4.1 Experiment no. 1

Aim: Types of topologies and types of transmission media.

Objectives:

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

Tools Required:

Software: Cisco Packet Tracer

Procedure:

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

Theory:

Cisco Packet Tracer:

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

Types of Topologies:

Bus Topology

In local area network, it is a single network cable runs in the building or campus and all nodes are connected along with this communication line with two endpoints called the bus or backbone. In other words, it is a multipoint data communication circuit that is easily control

data flow between the computers because this configuration allows all stations to receive every transmission over the network. For bus topology we build network using three genericpc which are serially connected with three switches using copper straight through cable and switches are interconnected using copper cross over cable.

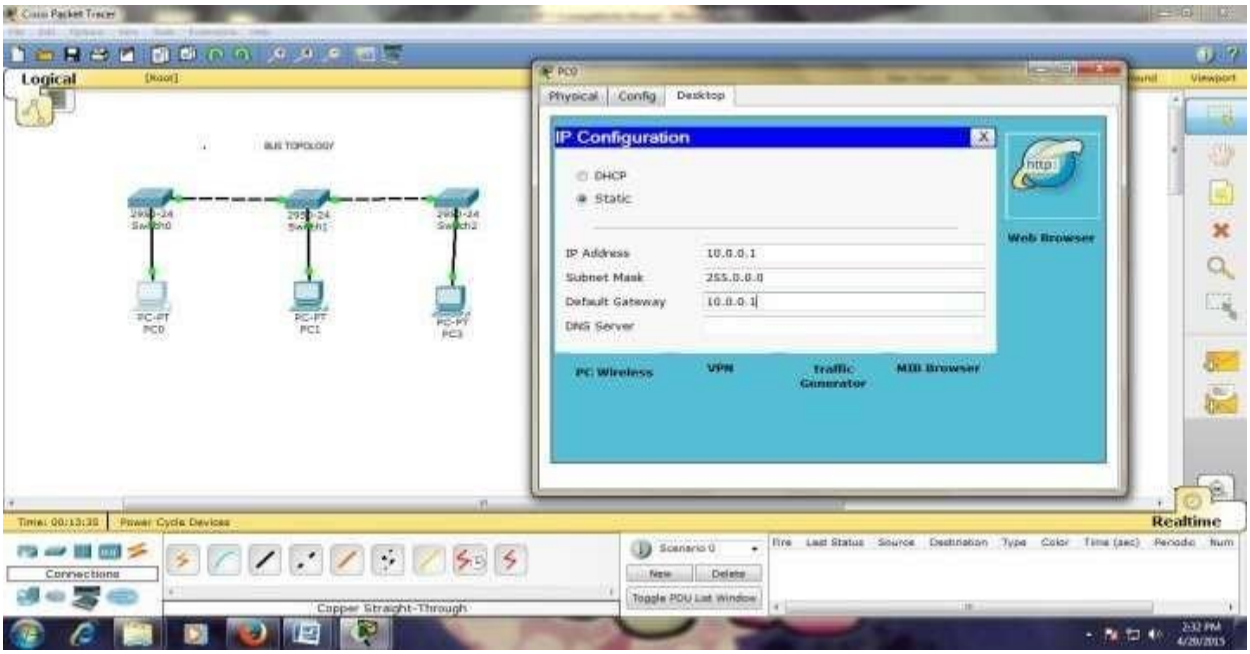


Fig -1: Design of bus topology

Star Topology:

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

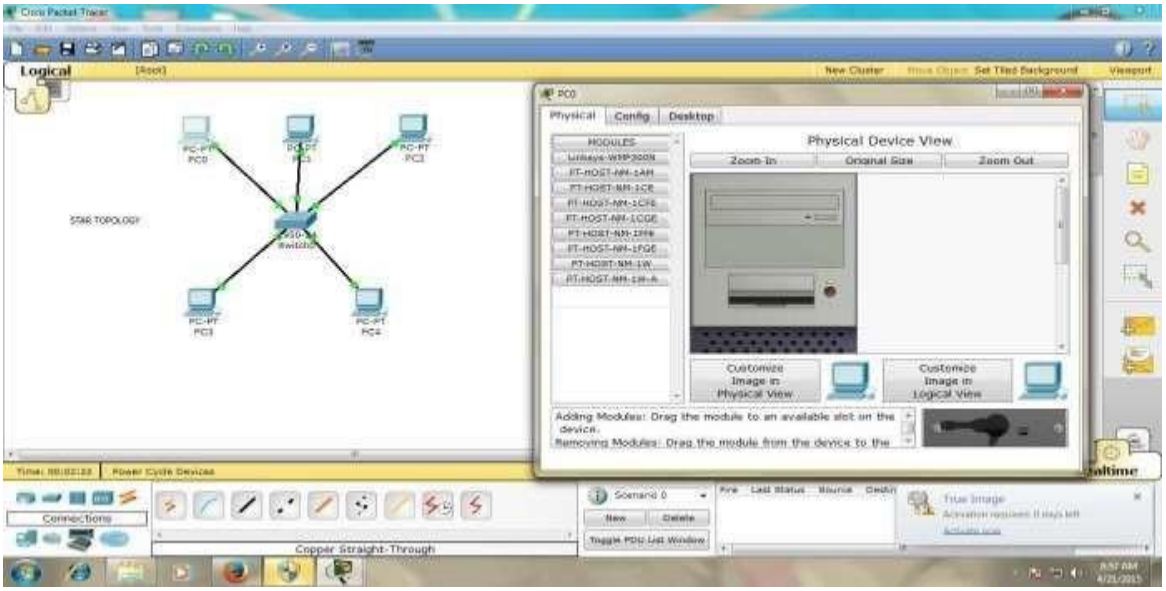


Fig -2: Design of star topology

Mesh Topology:

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

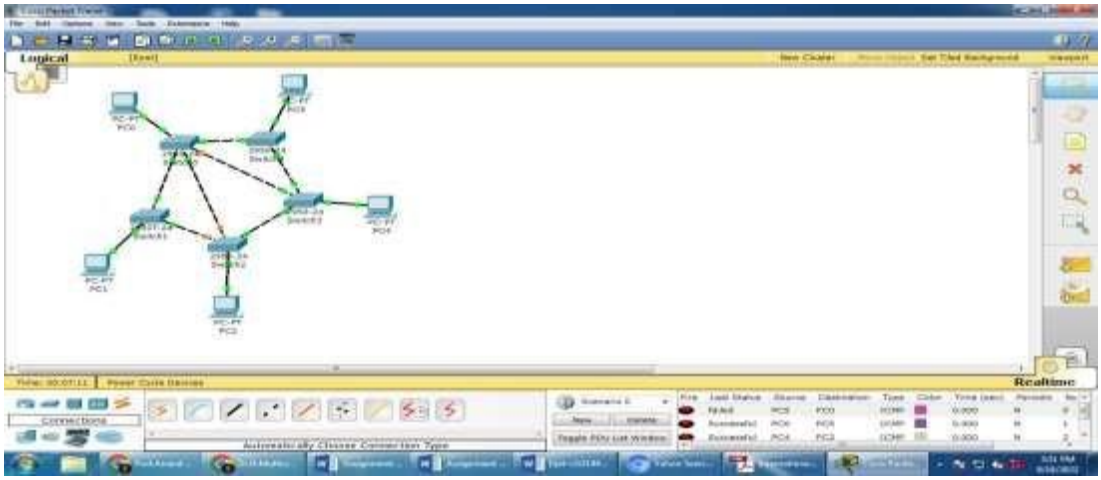


Fig -3: Design of mesh topology

Configuration of component:

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

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

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

Simulation of network topology:

1. **Simulation of bus topology:**

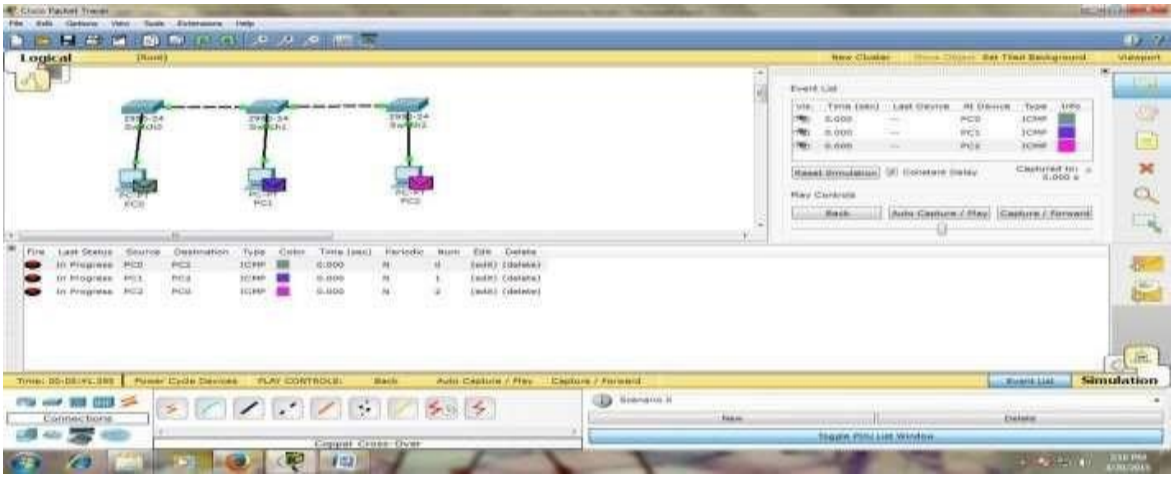


Fig.4 Simulation of Bus Topology

2. Simulation of star topology:

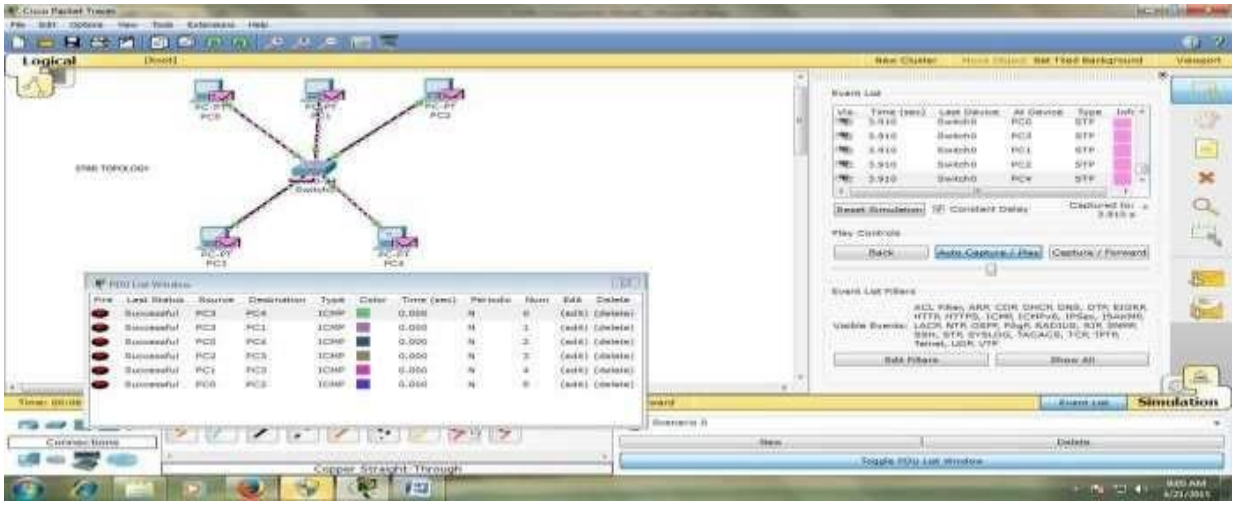


Fig.5 Simulation of Star Topology

3. Simulation of Mesh topology:

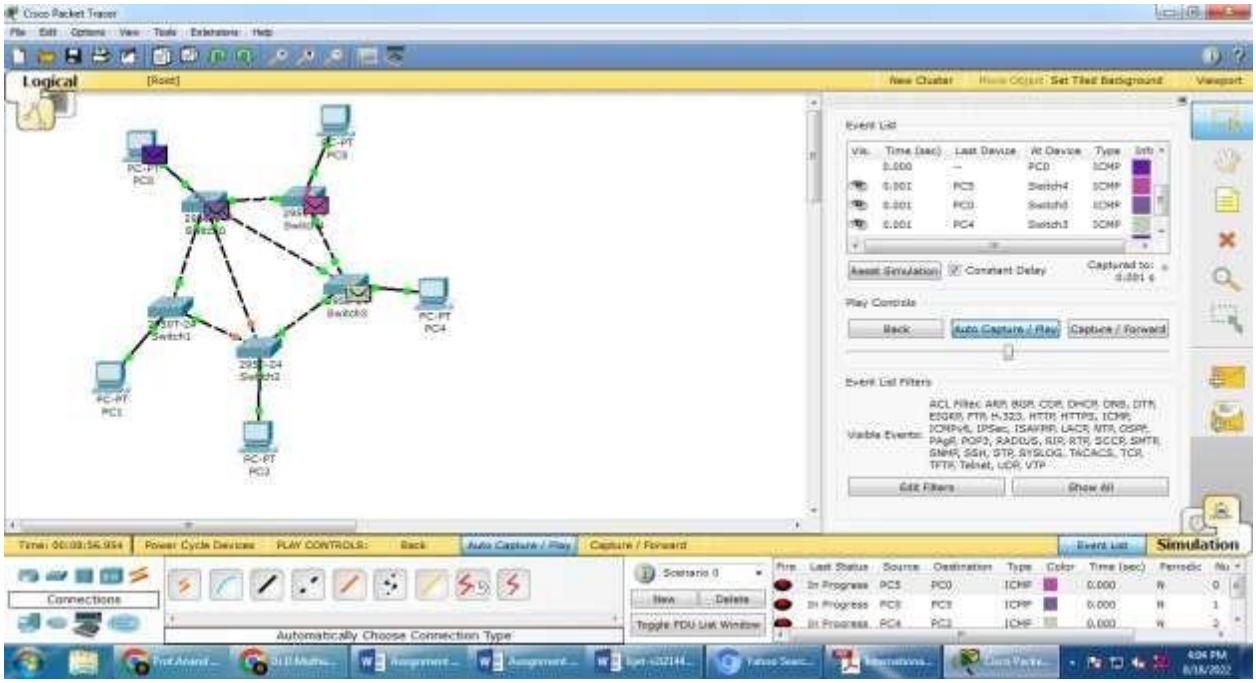


Fig.6 Simulation of Mesh Topology

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

Outcomes:

Demonstrate types of technologies and types of transmission media using Cisco Packet tracertool.

FAQs:

1) What is Cisco Packet Tracer?
2) What are different types of Topologies?
3) Difference between the Topologies?
4) What is Transmission Media?

4.3 Experiment 3

- **Aim:** Use packet Tracer tool for configuration of 3 router networkusing one of the following protocol RIP/OSPF/BGP

  - **Prerequisite**

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

  - **Learning Objectives**:

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

1.3 Theory:

1.3.1 Introduction

ROUTING INFORMATION PROTOCOL:

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

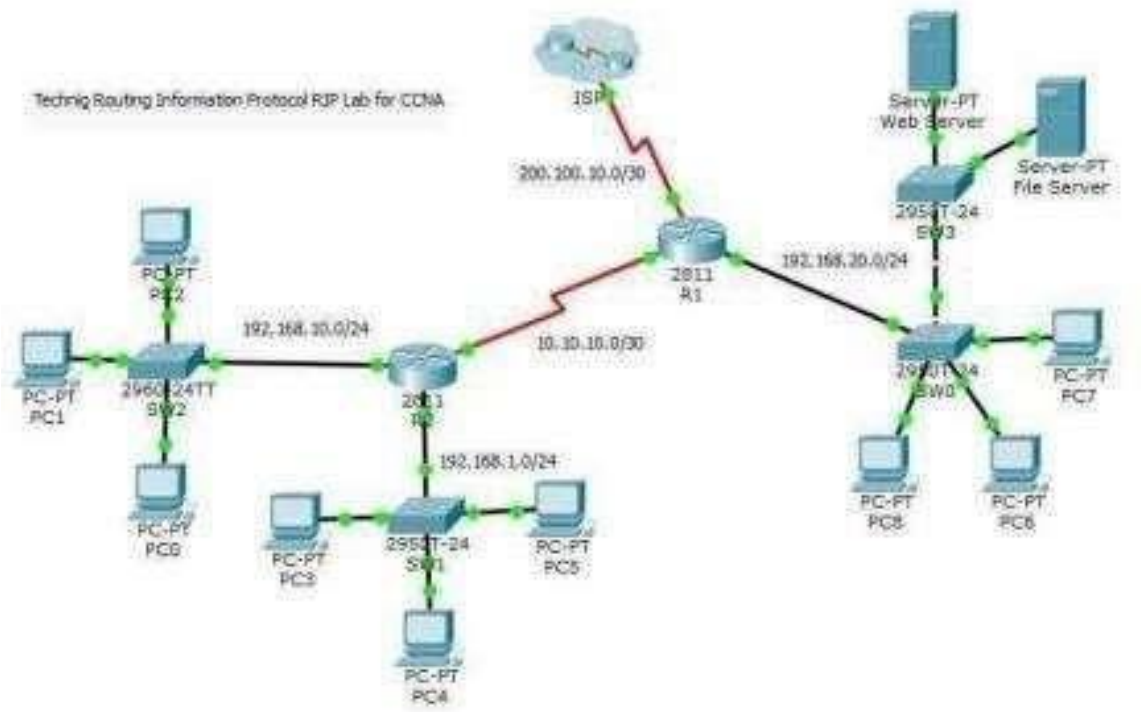
Originally, each RIP router transmitted full updates every 30 seconds. In the early deployments, routing tables were small enough that the traffic was not significant. As networks grew in size, however, it became evident there could be a massive traffic burst every 30 seconds, even if the routers had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

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

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

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

CONFIGURE ROUTING INFORMATION PROTOCOL (RIP)



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

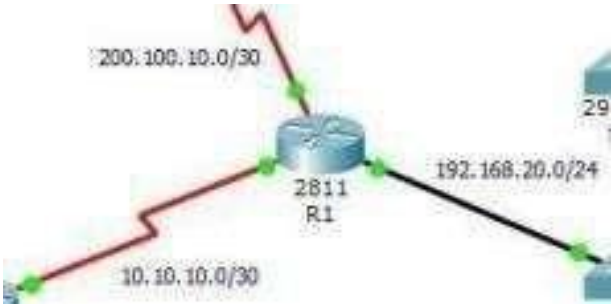
```
R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 200.100.10.0
```

DHCP message types:

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

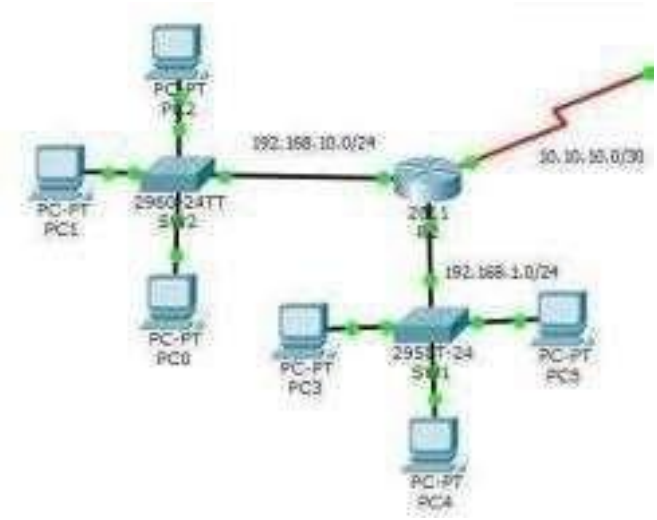
After enabling router with enable command then go to privileged mode with configure terminal command. Now with router rip command, enable routing for all routers. Theversion 2 Command, configure routing information protocol with version two. And next set

all network id like the above network command. I have set all three network which connectdirectly to R1.



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

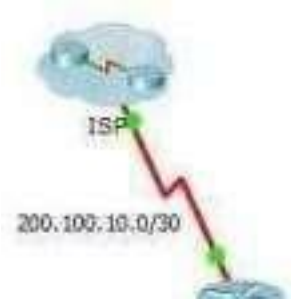
```
R2>enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 10.10.10.0
R2(config-router)#network 192.168.10.0
R2(config-router)#network 192.168.1.0
R2(config-router)#
```



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

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

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



- **CONFIGURE OSPF ROUTING PROTOCOL**

In the router R1 configure OSFP routing with Router ospf command

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

The router OSPF command is enable OSPF routing on the router, and the 1 before OSFP is the process ID of the OSFP Protocol. You can set different process id from "1-65535" for each router. The network command with network ID "network 20.10.10.0" is the network identifier,and the " 0.0.0.3″ is the wildcard mask of 20.10.10.0 network. Wildcard mask determine which interfaces to advertise, because OSPF advertise interfaces, not networks.

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

R3>**enable**

*R3#***configure terminal**

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

*R3(config)#***router ospf 1**

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

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

Don? So do the following for router R2.

R2>**enable**

*R2#***configure terminal**

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

*R2(config)#***router ospf 1**

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

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

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

Conclusion:

Hence we have studied Packet Tracer Properly.

Outcomes:

To Understand Simulation Tool.and should Able to Configure Routing Protocols.

FAQs:

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

4.3 Experiment no.3

Aim:

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

Theory:

Subnetting

Subnetting is a process of dividing large network into the smaller networks based on layer 3 IP address. Every computer on network has an IP address that represent its location on network. Two version of IP addresses are available IPv4 and IPv6. In this article we will perform subnetting on IPv4.

IPv4

IP addresses are displayed in dotted decimal notation, and appear as four numbers separated by dots. Each number of an IP address is made from eight individual bits known as octet. Each octet can create number value from 0 to 255. An IP address would be 32 bits long in binary divided into the two components, network component and host component. Network component is used to identify the network that the packet is intended for, and host component is used to identify the individual host on network.

IP addresses are broken into the two components:

Network component :- Defines network segment of device.

Host component :- Defines the specific device on a particular network segment

IP Classes in decimal notation

1. Class A addresses range from 1-126
2. Class B addresses range from 128-191
3. Class C addresses range from 192-223
4. Class D addresses range from 224-239
5. Class E addresses range from 240-254

- 0 [Zero] is reserved and represents all IP addresses.
- 127 is a reserved address and is used for testing, like a loop back on an interface.
- 255 is a reserved address and is used for broadcasting purposes

Subnet mask

Subnet mask is a 32 bits long address used to distinguish between network address and host address in IP address. Subnet mask is always used with IP address. Subnet mask has only one purpose, to identify which part of an IP address is network address and which part is hostaddress.

For example how will we figure out network partition and host partition from IP address 192.168.1.10 ? Here we need subnet mask to get details about network address and host address.

- In decimal notation subnet mask value 1 to 255 represent network address and value 0 [Zero] represent host address.
- In binary notation subnet mask **ON** bit [1] represent network address while **OFF** bit[0] represent host address.

In decimal notation

IP address	192.168.1.10
Subnet mask	255.255.255.0

Network address is **192.168.1** and host address is **10**.

In binary notation

IP address	11000000.10101000.00000001.00001010
Subnet mask	11111111.11111111.11111111.00000000

Network address is 11000000.10101000.00000001 and host address is 00001010

IP Class	**Default Subnet**	**Network bits**	**Host bits**	**Total hosts**	**Valid hosts**
A	255.0.0.0	First 8 bits	Last 24 bits	16, 777, 216	16, 777, 214
B	255.255.0.0	First 16 bits	Last 16 bits	65,536	65,534
C	255.255.255.0	First 24 bits	Last 8 bits	256	254

Network ID

First address of subnet is called network ID. This address is used to identify one segment or broadcast domain from all the other segments in the network.

Block Size

Block size is the size of subnet including network address, hosts addresses and broadcast address.

Broadcast ID

There are two types of broadcast, direct broadcast and full broadcast.

Direct broadcast or local broadcast is the last address of subnet and can be heard by all hosts in subnet.

Full broadcast is the last address of IP classes and can be heard by all IP hosts in network. Full broadcast address is 255.255.255.255

The main difference between direct broadcast and full broadcast is that routers will not propagate local broadcasts between segments, but they will propagate directed broadcasts.

Host Addresses

All address between the network address and the directed broadcast address is called host address for the subnet. You can assign host addresses to any IP devices such as PCs, servers, routers, and switches.

Single class C IP range can fulfill this requirement, still you have to purchase 2 class C IP range, one for each network. Single class C range provides 256 total addresses and we need only 30 addresses, this will waste 226 addresses. These unused addresses would make additional route advertisements slowing down the network.

With subnetting you only need to purchase single range of class C. You can configure router to take first 26 bits instead of default 24 bits as network bits. In this case we would extend default boundary of subnet mask and borrow 2 host bits to create networks. By taking two bits from the host range and counting them as network bits, we can create two new subnets, and assign hosts them. As long as the two new network bits match in the address, they belong to the same network. You can change either of the two bits, and you would be in a new subnet.

Advantage of Subnetting

- Subnetting breaks large network in smaller networks and smaller networks are easier to manage.
- Subnetting reduces network traffic by removing collision and broadcast traffic, that overall improve performance.
- Subnetting allows you to apply network security polices at the interconnection between subnets.
- Subnetting allows you to save money by reducing requirement for IP range.

Default subnet mask

Class	Subnet Mask	Format
A	255.0.0.0	Network.Host.Host.Host
B	255.255.0.0	Network.Network.Host.Host
C	255.255.255.0	Network.Network.Network.Host

Key terms to remembers

- A subnet is a smaller portion of large network treated as its own separate network. To create subnet we borrow bits from host portion and assign them as network bits. This mean more networks, fewer hosts.
- If the network bits on two addresses do not match, then the two packets are intended for two separate networks.
- On a 32 bits IP address at least eight bits must belong to the network portion and at least 2 bits must belong to the host portion.
- Each IP address has a predefined IP class and that cannot be changed.
- Each class has a predefined default subnet mask that tell us the octets, which are already part of the network portion, as well as how many bits we have available to work with.
- Whatever network class is it, we cannot change those bits that are already assigned.We cannot assign the network ID and the broadcast address to a host.
- Regardless how many bits are left in the host field, network ID and the broadcast address must be reserved.
- Subnet bits start at the left and go to the right, without skipping bits.

Conclusion:

Successfully implemented the subnetting and subnet mask program.

4.4 Experiment no.4

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

Objective: To understand working of Distance vector routing protocol.

Prerequisite:

1. Shortest path finding
2. Classification of routing Algorithm

Learning Objectives:

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

Theory:

Introduction:

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

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

Information kept by DV router -

- Each router has an ID

Associated with each link connected to a router,

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

Distance Vector Table Initialization -

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

Distance vector Algorithm:

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

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

Dx(y) = Estimate of least cost from x to y

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

Dx = [Dx(y): y ∈ N] = Node x maintains distance vector

Node x also maintains its neighbors' distance vectors
– For each neighbor v, x maintains Dv = [Dv(y): y ∈ N]

Distance Vector Routing:

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

Link State Routing:

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

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

Outcome: Understand working of Distance vector routing protocol.

FAQs:

1) What is Link State Algorithm?
2) What is Distance Vector Algorithm?
3) Difference Between Link State and distance vector algorithm?

4.5 Expirement no. 5

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

Prerequisite:

a) Socket Header b) Network Programming c) Ports

Objectives:

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

Theory:

1.3.1 Introduction

TCP:

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

T**he client server model**

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

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

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

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

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

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

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

Algorithm: Server Program

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

Algorithm: Client Program

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

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

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

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

TYPES OF SOCKETS

Socket Types

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

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

- Stream Sockets − Delivery in a networked environment is guaranteed. If you send through the stream socket three items "A, B, C", they will arrive in the same order − "A, B, C". These sockets use TCP (Transmission Control Protocol) for data transmission. If delivery is impossible, the sender receives an error indicator. Data records do not have any boundaries.
- Datagram Sockets − Delivery in a networked environment is not guaranteed. They're connectionless because you don't need to have an open connection as in Stream Sockets − you build a packet with the destination information and send it out. They use UDP (User Datagram Protocol).
- Raw Sockets − These provide users access to the underlying communication protocols, which support socket abstractions. These sockets are normally datagram oriented, though their exact characteristics are dependent on the interface provided by the protocol. Raw sockets are not intended for the general user; they have been provided mainly for those interested in developing new communication protocols, or for gaining access to some of the more cryptic facilities of an existing protocol.

Here is the description of the parameters −

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

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

SERVER SOCKET METHODS

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

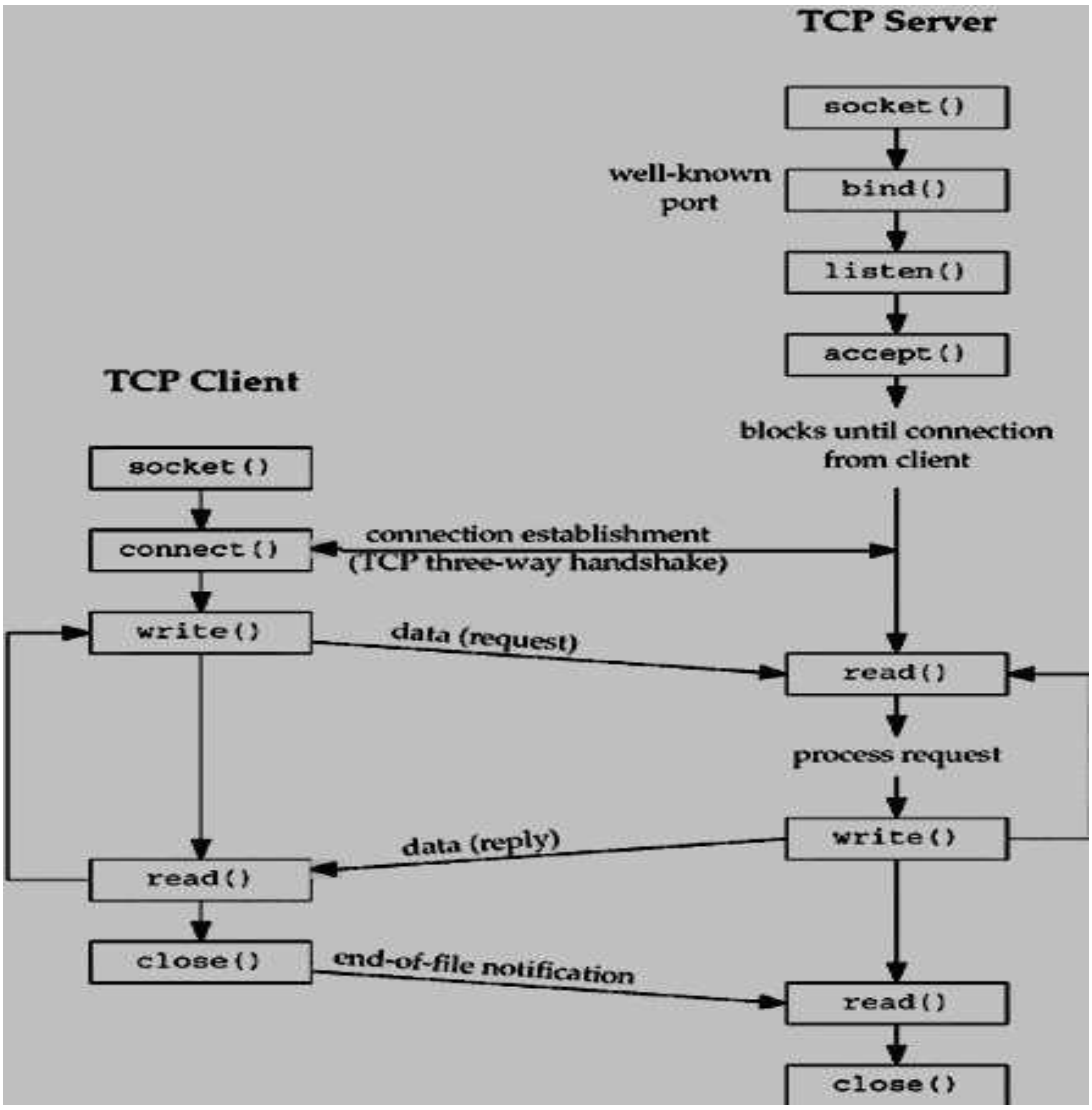
CLIENT SOCKET METHODS

Sr.No.	Method & Description
1	**s.connect()**This method actively initiates TCP server connection.

GENERAL SOCKET METHODS

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

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



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

Outcomes: Develop Work of Socket and Different methods associated with Client & Server Socket

FAQs:

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

3. Why is the port number required?

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

5.What are the Methods Associate with Server Socket ?

6. What are the methods associated with Client Socket ?

4.6. Experiment no. 6

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

1.1Prerequisite:

a) Socket Header b) Network Programming c) Ports

Learning Objectives:

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

Theory:

1.3.1 Introduction

What is UDP?

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

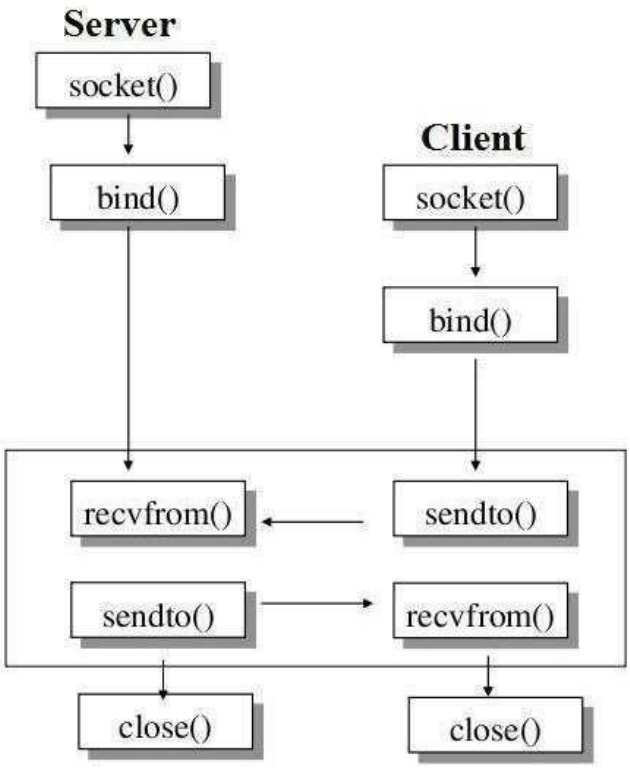
Why UDP is faster than TCP?

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

LINUX SOCKET PROGRAMMING:

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

Connectionless Protocol



The header files:

The Berkeley socket development library has many associated header files.They include: ***<sys/socket.h>***

Definitions for the most basic of socket structures with the BSD

socket API *<sys/socket.h>*

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

Socket API<sys/types.h>

Definitions for the socketaddr_in{} and other base data structures

<sys/un.h>

Definitions and data type declarations for SOCK_UNIX streams

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

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

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

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

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

Client: A simple demo to send an UDP packet containing "Hello World!" to address127.0.0.1, port 7654 might look like this:

#include #include #include #include #include#include int

main(int argc, char *argv[])

{

int sock; struct sockaddr_in sa;

int bytes_sent, buffer_length;

char buffer[200]; sprintf(buffer,

"Hello World!");

buffer_length = strlen(buffer) + 1;

sock = socket(PF_INET, SOCK_DGRAM, 0);

sa.sin_family = AF_INET;

sa.sin_addr.s_addr = htonl(0x7F000001);

sa.sin_port = htons(7654); bytes_sent = sendto(sock, buffer, buffer_length, 0, &sa,sizeof(struct sockaddr_in));

if(bytes_sent < 0) printf("Error sending packet: %s\n", strerror(errno));return 0;

}

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

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

Typical UDP Server code

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

APPLICATION :

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

Conclusion Thus we have successfully implemented the socket programming for TCP

Outcome: Thus we have studied Working of UDP Socket.

FAQs:

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

Conclusion Thus we have successfully implemented the socket programming for TCP

4.7. Experiment no. 7

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

Objectives:

To understand the concept of HTTP, HTTPS and FTP Protocol.

Theory

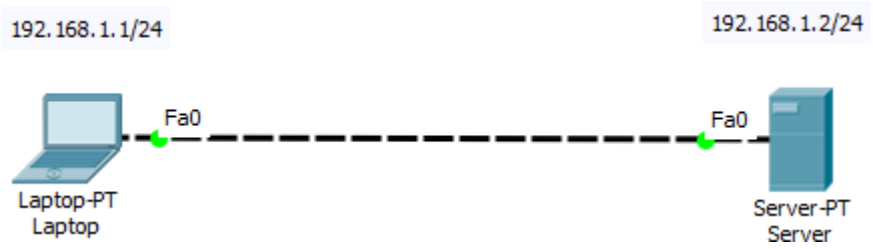
The File Transfer Protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

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

Worth noting: Although FTP does support user authentication, all data is sent in clear text, including usernames and passwords. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

Let's now do FTP configuration in Packet Tracer:

1. Build the network topology.



FTP topology.PNG

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

Laptop: IP address: 192.168.1.1 Subnet Mask: 255.255.255.0

Server: IP address: 192.168.1.2 Subnet Mask: 255.255.255.0

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

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

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

ftp from laptop.PNG

You are now in the FTP prompt .

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

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

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

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

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

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

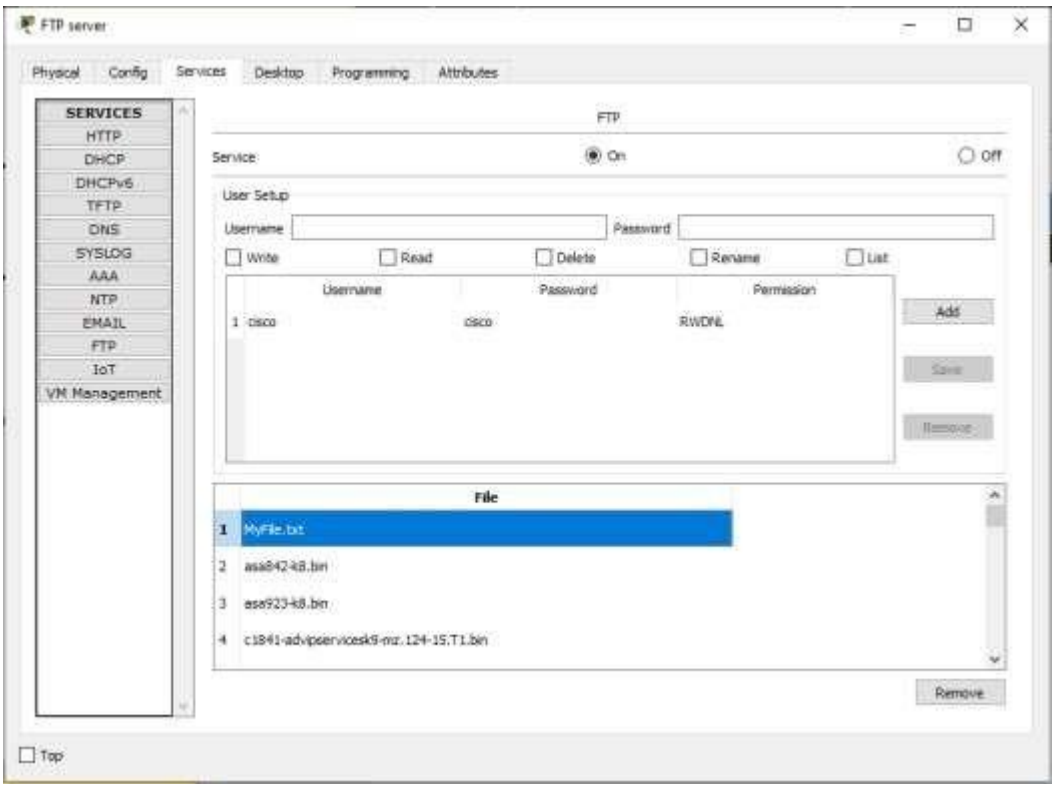
3. Now upload the file from the Laptop to the server using FTP. (An FTP connection has to be started first. But this is what we've done in step 3)

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

put MyFile.txt

put MyFile to FTP directory.PNG

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



MyFile.txt really send to sever.PNG

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

All FTP commands supported

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

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

For example: get MyFile.txt

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

For example: delete MyFile.txt

Rename– used to Rename a file

cd – used to change directory.

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

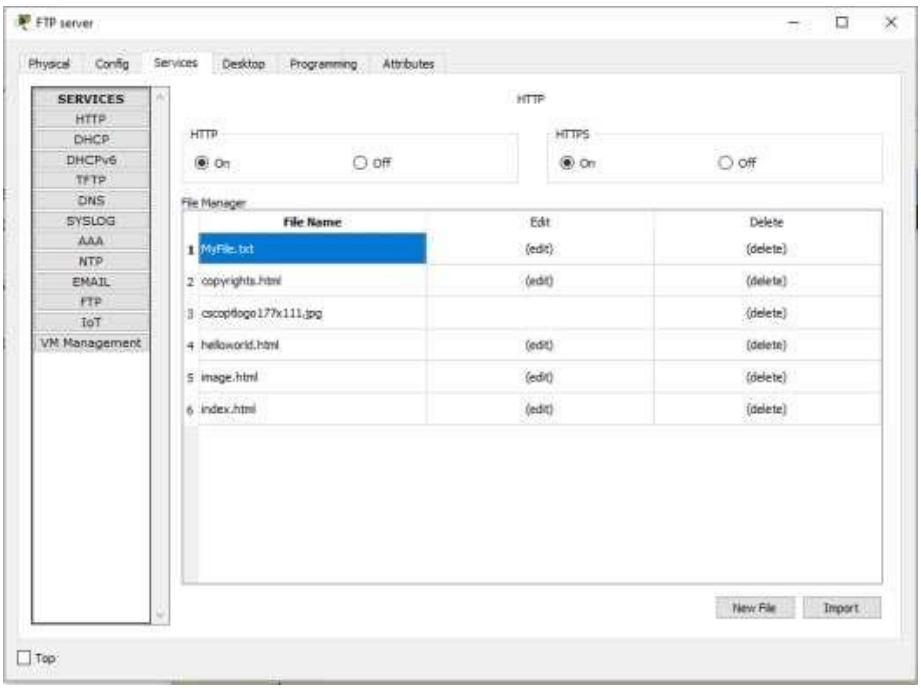
Once the http directory is open, you can upload a file to the HTTP server. You're now uploadinga file to an HTTP folder(directory) using FTP.

For example: put MyFile.txt

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

changing directory then put files to HTTP directory using FTP

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



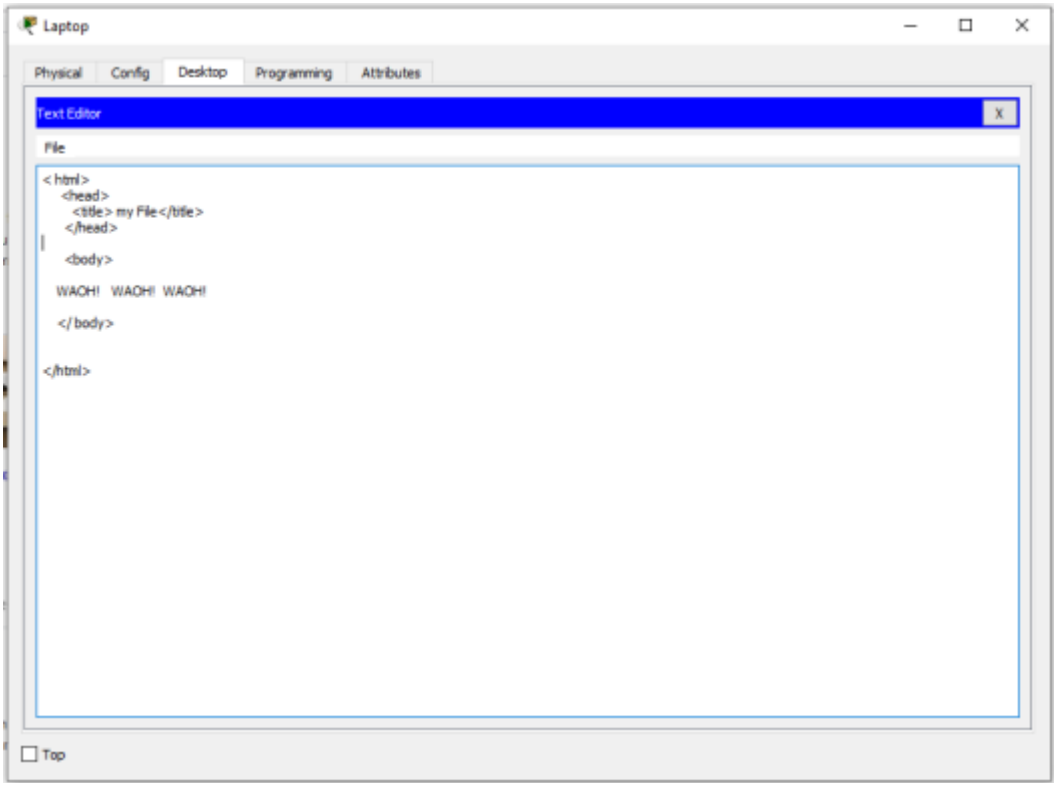
MyFile.txt really send to HTTP server

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

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

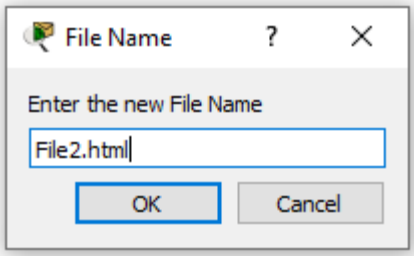
We'll create an html file in our Laptop, upload it to HTTP server directory using FTP, then try to access the file from the Laptop's browser.

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



File2 HTML code

Save your file as an html file like this:



File2 html.PNG

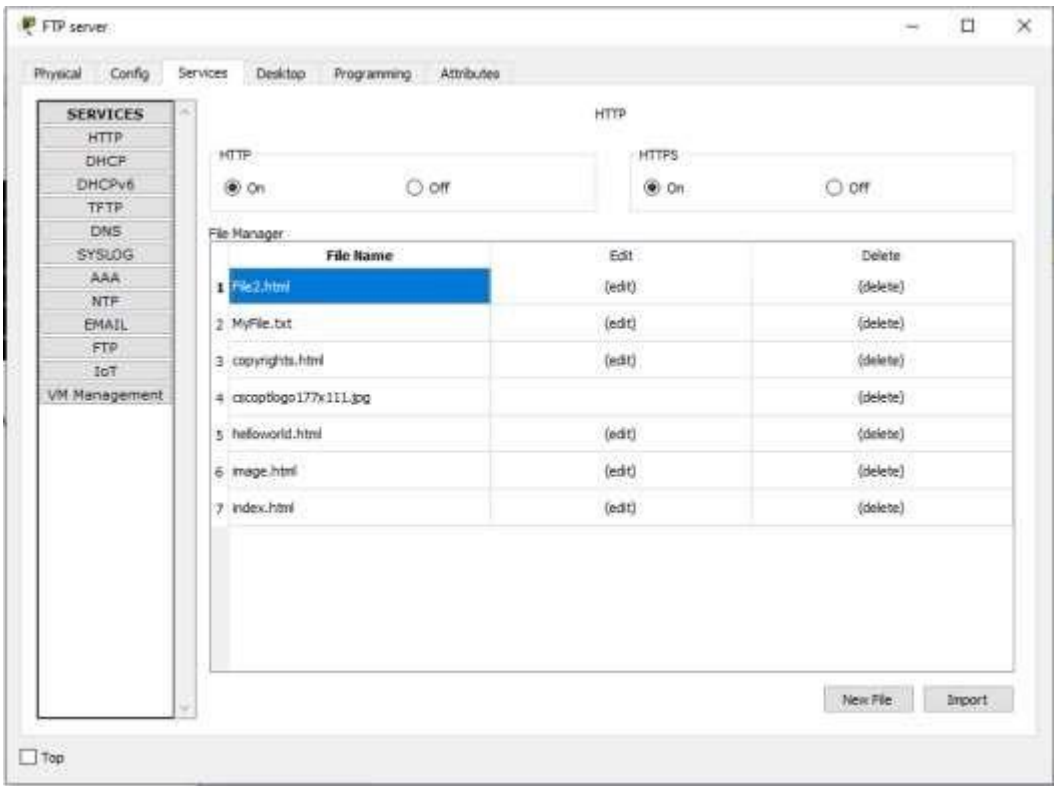
Now upload the file(File2.html) to the HTTP server using FTP. This is easy. We've already done it previously!

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

Sending File2. html to HTTP directory.PNG

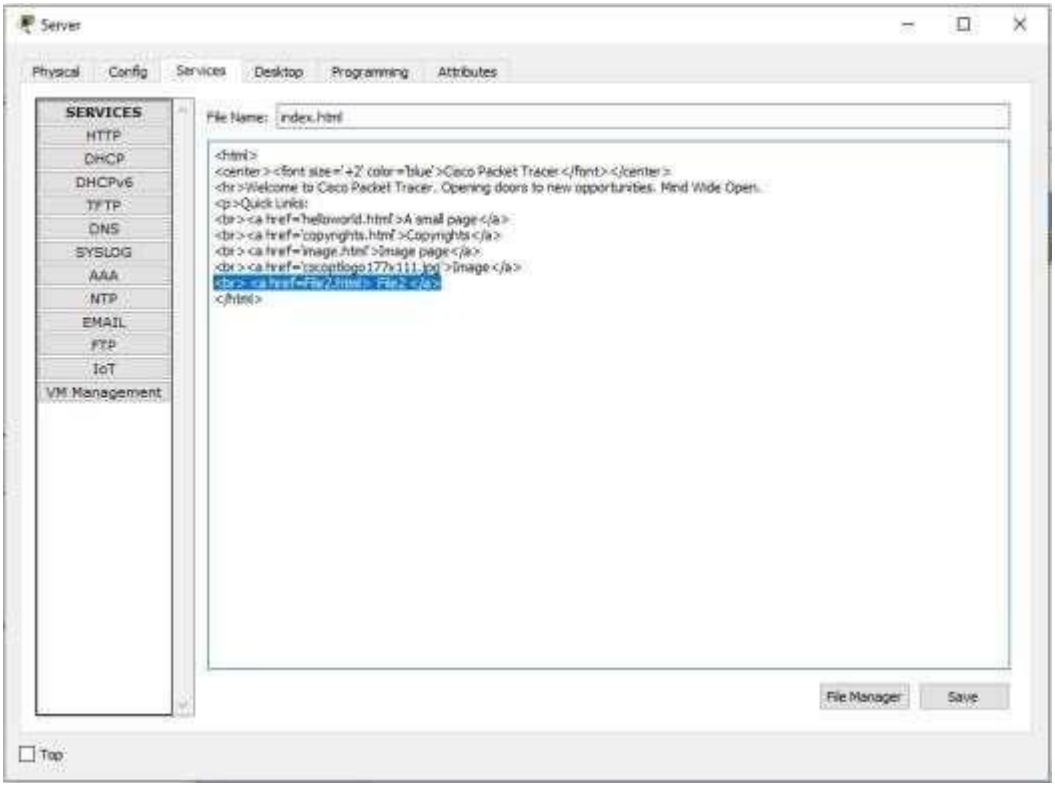
Check whether the html file uploaded has been received in the HTTP directory:Go

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



File2 HTML really uploaded into HTTP directory.PNG

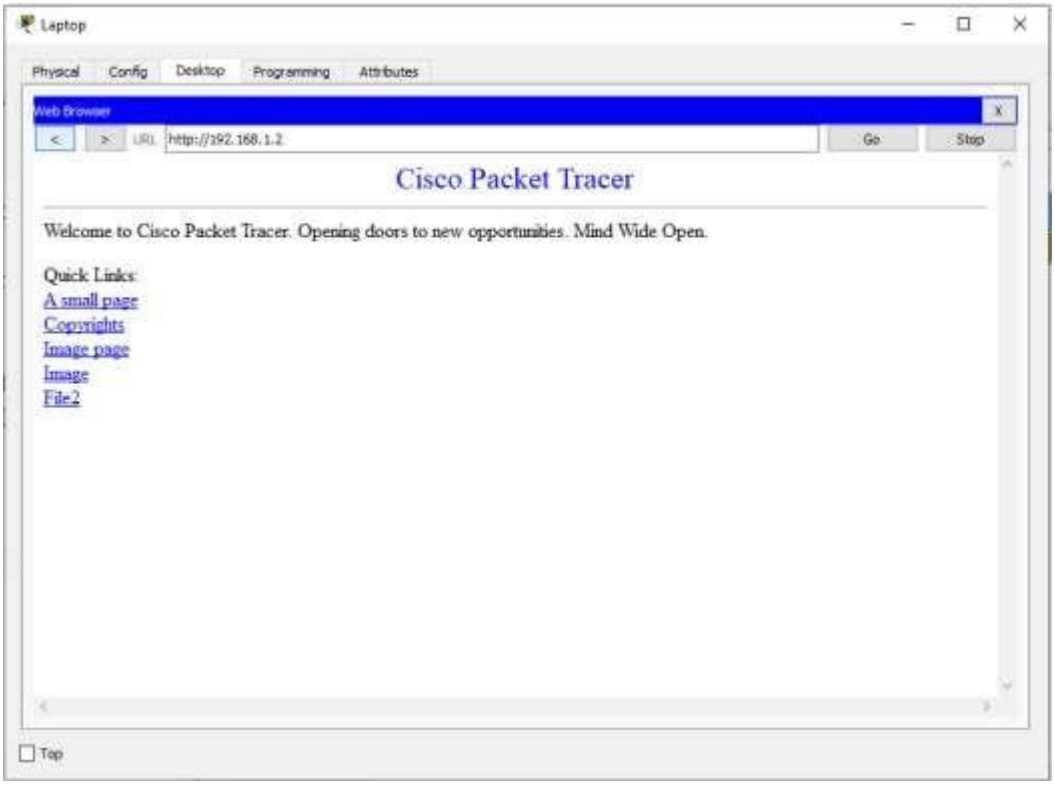
Now edit index.html file in the HTTP directory so as to include a link to File2 that we've just uploaded. This will make File2 accessible from the Laptop's browser. To do this, locate index.html then click edit. Proceed to edit it as shown below. Then save and accept overwrite.Index.html editing to include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



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

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

Http response with File2.PNG



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

Conclusion:

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

Outcomes: Analyze the performance of HTTP,HTTPS and FTP protocol using Packet tracertool.

FAQs:

1. Can multiple clients connect to same FTP socket?
2. Which socket is used in FTP?
3. Difference between HTTP and HTTPs?
4. How do you specify socket type for HTTP?

4.8.Experiment no.08

Title:

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

Theory:

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

How does SSL/TLS work?

In order to provide a high degree of privacy, SSL encrypts data that is transmitted across the web. This means that anyone who tries to intercept this data will only see a garbled mix of characters that is nearly impossible to decrypt.

SSL initiates an authentication process called a handshake between two communicating devices to ensure that both devices are really who they claim to be.

SSL also digitally signs data in order to provide data integrity, verifying that the data is not tampered with before reaching its intended recipient.

There have been several iterations of SSL, each more secure than the last. In 1999 SSL was updated to become TLS.

Why is SSL/TLS important?

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

SSL was created to correct this problem and protect user privacy. By encrypting any

data that goes between a user and a web server, SSL ensures that anyone who intercepts the data can only see a scrambled mess of characters. The consumer's credit card number is now safe, only visible to the shopping website where they entered it.

SSL also stops certain kinds of cyber attacks: It authenticates web servers, which is important because attackers will often try to set up fake websites to trick users and steal data. It also prevents attackers from tampering with data in transit, like a tamper-proof seal on a medicine container.

Are SSL and TLS the same thing?

SSL is the direct predecessor of another protocol called TLS (Transport Layer Security). In 1999 the Internet Engineering Task Force (IETF) proposed an update to SSL. Since this update was being developed by the IETF and Netscape was no longer involved, the name was changed to TLS. The differences between the final version of SSL (3.0) and the first version of TLS are not drastic; the name change was applied to signify the change in ownership.

Since they are so closely related, the two terms are often used interchangeably and confused. Some people still use SSL to refer to TLS, others use the term "SSL/TLS encryption" because SSL still has so much name recognition.

SSL PROTOCOL Wireshark :

Step 1: Open a Trace

1. Open the Wireshark trace

You should see the following trace.

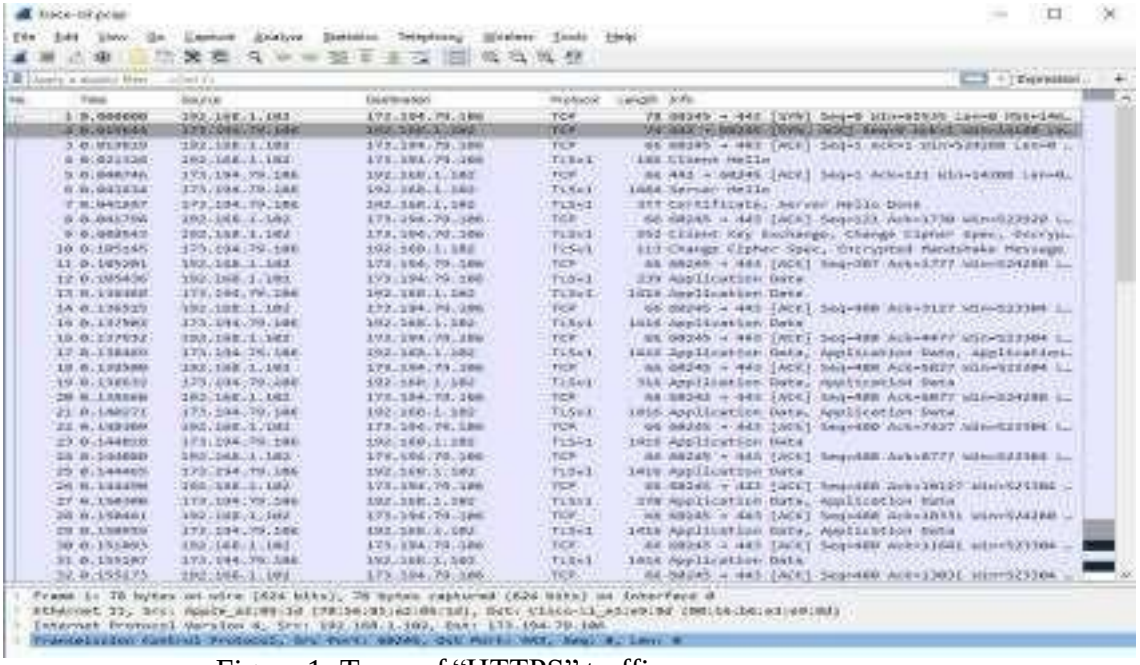


Figure 1: Trace of "HTTPS" traffic

1

Step 2: Inspect the Trace

Now we are ready to look at the details of some "SSL" messages.

*2. To begin, enter and apply a display filter of "*ssl*". (see below)*

This filter will help to simplify the display by showing only SSL and TLS messages. It will exclude other TCP segments that are part of the trace, such as Acks and connection open/close.

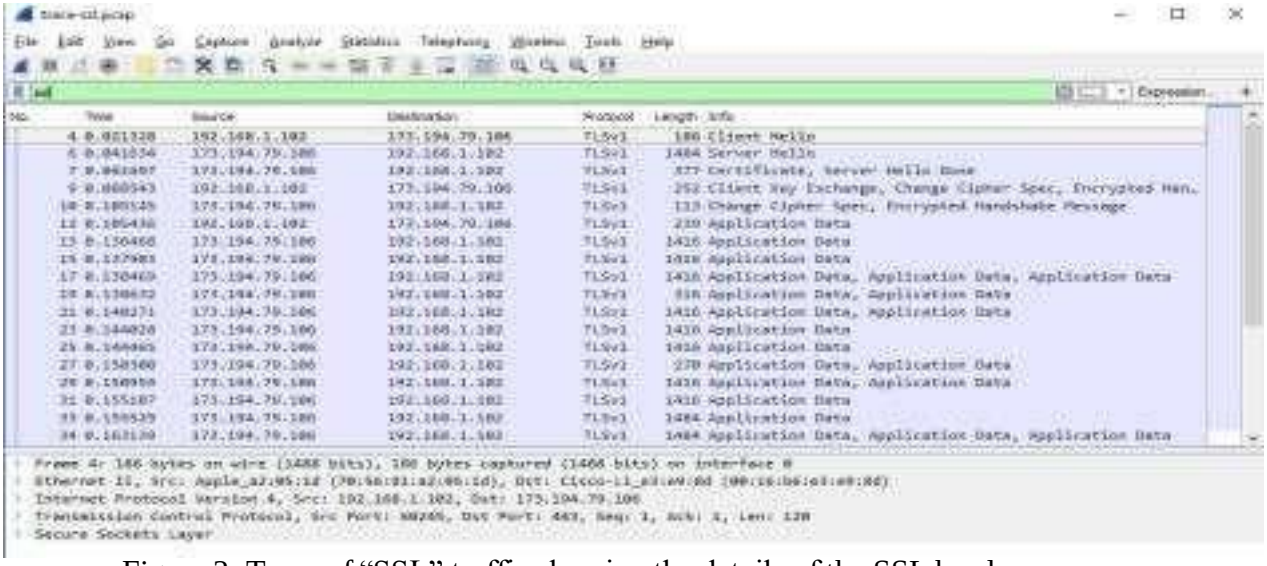
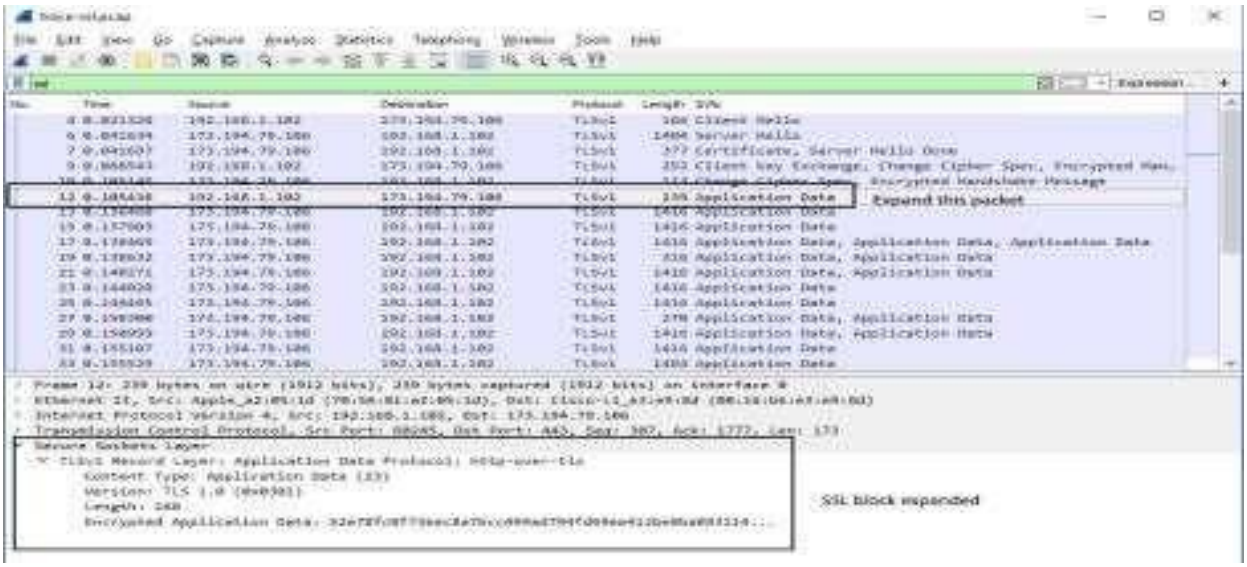


Figure 2: Trace of "SSL" traffic showing the details of the SSL header

3. *Select a TLS message somewhere in the middle of your trace for which the Info reads "Applica tion Data" & expand its Secure Sockets Layer block (by using the "+" expander or icon).* For in stance, packet #12 (see below).



2

Application Data is a generic TLS message carrying contents for the application, such as the web page. It is a good place for us to start looking at TLS messages.

The lower layer protocol blocks are TCP and IP because SSL runs on top of TCP/IP. The SSL layer contains a "TLS Record Layer". This is the foundational sublayer for

TLS. All messages contain records. Expand this block to see its details. Each record starts with a Content Type field. This tells us what is in the contents of the record. Then comes a Version identifier. It will be a constant value for the SSL connection. It isfol lowed by a Length field giving the length of the record. Last comes the contents of the record. Applica tion Data records are sent after SSL has secured the connection, so the contents will show up as encrypted data. To see within this block, we could configure Wireshark with the decryption key. This is possible, but outside of our scope.Note that, unlike other protocols we will see such as DNS, there may be multiple records in a single message. Each record will show up as its own block. Look at the Info column, and you will see messages with more than one block.

The Content-Type for a record containing "Application Data" is 23. The version constant used in this trace is 0x0301 which represents TLS 1.0. The Length coversonly the payload of the Record Layer.

Step 3: The SSL Handshake

An important part of SSL is the initial handshake that establishes a secure connection. The handshake proceeds in several phases. There are slight differences for different versions of TLS and depending on the encryption scheme that is in use. The usual outline for a brand-new connection is:

a. Client (the browser) and Server (the web server) both send their Hellos
b. Server sends its certificate to Client to authenticate (and optionally asks for Client Certificate) c. Client sends keying information and signals a switch to encrypted data.
d. Server signals a switch to encrypted data.
e. Both Client and Server send encrypted data.
f. An Alert is used to tell the other party that the connection is closing.

Note that there is also a mechanism to resume sessions for repeat connections betweenthe same client and server to skip most of steps b and c. However, we will not study session resumption.

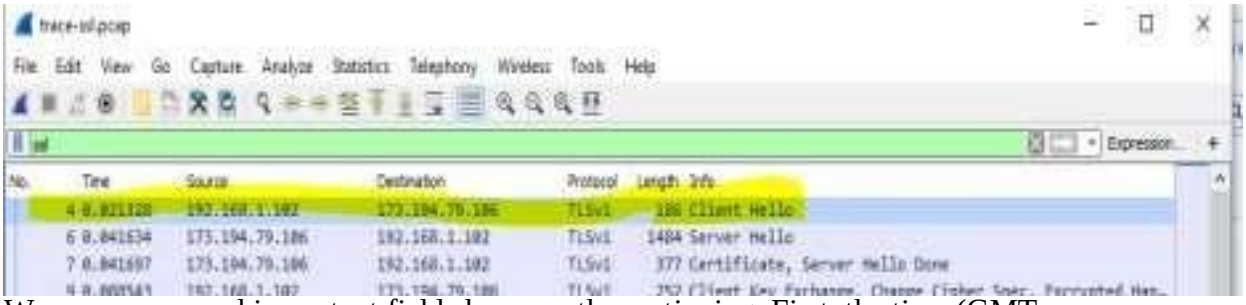
Hello Messages

*Next we will find and inspect the details of the Client Hello and Server Hello messages, including expand ing the Handshake protocol block within the TLS Record.*For these initial messages, an encryption scheme is not yet established so the contentsof the record are visible to us. They contain details of the secure connection setup in a

Handshake protocol format.

3

4. Select packet #4, which is a TLS Client Hello message

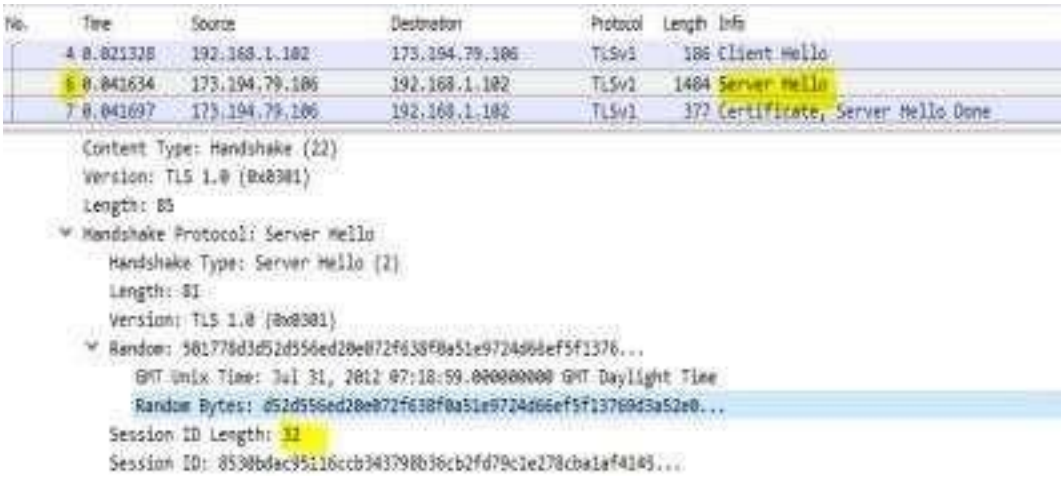


We can see several important fields here worth mentioning. First, the time (GMT seconds since midnight Jan 1, 1970) and random bytes (size 28) are included. This willbe used later in the protocol to generate our symmetric encryption key. The client can send an optional session ID to quickly resume a previous TLS connection and skip portions of the TLS handshake. Arguably the most important part of the Cli entHello message is the list of cipher suites, which dictate the key exchange algorithm, bulk encryption algorithm (with key length), MAC, and a psuedo-random function. The list should be ordered by client preference. The collection of these choices is a “cipher suite”, and the server is responsible for choosing a secure one it supports or return an error if it doesn’t support any. The final field specified in the specification is for compression methods. However, secure clients will advertise that they do not support compression (by passing “null” as the only algorithm) to avoid the CRIME attack.

Finally, the ClientHello can have a number of different extensions. A common one is server_name, which specifies the host name the connection is meant for, so webservershosting multiple sites can present the correct certifi cate.

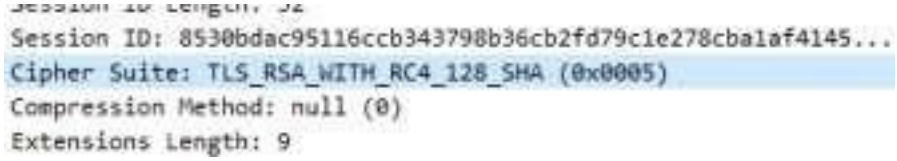
5. Select packet #6, which is a TLS Server Hello message

The session ID sent by the server is 32 bytes long. This identifier allows later resumption of the session with an abbreviated handshake when both the client and server indicate the same value. In our case, the client likely sent no session ID as there was nothing to resume (see below)



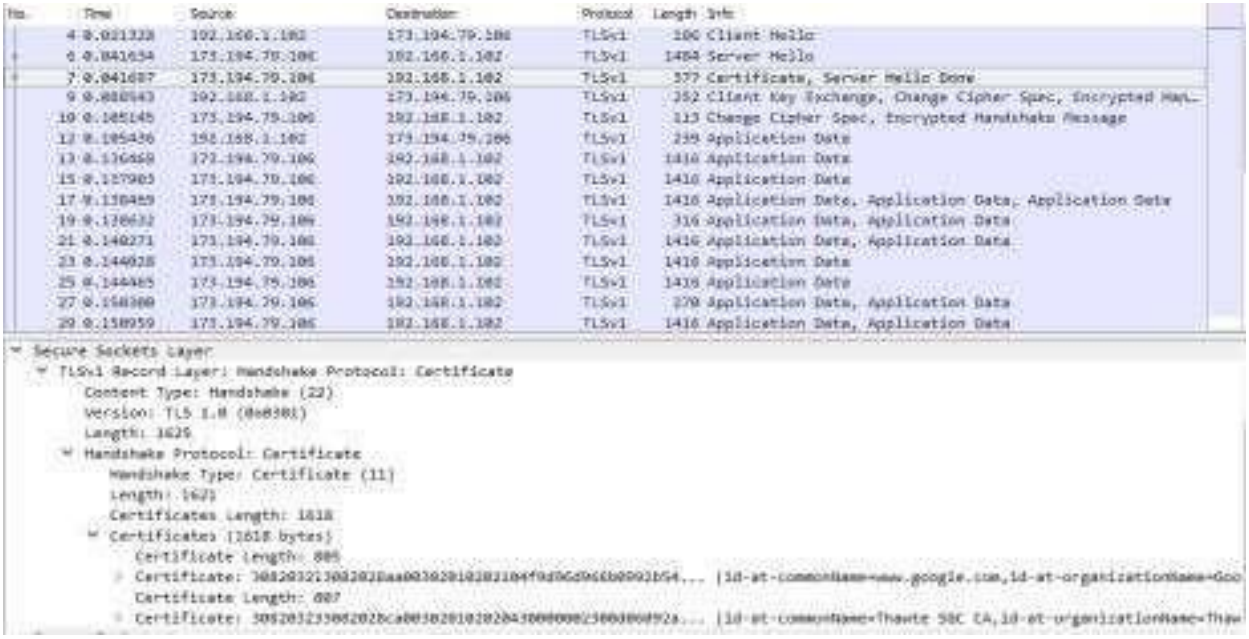
4

The Cipher method chosen by the Server is *TLS_RSA_WITH_RC4_128_SHA (0x0005)*. The Client will list the different cipher methods it supports, and the Server will pick one of these methods to use.



Certificate Messages

6. *Next, find and inspect the details of the Certificate message including expanding the Handshake protocol block within the TLS Record (see below for expansion of packet #7).*



As with the Hellos, the contents of the Certificate message are visible because an encryption scheme is not yet established. It should come after the Hello messages.

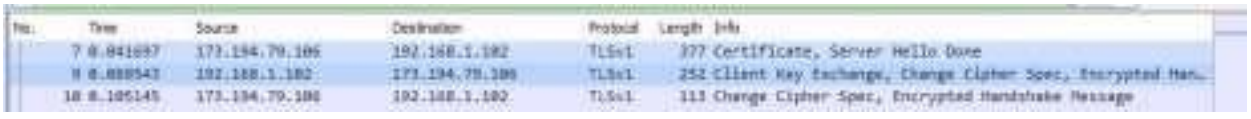
Note it is the server that sends a certificate to the client, since it is the browser that wants to verify the identity of the server. It is also possible for the server to request certificates from the client, but this be havior is not normally used by web applications.

A Certificate message will contain one or more certificates, as needed for one party to verify the identity of the other party from its roots of trust certificates. You can inspect those certificates in your browser.

5

Client Key Exchange and Change Cipher Messages

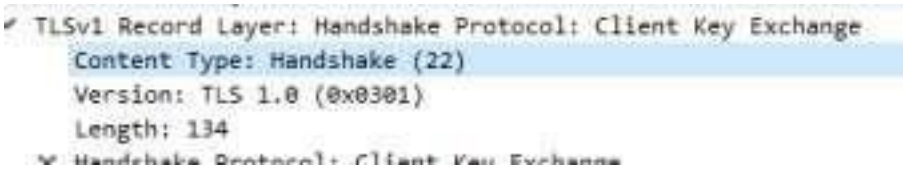
7. Find and inspect the details of the Client Key Exchange and Change Cipher messages i.e. packet #9 (see below)



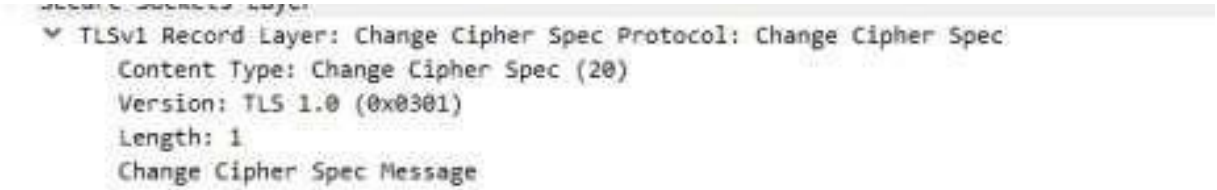
The key exchange message is sent to pass keying information so that both sides will have the same se cret session key. The change cipher message signal a switch to a new encryption scheme to the other party. This means that it is the last unencrypted

message sent by the party.

Note how the Client Key Exchange has a Content-Type of 22, indicating the Handshake protocol. This is the same as for the Hello and Certificate messages, asthey are part of the Handshake protocol.



The Change Cipher Spec message has a Content-Type of 20, indicating the Change Cipher Spec protocol (see packet #10 – see below).



That is, this message is part of its own protocol and not the Handshake protocol. Both sides send the Change Cipher Spec message immediately before they switch to sending encrypted contents. The message is an indication to the other side. The contents of the Change Cipher Spec mes sage are simply the value 1 as a single byte. Actually, it is the value "1" encrypted under the current scheme, which uses no encryption for the handshake so that we can see it.

Alert Message

8. Finally, find and inspect the details of an Alert message at the end of the trace (packet #42).

The Alert message is sent to signal a condition, such as notification that one party is closing the connection. You should find an Alert after the Application Data messages that make up the secure web fetch.

Note, the Content-Type value is 21 for Alert. This is a new protocol, different from the Handshake, Change Cipher Spec and Application Data values that we have already seen.

The alert is encrypted; we cannot see its contents. Wireshark also describes the message as an “Encrypted Alert”. Presumably is it a “close_notify” alert to signal thatthe connection is ending, but we can not be certain.

Conclusion:

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

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

FAQs:

1) What is SSL Protocol?
2) On which layer SSL Protocol works?

4.9. Experiment no.09

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

Learning Objectives:

1. Understand the concept and working of Encrypted mails

Theory

MIME (Secure/Multipurpose Internet Mail Extensions)

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

How does S/MIME work?

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

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

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

Benefits of encrypted email

1. Safeguards sensitive data

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

2. Economical

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

3. Timesaving

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

4. Regulation compliance

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

5. Protects against malware

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

How does email encryption work?

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

There are generally three encryption types available:

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

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

How to send encrypted email in Outlook

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

1. Create a digital certificate

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

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

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

2. Use your digital signature

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

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

3. Encrypt Outlook messages

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

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

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

4. Encrypt all Outlook messages

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

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

How to Send Encrypted Email

Have you ever wondered about the security of your private email conversations? Whether at work, school, or home, sending emails comes with a bit of a risk. There's one thing you can do to

discourage data breaches and attacks on your sensitive data, however. Use encrypted email. Learn how to practice this common-sense method for communicating in our step-by-step guide. But first, let's look at why you should embrace encryption for your email correspondence.

How to Encrypt Email and Send Secure Messages

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

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

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

Outcome:

Demonstrate the steps for implementation of S/MIME email security throughMicrosoft® Office Outlook.

FAQs:

1) What is MIME?
2) How MIME works?
3) How does email encryption work?

4.10 Experiment No. 10

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

.

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.

What Ports Does IPSEC Operate On?

UDP port 500 should be opened as should IP protocols 50 and 51. UDP port 500 should be opened to allow for ISAKMP to be forwarded through the firewall while protocols 50 and 51 allow ESP and AH traffic to be forwarded respectively.2

What is ISAKMP?

ISAKMP stands for Internet Security Association and Key Management Protocol. These are two key components of an IPSEC VPN that must be in place in order for it to function normally and protect the public traffic that is being forwarded between the client and VPN server or VPN server to VPN server.

What are ESP and AH?

No, ESP is not Extra-Sensory Perception! ESP stands for Encapsulating Security Protocol and AH stands for Authentication Header.

Encapsulating Security Protocol

ESP gives protection to upper layer new protocols, with a Signed area indicating where a protected data packet has been signed for integrity, and an Encrypted area which

indicates the information that's protected with confidentiality. Unless a data packet is being tunneled, ESP protects only the IP data payload (hence the name), and not the IP header.

ESP may be used to ensure confidentiality, the authentication of data origins, connectionless integrity, some degree of traffic-level confidentiality, and an anti-replay service (a form of partial sequence integrity which guards against the use of commands or credentials which have been captured through password sniffing or similar attacks).3

Authentication Header

Authentication Header (AH) is a new protocol and part of the Internet Protocol Security (IPsec) protocol suite, which authenticates the origin of IP packets (datagrams) and guarantees the integrity of the data. The AH confirms the originating source of a packet and ensures that its contents (both the header and payload) have not been changed since transmission.

If security associations have been established, AH can be optionally configured to defend against replay attacks using the sliding window technique.4

How Do They All Work Together?

When properly configured, an IPSEC VPN provides multiple layers of security that ensure the security mode and integrity of the data that is being transmitted through the encrypted tunnel. This way an organization can feel confident that the data has not beenintercepted and altered in transit and that they can rely on what they are seeing.

IPsec Protocols

AH and/or ESP are the two protocols that we use to actually protect user data. Both of them can be used in transport or tunnel mode, let's walk through all the possible options.

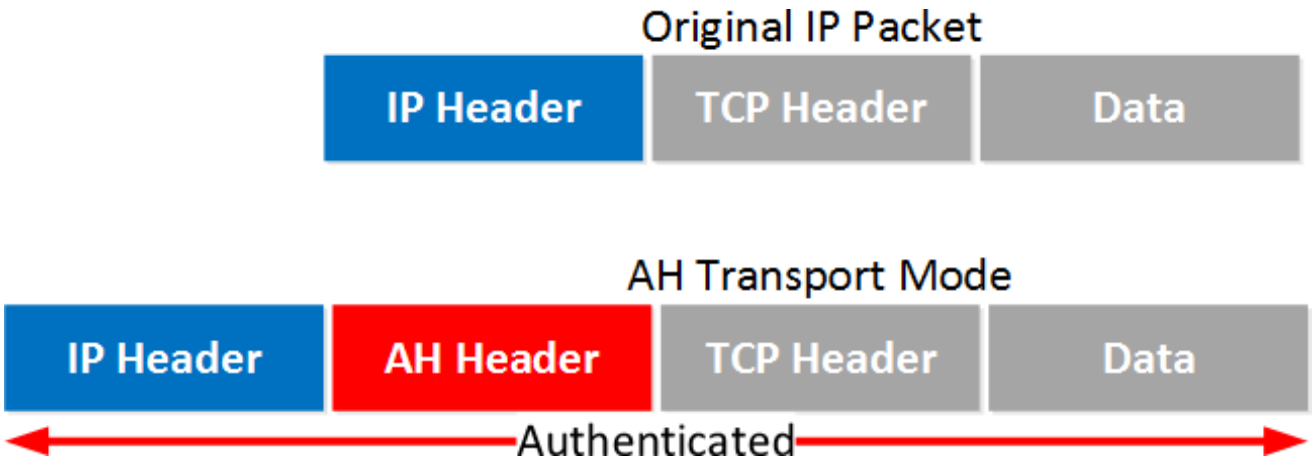
Authentication Header Protocol

AH offers authentication and integrity but it doesn't offer any encryption. It protects the IP packet by calculating a hash value over almost all fields in the IP header. The fields it excludes are the ones that can be changed in transit (TTL and header checksum). Let's

start with transport mode…

Transport Mode

Transport mode is simple, it just adds an AH header after the IP header. Here's an example of an IP packet that carries some TCP traffic:



And here's what that looks like in Wireshark:

```
Frame 1: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
Ethernet II, Src: Cisco_8b:36:d0 (00:1d:a1:8b:36:d0), Dst: Cisco_ed:7a:f0 (00:17:5a:ed:7a:f0)
Internet Protocol Version 4, Src: 192.168.12.1 (192.168.12.1), Dst: 192.168.12.2 (192.168.12.2)
    Version: 4
    Header Length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 124
    Identification: 0x0028 (40)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 255
    Protocol: Authentication Header (51)
  ⊞ Header checksum: 0x21d3 [validation disabled]
    Source: 192.168.12.1 (192.168.12.1)
    Destination: 192.168.12.2 (192.168.12.2)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Authentication Header
    Next Header: ICMP (0x01)
    Length: 24
    AH SPI: 0xcf54ccdf
    AH Sequence: 30
    AH ICV: aa9cafe5ed06d6c74cb3c671
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x7994 [correct]
    Identifier (BE): 8 (0x0008)
    Identifier (LE): 2048 (0x0800)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
    [Response frame: 2]
  ⊞ Data (72 bytes)
```

Above you can see the AH header in between the IP header and ICMP header. This is acapture I took of a ping between two routers. You can see that AH uses 5 fields:

Next Header: this identifies the next protocol, ICMP in our example.

Length: this is the length of the AH header.

SPI (Security Parameters Index): this is an 32-bit identifier so the receiver knows towhich flow this packet belongs.

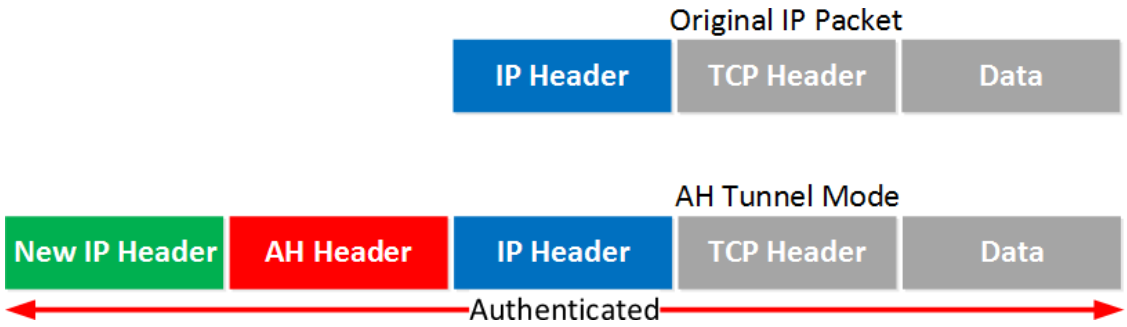
Sequence: this is the sequence number that helps against replay attacks.

ICV (Integrity Check Value): this is the calculated hash for the entire packet. The receiver also calculates a hash, when it's not the same you know something is wrong.

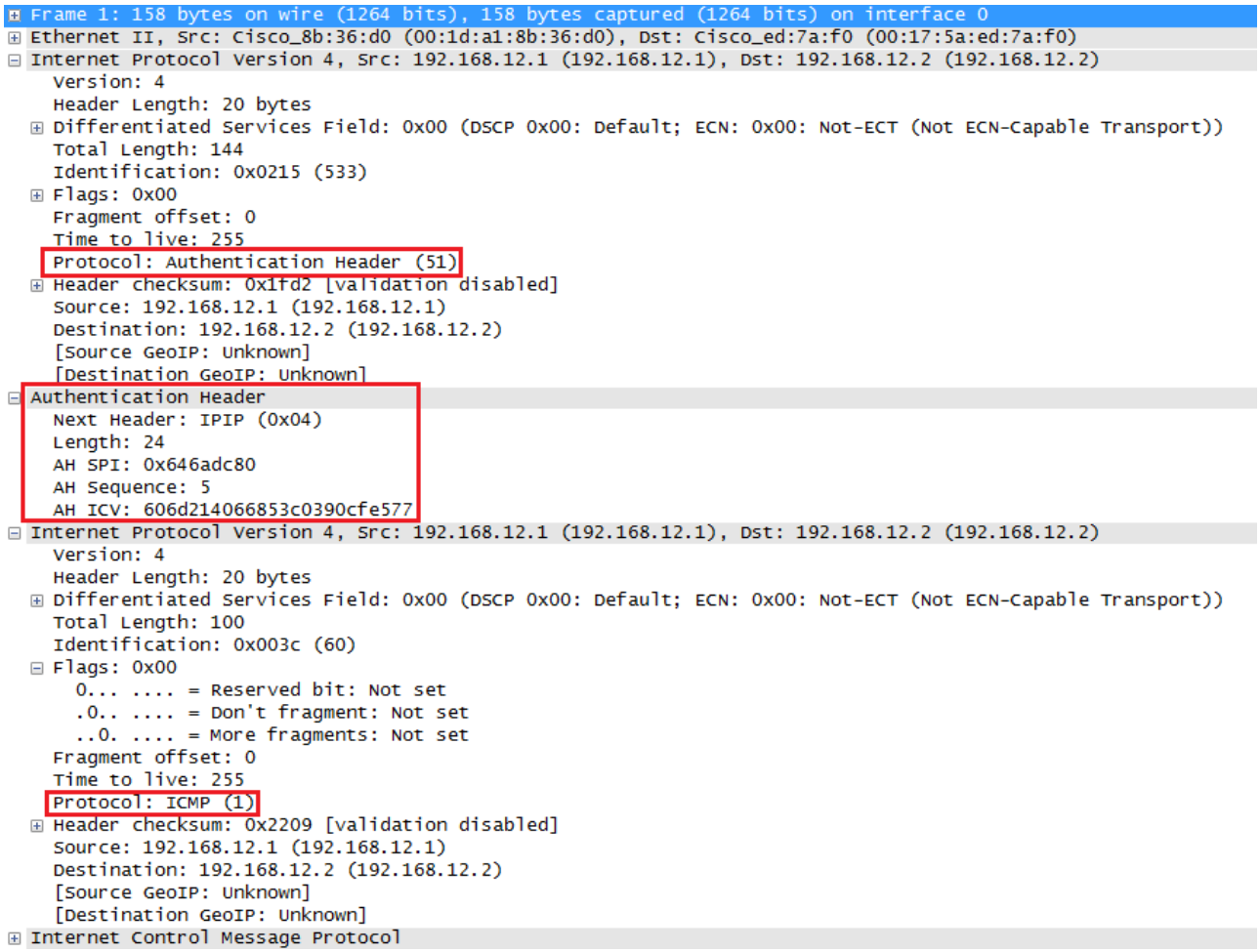
Let's continue with tunnel mode.

Tunnel Mode

With tunnel mode we add a new IP header on top of the original IP packet. This could be useful when you are using private IP addresses and you need to tunnel your traffic over the Internet. It's possible with AH but it doesn't offer encryption:



The entire IP packet will be authenticated. Here's what it looks like in wireshark:



Above you can see the new IP header, then the AH header and finally the original IP packet that carries some ICMP traffic.

One problem with AH is that it doesn't play well with NAT / PAT. Fields in the IP header like TTL and the checksum are excluded by AH because it knows these will change. The IP

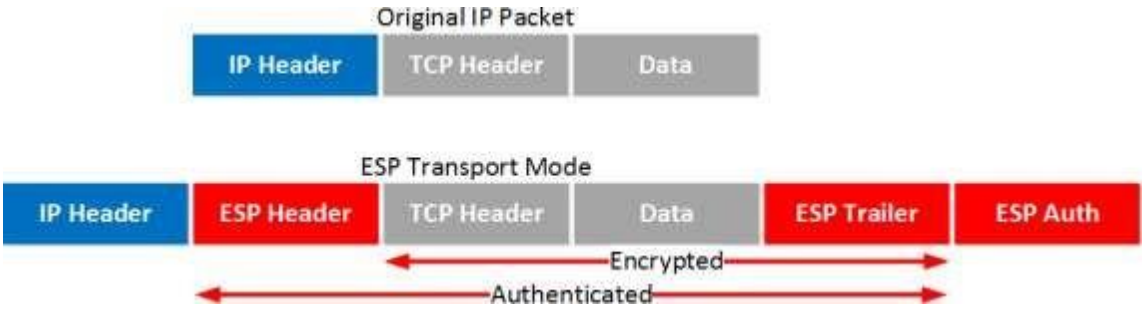
Let's continue with ESP…

ESP (Encapsulating Security Payload) Protocol

ESP is the more popular choice of the two since it allows you to encrypt IP traffic. Wecan use it in transport or tunnel mode, let's look at both.

Transport Mode

When we use transport mode, we use the original IP header and insert an ESP header. Here's what it looks like:



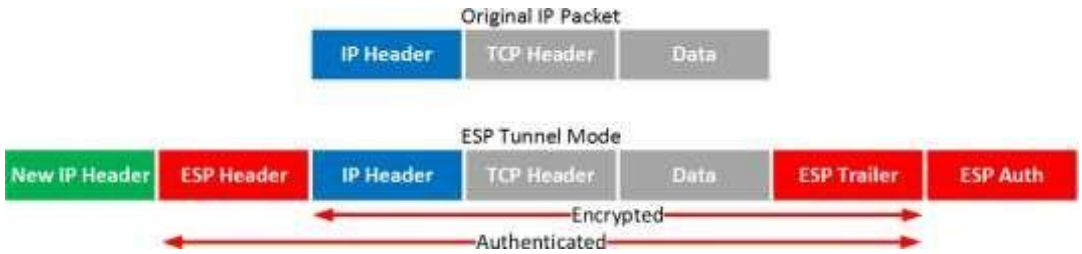
Above you can see that we add an ESP header and trailer. Our transport layer (TCP for example) and payload will be encrypted. It also offers authentication but unlike AH, it's not for the entire IP packet. Here's what it looks like in wireshark:

```
Frame 1: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface 0
Ethernet II, Src: Cisco_8b:36:d0 (00:1d:a1:8b:36:d0), Dst: Cisco_ed:7a:f0 (00:17:5a:ed:7a:f0)
Internet Protocol Version 4, Src: 192.168.12.1 (192.168.12.1), Dst: 192.168.12.2 (192.168.12.2)
   Version: 4
   Header Length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
   Total Length: 152
   Identification: 0x0042 (66)
   Flags: 0x00
     0... .... = Reserved bit: Not set
     .0.. .... = Don't fragment: Not set
     ..0. .... = More fragments: Not set
   Fragment offset: 0
   Time to live: 255
   Protocol: Encap Security Payload (50)
   Header checksum: 0x219e [validation disabled]
   Source: 192.168.12.1 (192.168.12.1)
   Destination: 192.168.12.2 (192.168.12.2)
   [Source GeoIP: Unknown]
   [Destination GeoIP: Unknown]
Encapsulating Security Payload
   ESP SPI: 0x36cb42df (919290591)
   ESP Sequence: 1
```

Above you can see the original IP packet and that we are using ESP. The IP header is incleartext but everything else is encrypted.

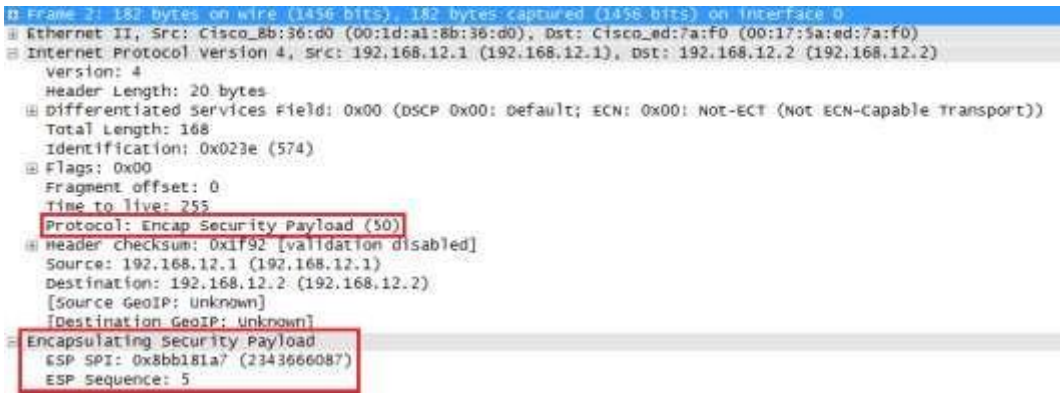
Tunnel Mode

How about ESP in tunnel mode? This is where we use a new IP header which is usefulfor site-to-site VPNs:



It's similar to transport mode but we add a new header. The original IP header is now also encrypted.

Here's what it looks like in wireshark:



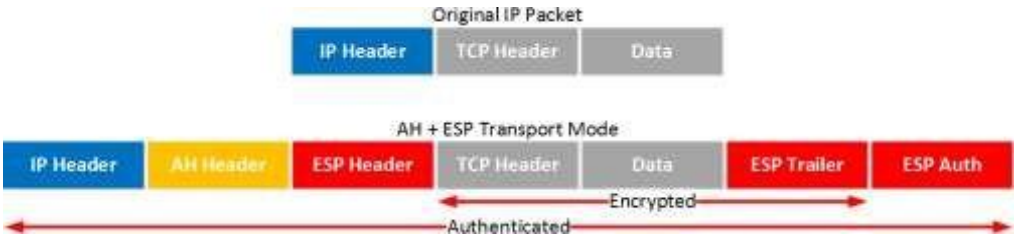
The output of the capture is above is similar to what you have seen in transport mode. The only difference is that this is a new IP header, you don't get to see the original IP header.

AH and ESP

This one confuses a lot of people, it's possible to use AH and ESP at the same time. Let's check it out!

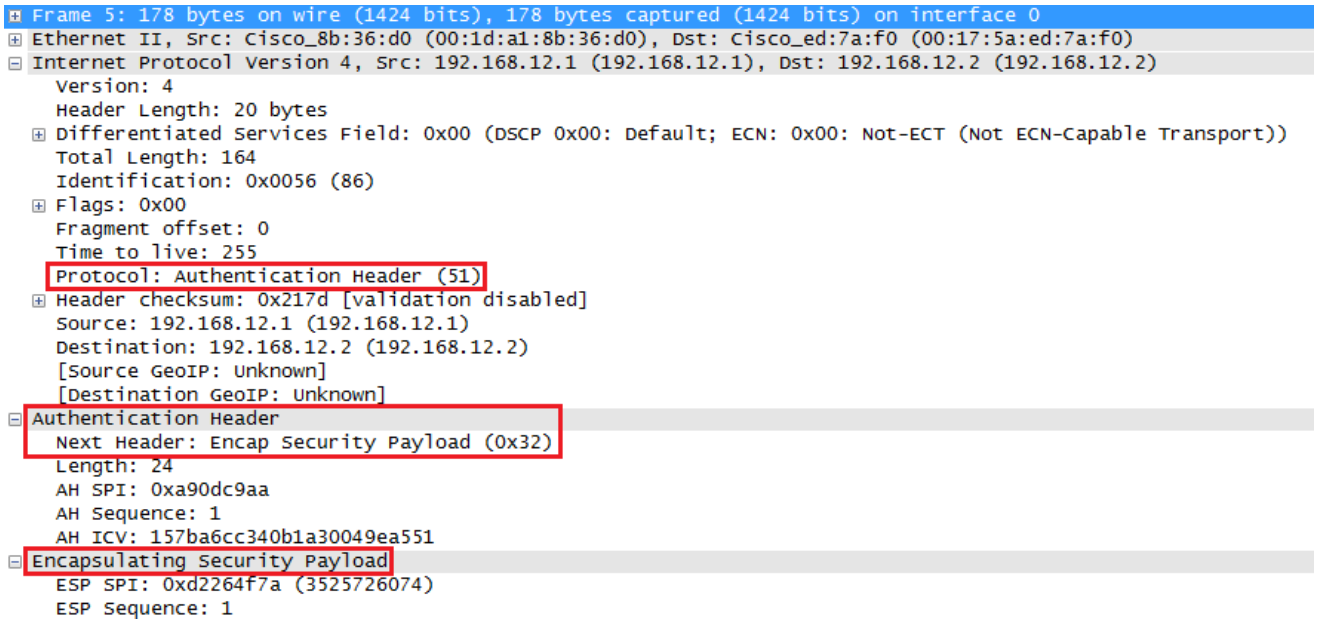
Transport Mode

Let's start with transport mode, here's what the IP packet will look like:



With transport mode we will use the original IP header, followed by an AH and ESP header. The transport layer, payload and ESP trailer will be encrypted.

Because we also use AH, the entire IP packet is authenticated. Here's what it looks likein wireshark:



Above you can see the original IP packet, the AH header and the ESP header.

Conclusion:

Hence we had studied the IPsec (ESP and AH) protocol by capturing the packets using Wireshark tool.

Outcomes:
Retrieve IPsec (ESP and AH) protocol by capturing the packets using Wireshark tool.

FAQs:
1) What is Wireshark?
2) Why Wireshark is used?
3) How the Packets are Captured in Wireshark?

4.11. Experiment no.11

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

Objectives:

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

New Concepts:

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

Theory:

Need for DNS:

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

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

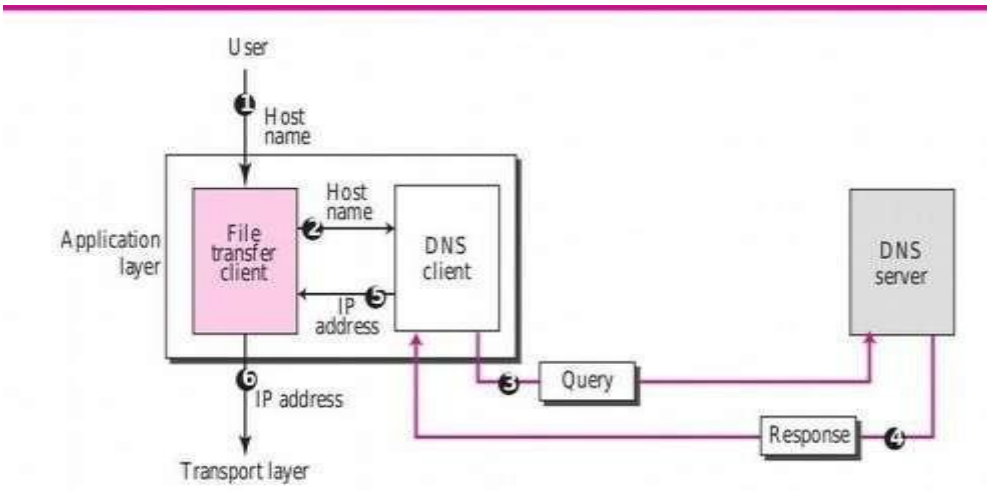


Figure 1: How TCP/IP uses a DNS client and a DNS server to map a name to an address; the reverse mapping is similar.

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

The following six steps map the host name to an IP address.

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

NAME SPACE:

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

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

Flat Name Space:

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section; if they do, it

has no meaning. The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

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

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

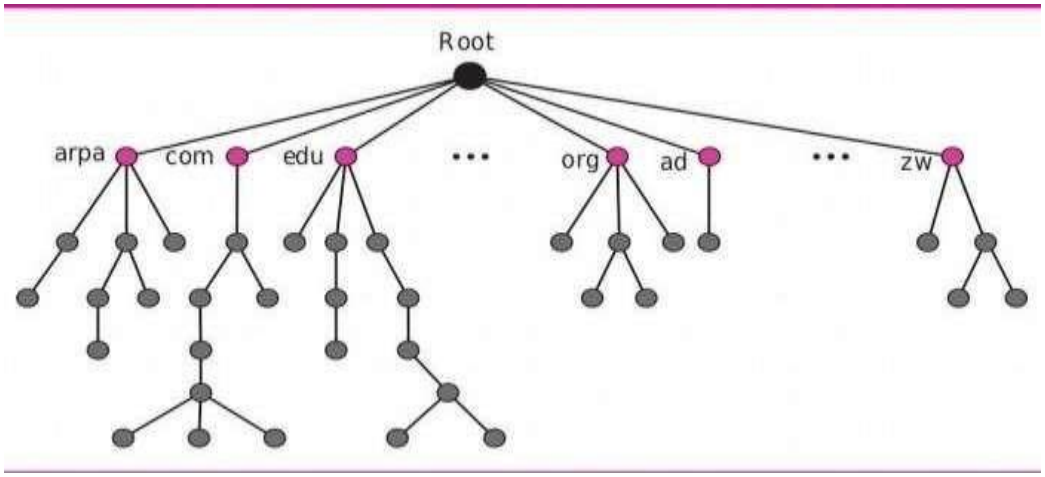


Figure 2: Domain Name Space

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

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

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

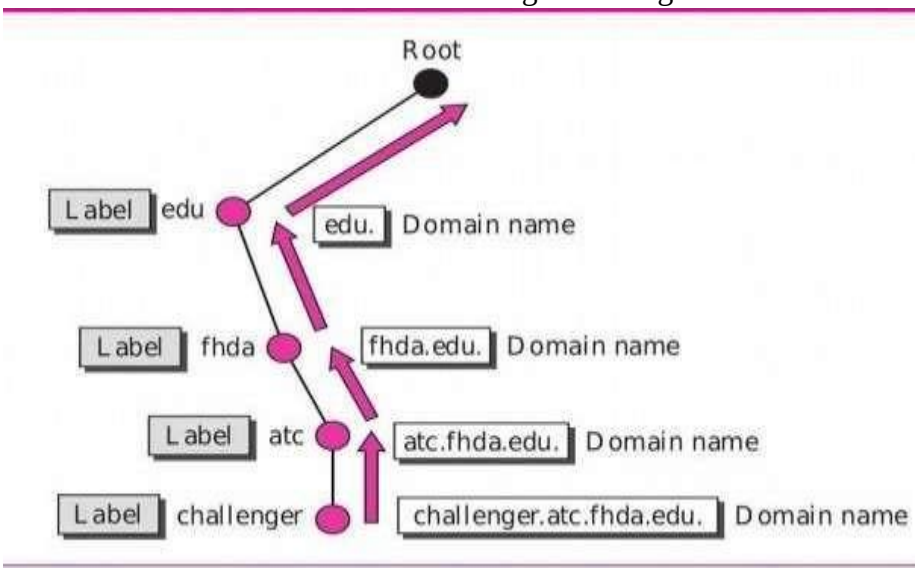


Figure 3 shows some domain names.

Domain:

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

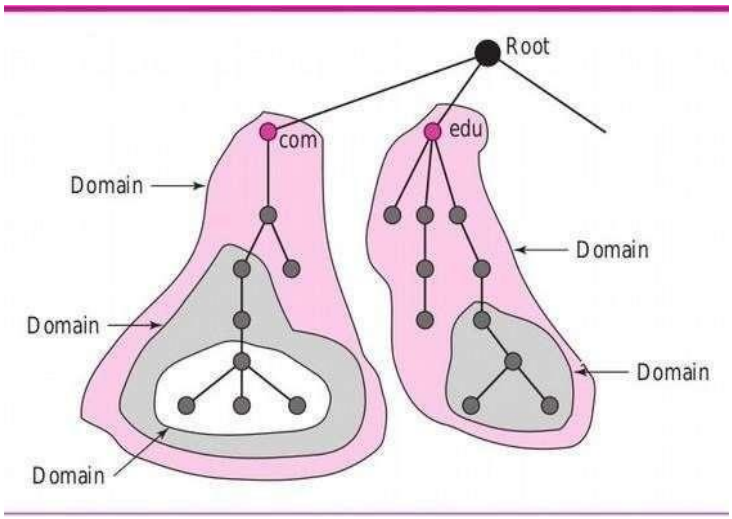


Figure 4 Domain

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

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

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

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

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

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

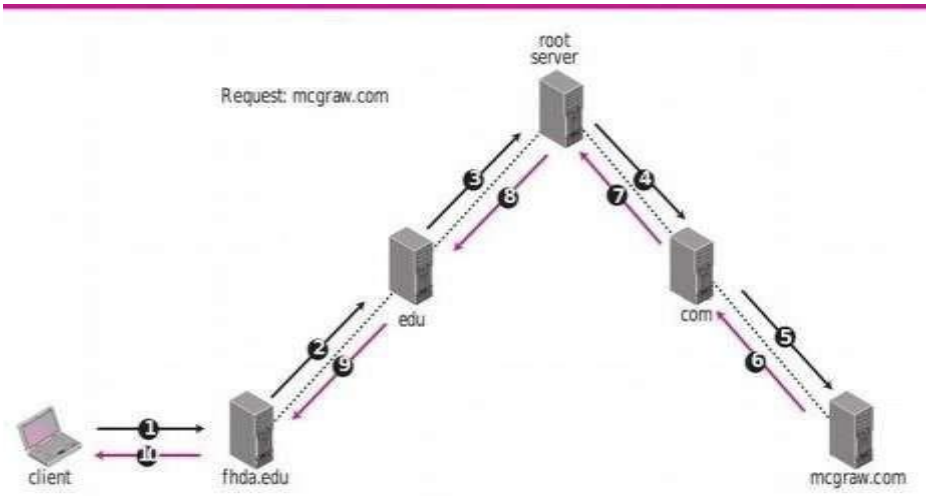


Figure 5: Recursive resolution.

Algorithm:

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

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

Conclusion:

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

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

FAQs:

1) What is DNS?
2) What is Need of DNS?
3) What is Domain?
4) What are different types of domain?

4.12. Experiment no. 12

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

Prerequisite:

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

Learning Objectives:

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

Theory

Introduction

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

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

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

Advantages of DHCP:-

DHCP has several major advantages over manual configurations.

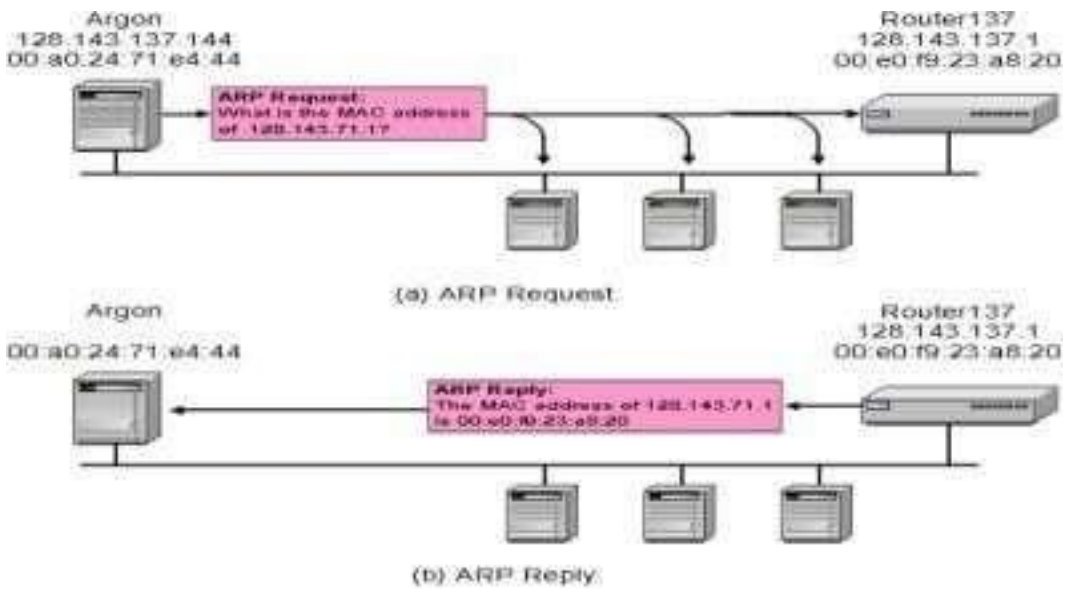
- Each computer gets its configuration from a "pool" of available numbers automatically for a specific time period (called a leasing period), meaning no wasted numbers.
- When a computer has finished with the address, it is released for another computer to use. Configuration information can be administered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

DHCP message types:

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

DHCP Operations:-

1. DHCP Discover



2. DHCP Offer

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

5.4.4 Installing DHCP in Ubuntu:

Open terminal and type following commands:-

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

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

3.

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

option subnet-mask 255.255.255.0;

option broadcast-address 10.1.32.255;

subnet 192.168.1.0 netmask 255.255.255.0

range 10.1.32.10 10.1.32.20; }

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

   sudo service isc-dhcp-server restart

   sudo service isc-dhcp-server start

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

Conclusion:
Hence we Installed and Configured DHCP and studied Installation of Software on remoteMachine.

Outcome: Understand the concept of DHCP and Configuring DHCP and installation of software.

FAQs:

1) What is DHCP protocol and on which layer it works?
2) DHCP stands for?
3) How DHCP protocol works?
4) What are DHCP message types: