be communicating with each other. One of the two processes, the client, connects to the other process, the server, typically to make a request for information. A socket is one end of an interprocess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



Sockets are one of the most important IPC mechanisms on UNIX. Originally introduced in 4.2BSD as a generalization of pipes, they later became the basis for the UNIX networking subsystem. Sockets are the only IPC mechanism that allows communication between processes running on different machines. Essentially, it is an end-point of communication which may be bound to a name. But enough with the bland introduction. A socket is just a way to allow processes to talk to one another.

TCP and UDP

As you should remember from networking, on top of IP, there are two major transport protocols on top of which all other protocols are built: TCP and UDP These act as transportation mechanisms for other, higher-level, protocols. TCP is a reliable, connection-oriented protocol that transmits data as a stream of bytes. UDP, on the other hand, is an unreliable, connectionless protocol that sends data in chunks called datagrams.

Creating a socket

Both the client and server need to create a socket before they can do anything. The client uses its socketto connect to a server whilst the server uses its socket for listening for new connections. Socket creation is done using the socket() call

socket()

Creates a new socket. Returns a file descriptor representing the socket end-point, or-1, if an error occurs.

int socket(int af, int type, int protocol);

af : is the address family to use with the socket. This could be either

AF_UNIXfor the UNIX address family(for local IPC), or

AF_INETfor the Internet address family (for network communication). TCP requires that you useAF_INET,

type: is the socket type. This can be SOCK_STREAM for connection-oriented, stream-based protocols like TCP, SOCK_DGRAM for datagram-based, connectionless protocols like UDP, or SOCK_RAW whereyou want to use your own transportation protocol

protocol :is the transport protocol to use. It's best to pass0, which lets the system decide.

Binding the socket to a well-known address

For clients to connect to a server, they need to know its address, but sockets are created without an address of their own and so must be assigned to one that the client will know about. This is done by binding a socket to a well-known address on the system. For instance, web-servers are usually bound to port 80 as this is the well-known port for the HTTP protocol

struct sockaddr

This is a generic socket address structure provided to allow flexibility in passing socket addresses to functions.

```
struct sockaddr
{
    unsigned short sa_family;   // Address family tag.
    char           sa_data[14]; // Padding.
};
```

struct sockaddr_in

structsockaddr_inis a specialised version ofstructsockaddrespecially for the AF_INETaddress family.

```
struct sockaddr_in
{
    unsigned short sin_family;  // Set to AF_INET.
    unsigned short sin_port;    // Port number to bind to.
    struct in_addr sin_addr;    // IP address.
    char           sin_zero[8]; // Padding.
};
```

structin_addr

structin_addrrepresents an IP address. Why this structure exists and wasn't just incorporated directly intostructsockaddr_inis anybody's guess.Setting its one field, s_addr, toINADDR_ANYwill leave it up to the server to choose an appropriate host IP address, and this is usually the best thing to do.

```
struct in_addr
{
    unsigned long s_addr;   // IP address.
};
```

bind()

Binds a socket to a well-known address. This will return0if successful and-1 if not.

int bind(intfd, structsockaddr* addr, intlen);

fd: is the file descriptor of the socket to bind.
addr: points to the address to bind the socket to.
len: is the length of the address in bytes.

intBindSocket(intfd, unsigned short port)

```
{
    struct sockaddr_in addr;

    addr.sin_family      = AF_INET;
    addr.sin_addr.s_addr = INADDR_ANY;  // Let the host decide.
    addr.sin_port        = htons(port); // Port to bind to.

    return bind(fd, (struct sockaddr*) &addr, sizeof(addr));
}
```

listen()
Makes the socket listen for incoming connections, and sets up a connection queue for the socket.

int listen(intfd, intlen);

fd: is the file descriptor of the socket to put in passive mode.
Len: is the length of the connection queue **5**is the usual value used.

gethostbyname()

Allows you to discover a host's details (including its address) by specifying its name. This returns a structure specifying the host's details, or NULL if it fails.

structhostent* gethostbyname(char* name);

connect()
Connects a socket to a given server, putting the socket inactive mode. Returns0 if successful, else-1

int connect(intfd, structsockaddr* addr, intlen);

fd: is the file descriptor of the socket to connect.
Addr: points to the address of the server to connect to.
len: is the length in bytes of the address

Accepting incoming connections
After the connection queue has been created, the server can accept incoming connections from clients wishing to talk to it. To accept one, it must call

accept()

Accepts a single incoming connection. Returns a file descriptor corresponding to the new connection, or-1if an error occurs.

int accept(intfd, structsockaddr* addr, int* len);

fd: is the file descriptor of the socket listening for incoming connections.

addr: is an address structure to hold the address of the client making the connection. Pass NULL here if you don't care about getting this information.

len: is pointer to a variable holding the length in bytes of the address structure passed in addr. On return-ing, this will hold the actual length of the client address. If you passed in NULLtoaddr, passNULLin here too.

Reading and Writing data
After the client has connected to the server and the server has accepted the connection, they can start
sending data back and forth. This is done with good old
read()
and
write()
. There are, however,
some caveats attached to reading data, however.

write()
Writes data to a file descriptor.

int write(intfd, void* buf, unsigned int n);

fd: is the file descriptor to write to.
buf: is a buffer containing data to write.
n: is the number of bytes from the buffer to write.

read()
Reads data from a file descriptor. It returns the number of bytes actually read, or-1 if an error occurs.

int read(intfd, void* buf, unsigned int n);
fd:is the file descriptor to read from.
buf:is a buffer to write the data read to.
n:is the size in bytes of the buffer.

Closing the socket

Once the client or server is finished with a socket, it should call **close()** to deallocate it.

close()

Closes a file descriptor.

int close(intfd);

fd: is the file descriptor to close.

No.	TCP	UDP
1	This Connection oriented protocol	This is connection-less protocol
2	The TCP connection is byte stream	The UDP connection is a message stream
3	It does not support multicasting and broadcasting	It supports broadcasting
4	It provides error control and flow control	The error control and flow control is not provided
5	TCP supports full duplex transmission	UDP does not support full duplex transmission
6	It is reliable service of data transmission	This is an unreliable service of data transmission
7	The TCP packet is called as segment	The UDP packet is called as user datagram.

Assignment B10

Title: UDP Socket Programming

Objective/s: To Implement UDP sockets to transfer files.

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

Software&/hardware requirements: JAVA/C (strike the one not used} /Pentium IV and above

Theory: //Here write answers to FAQ questions given below.

11. What are raw sockets?
12. What is socket address?
13. A) What is the need for port number? B) What is the size of port number? C) How are port numbers classified? D)what is the role of IANA
14. What Is the Difference Between TCP And UDP?
15. What is the difference between connection-oriented and connectionless services?
16. Give examples for reliable and unreliable services?
17. Can UDP be used in real time data? Justify
18. What application uses UDP?
19. What are the socket primitives? And What are socket primitives for UDP?
20. Which socket is used in UDP?
21. What is the difference between TFTP and FTP

Conclusion: Thus, we have successfully implemented the socket programming for UDP.

THEORY:

UDP:

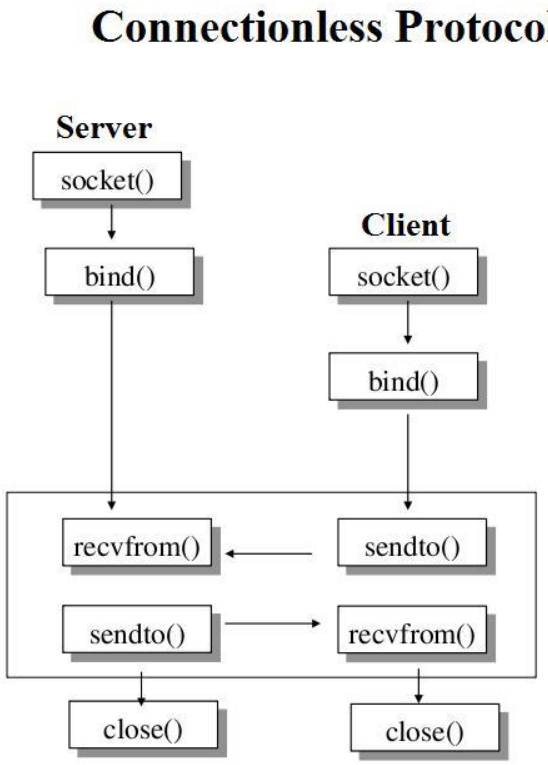
UDP (User Datagram Protocol) is a communication protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). UDP is an alternative to the Transmission Control Protocol (TCP) and, together with IP, is sometimes referred to as UDP/IP. Like the Transmission Control Protocol, UDP uses the Internet Protocol to actually get a data unit (called a datagram) from one computer to another. Unlike TCP, however, UDP does not provide the service of dividing a message into packets (datagrams) and reassembling it at the other end. Specifically, UDP doesn't provide sequencing of the packets that the data arrives in. This means that the application program that uses UDP must be able to make sure that the entire message has arrived and is in the right order. Network applications that want to save processing time because they have very small data units to exchange (and therefore very little message reassembling to do) may prefer UDP to TCP. The Trivial File Transfer Protocol (TFTP) uses UDP instead of TCP.

With a UDP socket a connection is NOT made, instead the sender just issues a message to its destination and hopes it gets there! The message uses a datagram of fixed length, often termed a record. Since there is no connection between client and server the client can send a datagram to one server and then immediately send a datagram to another server using the same socket UDP is a connectionless protocol.

Trivial File Transfer Protocol (TFTP) is a simple, lock-step, file transfer protocol which allows a client to get from or put a file onto a remote host.

TFTP is a simple protocol for transferring files, implemented on top of the UDP/IP protocols using IANA registered port number 69. TFTP was designed to be small and easy to implement,

and therefore it lacks most of the advanced features offered by more robust file transfer protocols. TFTP only reads and writes files from or to a remote server. It cannot list, delete, or rename files or directories and it has no provisions for user authentication. Today TFTP is generally only used on local area networks (LAN).



Assignment C11

Title: DNS Lookup

Objective: To understand the concept of DNS Lookup.

Problem statement: Write a program for DNS lookup. Given an IP address as input, it should return URL & vice versa.

Software&/ hardware requirements: Python / Java

Theory: //here, write answers to FAQs given below.

1. What is Name Space n explain its types.
2. What is DNS?
3. What are different DNS zones?
4. What is DNS server & what are the main types of DNS server.
5. Why there is a need of using DNS?
6. How DNS lookup process work, describe some steps.
7. Which TCP/IP port is used by DNS server?
8. What are DNS resolution techniques?
9. Draw n explain DNS message header format.
10. What are the types of records in DNS?

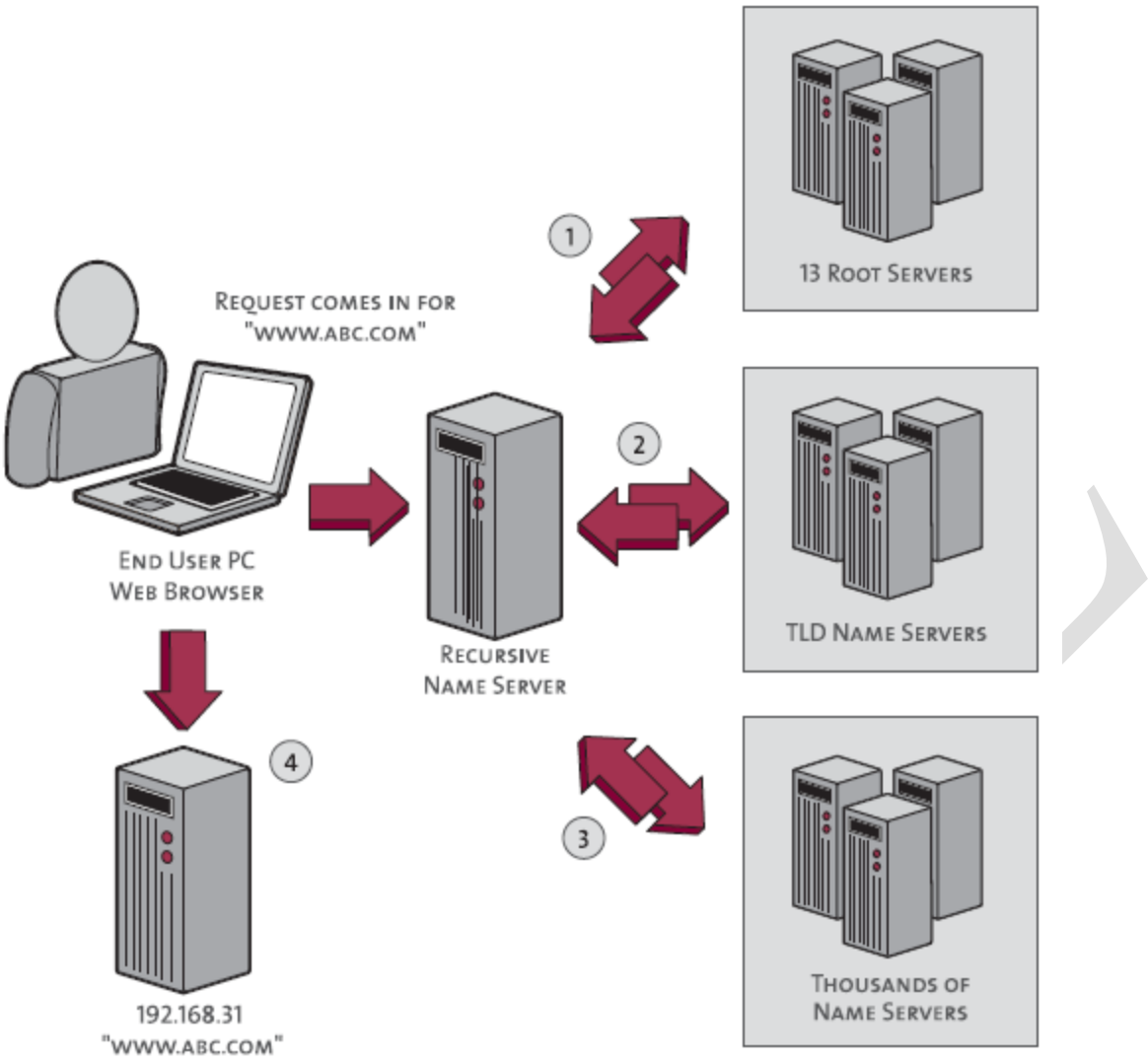
Conclusion: Studied and analyzed the concept of DNS Lookup

Theory:-

A DNS lookup, in a general sense, is the process by which a DNS record is returned from a DNS server. This is like looking up a phone number in a phone book - that is why it is referred to as a "lookup". Interconnected computers, servers and smart phones need to know how to translate the email addresses and domain names people use into meaningful numerical addresses. A DNS lookup performs this function.

The basic idea of DNS is that humans can't easily remember long strings of digits like machines can, but can much more easily remember words. So, when you type in a domain name like www.techopedia.com, the request is forwarded to a DNS server (whether locally or at an ISP), which returns the corresponding IP address. This address is then used by all the computers and routers to channel the request and responses of a user's session. The result is the user sees web pages as expected or has email show up in an in-box.

The two types of DNS lookups are forward DNS lookups and reverse DNS lookups.Forward DNS lookup is using an Internet domain name to find an IP address. Reverse DNS lookup is using an Internet IP address to find a domain name. When you enter the address for a Web site at your browser (the address is formally called the Uniform Resource Locator, or URL), the address is transmitted to a nearby router which does a forward DNS lookup in a routing table to locate the IP address. Forward DNS (which stands for domain name system) lookup is the more common lookup since most users think in terms of domain names rather than IP addresses. However, occasionally you may see a Web page with a URL in which the domain name part is expressed as an IP address (sometimes called a dot address) and want to be able to see its domain name. An Internet facility that lets you do either forward or reverse DNS lookup yourself is called nslookup. It comes with some operating systems or you can download the program and install it in your computer.



First your computer queries the name server (DNS server) it is set up to use. This is the recursive name server shown above.

The name server doesn't know the IP address for www.abc.com, so it will start the following chain of queries before it can report back the IP address to your computer (the numbers below correspond to the numbers in the image).

1. Query the **Internet root servers** to get the name servers for the .com TLD.
2. Query the .com **TLD name servers** to get the authoritative name servers for abc.com.
3. Query the **authoritative name servers for abc.com** to finally get the IP address for the host www.abc.com, then return that IP address to your computer.
4. Done! Now that your computer has the IP address for www.abc.com, it can access that host.

Assignment C13

Title: Capture packets using Wireshark and analyses Facebook traffic.

Objective:

Capture packets using Wireshark

Work with capture files and analyze packets

Problem statement: Capture all TCP and HTTP traffic to/from Facebook, during the time when you log in to your Facebook account and analyze.

Software&/ hardware requirements: Wireshark 3.6.0

Theory: //here, write answers to FAQs given below.

11. What is wireshark? Name any other equivalent tool?
12. Difference between display and capture filter?
13. Difference between monitor and promiscuous mode?
14. Explain TCP header?
15. Explain UDP header?
16. Explain IPV4 header
17. Explain IPV6 header
18. Explain owner/organization, use , layer applicable and size of MAC address, IP address (version 4 and 6) and port address?
19. Convert hex to decimal
20. Analyze the given UDP dump Ex: **06 32 00 0D 00 1C E2**

Conclusion: Studied and analyzed Facebook traffic.

Wireshark

Wireshark is a network protocol analyzer that can be installed on Windows, Linux, and Mac. It provides a comprehensive capture and is more informative than Fiddler.

Wireshark captures the data coming or going through the NICs on its device by using an underlying packet capture library. By default, Wireshark captures on-device data only, but it can capture almost all the data on its LAN if run in promiscuous mode. Currently, Wireshark uses NMAP's Packet Capture library(called npcap).

To use:

1. Install Wireshark.
2. Open your Internet browser.
3. Clear your browser cache.
4. Open Wireshark
5. Click on "**Capture** > **Interfaces**". A pop-up window will display.
6. You'll want to capture traffic that goes through your ethernet driver. Click on the **`Start`** button to capture traffic via this interface.
7. Visit the URL that you wanted to capture the traffic from (Eg. Login to facebook to analyze facebook traffic).
8. Go back to your Wireshark screen and **press Ctrl + E** to stop capturing.
9. After the traffic capture is stopped, please save the captured traffic into a **`*.pcap`** format file and attach it to your support ticket.

Assignment C14

Title: HTTP, HTTPS, FTP using packet tracer

Objective: To study the working of application layer protocols HTTP, HTTPS, and FTP using packet tracer.

Problem statement: Study and analyze the performance of HTTP, HTTPS, and FTP protocols using packet tracer tool.

Software&/ hardware requirements: Cisco Packet Tracer

Theory: //here, write answers to FAQs given below.

1. What is HTTP and HTTPS? State the port number for HTTP at transport layer.
2. Explain HTTP Request Message format with neat labeled diagram.
3. Explain HTTP Response Message format with neat labeled diagram.
4. What are HTTP Request Methods?
5. What are Persistent Connections? And differentiate between persistent and non-persistent HTTP.
6. What are (400 Bad Request) and (200 OK) response codes in HTTP?
7. What is file transfer protocol (FTP)? State the port number for FTP at transport layer.
8. State the FTP commands and their syntax with example: a) to receive a text file at client from ftp server, and b) to transfer the text file from client to ftp server
9. With diagram explain the two types of connections in FTP and their port numbers.
10. Different active mode and passive mode in FTP.

Conclusion: Studied and analyzed study the working of application layer protocols HTTP, HTTPS, and FTP using packet tracer.

Theory:

File transfer protocol server configuration using cisco packet tracer:

Objectives:

- To Configure FTP Services on Server.
- To Upload a File into the FTP Server from Remote PC.
- To Download a File from the FTP Server from Remote PC.

Procedure:

Step-1: (Configuring Router0):

1. Select a 2911 Router from Network Devices and drag and drop to the workspace.
2. Select Router0 and Go to Config.
3. Configure the GigabitEthernet0/0 by assigning IP address as 192.168.1.1 and subnet mask as 255.255.255.0 and turn on the port status
4. Configure the GigabitEthernet0/1 by assigning IP address as 10.0.0.1 and subnet mask as 255.0.0.0 and turn on the port status.

Router0 Command Line Interface:

```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface GigabitEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#shutdown
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
```

Step-2: (Configuring PCs):

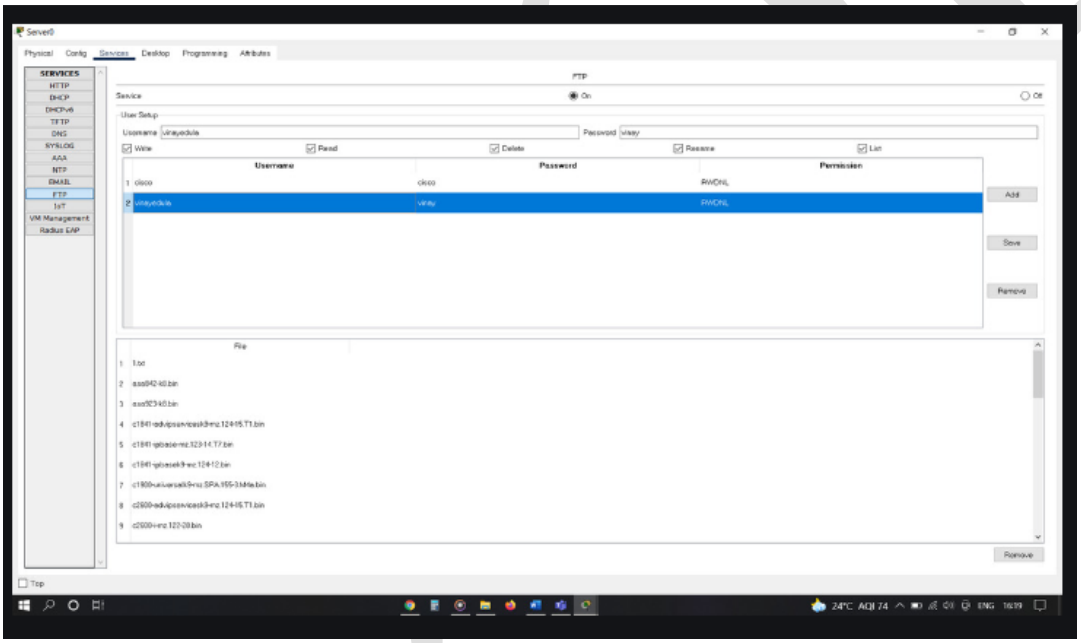
1. Select two PC-PT type PCs from End devices and drag and drop to the workspace.
2. Select PC0 and go to FastEthernet0 in config and assign IP address and subnet mask for the PC0 as 192.168.1.2, 255.255.255.0

3. Select PC1 and go to FastEthernet0 in config and assign IP address and subnet mask for the PC1 as 192.168.1.3, 255.255.255.0
4. For both the PCs (PC0, PC1) go to Global settings in config and Assign default gateway as 192.168.1.1

Step-3: (Configuring Server0):

1. Select a server from End devices and drag and drop to the workspace.
2. Go to the global settings in config and assign default gateway as 10.0.0.1
3. Go to FastEthernet0 and assign IP address and subnet mask as 10.0.0.2, 255.0.0.0
4. Go to services and open FTP Service.
5. Go to user setup and create a username and password.
6. Select all the permissions (Write, Read, Delete, Rename, List) and add the user.

User setup:



Step-4: (Configuring Switch and Making connections):

1. Select a 2950-24 Switch from the network devices and drag and drop to the workspace.
2. Connect FastEthernet0 port of PC0 to the FastEthernet0/1 port of switch0 using Copper Straight-Through cable.
3. Connect FastEthernet0 port of PC1 to the FastEthernet0/2 port of switch0 using Copper Straight-Through cable.
4. Connect FastEthernet0/3 port of switch0 to the GigabitEthernet0/0 of Router0 using Copper Straight-Through cable.

5. Connect GigabitEthernet0/1 port of Router0 to the FastEtherner0 of server0 using Copper Straight-Through cable.

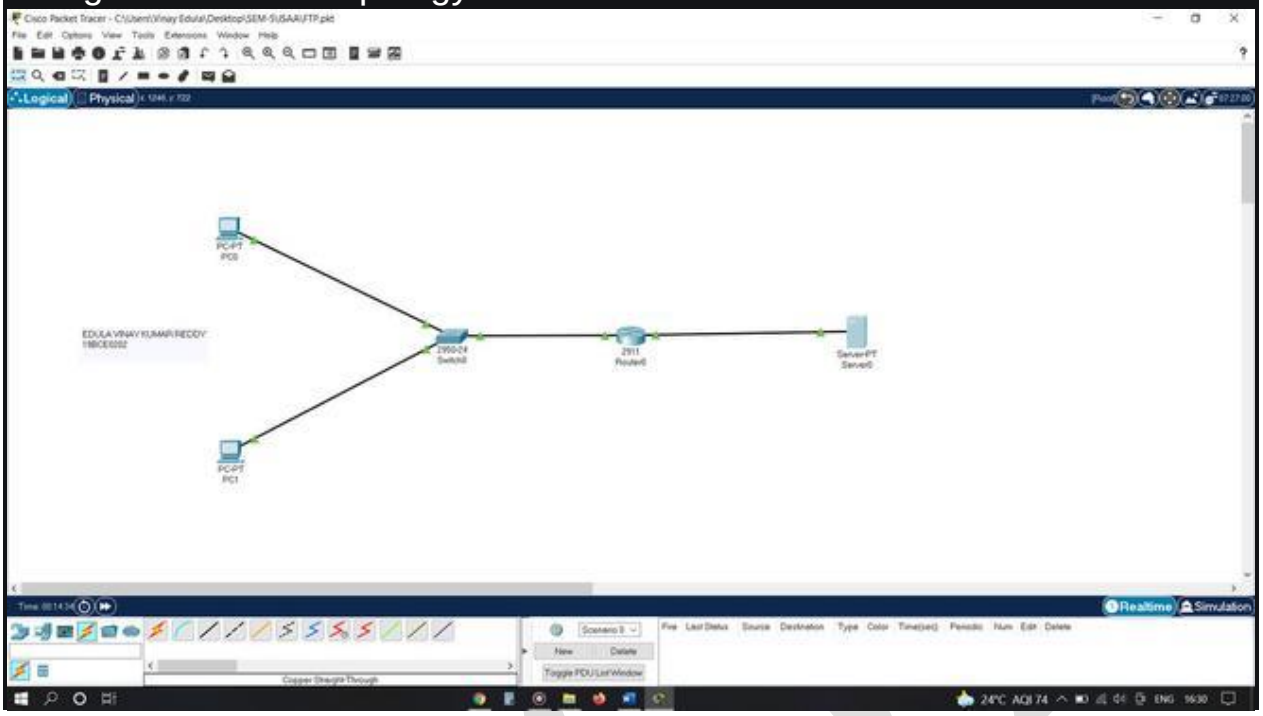
Router Configuration Table:

Device Name	IP address GigabitEthernet0/0	Subnet Mask	IP AddressGigabitEthernet0/1	Subnet Mask
2911 Router0	192.168.1.1	255.255.255.0	10.0.0.1	255.0.0 0

PC Configuration Table:

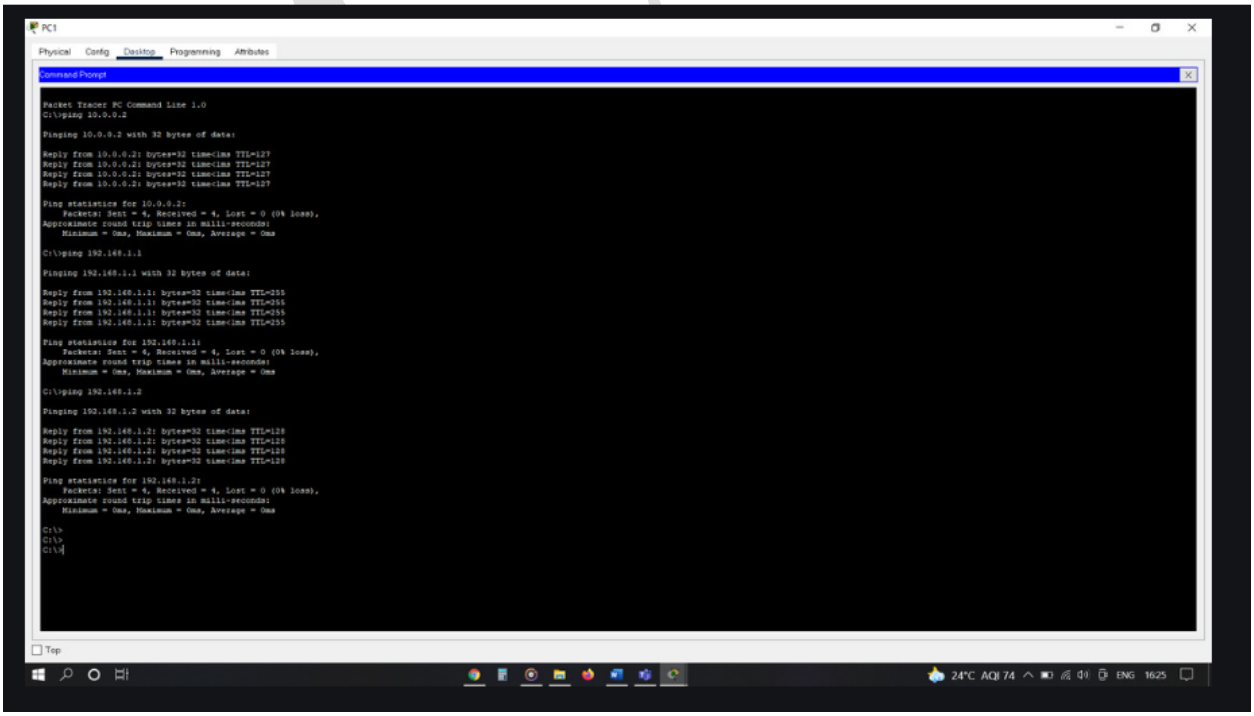
Device Name	Device Type	IP address	Subnet Mask	Default Gateway
PC 0	PC-PT	192.168.1.2	255.255.255.0	192.168.1.1
PC 1	PC-PT	192.168.1.3	255.255.255.0	192.168.1.1

Designed Network topology:

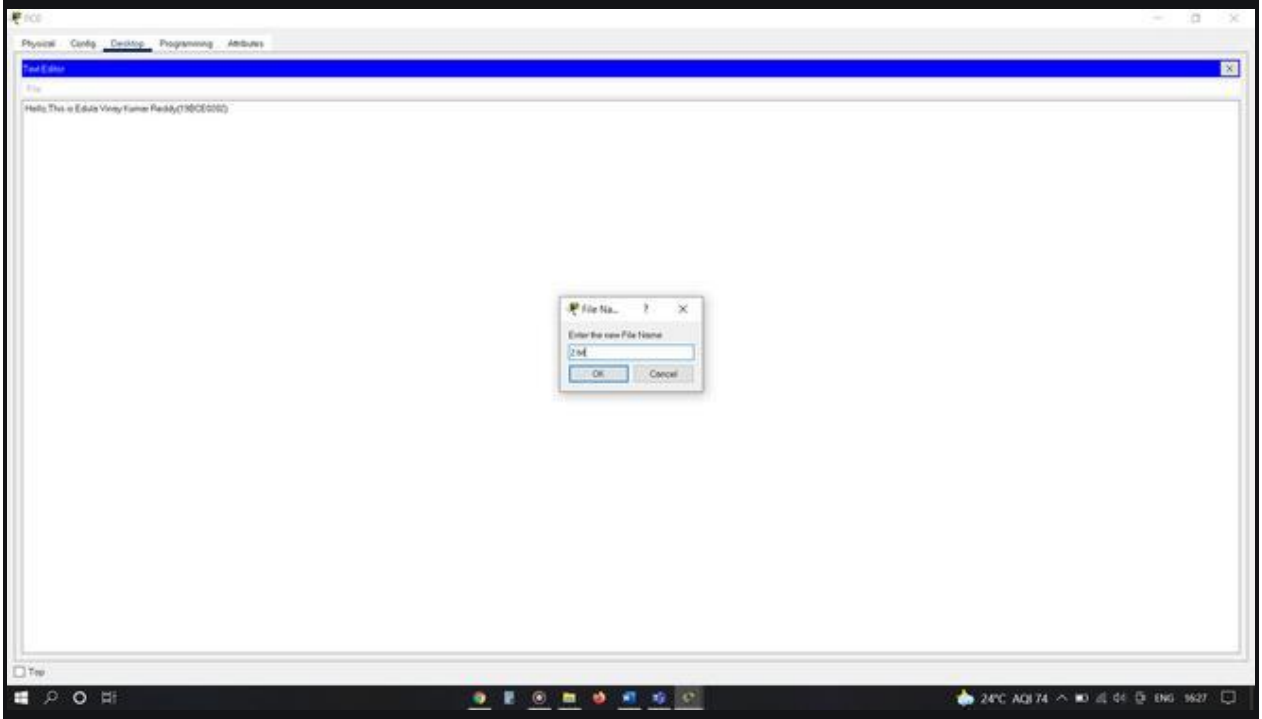


Checking connections from PC0 to the other hosts in the network using ping Command in Command Prompt.

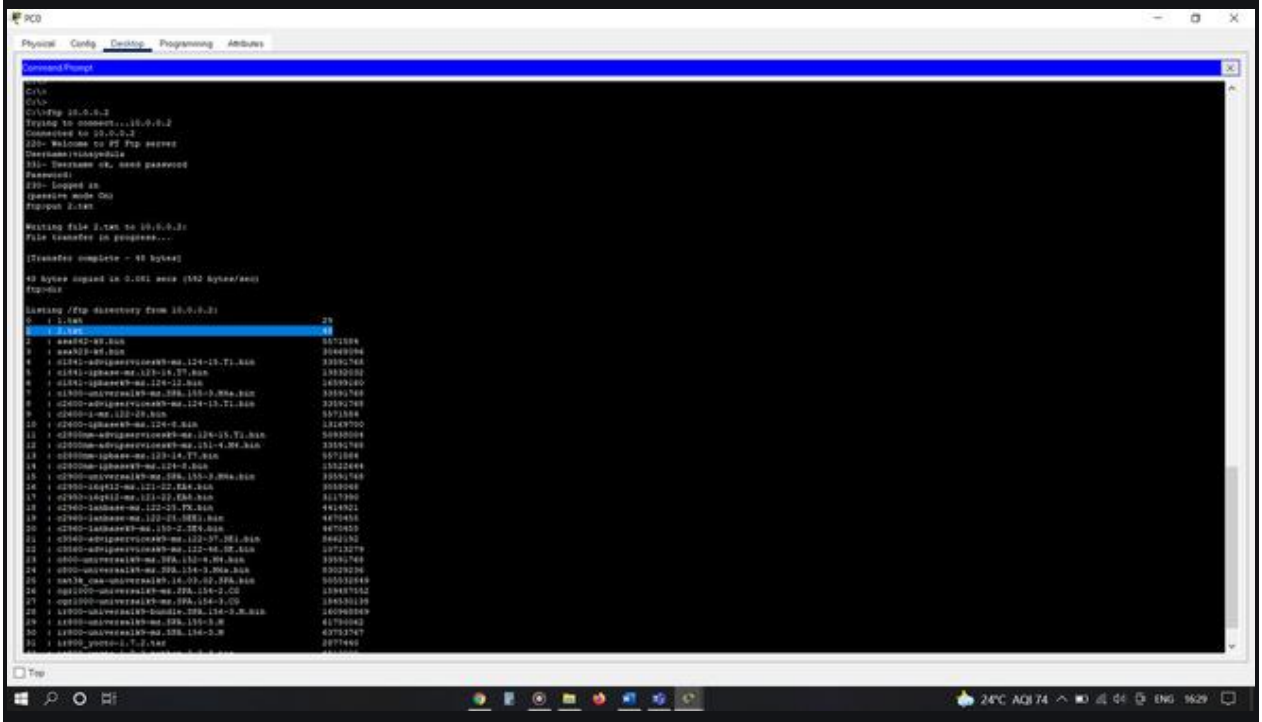
Checking connections from PC0 to the other hosts in the network using ping Command in Command Prompt.



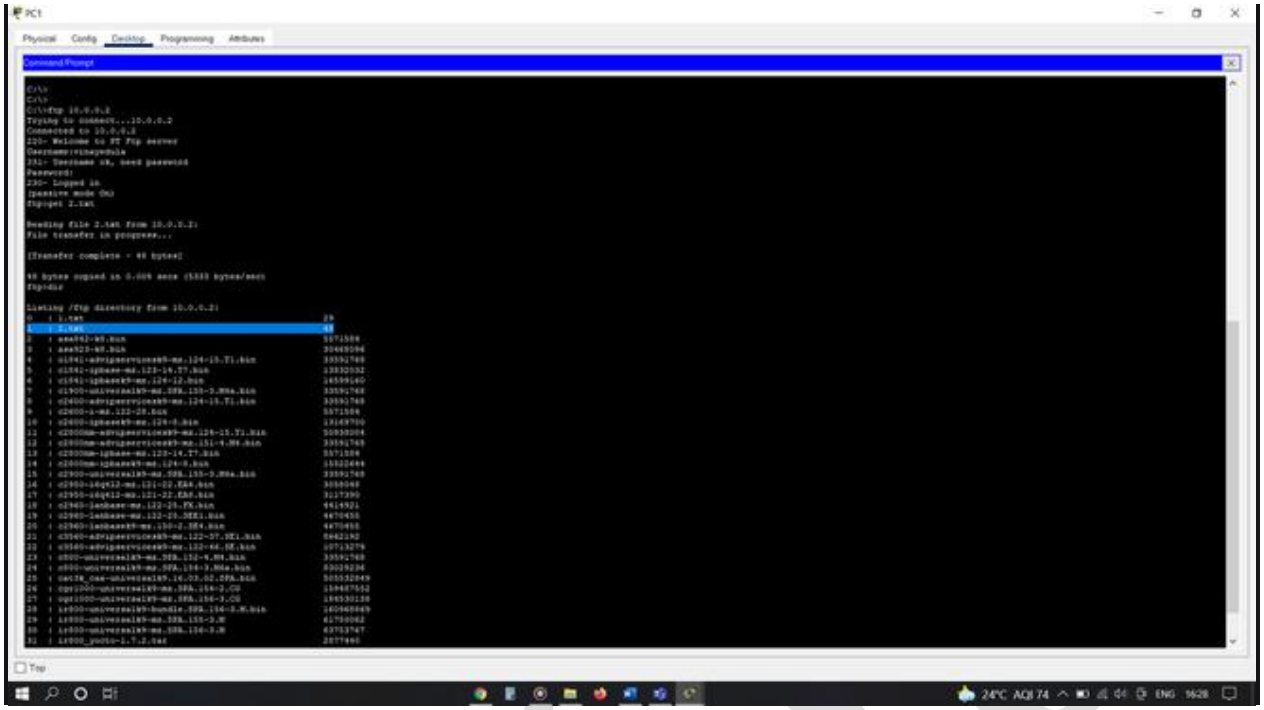
Creating a file named 2.txt for writing(uploading) into FTP Server.



Writing(uploading) the file named 2.txt into FTP Server from PC0 using put 2.txt command and verifying this file transfer using dir command.



Reading(Downloading) the file named 2.txt present in FTP Server from PC1 using get 2.txt command and verifying this file transfer using dir command.



Similarly we have to do for HTTP and HTTPS configurations.

Assignment C15

Title: SSL

Objective: To study SSL protocol.

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

Software&/ hardware requirements: Wireshark 3.6.0

Theory: //here, write answers to FAQs given below.

1. What is difference between HTTP , HTTPs and SSL
2. How does SSL work.
3. Difference between SSL and TLS
4. Justify " compression is done followed by encryption" . Name few compression algorithms.
5. Define Confidentiality, integrity and Authentication with respect to SSL
6. Name algorithms used for confidentiality and integrity?
7. How is authentication achieved during communication in public network.
8. Where is SSL located in TCP/IP protocol stack
9. Draw and explain SSL protocol stack?
10. What is a CA? and name few CAs?

Conclusion: studied SSL protocol by capturing the packets using wireshark tool .

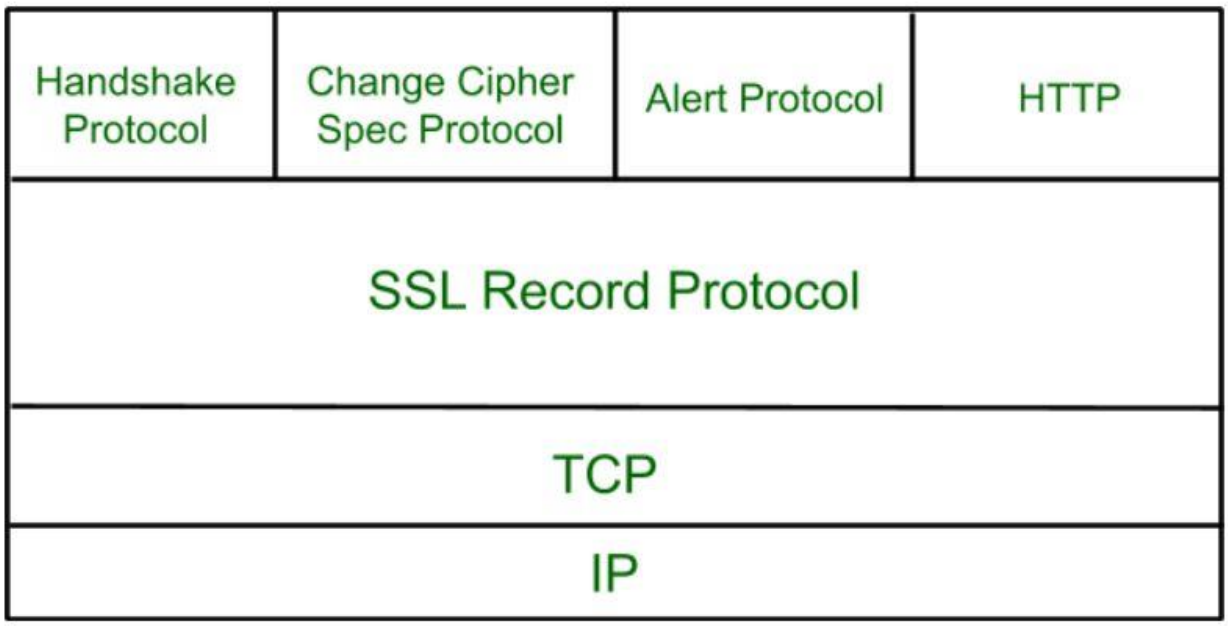
Theory:

Secure Socket Layer (SSL) provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

Secure Socket Layer Protocols:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol
- Alert protocol

SSL Protocol Stack:

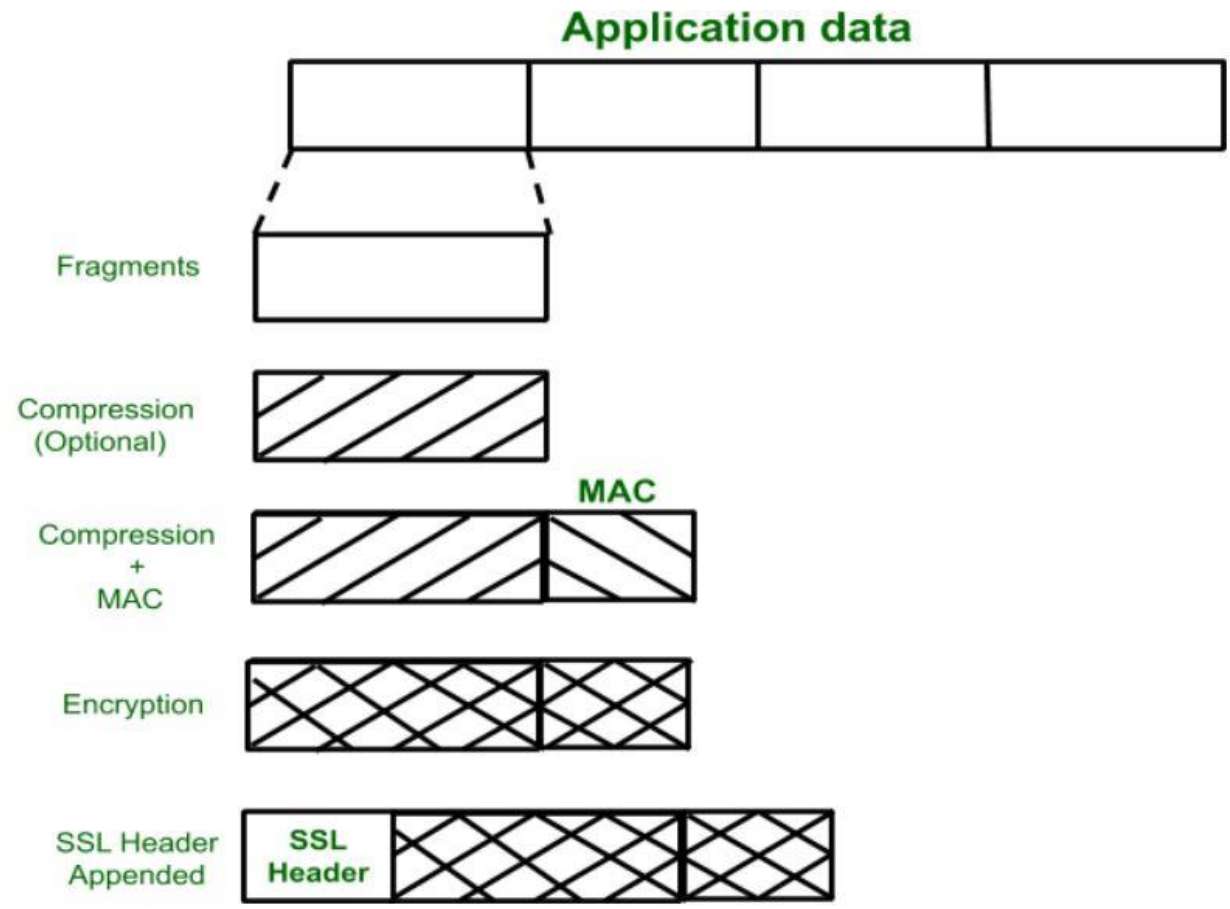


SSL Record Protocol:

SSL Record provides two services to SSL connection.

- Confidentiality
- Message Integrity

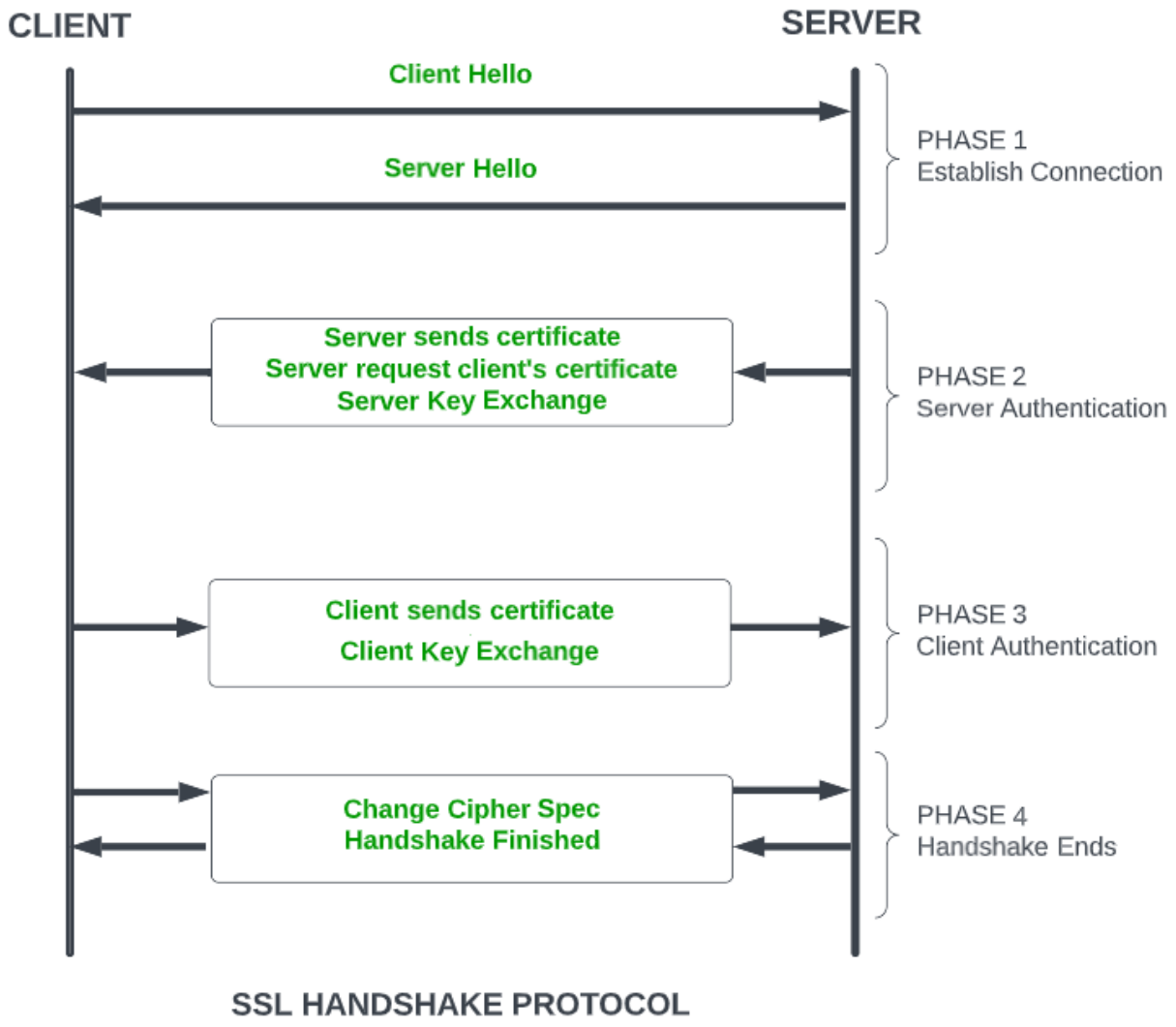
In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.



Handshake Protocol:

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

- **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
- **Phase-2:** Server sends his certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.
- **Phase-3:** In this phase, Client replies to the server by sending his certificate and Client-exchange-key.
- **Phase-4:** In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends.



SSL Handshake Protocol Phases diagrammatic representation

Change-cipher Protocol:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current state.
Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.

Alert Protocol:

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.

The level is further classified into two parts:

Warning (level = 1):
This Alert has no impact on the connection between sender and receiver. Some of them are:

Bad certificate: When the received certificate is corrupt.
No certificate: When an appropriate certificate is not available.
Certificate expired: When a certificate has expired.
Certificate unknown: When some other unspecified issue arose in processing the certificate, rendering it unacceptable.
Close notify: It notifies that the sender will no longer send any messages in the connection.

Fatal Error (level = 2):

This Alert breaks the connection between sender and receiver. The connection will be stopped, cannot be resumed but can be restarted. Some of them are :

Handshake failure: When the sender is unable to negotiate an acceptable set of security parameters given the options available.
Decompression failure: When the decompression function receives improper input.
Illegal parameters: When a field is out of range or inconsistent with other fields.
Bad record MAC: When an incorrect MAC was received.
Unexpected message: When an inappropriate message is received.

The second byte in the Alert protocol describes the error.

Silent Features of Secure Socket Layer:

- The advantage of this approach is that the service can be tailored to the specific needs of the given application.
- Secure Socket Layer was originated by Netscape.
- SSL is designed to make use of TCP to provide reliable end-to-end secure service.
- This is a two-layered protocol.

Assignment C17

Title: To study the IPsec (ESP and AH) protocol by capturing the packets using Wireshark tool.
Objective: Internet Protocol Security

Problem statement: To study the IPsec (ESP and AH) protocol by capturing the packets using Wireshark tool.

Software&/ hardware requirements: Wireshark 3.6.0

Theory: //here, write answers to FAQs given below.

1.What is a security protocol, and what is its purpose?

2. Give examples of services that can be provided by security protocols.
3. Briefly describe three major benefits of using IPSec.

4. What are the three security services that can be provided by IPsec?

5. Briefly explain the type of mechanism used to provide each of these services.

Encapsulating Security Payload (ESP) is an IPSec protocol that can be run in two modes: transport mode and tunnel mode.

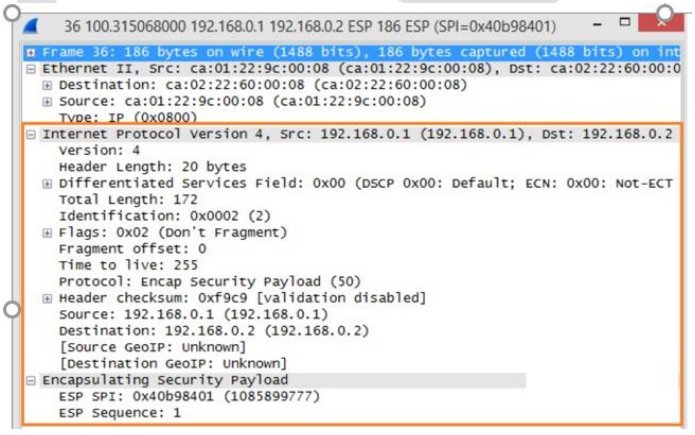
6.Explain the main difference in packet processing between these two modes.

7.Briefly describe the most typical application scenario for ESP in tunnel mode.

8. Briefly describe an application scenario for ESP in transport mode.

9. Briefly explain the additional security services provided by using ESP in tunnel mode as opposed to using ESP in transport mode.

10.Explain the following with respect to IPSEC protocol



Conclusion: Thus understood the working principle of IPSEC.

Theory:

The **IP security (IPSec)** is an Internet Engineering Task Force (IETF) standard suite of protocols between 2 communication points across the IP network that provide data authentication, integrity, and confidentiality. It also defines the encrypted, decrypted and authenticated packets. The protocols needed for secure key exchange and key management are defined in it.

Uses of IP Security –

IPsec can be used to do the following things:

- To encrypt application layer data.
- To provide security for routers sending routing data across the public internet.
- To provide authentication without encryption, like to authenticate that the data originates from a known sender.
- To protect network data by setting up circuits using IPsec tunneling in which all data is being sent between the two endpoints is encrypted, as with a Virtual Private Network(VPN) connection.

Components of IP Security –

It has the following components:

1. **Encapsulating Security Payload (ESP) –**

   It provides data integrity, encryption, authentication and anti replay. It also provides authentication for payload.

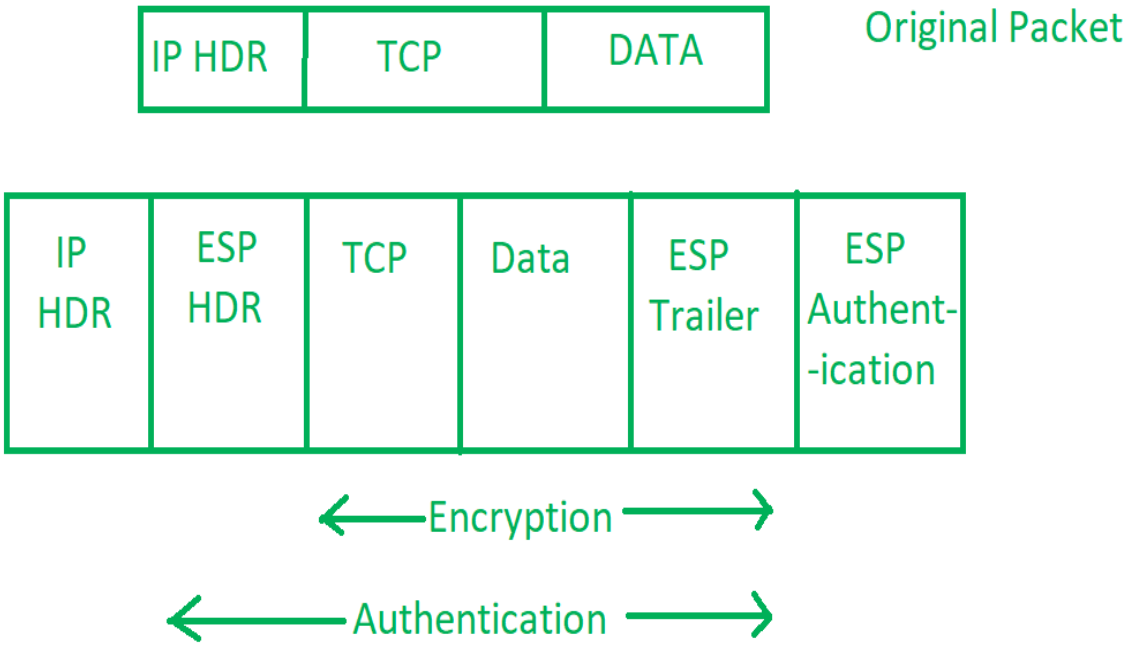
2. **Authentication Header (AH) –**

   It also provides data integrity, authentication and anti replay and it does not provide encryption. The anti replay protection, protects against unauthorized transmission of packets. It does not protect data's confidentiality.

3. Internet Key Exchange (IKE) –

It is a network security protocol designed to dynamically exchange encryption keys and find a way over Security Association (SA) between 2 devices. The Security Association (SA) establishes shared security attributes between 2 network entities to support secure communication. The Key Management Protocol (ISAKMP) and Internet Security Association which provides a framework for authentication and key exchange. ISAKMP tells how the set up of the Security Associations (SAs) and how direct connections between two hosts that are using IPsec.

Internet Key Exchange (IKE) provides message content protection and also an open frame for implementing standard algorithms such as SHA and MD5. The algorithm's IP sec users produces a unique identifier for each packet. This identifier then allows a device to determine whether a packet has been correct or not. Packets which are not authorized are discarded and not given to receiver.



Working of IP Security –

1. The host checks if the packet should be transmitted using IPsec or not. These packet traffic triggers the security policy for themselves. This is done when the system sending the packet apply an appropriate encryption. The incoming packets are also checked by the host that they are encrypted properly or not.
2. Then the **IKE Phase 1** starts in which the 2 hosts (using IPsec) authenticate themselves to each other to start a secure channel. It has 2 modes. The **Main mode** which provides the greater security and the **Aggressive mode** which enables the host to establish an IPsec circuit more quickly.

3. The channel created in the last step is then used to securely negotiate the way the IP circuit will encrypt data across the IP circuit.
4. Now, the **IKE Phase 2** is conducted over the secure channel in which the two hosts negotiate the type of cryptographic algorithms to use on the session and agreeing on secret keying material to be used with those algorithms.
5. Then the data is exchanged across the newly created IPsec encrypted tunnel. These packets are encrypted and decrypted by the hosts using IPsec SAs.
6. When the communication between the hosts is completed or the session times out then the IPsec tunnel is terminated by discarding the keys by both the hosts.