**EXPERIMENT-1**

**Aim:** Identify different types of Network cables and practically implement the cross-wired cable and straight through cable using crimping tool.

**Equipment’s/Tools:**

CAT 5 network cable, RJ-45 Connectors, Crimping tool, Cable tester.

**Description:**

Network media refers to the communication channels used to interconnect nodes on a computer network. Typical examples of network media include copper coaxial cable, copper twisted pair cables and optical fiber cables used in wired networks, and radio waves used in wireless data communications networks.

Networking cables are networking hardware used to connect one network device to other network devices or to connect two or more computers to share printers, scanners etc. Different types of network cables, such as coaxial cable, optical fiber cable, and twisted pair cables, are used depending on the network's physical layer, topology, and size. The devices can be separated by a few meters (e.g. via Ethernet) or nearly unlimited distances (e.g. via the interconnections of the Internet).

There are several technologies used for network connections. Patch cables are used for short distances in offices and wiring closets. Electrical connections using twistedpair or coaxial cable are used within a building. Optical fiber cable is used for long distances or for applications requiring high bandwidth or electrical isolation. Many installations use structured cabling practices to improve reliability and maintainability. In some home and industrial applications power lines are used as network cabling.

There are two types of Media:

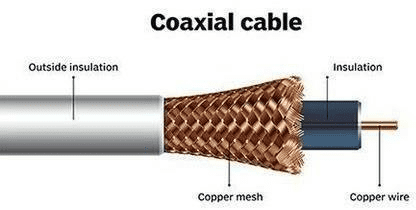
**A. Guided Media**

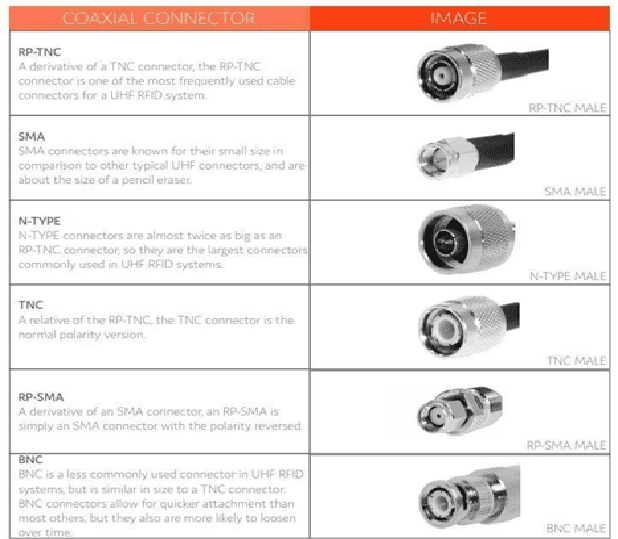
It is also referred to as Wired or Bounded transmission media. Signals being transmitted are directed and confined in a narrow pathway by using physical links. There are 3 major types of Guided Media:

**Coaxial Cables:**

Coaxial cables form a transmission line and confine the electromagnetic wave inside the cable between the center conductor and the shield. The transmission of energy

in the line occurs totally through the dielectric inside the cable between the conductors. Coaxial lines can therefore be bent and twisted (subject to limits) without negativeeffects, and they can be strapped to conductive supports without inducing unwanted currents in them. Early Ethernet, 10BASE5 and 10BASE2, used baseband signaling over coaxial cables. In the 20th century the L-carrier system used coaxial cable for long- distance calling. Coaxial cables are commonly used for television and other broadband signals. Although in most homes coaxial cables have been installed for transmission of TV signals, new technologies (such as the ITU-T G.hn standard) open the possibility of using home coaxial cable for high-speed home networking applications (Ethernet over coax).

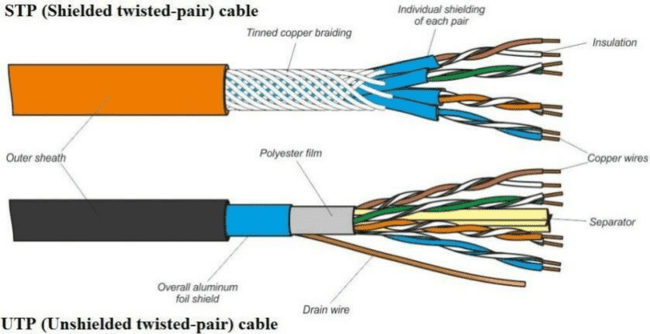
There are different types of connectors used for Coaxial cables:



**Twisted Pair:**

Twisted pair cabling is a form of wiring in which pairs of wires (the forward and return conductors of a single circuit) are twisted together for the purposes of canceling out electromagnetic interference (EMI) from other wire pairs and from external sources. This type of cable is used for home and corporate Ethernet networks. Twisted pair cabling is used in short patch cables and in the longer runs in structured cabling. An Ethernet crossover cable is a type of twisted pair Ethernet cable used to connect computing devices together directly that would normally be connected via a network switch, Ethernet hub or router, such as directly connecting two personal computers via their network adapters. Most current Ethernet devices support Auto MDI-X, so it doesn't matter whether you use crossover or straight cables. Based on how pairs are stripped in the plastic sheath, there are two types of twisted-pair cable; UTP and STP.

● In the UTP (Unshielded twisted-pair) cable, all pairs are wrapped in a single plastic sheath.

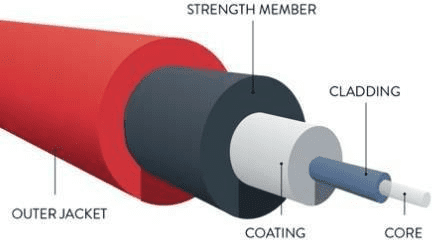
● In the STP (Shielded twisted-pair) cable, each pair is wrapped with an additional metal shield, then all pairs are wrapped in a single outer plastic sheath.

There are different types of connectors used for Twisted pair:

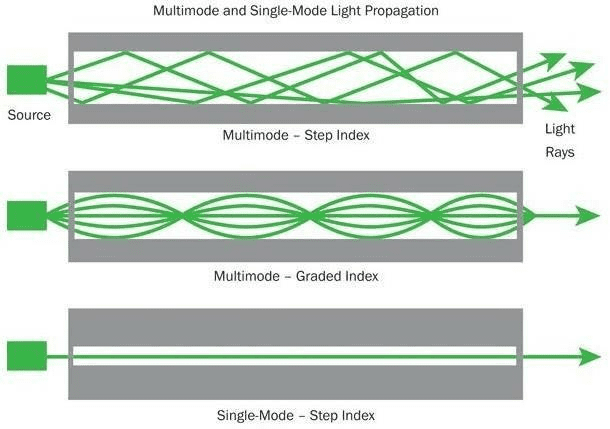
| **RJ-11**: This connector supported the early 10BASE-T networks and Voice Grade telephone connections with 6-pin capability. While still used in telephone implementations, RJ-11 has been surpassed in networking by RJ-45which will be described later. RJ-11 supports speeds of 10Mbps, which is adequate for voice, but highly inefficient for networking which can attain speeds of 1000Mbps. |  |
| --- | --- |
| **RJ-45:** This connector succeeded RJ-11 for network use. RJ-45 uses four twisted pairs of wire configured to reduce cross-talk and other cable related problems. In its initial configuration, RJ-45 was capable of matching RJ-11 speeds, 10Mbps, which was the standard for the time. Changes in network hardware made it possible for this connector to reach speeds of 100Mbps. As the technology has advanced, even greater speeds can be attained by making enhancements to the internal configuration of the connector and the cable while leaving the original shape and size of the connector unchanged. |  |

**Fibre Optics:**

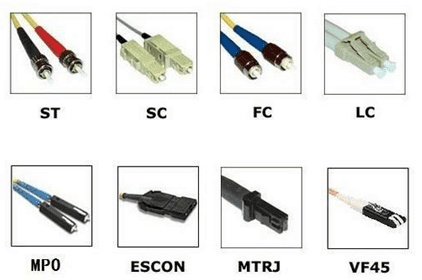
An optical fibre cable consists of a centre glass core surrounded by several layers of protective material. Optical fibre deployment is more expensive than copper but offers higher bandwidth and can cover longer distances.



There are two major types of optical fiber cables: shorter-range multi-mode fiber and long-range single-mode fiber.



There are different types of connectors used for Fiber Optic Media:



**Straight through cabling V/s Cross-wired cable:**

Ethernet cables can be wired as straight through or crossover. The straight through is the most common type and is used to connect computers to hubs or switches. They are most likely what you will find when you go to your local computer store and buya patch cable. Crossover Ethernet cable is more commonly used to connect a computer to a computer and may be a little harder to find since they aren’t used nearly as much as straight through Ethernet cable.

**Straight through cabling:**

The wire has two sides. Let one side is Side A and the other side is Side B. Do the following steps with Side A of the wire.

● Step 1. Remove the plastic cover from the cable up to two inches. You will see 4 twisted pairs (total 8 wires). In each twisted pair one wire will be colored and the other will be white. For example one will be Green and the other will be White having Green marks. The latter is called Green-White. Similarly there will be Brown wire twisted with Brown-White, Blue wire twister with Blue-White, Orange twisted with Orange-White

● Step 2. Untwist the wires and make them smooth(don’t remove the plastic covers from the metal wires).

● Step 3. Arrange the wires in the order: Orange-White, Orange, Green-White, Blue, Blue-White, Green, Brown-White, Brown.

● Step 4. Cut the wires in straight fashion and insert in the RJ-45 Jack.

● Step 5.Using the Crimping tool, punch it properly.

For Side B: Perform Step 1-5 for Side B.

Using a Cable Tester test for correctness of crimping.

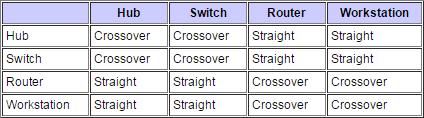
**Cross Over cable**



Side A: Repeat Steps 1 to 5 for side A similar to the process for straight through cable.

Side B: Arrange the wires as: green-white, green, orange-white, blue, blue-white, orange, brown-white, brown. And punch it properly.

Using a Cable Tester test for correctness of crimping.

The following table shows the list of devices and their Twisted pair media connectivity between them:

**B.** **Unguided** **Media:**

It is also referred to as Wireless or Unbounded transmission media. No physical medium is required for the transmission of electromagnetic signals.

**Result:** Thus the types of Network cables and practically implement the cross-wired cable and straight through cable using crimping tool is performed.

**EXPERIMENT-2**

**Aim:** Demonstrate connectivity of wired & wireless devices in Local Area Network using Hub, Switch and Router.

**Description:**

Network devices are the devices that interconnect networks. Because these devices connect network entities, they are known as connectivity devices. These devices include:

● Network Interface Card (NIC)

● Repeater

● Hub

● Bridge

● Switch

● Router

**Network Interface Card (NIC):**

A network interface card (NIC) is a hardware component without which a computer cannot be connected over a network. It is a circuit board installed in a computer that provides a dedicated network connection to the computer. It is also called networkinterface controller, network adapter or LAN adapter.

**NIC cards are of two types:**

● Internal Network Cards- In internal networks cards, motherboard has a slot for the network card where it can be inserted. It requires network cables to provide network access. Internal network cards are of two types. The first type uses Peripheral Component Interconnect (PCI) connection, while the second type uses Industry Standard Architecture (ISA).

● External Network Cards- In desktops and laptops that do not have an internal NIC, external NICs are used. External network cards are of two types: Wireless and USB based. Wireless network card needs to be inserted into the motherboard, however no network cable is required to connect to the network. They are useful while traveling or accessing a wireless signal.

**Repeater:**

A repeater is an electronic device that receives a signal and retransmits it. Repeaters are used to extend transmissions so that the signal can cover longer distances or be received on the other side of an obstruction.

When an electrical signal is transmitted via a channel, it gets attenuated depending upon the nature of the channel or the technology. This poses a limitation upon the length of the LAN or coverage area of cellular networks. This problem is alleviated by installing repeaters at certain intervals.

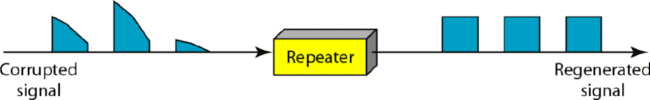
Repeaters amplify the attenuated signal and then retransmit it. Digital repeaters can even reconstruct signals distorted by transmission loss. So, repeaters are popularly incorporated to connect between two LANs thus forming a large single LAN.

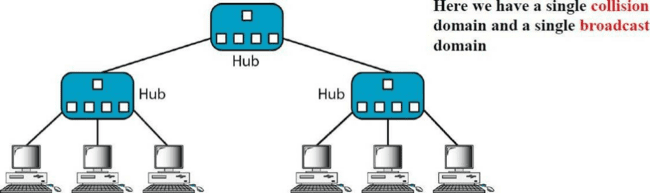
**Hub:**

Hubs are networking devices operating at a physical layer of the OSI model that are used to connect multiple devices in a network. They are generally used to connect computers in a LAN.

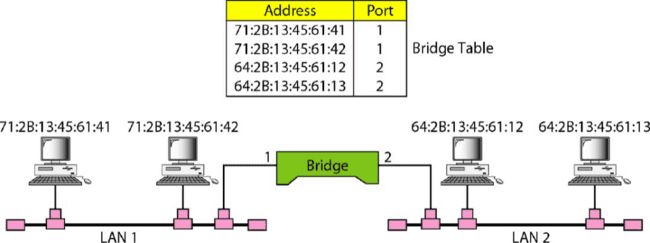
A hub has many ports in it. A computer which intends to be connected to the network is plugged in to one of these ports. When a data frame arrives at a port, it is broadcast to every other port, without considering whether it is destined for a particular destination device or not. The entire hub forms a single collision domain, and a single Broadcast domain. (Collision domain: is that part of the network when two or more nodes transmitat the same time collision will happen).

**There are two types of hubs:**

● Passive Hubs − Passive hubs connects nodes in a star configuration by collecting wiring from nodes. They broadcast signals onto the network without amplifying or regenerating them. As they cannot extend the distance between nodes, theylimit the size of the LAN.

● Active Hubs − Active hubs amplify and regenerate the incoming electrical signals before broadcasting them. They have their own power supply and serves both as a repeater as well as connecting center. Due to their regenerating capabilities, they can extend the maximum distance between nodes, thus increasing the size of LAN.

**Bridge**

A bridge is a network device that connects multiple LANs (local area networks) together to form a larger LAN. The process of aggregating networks is called network bridging. A bridge connects the different components so that they appear as parts of a single network. Bridges operate at the data link layer of the OSI model. They are mainly used to divide the collision domain caused by hubs.

Bridges are used to divide (segment) the LAN into smaller LANs segments, or to connect LANs that use identical physical and data link layers protocol. Each LAN segment is a separate collision domain. Bridge does not send the received frame to all other interfaces like hubs and repeaters, but it performs filtering. A bridge has a table used in filtering decisions

**Switches:**

Switches are networking devices operating at layer 2 or a data link layer of the OSI model. They connect devices in a network and use packet switching to send, receive or forward data packets or data frames over the network.

A switch has many ports, to which computers are plugged in (sometimes referred to as multiport bridges). When a data frame arrives at any port of a network switch, it examines the destination address, performs necessary checks and sends the frame to the corresponding device(s). It supports unicast, multicast as well as broadcast communications.

There are varieties of switches that can be broadly categorized into 4 types:

● Unmanaged Switch − These are inexpensive switches commonly used in home networks and small businesses. They can be set up by simply plugging in to the network, after which they instantly start operating. When more devices needs to be added, more switches are simply added by this plug and play method. They are referred to as u managed since they do not require to be configured or monitored.

● Managed Switch − These are costly switches that are used in organisations with large and complex networks, since they can be customized to augment the functionalities of a standard switch. The augmented features may be QoS (Quality of Service) like higher security levels, better precision control and complete network management. Despite their cost, they are preferred in growing organizations due to their scalability and flexibility. Simple Network Management Protocol (SNMP) is used for configuring managed switches.

● LAN Switch − Local Area Network (LAN) switches connects devices in the internal LAN of an organization. They are also referred as Ethernet switches or data switches. These switches are particularly helpful in reducing network congestion or bottlenecks. They allocate bandwidth in a manner so that there is nooverlapping of data packets in a network.

● PoE Switch − Power over Ethernet (PoE) switches are used in PoE Gogabit Ethernets. PoE technology combine data and power transmission over the same cable so that devices connected to it can receive both electricity as well as data over the same line. PoE switches offer greater flexibility and simplifies the cabling connections

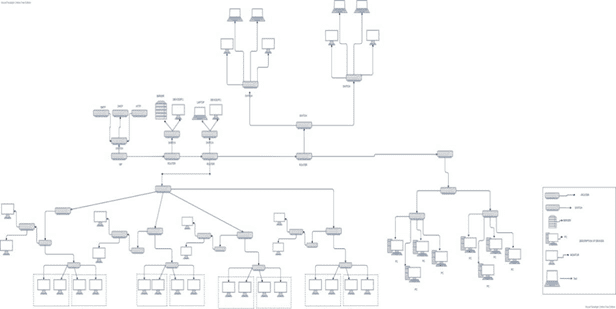
The main advantage of using a switch instead of a hub is that the traffic received by a device is reduced because only frames addressed to a specific device are forwarded to the port on which that device is connected. Switches read the source and destination MAC addresses in the frames and therefore can keep track of who is where, and who is talking to whom, and send data only where it needs to go. If the switch receives a frame whose destination address indicates that it is a broadcast (information meant for everyone) or multicast (information meant for a group), by default it sends the frame out all ports (except for the one on which it was received). All devices connected to one switch port are in the same collision domain, but devices connected to different ports are in different collision domains. By default, all devices connected to a switch are in the same broadcast domain.

**Router:**

Routers are networking devices operating at layer 3 or a network layer of the OSI model. The router is a device which connects multiple logical networks. They are responsible for receiving, analysing, and forwarding data packets among the connected computer networks. When a data packet arrives, the router inspects the destination address, consults its routing tables to decide the optimal route and then transfers the packet along this route. The functioning of a router depends largely upon the routing table stored in it. The routing table stores the available routes for all destinations. The router consults the routing table to determine the optimal route through which the data packets can be sent.

**Task:**

Analyze the college needs and design a network for your college. Use appropriate network devices. Justify your physical architecture. (You can use any network drawing software available online)



**Result** **:** Thus the connectivity of wired & wireless devices in Local Area Network using Hub, Switch and Router is studied.

**EXPERIMENT – 6**

**Aim:** Implement the error detection mechanism i.e. Cyclic Redundancy Check (CRC)

of data link layer.

**Description:**

* The Data Link Layer is the second layer in the OSI model, above the Physical

Layer, which ensures that the error free data is transferred between the adjacent nodes in

the network. There are two basic strategies for dealing with errors. One way is to include

enough redundant information (extra bits are introduced into the data stream at the

transmitter on a regular and logical basis) along with each block of data sent to enable the

receiver to deduce what the transmitted character must have been. The other way is to

include only enough redundancy to allow the receiver to deduce that error has occurred,

but not which error has occurred and the receiver asks for a retransmission. The former

strategy uses Error-Correcting Codes and latter uses Error-detecting Codes.

Redundancy is achieved through various coding schemes. The sender adds

redundant bits through a process that creates a relationship between the redundant bits

and the actual data bits. The receiver checks the relationships between the two sets of

bitsto detect errors. We can divide coding schemes into two broad categories: block

coding and convolution coding. Cyclic codes are special linear block codes. A subset of

cyclic codes is the cyclic redundancy check (CRC).

**CRC (Cyclic Redundancy Check)**

* This Cyclic Redundancy Check is the most powerful and easy to implement

technique. Unlike checksum scheme, which is based on addition, CRC is based on binary

division. In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are

appended to the end of data unit so that the resulting data unit becomes exactly divisible

by a second, predetermined binary number. At the destination, the incoming data unit is

divided by the same number. If at this step there is no remainder, the data unit is assumed

to be correct and is therefore accepted. A remainder indicates that the data unit has been

damaged in transit and therefore must be rejected.

❖ Bit strings are created as representation of polynomials with coefficients ‘0’  and ‘1’only.

❖ A k-bit frame is regarded as coefficients list for a polynomial with ‘k’ terms (xk-1 to x0)

**Eg: x5 + x4 +x0 = 110001**

❖ When this method is used, the sender and the receiver should agree upon  a generator polynomial, G(x) in advance.

❖ Both the high and low order bits of G(x) must be ‘1’

❖ To compute checksum for some frame with ‘m’ bits ( polynomial = M(x), append  ‘r’ zero bits to the lower end of the frame (r = degree of the generator polynomial)  so that this check summed frame is divisible by G(x).

❖ Divide M(x) by G(x) using modulo-2 division and subtract the remainder  from M(x) using modulo-2subtraction. let the resultant be called as T(x) ❖ T(x) is passed to the receiver and the receiver divides it by G(x). ❖ If there is a remainder, there has been a transmission error.

**Algorithm for computing checksum:**

1. Let ‘r’ be the degree of G(x). append ‘r’ to the lower end of the frame so that  it contains ( m + r) bits.

2. Divide M(x) by G(x) using MOD-2 division.

3. Subtract the remainder from M(x) using MOD-2 subtraction.

4. The result is the check summed frame to be transmitted.

**Program:**

#include <stdio.h>

#include <string.h>

#define max 30

int main() {

char dataword[max], poly[max], tmp[max], t[max];

int i, j, k, p, d;

printf("Enter Data Word:\n");

scanf("%s", dataword);

printf("Enter Divisor:\n");

scanf("%s", poly);

strcpy(tmp, dataword);

d = strlen(dataword);

p = strlen(poly);

for(i = 1; i <= p; i++)

strcat(tmp, "0");

for(i = 0; i < p; i++)

t[i] = tmp[i];

t[i] = '\0';

int x = p;

for(i = 0; i < d - 1; i++) {

if(t[0] == '1') {

for(j = 0; j < p; j++) {

if(t[j] == poly[j]) t[j] = '0';

else t[j] = '1';

}

}

for(k = 0; k < strlen(t) - 1; k++)

t[k] = t[k + 1];

t[k] = tmp[x++];

t[k + 1] = '\0';

}

for(k = 0; k < strlen(t) - 1; k++)

t[k] = t[k + 1];

t[k] = '\0';

strcat(dataword, t);

printf("The Code Word is:\n");

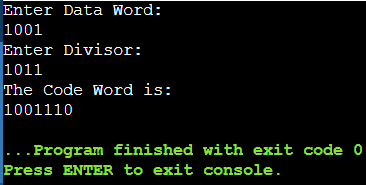
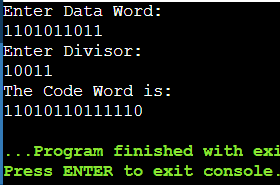
printf("%s", dataword);

return 0;

}

**Output (Test Cases):  CRC**

| **Input** | **Output** |
| --- | --- |
| Enter Data Word:  1001  Enter Divisor:  1011 | The Code Word is:  1001110 |
| Enter Data Word:  1101011011  Enter Divisor:  10011 | The Code Word is:  11010110111110 |

**Result:**Thus the program for error detection i.e. cyclic redundancy check(CRC) is  executed.

**EXPERIMENT-5**

**Aim:** Implement the data link layer framing mechanisms such as Bit stuffing & Character Stuffing.

**Description:**

1. What is Framing?

Since the physical layer merely accepts and transmits a stream of bits without any regard to meaning or structure, it is up to the data link layer to create and recognize frame boundaries. This is called Framing. This can be accomplished by attaching special bit/byte patterns to the beginning and end of the frame. If these bit/byte patterns can accidentally occur in data, special care must be taken to make sure these patterns are not incorrectly interpreted as frame delimiters.

Frames can be of fixed or variable size.

* In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.
* In variable-size framing, we need a way to define the end of one frame and the beginning of the next. Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

In character-oriented (or byte-oriented) framing, data to be carried are 8-bit characters from a coding system such as ASCII. To separate one frame from the next, an 8-bit (1-byte) flag usually protocol dependent is added at the beginning and the end of a frame. Character- oriented framing was popular when only text was exchanged by the data-link layers. The character- oriented framing approaches that are widely used are:

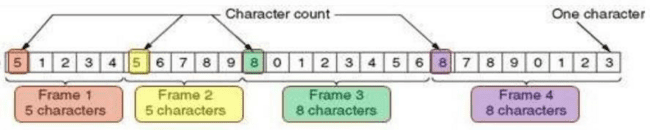
* Character stuffing/Byte Stuffing
* Character count

**Character Stuffing (Byte Stuffing):**

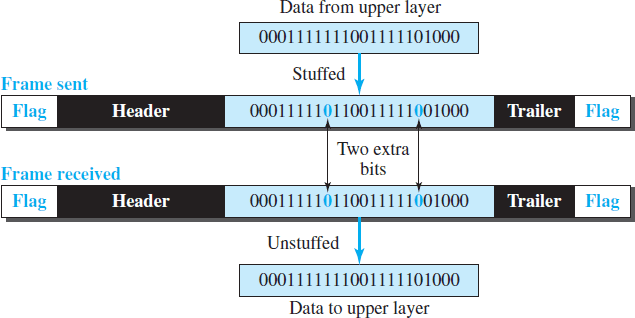
In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC) and has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not as a delimiting flag. If the text contains one or more escape characters followed by a byte with the same pattern as the flag the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text.

**Character** **Count:**

This method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow, and hence where the end of the frame is. The disadvantage is that if the count is garbled by a transmission error, the destination will lose synchronization and will be unable to locate the start of the next frame. So, this method is rarely used.



In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame. The bit-oriented framing approaches that are widely used are:

* Bit stuffing
* Physical layer coding violations 

**Bit stuffing:**

The third method allows data frames to contain an arbitrary number of bits and allows character

codes with an arbitrary number of bits per character. At the start and end of each frame is a flag byte consisting of the special bit pattern 01111110 . Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a zero bit into the outgoing bit stream. This technique is called bit stuffing. When the receiver sees five consecutive 1s in the incoming data stream, followed by a zero bit, it automatically destuffs the 0 bit. The boundary between two frames can be determined by locating the flag pattern.

**Physical** **layer** **coding** **violations:**

The final framing method is physical layer coding violations and is applicable to networks in which the encoding on the physical medium contains some redundancy. In such cases normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The combinations of low- low and high-high which are not used for data may be used for marking frame boundaries.

**Program:**

**Bit Stuffing:**

#include<stdio.h>

#include<string.h>

#define max 100

int main() {

int n, j = 0, i = 0, count;

printf("Enter Number of Frames : ");

scanf("%d", &n);

char frame[max][max]; //frame is a 2D array which holds n no.of bit sequence

for(i=0;i<n;i++){

printf("Enter Frame %d: ", i + 1);

scanf("%s", frame[i]);

}

printf("The transmitted frame is :\n");

for(i = 0; i < n; i++) {

count = 0;

printf("01111110 "); //At the start of each frame flag(01111110) is stuffed

for(j=0; frame[i][j]!='\0'; j++) {

printf("%c", frame[i][j]);

if(frame[i][j] == '1')

count++;

else

count = 0;

if(count == 5) { //if there are five consecutive 1s in data --zero is stuffed

printf("0"); //At the end of each frame flag(01111110) is stuffed

count=0;

}

}

printf(" 01111110\n");

}

return 0;

}

**Character Stuffing:**

#include<stdio.h>

#include<string.h>

int main() {

int n,i,j,k,count;

char flag[10],tmp[10]; char escape[10];

char frame[10][20],temp[10][20];

printf("Enter Number of Frames: ");

scanf("%d", &n);

printf("Enter Flag: ");

scanf("%s", flag);

while(1){

if(strlen(flag)==8) break;

else {

printf("Re-Enter flag of 8 bits :");

scanf("%s", flag);

}

}

printf("Enter Escape Character: ");

scanf("%s",escape);

while(1){

if(strlen(flag)==8) break;

else{

printf("Re-Enter escape of 8 bits :");

scanf("%s", flag);

}

}

for(i=0;i<n;i++) {

printf("Enter Frame %d : ",i+1);

scanf("%s",frame[i]);

while(strlen(frame[i])%8!=0) {

printf("Re-enter the Frame %d of length--Multiples of 8 : ",i+1);

scanf("%s",frame[i]);

}

}

printf("\nThe transmitted frame is:\n");

for(i=0;i<n;i++) {

printf("%s",flag);

for(j=0;j<(strlen(frame[i]))/8;j++) {

count=0;

for(k=j\*8;count<8;k++) {

tmp[count]=frame[i][k];

count++;

}

tmp[count]='\0';

if(!strcmp(tmp,flag))

printf("%s",escape);

else if(!strcmp(tmp,escape))

printf("%s",escape);

printf("%s",tmp);

}

printf("%s",flag);

}

return 0;

}

**Bit Stuffing:**

| **Input** | **Output** |
| --- | --- |
| Enter number of frames: 2  Enter frames:  01101010  01111110 |  |
| Enter Number of Frames: 1  Enter Frame 1: 11111111 |  |

**Character Stuffing:**

| **Input** | **Output** |
| --- | --- |
| Enter Number of Frames: 1  Enter Flag: 10101010  Enter Escape Character: 11110000  Enter Frame 1 : 1111000000110011 |  |
| Enter Number of Frames: 2  Enter Flag: 10101010  Enter Escape Character: 11110000  Enter Frame 1 : 0011001110101010  Enter Frame 2 : 1111000010101010 |  |

**Result:** Thus the program for Framing i.e. bit stuffing and character stuffing is executed

**EXPERIMENT-9**

**Aim:** Implement Unicast Routing Algorithm using any Programming Language (Distance Vector Routing or Dijstraw’s Routing Algorithm).

**Description:**

**Dijstraw’s Routing Algorithm:**

Several algorithms for computing the shortest path between two nodes of a graph

are known. This one is due to Dijkstra (1959). Each node is labeled (in parentheses) with

its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be either tentative or permanent. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

To illustrate how the labeling algorithm works, look at the weighted, undirected

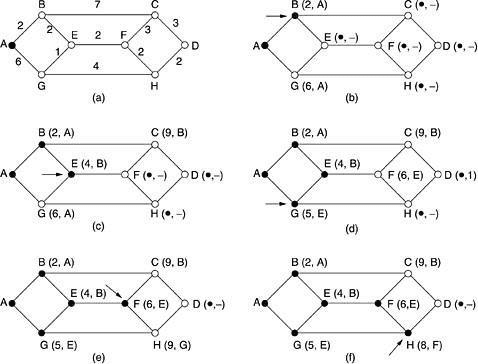
graph of Fig. (a), where the weights represent, for example, distance. We want to find the

shortest path from A to D. We start out by marking node A as permanent, indicated by a

filled-in circle. Then we examine, in turn, each of the nodes adjacent to A (the working node), relabeling each one with the distance to A. Whenever a node is relabeled, we also label it with the node from which the probe was made so that we can reconstruct the final path later. Having examined each of the nodes adjacent to A, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, as shown in Fig. (b). This one becomes the new working node.

We now start at B and examine all nodes adjacent to it. If the sum of the label on B and the distance from B to the node being considered is less than the label on that node, we have a shorter path, so the node is relabeled.

After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively-labeled node with the smallest value. This node is made permanent and becomes the working node for the next round. Figure shows the first five steps of the algorithm.

 Figure: The first five steps used in computing the shortest path from A to D. The arrows

indicate the working node

**Distance Vector Routing Algorithm:**

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

Distance Vector Table Initialization -

Distance to itself = 0

Distance to ALL other routers = infinity number. Neighbor – Null

**Routing Process:**

● A router transmits its distance vector to each of its neighbors in a routing packet.

● Each router receives and saves the most recently received distance vector from each of its neighbors.

● A router recalculates its distance vector when:

o It receives a distance vector from a neighbor containing different information than before.

o It discovers that a link to a neighbor has gone down.

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

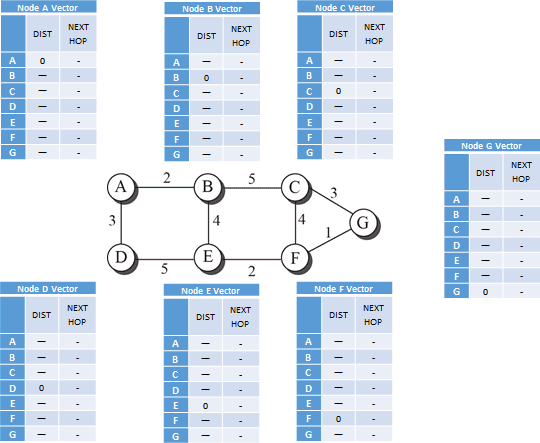
Dx(y) = Estimate of least cost from x to y C(x,v) = Node x knows cost to each neighbor v

Dx= [Dx(y): y ∈ N ] = Node x maintains distance vector Node x also maintains its neighbor’s distance vectors

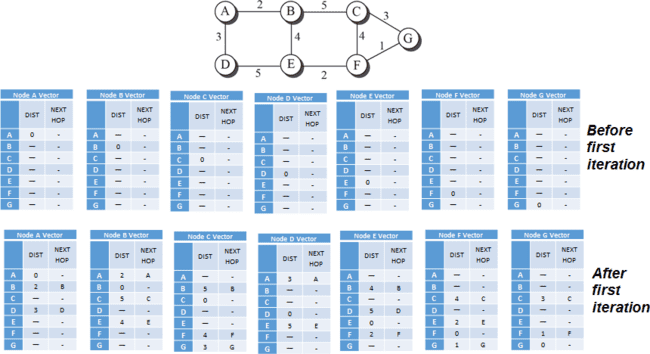
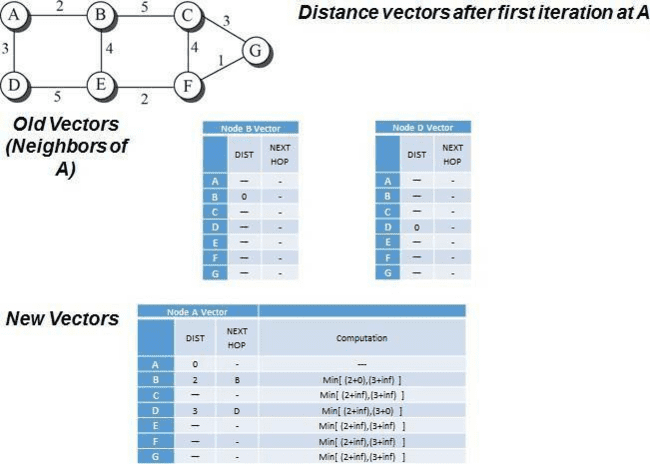
– For each neighbor v, x maintains Dv = [Dv(y): y ∈ N ]

● From time-to-time, each node sends its own distance vector estimate to neighbors.

● When a node x receives new DV estimate from any neighbor v, it saves v’s distance vector and it updates its own DV using B-F equation: Dx(y) = min { C(x,v) + Dv(y), Dx(y) } for each node y ∈ N



Distance Vectors of all Nodes Initially

 Distance Vectors of all Nodes after First Iteration 

**Program:**

#include <stdio.h>

struct node {

int dist[20];

int from[20];

} route[10];

int main() {

int dm[20][20], no,i,j,c=1;

printf("Enter no of nodes." ); scanf("%d",&no);

printf("Enter the distance matrix:");

for ( i = 0; i < no; i++) {

for ( j = 0; j < no; j++) {

scanf("%d",&dm[i][j]); /\* Set distance from i to i as 0 \*/

dm[i][i] = 0;

route[i].dist[j] = dm[i][j];

route[i].from[j] = j;

}

}

printf("------step %d \n",c);

for(i=0;i<no;i++){

printf(" At router %c\n",i+65);

printf("Dest dist next hop\n");

for (int j = 0; j < no; j++) {

if(route[i].dist[j]==999)

printf("%c ~ %d\n", j+65, route[i].dist[j] );

else

printf("%c %d %c\n", j+65, route[i].dist[j] ,route[i].from[j]+65);

}

printf("\n");

}

int flag;

do {

c++;

flag = 0;

printf("------step %d \n",c);

for(int i = 0; i < no; i++) {

for(int j = 0; j < no; j++) {

for(int k = 0; k < no; k++) {

if ((route[i].dist[j]) > (route[i].dist[k] + route[k].dist[j])) {

route[i].dist[j] = route[i].dist[k] + route[k].dist[j];

route[i].from[j] = k;

flag = 1;

}

}

}

printf(" At router %c\n",i+65);

printf("Dest dist next hop\n");

for (int j = 0; j < no; j++)

printf("%c %d %c\n", j+65, route[i].dist[j] ,route[i].from[j]+65);

printf("\n");

}

} while (flag);

return 0;

}

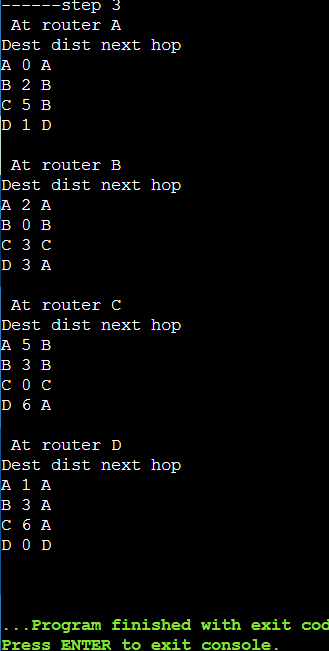
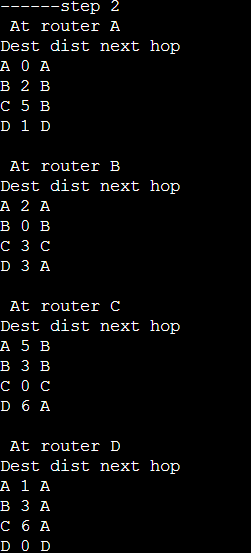
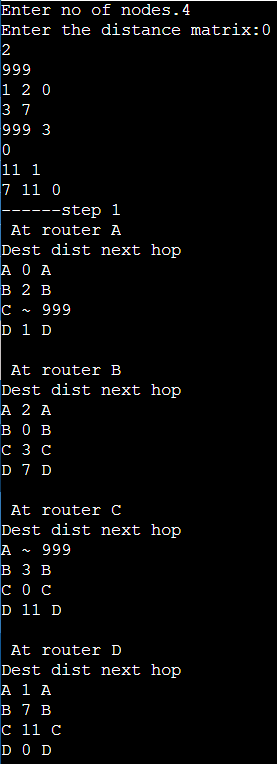
**Dijkstra’s Algorithm:**

| **Sample** **Input** | **Sample** **Output** |
| --- | --- |
| **Enter** **Number** **of** **Nodes:** **5**  **Enter** **Adjacency** **Matrix** **for** **the** **following** **Graph**    **Enter** **Destination:** **3** | **Shortest** **Path:** **0** → **1**→ **2** → **3**  **Distance:** **6** |

**Distance Vector Routing Algorithm:**

| **Input** |  |
| --- | --- |
| **Output** | |
|  | |

**OUTPUT :**

****

**Result:** Thus, the program for implementing Unicast Routing Algorithm using any Programming Language (Distance Vector Routing or Dijkstra’s Routing Algorithm) is executed.

**EXPERIMENT-4**

**Aim:** Understand the interface and Analyze Network Traffic Using Wireshark tool.

**Description:** Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. We could think of a network packet analyzer as a measuring device used to examine what’s going on inside a network cable, just like a voltmeter is used by an electrician to examine what’s going on inside an electric cable. In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed. Wireshark is perhaps one of the best open source packet analyzers available today. Wireshark is an open source software project, and is released under the GNU General Public License (GPL). We can freely use Wireshark on any number of computersyou like, without worrying about license keys or fees or such. In addition, all source code is freely available under the GPL.

Here are some things Wireshark does not provide:

● Wireshark isn’t an intrusion detection system. It will not warn you when someone does strange things on your network that he/she isn’t allowed to do. However, if strange things happen, Wireshark might help you figure out what is really going on.

● Wireshark will not manipulate things on the network, it will only "measure" things from it. Wireshark doesn’t send packets on the network or do other active things (except for name resolutions, but even that can be disabled).

People use Wireshark for:

● Network administrators use it to troubleshoot network problems

● Network security engineers use it to examine security problems

● Developers use it to debug protocol implementations

● People use it to learn network protocol internals

● Beside these examples Wireshark can be helpful in many other situations too.

Features Wireshark provides:

● Available for UNIX and Windows.

● Capture live packet data from a network interface.

● Open files containing packet data captured with tcpdump/WinDump,Wireshark, and a number of other packet capture programs.

● Import packets from text files containing hex dumps of

● Display packets with very detailed protocol information.

● Save packet data captured.

● Export some or all packets in a number of capture file formats.

● Filter packets on many criteria.

● Search for packets on many criteria.

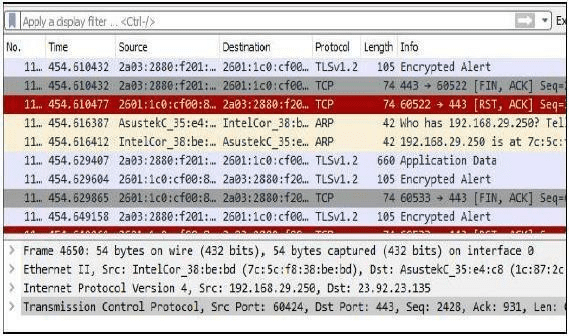
● Colorize packet display based on filters.

● Create various statistics.

● Live capture from many different network media

Wireshark can capture traffic from many different network media types - and despite its name - including wireless LAN as well. Which media types are supported, depends on many things like the operating system you are using.

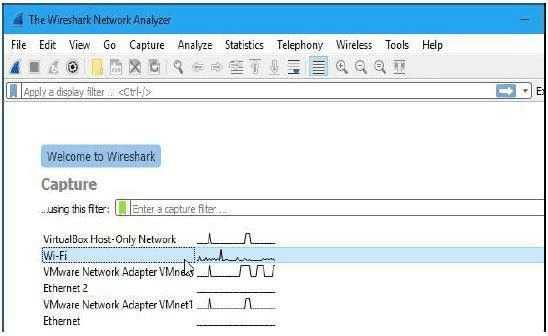
**How** **to** **Use** **Wireshark** **to** **Capture,** **Filter** **and** **Inspect** **Packets**

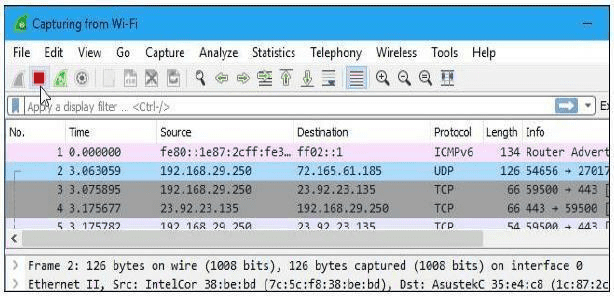


Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color coding, and other features that let you dig deep into network traffic and inspect individual packets. It captures the packets, filtering them, and inspecting them. You can use Wireshark to inspect a suspicious program’s network traffic, analyze the traffic flow on your network, or troubleshoot network problems.

**Getting Wireshark:** We can download Wireshark for Windows or macOS from its official website. If you’re using Linux or another UNIX-like system, you’ll probably find Wireshark in its package repositories. For example, if you’re using Ubuntu, you’ll find Wireshark in the Ubuntu Software Center.

**Capturing Packets:** After downloading and installing Wireshark, you can launch it and double-click the name of a network interface under Capture to start capturing packets on that interface. For example, if you want to capture traffic on your wireless network, click your wireless interface. You can configure advanced features by clicking Capture > Options, but this isn’t necessary for now.

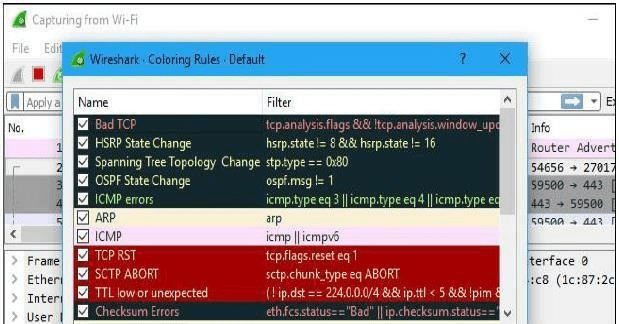


As soon as you click the interface’s name, you’ll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.

If you have promiscuous mode enabled it’s enabled by default you’ll also see all the other packets on the network instead of only packets addressed to your network adapter. To check if promiscuous mode is enabled, click Capture > Options and verify the“Enable promiscuous mode on all interfaces” checkbox is activated at the bottom of this window

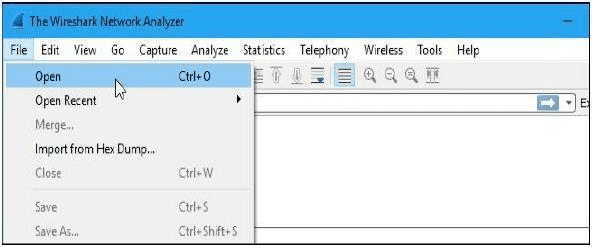
Click the red “Stop” button near the top left corner of the window when you want to stop capturing traffic.

**Color Coding:** You’ll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors—for example, they could have been delivered out of order. To view exactly what the color codes mean, click View >Coloring Rules. You can also customize and modify thecoloring rules from here, if you like.



**Sample Captures:**

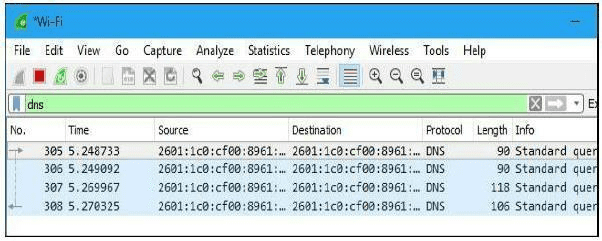
If there’s nothing interesting on your own network to inspect, Wireshark’s wiki has you covered. The wiki contains a page of sample capture files that you can load and inspect. Click File > Open in Wireshark and browse for your downloaded file to open one.

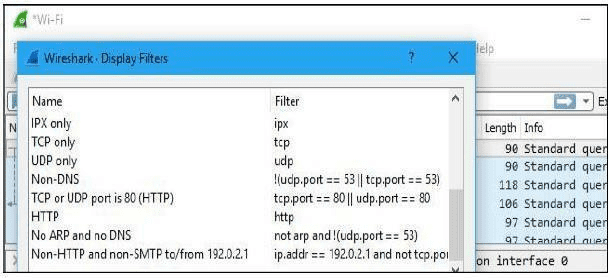
You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.

**Filtering Packets:**

If you’re trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you’ll likely have a large amount of packets to sift through. That is where Wireshark’s filters come in.

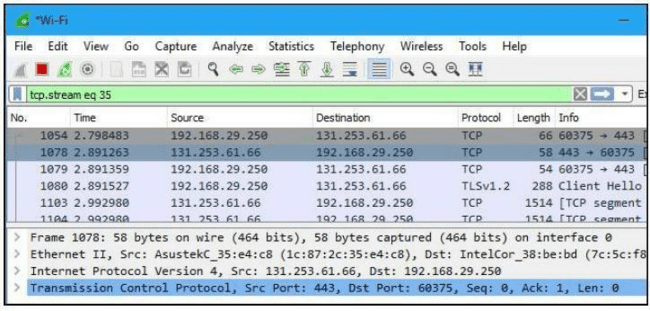
The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type “dns” and you’ll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



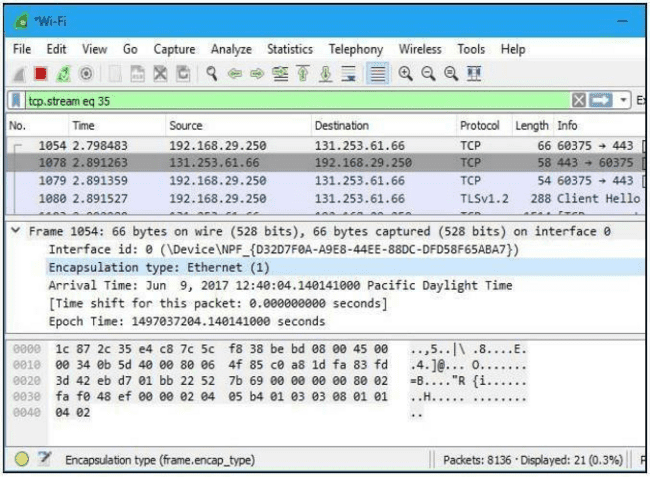
You can also click Analyze> Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.For more information on Wireshark’s display filtering language, read the Building display filter expressions page in the official Wireshark documentation.

Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

You’ll see the full TCP conversation between the client and the server. You can also click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.

Close the window and you’ll find a filter has been applied automatically. Wireshark is showing you the packets that make up the conversation.

**Inspecting** **Packets**

Click a packet to select it and you can dig down to view its details.

You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.

Wireshark is an extremely powerful tool, and this tutorial is just scratching the surface of what you can do with it. Professionals use it to debug network protocol implementations, examine security problems and inspect network protocol internals.

**Tasks** **to-do** **using** **Wireshark:**

• Investigate IP protocol by capturing and studying IP datagrams.

• Capture and study ICMPv4 packets generated by other utility programs such as ping and traceroute.

• Retrieve web pages using HTTP. We use Wireshark to capture packets for analysis. We learn about the most common HTTP messages. We also capture response messages and analyze them.

**Result:** Thus the types of Network cables and practically implement the cross-wired cable and straight through cable using crimping tool is performed.

**EXPERIMENT-3**

**Aim:** Experiment with the basic network commands like Ping, IPCONFIG, Tracert etc.. in real networks.

**Description:** When configuring and managing a computer network, or diagnosing problems in a network, you need to use the correct tools for the task. Most often these tools are software applications. There are various tools available on most computers that can be used to support common networking tasks including:

● Viewing and changing the configuration of your computer’s network interface, such as addresses and other protocol parameters.

● Testing your computer’s network connectivity, such as ability to communicate with other computers and statistics of the communication.

● View and analyse traffic sent/received by your computer, as well as other computers on a network.

The tools that can be used to manage the network vary on different operating systems. For example, Microsoft Windows has different programs than Unix variants such as Ubuntu and Apple MAC OS. Combined with this, many operating systems will have two different interfaces to the same tool: a graphical user interface (GUI) and a command line (text) interface.

Although the programs may be different (including interface and options), the majority of them provide similar level of functionality. Therefore once you learn the functionality using one tool, it will not be too hard for you to perform the same functionality in another operating system. This experiment we use command line interface.

**Testing Network Connectivity**

A basic task for diagnosing the connectivity of a network is to test whether one computer can communicate with another. This is normally performed using the Internet Control Message Protocol (ICMP). A user application that implements ICMP for testing connectivity is ‘ping’.

**1) ping command:**

ping sends a message from your computer to some destination computer, which then immediately responds. ping measures the time it takes from sending the message, to

when the response is received. That is, the delay to the destination and back, i.e. the round trip time (RTT).

The simplest way to use ping is to specify the destination as the first parameter:

>ping DESTINATION;

where DESTINATION is the IP address or domain name of the computer you want to test connectivity with. You can stop the ping by pressing Ctrl-C, or you can limit the number of messages sent by ping to COUNT messages using the -c parameter:

>ping -c COUNT DESTINATION

**Viewing** **Network** **Interface** **Information**

Every computer connects to the LAN via one of its Network Interface Cards (NIC). Almost all operating systems allow the user to view information about the current NIC connection, including:

• MAC (or hardware) address • IP address and subnet mask

• Addresses of other important nodes (servers) on the network • Traffic sent/received by the NIC

Operating systems often allow administrator users to modify some of the above information as well. To view the basic network information from your computer the ‘ipconfig’ command is used. The main command to view and edit the specific network interface information is ‘ifconfig’.

**ipconfig:**

ipconfig result will differ depending on your network setup and the type of network adapters installed on your computer.

To view the network information:

>ipconfig;

| **Parameter** | **Description** |
| --- | --- |
| /all | Display the full TCP/IP configuration information for all network adapters. |
| /release | Release the IPv4 address for the specified adapter. |
| /release6 | Release the IPv6 address for the specified adapter. |
| /renew | Renew the IPv4 address for the specified adapter. |
| /renew6 | Renew the IPv6 address for the specified adapter. |

| **Parameter** | **Description** |
| --- | --- |
| /flushdns | Purges the DNS Resolver cache. |
| /displaydns | Display the contents of the DNS Resolver Cache. |
| /? | Displays help information. |

**ifconfig:**

The operating system assigns names to each interface, such as eth0 for on Ethernet NIC and eth1 for another. As the name/number assigned to an interface is automatic, you cannot assume the same scheme is used in different computers, nor can you assume itwill be the same each time you start the same computer. The special loopback interface (which isn’t a real physical interface, but a virtual interface implemented in software inside the OS) is often given the name lo.

To view the information for all interfaces:

> ifconfig

To view the details of a specific interface, such as eth0:

>ifconfig eth0;

**Testing** **a** **Route**

Another useful network connectivity test is to determine the path (or route) that a message takes. That is, what routers does the message pass via on the way to thedestination. Like ping, an ICMP message called ‘tracepath or tracert’ is sent to the destination and returned, but with tracepath/tracert the set of routers along the way also send a response to the source. The tracepath/tracert application can be used by giving a destination IP address or domain name as a parameter:

>tracepath DESTINATION >tracert DESTINATION

The ‘pathping’ provides information about network latency and network loss at intermediate hops between a source and destination. This command sends multiple echo Request messages to each router between a source and destination, over a period of time, and then computes results based on the packets returned from each router. Because this command displays the degree of packet loss at any given router or link, you can determine which routers or subnets might be having network problems.

> pathping /n <ipaddress>

**Converting** **IP** **Addresses** **to** **Hardware** **Addresses** **(Address** **resolution** **protocol)**

Remember that IP addresses are logical addresses. For a computer to send data to another computer on the same LAN/WAN they must use hardware (or MAC) addresses. We can view the information ARP has in your computer using the application arp. Running arp will return a table (called the ARP table or ARP cache) of IP addresses and corresponding hardware addresses that your computer currently knows about:

> arp –a

**Network** **Statistics**

A tool that allows you to view many different network statistics is netstat. For example, you can view interface statistics (similar to ifconfig), routing table statistics (same as route), connection statistics and TCP/IP packet statistics.

You can view the active TCP connections:

>netstat -n –t

You can also view summary TCP/IP statistics:

>netstat –s

Finding MAC address

To get the MAC address of the network adapters (or NIC for short) in your computer

>getmac

The getmac utility can also be used to find the MAC address of network adapters on a remote computer. To be able to do this, you will need the following information:

The IP address or hostname of the remote computer.

Administrator login credential (user name and password) on the remote computer. If the computer is part of a network domain, then you can use your domain administrator credential.

>getmac /s <IPaddress> /u <username> /p <password>

The getmac utility can only remotely retrieve MAC addresses of computers running a Windows operating system only

To get the MAC address of the network adapters with the results formatted in a table with extra information

>getmac/fotable/v

**Tasks** **&** **Output** **(for** **each** **task):**

• View the configuration details, including addresses, of your computers network interfaces.

• Test the network connectivity between your computer and several other computers: another PC in the lab; external web servers

• Using one of the publicly available websites for ping/traceroute, test the connectivity to several external websites.

• Find the IP addresses of several web servers (domains), using several different DNS servers.

• View your ARP cache. Find the hardware address of another computer in the lab using ARP

• View the active TCP connections that your computer has, especially after you visited a website.

• View and browse through the summary network statistics.

• View the DHCP lease information for your computer, and see how it changes as you renew/refresh the lease.

• Find the mac address of another computer in the network.

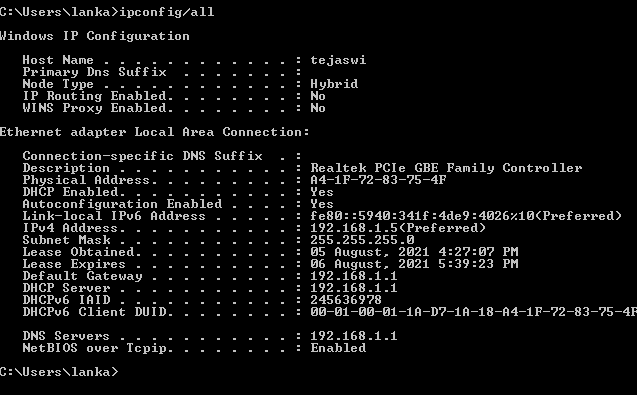
**1** **)Command:>ipconfig/all**

**Description:**

To view the basic network information from your computer the **‘ipconfig’** command is used. **ipconfig** **/all**:Display the full TCP/IP configuration information for all network adapters.

Every computer connects to the LAN via one of its Network Interface Cards (NIC). Almost all operating systems allow the user to view information about the current NIC Connection. ipcongifg/all displays information like:

• MAC (or physical) address

• IP address -192.168.1.5 (from above example) • subnet mask - 255.255.255.0

• Addresses of other important nodes (servers) on the network • DHCP server-192.168.1.1

• Default gateway – system’s router’s address - 192.168.1.1

**2)** **Command:** **>ping** **DESTINATION**

Where, DESTINATION is the IP address or domain name

**Description:**

**P**ing sends a message to the specified device to see if it is active. If that device is active it sends response message. If the source device don’t receive the response message before **Round** **Trip** **Time-**RTT (time taken when a browser sends a request to when it receives a response from a server)

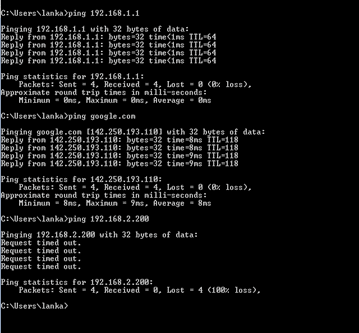
it says **Request** **Timed** **Out**

**T**otal four packets (messages) are sent to the device. If Request timed out is displayed for all four packets then the system of specified ip address/domain name is not in the network (or not working).

Sent:Total packets sent

Receive:Total packets received by receiver Lost:No.of packets lost during transmission

In above example google and 192.168.1.1 are pinged successfully, 192.168.2.200 is not.



**3)Command:>ping** **<destination>**



**Command:** **>tracert** **<destination>**

**Description**: **ping** **:** ping sends a message from your computer to some destination computer, which then immediately responds.

**tracert**:It is used to determine the path (or route) that a message takes. That is, what routers does the message pass via on the way to the destination. Like ping, an ICMP message called ‘tracepath or tracert’ is sent to the destination and returned, but with tracepath/tracert the set of routers along the way also send a response to the source. The tracepath/tracert application can be used by giving a destination IP address or domain name as a parameter

in above example [‘**www.facebook.com’** h](http://www.facebook.com/)as been pinged and traced successfully.

In tracing of facebook , it gave the routers in between like router of system, router of local broadband so on….

**4)** **Command:** **>** **nslookup** **<domain** **name** **or** **IP** **address>**

**Description:**

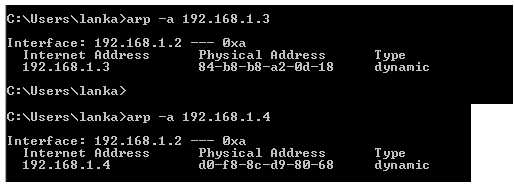
**Nslookup** (stands for “Name Server Lookup”) is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems.

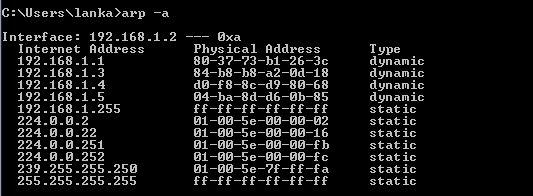
**nslookup** [**www.google.com**](http://www.google.com/) goes to the DNS server and gets back the corresponding IP address **142.250.195.228** from the records.

**5)** **Command:** **>arp** **–a**

Displays current ARP entries by interrogating the current protocol data. If inet\_addr is specified, the IP and Physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.

Running **arp** **–a** will return a table (called the ARP table or ARP cache) of IP addresses and corresponding hardware addresses that the computer currently knows about.(above example)





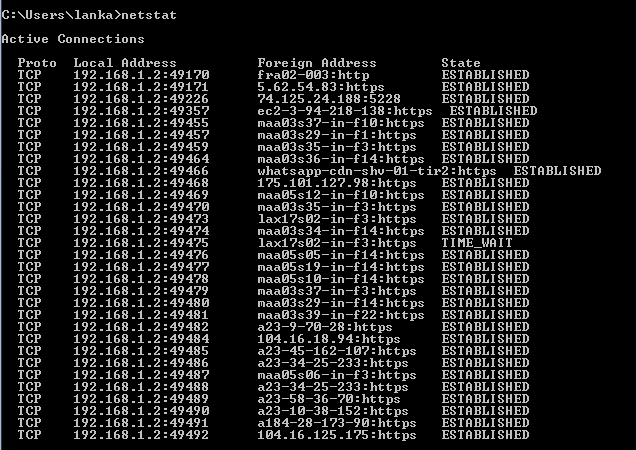
An IP address is specified with the command. So, it displayed the physical address of that particular device.

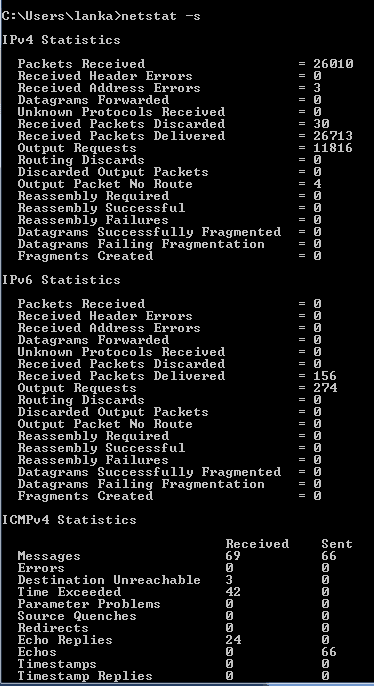
Ex:-physical address of 192.168.1.4 is d0-f8-8c-d9-80-68.

**6)Command:>** **netstat**

**Description:**

**netstat** allows to view the active TCP connections.

The above example shows the established connections with different IP addresses( can be clearly observed when some websites are visited before running the command),their foreign addresses ,protocol type and state of connection(established or time wait).

**7.** **Command:>netstat** **-s**

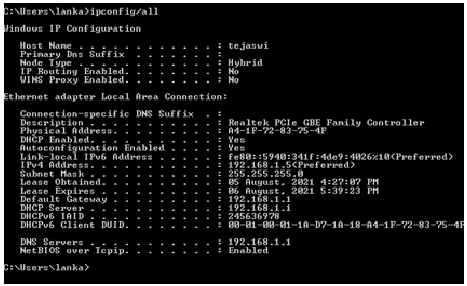
**Description:**

A tool that allows you to view many different network statistics is netstat. For example, you can view interface statistics (similar to ifconfig), routing table statistics (same as route), connection statistics and TCP/IP packet statistics, as shown in above example.

**8.** **View** **the** **DHCP** **lease** **information** **for** **your** **computer,** **and** **see** **how** **it** **changes** **as** **you** **renew/refresh** **the** **lease.**

**Command:>ipconfig** **/all**

Display the full TCP/IP configuration information for all network adapters.

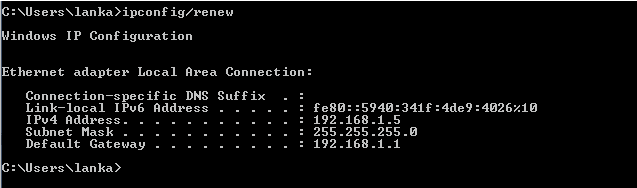


**Lease** **information:**

Lease Obtained : 05 August, 2021 4:27:07 PM 32

Lease Expires : 06 August, 2021 5:39:23 PM

**Command:>** **ipconfig/renew**



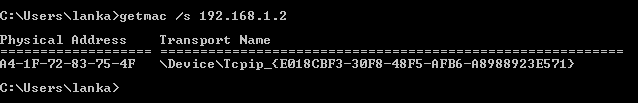
**Description:**

Ipconfig/renew command orders the DHCP client to renegotiate an IP address lease with the DHCP server on the system’s router.

This will either renew the current local IP address with the router or get your computer a new local IP address if the router has not already reassigned the current address.

In above example, the IP address is renewed.

**9.Command:>** **getmac** **/s** **<IP** **address** **>**



The mac address of 192.168.1.2 is displayes in above example.

The getmac utility can be used to find the MAC address(physical address) of network adapters on a remote computer.

**/s** specifies the name or IP address of a remote computer. The default is the local computer.

This example is for the mac address of local computer (192.168.1.5)



**Result:** Thus the basic network commands like Ping, IPCONFIG, Tracert etc.. in realnetworks is executed.

**EXPERIMENT-7**

**Aim:**

Experiment with configuration of Host IP, Subnet Mask and Default Gateway of a device in LAN and establish Peer to Peer network connection.

**Description:**

Every network device has a physical address called a MAC address, which is assigned to the device at the factory. When you buy a network interface card to install into a computer, the MAC address of that card is fixed and can’t be changed. Logical address in terms of computer networking is an address assigned to a device which is uniquely identify a device in a network. Logical address is basically the IP address. IP address is a 32 bit number and it works on layer 3 of the OSI model. Logical addresses are created and used by Network layer protocols such as IP or IPX. The Network layer protocol such as ARP translates logical addresses to MAC addresses. When using a version 4 of IP protocol, the IP address is specified by 32 bits. When using a version 6 of IP protocol, the IP address is specified by 128 bits.

**IPv4 Addressing**

IPv4 addresses are 32-bit numbers that are typically displayed in dotted decimal notation. A 32-bit address contains two primary parts: the network prefix and the host number. All hosts within a single network share the same network address. Each host also has an address that uniquely identifies it. An IPv4 address is written in decimal digits, divided into four 8-bit fields that are separated by periods. Each 8-bit field represents a byte of the IPv4 address. This form of representing the bytes of an IPv4 address is often referred to as the dotted-decimal format.

* Address - The unique number ID assigned to one host or interface in a network.
* Subnet - A portion of a network that shares a particular subnet address.
* Subnet mask - A subnet mask is a 32 bits address used to distinguish between a network address and a host address in IP address. The Network part is
* Interface - A network connection.

There are four different types of IP addresses: public, private, static, and dynamic. While the public and private are indicative of the location of the network—private being used inside a network while the public is used outside of a network when connected to WAN such as Internet—static and dynamic indicate permanency.

In the IPv4 IP address space, there are five classes: A, B, C, D and E. Each class has a specific range of IP addresses (and ultimately dictates the number of devices you can have on your network). Primarily, class A, B, and C are used by the majority of devices on the Internet. IP addresses, before 1993 use the classful addressing where classes have a fixed number of blocks and each block has a fixed number of hosts.

**Classless Interdomain Routing (CIDR)**

Classless Interdomain Routing (CIDR) was introduced in order to improve both address space utilization and routing scalability in the Internet. It was needed because of the rapid growth of the Internet and growth of the IP routing tables held in the Internet routers.

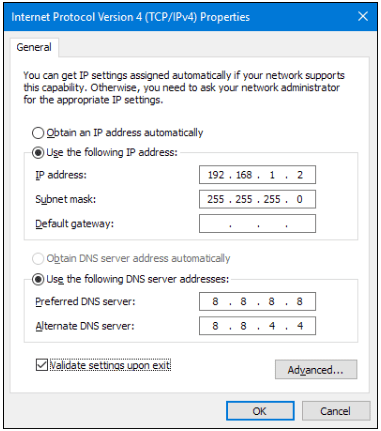
CIDR moves away from the traditional IP classes (Class A, Class B, Class C, and so on). In CIDR , an IP network is represented by a prefix, which is an IP address and some indication of the length of the mask.

To alleviate address depletion, two strategies were proposed and, to some extent, implemented: subnetting and supernetting. While subnetting was devised to divide a large block into smaller ones, supernetting was devised to combine several blocks into a larger block.

Subnetting is the practice of dividing a network into two or smaller networks. It increases routing efficiency, which helps to enhance the security of the network and reduces the size of the broadcast domain. IP Subnetting designates high-order bits from the host as part of the network prefix. This method divides a network into smaller subnets.

**To set a static IP address in Windows 7, 8, and 10:**

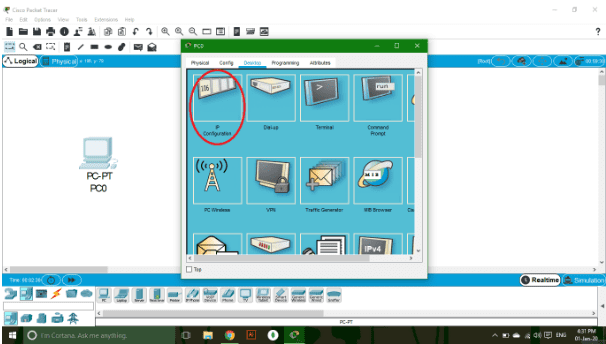
* Click Start Menu > Control Panel > Network and Sharing Center or Network and Internet > Network and Sharing Center.
* Click Change adapter settings.
* Right-click on Wi-Fi or Local Area Connection.
* Click Properties.
* Select Internet Protocol Version 4 (TCP/IPv4).
* Click Properties.
* Select Use the following IP address.
* Enter the IP address, Subnet mask, Default gateway, and DNS server.
* Click OK.

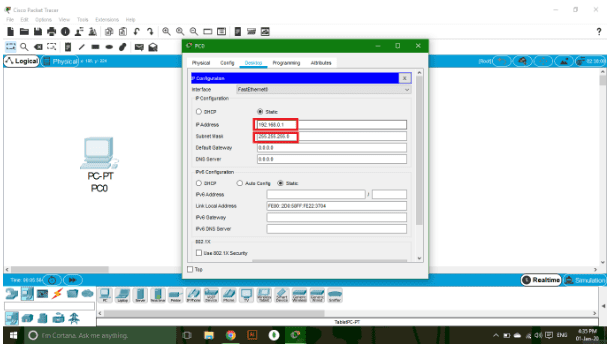


**To set a static IP address in Packet Tracer:**

To set an IP address to a PC in Cisco Packet Tracer follow the instruction below.

* Click on the PC.
* Go to Desktop tab.
* Click on IP Configuration.
* Fill the IP Address. Subnet mask will be auto generated.
* Check you IP Address.





**Steps for establishing peer to peer connectivity:**

* open software packet tracer
* click End Devices icon (lower left corner) or press CTRL + ALT + V
* drag icon general (Personal Computer) and drop to worksheets.
* click Connections icon or press CTRL + ALT + 0 , then click Automatically Choose Connection Type.
* click PC0 then click PC1.
* double click PC0.
* Desktop tab, then click IP Configuration.
* set IP Address for Subnet Mask.(if gateway address is necessary specify it)

IP Address PC0 = 192.168.2.1

Subnet Mask = 255.255.255.0

* close window PC0
* double click PC1
* Desktop tab, then click IP Configuration.
* set IP Address for Subnet Mask.(if gateway address is necessary specify it)

IP Address PC0 = 192.168.2.2

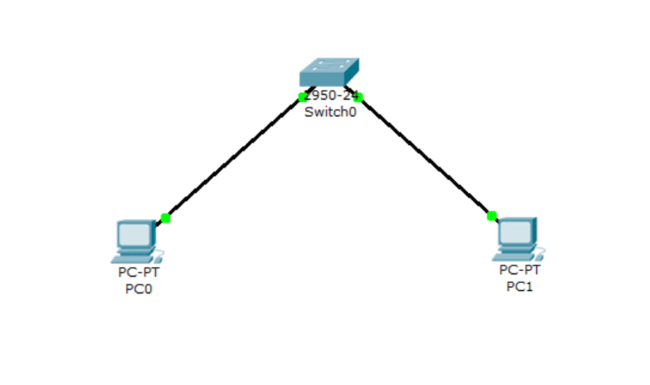
Subnet Mask = 255.255.255.0

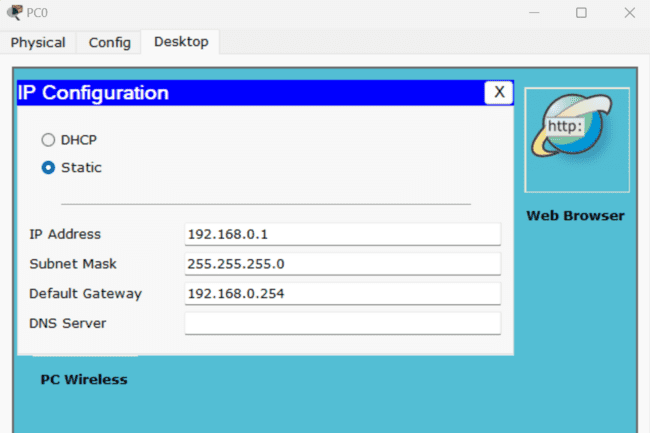
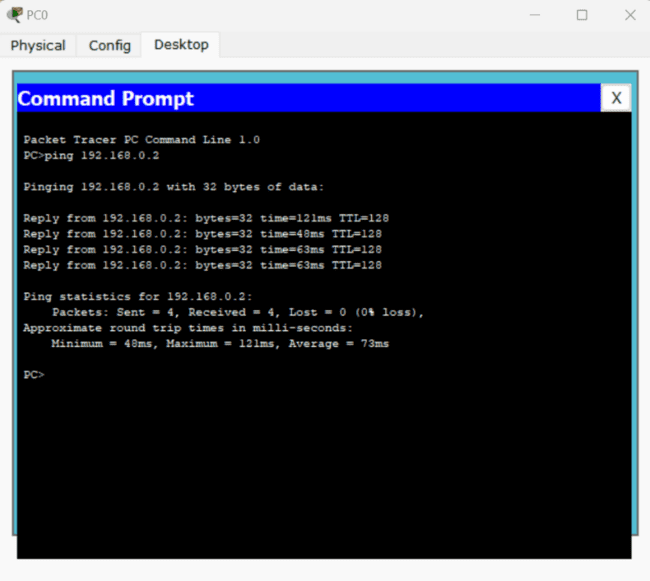
* Desktop tab, then click Command Prompt
* type ping 192.168.2.1 then enter
* if it appears as shown below, it means PC0 and PC1 are connected and successful.

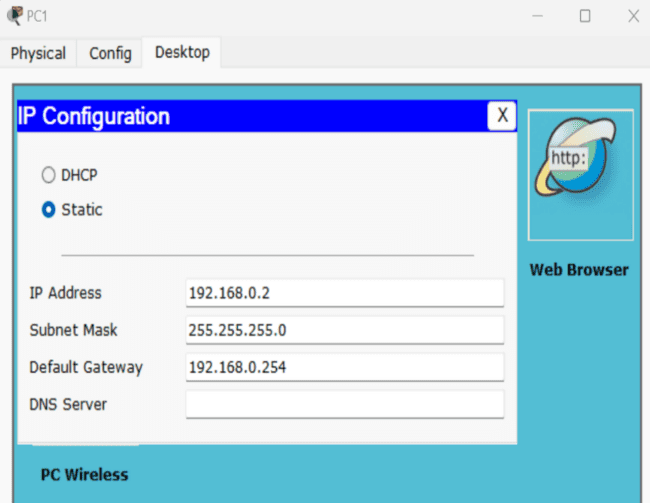
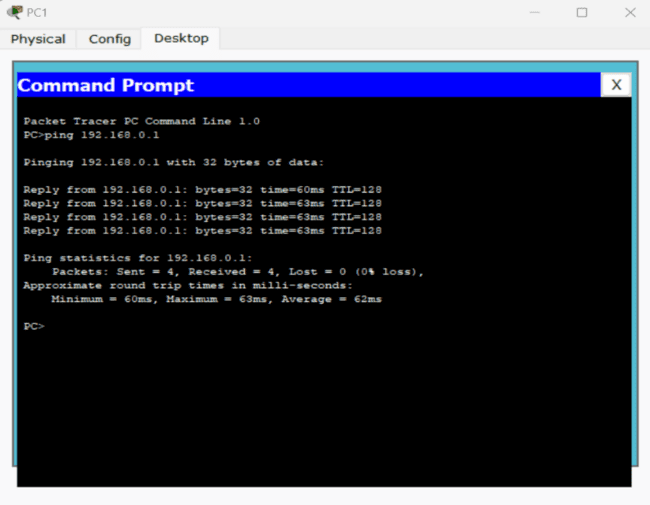
**Task:**

* Use Class B IP Address and establish peer to peer connectivity between two PC’s in packet tracer.
* Connect the following devices and establish peer to peer connectivity by assigning IP address and appropriate cable between the devices :

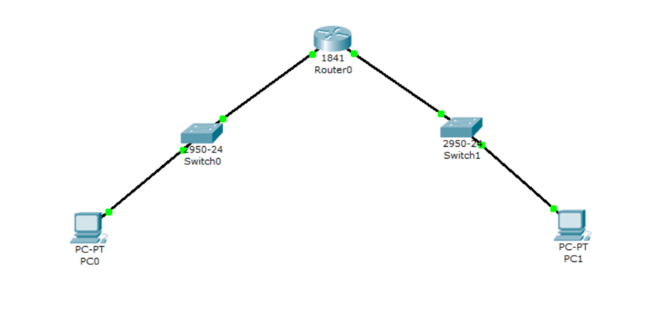
**PC 🡪 Switch 🡪 PC**

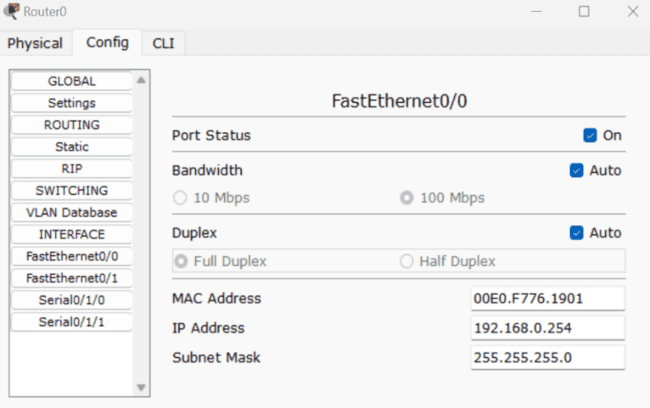
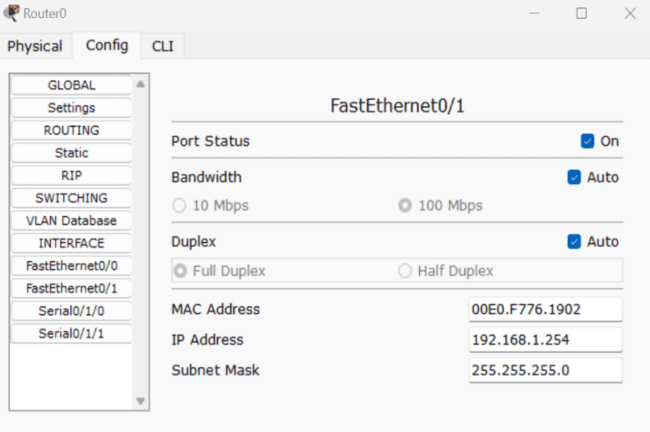


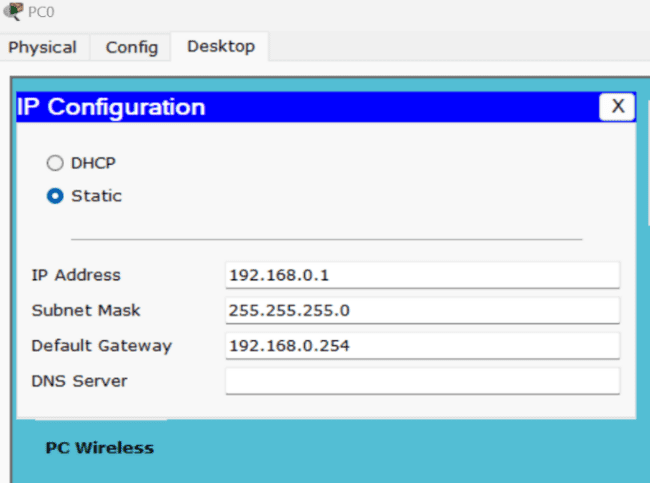
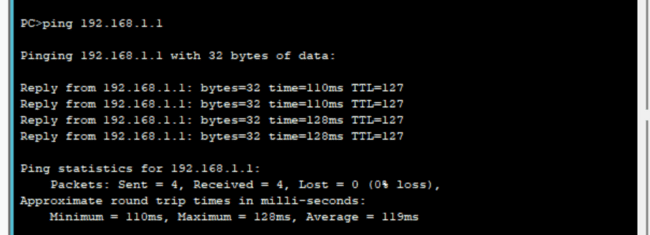
 

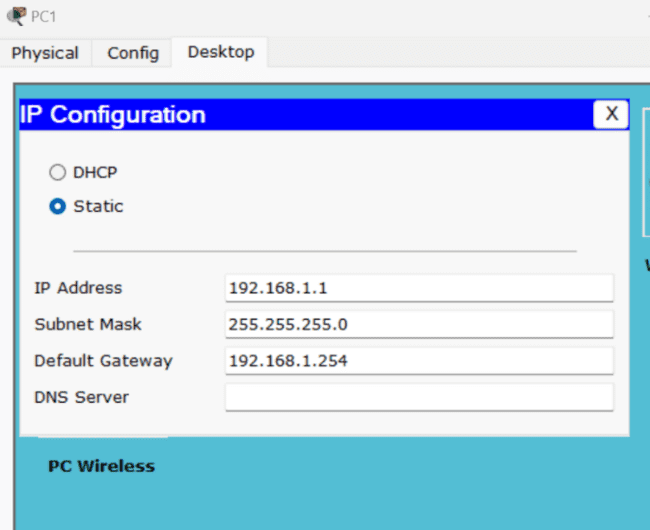
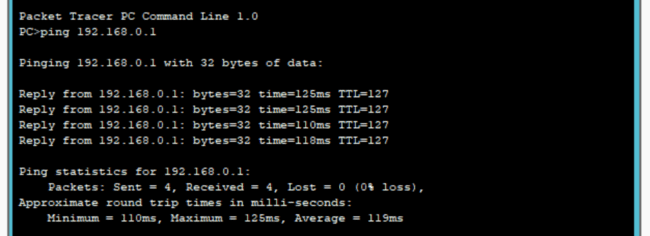
 

**PC 🡪 Switch 🡪 Router 🡪 Switch 🡪 PC**

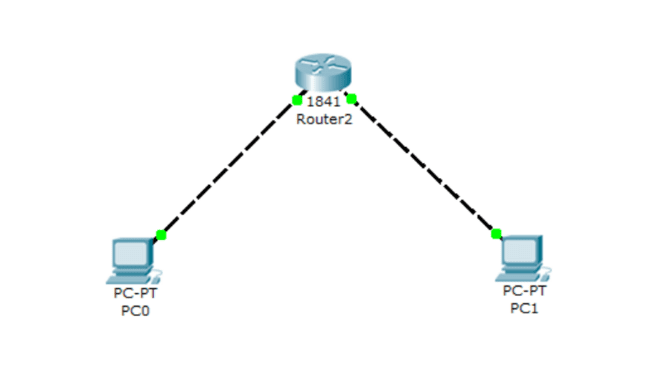
****

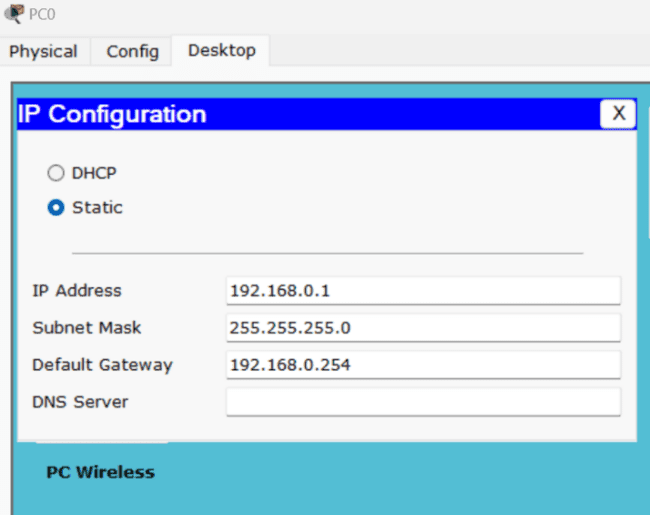
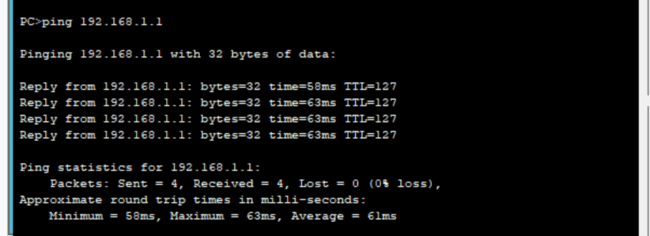
 

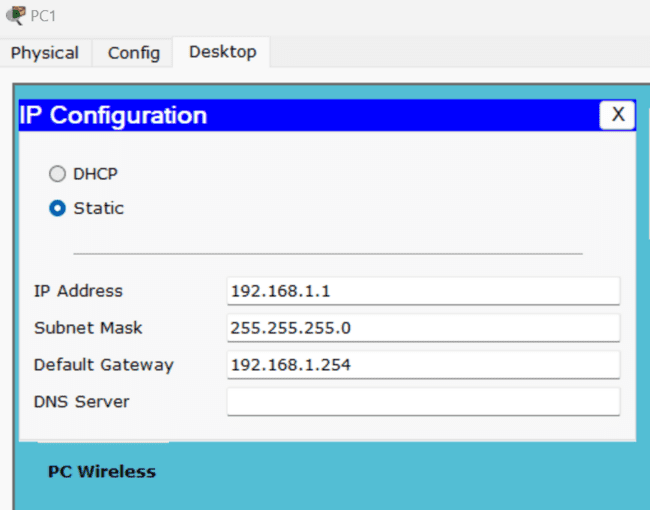
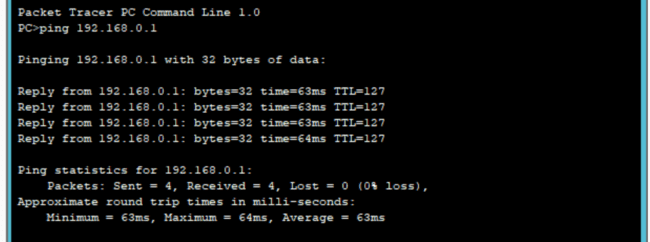
 

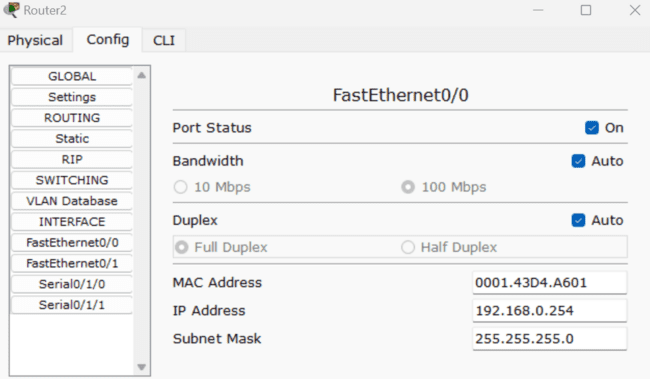
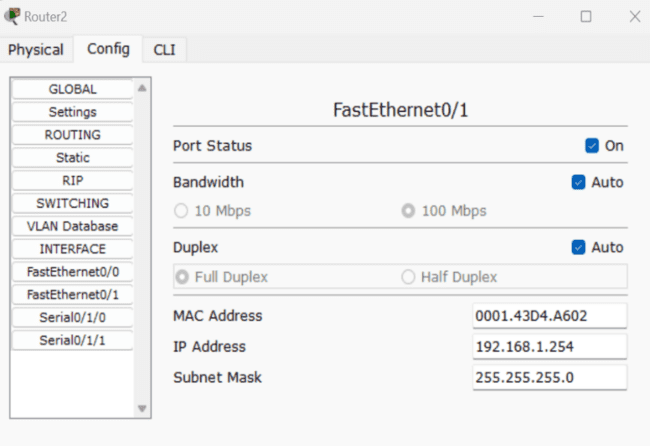
 

**PC 🡪 Router 🡪 PC**



**Result:**

Thus the configuration of Host IP, Subnet Mask and Default Gateway of a device in LAN and establish Peer to Peer network connection is executed.

**EXPERIMENT-8**

**Aim:**

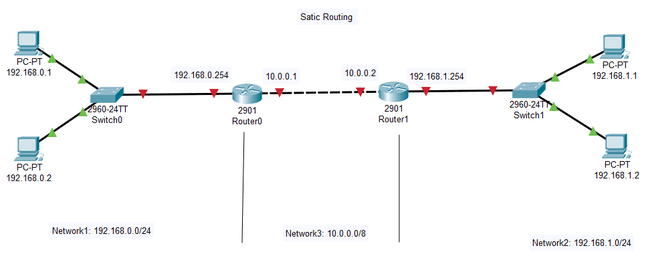
Demonstrate Static Routing and Dynamic Addressing Mechanisms.

**Description:**

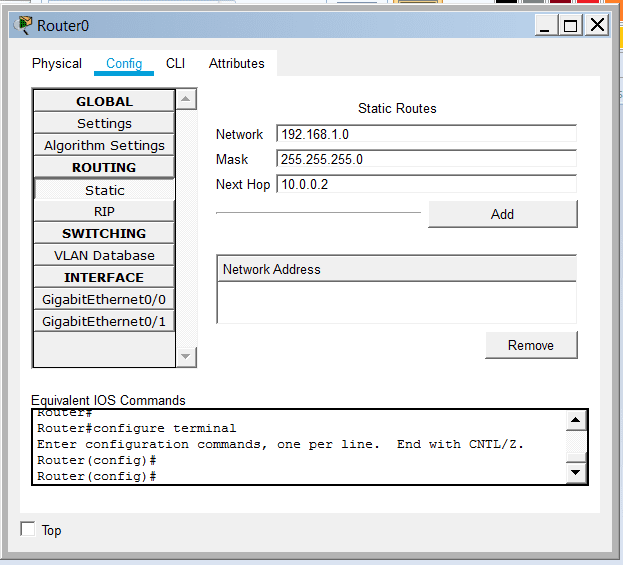
**Static Routing:**

Static routing is a routing type in which a network administrator configures the routes into the routing table to be used by the router to send packets to a destination network.

Steps for assigning static route:

1. Configure a network in Packet Tracer as shown in figure below by assigning the IP addresses, Subnet masks and gateway address. Also assign the interface IP addresses at each of the interfaces of the routers as shown below.
2. At Router0, the packets coming from Network1 has to be routed to Network2, we need to establish a static route. This can be done at Router0 by creating a route to destination network2 passing the data through the interface 10.0.0.2 as the next hop from Router0. To set the route we need to go to the config tab under router0 and click on static button on the left menu. Set the following parameters:

* Network: give the destination network address: 192.168.1.0
* Mask: give the destination network subnet mask address: 255.255.255.0
* Next Hop: give the next hop address to the destination network at router0 i.e. 10.0.0.2



1. Now click on add button below to create a route.
2. Similarly set the static route parameters at router1 for the reverse path as given below and add the route.

* Network: give the destination network address: 192.168.1.0
* Mask: give the destination network subnet mask address: 255.255.255.0
* Next Hop: give the next hop address to the destination network at router0 i.e. 10.0.0.2

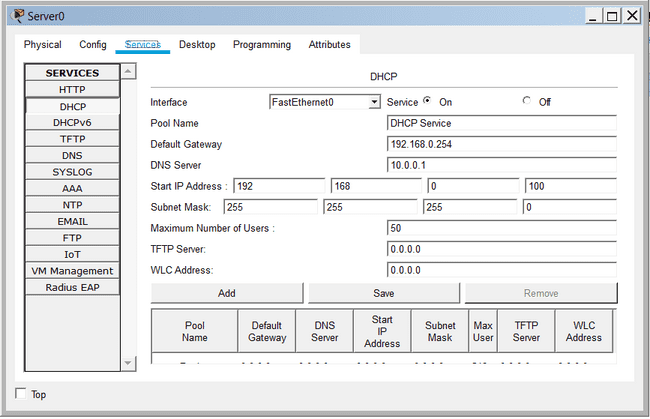
1. Check the connectivity between the devices of Network1 & Network2.

**Dynamic Addressing:**

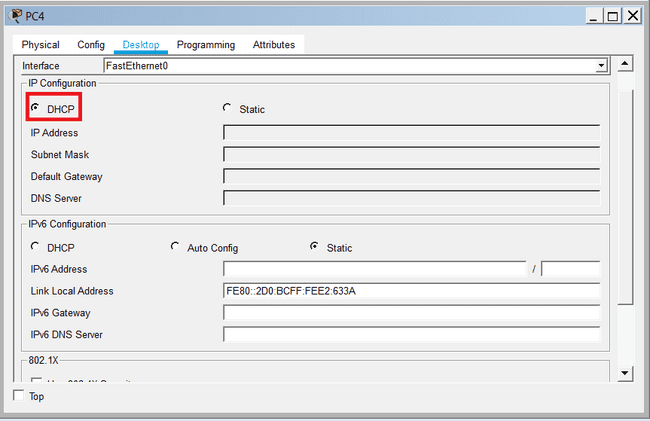
A dynamic IP address is an IP address that is assigned by a device/server in network that use temporarily. If a dynamic address is not in use, it can be automatically assigned to a different device by the server. Dynamic IP addresses are assigned using a protocol called Dynamic Host Configuration Protocol (DHCP). The device that assigns the dynamic addresses to the devices connected to network is called DHCP server. The devices assigned IP addresses are called DHCP Clients.

Steps for dynamic addressing:

* For a device to obtain IP address dynamically we need to configure the DHCP server. The DHCP server can be configured by connecting a server device in the network from the end devices menu.
* Assign an IP address of that network to the sever.
* Under the services section of the server, click on DHCP and set the following parameters and click on Add button.
  + Interface: select interface and set service ON.
  + Pool Name: Assign the pool Name
  + Default Gateway: If gateway is present then specify its address.
  + DNS Server: If DNS server is present, then specify its address.
  + Start IP address: Assign the starting IP address that can be given to DHCP clients.
  + Subnet Mask: Assign the subnet mask of the network.
  + Maximum Number of Users: Enter the number of users that can be connected with DHCP(i.e. clients)



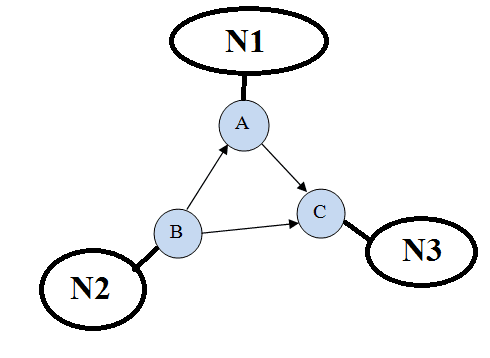
* Connect an End device to the same network and go to IP address settings. Then set DHCP under IP Configuration.

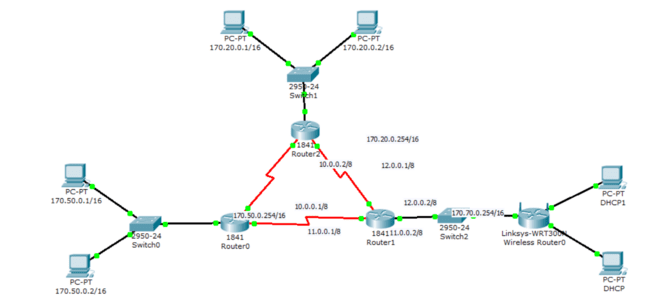


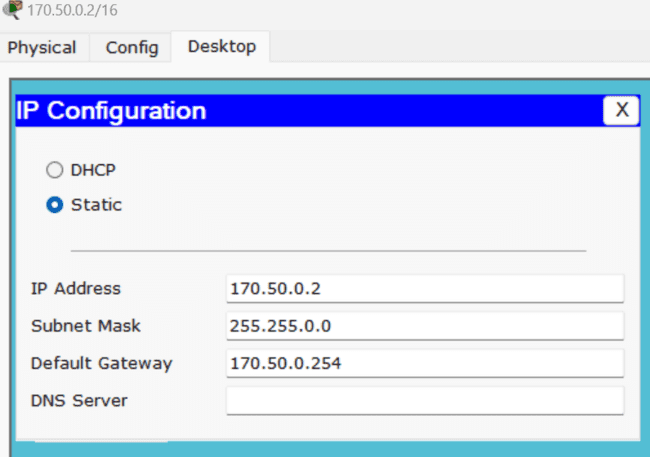
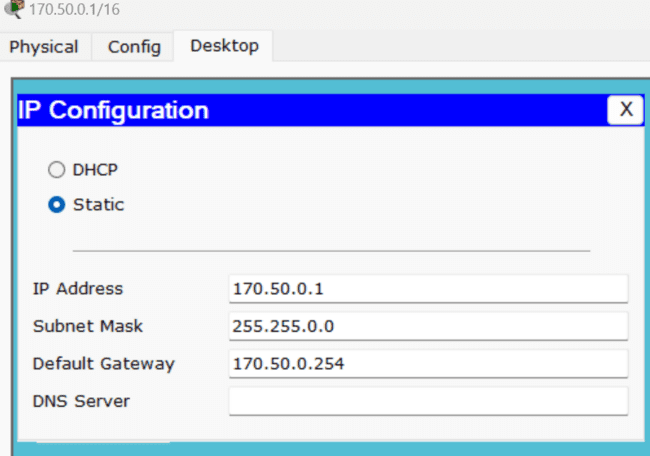
* With a few seconds you will observe that the IP address is assigned to the device.
* You can verify it from command prompt and test its connectivity.

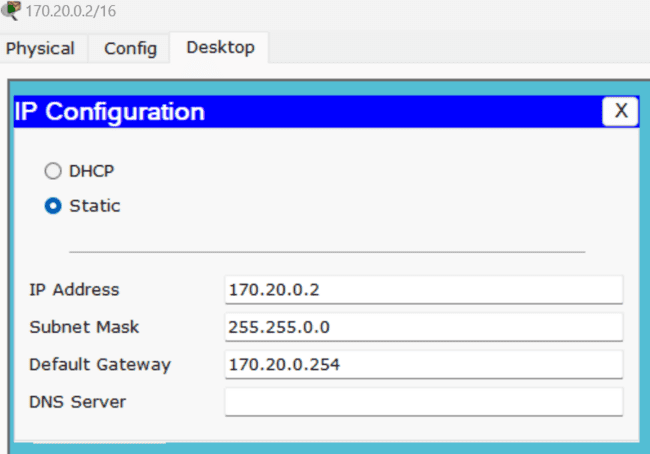
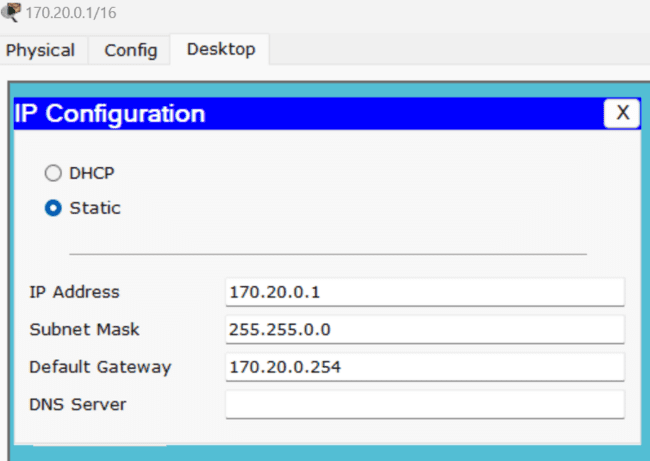
**Task:**

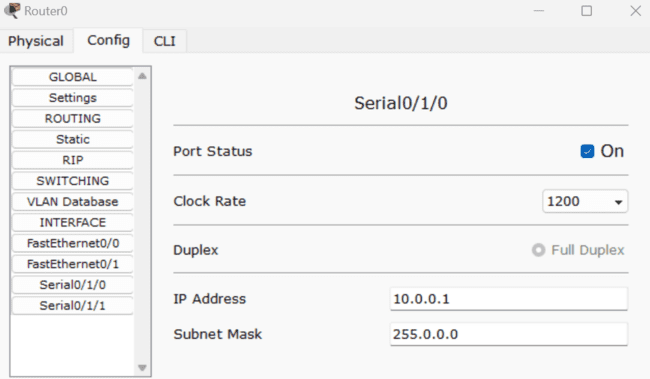
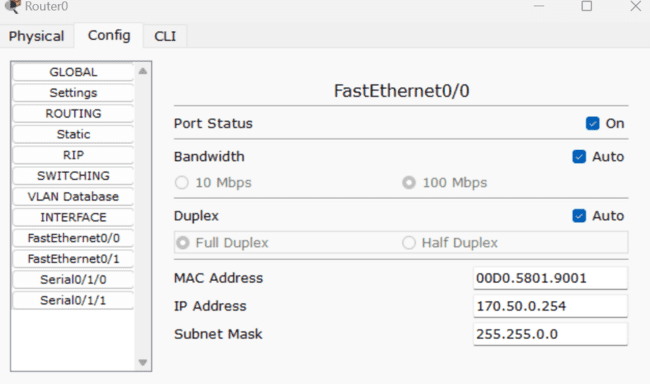
* Design a network for the topology given below and implement static routing & dynamic addressing. Verify the connectivity between the networks.
  + A, B, C are Routers
  + N1, N2, N3 are the networks with two PC’s
  + N1 & N2 are networks with Two PC’s and using static addressing mechanism
  + N3 is the network with Two PC’s and One server with Dynamic Addressing mechanism

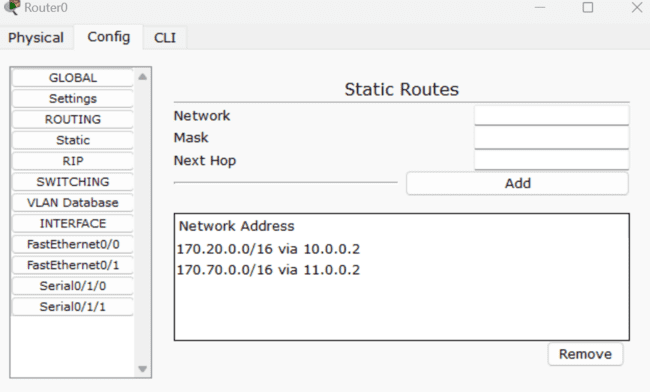
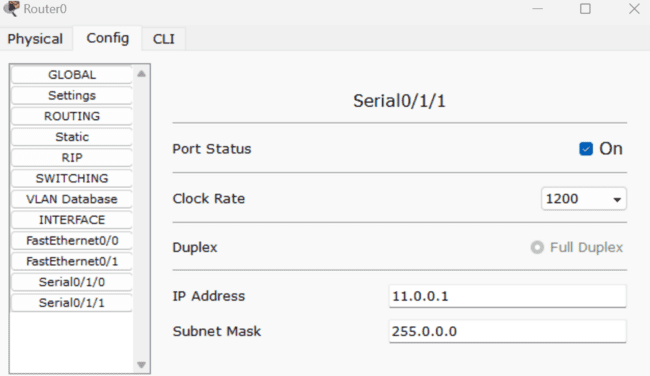
****

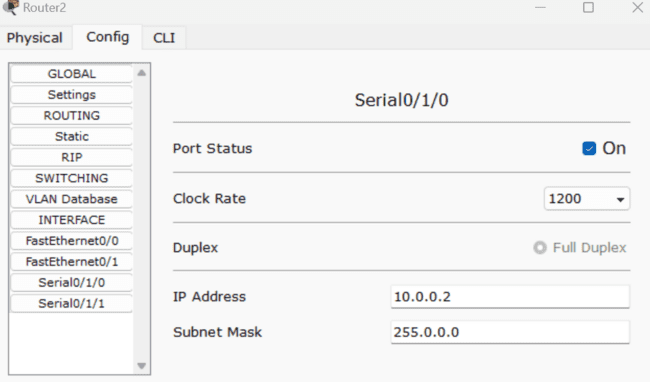
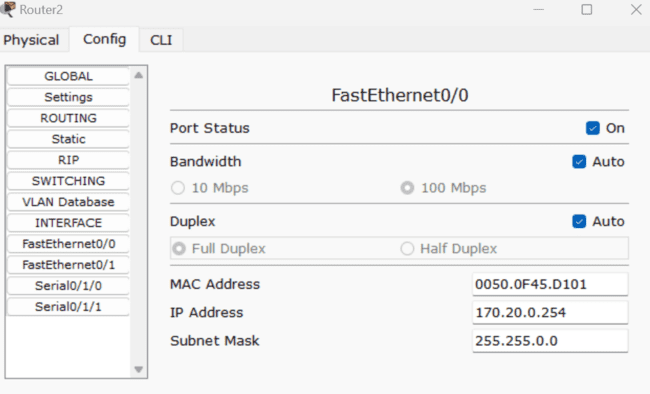


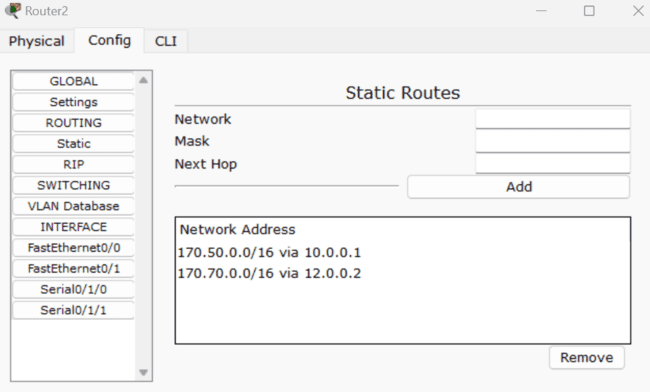
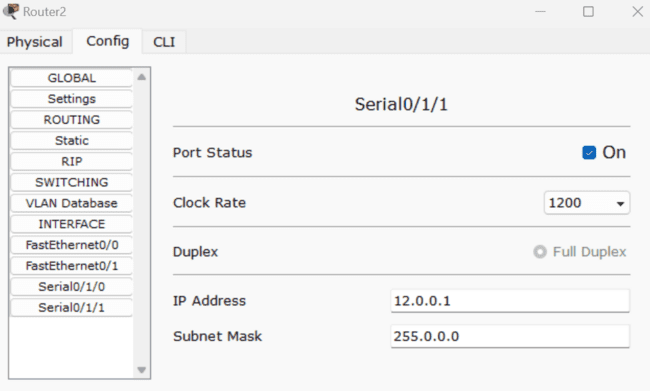


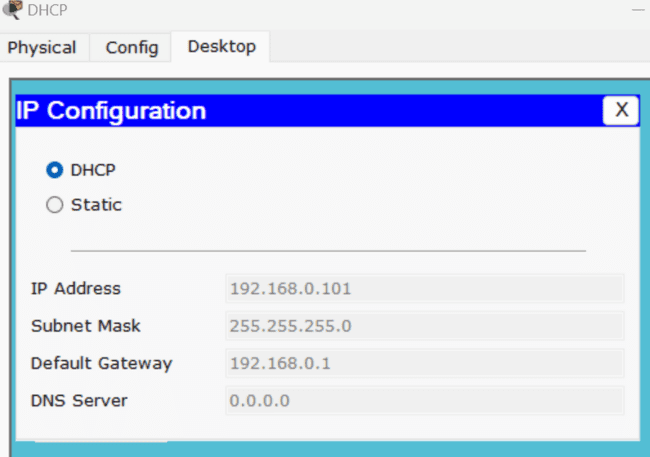
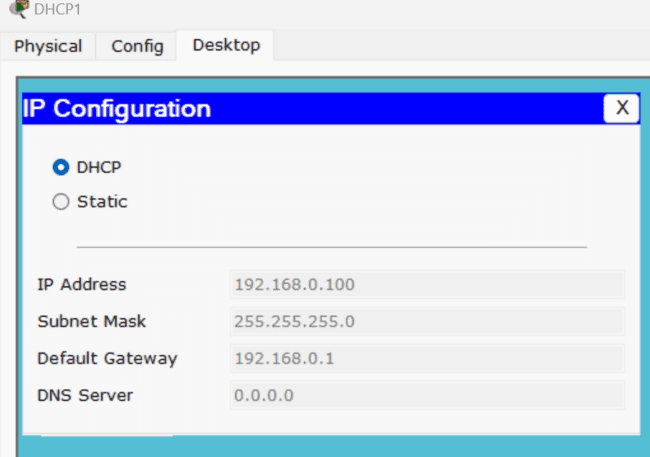


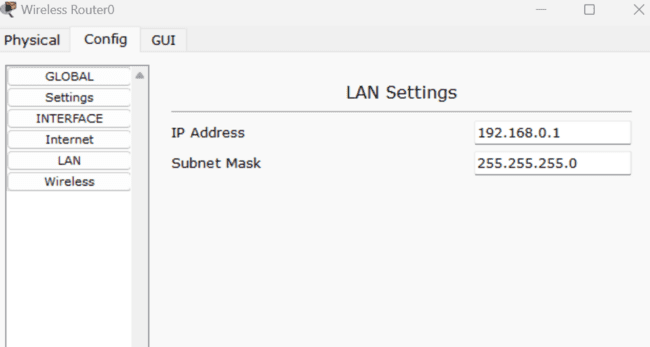
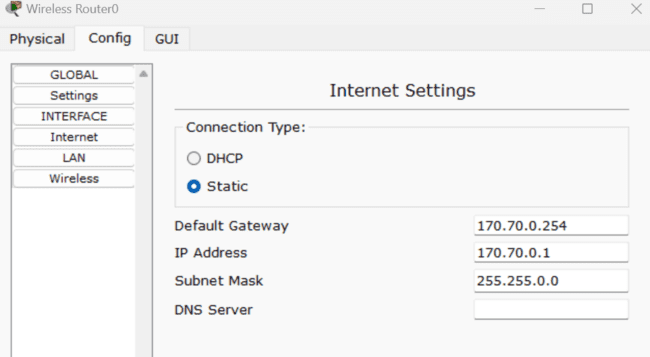


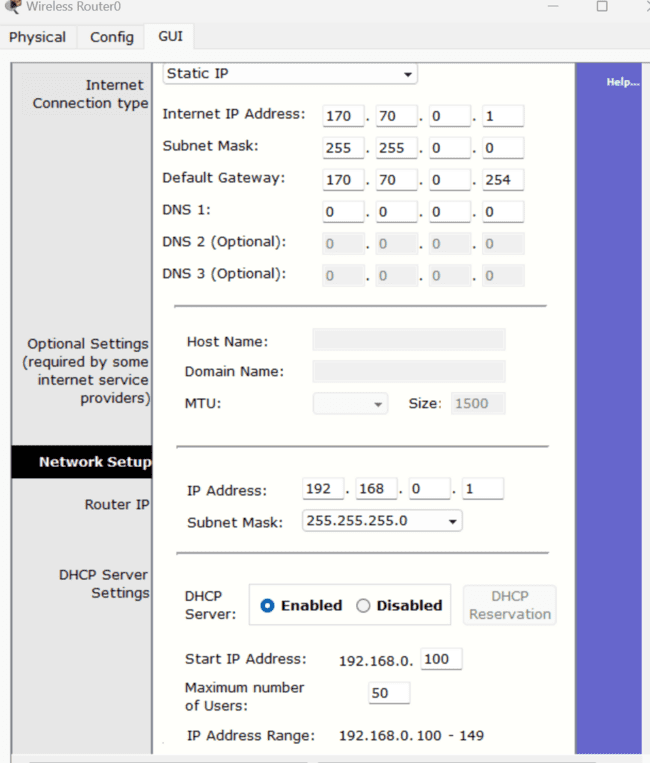


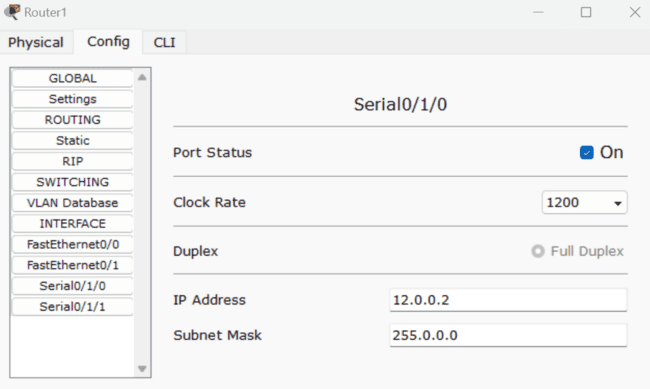
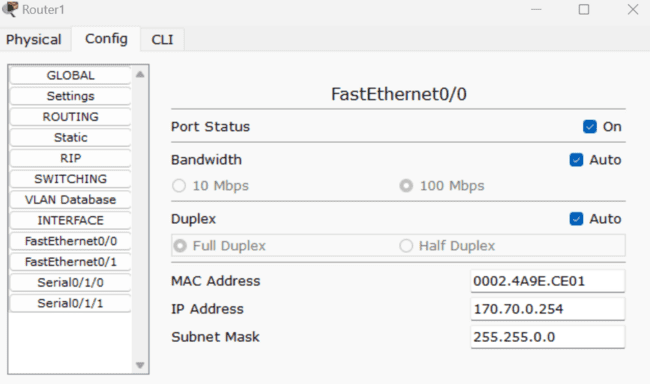


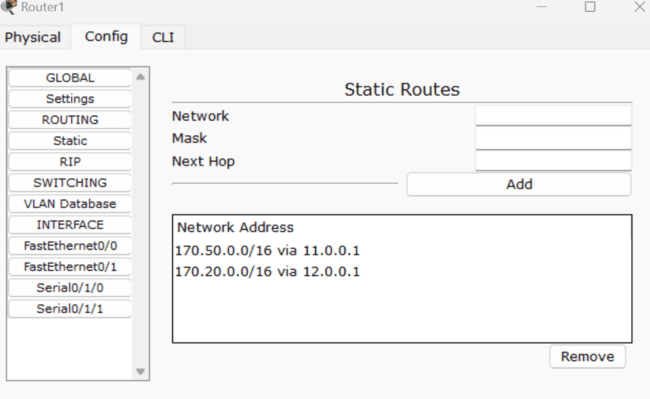
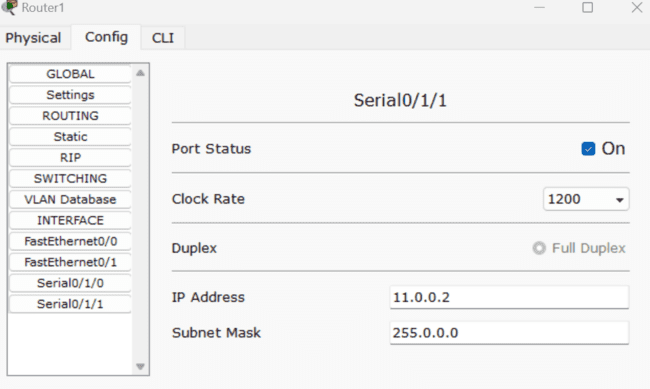


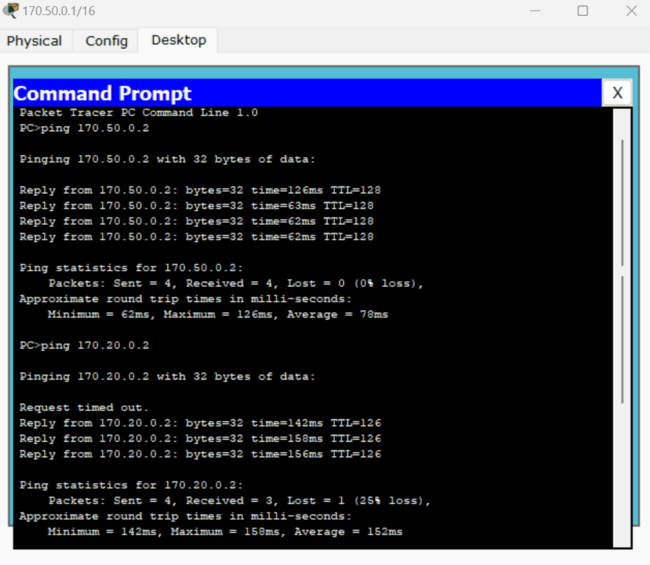


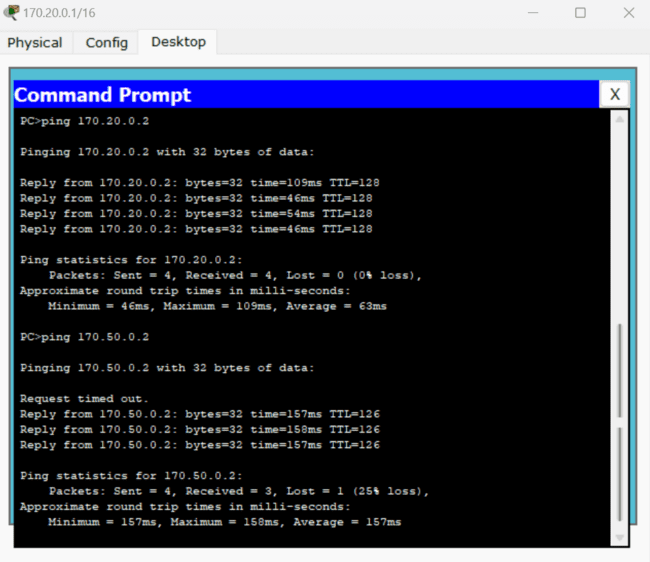


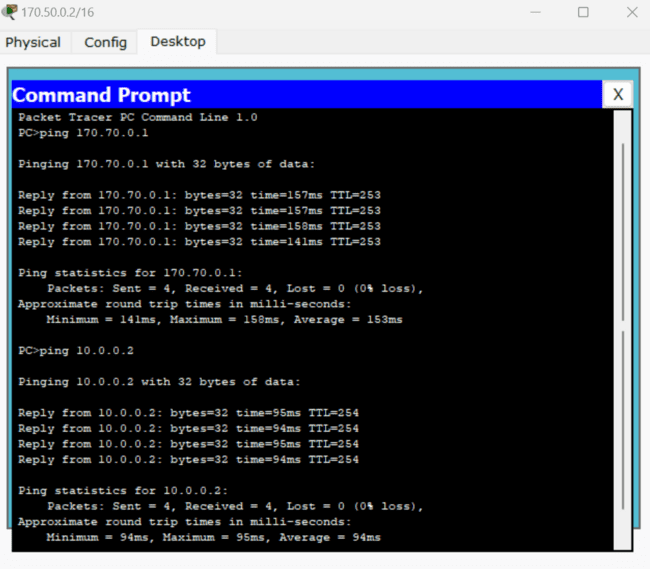


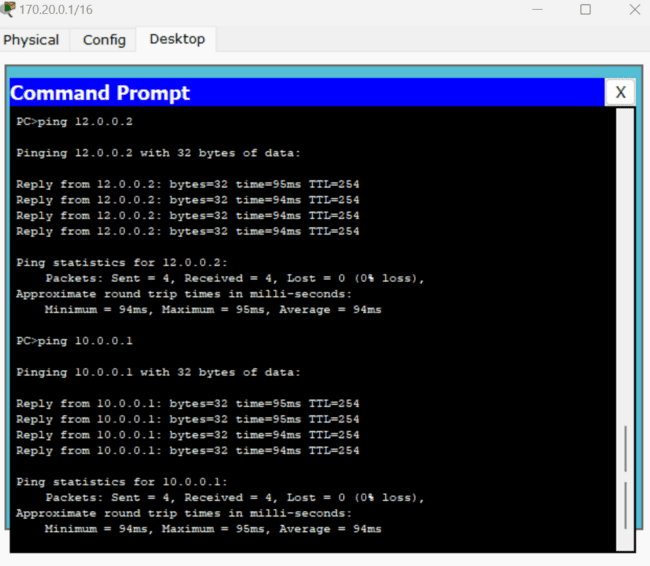


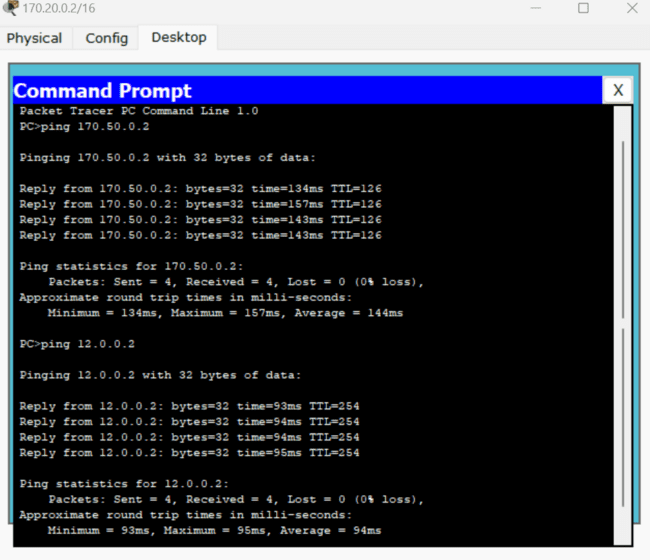












**Result:**

Thus the experiment to show Static Routing and Dynamic Addressing Mechanisms is executed.

**EXPERIMENT-11**

**Aim:**

Show the working of Application Layer Protocols – HTTP, FTP & DNS.

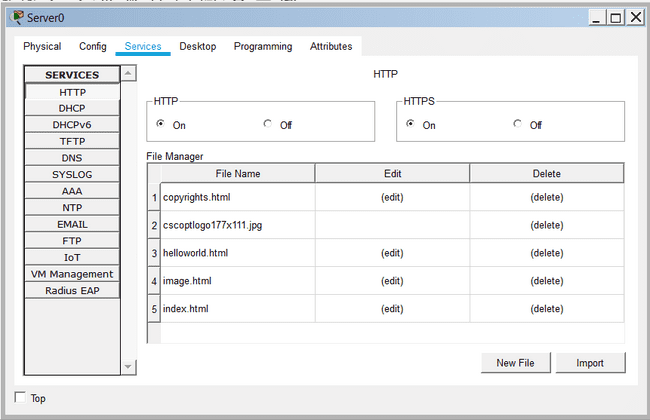
**Description:**

**Hyper Text Transfer Protocol (HTTP):**

Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes. HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response. HTTP is a stateless protocol, meaning that the server does not keep any data (state) between two requests.

Configuration/Implantation of HTTP mechanism in Packet Tracer:

* Configure a simple network and add a server device to the network.
* In the server machine go to services and select HTTP.
* Configure the settings as shown below.
* Now using browser on the remote device check for the web page by clicking on the IP address in the address bar.

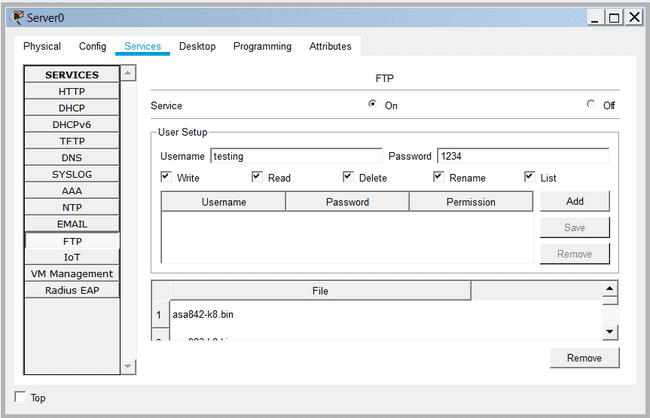


**File Transfer Protocol (FTP):**

The File Transfer Protocol (FTP) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network. FTP is built on a client–server model architecture using separate control and data connections between the client and the server.

Configuration/Implantation of FTP mechanism in Packet Tracer:

* Configure a simple network and add a server device to the network.
* In the server machine go to services and select FTP.
* Configure the settings as shown below.
  + Enable FTP service
  + Assign Username & Password
  + Giver Permissions and ADD user
* Now open the terminal in the FTP Server. And login into the server using ‘ftp’ command. Check for the files present.
* From a remote device you can login using the command ‘ftp <IP\_address>’ and access the files using the basic commands of ftp.

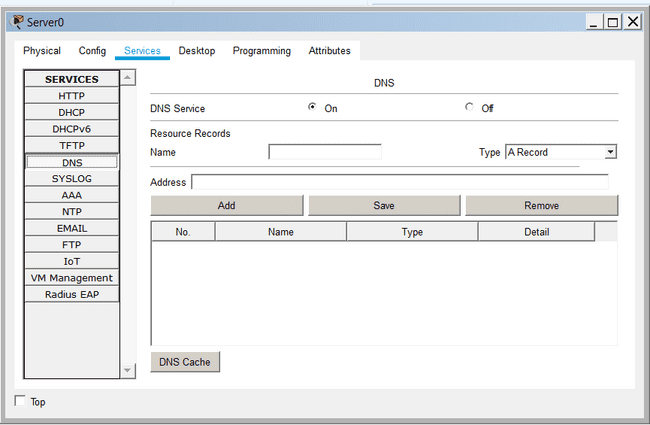


**Domain Name System (DNS):**

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols.

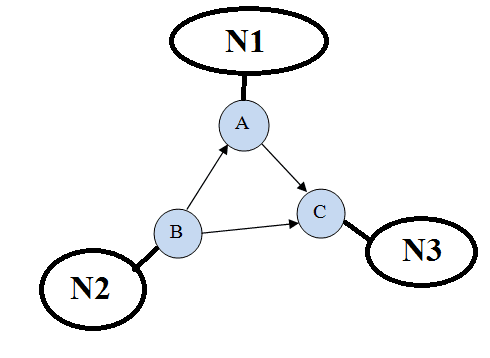
Configuration/Implantation of DNS mechanism in Packet Tracer:

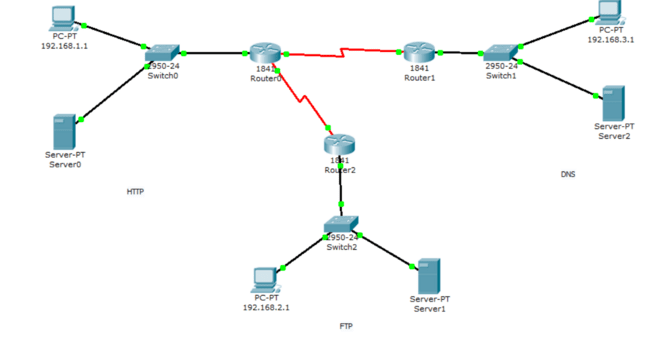
* Configure a simple network and add a server device to the network.
* In the server machine go to services and select DNS.
* Configure the settings as shown below. Give a Name to every PC in the network add Add the resource record.
* Test domain name – IP resolution. Ping the hosts from one another using their names instead of their IP addresses. If the DNS service is turned on and all IP configurations are okay, then ping should work.



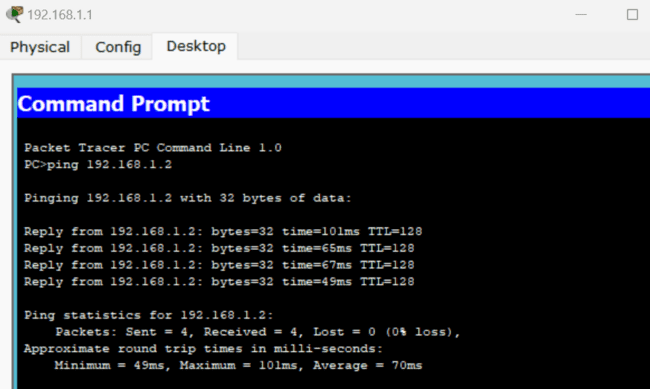
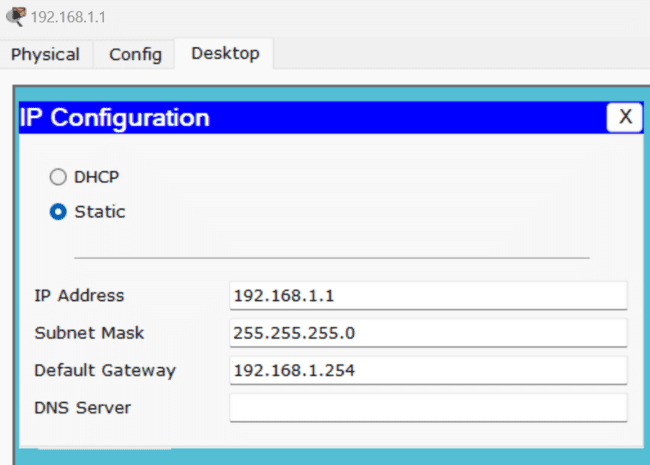
**Task:**

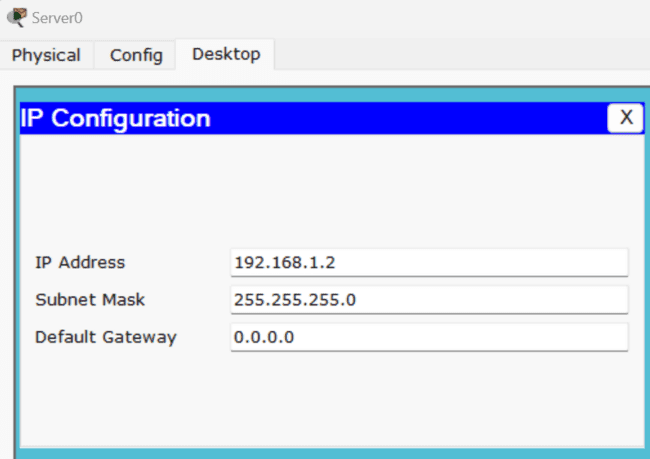
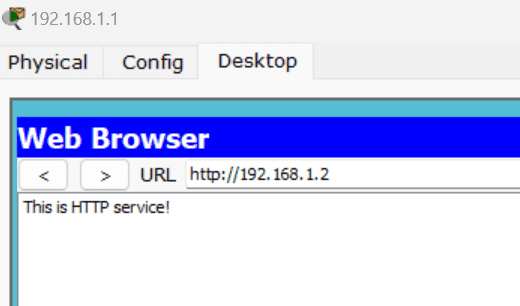
* Design a network for the topology given below and implement Application Layer Services such as HTTP, FTP & DNS. Verify the connectivity between the devices of networks and their services offered.
  + A, B, C are Routers
  + N1, N2, N3 are the networks with one server and one PC’s.
  + N1 contains HTTP server, N2 contains FTP server & N3 contains DNS server.

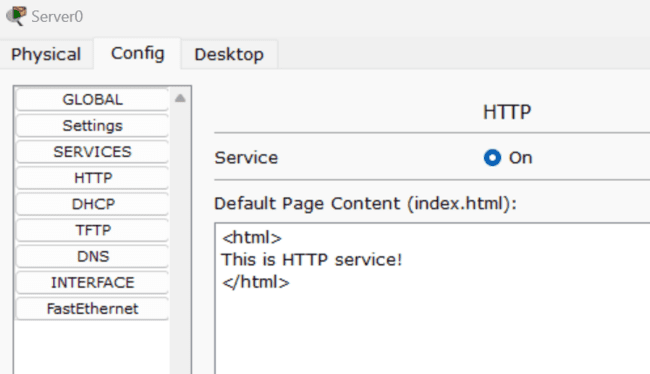
****

****

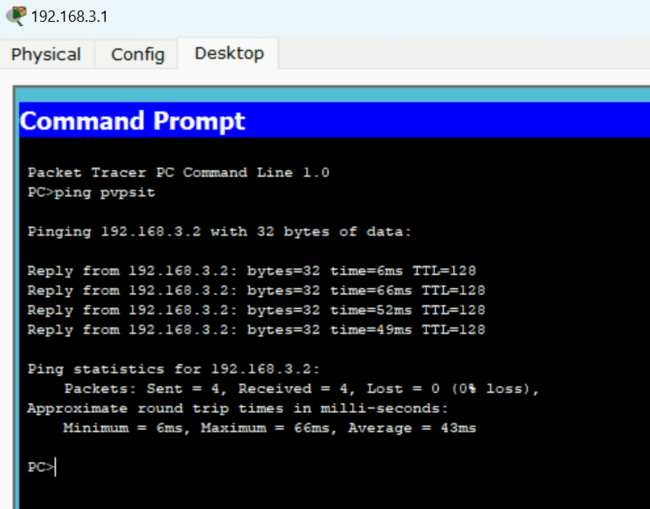
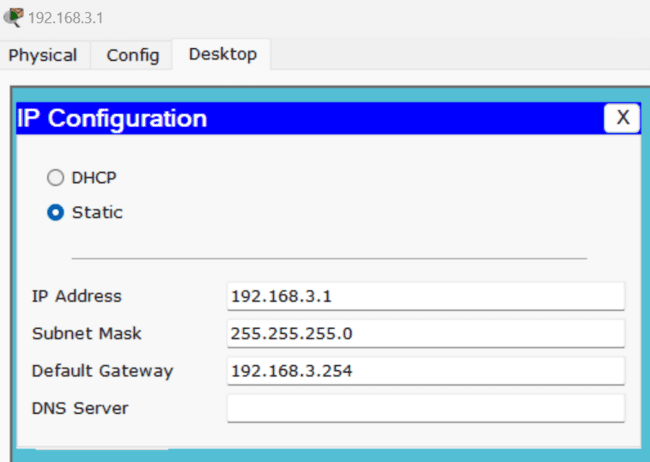
**HTTP:**

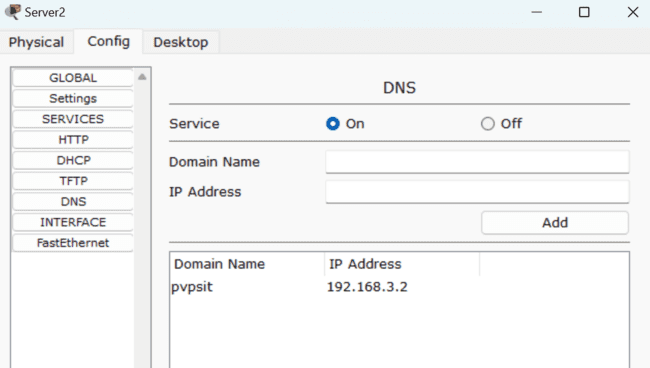
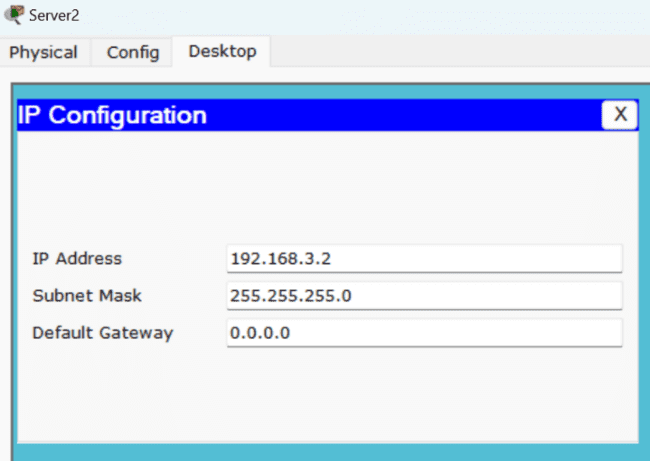
****

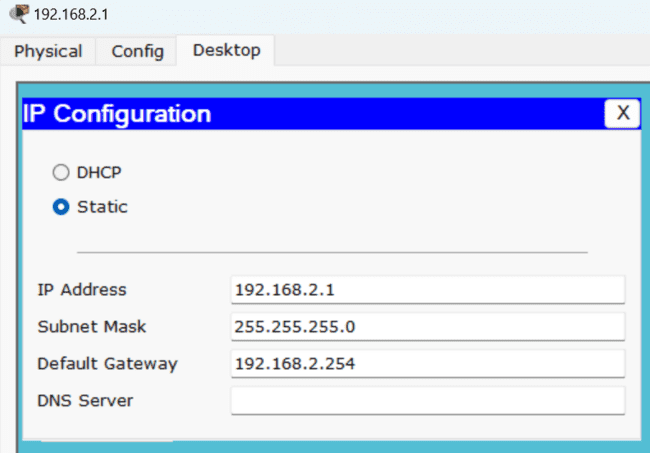
****

****

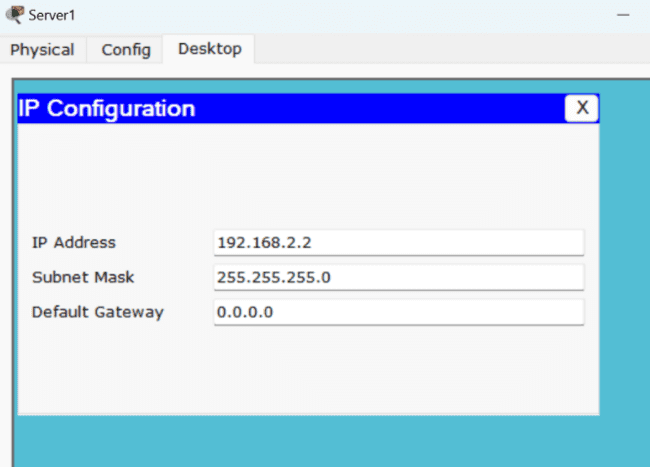
**DNS:**

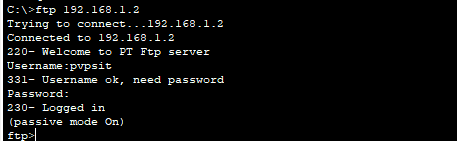
****

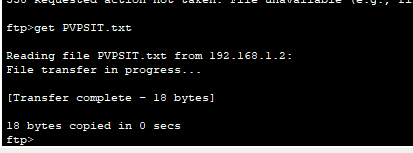
****

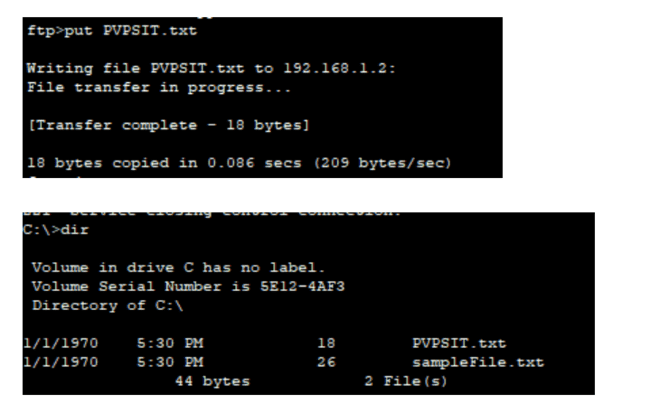
****

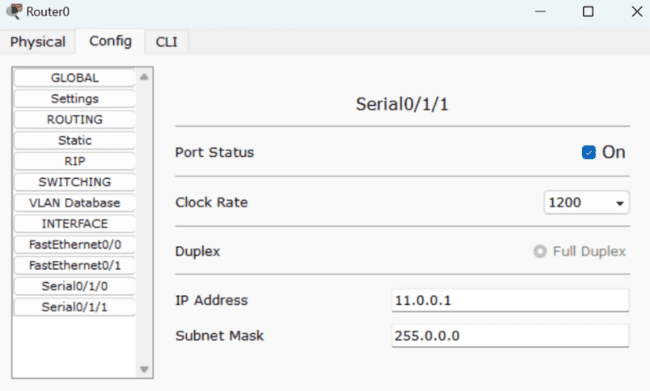
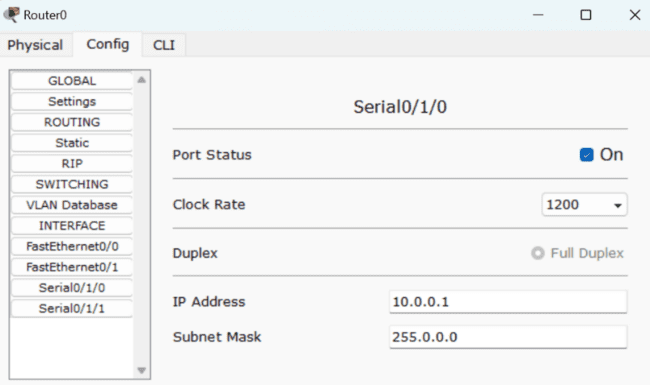
**FTP:**

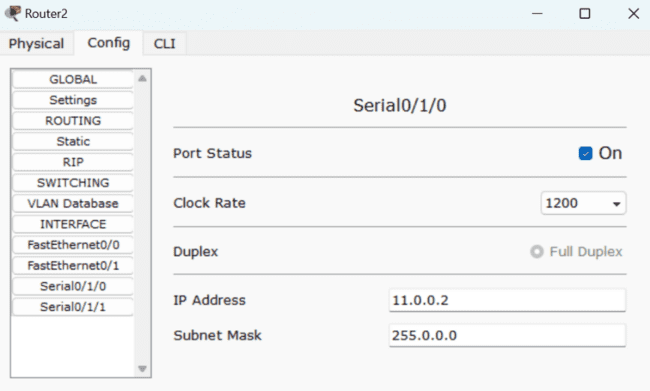
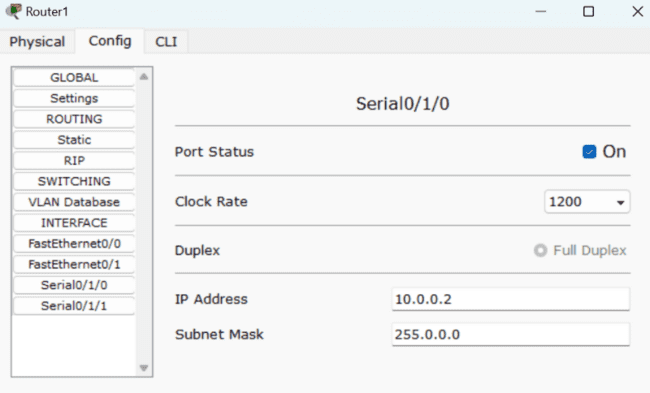
****

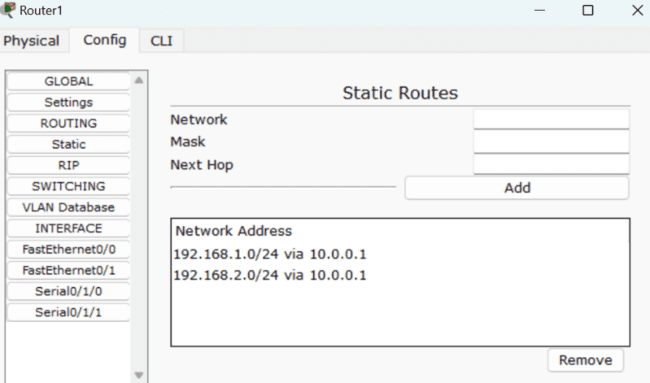
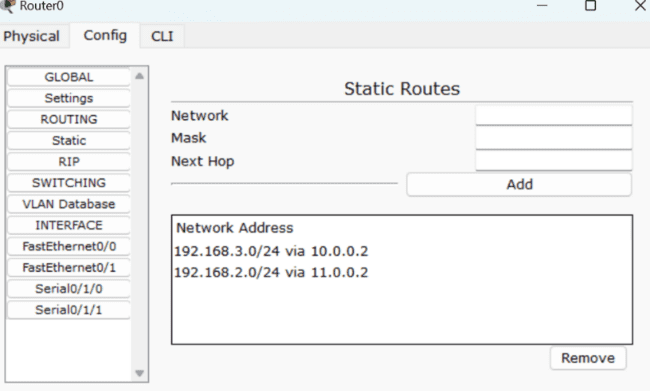
****

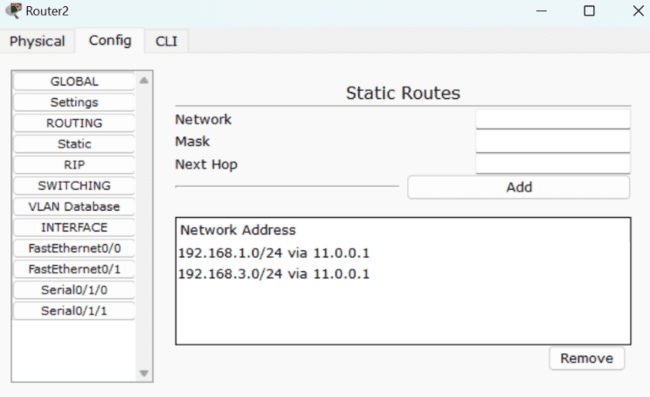
****

****

****

****

****

****

**Result:**

Thus the experiment to show working of Application Layer Protocols is executed.

**EXPERIMENT-12**

**Aim:**

Case study: Analyze, Design and build a network for an organization using Network Simulation tool.

**Task:**

This case study analyzes the network infrastructure of PVP Siddhartha Institute of Technology (PVPSIT), a technical education institute. The institute has provided you with a short description of the current situation and its plans. As a network designer, it is your job to identify all the organization's requirements and data that will allow you to provide an effective solution.

Organizational Facts

PVPSIT is a medium-sized organization located in Vijayawada, with approximately 250 staff members supporting up to 1000 students in each department. The institute is interested in updating its main facility (which uses equipment from various vendors) in its Layer 2 campus. You are meeting to define the client's requirements.

PVPSIT has 8 departments in total on the campus, plus 5 sections for administration. The organization is comprised of a main building having three floors on both sides, with four wiring closets per floor.

The both sides of the main building are connected to the central network administrative (CNA) room; the switches from these three floors are connected with fiber connections to the CNA.

The Computer Science Department has three corridors, with three wiring closets per corridor. The three corridors totally comprise of with 4 laboratories, 10 class rooms, and 15 staff rooms.

The network architect is new to the organization. The institute is aggressively expanding its network architecture. Due to student growth and online teaching in general, plans to enlarge the main campus are also under way. The organization is doing fairly well financially.

It wants to selectively deploy cutting-edge technology for better connectivity and performance. Management is tired of network downtime and slowness affecting teaching process. Network manageability is important because PVPSIT has a tradition of basing operations on small support staffs with high productivity. PVPSIT wants to upgrade in urgent need for a single department of computer science initially.

Current Situation

The current network uses inexpensive switches from several vendors, purchased over time. They comply with various standards, depending on when they were purchased. The switches are not manageable.

Within each of the three department corridors is a corridor switch. One main switch from each corridor connects to the main switch in CNA. The other switches connect either directly to the each floor switch or via a daisy chain of switches, depending on which was most convenient at the time.

Currently, the LAN spans the entire campus. No Layer 3 switching is present. The address space is 172.16.0.0 /16. Addresses are coded sequentially into PCs as they are deployed. Staff members have been meaning to deploy DHCP but have not had the time.

The applications that the organization is currently running include standard office applications, plus some specialized tools running over IP in each lab. Web server, file server, and other servers are present in the CNA room located in computer science department. All the new servers are capable of using Gigabit or Gigabit Ether Channel connectivity.

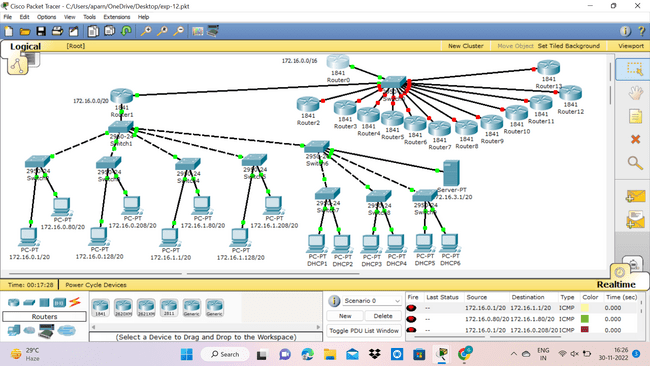
The WAN uses 56-kbps links to three floors of the college. The one router uses static routing that was configured by a previous network designer.

The staff members have frequently complained about slow response times. There appears to be severe congestion of the LAN, especially at peak hours.

Each Lab in the computer science department has 80 computers. All staff rooms are to be equipped with Static LAN Connections and Wireless connectivity. The class rooms are to be connected wirelessly comprising of 80 students. The web server has a public IP. The file server has local IP within the department.

You task is to analyze, document the requirements and to reconfigure the entire network of the computer science department keeping in view of the organization and existing infra-structure in Packet Tracer.

Note: Use Addressing Mechanism, Proper Devices, Connectivity etc. while designing the network.



**Result:**

Thus the case study / experiment to design a network based on the requirements is executed.