Computer Networks

Experiment - 1

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim: To study various networking devices and networking topologies

Networking Devices:

Definition:

The devices which are used for communication between different hardware used in the computer network are known as network devices. These devices are also known as physical devices, networking hardware, and network equipment otherwise computer networking devices. In a computer network, each network device plays a key role based on their functionality, and also works for different purposes at different segments.

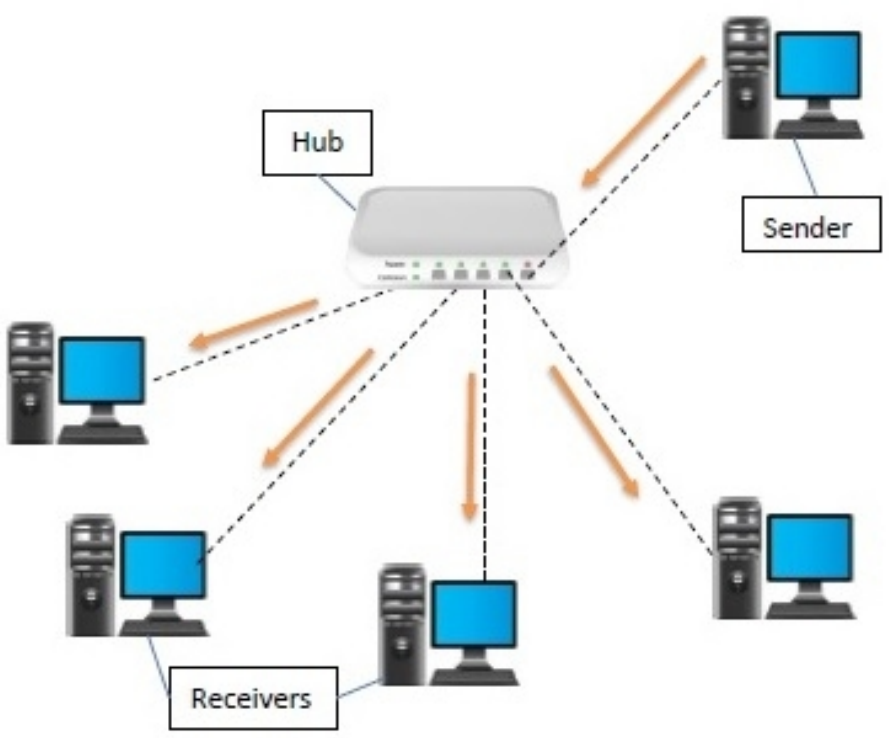
Types of Network Devices

There are different types of network devices used in a computer network which include the following.

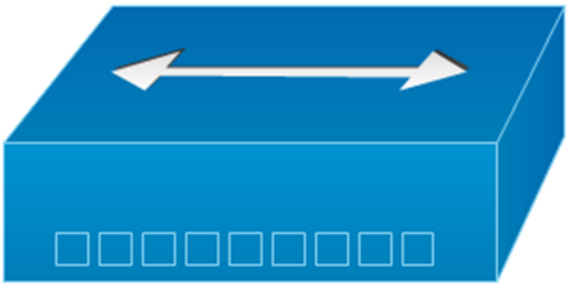
- Network Hub
- Network Switch
- Network Router
- Gateway
- Bridge
- Repeater

1) Network Hub: The network hub is one kind of networking device in a computer network, used to communicate with various network hosts and also for data transferring. The transferring of data in a computer network can be done in the form of packets. Whenever the data processing can be done from a host to a network hub, then the data can transmit to all the connected ports. Similarly, all the ports identify the data path which leads to inefficiencies & wastage. Because of this working, a network hub cannot be so safe and secure. In addition, copying the data packets on all the ports will make the hub slower which leads to the utilise of the network switch

Diagram:



Logical Symbol:



Working:

When a hub receives a packet of data at one of its ports from a network device, it transmits (repeats) the packet to all of its ports to all of the other network devices. If two network devices on the same network try to send packets at the same time a collision is said to occur.

Hubs operate in such a way that all data received through one port is sent to all other ports. This type of operation creates an extremely insecure environment and anyone can sniff the network using a sniffer and any unencrypted traffic over the network is not secure. Hubs are unsecure LAN devices that should be replaced with switches for security and increased bandwidth.

Advantages of Hub

- It provides support for different types of Network Media.
- It can be used by anyone as it is very cheap.
- It can easily connect many different media types.
- The use of a hub does not impact on the network performance.
- Additionally, it can expand the total distance of the network.

Disadvantages of Hub

- It has no ability to choose the best path of the network.
- It does not include mechanisms such as collision detection.
- It does not operate in full-duplex mode and cannot be divided into the Segment.
- It cannot reduce the network traffic as it has no mechanism.
- It is not able to filter the information as it transmits packets to all the connected segments.
- Furthermore, it is not capable of connecting various network architectures like a ring, token, and ethernet, and more

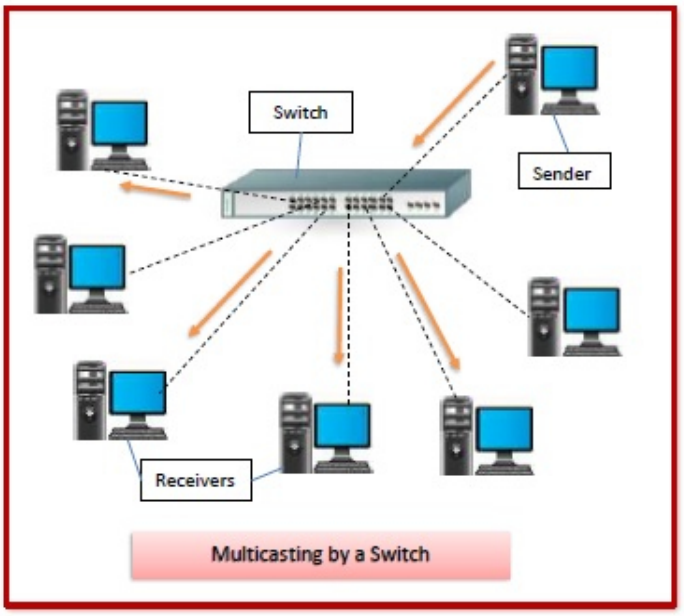
Applications:

- Hub is used to create small home networks.
- It is used for network monitoring.
- They are also used in organisations to provide connectivity.
- It can be used to create a device that is available throughout the network.

2) Switch :

Introduction: When a user accesses the internet or another computer network outside their immediate location, messages are sent through the network of transmission media. This technique of transferring the information from one computer network to another network is known as switching.

Diagram:



Logical Symbol:

Working of Switch:

When the source wants to send the data packet to the destination, the packet first enters the switch and the switch reads its header and finds the MAC address of destination to identify the device then it sends the packet out through the appropriate ports that leads to the destination devices.

Switch establishes a temporary connection between source and destination for communication and terminates the connection once conversation is done. Also, it offers full bandwidth to network traffic going to and from a device at the same time to reduce collision.

Switching techniques are used to decide the best route for data transmission between source and destination.

Advantages of Switching:

- Switch increases the bandwidth of the network.
- It reduces the workload on individual PCs as it sends the information to only that device which has been addressed.
- It increases the overall performance of the network by reducing the traffic on the network.
- There will be less frame collision as the switch creates the collision domain for each connection.

Disadvantages of Switching:

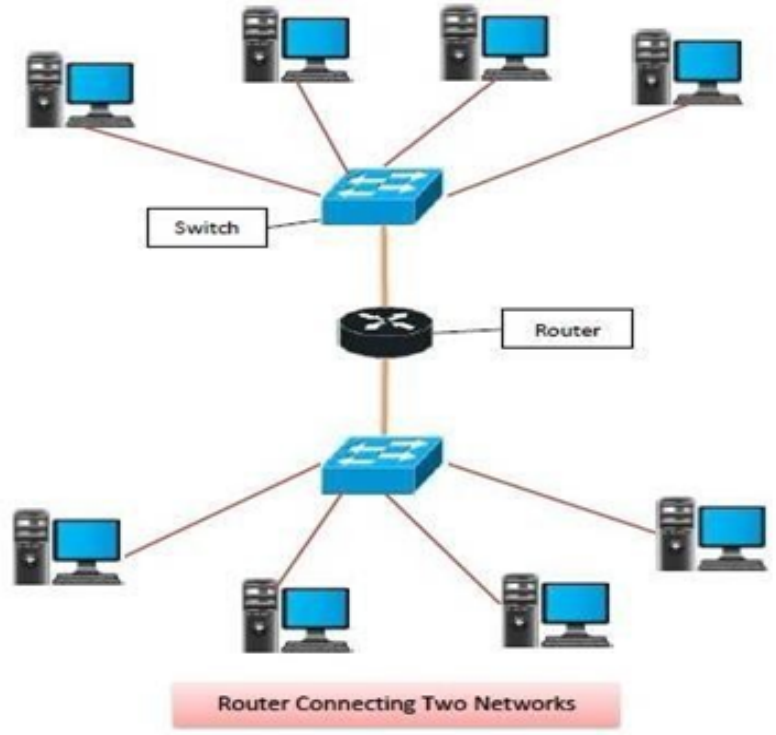
- A Switch is more expensive than network bridges.
- A Switch cannot determine the network connectivity issues easily.
- Proper designing and configuration of the switch are required to handle multicast packets.

3) Router

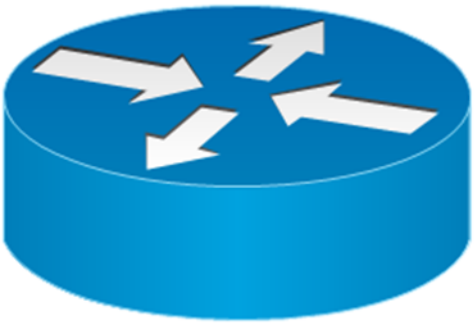
A router is a device that connects two or more packet-switched networks or subnetworks. It serves two primary functions: managing traffic between these networks by forwarding data packets to their intended IP

addresses, and allowing multiple devices to use the same Internet connection.

Diagram:



Logical Symbol:



Working:

How a router works

A router examines a packet header's destination IP address and compares it against a routing table to determine the packet's best next hop. Routing tables list directions for forwarding data to particular network destinations, sometimes in the context of other variables, like cost. They amount to an algorithmic set of rules that calculate the best way to transmit traffic toward any given IP address.

A routing table often specifies a default route, which the router uses whenever it fails to find a better forwarding option for a given packet. For example, the typical home office router directs all outbound traffic along a single default route to its internet service provider (ISP).

Routing tables can be static -- i.e., manually configured -- or dynamic. Dynamic routers automatically update their routing tables based on network activity, exchanging information with other devices via routing protocols.

Many routers also perform network address translation (NAT), shielding the private IP addresses of a local area network (LAN) by readdressing all outgoing traffic with a single shared public IP address. NAT helps both conserve globally valid IP addresses and improve network security.

Advantages of Routers:

1) It provides connection between different network architectures such as ethernet & token ring etc.

2)It can choose the best path across the internetwork using dynamic routing algorithms.

3)It can reduce network traffic by creating collision domains and also by creating broadcast domains.

4)It provides sophisticated routing, flow control and traffic isolation.

5)They are configurable which allows network manager to make policy based on routing decisions.

6)Drawbacks or disadvantages of Routers

Disadvantages of Routers:

1) They operate based on routable network protocols.

2)They are expensive compared to other network devices.

3) Dynamic router communications can cause additional 4)network overhead. This results in less bandwidth for user data.

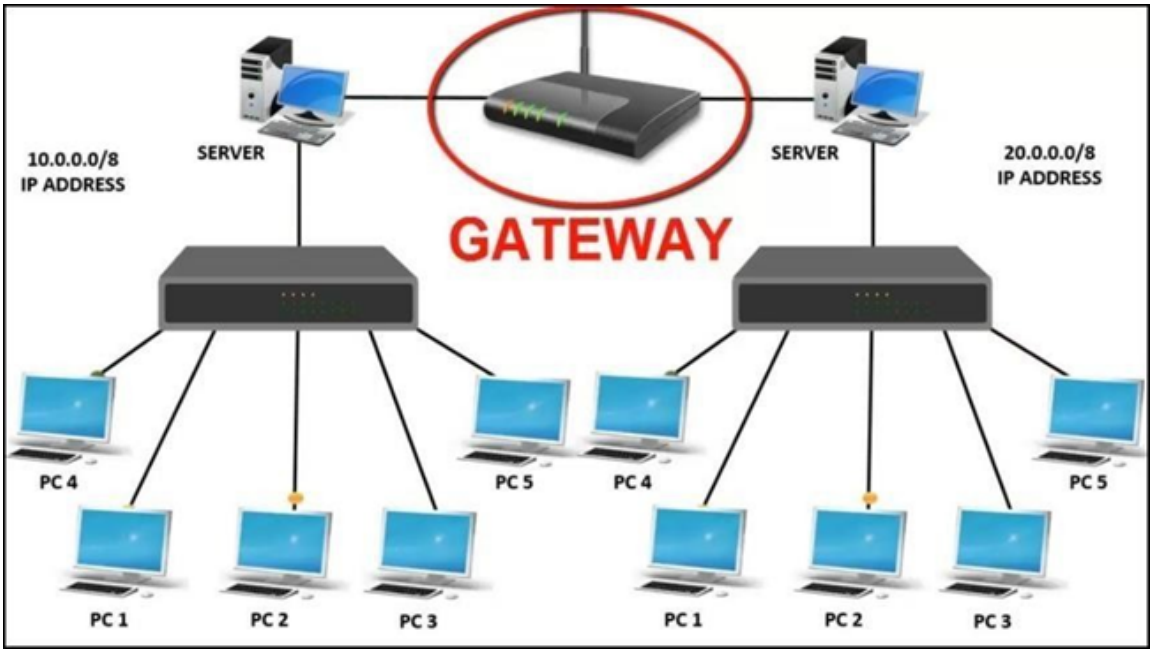
5)They are slower as they need to analyse data from layer-1 through layer-3.

6)They require a considerable amount of initial configurations.

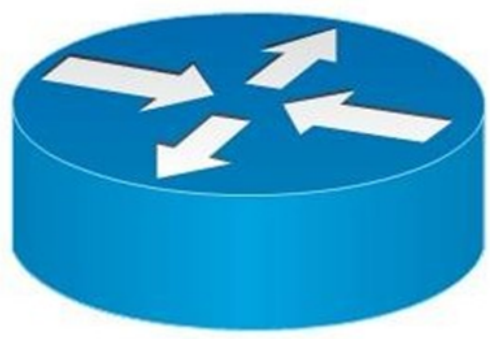
7) They are protocol dependent devices which must understand the protocol they are forwarding.

4) Gateway: A gateway is simply a device or hardware that acts as a gate between the networks. We can also define it as a node that acts as an entry for other network nodes. It is also responsible for facilitating the traffic flow within the network. Gateway uses more than one communication protocol, so its activities are more complicated than a router or a switch.

Diagram:



Symbol:



Working:

It is a point of a network that can access other networks. Usually, in the intranet, a router or node can act as a gateway node or the router that links the networks are called gateways. In large scale enterprises, the computers manage the traffic between enterprise networks are termed as gateway nodes. Such as that the computers used by Internet service providers to link varied users to each other at an instant time to the internet are gateway nodes. In any development team of any commercial enterprise computer server functions as gateway nodes and it may also be a proxy server or a firewall at times.

It can be linked-to router since a router accurately knows about the routing path of data packets that appears at gateway then a switch decides on the suitable in and out the path of the gateway for the designated packet. The gateway is a mandatory attribute of routes even though the other devices can act well as a gateway. But the operating system used here with internet sharing behaves like a gateway and establishes the connection with internal networks.

Advantages :

Connectivity: Gateway provides better connectivity with other different networks, and it also helps to scale the network by linking the multiple computers along with other systems together. So, due to this, different types of computers have the ability to access the same type of information.

Better Flexibility: The gateway is getting more flexibility for your network because it is capable of translating all information from computers along with different kinds of systems.

Protection from Unwanted Users: Gateway allows user authentication for getting to improve the security. In concern of security, user ID and Password are installed on the network gateway, so unwanted users are not able to access any type of information from the network. Due to this security, authorised users have only the right to access all information over the entire network.

Disadvantages:

Configuration: It is getting more difficult for Configuration of devices through a gateway. So, a skilled system administrator is needed for this purpose.

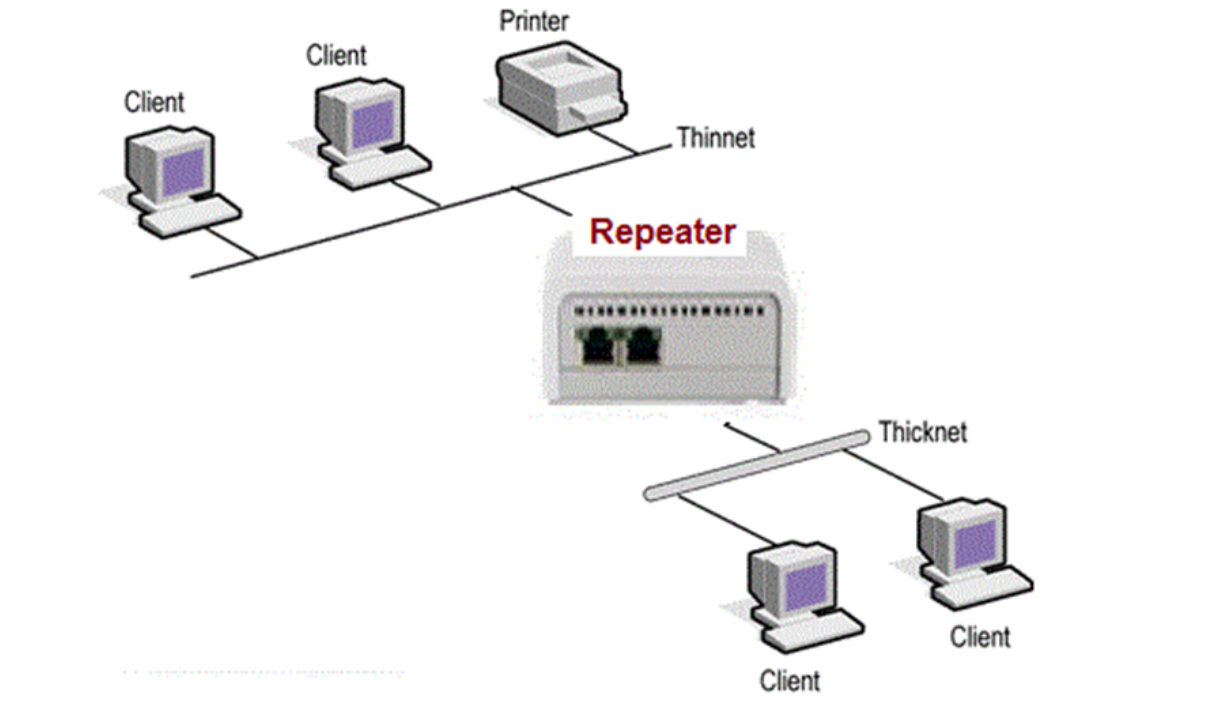
Time Delay: Network gateway takes more time for translating the information, so it cannot instant transfer information. It gets back old cache information which is not cleared. Due to this, it can take more time for producing the accurate result.

Implementation: Oftenally, gateways on default are configured on the router itself. Network administrators are having more difficulties installing them. Due to this, its cost can get higher.

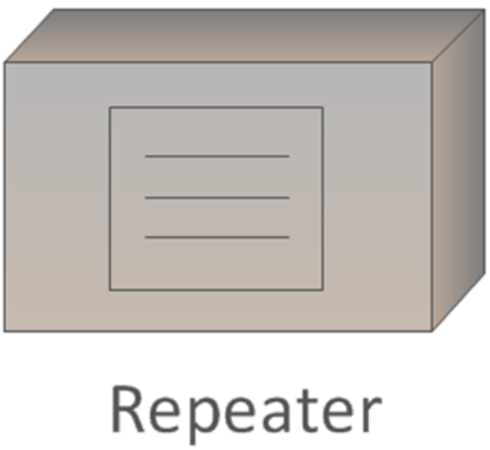
5) Repeater:

Repeaters are network devices operating at the physical layer of the OSI model that amplify or regenerate an incoming signal before retransmitting it. They are incorporated in networks to expand its coverage area. They are also known as signal boosters.

Diagram:



Symbol:



Working :

A repeater is a network device that retransmits a received signal with more power and to an extended geographical or topological network boundary than what would be capable with the original signal.

Advantages of Repeaters:

- Repeaters are simple to install and can easily extend the length or the coverage area of networks.
- They are cost effective.
- Repeaters don't require any processing overhead. The only time they need to be investigated is in case of degradation of performance.
- They can connect signals using different types of cables.

Disadvantages of Repeaters:

- Repeaters cannot connect dissimilar networks.
- They cannot differentiate between actual signal and noise.
- They cannot reduce network traffic or congestion.
- Most networks have limitations upon the number of repeaters that can be deployed.

6) Bridge:

Bridges are used to connect two subnetworks that use interchangeable protocols. It combines two LANs to form an extended LAN. The main difference between the bridge and repeater is that the bridge has a penetrating efficiency.

Figure-

Logical Symbol-

Working-

A bridge accepts all the packets and amplifies all of them to the other side. The bridges are intelligent devices that allow the passing of only selective packets from them. A bridge only passes those packets addressed from a node in one network to another node in the other network.

Advantages-

1) **Network Scalable**: Sometimes, bridges play the role as repeaters for scaling the physical network. Bridge allows support for making connections with each other of dissimilar network architectures.
2) **Enhanced Bandwidth**: Some nodes exist on the network that help to share the separate collision domain. So, these bridges are getting to enhance the bandwidth for those individual nodes.
3) **Best Reliability**: Bridges help to provide higher reliability over the entire network that allows easy maintenance of the computer network. Due to small segments of the LAN network, network congestion gets to decrease.

Disadvantages-

1) **Less Speed**: Bridge gets to make a buffer of frames and then introduces more relays. So, this makes them slower than repeaters.
2) **Down Performance**: Bridges get additional processing by monitoring of all MAC addresses. So, due to this, it gets to downgrade the entire performance in networking.
3) **Broadcast Filtering**: Bridges are not capable of individually filter the broadcast traffic. So, they forward broadcast packets.

Applications-

Bridges connect two or more different LANs that has a similar protocol and provides communication between the devices (nodes) in them. By joining multiple LANs, bridges help in multiplying the network capacity of a single LAN. The bridges in networks are used to divide the local area network (LAN) into various segments. These work on the OSI model under the data link layer. These are used to store the MAC address of computers in the network. These are used to reduce the traffic in the network. These are used to filter the content by using the MAC address of the source and destination. Used for interconnecting two LANs using the single and same protocol

Networking Topology

Topology defines the structure of the network of how all the components are interconnected to each other.

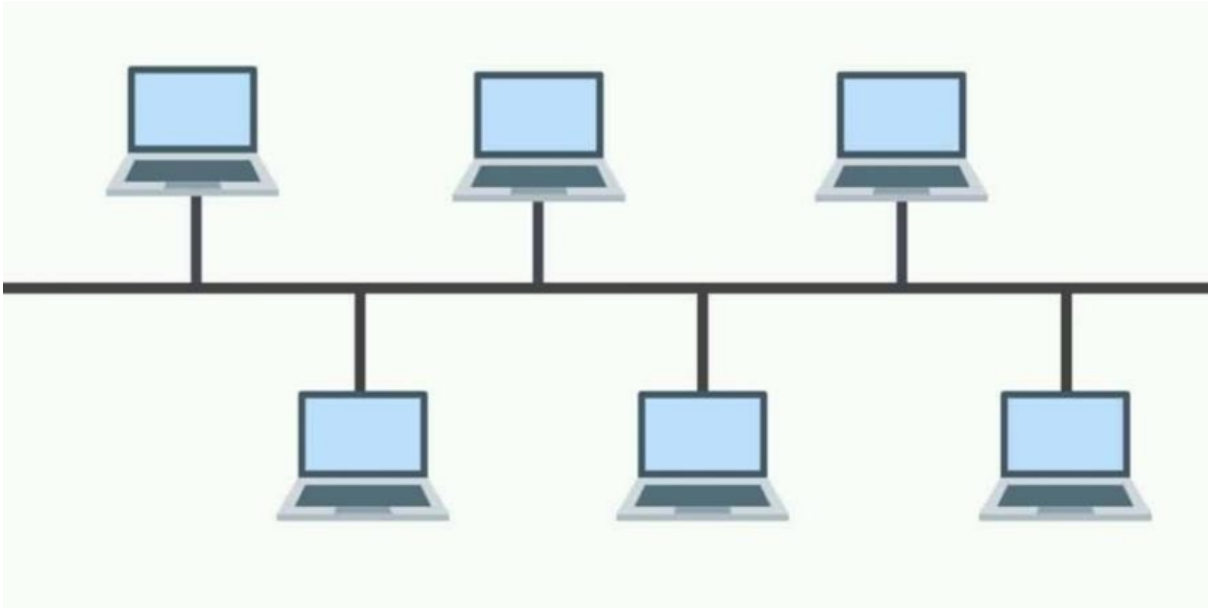
Here is the common topologies list:

1) Bus
2) Star
3) Mesh
4) Tree
5) Hybrid
6) Ring

1)Bus-

Introduction-

Bus topology is a network type in which every computer and network device is connected to a single cable. It transmits the data from one end to another in a single direction. No bi-directional feature is in bus topology. It is a multi-point connection and a non-robust topology because if the backbone fails the topology crashes.



Architecture-

Bus topology uses a single cable which connects all the included nodes. The main cable acts as a spine for the entire network. One of the computers in the network acts as the computer server. When it has two endpoints, it is known as a linear bus topology.

Applications-

1) Small workgroup local area networks (LANs) whose computers are connected using a thinnet cable.
2) Trunk cables connecting hubs or switches of departmental LANs to form a larger LAN.
3) Backboning, by joining switches and routers to form campus-wide networks.

Advantages-

1) If N devices are connected to each other in a bus topology, then the number of cables required to connect them is 1, which is known as backbone cable, and N drop lines are required.
2) If N devices are connected to each other in a bus topology, then the number of cables required to connect them is 1, which is known as backbone cable, and N drop lines are required.

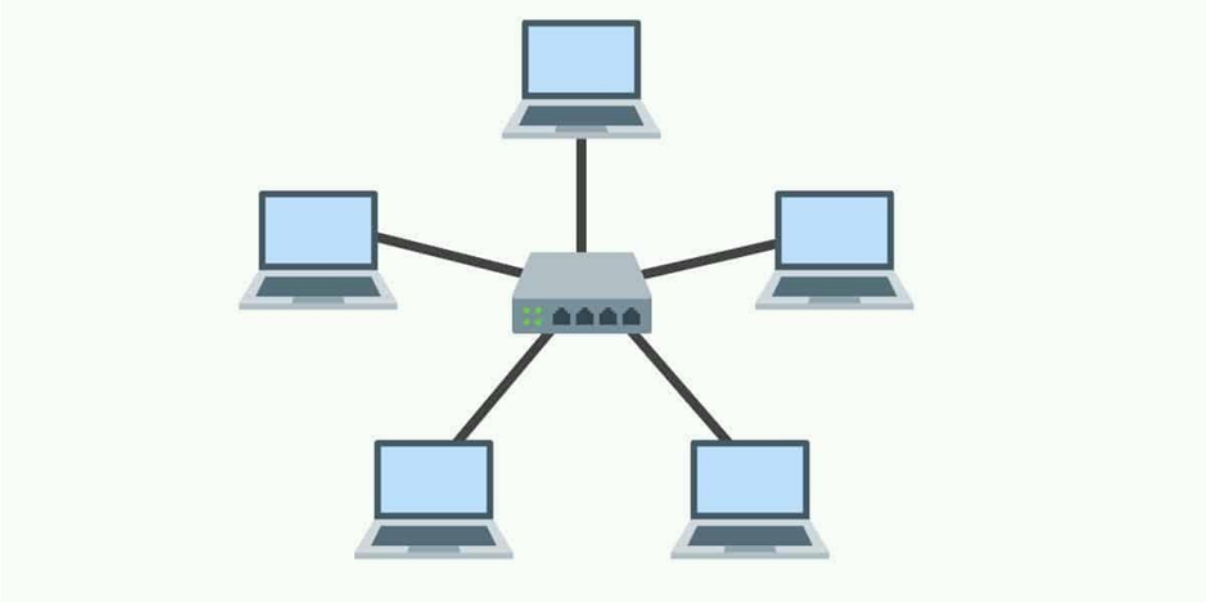
Disadvantages-

1) Security is very low.
2) If the network traffic is heavy, it increases collisions in the network. To avoid this, various protocols are used in the MAC layer known as Pure Aloha, Slotted Aloha, CSMA/CD, etc.
3) If the common cable fails, then the whole system will crash down.

2)Star-

Introduction-

Star topology is a network topology in which each network component is physically connected to a central node such as a router, hub or switch.



Architecture-

In star topology, all the devices are connected to a single hub through a cable. This hub is the central node and all other nodes are connected to the central node. The hub can be passive in nature i.e., not an intelligent hub such as broadcasting devices, at the same time the hub can be intelligent known as an active hub.

Applications-

1) They use LAN connections for high speed up to 100MBPS.
2) Star topology is used in small networks.
3) They are very easy in maintenance and up gradation.

Advantages-

1) If N devices are connected to each other in a star topology, then the number of cables required to connect them is N. So, it is easy to set up.

2) Each device requires only 1 port i.e. to connect to the hub, therefore the total number of ports required is N.

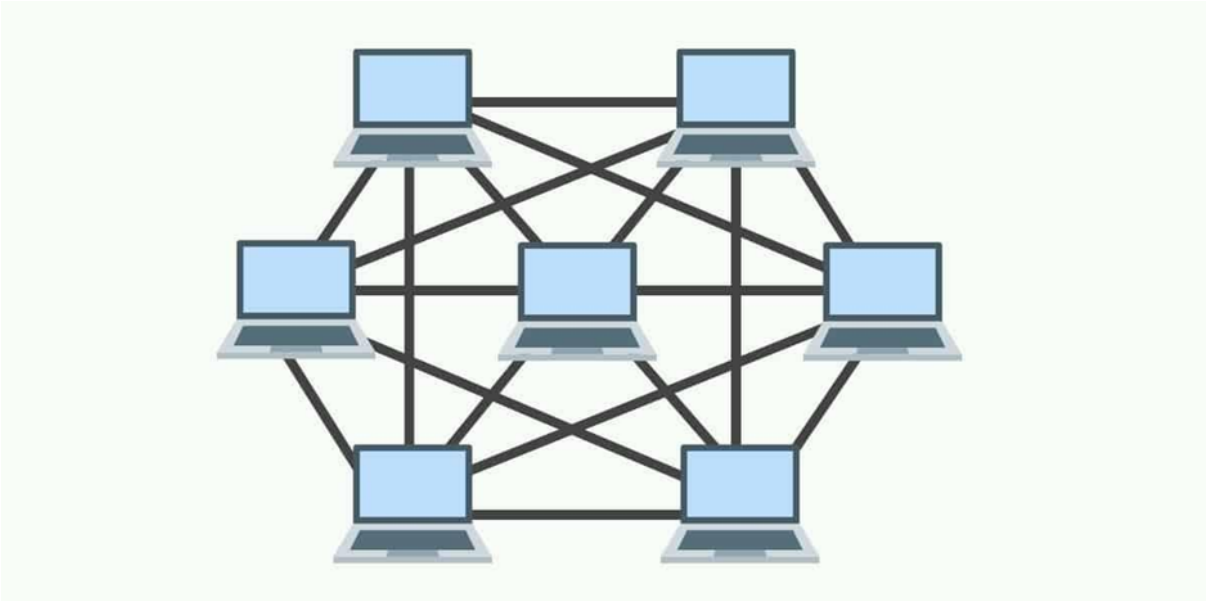
Disadvantages-

1) If the concentrator (hub) on which the whole topology relies fails, the whole system will crash down.
2) The cost of installation is high.
3) Performance is based on the single concentrator i.e. hub.

3)Mesh-

Introduction-

In a mesh topology there is no central connection point. Instead, each node is connected to at least one other node and usually to more than one. Each node is capable of sending messages to and receiving messages from other nodes. The nodes act as relays, passing on a message towards its final destination.



Architecture-

In Mesh Topology, the connections between devices take place randomly. The connected nodes can be computers, switches, hubs, or

any other devices. In this topology setup, even if one of the connections goes down, it allows other nodes to be distributed.

Applications-

1) Home monitoring and control: It's a snap to turn lights off and on or dim them.
2) Building monitoring and control: Monitoring and controlling lights, HVAC, and other functions in large office buildings, hotels, hospitals, and other structures can yield huge energy savings.
3) Military communications and reconnaissance: A mesh makes soldier-to-soldier communications more reliable with longer range. Meshes also help tie together and coordinate many weapons and systems in monitoring and managing the battlefield.

Advantages-

1) Failure during a single device won't break the network.
2) There is no traffic problem as there is a dedicated point to point links for every computer.
3) Fault identification is straightforward.

Disadvantages-

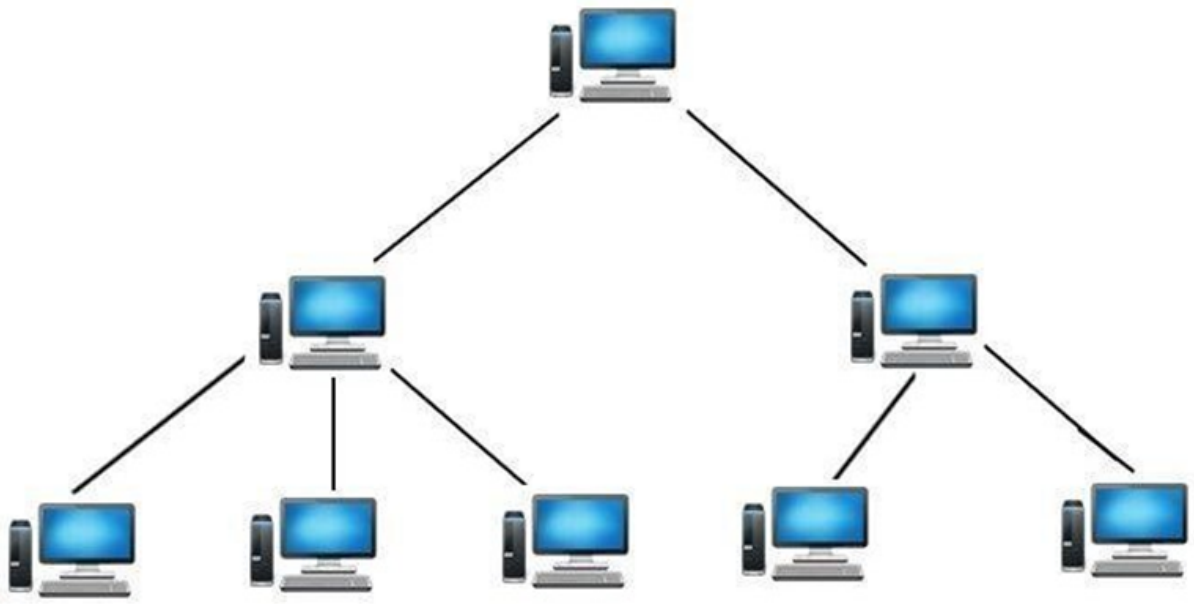
1) It's costly as compared to the opposite network topologies i.e. star, bus, point to point topology.
2) Installation is extremely difficult in the mesh.
3) Power requirement is higher as all the nodes will need to remain active all the time and share the load.
4) Complex process.

4)Tree-

Introduction-

A tree topology, or star-bus topology, is a hybrid network topology in which star networks are interconnected via bus networks. Tree networks

are hierarchical, and each node can have an arbitrary number of child nodes.



Architecture-

In this topology, the various secondary hubs are connected to the central hub which contains the repeater. In this data flow from top to bottom i.e. from the central hub to secondary and then to the devices or from bottom to top i.e. devices to the secondary hub and then to the central hub. It is a multi-point connection and a non-robust topology because if the backbone fails the topology crashes.

Applications-

Tree topologies are commonly used to arrange data in databases and workstations in corporate networks.

Advantages-

1) The advantages of centralization that are achieved in a star topology are inherited by the individual star segments in a tree network.
2) Each star segment gets a dedicated link from the central bus. Thus, failing of one segment does not affect the rest of the network. 3) Fault identification is easy.

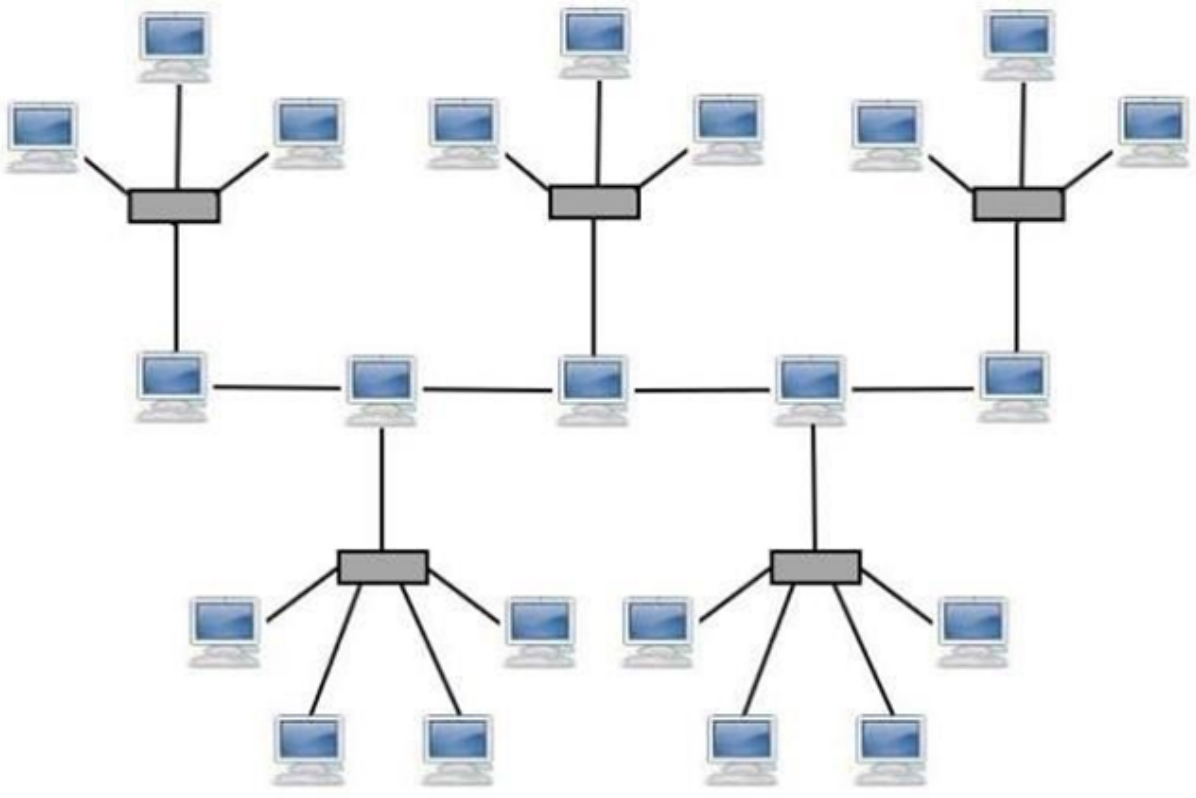
Disadvantages-

1) The advantages of centralization that are achieved in a star topology are inherited by the individual star segments in a tree network.
2) Each star segment gets a dedicated link from the central bus. Thus, failing of one segment does not affect the rest of the network. 3) Fault identification is easy.

5)Hybrid-

Introduction-

Hybrid topology is an integration of two or more different topologies to form a resultant topology which has many advantages (as well as disadvantages) of all the constituent basic topologies rather than having characteristics of one specific topology.



Architecture-

A hybrid cloud network architecture consists of private servers, public cloud virtual servers, and the network that connects them. Public cloud providers typically utilize direct MPLS or Ethernet connections to move data between the client's private cloud and the service provider's public cloud.

Applications-

The hybrid topology is more useful when you need to fulfill diversity in Computer Network. In this topology, all network sections can include the configuration of different Network Topology. For instance, you can have a Hybrid network made by two different networks Star Backbone and the Ring Network.

Advantages-

1) Reliability- Among the networking topologies, the hybrid topology is the most reliable and safe for use. Because of its branching factor, the error detection is very fast in hybrid and for that troubleshooting is very easy.
2) Effectiveness of Networks- Since the combination of various topologies makes the hybrid structure more effective, the overall effectiveness is improved greatly that not only enhances the strengths of the networks but also neutralizes the weak networks of different topologies.
3) Flexibility- Hybrid topology offers great flexibility in usage since the overall configurations and modifications can be planned and designed according to the requirements of the users and the organizations that optimize the overall resources of the networks.

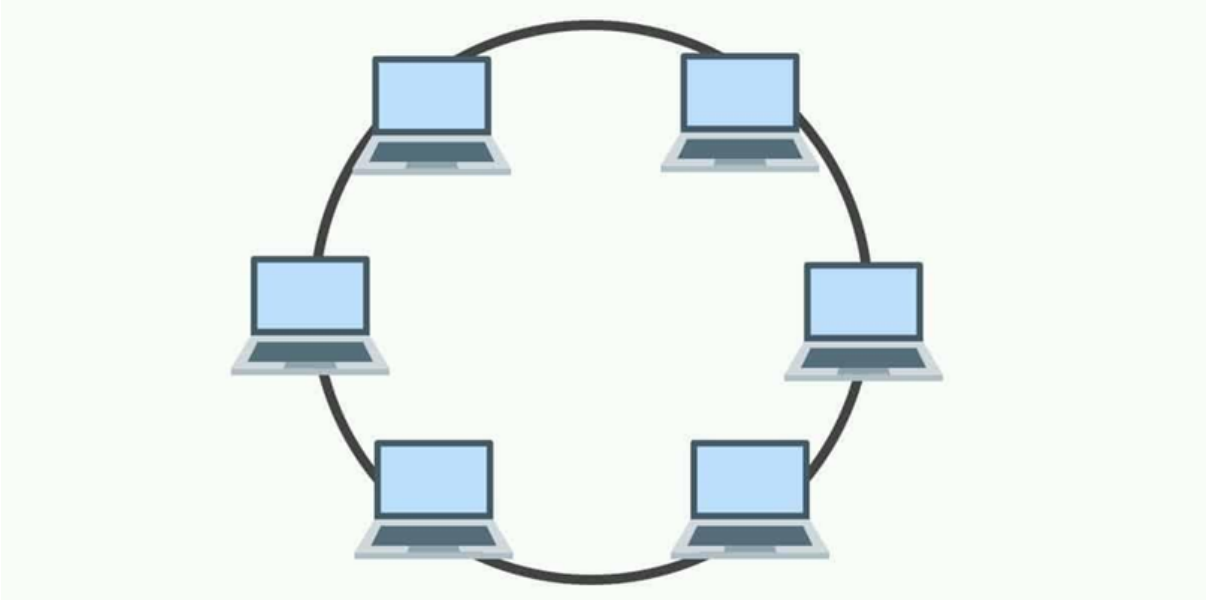
Disadvantages-

1) It is a type of network expensive.
2) Design of a hybrid network is very complex.
3) There is change hardware in order to connect topology with another topology.

6)Ring-

Introduction-

In this topology, it forms a ring connecting devices with exactly two neighbouring devices.



Architecture-

A number of repeaters are used for Ring topology with a large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass

through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the network.

Applications-

1) It is used in the Wide Area Network (WAN) and in Metropolitan Area Network (MAN) is used in vast areas for connecting all the (LAN) it is also used in-ring network also the most crucial part in a ring topology.
2) Local Area Network (LAN) is used in all computer machines connected to the ring network for the data flow in the unidirectional and bidirectional path.

Advantages-

1) The possibility of collision is minimum in this type of topology.

2) Cheap to install and expand.

Disadvantages-

1) Troubleshooting is difficult in this topology.
2) The addition of stations in between or removal of stations can disturb the whole topology. 3) Less secure.

Conclusion:

Thus, we successfully studied different networking devices and topologies

EXPERIMENT - 2

Name: Meet Patel **SapID: 60004200104**

Batch: B1

AIM : To study and execute different networking commands

Commands:

ipconfig :IT stands for Internet Protocol Configuration.The ipconfig command lists the network interfaces attached to the PC along with other statistics such as the IP addresses associated with each interface, subnet mask and default gateway for all adapters. This is a command-line application which displays all the current TCP/IP(Transmission Control Protocol / Internet Protocol) network configuration, refreshes the DHCP (Dynamic Host Configuration Protocol) and DNS (Domain Name Server).

Display the basic TCP/IP configuration for all adapters :

```
C:\Users\djsce.student>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::e01d:4e6c:666a:9675%3
   IPv4 Address. . . . . . . . . . . : 10.120.63.64
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.120.63.1

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::137:b2ba:2232:5345%5
   IPv4 Address. . . . . . . . . . . : 192.168.119.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

ipconfig -all :Displays the full TCP/IP configuration for all adapters. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connections.

Display the full TCP/IP configuration for all adapters :

```
C:\Users\djsce.student>ipconfig -all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : MUm915CPU1594
   Primary Dns Suffix  . . . . . . . : SVKMGRP.COM
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : SVKMGRP.COM

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GBE Family Controller
   Physical Address. . . . . . . . . : EC-B1-D7-65-0A-7B
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::e01d:4e6c:666a:9675%3(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.120.63.64(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.120.63.1
   DHCPv6 IAID . . . . . . . . . . . : 65843671
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-21-68-B9-31-EC-B1-D7-65-0A-7B
   DNS Servers . . . . . . . . . . . : 192.168.2.51
                                       192.168.2.52
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
   Physical Address. . . . . . . . . : 00-50-56-C0-00-01
   DHCP Enabled. . . . . . . . . . . : No
```

```
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::137:b2ba:2232:5345%5(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.119.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 83906646
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-21-68-B9-31-EC-B1-D7-65-0A-7B
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

ping : Short for packet internet groper, the ping command is used to check connectivity between 2 systems or servers.Verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) echo Request messages. The receipt of corresponding echo Reply messages are displayed, along with round-trip times. ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution. You can also use this command to test both the computer name and the IP address of the computer.

To ping the destination 10.120.63.65:

```
C:\Users\djsce.student>ping 10.120.63.65

Pinging 10.120.63.65 with 32 bytes of data:
Reply from 10.120.63.65: bytes=32 time=2ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128

Ping statistics for 10.120.63.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

ping -t : Specifies ping continue sending echo Request messages to the destination until interrupted. To interrupt and display statistics, press CTRL+ENTER. To interrupt and quit this command, press CTRL+C

To ping -t the destination 10.120.63.65:

```
C:\Users\djsce.student>ping -t 10.120.63.65

Pinging 10.120.63.65 with 32 bytes of data:
Reply from 10.120.63.65: bytes=32 time=2ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=2ms TTL=128
Reply from 10.120.63.65: bytes=32 time=2ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128
Reply from 10.120.63.65: bytes=32 time=1ms TTL=128

Ping statistics for 10.120.63.65:
    Packets: Sent = 12, Received = 12, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
Control-C
^C
```

netstat : The netstat command displays a variety of network statistics about a computer's active TCP/IP connections. . It can display the routing table, ports that various services are listening on, TCP connections. This command has a number of different functions, but the most useful of these is to display network summary information for the device.

Display network interfaces attached to your PC:

```
C:\Users\djsce.student>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.120.63.64:50076     w-srv-sccm:10123       ESTABLISHED
  TCP    10.120.63.64:50519     bom07s29-in-f14:https  ESTABLISHED
  TCP    10.120.63.64:50551     sa-in-f188:5228        ESTABLISHED
  TCP    10.120.63.64:50621     bom12s20-in-f14:https  ESTABLISHED
  TCP    10.120.63.64:52870     bom12s21-in-f10:https  ESTABLISHED
  TCP    10.120.63.64:52932     se-in-f101:https       ESTABLISHED
  TCP    10.120.63.64:52942     bom12s01-in-f14:https  ESTABLISHED
```

```
TCP    10.120.63.64:52948     bom07s32-in-f3:https   TIME_WAIT
TCP    10.120.63.64:52950     216.239.34.117:https   TIME_WAIT
TCP    10.120.63.64:52953     bom07s35-in-f19:https  ESTABLISHED
TCP    10.120.63.64:52967     bom07s20-in-f3:https   ESTABLISHED
TCP    10.120.63.64:52969     20.42.73.24:https      TIME_WAIT
```

netstat -an : The netstat -an command prints out the TCP connections as well as UDP connections.

```
C:\Users\djsce.student>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:902            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:912            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2701           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49153          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49154          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49155          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49156          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:59698          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:59711          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:59717          0.0.0.0:0              LISTENING
  TCP    10.120.63.64:139       0.0.0.0:0              LISTENING
  TCP    10.120.63.64:50076     192.168.2.168:10123    ESTABLISHED
  TCP    10.120.63.64:50519     142.250.182.238:443    ESTABLISHED
  UDP    0.0.0.0:123            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:3389           *:*
  UDP    0.0.0.0:4500           *:*
  UDP    192.168.119.1:59735    *:*
  UDP    192.168.225.1:137      *:*
  UDP    192.168.225.1:138      *:*
  UDP    192.168.225.1:1900     *:*
  UDP    192.168.225.1:59736    *:*
```

```
  UDP    [::]:123               *:*
  UDP    [::]:500               *:*
  UDP    [::]:3389              *:*
  UDP    [fe80::137:b2ba:2232:5345%5]:546  *:*
  UDP    [fe80::137:b2ba:2232:5345%5]:1900  *:*
```

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

```
C:\Users\djsce.student>pathping www.mu.ac.in

Tracing route to www.mu.ac.in [14.139.125.195]
over a maximum of 30 hops:
  0  MUm915CPU1594.SVKMGRP.COM [10.120.63.64]
  1  10.120.63.1
  2  10.120.138.18
  3    *       *       *
Computing statistics for 50 seconds...
^C
```

arp -a: The ARP command corresponds to the Address Resolution Protocol. Although it is easy to think of network communications in terms of IP addressing, packet delivery is ultimately dependent on the Media Access Control (MAC) address of the device's network adapter. This is where the Address Resolution Protocol comes into play. Its job is to map IP addresses to MAC addresses.

```
C:\Users\djsce.student>arp -a

Interface: 10.120.63.64 --- 0x3
  Internet Address      Physical Address      Type
  10.120.63.1           00-87-31-c8-2a-d6     dynamic
  10.120.63.46          ec-b1-d7-64-b3-c0     dynamic
  10.120.63.49          ec-b1-d7-64-b3-71     dynamic
  10.120.63.51          ec-b1-d7-64-f6-00     dynamic
  10.120.63.52          40-a8-f0-a7-a0-d7     dynamic
  10.120.63.253         ec-b1-d7-67-99-a2     dynamic
```

```
  10.120.63.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.119.1 --- 0x5
  Internet Address      Physical Address      Type
  192.168.119.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.225.1 --- 0x7
  Internet Address      Physical Address      Type
  192.168.225.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

nslookup: The nslookup utility is a command-line tool that is used for making DNS lookups in a bid to retrieve domain names and A records. Type the nslookup command, and Windows will display the name and IP address of the device's default DNS server. From there, you can type host names in an effort to see if the DNS server is able to resolve the specified host name.

```
C:\Users\djsce.student>nslookup
Default Server:  mumdc-prim.svkmgrp.com
Address:  192.168.2.51

>
```

EXPERIMENT - 3

Name: Meet Patel **SapID: 60004200104**
Batch: B1

Aim: To implement CRC and Hamming Code as error detection and correction codes.

Theory:

Hamming Code:

Hamming code is a set of error-correction codes that can be used to detect and correct the errors that can occur when the data is moved or stored from the sender to the receiver.
Redundant bits are extra binary bits that are generated and added to the information- carrying bits of data transfer to ensure that no bits were lost during the data transfer. These redundancy bits are placed at the positions which correspond to the power of 2.

Parity bits –

A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1’s in the data is even or odd. Parity bits are used for error detection. There are two types of parity bits:

Even parity bit:

In the case of even parity, for a given set of bits, the number of 1’s are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences

of 1’s an even number. If the total number of 1’s in a given set of bits is already even, the parity bit’s value is 0.

Odd Parity bit –

In the case of odd parity, for a given set of bits, the number of 1’s are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1’s an odd number. If the total number of 1’s in a given set of bits is already odd, the parity bit’s value is 0.

Cyclic Redundancy Check (CRC):

CRC or Cyclic Redundancy Check is a method of detecting accidental changes/errors in the communication channel. CRC uses **Generator Polynomial** which is available on both sender and receiver side. An example generator polynomial is of the form like $x^3 + x + 1$. This generator polynomial represents key 1011.

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.

Code(CRC):

```
#include <stdio.h>

 void modulo2Div(int n,int l,int g[], int d[],int code[]){
   int key[l+n],m,j,k;
```

```
    for(int i=0;i<l;i++){
        m=0;
        k=key[i];
        for(j=i;j<i+4;j++){
            if(i==0){
                if(g[m]==d[m]){
                    key[j]=0;
                }else{
                    key[j]=1;
                }
            }
            else if(k==0){
                    if(key[j]==0){
                        key[j]=0;
                    }else{
                        key[j]=1;
                    }
                }
            else{
                if(key[j]==g[m]){
                    key[j]=0;
                }else{
                    key[j]=1;
                }
            }
        m++;
        }
        key[j]=d[i+4];
    }
     for(int i=0;i<l;i++){
     code[i]=d[i];
     }

     for(int i=l;i<n+l;i++){
     code[i]=key[i];
     }

}

int main()
{
     int n, error=0;
     printf("Enter the highest degree of G(x) : ");
     scanf("%d",&n);
     int g[n];
     for(int i=0;i<=n;i++){
         printf("\nEnter the coefficient of x^%d : ",n-i);
         scanf("%d",&g[i]);
```

```
 }
 int l;
 printf("Enter the length of original data : ");
 scanf("%d",&l);
 int d[l+n];
    printf("\nEnter the original data : ");
 for(int i=0;i<l;i++){
    scanf("%d",&d[i]);
 }
 for(int i=0;i<n;i++){
    d[l+i]=0;
 }
int code[l+n];
modulo2Div(n,l,g,d,code);

printf("\nGenerator : ");

 for(int i=0;i<=n;i++){
 printf("%d",g[i]);
 }
 printf("\n");
printf("\nData : ");

 for(int i=0;i<l;i++){
 printf("%d",d[i]);
 }
 printf("\n");
printf("\nRemainder : ");

 for(int i=l;i<l+n;i++){
 printf("%d",code[i]);
 }
 printf("\n");

printf("\nCodeword : ");

 for(int i=0;i<l+n;i++){
 printf("%d",code[i]);
 }
 printf("\n");
modulo2Div(n,l,g,code,code);

 for(int i=l;i<l+n;i++){
     if(!code[i]==0){
         error=1;
     }
 }
```

```
    if(error){
        printf("\nError!");
    }
    else{
        printf("\nRemainder at receiving side : ");
        for(int i=l;i<l+n;i++){
        printf("%d",code[i]);
        }
        printf("\n");
        printf("\nReceived successfully");
    }
   return 0;
}
```

Output:

```
Enter the highest degree of G(x) : 3

Enter the coefficient of x^3 : 1

Enter the coefficient of x^2 : 1

Enter the coefficient of x^1 : 0

Enter the coefficient of x^0 : 1
Enter the length of original data : 6

Enter the original data : 1
0
0
1
0
0

Generator : 1101

Data : 100100

Remainder : 001

Codeword : 100100001

Remainder at receiving side : 000

Received successfully

...Program finished with exit code 0
Press ENTER to exit console.
```

Code(Hamming Code):

```
#include <stdio.h>
#include <math.h>

int hamming_calculate(int position, int length,int code[])
{
int count = 0; int i,j,k;
i = position - 1;

while(i<length)
{
for( j = i;j<i+position;j++)
{
if(code[j]==1){

count++;
}
}

i = i + 2 * position;

}

if(count % 2 == 0){
return 0;

}
else {
return 1;
}

}


void print(int n,int p,int code[])
{
for(int i=0;i<n+p;i++)
{
printf(" %d \t",code[i]);
```

```
}
printf("\n"); int j=0;
for(int i=0;i<n+p;i++){ if(i == (pow(2,j) - 1)){
printf(" P%d \t",i+1); j++;
}
else {
printf(" D%d \t",i+1);

}
}

}


int main()
{
int code[50], input[50]; int n,p;
int counter;
printf("Enter the size of hamming code : "); scanf("%d",&n);
for(int i=1;i<n;i++){ if(pow(2,i) >= n+i+1 ){
p = i; break;
}
}

printf("Enter the data of hamming code \n"); for(int i=0;i<n;i++){
printf("Enter the value of bit %d : ",i+1); scanf("%d",&input[i]);
printf("\n");
}

int j = 0,k = 0;
for(int i=0;i<n+p;i++){ if(i == (pow(2,j)-1)){
code[i] = 0; j++;

}
else {
code[i] = input[k]; k++;
}
}



printf("The Intial Hamming code is :\n"); print(n,p,code);
```

```
for(int i=0;i<p;i++)
{
int position = pow(2,i);
int value = hamming_calculate(position, n+p,code); code[position - 1] = value;
}

printf("\n\n");
printf("The Final Hamminag Code is : \n"); print(n,p,code);

return 0;
}
```

```
Enter the size of hamming code : 10
Enter the data of hamming code
Enter the value of bit 1 : 1

Enter the value of bit 2 : 0

Enter the value of bit 3 : 1

Enter the value of bit 4 : 0

Enter the value of bit 5 : 1

Enter the value of bit 6 : 0

Enter the value of bit 7 : 1

Enter the value of bit 8 : 0

Enter the value of bit 9 : 1

Enter the value of bit 10 : 0

The Intial Hamming code is :
 0       0       1       0       0       1       0       0       1       0       1       0       1       0
 P1      P2      D3      P4      D5      D6      D7      P8      D9      D10     D11     D12     D13     D14

The Final Hamminag Code is :
 0       1       1       0       0       1       0       1       1       0       1       0       1       0
 P1      P2      D3      P4      D5      D6      D7      P8      D9      D10     D11     D12     D13     D14

...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion: Thus we successfully implemented Error detection and correction techniques

Experiment - 4

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim: To implement Djikstra's shortest path algorithm .

Theory: Dijkstra Algorithm is a very famous greedy algorithm. It is used for solving the single source shortest path problem. It computes the shortest path from one particular source node to all other remaining nodes of the graph. Condition: It is important to note the following points regarding Dijkstra AlgorithmDijkstra algorithm works only for connected graphs. Dijkstra algorithm works only for those graphs that do not contain any negative weight edge. The actual Dijkstra algorithm does not output the shortest paths. It only provides the value or cost of the shortest paths. By making minor modifications in the actual algorithm, the shortest paths can be easily obtained. Dijkstra algorithm works for directed as well as undirected graphs. Working of Djikstras Algorithm: Dijkstra's Algorithm works on the basis that any subpath B -> D of the shortest path A -> D between vertices A and D is also the shortest path between vertices B and D. Each subpath is the shortest path Djikstra used this property in the opposite direction i.e we overestimate the distance of each vertex from the starting vertex. Then we visit each node and its neighbors to find the shortest subpath to those neighbors. The algorithm uses a greedy approach in the sense that we find the next best solution hoping that the end result is the best solution for the whole problem.

Code:

```
#include <stdio.h>

int main()
{
   int n=6,p,check,min;
//    printf("Enter the number of nodes : ");
//      scanf("%d",&n);

    // int cost[n][n], D[n], V[n];

    // printf("\nEnter the Distances\n");
    // for (int i = 0; i < n; i++) {
    //      for (int j = 0; j <n ; j++) {
    //          if(i==j){
    //              cost[i][j]==0;
    //          }
    //          else{
    //              printf("\nDistance of node %d to %d : ",i+1,j+1);
    //              scanf("%d",&cost[i][j]);
    //          }
    //      }
    //      D[i]=50;
    //      V[i]=0;
    // }
  int
cost[6][6]={{0,7,50,50,50,3},{7,0,4,50,50,2},{50,4,0,8,5,5},{50,50,8,0,3,5
0},{50,50,5,3,0,6},{3,2,5,50,6,0}};
  int V[6]={0,0,0,0,0,0};
  int D[6]={50,50,50,50,50,50};
```

```
    printf("\nEnter the source node : ");
    scanf("%d",&p);
    p=p-1;
    D[p]=0;
    for(int i=0;i<n-1;i++){

        for(int i=0;i<n;i++){
            check=D[p]+cost[p][i];
            if(check<D[i]){
                D[i]=check;
            }
        }
        V[p]=1;
        min=50;
        for(int i=1;i<n;i++){
            if(min>D[i] && V[i]==0){
                min=D[i];
                p=i;
            }
        }

    }

    printf("\nShortest path using Dijstra's is : ");
    for(int i=0;i<n;i++){
        printf("%d ",D[i]);
    }

    return 0;
}
```

Output:

```
Enter the source node : 1

Shortest path using Dijstra's is : 0 5 8 12 9 3

...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion: Dijkstra's shortest path algorithm has been successfully executed.

Experiment - 5

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim:To study and implement different framing techniques.

Theory:

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.

Character Stuffing

Character stuffing is also known as byte stuffing or character-oriented framing and is same as that of bit stuffing but byte stuffing actually operates on bytes whereas bit stuffing operates on bits.

Here the data is stuffed at start and end with characters not present int data word itself. Thus, we return the data by adding the unique start and ending characters or a special byte that is basically known as ESC (Escape Character) that has predefined pattern is generally added to data section of the data stream or frame when there is message or character that has same pattern as that of flag byte.

But receiver removes this ESC and keeps data part that causes some problems or issues. In simple words, we can say that character stuffing is addition of 1 additional byte if there is presence of ESC or flag in text.

Bit Stuffing

Bit stuffing is also known as bit-oriented framing or bit-oriented approach. In bit stuffing, extra bits are being added by network protocol designers to data streams. It is generally insertion or addition of extra bits into transmission unit or message to be transmitted as simple way to provide and give signalling information and data

to receiver and to avoid or ignore appearance of unintended or unnecessary control sequences. It is type of protocol management simply performed to break up bit pattern that results in transmission to go out of synchronisation. Bit stuffing is very essential part of transmission process in network and communication protocol. It is also required in USB.

Code:

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
   int opt;
   do
   {
```

```
    cout << "1. Character Count\n2. Character Stuffing\n3. Bit
Stuffng.\n4. Exit";
    cout << "\n\nEnter an option: ";
    string data, finalData, temp;
    cin >> opt;
    switch (opt)
    {
    case 1:
    {
       cout << "Enter the number of frames: ";
       int frames;
       cin >> frames;
       while (frames--)
       {
          cout << "Enter frame data: ";
          cin >> temp;
          finalData += (std::to_string(temp.length()) + temp);
       }
       cout << "\nFinal data: " << finalData << "\n\n";
       break;
    }
    case 2:
    {
       cout << "Enter the data: ";
       cin >> data;
       string stx = "STX", dle = "DLE";
       int i = 0, count = 0;
       finalData += stx + dle;
       while (i != data.length())
       {
          finalData += data[i];
          if (data[i] == 'D')
          {
             count++;
          }
          else if (count == 1 && data[i] == 'L')
```

```
        {
            count++;
        }
        else if (count == 2 && data[i] == 'E')
        {
            finalData += "DLE";
            count = 0;
        }
        else
        {
            count = 0;
        }
        i++;
    }
    finalData += dle + stx;
    cout << "\nFinal data: " << finalData << "\n\n";
    break;
}
case 3:
{
    cout << "Enter the data: ";
    cin >> data;
    int i = 0, count = 0;
    string flag = "01111110";
    finalData += flag;
    while (i != data.length())
    {
        if (data[i] == '1' && count == 5)
        {
            count = 0;
            finalData += '0';
        }
        else if (data[i] == '1')
        {
            count++;
            finalData += data[i];
```

```
            }
            else if (count > 0)
            {
               count = 0;
               finalData += data[i];
            }
            i++;
         }
         finalData += flag;
         cout << "\nFinal data: " << finalData << "\n\n";
         break;
      }
      case 4:
      {
         cout << "Exiting..!\n";
         break;
      }
      default:
      {
         cout << "Enter an valid OPTION!!";
      }
      }
   } while (opt != 4);
   return 0;
}
```

Output:

```
1. Character Count
2. Character Stuffing
3. Bit Stuffng.
4. Exit

Enter an option: 1
Enter the number of frames: 3
Enter frame data: bfadvczv
Enter frame data: fds
Enter frame data: fvbfd

Final data: 8bfadvczv3fds5fvbfd

1. Character Count
2. Character Stuffing
3. Bit Stuffng.
4. Exit

Enter an option:
```

```
1. Character Count
2. Character Stuffing
3. Bit Stuffng.
4. Exit

Enter an option: 2
Enter the data: dfsfgDLEvfdvd

Final data: STXDLEdfsfgDLEDLEvfdvdDLESTX

1. Character Count
2. Character Stuffing
3. Bit Stuffng.
4. Exit

Enter an option:
```

```
Enter an option: 3
Enter the data: 10101111111111110111101100

Final data: 011111101010111110111110111101100111111O

1. Character Count
2. Character Stuffing
3. Bit Stuffng.
4. Exit

Enter an option: 4
Exiting..!


...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion: We have Successfully Implemented Different Framing Techniques like Character Count, Character Stuffing and Bits Stuffing used during Communication

Experiment - 6

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim:To implement socket communication in java

Theory:

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

Java Socket programming is used for communication between the applications running on different JRE. Java Socket programming can be connection-oriented or connectionless. Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connectionless socket programming.

The client in socket programming must know two information:

1. IP Address of Server, and
2. Port number.

Socket Class

A socket is simply an endpoint for communications between the machines. The Socket class can be used to create a socket.

ServerSocket Class

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

User Datagram Protocol(UDP):

DatagramSockets are Java's mechanism for network communication via UDP instead of TCP. Java provides DatagramSocket to communicate over UDP instead of TCP. It is also built on top of IP. DatagramSockets can be used to both send and receive packets over the Internet.

One of the examples where UDP is preferred over TCP is the live coverage of TV channels. In this aspect, we want to transmit as many frames to live audience as possible not worrying about the loss of one or two frames. TCP being a reliable protocol add its own overhead while transmission. Another example where UDP is preferred is online multiplayer gaming. In games like counter- strike or call of duty, it is not necessary to relay all the information but the most important ones. It should also be noted that most of the applications in real life uses careful blend of both UDP and TCP; transmitting the critical data over TCP and rest of the data via UDP.

Code(TCP):
Server:

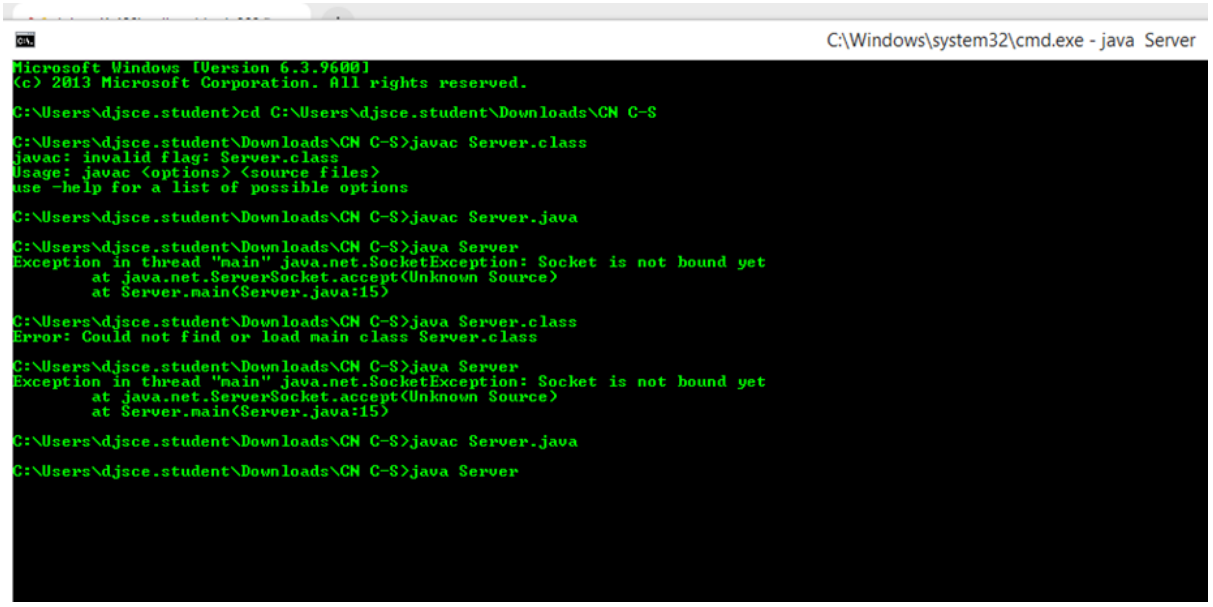
```
import java.io.*;
import java.net.*;

class Server{
public static void main(String[] args) throws Exception{

String msg;
ServerSocket ss = new ServerSocket(80);

while(true){

Socket s1=ss.accept();
String m[]={"m","t","w","th","f","sa","su"};

int i=(int)(Math.random()*m.length);

msg=m[i];
PrintStream ps=new PrintStream(s1.getOutputStream());

ps.println(msg);

}

}
}
```

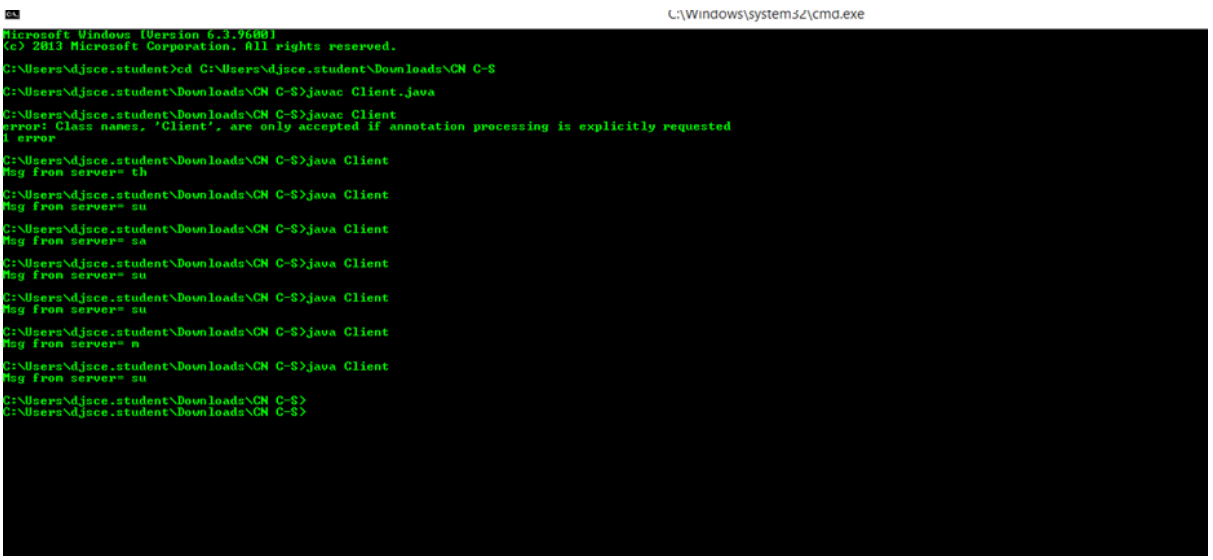
Client:

```
import java.io.*; import java.net.*; class Client
{
public static void main(String[] args) throws Exception {
```

```
Socket cs = new Socket("localhost",80); BufferedReader br=new
BufferedReader(new
InputStreamReader(cs.getInputStream()));
String m=br.readLine(); System.out.println("Msg from server= "+m);
cs.close();
}
}
```

Output:





Code(UDP):
(Server Side):

```
import java.io.IOException; import java.net.DatagramPacket; import
java.net.DatagramSocket; import java.net.InetAddress; import
java.net.SocketException;

public class udpserver
{
public static void main(String[] args) throws IOException
{
DatagramSocket ds = new DatagramSocket(1234); byte[] receive =
new byte[65535];

DatagramPacket DpReceive = null; while (true)
{

DpReceive = new DatagramPacket(receive, receive.length);
ds.receive(DpReceive);
System.out.println("Client:-" + data(receive));

if (data(receive).toString().equals("bye"))
{
System.out.println("Client sent bye.     EXITING");
break;
}
receive = new byte[65535];
}
}


public static StringBuilder data(byte[] a)
```

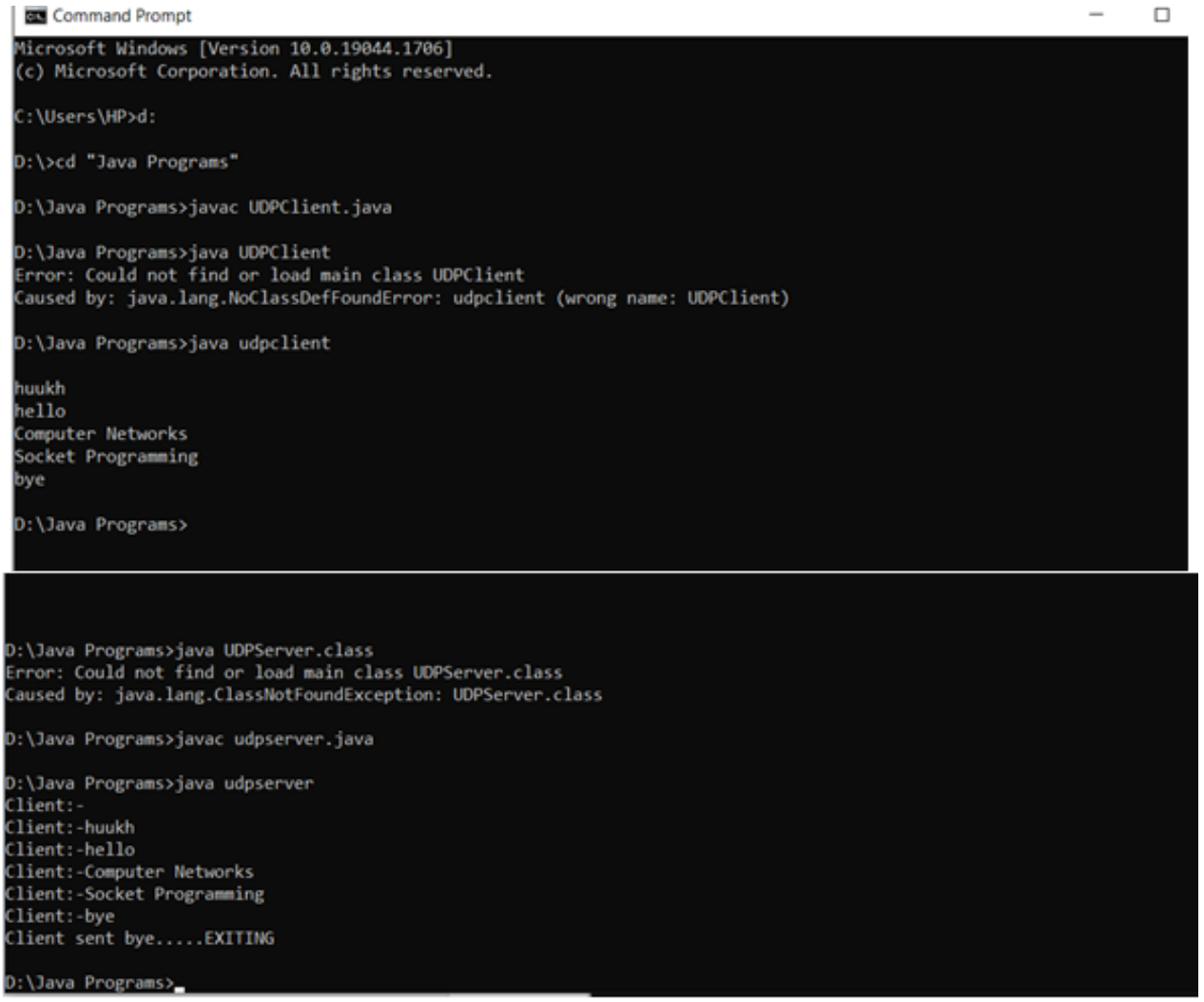
```
{
if (a == null) return null;
StringBuilder ret = new StringBuilder(); int i = 0;
while (a[i] != 0)
{
ret.append((char) a[i]); i++;
}
return ret;
}
}
```

(Client Side):

```
import java.io.IOException; import java.net.DatagramPacket; import
java.net.DatagramSocket; import java.net.InetAddress; import
java.util.Scanner;

public class udpclient
{
public static void main(String args[]) throws IOException
{
Scanner sc = new Scanner(System.in);


DatagramSocket ds = new DatagramSocket();


InetAddress ip = InetAddress.getLocalHost(); byte buf[] = null;

while (true)
{
String inp = sc.nextLine(); buf = inp.getBytes();
```

```
DatagramPacket DpSend =
new DatagramPacket(buf, buf.length, ip, 1234);


ds.send(DpSend);


if (inp.equals("bye")) break;
}
}
}
```

Output:



Conclusion: We have Successfully implemented Socket programming using TCP and UDP protocols in JAVA.

Experiment - 7

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim: Creation of Duplex link in ns2 between two nodes

Theory:
Duplex is a bidirectional communication system that allows both end nodes to send and receive communication data or signals, simultaneously and one at a time. Both nodes have the ability to operate as sender and receiver at the same time, or take turns sending or receiving data.

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

The duplex links between n0 and n2, and n1 and n2 have 2 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 1.7 Mbps of bandwidth and 20 ms of delay. Each node uses a DropTail queue that has a maximum size of 10.

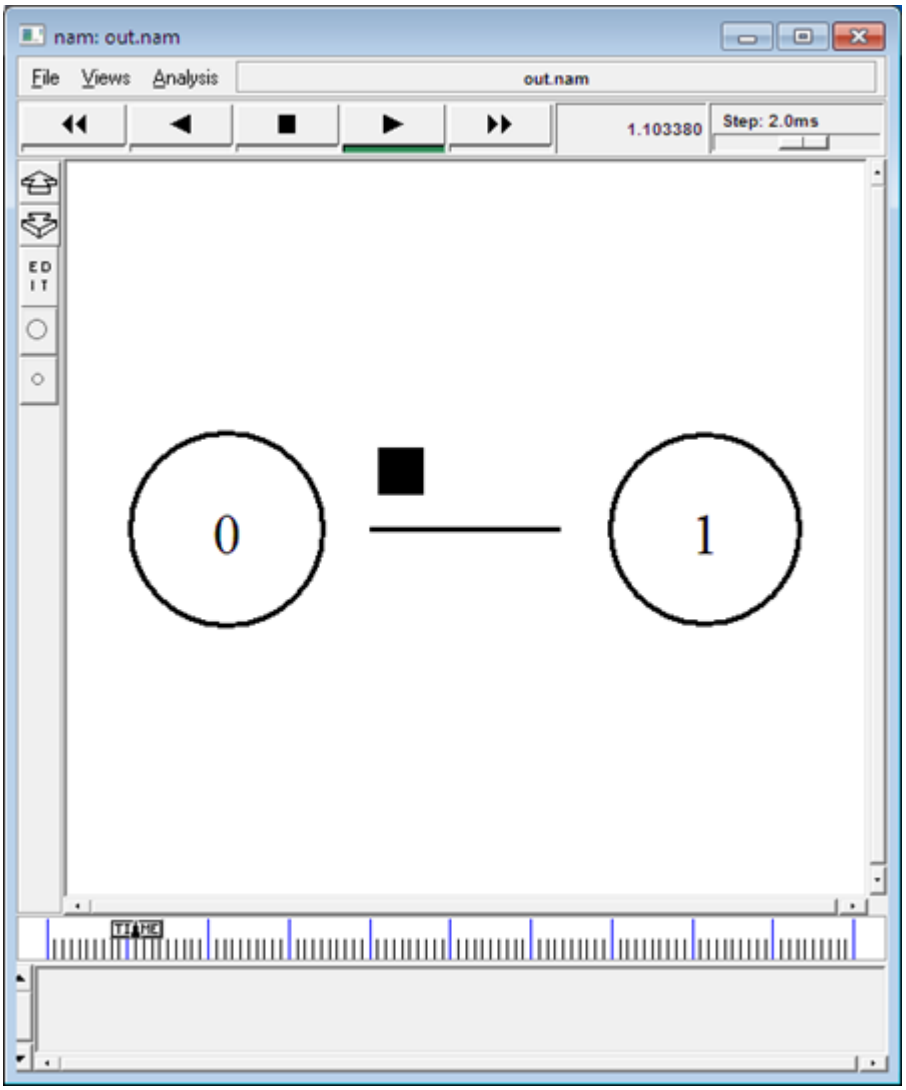
Commands:

```
#=================================
#       Simulation parameters setup
#=================================
set val(stop) 10.0       ;# time of simulation end
```

```
#==================================
#       Initialization
#==================================
#Create a ns simulator set ns [new Simulator]

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file set namfile [open out.nam w]
$ns namtrace-all $namfile
#==================================
#       Nodes Definition #==================================
#Create 2 nodes set n0 [$ns node] set n1 [$ns node]

#==================================
#       Links Definition #==================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right

#==================================
#       Agents Definition #==================================
#Setup a TCP connection set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500
```

```
#==================================
#       Applications Definition
#==================================
#Setup a FTP Application over TCP connection set ftp0 [new
Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#==================================
#       Termination
#==================================
#Define a 'finish' procedure proc finish {} {
global ns tracefile namfile
$ns flush-trace close $tracefile close $namfile
exec nam out.nam & exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"

$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

Output:



Conclusion:

Thus, we successfully created a duplex link in NS2.

Experiment - 8

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim: To implement and understand tcp-udp scenario in NS2

Theory:

TCP is a connection-oriented protocol, whereas UDP is a connectionless protocol. A key difference between TCP and UDP is speed, as TCP is comparatively slower than UDP. Overall, UDP is a much faster, simpler, and efficient protocol, however, retransmission of lost data packets is only possible with TCP.

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

A “UDP” agent that is attached to n0 is connected to a “null” agent attached to n3. A “null” agent frees the packets received. An “FTP” and a “CBR” traffic generator are respectively attached to “TCP” and “UDP” agents, and the “CBR” is configured to generate 1 Kbytes packets at the rate of 100 packets per second.
A “TCP” agent is attached to n1, and a connection is established to a TCP “sink” agent attached to n3. A TCP “sink” agent generates and sends ACK packets to the sender (TCP agent) and frees the received packets. A “UDP” agent that is attached to n0 is connected to a “null” agent attached to n3.

Commands:

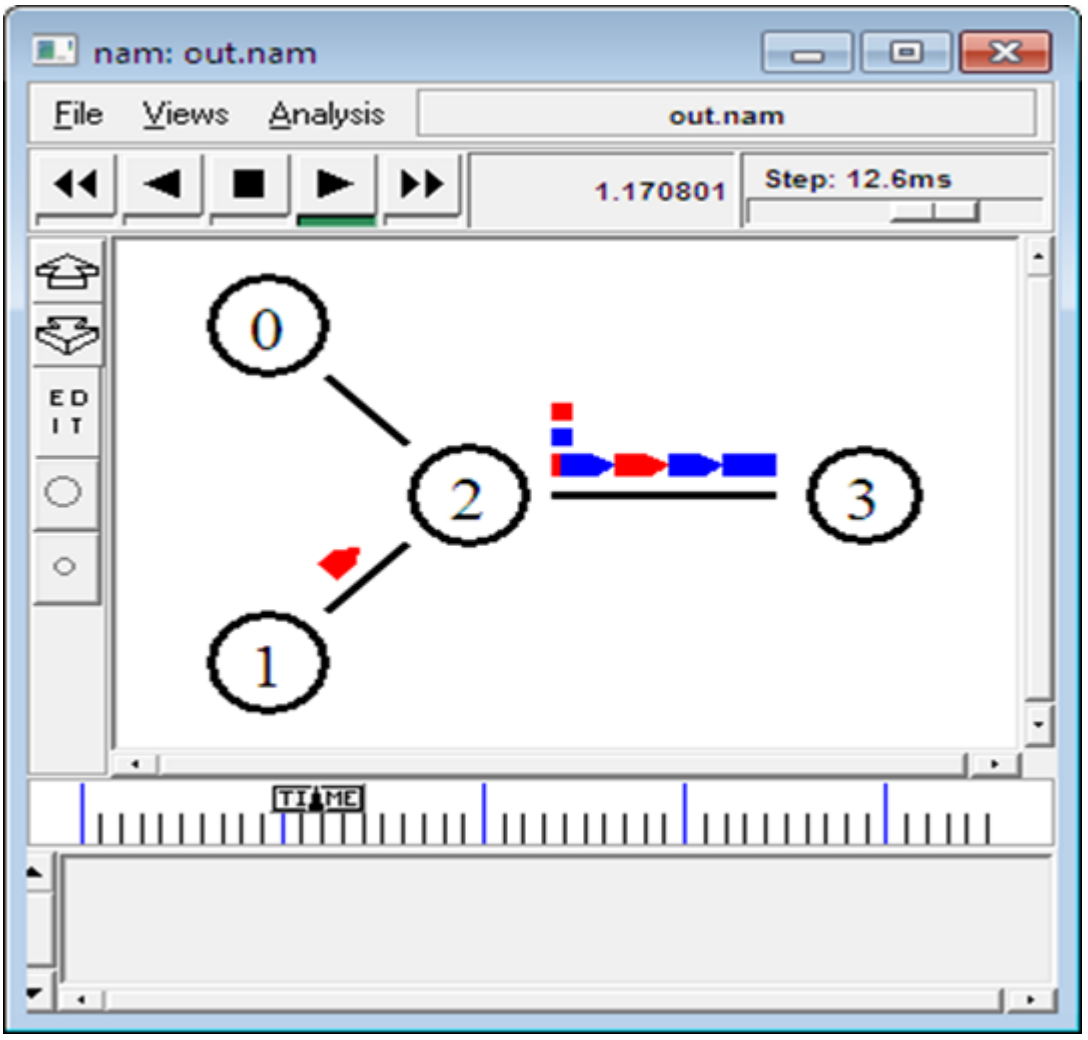
#Create a simulator object set ns [new Simulator]

#Define different colors for data flows (for NAM)

```
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure proc finish {} {
global ns nf
$ns flush-trace
#Close the NAM trace file close $nf
#Execute NAM on the trace file exec nam out.nam &

exit 0
}

#Create four nodes set n0 [$ns node] set n1 [$ns node] set n2 [$ns node] set
n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
```

```
$tcp set fid_ 1

#Setup a FTP over TCP connection set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

#Setup a UDP connection set udp [new Agent/UDP]
$ns attach-agent $n1 $udp set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"

$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]" puts "CBR interval = [$cbr set
interval_]"

#Run the simulation
$ns run
```

Output:



Conclusion: Thus, we studied TCP-UDP scenario in NS2

Experiment - 9

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim: Creation of Stop and Wait using ns2

Theory:

Stop-and-wait ARQ, also referred to as alternating bit protocol, is a method in telecommunications to send information between two connected devices. It ensures that information is not lost due to dropped packets and that packets are received in the correct order.

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

Commands:

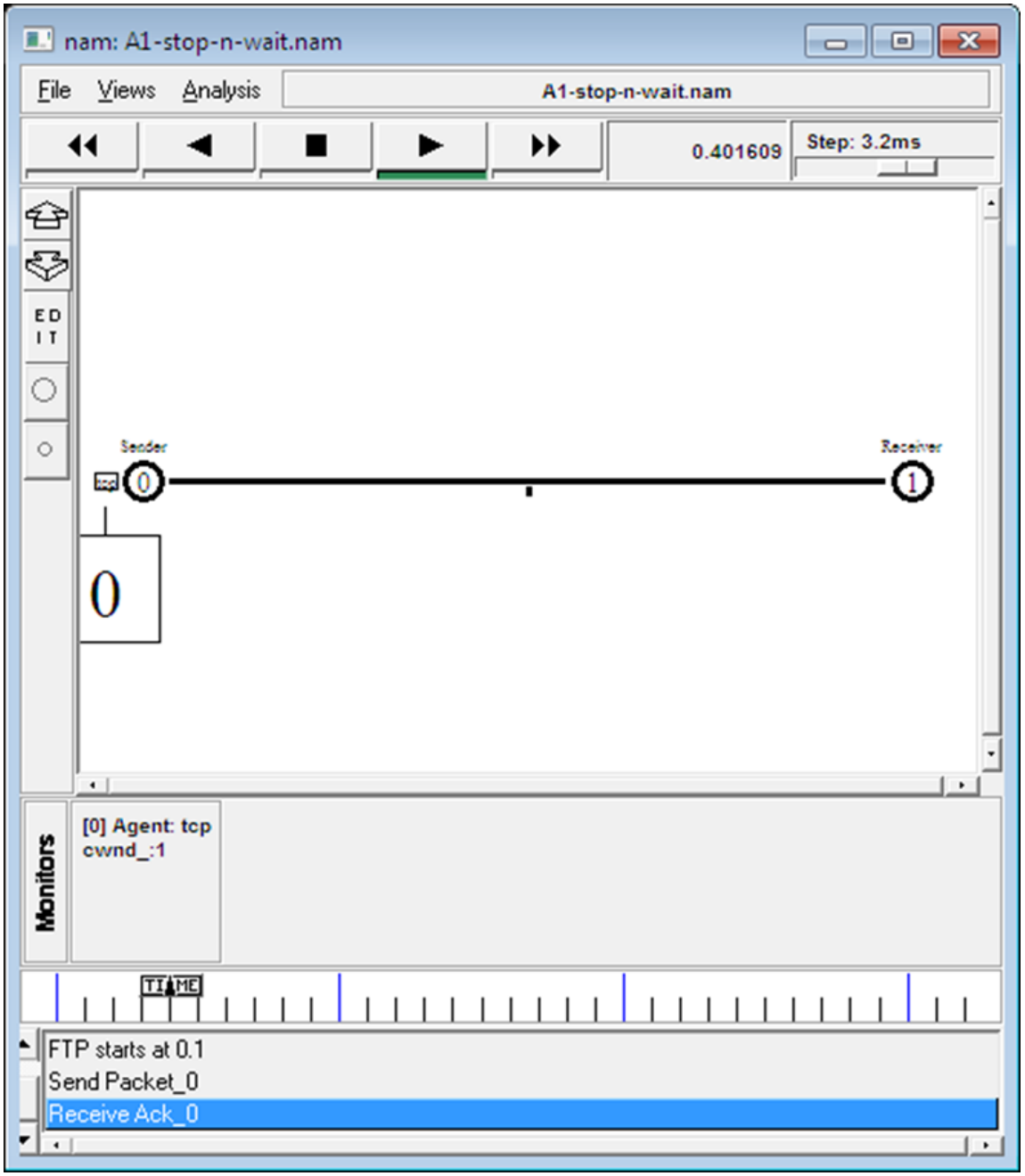
```
# stop and wait protocol in normal situation

# features : labeling, annotation, nam-graph, and window size monitoring set ns [new Simulator]

set n0 [$ns node]

set n1 [$ns node]

$ns at 0.0 "$n0 label Sender"
```

```
$ns at 0.0 "$n1 label Receiver"


set nf [open A1-stop-n-wait.nam w]
$ns namtrace-all $nf
set f [open A1-stop-n-wait.tr w]
$ns trace-all $f


$ns duplex-link $n0 $n1 0.2Mb 200ms DropTail
$ns duplex-link-op $n0 $n1 orient right
$ns queue-limit $n0 $n1 10
Agent/TCP set nam_tracevar_ true set tcp [new Agent/TCP]
$tcp set window_ 1
$tcp set maxcwnd_ 1
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns add-agent-trace $tcp tcp
$ns monitor-agent-trace $tcp
$tcp tracevar cwnd_
$ns at 0.1 "$ftp start"
$ns at 3.0 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n1 $sink"
```

```
$ns at 3.5 "finish"

$ns at 0.0 "$ns trace-annotate \"Stop and Wait with normal operation\""

$ns at 0.05 "$ns trace-annotate \"FTP starts at 0.1\""

$ns at 0.11 "$ns trace-annotate \"Send Packet_0\""

$ns at 0.35 "$ns trace-annotate \"Receive Ack_0\""

$ns at 0.56 "$ns trace-annotate \"Send Packet_1\""

$ns at 0.79 "$ns trace-annotate \"Receive Ack_1\""

$ns at 0.99 "$ns trace-annotate \"Send Packet_2\""

$ns at 1.23 "$ns trace-annotate \"Receive Ack_2 \""

$ns at 1.43 "$ns trace-annotate \"Send Packet_3\""

$ns at 1.67 "$ns trace-annotate \"Receive Ack_3\""

$ns at 1.88 "$ns trace-annotate \"Send Packet_4\""

$ns at 2.11 "$ns trace-annotate \"Receive Ack_4\""

$ns at 2.32 "$ns trace-annotate \"Send Packet_5\""

$ns at 2.55 "$ns trace-annotate \"Receive Ack_5          \""

$ns at 2.75 "$ns trace-annotate \"Send Packet_6\""

$ns at 2.99 "$ns trace-annotate \"Receive Ack_6\""

$ns at 3.1 "$ns trace-annotate \"FTP stops\"" proc finish {} {

global ns nf

$ns flush-trace close $nf

puts "filtering..."

exec tclsh ../ns-allinone-2.1b5/nam-1.0a7/bin/namfilter.tcl A1-stop-n-wait.nam puts
"running nam..."

exec nam A1-stop-n-wait.nam & exit 0

}
```

$ns run

Output:



Conclusion:

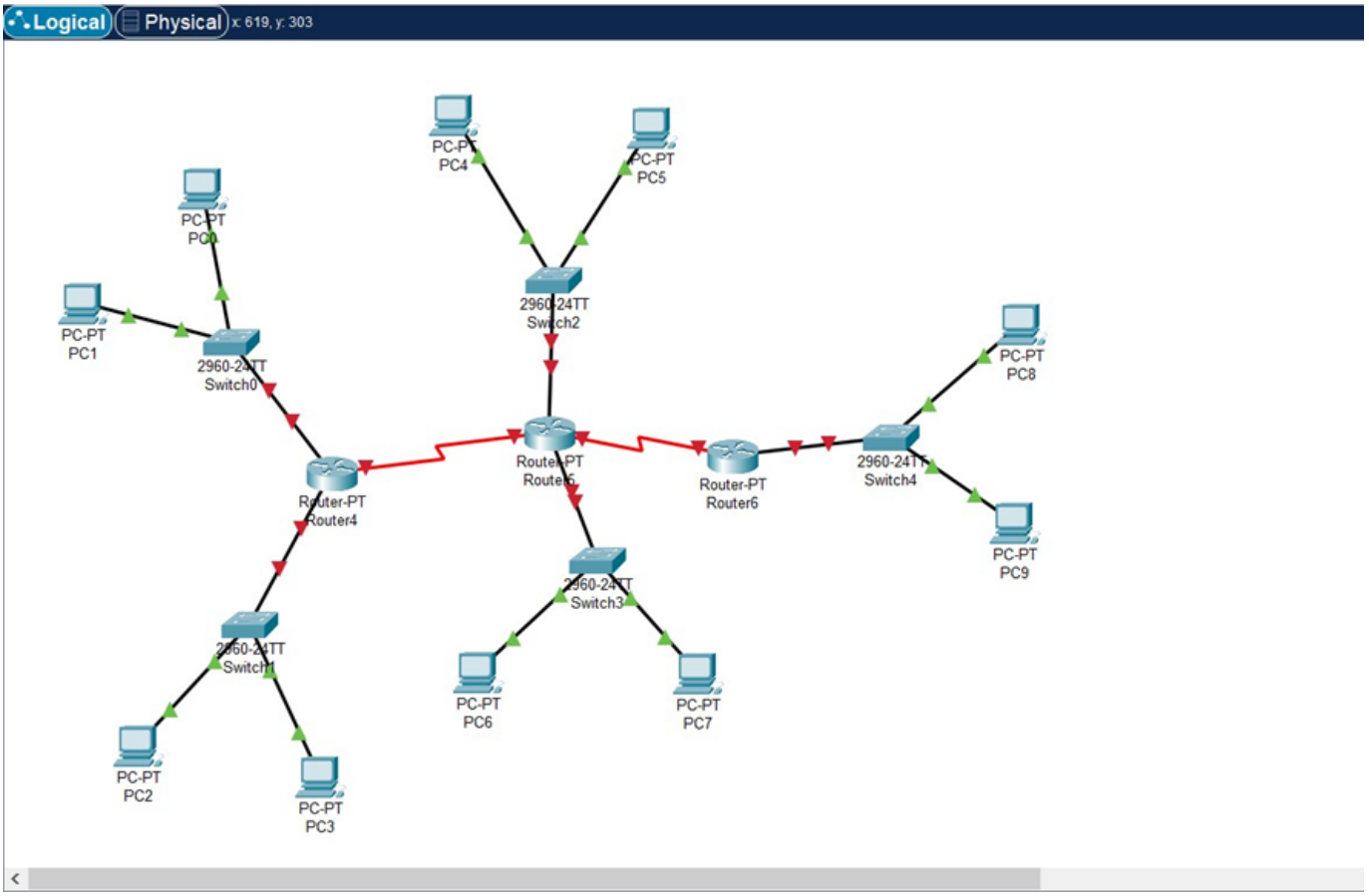
Thus, we studied stop and wait protocols in NS2

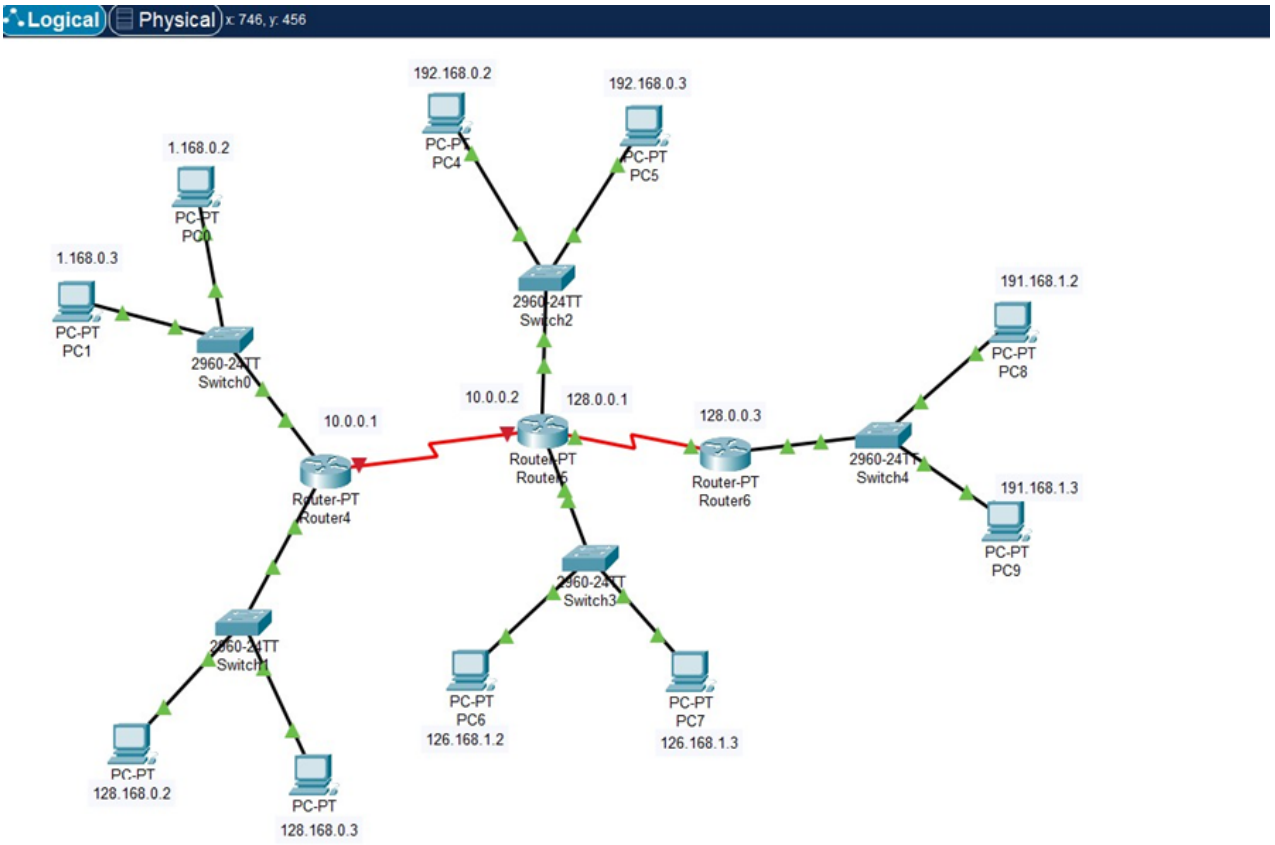
Experiment - 10

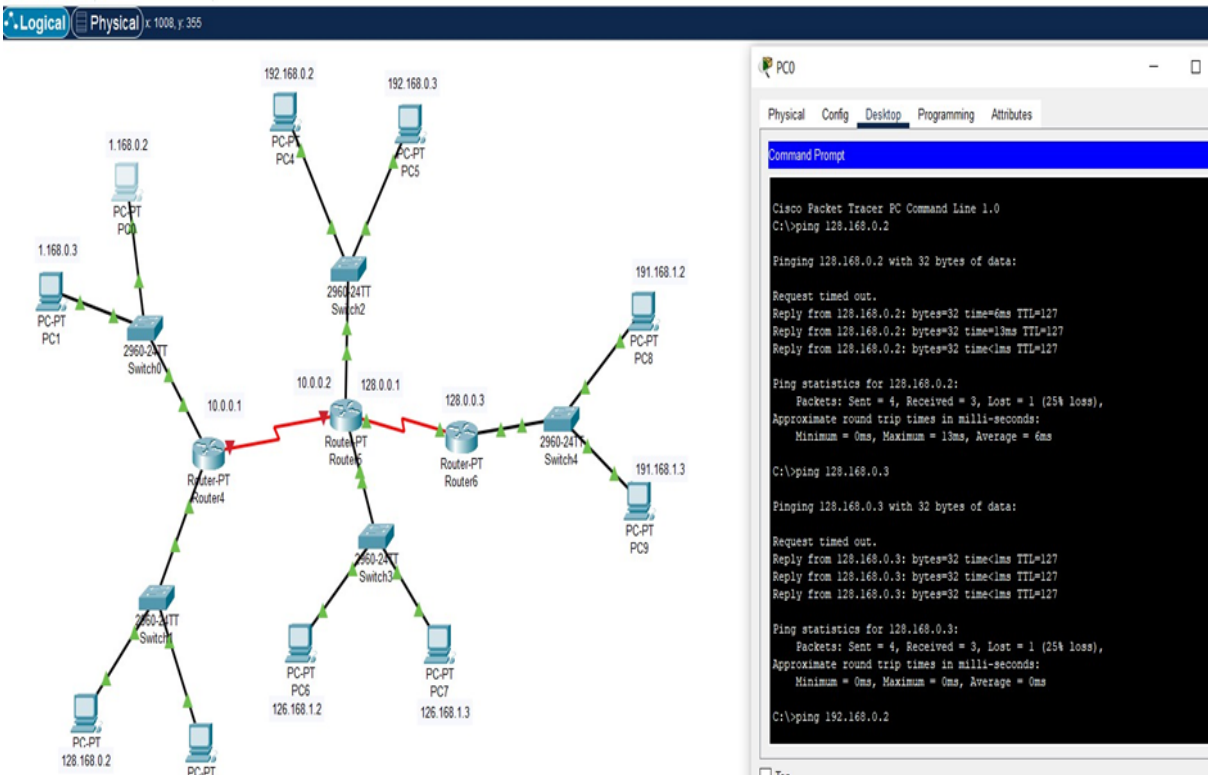
Name: Meet Patel **SapId:60004200104**

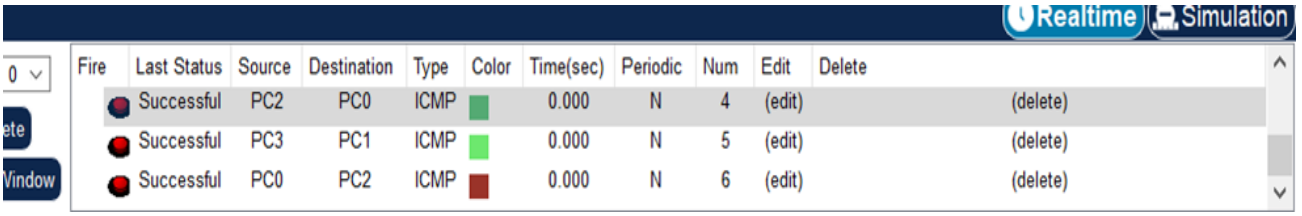
Batch: B1

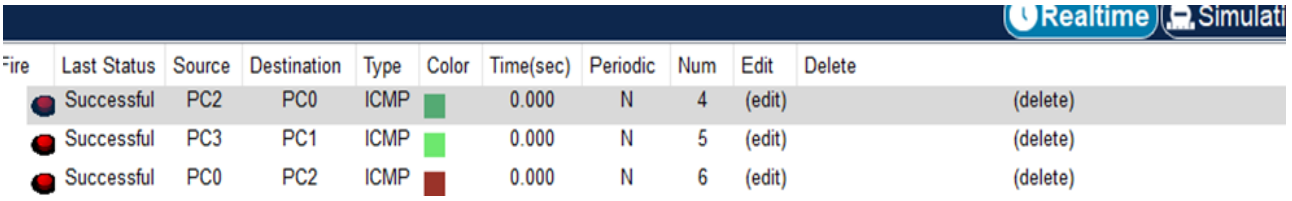
Aim: Create different networking topologies in NS2.




Logical
Physical
x: 746, y: 456
192.168.0.2
PC-PT
PC4
192.168.0.3
PC-PT
PC5
1.168.0.2
PC-PT
PC0
1.168.0.3
PC-PT
PC1
2960-24TT
Switch2
2960-24TT
Switch0
191.168.1.2
PC-PT
PC8
10.0.0.2
128.0.0.1
10.0.0.1
Router-PT
Router5
128.0.0.3
Router-PT
Router4
Router-PT
Router6
2960-24TT
Switch4
191.168.1.3
PC-PT
PC9
2960-24TT
Switch3
2960-24TT
Switch1
PC-PT
PC6
126.168.1.2
PC-PT
PC7
126.168.1.3
PC-PT
128.168.0.2
PC-PT
128.168.0.3


Logical
Physical
x: 1008, y: 355
PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 128.168.0.2
Pinging 128.168.0.2 with 32 bytes of data:
Request timed out.
Reply from 128.168.0.2: bytes=32 time=6ms TTL=127
Reply from 128.168.0.2: bytes=32 time=13ms TTL=127
Reply from 128.168.0.2: bytes=32 time<1ms TTL=127
Ping statistics for 128.168.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 13ms, Average = 6ms
C:\>ping 128.168.0.3
Pinging 128.168.0.3 with 32 bytes of data:
Request timed out.
Reply from 128.168.0.3: bytes=32 time<1ms TTL=127
Reply from 128.168.0.3: bytes=32 time<1ms TTL=127
Reply from 128.168.0.3: bytes=32 time<1ms TTL=127
Ping statistics for 128.168.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.0.2


Realtime
Simulation
0
ete
Vindow
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
Successful PC2 PC0 ICMP 0.000 N 4 (edit) (delete)
Successful PC3 PC1 ICMP 0.000 N 5 (edit) (delete)
Successful PC0 PC2 ICMP 0.000 N 6 (edit) (delete)


Realtime
Simulati
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
Successful PC2 PC0 ICMP 0.000 N 4 (edit) (delete)
Successful PC3 PC1 ICMP 0.000 N 5 (edit) (delete)
Successful PC0 PC2 ICMP 0.000 N 6 (edit) (delete)

Experiment - 11

Name: Meet Patel **SapId:60004200104**

Batch: B1

Aim: To implement RIP in Packet Tracer

Theory:

Routing Information Protocol (RIP) is one of the oldest distance vector routing protocols, invented in the 1980s. Two versions of the protocol were developed:

Version 1 - supports only classful routing and doesn't send subnet masks in routing updates. Uses broadcasts for updates.

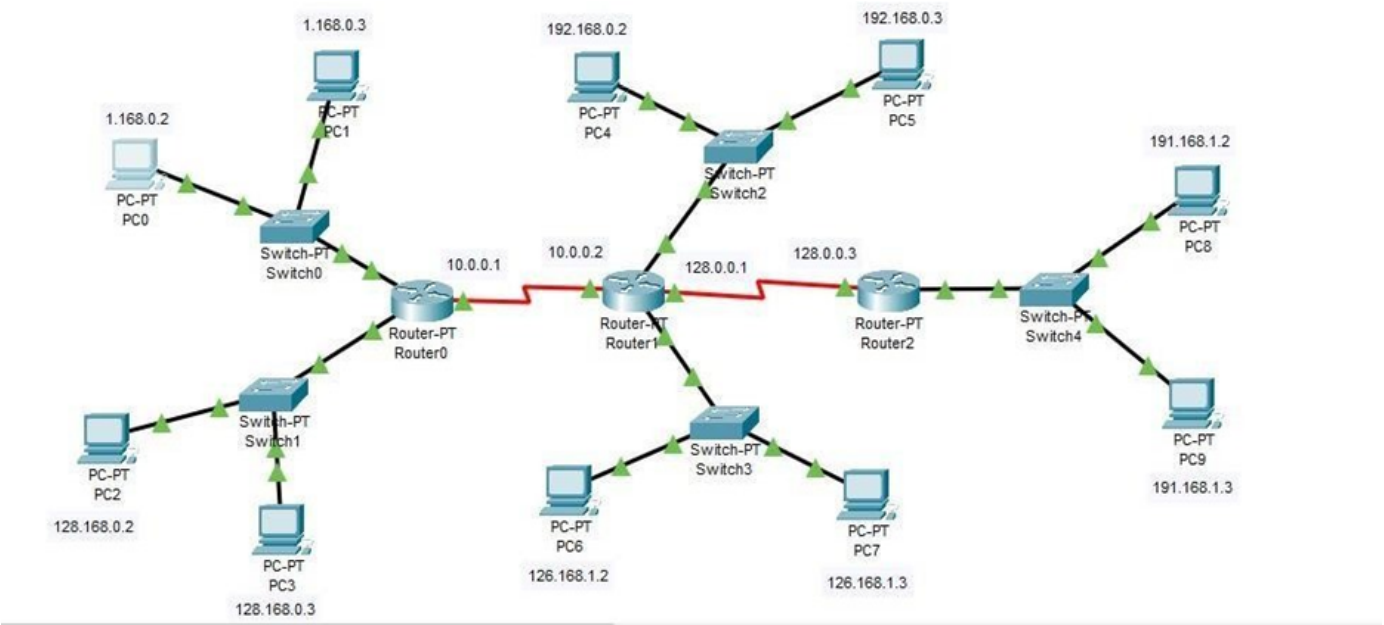
Version 2- supports classless routing and sends subnet masks in routing updates. This version uses the multicast address of 224.0.0.9to send routing updates.

There is also a version of RIP developed for IPv6 networks called RIPng.

RIP has a default administrative distance of 120. It uses the hop count (the number of routers between the source and destination network) as the metric. The hop count limit is

15. Any route with a higher hop count will be marked as unreachable.

Code:



Output:

Packet Tracer PC

Command Line 1.0

C:\>ping 128.168.0.2

Pinging 128.168.0.2 with 32

bytes of data: Request

timed out.

Reply from 128.168.0.2:
bytes=32 time<1ms TTL=127

Reply from 128.168.0.2:
bytes=32 time=16ms TTL=127
Reply from 128.168.0.2:
bytes=32 time=1ms TTL=127
Ping statistics for 128.168.0.2:

Packets: Sent = 4, Received = 3,
Lost = 1 (25% loss),
Approximate round trip times in
milli- seconds:

Minimum = 0ms, Maximum =
16ms, Average = 5ms
C:\>ping 128.168.0.3

Pinging 128.168.0.3 with 32

bytes of data: Request

timed out.

Reply from 128.168.0.3:
bytes=32 time=11ms TTL=127

Reply from 128.168.0.3:
bytes=32 time<1ms TTL=127

Reply from 128.168.0.3:
bytes=32 time<1ms TTL=127

Ping statistics for 128.168.0.3:

Packets: Sent = 4, Received = 3,
Lost = 1 (25% loss),
Approximate round trip times in
milli- seconds:

Minimum = 0ms, Maximum =
11ms, Average = 3ms

C:\>ping 192.168.0.2

Pinging 192.168.0.2 with 32

bytes of data: Request

timed out.

Reply from 192.168.0.2:
bytes=32 time=1ms TTL=126

Reply from 192.168.0.2:
bytes=32 time=1ms TTL=126

Reply from 192.168.0.2:
bytes=32 time=2ms TTL=126
Ping statistics for
192.168.0.2:

Packets: Sent = 4, Received = 3,
Lost = 1 (25% loss),
Approximate round trip times in
milli- seconds:

Minimum = 1ms, Maximum =
2ms, Average = 1ms

C:\>ping 192.168.0.3

Pinging 192.168.0.3 with 32

bytes of data: Request

timed out.

Reply from 192.168.0.3:
bytes=32 time=1ms TTL=126

Reply from 192.168.0.3:
bytes=32 time=15ms TTL=126
Reply from 192.168.0.3:

bytes=32 time=1ms TTL=126
Ping statistics for 192.168.0.3:

Packets: Sent = 4, Received = 3,
Lost = 1 (25% loss),
Approximate round trip times in
milli- seconds:

Minimum = 1ms, Maximum =
15ms, Average = 5ms

C:\>ping 126.168.1.2

Pinging 126.168.1.2 with 32

bytes of data: Request

timed out.

Reply from 126.168.1.2:
bytes=32 time=1ms TTL=126

Reply from 126.168.1.2:
bytes=32 time=14ms TTL=126
Reply from 126.168.1.2:
bytes=32 time=2ms TTL=126
Ping statistics for 126.168.1.2:

Packets: Sent = 4, Received = 3,
Lost = 1 (25% loss),
Approximate round trip times in
milli- seconds:

Minimum = 1ms, Maximum =
14ms, Average = 5ms

C:\>ping 126.168.1.3

Pinging 126.168.1.3 with 32

bytes of data: Request

timed out.

Reply from 126.168.1.3:
bytes=32 time=1ms TTL=126

Reply from 126.168.1.3:
bytes=32 time=24ms TTL=126

Reply from 126.168.1.3:
bytes=32 time=15ms TTL=126
Ping statistics for 126.168.1.3:

Packets: Sent = 4, Received = 3,
Lost = 1 (25% loss),
Approximate round trip times in
milli- seconds:

Minimum = 1ms, Maximum =
24ms, Average = 13ms

C:\>ping 191.168.1.2

Pinging 191.168.1.2 with 32 bytes of data:

Request timed out.

Reply from 191.168.1.2:
bytes=32 time=2ms TTL=125

Reply from 191.168.1.2: bytes=32 time=13ms TTL=125 Reply from 191.168.1.2: bytes=32 time=2ms TTL=125 Ping statistics for 191.168.1.2: Packets: Sent = 4, Received = 3, Lost = 1 (25%

loss), Approximate round trip times in milli- seconds:

Minimum = 2ms, Maximum = 13ms, Average = 5ms

C:\>ping 191.168.1.3

Pinging 191.168.1.3 with 32

bytes of data: Request

timed out.

Reply from 191.168.1.3: bytes=32 time=5ms TTL=125

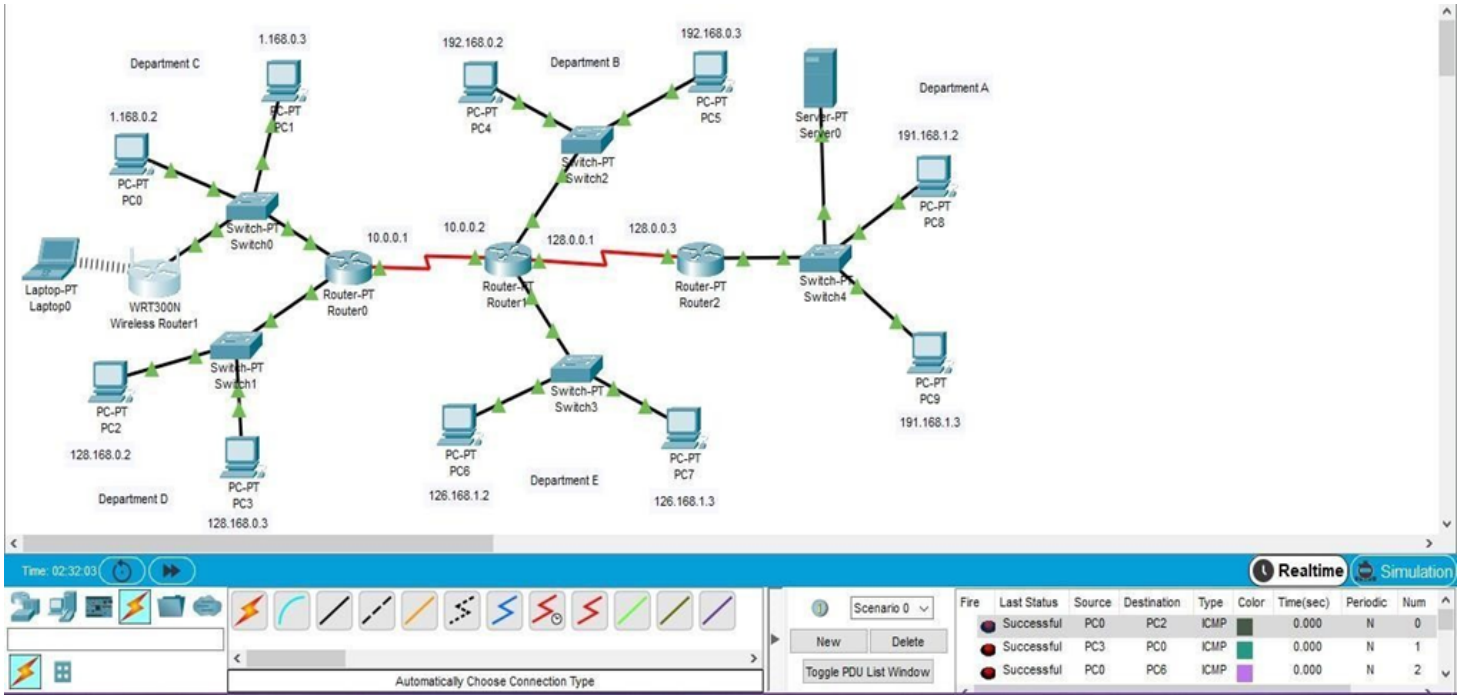
Reply from 191.168.1.3: bytes=32 time=4ms TTL=125

Reply from 191.168.1.3: bytes=32 time=2ms TTL=125
Ping statistics for 191.168.1.3:

Packets: Sent = 4, Received = 3, Lost = 1 (25% loss), Approximate round trip times in milli- seconds:

Minimum = 2ms, Maximum = 5ms, Average = 3ms

C:\>



Conclusion: Thus, we successfully implemented RIP packet tracer.