1 Study of Topologies & Transmission media

Title : Demonstrate the different types of topologies and types of transmission media by using packet tracer

Theory

Topology (from the Greek topos: "place", and logis: "study") is the mathematical study of shapes.

Topology is of two types : physical & logical

The topology of a network is the geometric representation of the relationship of all the links and linking devices(usually called nodes)to one another.

Physical topology: refers to the way in which a network is laid out physically.(actual arrangements of nodes & links)

Refer Figure 1

Logical topology :is the way that the signals act on the network media,

OR

the way that the data passes through the network from one device to the next device without considering the physical topology.

A network's logical topology is not necessarily the same as its physical topology.

Refer figure 2

Comparison table of topologies

	Mesh	**Bus**	**Ring**	**Star**
Logical diagram	Figure 1	Figure 1	Figure 1	Figure 1
Principle	Every node is connected to every other node by separate dedicated links	All nodes connected via common link	Every node connected to two other nodes to form a ring	All nodes are connected to a central device (hub or switch)
No. of links	N(N-1) /2	1 common link and N drop lines	N	N
When Link breaks	Only that path down	Whole network down	Whole network down	Only that path down .But when switch is down, whole N/W down
cost	Most costly	Least cost	moderate	moderate
Speed	Most fast since dedicated paths in two nodes	Least speed since collisions	moderate	moderate
Use	1)telephone exchanges 2) a backbone connecting the main computers of all networks	obsolete	IBMs token ring LAN , Now obsolete	most commonly used in LANs

Combination of any of these topologies is called **Hybrid** topology, which combines advantages of these.

Comparison of transmission media (wired / guided) (refer figure 3)

	Co axial	**Twisted pair**	**Fiber optic cable**
Diagram	Figure 3	Figure 3	Figure 3
Signal sent in form of	Electrical	Electrical	light
Structure	Made of copper, outer	two twisted wires Made	Made of glass or plastic ,

	conductor in form of braid	of Aluminium /copper	density of outer cladding is less than inner core
Signal flow	Both directions	Both directions	One FOC sends Only in one direction , so two FOC required
Frequency range(bandwidth)	Moderate(500 MHz)	Least(1MHz)	High(186 to 370 THz)
Speed based on subtype	In Mbps, more than UTP	10, 100,200,600 Mbps	In Gbps
Used in topology	Bus	Star	Star , point to point
Sub types	Thick , thin	UTP , STP	Multimode , single mode
Cost	Moderate	Least	most
Noise immunity	High	Least	Highest
Distance in repeaters	9 km	2 Km	More than 40 Km
Use	In analog telephone&LAN But now replaced by UTP or FOC	LAN	Cable TV ,LAN, MAN, WAN

Comparison of transmission media (wireless / unguided)

	Radio	**Microwave**	**Infrared**
Range of freq:	3KHz to 1GHz	1 to 300GHz	300 GHz to 400 THz
Propagation	Ground & Sky	Line of sight	Line of sight
Distance(based on frequency used)	Long; few hundred Km	Moderate ;50 to 80km	Short ;few mtrs
Obstacles	Pass through Obstacles	they are reflected by metal; they pass through glass, paper, plastic, and similar materials; and they are absorbed by foods.	Can't Pass through Obstacles
Applications in wireless communication	AM , FM radio, TV, sea ,police , military ,aviation communication	Mobile telephones, WiFi(WLAN), Bluetooth	Remotes for gadgets like TV, mouse, keyboard, printer etc

Simulation of topologies

Bus topology : connect PCs to each **hub** by straight cable & hubs by cross cables, Set IP address & subnet mask in each PC

Star topology : connect all PCs to single switch by straight cables , set IP address & subnet mask in each PC

Simulation of transmission media

Fiber optic cable media

Insert 1FFE (fiber fast Ethernet) interface in PCs & two 1FFE (fiber fast Ethernet) interface in switch

Wireless media

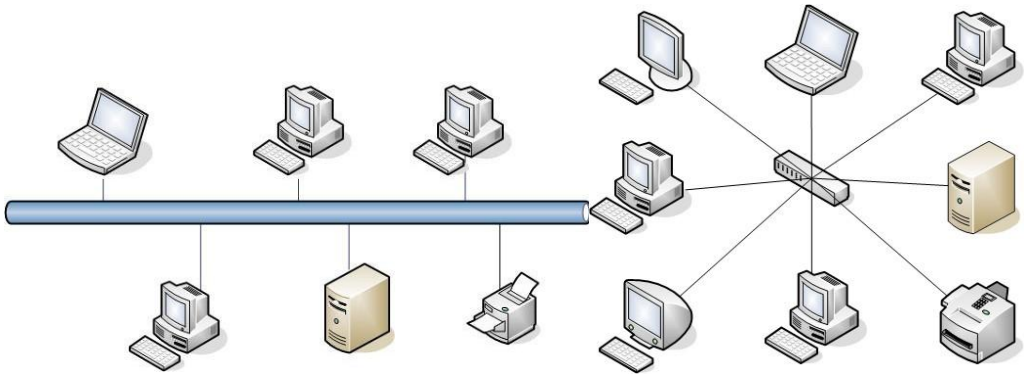
Insert WPC 300N wireless interface in PC & laptop . Use wireless access point (model AP-PT)

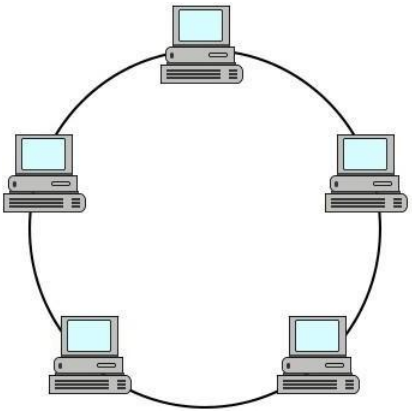
In all above topologies , Verify the connectivity between any two PCs , both in **real mode** (by using ping command) & in simulation mode.

Attach

the print of screen shots of network diagram & verification done using real mode (ping command)

Figure 1 physical topologies





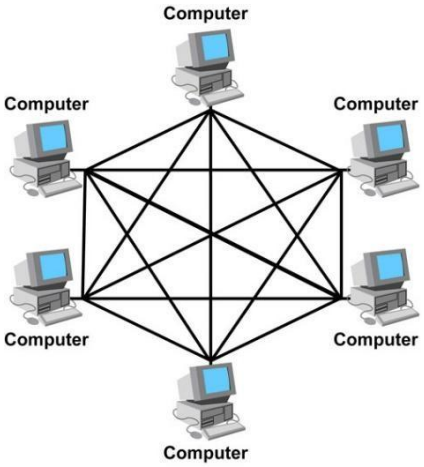
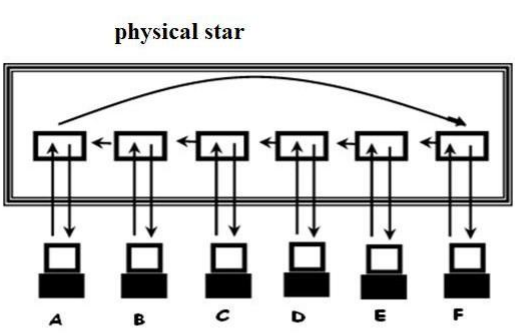
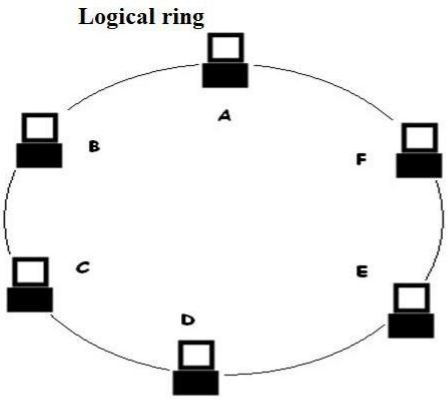
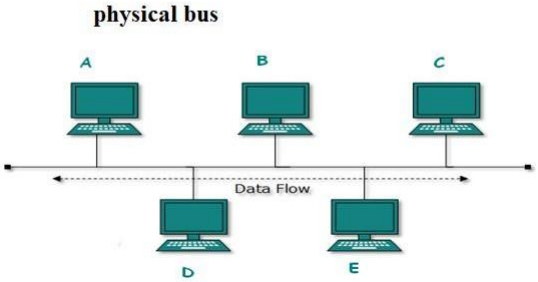


Figure 2 logical topologies







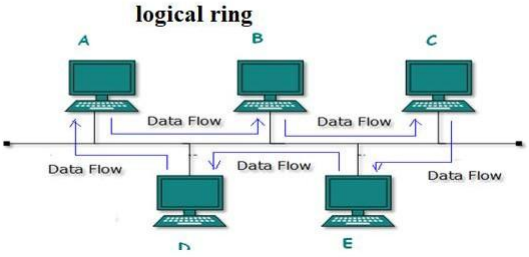
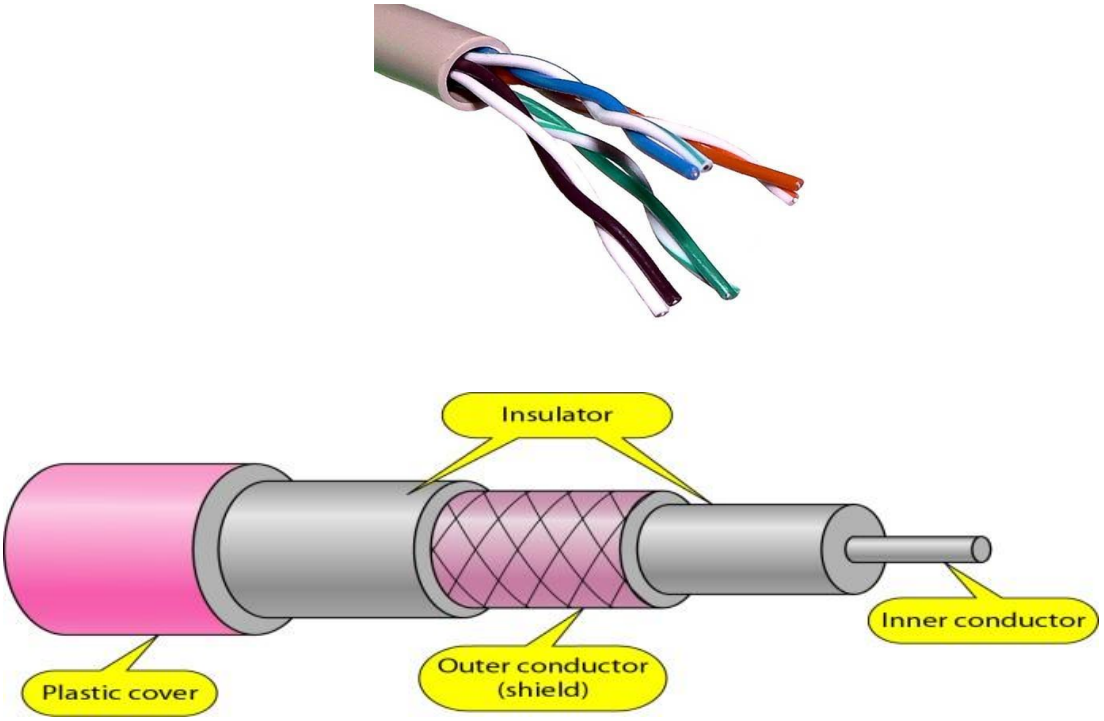
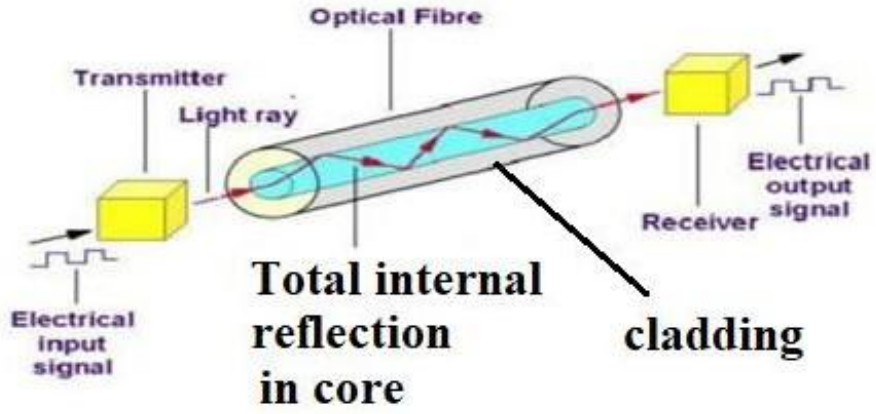


Figure 3 transmission media types





Experiment 2

Title :Setup a WAN

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 LAN 1 (wired LAN) to LAN2 (Wireless LAN).

Theory :

LAN: A local area network (LAN) is privately owned and links the devices in a single office, building, or campus. LAN size is limited to a few kilometers. LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium.

Refer figure 1

LAN can be wired or wireless .

Wired LAN: common wired LAN topologies are bus, ring ,star, mesh & hybrid

WirelessWLAN : also called IEEE 802.11. standard defines two kinds of services: the basic service set (BSS) and the extended service set (ESS). The BSS without anAP is a stand-alone network and cannot send data to other BSSs. It is called an *ad hoc architecture.* In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an *infrastructure* network.

wireless access point : is a device that creates a wireless local area network, or WLAN. An access point connects to a wired router, switch, or hub via an Ethernet cable, and spreads a WiFi signal to a designated area.

Wireless router is a router with integrated (built in) access point.

Refer figure 2

WAN: provides long-distance transmission of information over large geographic areas that can cover a country, a continent, or across the world.

Refer figure 3

Procedure :

Refer figure 4

By using packet tracer

Create a wired LAN with two end devices (PCs) connected to switch(model 2960) In one PC

, set static IPv4 address =192.168.1.2 &Gateway= 192.168.1.1

In second PC , set static IPv4 address =192.168.1.3 &Gateway= 192.168.1.1

Create a wireless LAN with two wireless devices (PC, laptop) connected to wireless access point (model AP-PT). then connect wireless access point to a switch(model 2960). In PC ,remove ethernet card & insert wireless card. set static IPv4 address =192.168.2.2 &Gateway= 192.168.2.1

In laptop , set static IPv4 address =192.168.2.3 &Gateway= 192.168.2.1

Insert two routers(model 1841). Add a serial module in each of the routers. Connect routers by using a serial line (DCE)

Connect switch of the wiredLAN to one router Connect switch of the wirelessLAN to second router

In router (connected to wired LAN) set IP for fast ethernet port=192.168.1.1 IP for serial port =11.0.0.1

In routing , select static routing and add a route 192.168.2.0 via 11.0.0.2

In router (connected to wireless LAN) set IP for fast ethernet port=192.168.2.1 IP for serial port =11.0.0.2

In routing , select static routing and add a route 192.168.1.0 via 11.0.0.1

Test the connectivity from any one PC in **wired** LAN to any device (PC or laptop) in **wireless** LAN using ping command in real mode or sending packet in simulation mode

Attach

the screen shot of the result of ping command in real mode.

Figure 1 wired LAN

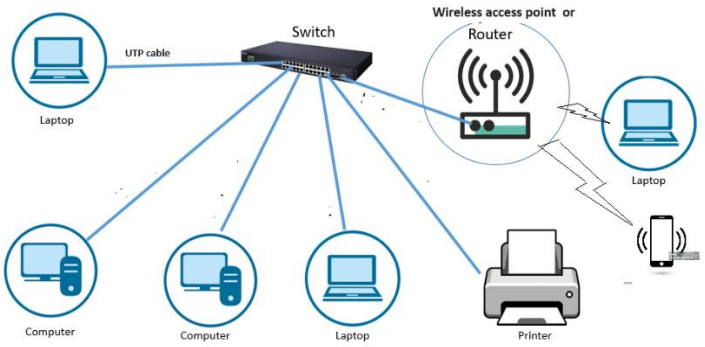


Figure 2 wireless LAN

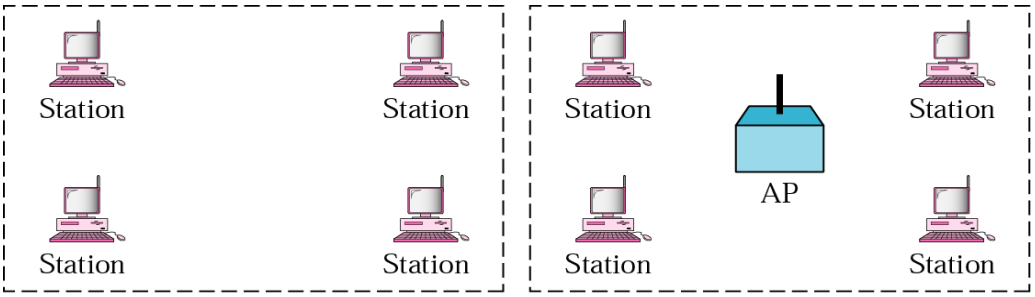


Figure 3 WAN

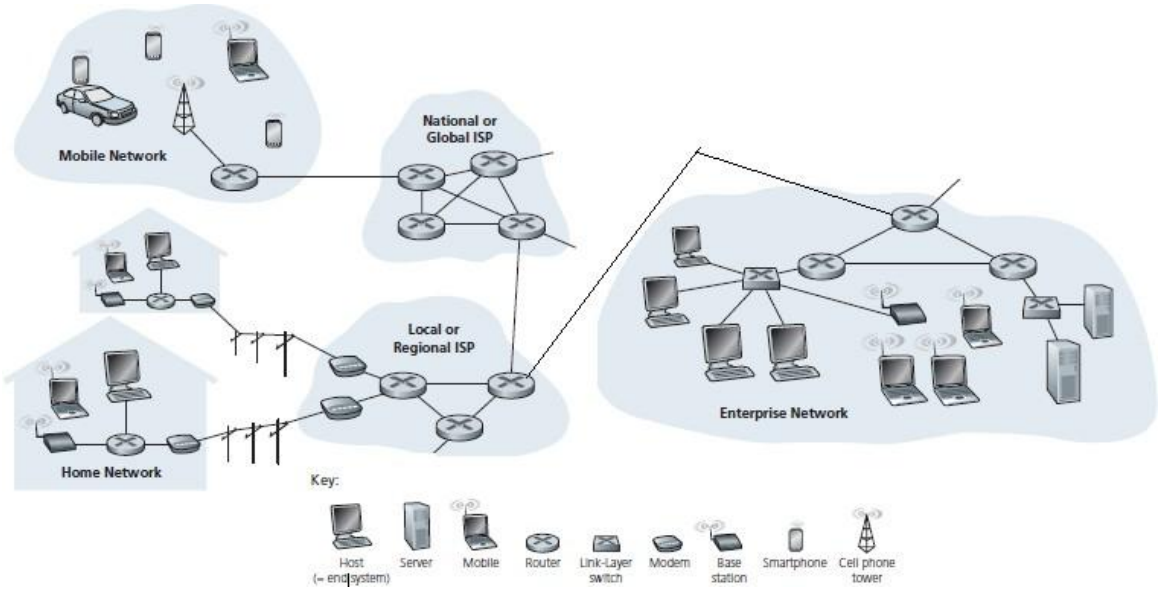
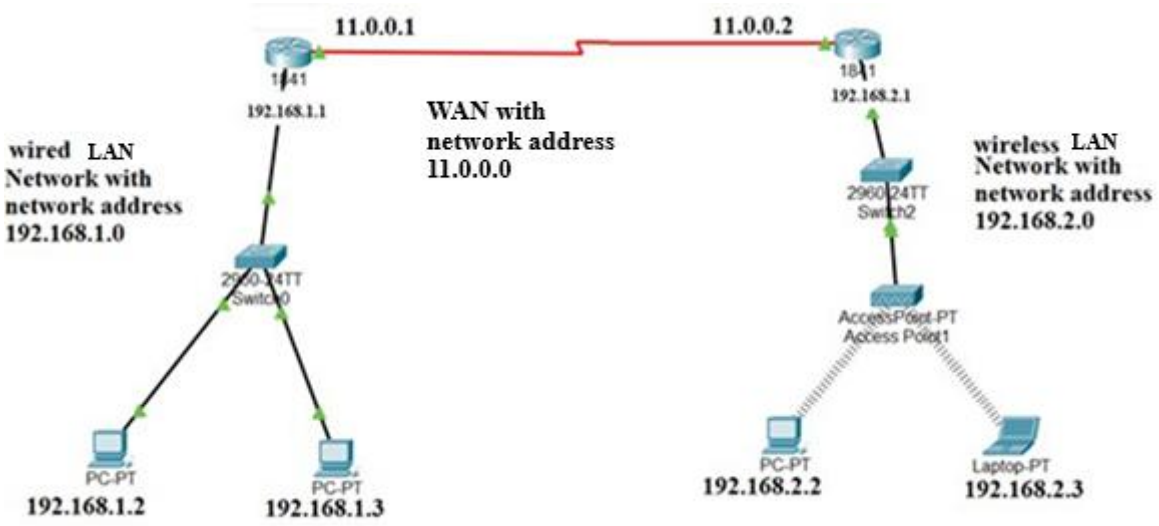


Figure 4 wired LAN to wireless LAN through WAN



Experiment 3
Title: Error detection and correction

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

THEORY:

Error detection & correction by Hamming code
Hamming Code is an error correction code .It is a technique developed by R.W. Hamming. At the data link layer of the sender node , for a given data word ,few Redundant bits(parity bits) are generated .then dataword is appended with the Redundant bits to form a codeword. This code word is then sent to the receiver node . The receiver can detect & correct 1 bit error with the help of Redundant bits in the received codeword.
The number of redundant bits can be calculated using the following formula:
$2^r - 1 - r \geq K.$
where, r = total number of redundant bits, K = total number of data bits in data word.
For 7bit data word, the Redundant bits are calculated Using above formula
$2^r - 1 - r \geq 7.$
For r=4 , the condition is satisfied. Thus, the number of redundant bits= 4.

Then these redundant bits are placed in dataword at the positions of power of 2.
i.e 2^0 =1; 2^1 =2; 2^2 =4 & 2^3 =8
So. If dataword to be transmitted is 1011001, the bits will be placed as follows:

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

Then redundant bits (**parity** bits) are placed as follows:

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

The bits used to **calculate** redundant (parity) bits at the **sender** are

Position	P8	P4	P2	P1
0001=1	0	0	0	1
0010=2	0	0	1	0
0011=3	0	0	1	1
0100=4	0	1	0	0
0101=5	0	1	0	1
0110=6	0	1	1	0
0111=7	0	1	1	1
1000=8	1	0	0	0
1001=9	1	0	0	1
1010=10	1	0	1	0
1011=11	1	0	1	1

Parity bit 1 is calculated by bit positions which have the least significant bit set: bit 3, 5, 7, 9, & 11. i.e. P1 = bit3 $\oplus$ bit5 $\oplus$ bit7 $\oplus$ bit9 $\oplus$ bit11
i.e. P1 is set to 1 if total no. of 1s in these bit positions is odd else set to 0.

Parity bit 2 covers all bit positions which have the second least significant bit set: bit 3, 6, 7, 10 & 11. i.e P2= bit3 ⊕ bit6 ⊕ bit7 ⊕ bit10 ⊕ bit11
i.e. P2 is set to 1 if total no. of 1s in these bit positions is odd else set to 0.

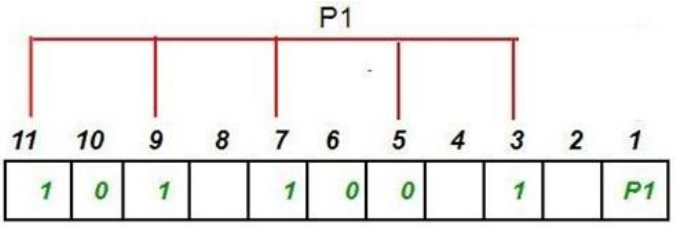
Parity bit 4 covers all bit positions which have the third least significant bit set: bits 5, 6 &7
P4= bit5 ⊕ bit6 ⊕ bit7
i.e. P4 is set to 1 if total no. of 1s in these bit positions is odd else set to 0.

Parity bit 8 covers all bit positions which have the fourth least significant bit set: bits 9,10 & 11. P8 = bit9 ⊕ bit10 ⊕ bit 11
i.e. P8 is set to 1 if total no. of 1s in these bit positions is odd else set to 0.

Example :

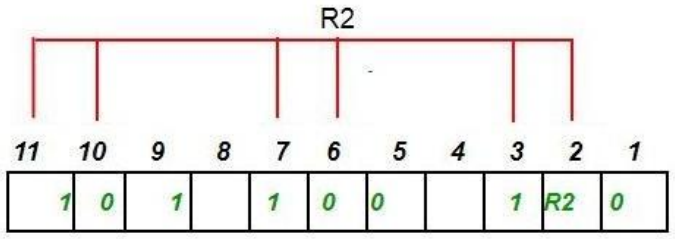
Part 1: Given a data word 1011001. Determining the Parity bits for generating code word

- determine P1



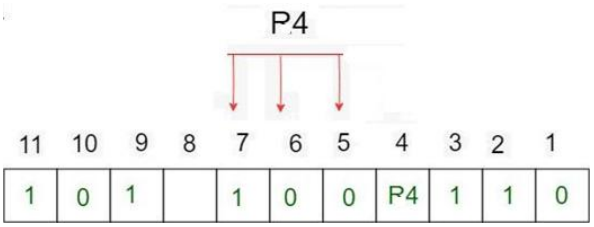
To find the redundant bit R1, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R1 is an even number, the value of R1 (parity bit's value) is = 0
P1 = bit3 ⊕ bit5 ⊕ bit7 ⊕ bit9 ⊕ bit11
= 1 ⊕ 0⊕ 1 ⊕ 1 ⊕ 1
=0

- determine P2



P2= bit3 ⊕ bit6 ⊕ bit7 ⊕ bit10 ⊕ bit11
= 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1
=1

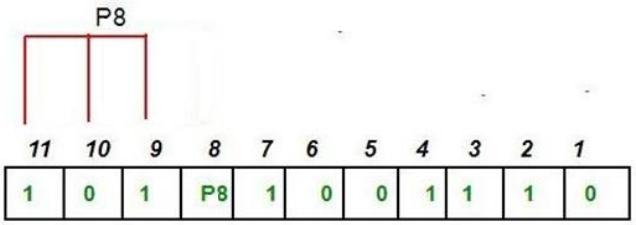
- determine P4



P4= bit5 ⊕ bit6 ⊕ bit7

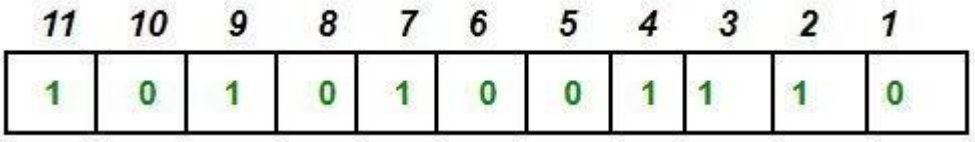
= 0 ⊕ 0 ⊕ 1
= 1

- determine P8 bit

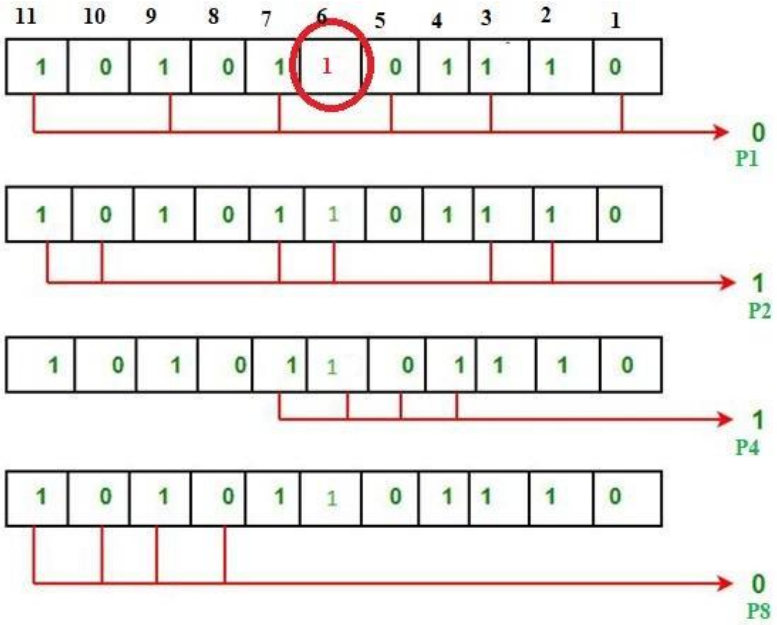


P8 = bit9 ⊕ bit10 ⊕ bit 11
= 1 ⊕ 0 ⊕ 1
= 0

Thus, the data transferred is:



Part 2 Error detection and correction:
Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission, then it gives new parity values in the binary number:



The parity bits are calculated at **receiver** as follows :
P1 = bit1 ⊕ bit3 ⊕ bit5 ⊕ bit7 ⊕ bit9 ⊕ bit11
P2= bit2 ⊕ bit3 ⊕ bit6 ⊕ bit7 ⊕ bit10 ⊕ bit11
P4= bit4 ⊕ bit5 ⊕ bit6 ⊕ bit7
P8 = bit8 ⊕ bit9 ⊕ bit10 ⊕ bit 11
These parity bits are used to find Syndrome .
Syndrome = 8*P8 + 4*P4 + 2*P2 + P1
If syndrome is zero, then there is no error.
If syndrome is non zero, then there is error at position equal to syndrome.
Syndrome = 8 * 0 + 4* 1 + 2*1 + 0
= 6
Thus, bit 6 has error. To correct the error, the 6th bit is changed from 1 to 0.

Algorithm for Hamming code

Sender side

1. Read / accept 7 bit data word from user & store in array D[11] at respective positions D[11, 10, 9, 7, 6, 5, 3]
2. Calculate parity bits P8 , P4, P2 & P1 & store in array D[11] at respective positions D [8,4,2,1]
3. Display resultant code word which is in D[]

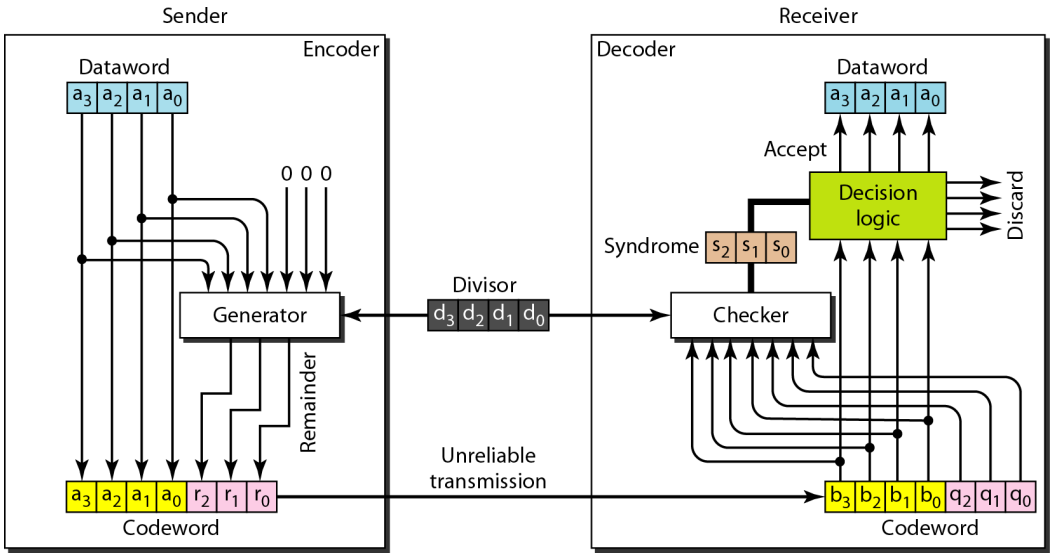
Receiver side

1. Read / accept 11 bit code word from user & store in array D[]
2. Calculate parity bits P8 , P4, P2 & P1
3. Calculate syndrome= 8*P8 + 4*P4 + 2*P2 + P1
4. If syndrome is zero then no error
5. Else there is error in position D[syndrome]
6. Reverse the bit at position D[syndrome]
7. Display the correct data word D[11,10,9,7,6,5,3]

Cyclic Redundancy Check: CRC

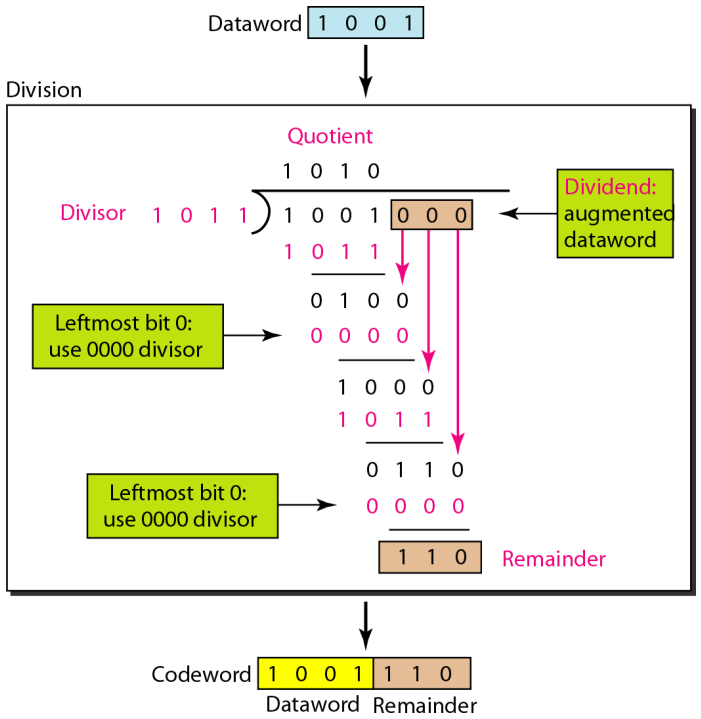
It is a type of linear block code used to only detect errors

- A k-bit dataword is divided by some predetermined number (called the generator). the remainder of the division (rbits) are appended to original k-bit dataword to create code word of n bits (where n = k+r). The sender sends this codeword to receiver.
- The receiver then divides the incoming codeword by the same number used by sender and, if remainder is 0, there is no error and receiver accepts codeword & extracts data word from it & if remainder is non zero then there is error, thenreceiver discards the received codeword.



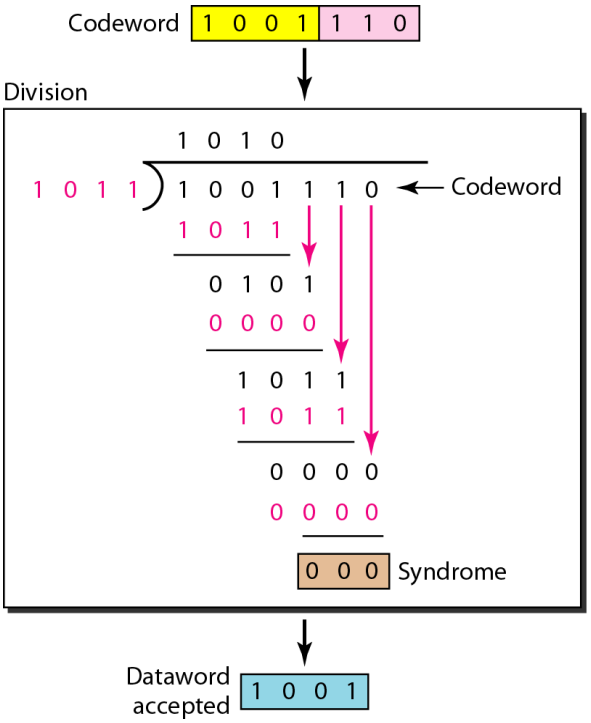
Example:

Sender side :

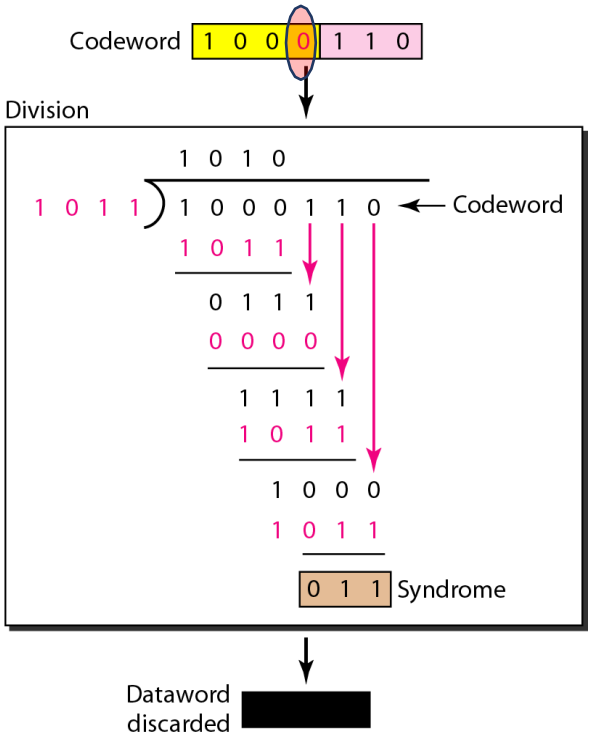


Receiver side:

Case 1: codeword received without error



Case 2: codeword received with error



References :

Online hamming code calculators

https://www.ecs.umass.edu/ece/koren/FaultTolerantSystems/simulator/Hamming/HammingCodes.html

https://www.omnicalculator.com/other/hamming-code

Attach :

print out of Hamming code program & its execution

Experiment 4: Sliding Window Protocol

Problem statement:

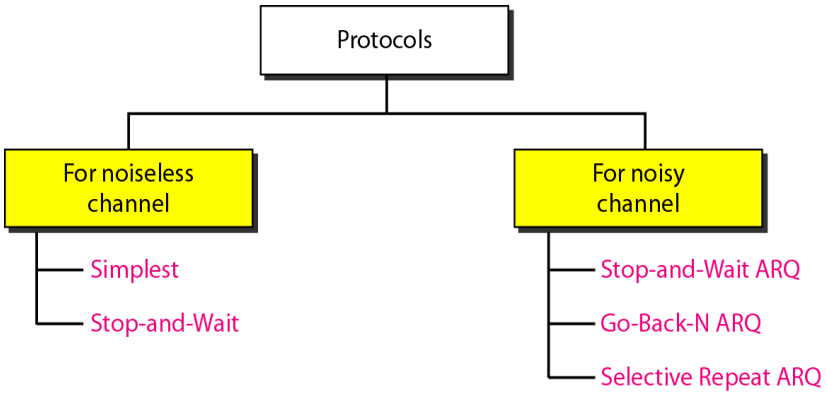
Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.

Theory

Functions of Data link layer (DLL) :

1. framing
2. Flow control
3. error control
4. access control

The taxonomy of flow & error control :



Error control can be done by two methods :

1) FEC

2) ARQ

1) FEC : (forward error correction) : In this the receiver can detect the error in the received data as well as corrects it on its own. Eg of error correction code : Hamming code

2) ARQ: (automatic repeat request) : In this the receiver will only detect that there is some error in the received data but does not correct it on its own.

when a receiver receives a frame with no errors , it sends ACK. But if it detects error in it , it does not correct it on its own , but tell the sender to resend it . (i.e. it does not send ACK frame to sender & when sender does not receive ACK within specified time it automatically sends same frame again)

Eg of error detection codes used by receiver: parity code , CRC code

These codes are used in following Flow & error control protocols :

1. stop & wait ARQ,
2. Go back N ARQ
3. Selective repeat ARQ

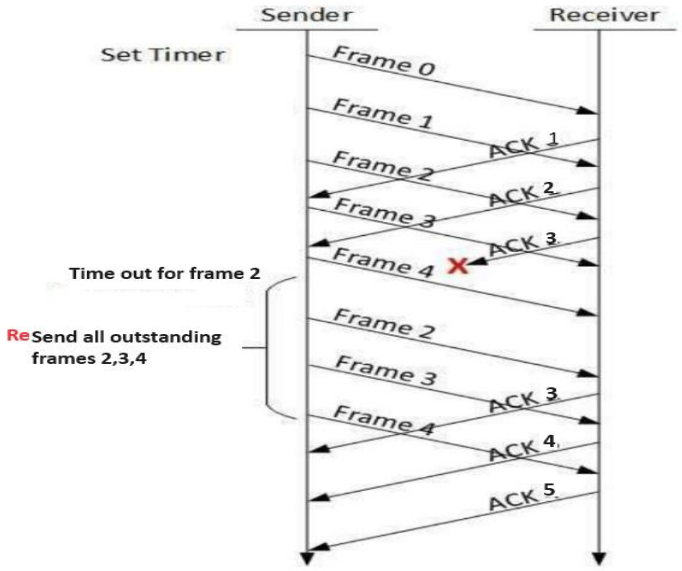
The 2 & 3 are called Sliding window protocols

In this sender can send N frames without checking ACK after each frame.

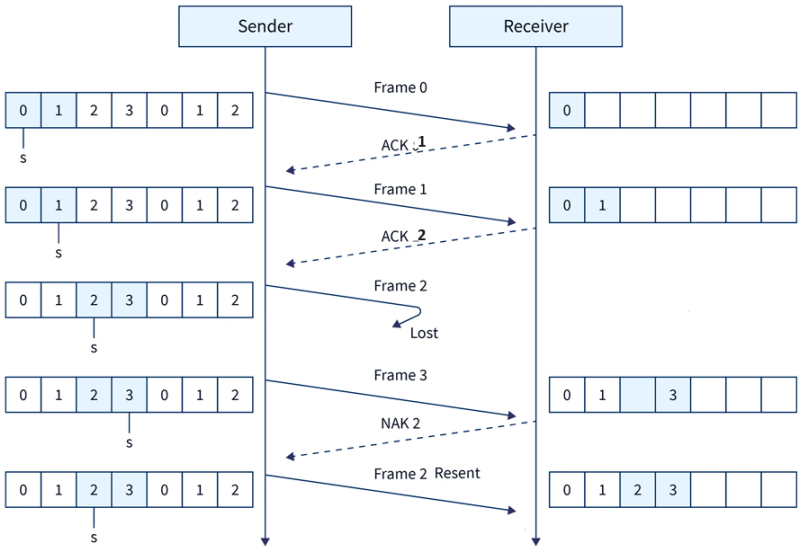
In this both sender & receiver, keep an abstract window in the program.

Go back N ARQ:

If it does not receive ACK for any of the frames sent within time , then it **resends ALL N** outstanding frames whose ACK are not received.



Selective repeat ARQ: Sender sends **only those** select frames for which it didn't receive ACK or receives NAK.



Comparison

	Go back N	**Selective repeat**
Principle	Sends multiple frames (max 2^m) at a time then checks for ACKs from Rx.	Sends multiple frames (max 2^{m-1}) at a time then checks for ACKs from Rx.
Frame numbering	Frames are numbered 0,1 till (2^m)-1	0,1 till (2^{m-1})
Tx window size	(2^m)-1	(2^{m-1})
Rx window size	1	Rx window size is same as TX window i.e. (2^{m-1})
frame receiving policy of Rx	Rx accepts frame only if it is the expected frame. Else it discards the frame.	RX accepts frames even if they are not in sequence. If it does not receive a frame in the sequence , it sends NAK for that frame
Number of timers required at sender	1	Many timers are required; one for each frame sent.
Policy if sent frame is lost/damaged or ACK is lost	If any frame is lost or any ACK is lost or delayed , all the outstanding frames (whose ACK has not been received) are resent by TX.	If any frame is lost or any ACK is lost or delayed or NAK for a particular frame is received only that outstanding frame is resent by TX.
Link utilization	better than stop n wait	better than Go back N
algorithm	Complex than Stop & wait ARQ	most complex

Attach :

1. **Print** of Programs for Go back N sliding window protocol
2. **screen shot** of execution of program
3 **Print** of Programs selective repeat sliding window protocol
4. **screen** shot of execution of program

Experiment 5: Subnetting

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

Theory:

IP address (Internet protocol Address)

Function of network layer is delivery of packets from source host in one network to destination in same or another network. for this, the IP Protocol is used at network layer.

IP protocol uses an unique number to identify each device connected to the Internet & is called the Internet protocol address or IP address. There are two versions of IP addresses 1) IPv4 2) IPv6.

An IPv4 address is a 32-bit address given to a host or a router which is to be connected to the Internet. IP addresses are unique i.e. each address defines one, and only one, device on the Internet. Two devices on the Internet can never have the same address. The total address space of IPv4 is 2^{32} or 4,294,967,296

IP addresses are displayed in dotted decimal notation, and appear as four decimal numbers separated by dots. Each number of an IP address is made from eight bits known as byte/ octet. Each octet can have a value from 0 to 255.

IPv4 has two schemes

1. Class full addressing
2. Classless addressing

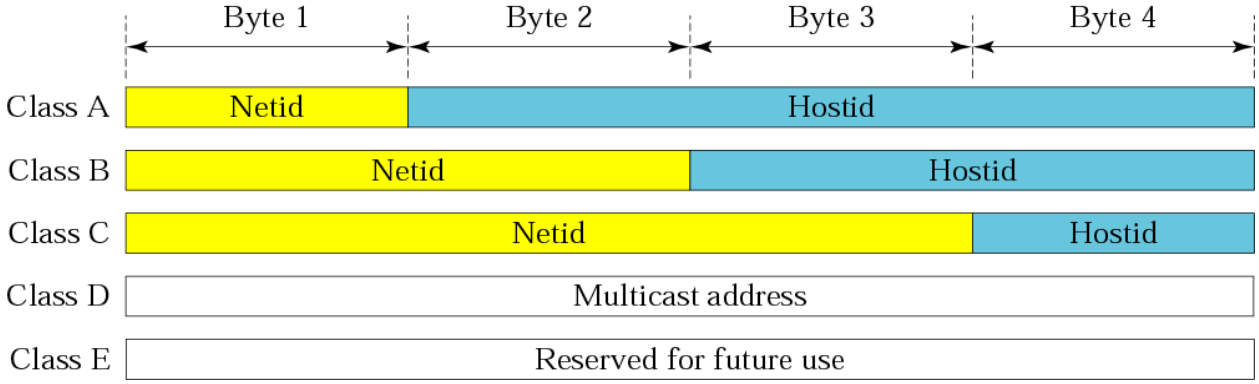
1. Class full addressing : In this IP addresses are divided in to 5 classes . Each class can be identified by the first byte value

In Class A addresses the first byte range from 1-126
In Class B addresses the first byte range from 128-191
In Class C addresses the first byte range from 192-223
In Class D addresses the first byte range from 224-239
In Class E addresses the first byte range from 240-254

IP addresses has two components:

Network id :- is the network address of network in which that host resides. i.e. all the hosts in same network will have same network address.

Host id :- denotes the specific host



Some IP addresses are reserved for some specific purpose

Special Address	*Netid*	*Hostid*	*Source or Destination*
Network address	Specific	All 0s	None
Direct broadcast address	Specific	All 1s	Destination
Limited broadcast address	All 1s	All 1s	Destination
This host on this network	All 0s	All 0s	Source
Specific host on this network	All 0s	Specific	Destination
Loopback address	127	Any	Destination

Addresses for private networks

A number of blocks are set aside for private use. They are not allowed outside that network. Since many organizations can use these same private addresses in their network.

Class	*Netids*	*Blocks*
A	10.0.0	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

Subnetting

Is **logically** dividing a large network into small networks. Its purpose is :

1. Subnetting breaks large network in smaller networks and smaller networks are easier to manage.
2. Subnetting reduces network traffic by limiting collision and broadcast traffic only in particular subnet, which improves overall performance.
3. Subnetting allows you to apply network security polices at the interconnection between Subnets

Subnet mask

To divide a network in subnets , a subnet mask is required. It is **AND**ed with IP address to identify network address and host address from a given IP address.

Mask required for subnetting & supernetting

Default masks for A, B & C class (Class D & E don't have masks)

Class	*Mask in binary*	*Mask in dotted-decimal*
A	**11111111** 00000000 00000000 00000000	**255.**0.0.0
B	**11111111 11111111** 00000000 00000000	**255.255.**0.0
C	**11111111 11111111 11111111** 00000000	**255.255.255.**0

Default notation for Class A networks W.X.Y.Z /8
Default notation for Class B networks W.X.Y.Z /16
Default notation for Class C networks W.X.Y.Z /24

Eg. For a given IP address 192.168.1.10 (it is a class C address)

	In decimal notation	In binary notation
IP address	192.168.1.10	11000000.10101000.00000001.00001010
Subnet mask	255.255.255.0	11111111.11111111.11111111.00000000
Network id	192.168.1	11000000.10101000.00000001
Host id	10	00001010

Block Size

Block size is the size of subnet (no. of IP addresses in that subnet) including network address, hosts addresses and broadcast address.

For class C network:

network id				No. of subnets you want	No. of addresses in each subnet	CIDR Notation
8bits	8bits	8bits	No. of host id bits used for subnet mask	5th column	6th column	7th column
X	X	X	0000 0000	No subnets , only one single network	256	/24
X	X	X	**0**000 0000	two subnets	128	/25
X	X	X	**00**00 0000	4 subnets	64	/26
X	X	X	**000**0 0000	8 subnets	32	/27
X	X	X	**0000** 0000	16 subnets	16	/28
X	X	X	**0000 0**000	32 subnets	8	/29
X	X	X	**0000 00**00	64 subnets	4	/30

5th column : how many subnets you want to make in a given network.

6th column: how many addresses can be there in each subnet. (while number of hosts = addresses-2 , since first address of each block is network address & last address of each block is broadcast address)

7th column: for these number of subnets , what subnet mask is required.

compare your output with some online subnet calculators

https://www.calculator.net/ip-subnet-calculator.html

https://www.site24x7.com/tools/ipv4-subnetcalculator.html

Check IP address & subnet mask in your PC/laptop

Attach:

Program print

Screen shot of execution

Experiment 6 DVR Routing protocol implementation using program

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

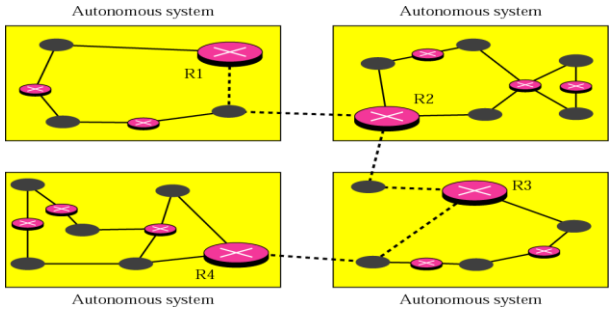
Theory

The function of Network layer is delivery of packets from source host in one network to destination host which is in the same or different network. The protocol used in network layer for this, is Internet protocol (IP). The routers are used to connect different networks. The IP protocol in each router use routing algorithm to decide to which output port (interface) a packet should be forwarded so that it reaches the destination with least cost(distance) of a chosen metric . The metric can be any one of : number of hops, delay , throughput etc.

Today, an internet is so large that one routing protocol cannot handle the task of updating the routing tables of all routers.

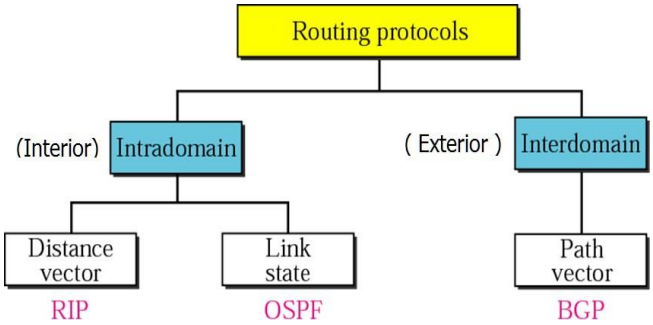
So the internet is divided into autonomous systems.

An **autonomous system** is a group of networks and routers under the authority of a single administration.



Classification of Routing protocols

1. intra domain routing (within same autonomous system);
2. inter domain routing (among different autonomous system)



RIP is an implementation of the distance vector algorithm.

OSPF is an implementation of the link state algorithm.

BGP is an implementation of the path vector algorithm

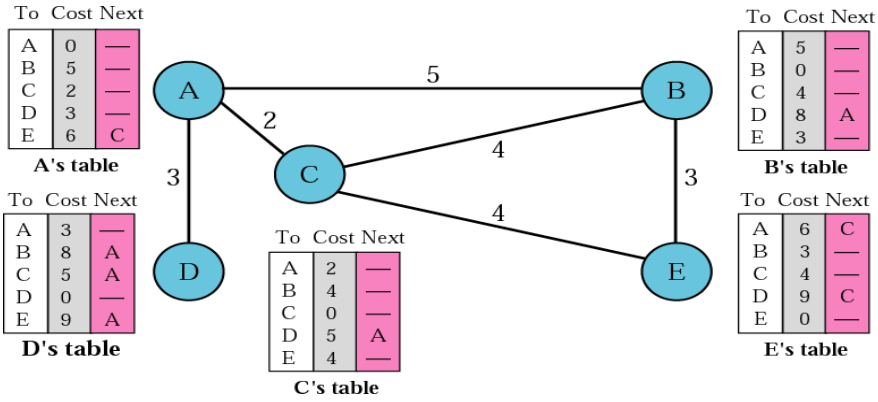
DVR	LSR
It is an intradomain routing algorithm	It is also an intradomain routing algorithm
Every node has only partial knowledge about whole topology	Every node has complete knowledge about whole topology
Each nodes sends info about its neighbours with respective distances to only its neighbours	Each nodes sends info about its neighbours with respective distances to its neighbours, &then neighbours send this to their neighbours. i.e. flooding

So Traffic for sending this info is less	So Traffic for sending this info is more
It generates routing tables slowly	It generates routing tables faster
Each router Uses Bellman -ford algorithm to find shortest paths to all other routers	Each router Uses Dijkshtra algorithm to find shortest paths to all other routers
It has problems of two & three node instability (or count to infinity problem)	It does not have this problem
Implemented in RIP protocol	Implemented in OSPF protocol

Distance Vector Routing (DVR) Protocol

Routing tables in all routers is updated by exchanging information with the **neighboring routers**. The distance vector routing algorithm is sometimes called by other names such as the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm. It was the routing algorithm used in the internet " ARPANET"

Each router keeps a table (called a vector) mentioning distance(cost) to all other routers & output port (interface) & next node to reach to all other routers. Then least distances (cost of the chosen metric), are computed using information from the neighbors' distance vectors.



Distance Vector Table Initialization -

- Distance to itself = 0
- Distance to neighboring routers = distance (cost of metric) as seen from graph
- Distance to ALL other routers(which are not directly connected to this router) = infinity number , but for practical purpose it is set to 99

Distance Vector procedure

1. A router transmits its distance vector to each of its neighbors using a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
   - It receives a distance vector from a neighbor containing different information than before.
   - It discovers that a link to a neighbor has gone down.
   - The Distance vector (DV) calculation is based on minimizing the cost to each destination
     Dx(i) = least distance/cost from x to i
     C(x,v) = cost from node x to each neighbor v
- From time-to-time, each node sends its own distance vector to neighbors.
- When a node x receives new DV from any neighbor v, it saves v's distance vector and it updates its own DV using **Bellman -Ford equation**:

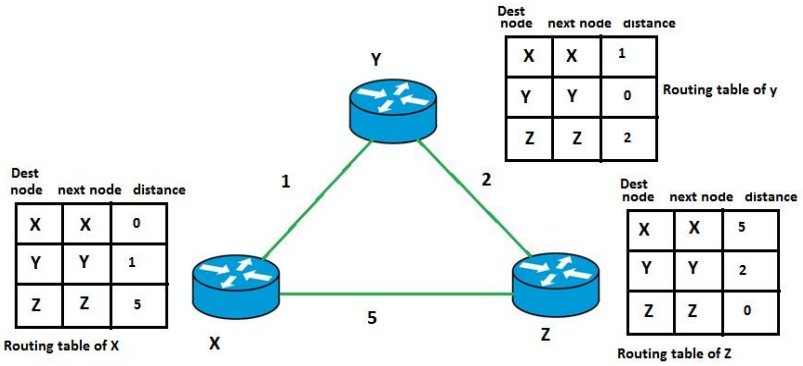
- **Dx(i) = min { C(x,v) + Dv(i) , Dx(i) } for each node i ∈ N (N is total number of nodes)**

  This means distance between node x & node i is the minimum among two alternatives:

  1) (distance from x to intermediate node v)+ (distance from node v to node i)

  2) distance between node x & node i

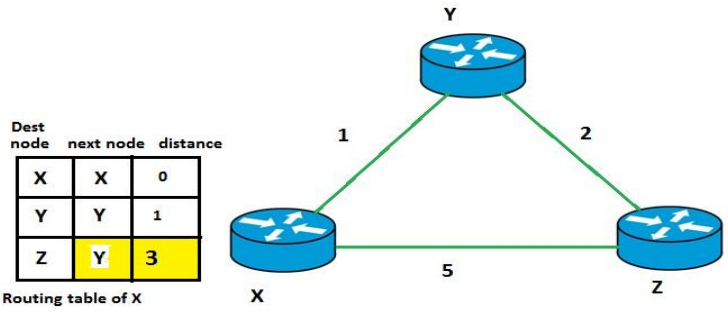
Example – Consider 3-routers X, Y and Z with distance as shown in graph.
1. Each router has its routing table. Every routing table will contain **initial** distance to the destination nodes.



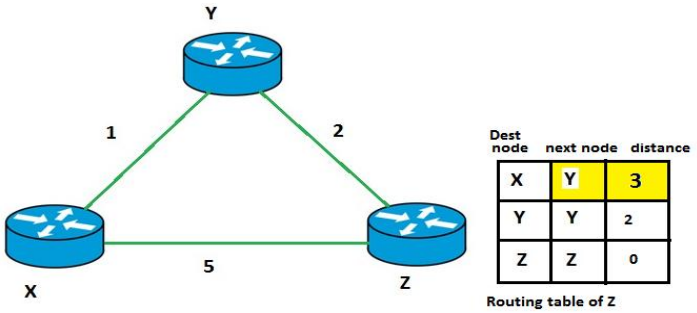
Or in short , the **initial** distance matrix can be written as

	X	Y	Z
X	0	1	5
Y	1	0	2
Z	5	2	0

2. Calculating least distances between all routers:
Consider router X.
X will share it routing table to neighbors and neighbors will share it routing table to it to X and distance from node X to destination I will be calculated using bellmen- ford equation.
Dx(i) = min { C(x,v) + Dv(i) , Dx(i)} for each node i ∈ N & next node v ∈ N



As we can see that distance will be less going from X to Z when Y is next node (hop) so it will be update in routing table of X.
Similarly for Z also , we can see that distance will be less going from Z to X when Y is next node (hop) so it will be update in routing table of Z.

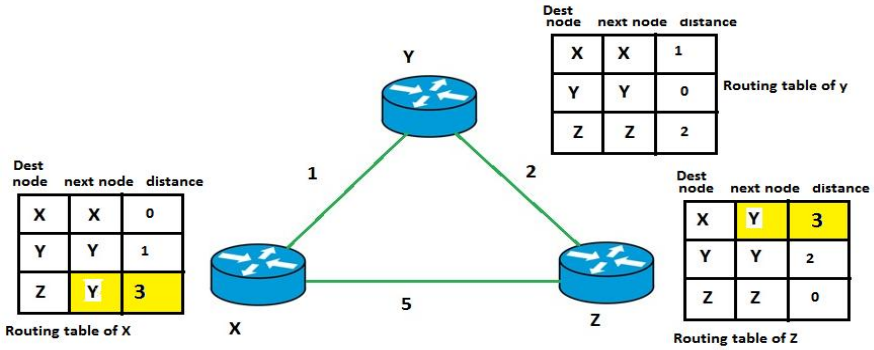


Dest node	next node	distance
X	Y	3
Y	Y	2
Z	Z	0

Routing table of Z

For router Y , there is are no other paths to reach X & Z with distances less than existing distances. So there is no change in routing table of Y.

So final tables are



Dest node	next node	distance
X	X	1
Y	Y	0
Z	Z	2

Routing table of y

Dest node	next node	distance
X	X	0
Y	Y	1
Z	Y	3

Routing table of X

Dest node	next node	distance
X	Y	3
Y	Y	2
Z	Z	0

Routing table of Z

And the final distance matrix is

	X	Y	Z
X	0	1	3
Y	1	0	2
Z	3	2	0

Algorithm:

1. By convention, the distance of the node to itself is assigned to zero and when a node is unreachable the distance is accepted as 99.
2. Accept the input distance matrix from the user (dm[][]) that represents the distance between each node in the network.
3. Store the distance between nodes in a suitable variable.
4. Calculate the minimum distance between two nodes by iterating.
   Dx(i) = min { C(x,v) + Dv(i) , Dx(i) } for each destination node i ∈ N & next node v ∈ N
5. If the existing distance between two nodes is more than the calculated alternate available path, replace the existing distance with the calculated distance.
6. store / Print the shortest path calculated.
7. repeat for all N nodes
8. Stop.

Attach:

Print of program & execution

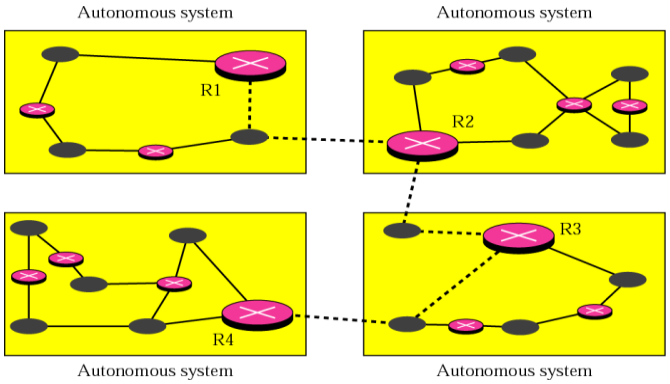
Experiment 7 Routing protocol implementation using Packet tracer

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

Theory & procedure:

The internet is so large that one routing protocol cannot handle the task of updating the routing tables of all routers. So the internet is divided into autonomous systems.

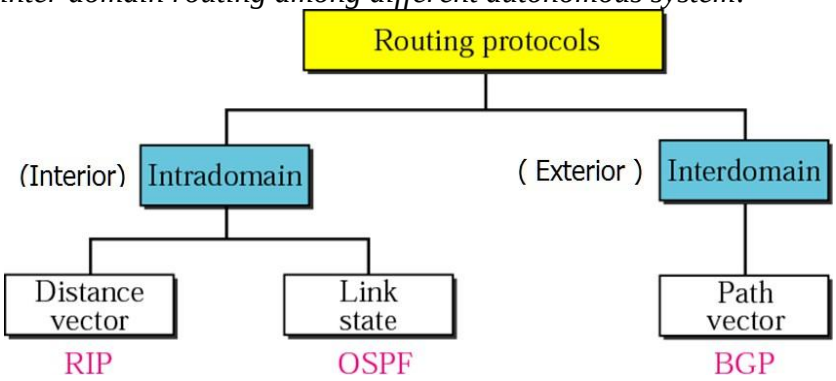
An autonomous system (AS) is a group of networks and routers under the authority of a single administration.



Classification of Routing protocols

intra domain routing within same autonomous system;

inter domain routing among different autonomous system.



(RIP) is an implementation of the distance vector protocol.

(OSPF) is an implementation of the link state protocol.

(BGP)is an implementation of the path vector protocol.

The Routing Information Protocol (RIP)

is an intradomain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing.

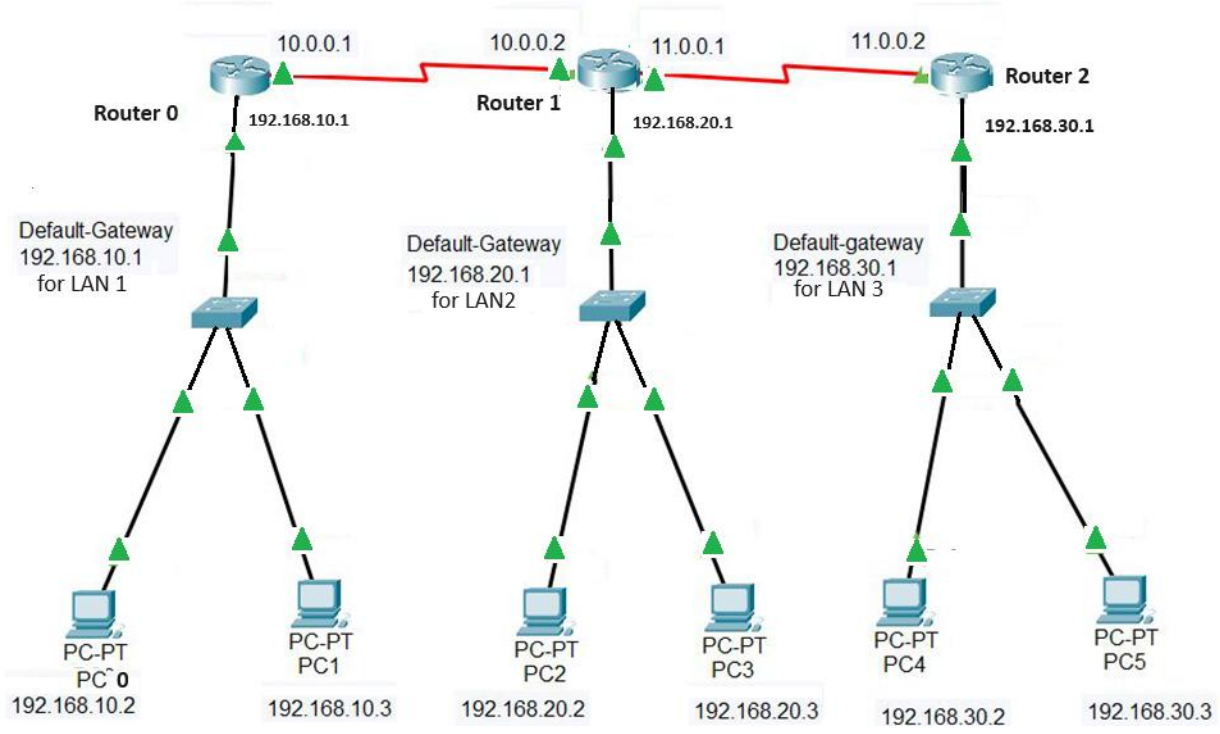
The metric used is the distance defined as the number of links (networks) to reach the destination. So, the metric in RIP is called **a hop count**.

Attach :

Print of ping command ouput for cheking connection between any two PCs in all three networks

Procedure:

Step 1 : create a network topology in packet tracer as shown below



Step 2: Configure the PCs (hosts) with IPv4 addresses,Subnet Masks & gateway according to the table given below.

Device	IPv4 Address	Subnet mask	Default Gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.1
PC1	192.168.10.3	255.255.255.0	192.168.10.1
PC2	192.168.20.2	255.255.255.0	192.168.20.1
PC3	192.168.20.3	255.255.255.0	192.168.20.1
PC4	192.168.30.2	255.255.255.0	192.168.30.1
PC5	192.168.30.3	255.255.255.0	192.168.30.1

Step 3: Select 3 switches (model 2960) to connect PCs in 3 LANs as shown.

Step 4: select 3 routers (model 1841) to connect 3 LANS as shown & configure them as below:

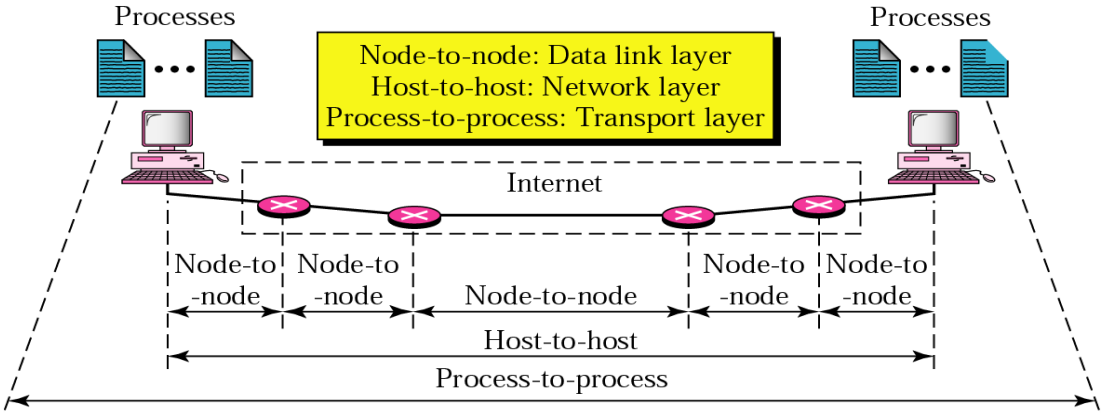
	IP address to connect to respective LANs	IP address to connect to another router
Router 0	192.168.10.1 (LAN 1)	10.0.0.1 (to connect to router 1)
Router 1	192.168.20.1 (LAN 2)	10.0.0.2 (to connect router 0) & 11.0.0.1 (to connect router 2)
Router 2	196.168.30.1 (LAN 3)	11.0.0.2 (to connect to router 1)

Experiment 8 TCP socket programming

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

Theory
TCP (Transmission control protocol)
TCP is used at transport layer.
It is used for process to process communication.
TCP features : reliable (error detection & ACK), connection-oriented (establishes connection before sending, receiving data), byte-stream, full duplex

Difference between data link layer, network layer & transport layer



Socket Programming
Socket is a pair of IP address & port address .
Socket programming is used for communication between the applications (processes) running on same or different machines.
Java Socket programming can be connection-oriented or connection-less & iterative or concurrent
Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming.

ServerSocket class

The **java.net.ServerSocket** class can be used to create a server socket object. This object is used to establish communication with the clients.

Methods

Method	Description
1)public Socket accept( )	establish a connection between server and client.
2) public void bind (Socket Address host, int backlog)	Binds the socket to the specified server and port in the Socket Address object.
2)public synchronized void close( )	Closes the server socket.

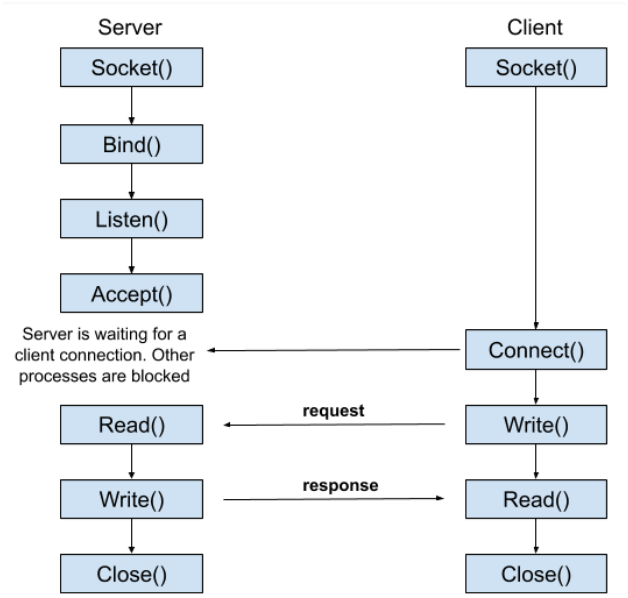
Socketclass

The java.net.Socket class represents the socket that both the client and the server use to communicate with each other. The client obtains a Socket object by instantiating one, whereas the server obtains a Socket object from the return value of the accept( ) method.

methods

Method	Description
1) public Socket(String host,int port)	This method attempts to connect to the specified server at the specified port
3)public Input Stream getInputStream( )	returns the Input Stream attached with this socket.
4)public Output Stream getOutputStream( )	Returns the Output Stream attached with this socket.
5)public synchronized void close( )	Closes this socket

state diagram



The algorithm for Hello:

Send string Hello to each other using getInputStream( )&getOutputStream( )

The algorithm for file transfer:

In server ,Get the file name and store into the BufferedReader, read the content until EOF is reached.

In client read the data send by server& write into a file

The algorithm for calculator:

In client program get two operands & one operation from user & sends to server.

Server does the operation on operands & sends result to the client

Client displays the result

Attach

Print of program & output

Experiment 9 UDP Sockets programming

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

Theory

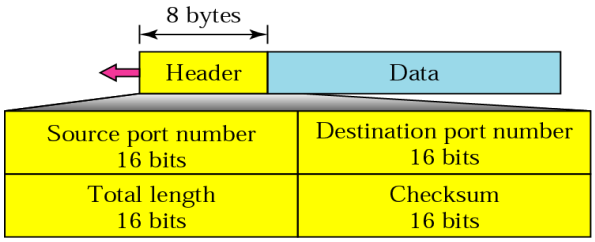
UDP (User datagram protocol)

UDP is used at transport layer.

Transport layer function is process to process communication.

It is connectionless & unreliable (if required , the error checking has to be done by application layer protocol which is using UDP at transport layer)

UDP datagram & header format



Length: defines the total length of the user datagram, header plus data. Total Length can be of 8 to 65,535 bytes

Checksum. used to detect errors over the entire user datagram (header plus data). If error , the datagram is dropped by UDP protocol. It is an **optional** field, which means that it depends upon the application, whether it wants to write the checksum or not. If it does not want to write the checksum, then all the 16 bits are zero; otherwise, it writes the checksum.

	UDP	TCP
Data unit	message	Byte (Stream)
Connection type	Connection less	Connection oriented
Casting	Multicasting & broadcasting	Unicasting
Reliability	unreliable	reliable
congestion control	No	Yes
flow control	No	Yes
security	NO	NO
applications	Video conferencing, audio & video streaming, DNS, VoIP, etc	HTTPS, HTTP, SMTP, POP, FTP, etc

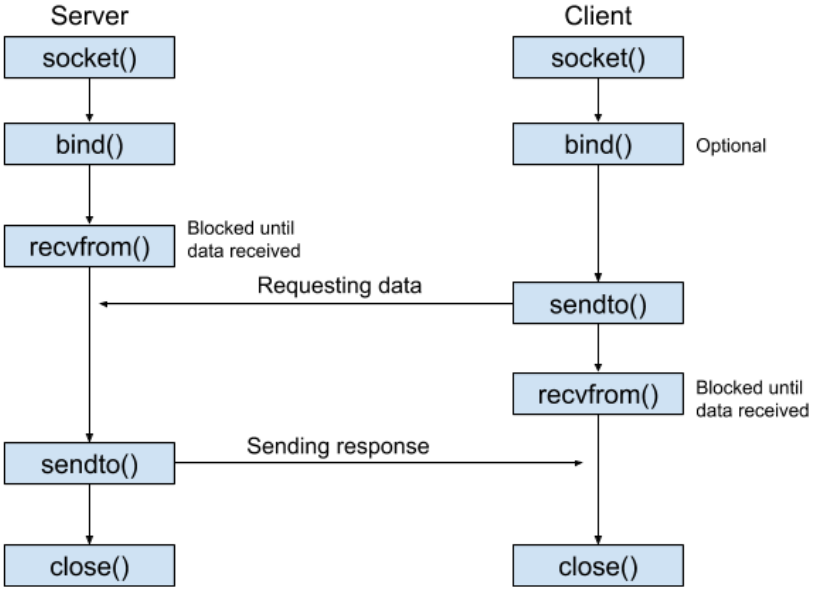
Server:

- Include appropriate header files.
- Create a UDP Socket using socket( ) system call.
- Specify the port where the service will be defined to be used by client.
- Bind the address and port using bind ( ) system call
- Receive a file name of text, audio or video from the Client using recvfrom( ) systemcall.
- Sends file to client using sendto( ) system call.
- Close the server socket

Client:

- Include appropriate header files.
- Create a UDP Socket using socket( ) system call.
- Fill in the socket address structure (with server information)
- Specify the port of the Server, where it is providing service
- Send a file name to the server using sendto( ) system call.
- Receive a file from the Server using recvfrom( ) system call.
- Close the client socket

Socket functions for UDP client/server in Connectionless Scenario



Attach:
Print of program & output

Experiment 10 DNS lookup

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

Theory

DNS (Domain Name System), translates human readable domain names (for example, www.amazon.com) to machine readable IP addresses (for example, 192.0.2.44)

To identify a host, the TCP/IP protocols use the IP address, which uniquely identifies the a host in the Internet.

However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name.

It is impossible to have one single host file to store every address with corresponding name and vice versa. The host file would be too large to store in every host. In addition, it would be impossible to update all the host files every time there is a change

Solution, is to divide this huge amount of information into smaller parts and store each part on a different computer.

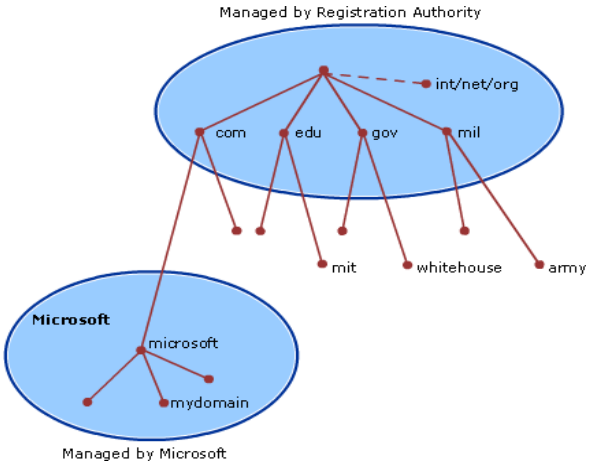
DNS Domain Name Hierarchy

It is hierarchical name space. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have maximum 128 levels.

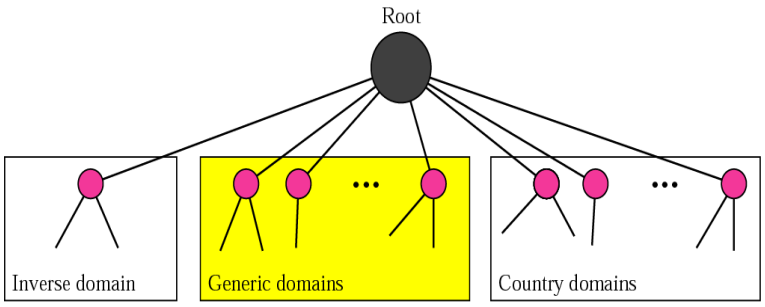
There are two types Fully Qualified Domain Name (FQDN) which end with a null label, but because null means nothing, the label ends with a dot (.) eg challenger.atc.fhda.edu.

Partially Qualified Domain Name (PQDN**)** If a label is not terminated by a null eg Eg challenger

Hierarchy of name servers



DNS in the Internet



Eg of generic domains : .com .org .edu etc
Eg of country domains: .us .uk .in etc

Inverse domain is used to get URL name from a given IP address

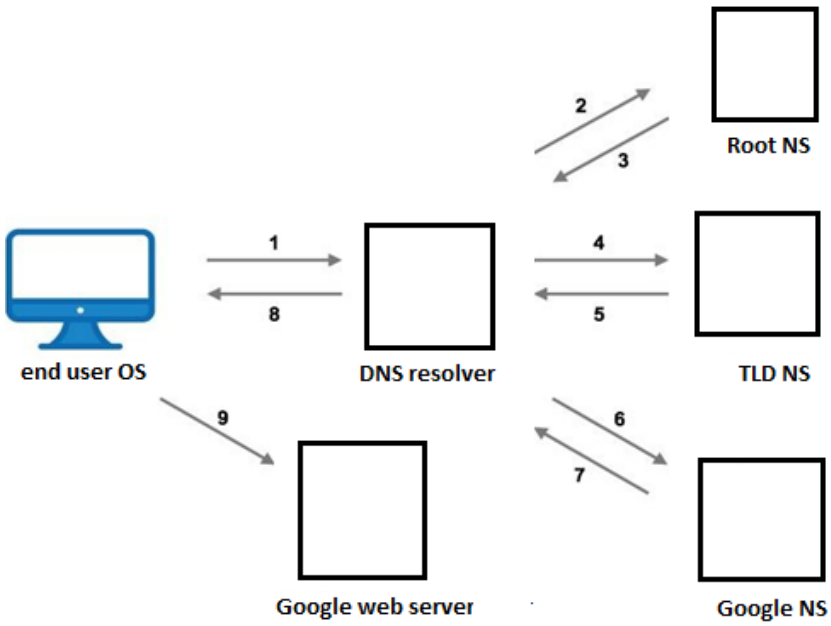
Working of DNS Lookup (DNS resolution)

To resolve DNS queries (to get IP address for a given domain name) , there are Domain Name Servers built across world that takes your DNS look-up request and resolve it. There are 5 types of Name Servers –

1. Caching Name Servers.
2. Recursive Name Servers.
3. Root Name Servers.
4. Top Level Domain (TLD) Name Servers.
5. Authoritative Name Servers.

Caching and Recursive Name Servers are generally provided by Internet Service Providers. Purpose of Caching Name Server is to store known domain names for certain amount of time (similar to caching in data storage). Recursive Name Server performs Full Name Resolution. There are 13 Root Name Servers across globe, responsible for directing requests to appropriate TLD Server.

The DNS Resolution process starts when the user types a URL address (eg google.com) on the browser and hits Enter. Then browser asks the operating system about its IP address



Step 1: OS Recursive Query to DNS Resolver

Since the operating system doesn't know where "www.google.com" is, it queries a DNS resolver. the response must be either an IP address or an error.

Step 2: DNS Resolver Iterative Query to the Root Server

Step 3: Root Server Response
These root servers hold the locations of all of the Top Level Domains (TLDs) such as .com, .edu .org etc.
so it returns the location of the .com servers. The root responds with a list of the 13 locations of the .com gTLD servers, listed as NS or "name server" records.

Step 4: DNS Resolver Iterative Query to the TLD Server
Then the resolver queries one of the .com name servers for the location of google.com..

Step 5: TLD Server Response
The .com gTLD server does not have the IP addresses for google.com, but it knows the location of google.com's name servers. The .com gTLD server responds with a list of all of google.com's NS records. In this case Google has four name servers, "ns1.google.com" to "ns4.google.com."

Step 6: DNS Resolver Iterative Query to the Google.com NS
Finally, the DNS resolver queries one of Google's name server for the IP of "www.google.com."

Step 7: Google.com NS Response
This time the queried Name Server knows the IPs and responds with IPv4 and IPv6 address record .

Step 8: DNS Resolver Response to OS
the resolver has finished the recursion process and sends IP address to operating system of requesting client.

Step 9: Browser Starts TCP Handshake
At this point the operating system, , provides the IP address to the Application (browser), which initiates the TCP connection to start loading the page.

Attach :
Print of program & its output

Experiment 11 Packets analysis using Wireshark

Problem statement : 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.

THEORY:

Packet analyzer:

A packet analyzer (also known as a network analyzer, protocol analyzer or packet sniffer) is a computer program or a computer hardware that can intercept and log traffic passing over a network.

As data streams across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet, and analyzes its content.

Wireshark is the world's widely-used network protocol analyzer. Wireshark is an open source software & is vailable for UNIX and Windows

Wireshark can:

- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and many other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured in various capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

Reasons people use Wireshark:

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems

- QA engineers use it to verify network applications
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

Wireshark does not provide:

- Wireshark isn't an intrusion detection system. It will not warn you when someone does strange things on your network. However, Wireshark might help you figure out what is really going on.
- Wireshark will not manipulate things on the network, it will only "measure" things from it. Wireshark doesn't send packets on the network or do other active things.

List of other packet sniffers:

tcpdump
Windump
Wireshark
tshark
Network Miner

Attach

Print of screenshots of 1) Capture all TCP traffic

2) Capture all HTTP**S** traffic
3) DISPLAY filter expression to count all TCP packets (captured under item #1) thathave the flags SYN, PSH, and RST set
4) count of TCp packets & HTTP packets

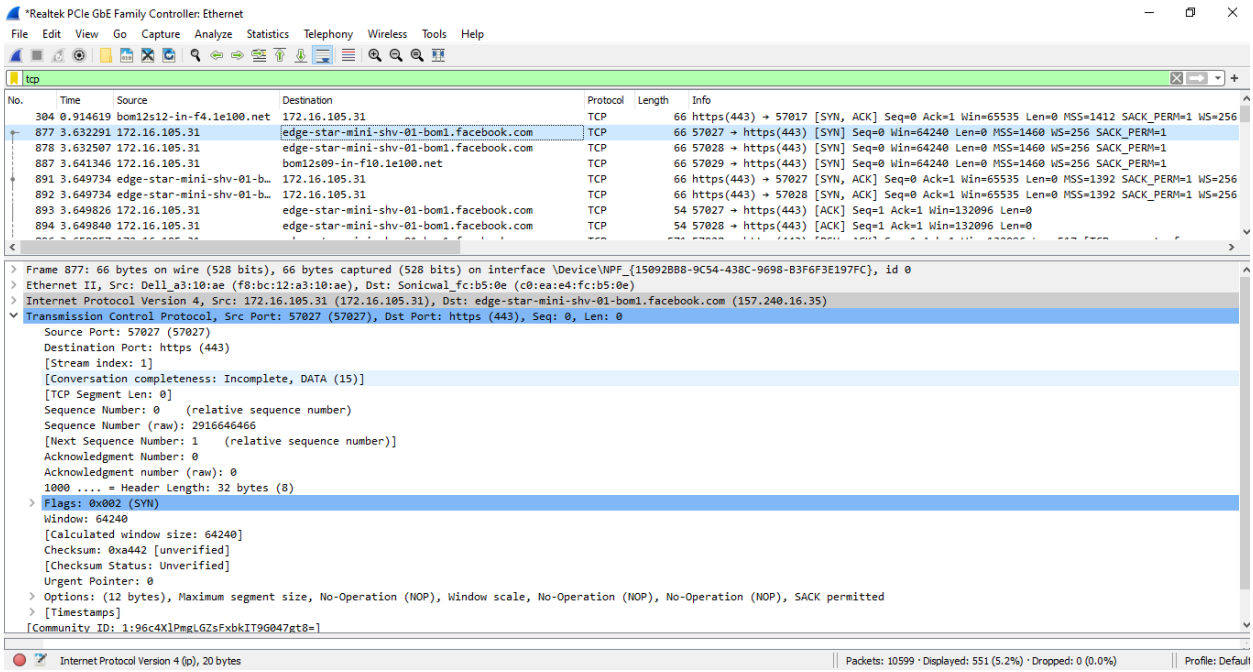
Procedures:

Before each task , clear the cache in the browser eg in chrome browser go to more tools ->clear history -> advanced -> clear data

1.Capture all TCP traffic to/from Facebook, during the time when you log in to your Facebook account

1. From the menu bar, select capture -> options -> interfaces.
2. In the interfaces, choose a particular Ethernet adapter or wireless adapter and note down its IP, and click the start button of the selected adapter.
3. Browse a particular web address to generate traffic to capture packets from the communication for e.g. facebook.com and return to Wireshark and stop the capture by selecting stop from the capture menu.
4. Now we have the captured packets and the captured packet list is on the screen. Since we are interested in only TCP packets, we shall be filtering out TCP packets from the packet pool. You can apply a filter in any of the following ways:
   i) In the display filter bar on the screen, enter TCP and apply the filter or
   ii) From analyze menu in the menu bar select display filters or
   iii) From capture menu in the menu bar select capture filters and then select TCP only..

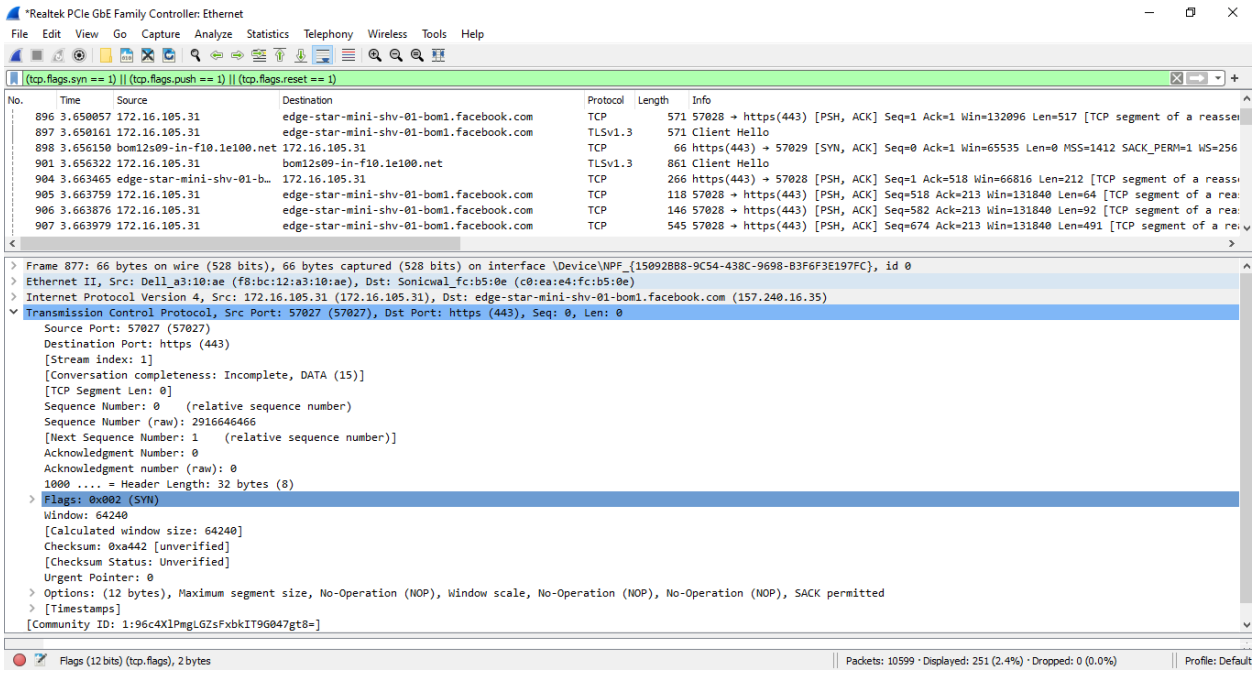
2. Capture all HTTP**S** traffic to/from Facebook, when you log in to your Facebook account
Use Same procedure as in 1.
Wireshark can't display https packets separately as they are sent inside TCP packets



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.

The expression to be typed in display filter is :

(tcp.flags.syn == 1) || (tcp.flags.push == 1) || (tcp.flags.reset == 1)

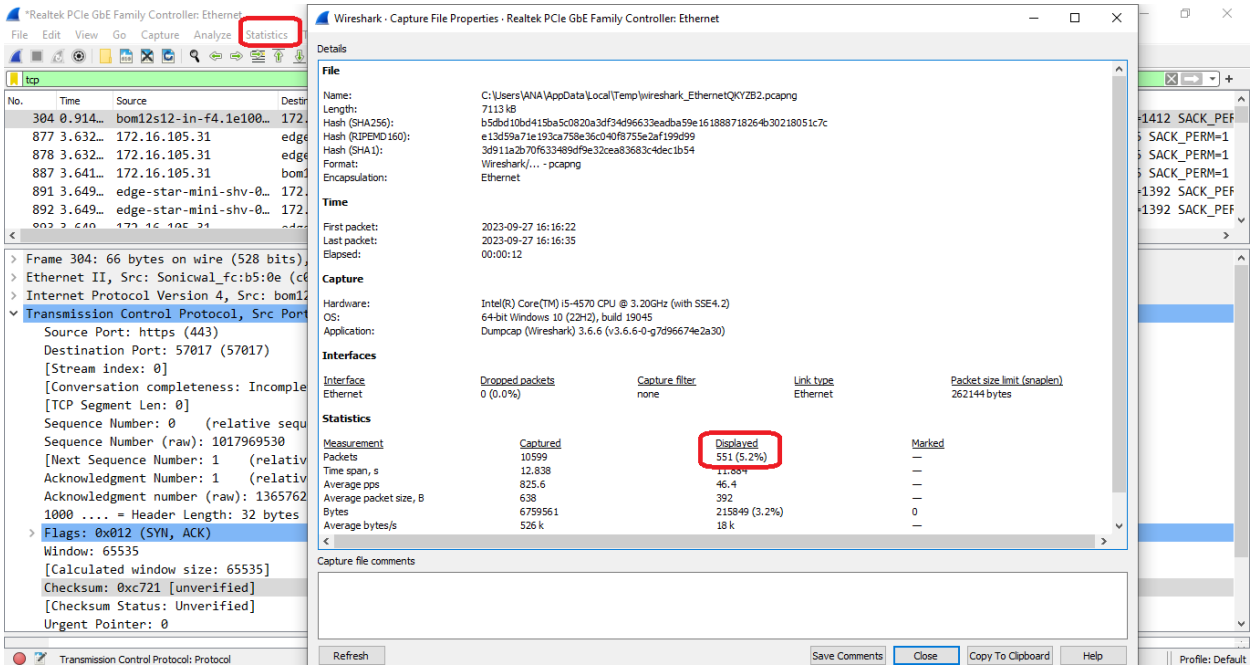


4. Count how many TCP packets you received from / sent to Face book, and how many of each were also HTTP packets.

In display filter , type tcp

Then in menu bar -> statistics -> capture file properties .

This will show count of TCP packets



Experiment 12 HTTP, HTTPS and FTP in packet tracer

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

Theory

FileTransferProtocol(FTP)

FTP is an application layer protocol which transfers files between local and remote file systems.FTP differs from other client-server applications which have only one connection between client & server. FTP establishes **two connections** between the hosts. FTP uses the services of TCP. So it needs two TCP connections. One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient.

In the server ,The well-known port 21 is used for the control connection, and the well-known port 20 is used for the data connection.

when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred**.**

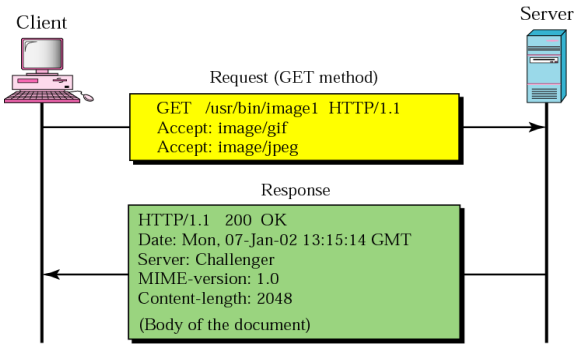
Example of FTP session:

```
$ ftp challenger.atc.fhda.edu
Connected to challenger.atc.fhda.edu
220 Server ready
Name: forouzan
Password: xxxxxxx
ftp > ls /usr/user/report
230 OK
150 Opening ASCII mode
.....
.....
226 transfer complete
ftp > close
221 Goodbye
ftp > quit
```

1. After the control connection is created, the FTP server sends the 220 (service ready) response on the control connection.
2. The client sends its user name.
3. The server responds with 331 (user name is OK, password is required)
4. The client sends the password (not shown).
5. The server responds with 230 (user log-in is OK).
6. The client sends the list command to find the list of files on the directory named report.
7. Now the server responds with 150 and opens the data connection.
8. The server then sends the list of the files or directories (as a file) on the data connection. When the whole list (file) is sent, the server responds with 226 (closing data connection) over the control connection.
9. The client now has two choices. It can use the QUIT command to request the closing of the control connection, or it can send another command to start another activity (and eventually open another data connection). In above example, the client sends a QUIT command.
10. After receiving the QUIT command, the server responds with 221 (service closing) and then closes the control connection.

Hypertext Transfer Protocol (HTTP)

The Hypertext Transfer Protocol (HTTP) is an application-level protocol.
HTTP uses the services of TCP on well-known port 80**.**
HTTPis a stateless protocol. i.e.server maintains no information about previous client requests.
HTTP was developed alongside HTMLto create the first interactive, text-based web browser to access the original World Wide Web.



In This example, the client retrieves a document from server. It uses GET method to retrieve an image with the path /usr/bin/image1. HTTP version used is (1.1). client can accept images in GIF and JPEG format. The request does not have a body.
The response message contains the status line and 4 lines of header. Status line shows HTTP version used is 1.1. and status code 200 means OK means request is successful. The header lines define the date, server, MIME version, and length of the document followed by document

HTTPS (HTTP over SSL or HTTP Secure)

HTTPS is an abbreviation of **Hypertext Transfer Protocol Secure**. It is a secure version of HTTP. This protocol is used for providing security tothe data sent between a website and the web browser. This protocol uses the 443 port number for communicating the data. This protocol is also called HTTP over SSL because the HTTPS communication protocols are encrypted using the SSL(SecureSocketLayer). But now SSL is replaced by new protocol: TSL. Those websites which need login credentials should use the HTTPS protocol for sending the data.

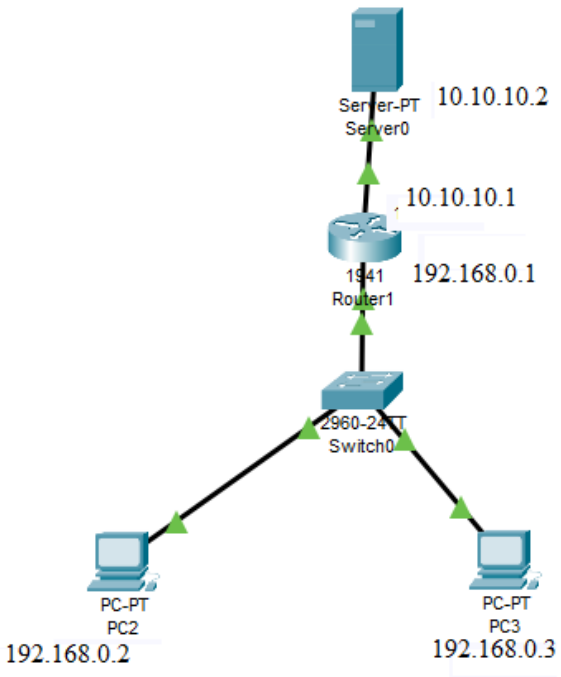
Difference between HTTP and HTTPS

HTTP	**HTTPS**
Hypertext Transfer Protocol	Hypertext Transfer Protocol Secure.
Operates at the application layer.	Operates at the transport layer.
The data which is transferred is plain text.	The data is encrypted, i.e. cipher text.
So It is un-secure.(any hacker can see the contents of packet using sniffer softwares	So It is highly secure.
Does not need any certificate.	Requires an SSL (Secure Socket Layer) certificate.
Uses port number80.	Uses port number443.
The URL start with http://	The URL start with https://
The speed is faster than HTTPS.	The speed is less than HTTP.
Examples of HTTP websites are Educational Sites, Internet Forums, etc.	Examples of HTTPS websites are shopping websites, banking websites, etc.

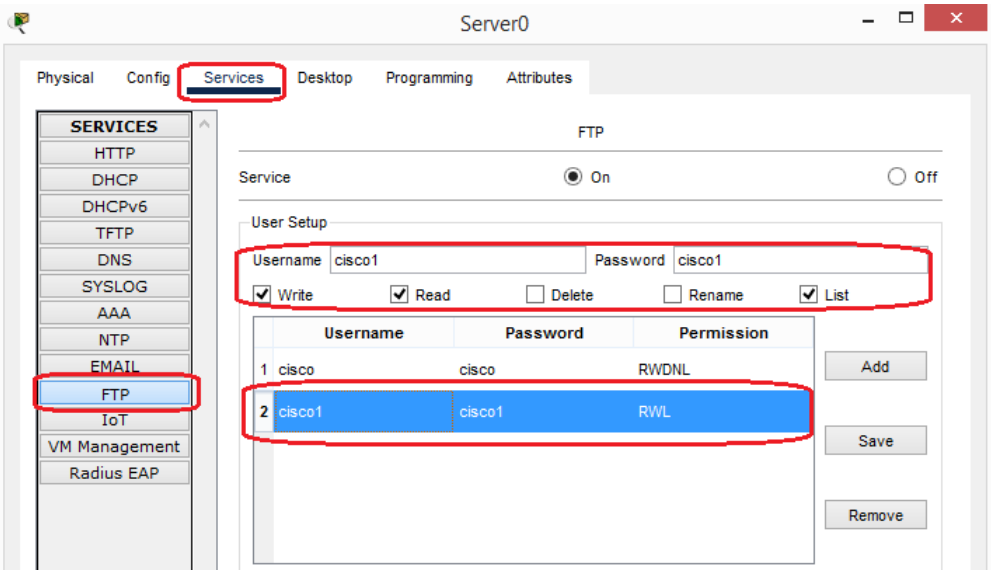
Implementation of FTP , HTTP , HTTPS using Packet tracer

Set up network as shown
Setup addresses for PC2 PC3 , router & server as shown

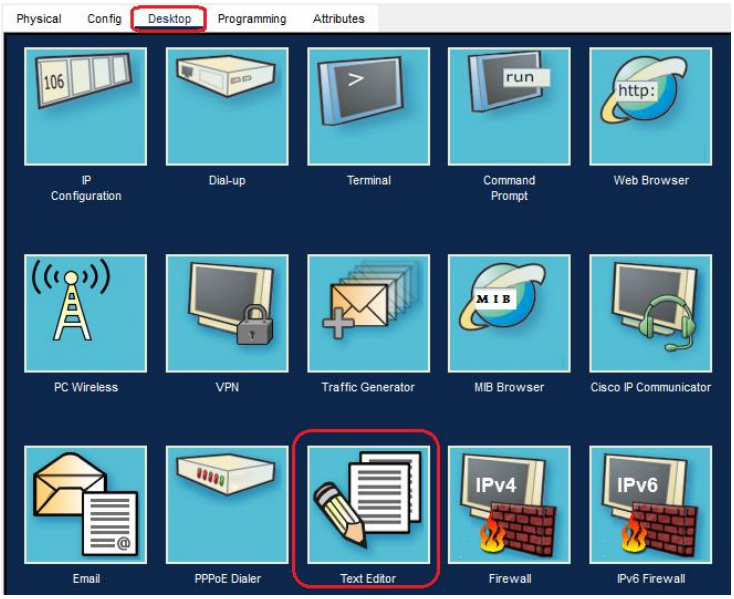
Device	IP address	subnet	gateway
PC2	192.168.0.2	255.255.255.0	192.168.0.1
PC3	192.168.0.3	255.255.255.0	192.168.0.1
Router Fast Ethernet 0/1	192.168.0.1	255.255.255.0	---
Router Fast Ethernet 0/2	10.10.10.1	255.0.0.0	----
Server	10.10.10.2	255.0.0.0	192.168.0.1



In server, select service“ FTP” , set new username & password as cisco1 with required permissions



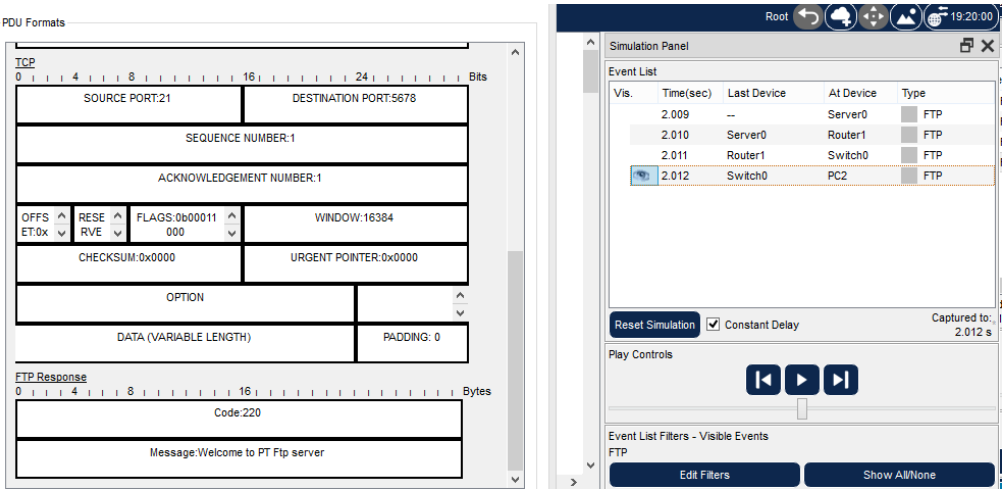
In PC2 create a file using text editor & save it by name= test.txt



select command prompt& login to server using command =ftp 10.10.10.2
Enter username and password as= cisco1

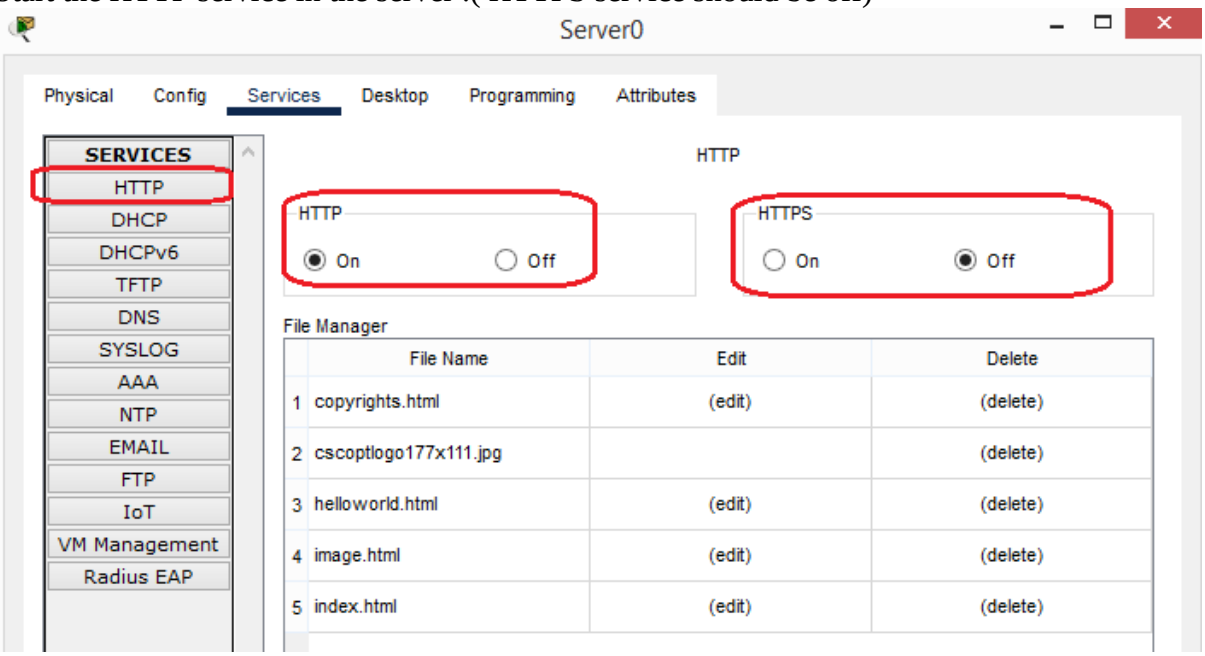
```
C:\>ftp 10.10.10.2
Trying to connect...10.10.10.2
Connected to 10.10.10.2
220- Welcome to PT Ftp server
Username:cisco1
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put test.txt

Writing file test.txt to 10.10.10.2:
File transfer in progress...

[Transfer complete - 40 bytes]

40 bytes copied in 0.075 secs (533 bytes/sec)
ftp>
```

Send file “test.txt” to server using command = Put test.txt
Now check if it is transferred to server by using command = dir
Now login through PC3
Download the file test.txt by using command = get 10.10.10.2
We can also see ftp packets being sent between PC and server , by creating complex PDU having FTP



Installation of HTTP / HTTPS server

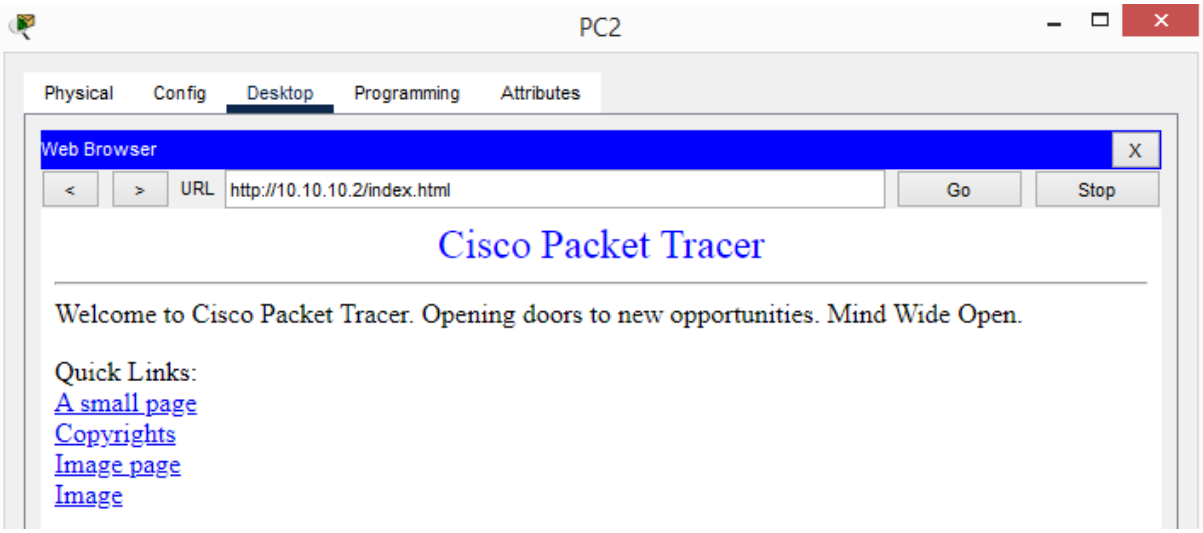
Start the HTTP service in the server .(HTTPS service should be off)



In PC2 open web browser through desktop tab

Enter url = 10.10.10.2

The web pages can be viewed.



Similarly , HTTPS service can be checked in packet tracer

FTP , HTTP packet analyzing using Wireshark

In Wireshark , start capturing

In windows command prompt Or in linux terminal, login to any open ftp server eg.

run the command = ftp ftp.gnu.org

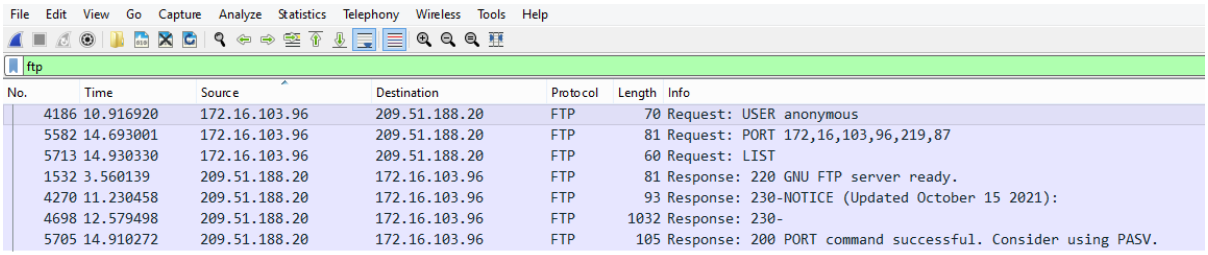
enter the username as= anonymous

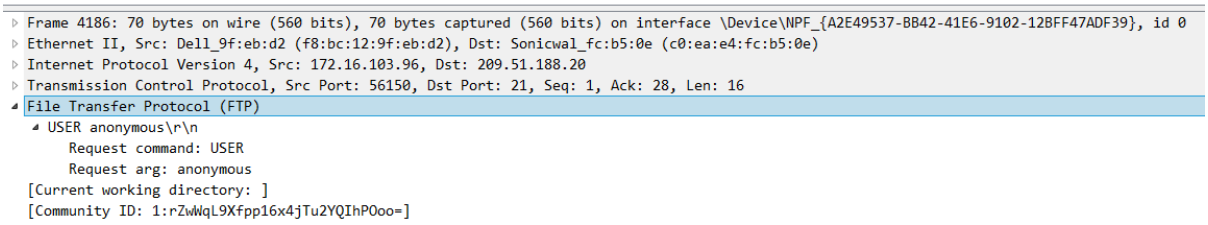
stop capturing

In display filter , type = ftp

We will see only ftp packets in packet list window.

Select ftp packets one by one and see details on packet details window



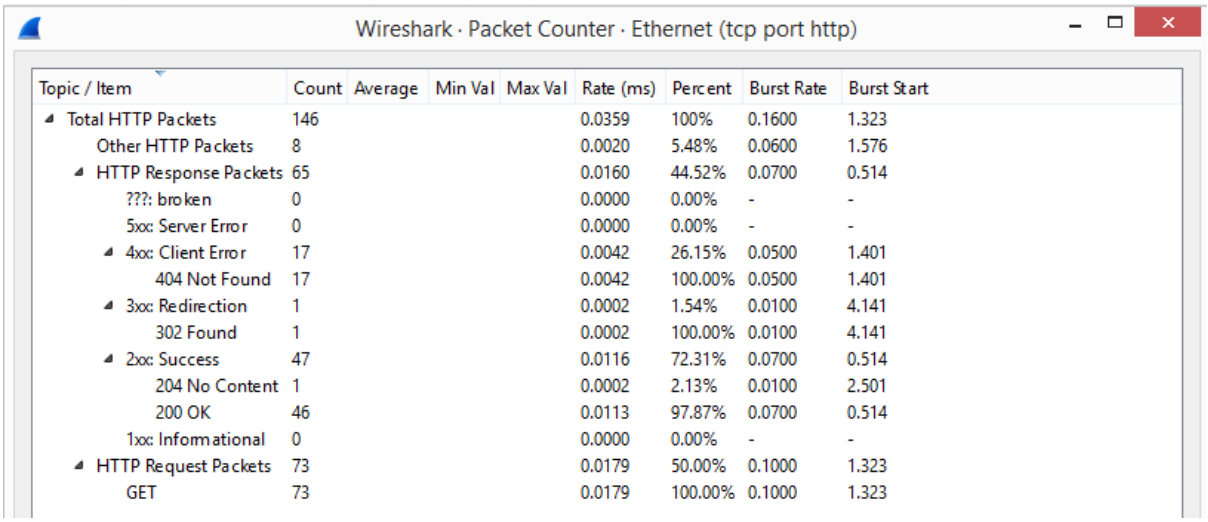


Capture HTTP traffic.

In packet details window , we can see details of HTTP packets.

Below is the screen shot of Get request from browser

Also in statistics tab choose HTTP protocol & see various statistics



Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total HTTP Packets	146				0.0359	100%	0.1600	1.323
Other HTTP Packets	8				0.0020	5.48%	0.0600	1.576
HTTP Response Packets	65				0.0160	44.52%	0.0700	0.514
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	17				0.0042	26.15%	0.0500	1.401
404 Not Found	17				0.0042	100.00%	0.0500	1.401
3xx: Redirection	1				0.0002	1.54%	0.0100	4.141
302 Found	1				0.0002	100.00%	0.0100	4.141
2xx: Success	47				0.0116	72.31%	0.0700	0.514
204 No Content	1				0.0002	2.13%	0.0100	2.501
200 OK	46				0.0113	97.87%	0.0700	0.514
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	73				0.0179	50.00%	0.1000	1.323
GET	73				0.0179	100.00%	0.1000	1.323

Experiment 13 SSL protocol

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

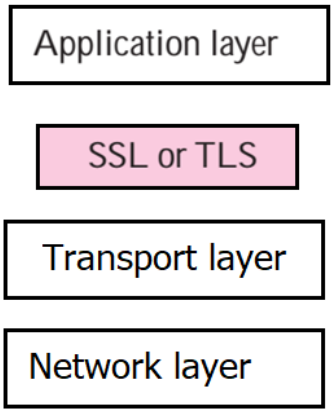
Theory

Network Security refers to the measures taken by any organization to secure its computer network and data against threats ,using both hardware and software systems.

Some of the threats are : Malware Attack, Phishing Attack, Password Attack, Man-in-the-Middle Attack, SQL Injection Attack, Denial-of-Service Attack & Cryptojacking etc

Secure Socket Layer (SSL)
Secure sockets layer (SSL) is a networking protocol at transport layer .
It is used for secure connection between web client and web server over an insecure network, eg the internet.
SSL was deprecated in 2015
Transport Layer Security (TLS)
is upgraded version of SSL. TLS is a more secure and efficient protocol

Application layer protocols use SSL for security , Eg, HTTPS is secured version of HTTP. & HTTPS uses SSL or TSL at transport layer.



SSL provides following security services on data received from the application layer.

- Message Integrity (message Authentication)
- Confidentiality of data (by encryption)
- Authentication (of web server by checking its SSL certificate)

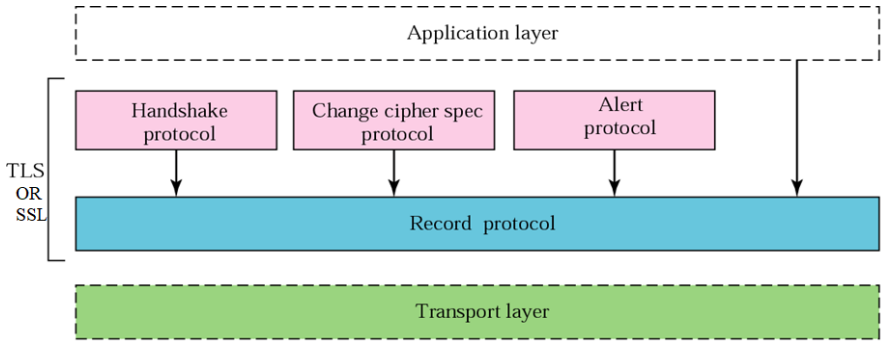
To provide these security services , SSL uses:

Key Exchange Algorithms,
the client and the server each need a set of cryptographic secrets. However, to create these secrets, one pre-master secret must be established between the two parties. SSL defines several key-exchange methods to establish this pre-master secret.
Encryption/Decryption Algorithms
The client and server also need to agree to a set of encryption and decryption algorithms.
Hash Algorithms
SSL uses hash algorithms to provide message integrity (message authentication).

Cipher Suite:
The combination of key exchange, hash, and encryption algorithms defines a cipher suite for each SSL session

SSL accomplishes its tasks , by using four protocols in two layers,

SSL Protocol Stack:



Summary of four protocols :

The Handshake Protocol provides security parameters for the Record Protocol. It establishes a cipher suite. It also authenticates the server to the client and the client to the server (if needed.)

The ChangeCipherSpec Protocol is used for signalling the readiness of cryptographic secrets.
The Alert Protocol is used to report abnormal conditions.

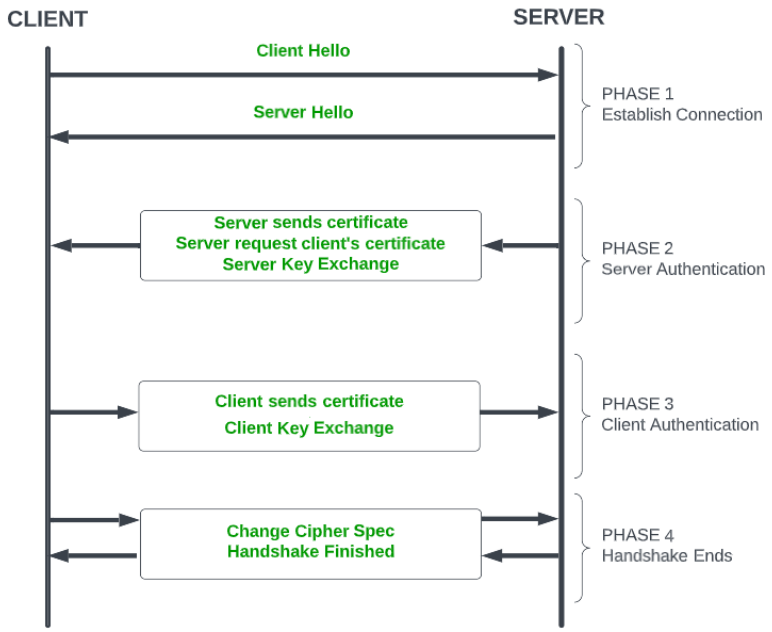
The Record Protocol is the carrier. It carries messages from three other protocols as well as the data coming from the application layer. Messages from the Record Protocol is sent to the transport layer, normally TCP.

Details of four protocols

Handshake Protocol:

It is used to establish sessions. It allows the client and server to authenticate each other by sending a series of messages to each other. It uses four phases.

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



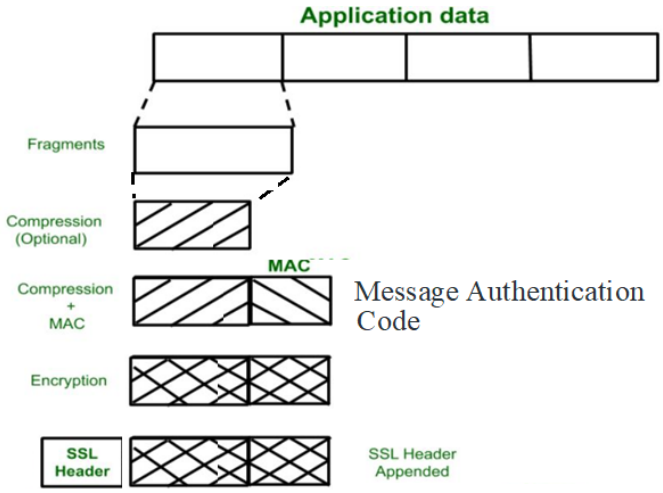
SSL HANDSHAKE PROTOCOL

SSL Record Protocol:

provides two services to SSL connection.

- Confidentiality
- Message Integrity

In this , data is divided into fragments. Each fragment is compressed (optional)and then encrypted MAC (Message Authentication Code) is generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) ,it is appended to each fragment. After that a SSL header is appended to it.



Change-cipher Protocol:

The “Change Cipher Spec” message lets the other party know that, it has generated the session key and is going to switch to encrypted communication.

Alert Protocol:

This protocol is used to send SSL-related alerts by server to client or vice versa.

Warning (level = 1):

This Alert does not close the connection between sender and receiver. Eg.

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

Fatal Error (level = 2):

This Alert terminates the connection between sender and receiver. Eg.

Handshake failure: When the sender is unable to negotiate an acceptable set of security parameters.
Decompression failure: error in decompression.
Illegal parameters: parameters are not valid
Bad record MAC: When an incorrect MAC was received.
Unexpected message: When an inappropriate message is received.

Attach :

Printout of screen shots of wireshark showing SSL packets after visiting any SSL secured website (banking, e-commerce etc.)

Experiment 14 IPsec

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

Theory

IPsec is a set of protocols that provide security by providing authentication & encryption of traffic between two nodes. IPsec works at the **network level (i.e. in IP)**. IPSec helps create authenticated and confidential packets for the IP layer.

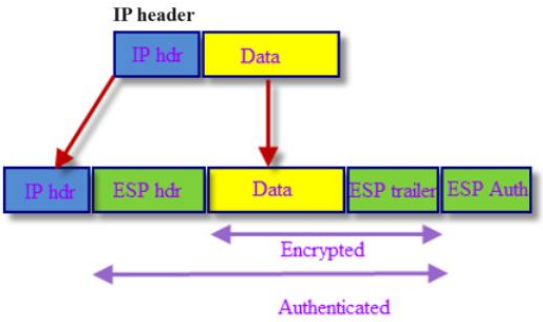
ISAKMP provides a framework for authentication, key management and supports many key exchange protocols like Oakley, Diffie-Hellman, the enhanced Diffie-Hellman and the RSA-based key exchange. AH protocol provides data origin authentication, data integrity, and replay protection. However, AH does not provide data confidentiality, which means that all of our data is sent in the clear text. Unlike AH, ESP protocol provides data confidentiality, and also optionally provides data origin authentication, data integrity checking, and replay protection. The difference between ESP and the Authentication Header (AH) protocol is that ESP provides encryption, while both protocols provide authentication, integrity checking, and replay protection

AH versus ESP

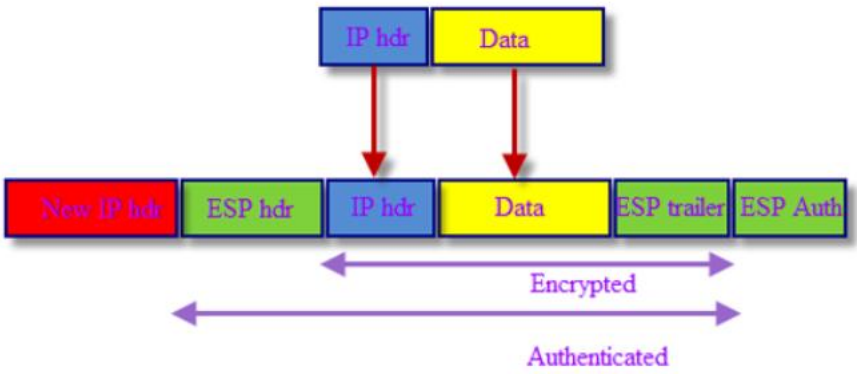
The ESP protocol was developed after the AH protocol. ESP does whatever AH does with additional functionality (i.e. confidentiality). Then, why do we need AH? The answer is that we don't. However, the implementation of AH is already included in some commercial products, which means that AH will remain part of the Internet until these products are phased out.

IPSec operates in one of two different modes: transport mode or tunnel mode

Transport mode : In this IP header is neither encrypted nor authenticated, only data is encrypted & authenticated by using ESP



Tunnel mode: In this entire IP packet including IP header is encrypted & authenticated

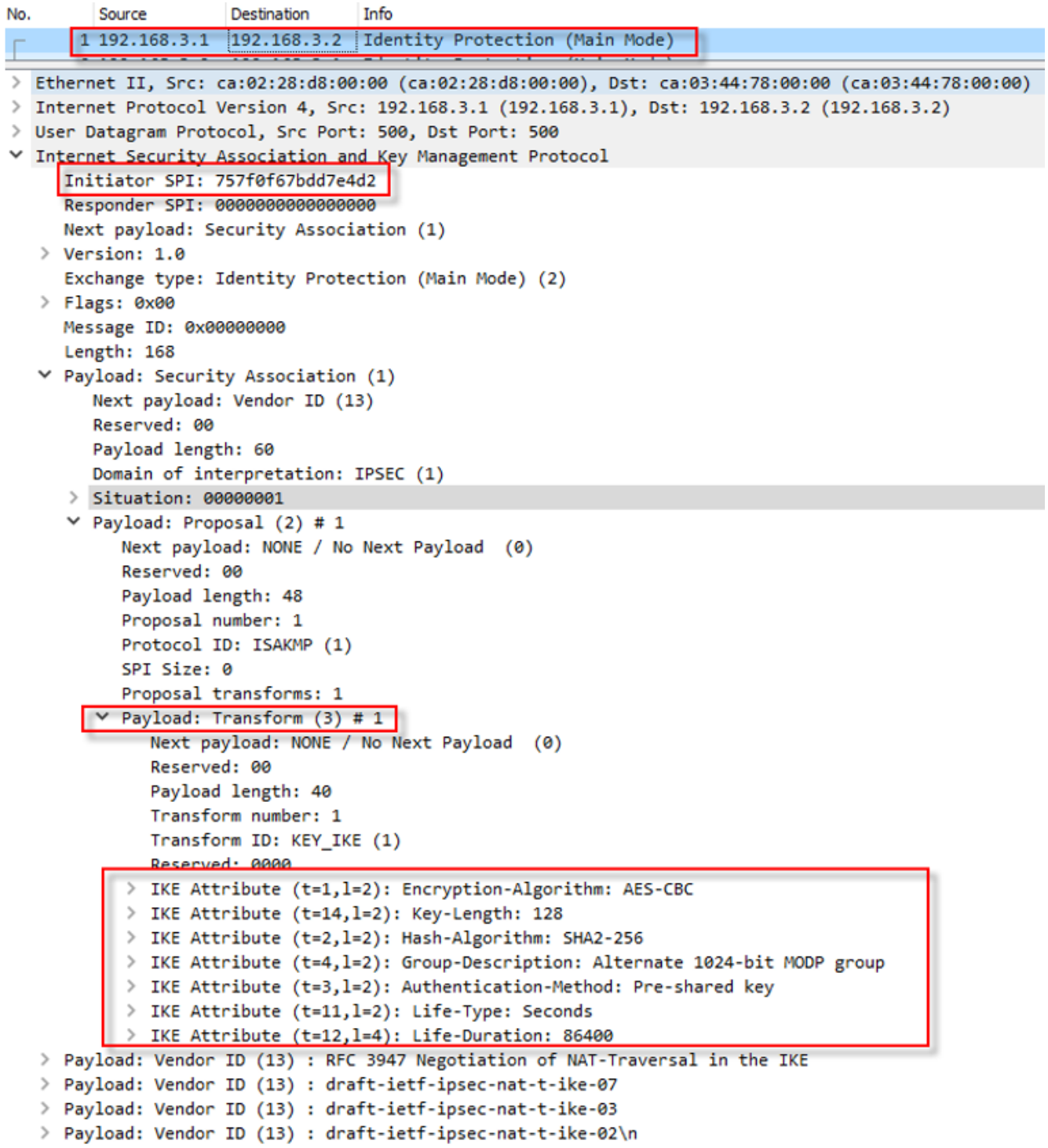


IPsec ISAKMP negotiations are made in two phases, Main Mode (Phase1) and Quick Mode (Phase2). Main mode (Phase1) authenticates the parties and is partially encrypted. Quick mode (Phase 2) negotiates the algorithms and agree on which traffic will be sent across the IPsec tunnel.

No.	Source	Destination	Info
1	192.168.3.1	192.168.3.2	Identity Protection (Main Mode)
2	192.168.3.2	192.168.3.1	Identity Protection (Main Mode)
3	192.168.3.1	192.168.3.2	Identity Protection (Main Mode)
4	192.168.3.2	192.168.3.1	Identity Protection (Main Mode)
5	192.168.3.1	192.168.3.2	Identity Protection (Main Mode)
6	192.168.3.2	192.168.3.1	Identity Protection (Main Mode)
7	192.168.3.1	192.168.3.2	Quick Mode
8	192.168.3.2	192.168.3.1	Quick Mode
9	192.168.3.1	192.168.3.2	Quick Mode
10	1.1.1.1	2.2.2.2	Echo (ping) request id=0x0000, seq=1/256, ttl=254 (reply in 11)
11	2.2.2.2	1.1.1.1	Echo (ping) reply id=0x0000, seq=1/256, ttl=254 (request in 10)
12	1.1.1.1	2.2.2.2	Echo (ping) request id=0x0000, seq=2/512, ttl=254 (reply in 13)
13	2.2.2.2	1.1.1.1	Echo (ping) reply id=0x0000, seq=2/512, ttl=254 (request in 12)
14	1.1.1.1	2.2.2.2	Echo (ping) request id=0x0000, seq=3/768, ttl=254 (reply in 15)
15	2.2.2.2	1.1.1.1	Echo (ping) reply id=0x0000, seq=3/768, ttl=254 (request in 14)
16	1.1.1.1	2.2.2.2	Echo (ping) request id=0x0000, seq=4/1024, ttl=254 (reply in 17)
17	2.2.2.2	1.1.1.1	Echo (ping) reply id=0x0000, seq=4/1024, ttl=254 (request in 16)

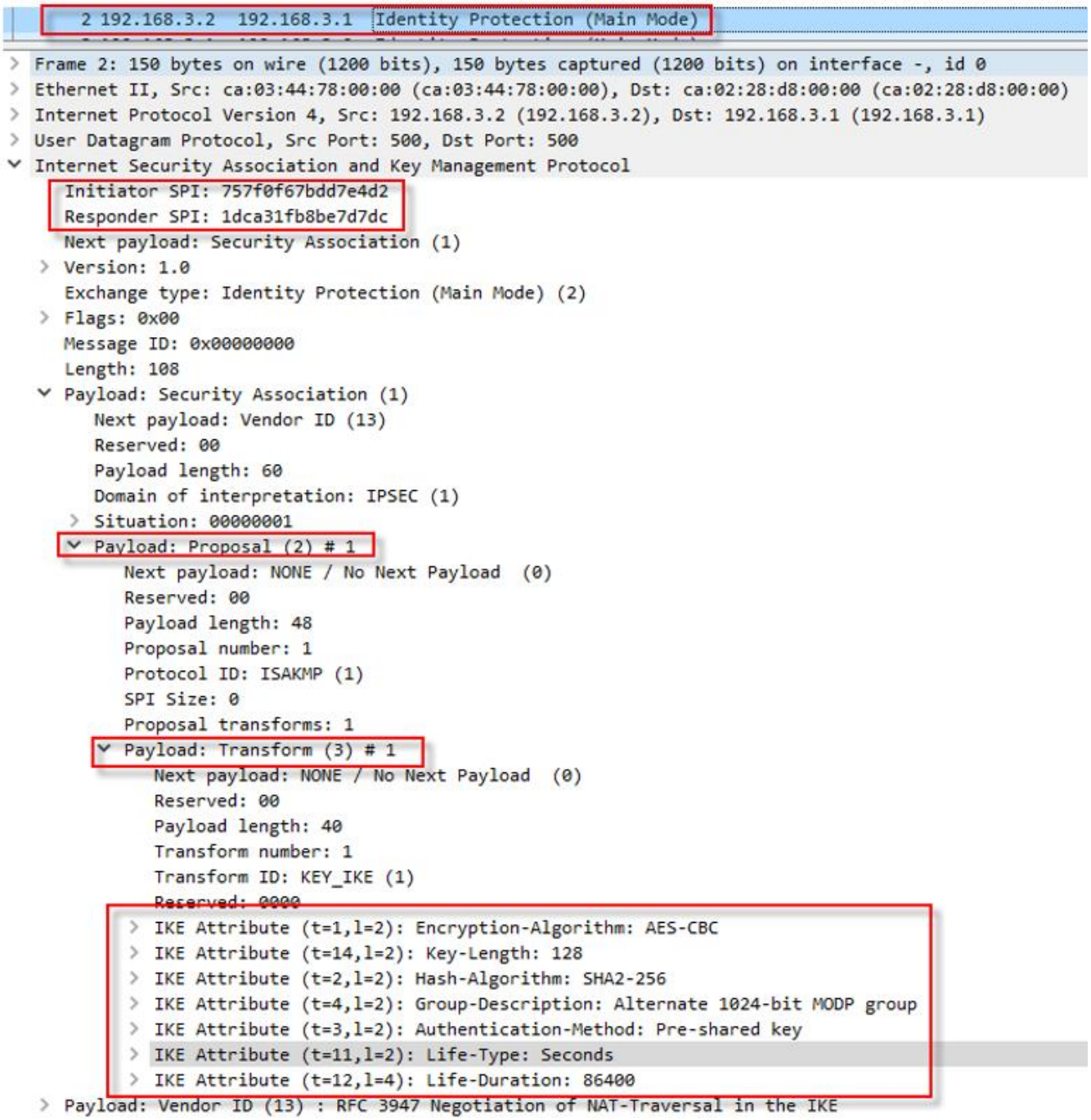
Packet 1 :

It is the first packet sent by the initiator(192.68.3.1) . It contains proposal for the security association(containing a set of security parameters) As seen below, the initiator negotiates for:
Encryption Algorithm as AES-CBC
AES-CBC key length as 128 bits,
Hash Algorithm as SHA2-256,
Diffie Helman Group as Group 2 (Alternate 1024-bit MODP group),
Authentication Method as Pre-shared key,
Life Type of the Phase 1 Tunnel as Seconds,
And Life Duration as 86400 seconds.



Packet 2:

The responder (192.68.3.2) sends a packet with its proposals (It picks one of the **Transforms)**



In next 2 Identity Protection packets, both peers exchange Diffie-Hellman public key values and nonces (random numbers) which will then allow both peers to agree on a shared secret key

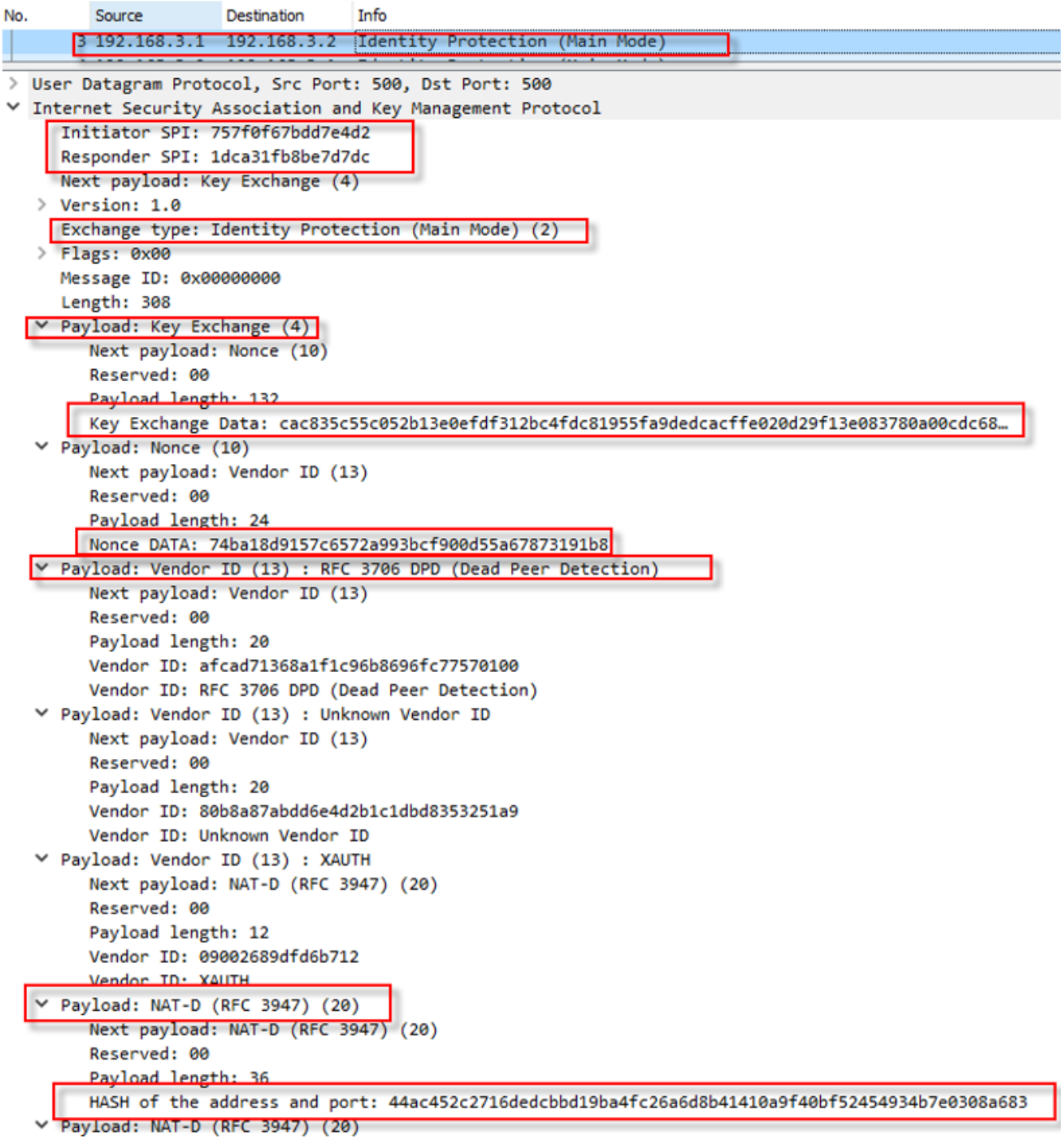
Packet number 3:

Initiator (192.68.3.1) sends next packet no.3 .This packet contains payload such as:

Key Exchange is used to send Diffie Helman public key.

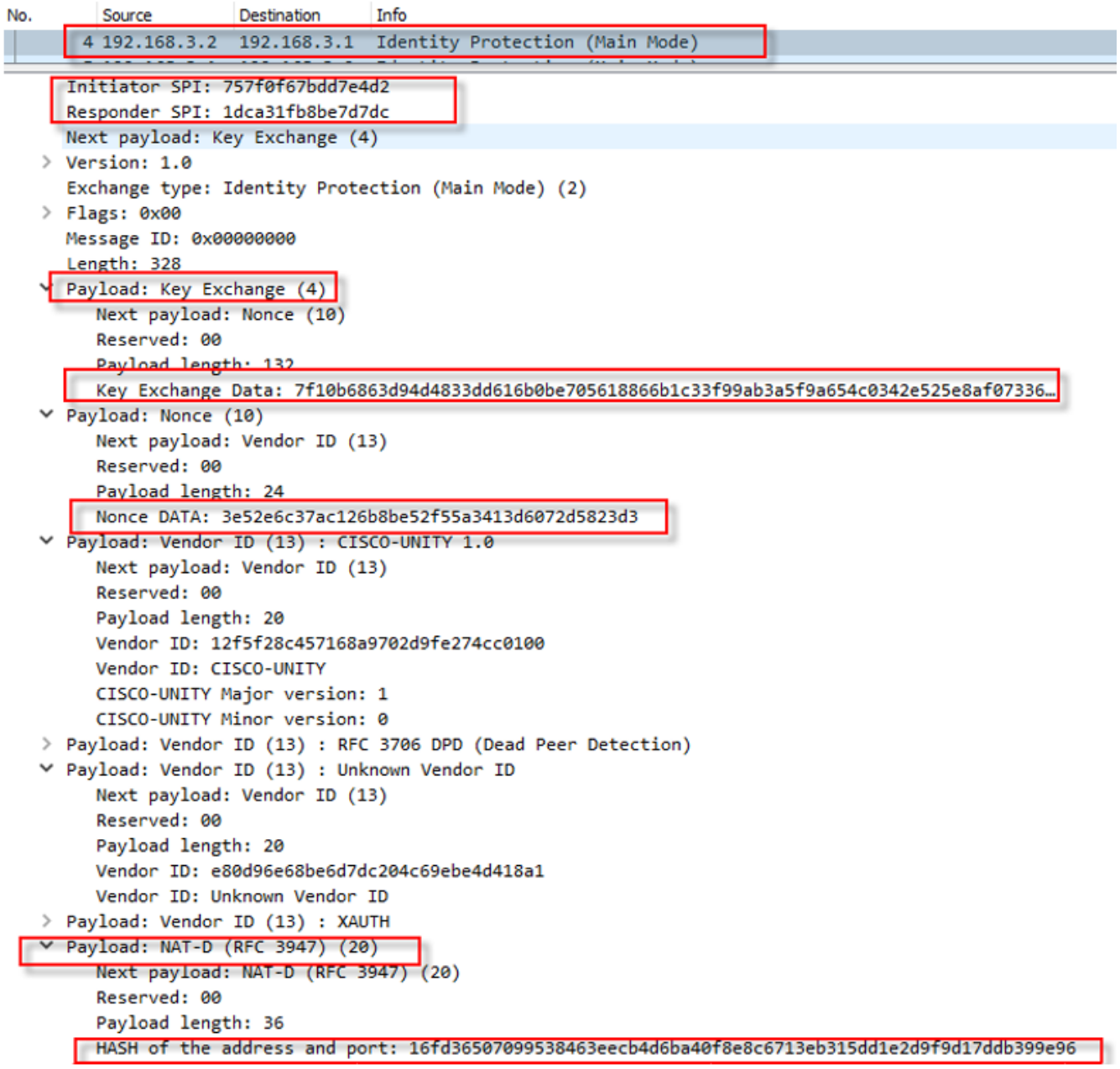
Nonce is a randomly generated number and used to prevent replay attacks from generating bogus SAs.

NAT-D is used to notice if a NAT device exists between gateways.



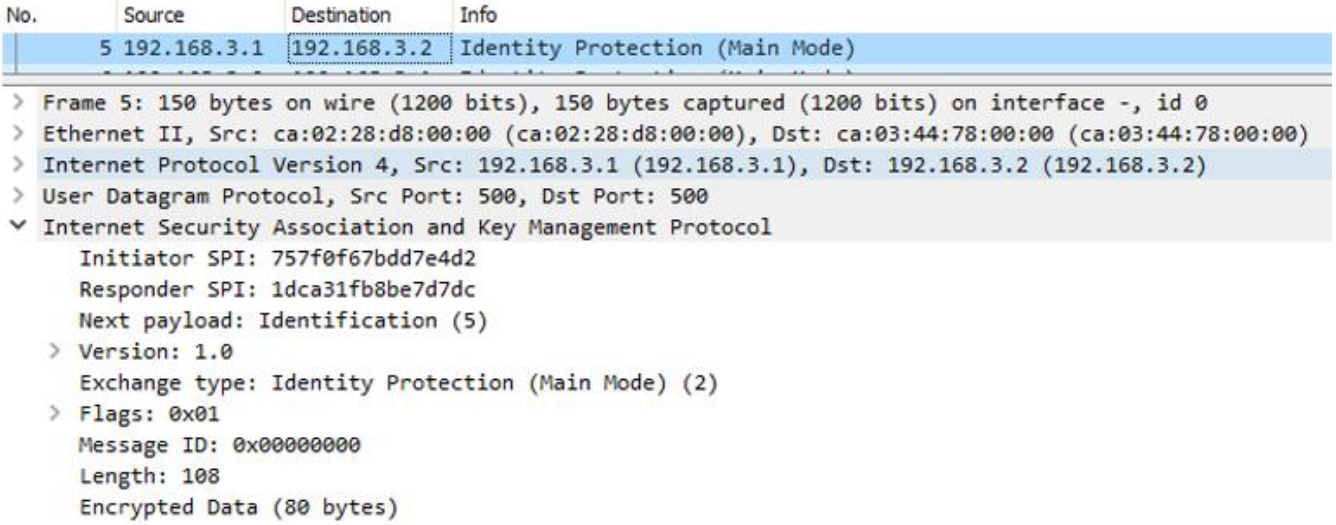
Packet number 4

The responder sends its payload such as Key Exchange, Nonce and NAT-D in return.



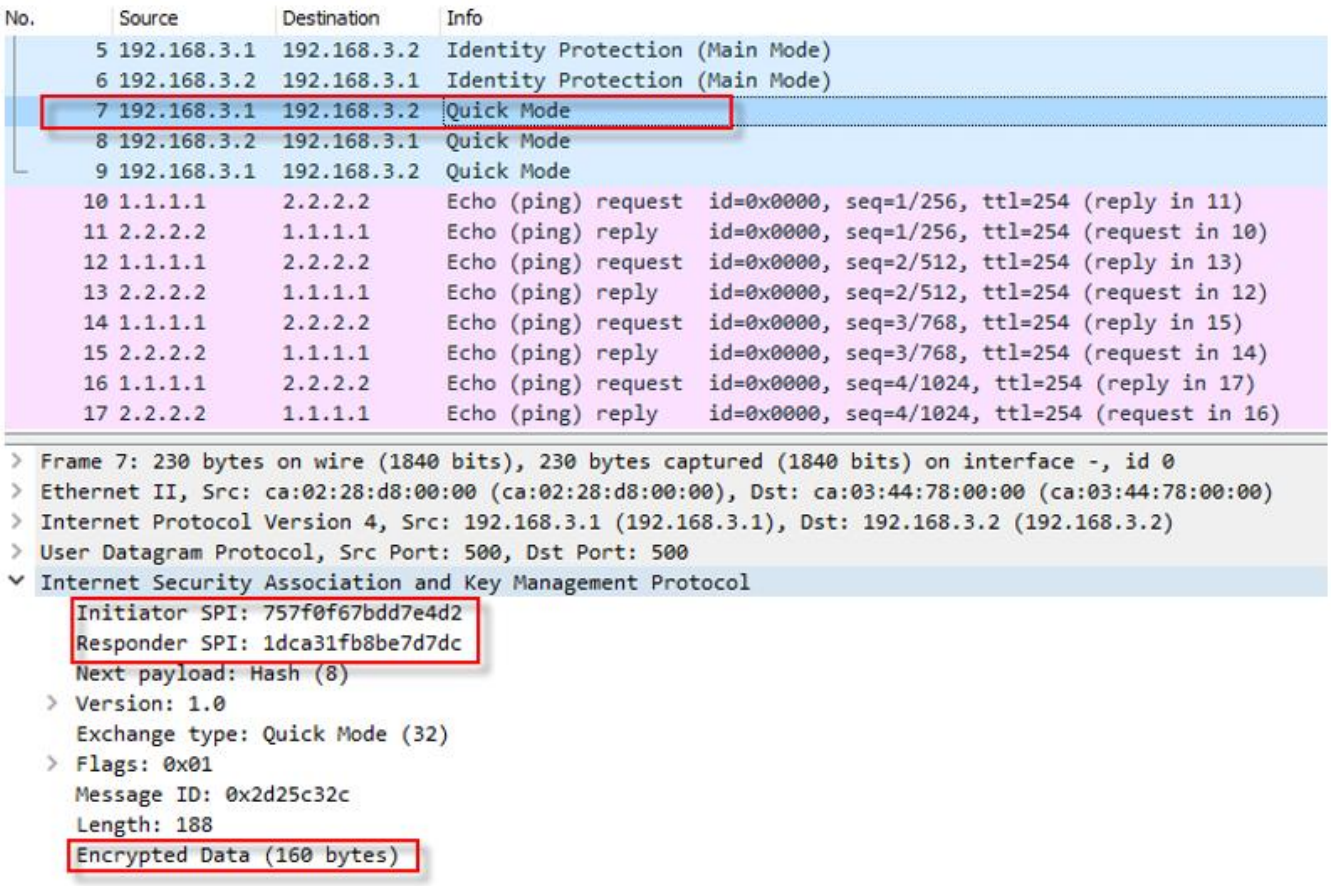
Packets number 5-6:

At this point, the traffic between both parties will be encrypted. These packets are used for identification and authentication of each peer



Packets number 7-9

These packets are fully encrypted. The packets contain the security association for IPsec tunnel protected by ESP or AH. It negotiates the parameters defined in the transform set configuration and mode of the tunnel (transport mode or tunnel mode)



Packets number 10-17

These packets contain the actual data tunneled by IPsec. Since we used AH for demonstration purpose, we can see the data in clear text. If we change the AH to ESP in the transform set configuration, the data will be encrypted. As seen below the ICMP packet from 1.1.1.1 to 2.2.2.2 is encapsulated with AH and tunneled through (192.168.3.1) to (192.168.3.2)