Computer Networks - Exp 1

Kartik Jolapara

60004200107 - B1

Aim

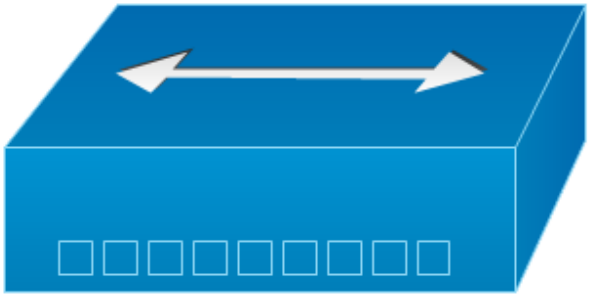
To study different networking devices and different networking topologies

Networking Devices

1. Hub

A hub is a common connection point, also known as a **network hub**, which is used for connection of devices in a network. It works as a central connection for all the devices that are connected through a hub. The hub has numerous ports. If a packet reaches one port, it is able to see all the segments of the network due to a packet being copied to the other ports. A network hub has no routing tables or intelligence (unlike a network switch or router), which is used to send information and broadcast all network data across each and every connection.

Logical Symbol



Working

Hubs work as a central connection between all network equipment and handle a data type, which is called **frames**. If a frame is received, it is transmitted to the port of the destination computer after amplifying it. A frame is passed to each of its ports in the hub, whether it is destined only for one port. It does not include the way of deciding a frame to which port it should be sent. Therefore, a frame has to transmit to **every port**, which ensures that it will reach its intended destination that generates a lot of traffic on the network and can be caused to damage the network. The hub is slower as compared to standard switch as it is not able to send or receive information at the same time, but a switch is more costly than a hub.

Advantages of Hub

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

Disadvantages of Hub

1. It has no ability to choose the best path of the network.
2. It does not include mechanisms such as collision detection.
3. It cannot reduce the network traffic as it has no mechanism.
4. It is not able to folder the information as it transmits packets to all the connected segments.

5. It is not capable of connecting various network architectures like a ring, token, and ethernet, and more.

Applications

1. Hub is used to create small home networks.
2. It is used for network monitoring.
3. They can be used in organizations to provide connectivity.
4. It can be used to create a device that is available throughout the network.

2. Switch

A **network switch** (also called switching hub, bridging hub) is a networking device 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. 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.

Logical Symbol

Working

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 lead 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 Switch

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

Disadvantages of Switch

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

Applications

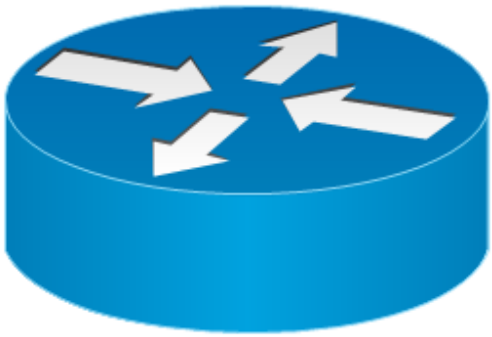
In larger networks, switches are often used as a way to offload traffic for analytic purposes. Nowadays, switches are used almost everywhere from **small office/home office** (SOHO) to **major ISPs** (Internet Service Providers). You can use them at your home office or small-sized area as you wish.

At its most basic, however, it is the simple task for a network switch to quickly and efficiently deliver packets from computer A to computer B, whether the computers are located across the hallway or halfway around the world. Several other devices contribute to this delivery along the way, but the switch is an essential part of the networking architecture.

3. Router

Routers are networking devices operating at layer 3 or a **network layer** of the OSI model. They are responsible for receiving, analyzing, and forwarding data packets among the connected computer networks. When a data packet arrives, the router inspects the **destination IP address**, consults its routing tables to decide the optimal route and then transfers the packet along this route.

Logical Symbol



Working

A router analyzes a destination IP address of a given packet header and compares it with the routing table to decide the packet's next path. The list of routing tables provides directions to transfer the data to a particular network destination. They have a set of rules that compute the best path to forward the data to the given IP address.

Routers use a **modem** such as a cable, fiber, or DSL modem to allow communication between other devices and the internet. Most of the routers have several ports to connect different devices to the internet at the same time. It uses the **routing tables** to determine where to send data and from where the traffic is coming. A routing table mainly defines the default path used by the router. So, it may fail to find the best way to forward the data for a given packet. For example, the office router along a single default path instructs all networks to its internet services provider.

There are two types of tables in the router that are **static and dynamic**. The static routing tables are configured manually, and the dynamic routing tables are updated automatically by dynamic routers based on network activity.

Advantages of Router

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 managers to make policy based on routing decisions.

Disadvantages of Router

1. They operate based on routable network protocols.
2. They are expensive compared to other network devices.

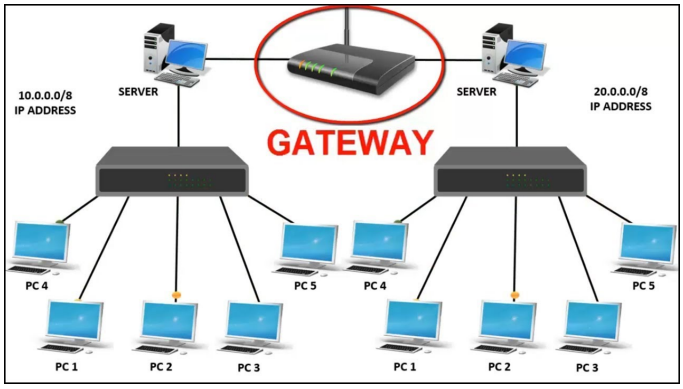
3. Dynamic router communications can cause additional network overhead. This results in less bandwidth for user data.
4. They are slower as they need to analyze data from layer-1 through layer-3.
5. They require a considerable amount of initial configurations.
6. They are protocol dependent devices which must understand the protocol they are forwarding.

Applications

Routers may also be used to connect two or more logical groups of computer devices known as subnets, each with a different network prefix. Routers may provide connectivity within enterprises, between enterprises and the Internet, or between internet service providers' (ISPs') networks.

The largest routers (such as the Cisco CRS-1 or Juniper PTX) interconnect the various ISPs, or may be used in large enterprise networks. Smaller routers usually provide connectivity for typical home and office networks. All sizes of routers may be found inside enterprises. The most powerful routers are usually found in ISPs, academic and research facilities. Large businesses may also need more powerful routers to cope with ever-increasing demands of intranet data traffic.

4. Gateway



A gateway is a network node that forms a passage between two networks operating with different transmission protocols. The most common type of gateways, the network gateway operates at layer 3, i.e. network layer of the OSI (open systems interconnection) model. However, depending upon the functionality, a gateway can operate at any of the seven layers of OSI model. It acts as the entry – exit point for a network since all traffic that flows across the networks should pass through the

gateway. Only the internal traffic between the nodes of a LAN does not pass through the gateway.

Logical Symbol

Working

It is a state of a network that can get to different networks. Typically, in the intranet, a node or router can go about as a router or the gateway node that interfaces the networks are called gateways. In big companies, the PCs that deal with the traffic between enterprise networks are named gateway nodes. For example, the PCs utilized by Internet service providers to connect fluctuated users at the moment time to the web are gateway nodes.

It very well may be connected to the router since a router precisely thinks about the routing path of data packets that shows up at the gateway, then a switch chooses in the reasonable in and out the way of the gateway for the assigned packet. The gateway is a required trait of courses even though different devices can act well as a gateway.

Advantages of Gateway

1. It can connect the devices of two different networks having dissimilar structures.
2. It is an intelligent device with filtering capabilities.
3. It has control over both collisions as well as a broadcast domain.
4. It uses a full-duplex mode of communication.
5. It has the fastest data transmission speed amongst all network connecting devices.
6. It can perform data translation and protocol conversion of the data packet as per the destination network's need.
7. It can encapsulate and decapsulate the data packets.
8. It has improved security than any other network connecting device.

Disadvantages of Gateway

1. It is complex to design and implement.
2. The implementation cost is very high.
3. It requires a special system administration configuration.

Applications

A typical function of a gateway is a gateway to connect from the user’s tenant to the normal capacity pool. It is given as a virtual switch devoted to users on repetitive equipment. A gateway server permits clients to approach their hosted webpage without approaching the remainder of the internet. It advances filtered solicitations to end-points however obstructs the wide range of various solicitations. This implies that when a client demands.

An IP gateway alludes to a device on a network that sends nearby network traffic to different networks. The subnet veil number helps characterize the connection between the host (switches, routers, computers, and so on) and the remainder of the network.

5. Repeater

A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network.
An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

Logical Symbol

Working

When an electrical signal is broadcasted through a channel, then it gets attenuated based on the nature of technology. This deploys a limitation depending upon the length of the LAN network. This issue is created by embedding the repeaters at the specific intervals.

Repeater gets to amplify the attenuated signal then retransmits it. Repeaters are getting popular for being incorporated to link between two small LAN and large LAN networks.

Advantages of Repeater

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

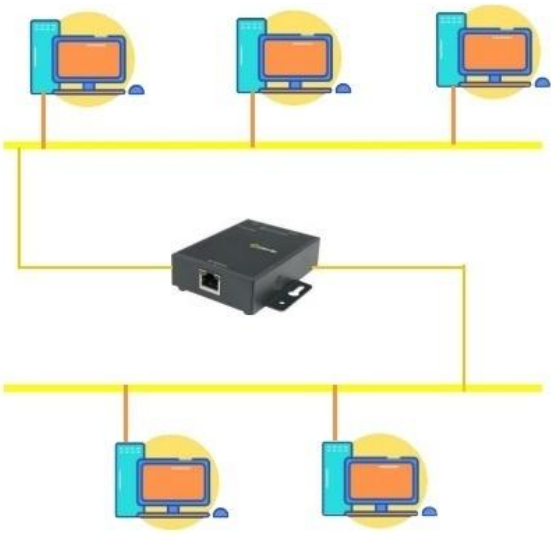
Disadvantages of Repeater

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

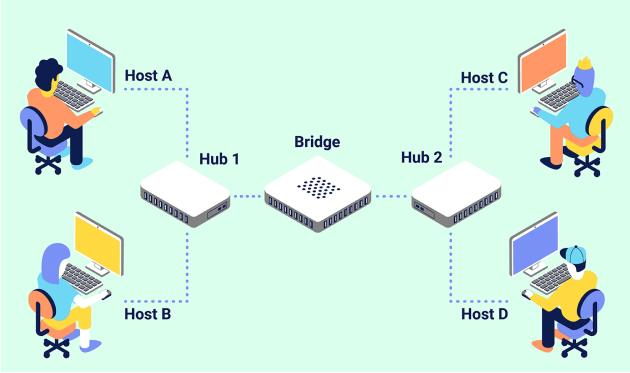
Applications

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 retransmits 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. This is shown in the following diagram



6. 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 and hence are also referred to as Layer 2 switches.

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 of Bridge

1. It reduces network traffic with minor segmentation
2. It reduces collisions
3. Bridge connects similar network types with different cabling

4. Bridge increase the number of attached workstation and network segments
5. It extends the physical network
6. Bridges also can reduce network traffic on a segment by subdividing network communications
7. It connects different architecture

Disadvantages of Bridge

1. It does not filter broadcasts
2. It is slower compare to repeaters due to the filtering process
3. It is more expensive compared to repeaters
4. Complex network topology, it can pose a problem for transparent bridge
5. A bridge is more expensive than repeaters or hubs'

Applications

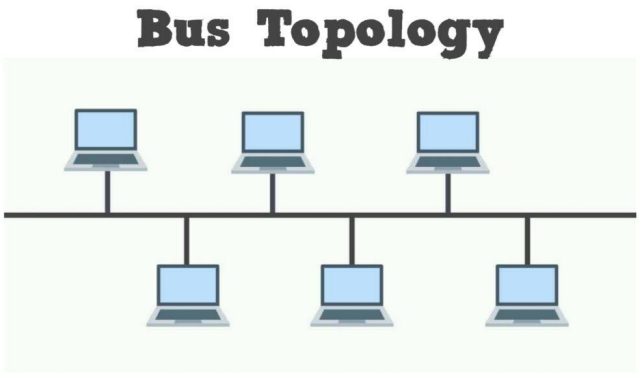
- Bridges are used to divide large busy networks into multiple smaller and interconnected networks to improve performance.
- Bridges also can increase the physical size of a network.
- Bridges are also used to connect a LAN segment through a synchronous modem relation to another LAN segment at a remote area.

Topologies

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**



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.

Advantages of Bus Topology

1. Works efficiently for small networks
2. Easy and cost-effective to install and add or remove devices
3. Doesn't require as much cabling as alternative topologies
4. If one device fails, other devices are not impacted

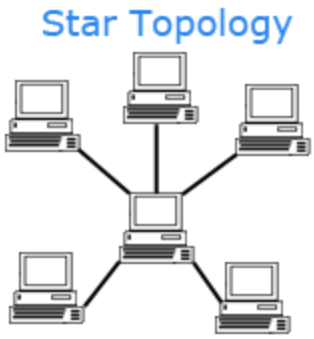
Disadvantages of Bus Topology

1. If the cable is damaged, the entire network will fail or be split
2. Difficult to troubleshoot problems
3. Very slow and not ideal for larger networks
4. Requires terminators at both ends of the cable to prevent bouncing signals that cause interference
5. Adding more devices and more network traffic decreases the entire network's performance
6. Low security due to all devices receiving the same signal from the source

Applications

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

2. Star



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. Active hubs have repeaters in them.

Advantages of Star Topology

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 of Star Topology

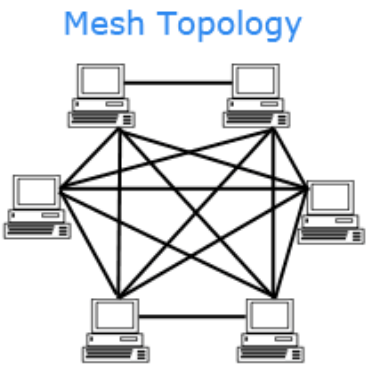
1. If the concentrator (hub) on which the whole topology relies fails, the whole system will crash down.
2. The cost of installation is high.
3. Performance is based on the single concentrator i.e. hub.

Applications

- Star topology is used in Local Area Network (LAN)
- High speed LAN often uses the star topology.
- Star topology is often used in homes and offices.

- Star topology is also used to transmit data along the central hub between the network nodes.
- By connecting all the systems to the central hubs, star topology can ease the probability of network failure.

3. Mesh



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.

Suppose, N number of devices are connected with each other in a mesh topology, the total number of ports that are required by each device is N-1. Suppose, N number of devices are connected with each other in a mesh topology, then the total number of dedicated links required to connect them is NC2.

Advantages of Mesh Topology

1. It is robust.
2. The fault is diagnosed easily. Data is reliable because data is transferred among the devices through dedicated channels or links.
3. Provides security and privacy.

Disadvantages of Mesh Topology

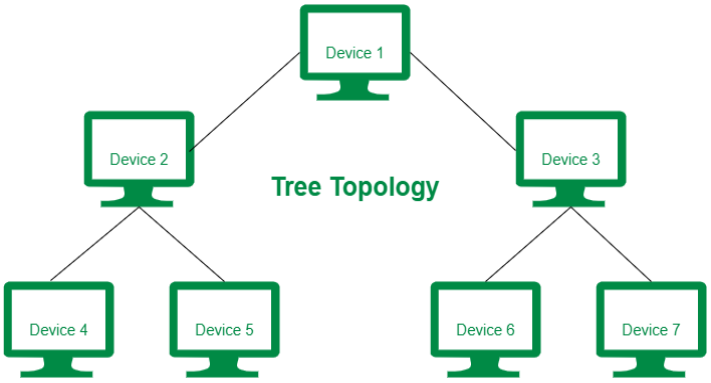
1. Installation and configuration are difficult.
2. The cost of cables is high as bulk wiring is required, hence suitable for less number of devices.
3. The cost of maintenance is high.

Applications

- **Home monitoring and control**: It's a snap to turn lights off and on or dim them.
- **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.
- **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.

4. Tree

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. This data flows 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.

Advantages of Tree Topology

1. It allows more devices to be attached to a single central hub thus it decreases the distance that is traveled by the signal to come to the devices.
2. It allows the network to isolate and also prioritize different computers.

Disadvantages of Tree Topology

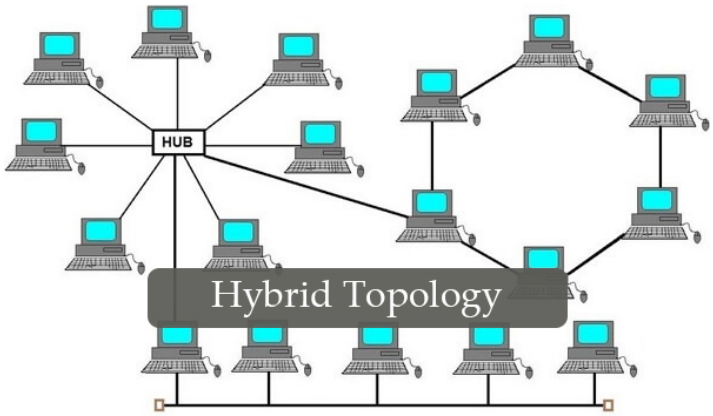
1. If the central hub fails the entire system fails.
2. The cost is high because of cabling.

Applications

- When you have a multi-story building and wish to establish clusters at each section of the network, you can utilize tree topology.
- If you have departments and sub-departments, you can segregate the whole Tree Network with the help of several switches that makes the entire network easy to maintain and more manageable.

5. Hybrid

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.

Advantages of Hybrid Topology

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 of Hybrid Topology

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

Applications

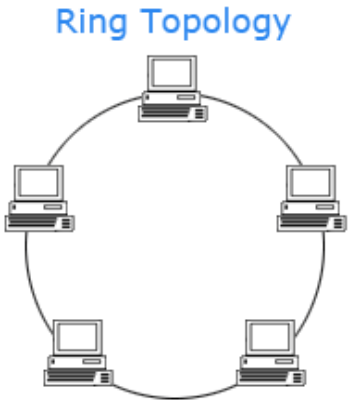
The examples and applications of hybrid topology are increasing rapidly. It has a super-power set up and flexible option and declared as a smart option; hence, the people choose to deploy it in-home or office. A compact is provided for the small-scale industries by this topology, as well as to their subunits. Thus, it is good to use for multi-floor buildings and departments such as an office or home. This

topology is placed to give its maximum efficiency on the basis of the requirements as it provides many benefits.

6. Ring

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

One station is known as a monitor station which takes all the responsibility to perform the operations. To transmit the data, the station has to hold the token. After the transmission is done, the token is to be released for other stations to use. When no station is transmitting the data, then the token will circulate in the ring. There are two types of token release techniques: Early token release releases the token just after transmitting the data and Delay token release releases the token after the acknowledgment is received from the receiver.

Advantages of Ring Topology

1. The possibility of collision is minimum in this type of topology.
2. Cheap to install and expand.

Disadvantages of Hybrid Topology

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.

Applications

- It is used in the Wide Area Network (WAN) and in the Metropolitan Area Network (MAN) is used in vast areas for connecting all the (LAN) it is also used in-ring networks, also the most crucial part in a ring topology.
- 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.

Computer Networks - Exp 2

Kartik Jolapara

60004200107 - B1

Aim

To study and execute different networking commands

Commands

1. 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\Greha>**ipconfig**

Windows IP Configuration

Unknown adapter ProtonVPN TUN:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Ethernet adapter vEthernet (WSL):

Connection-specific DNS Suffix . :

Link-local IPv6 Address . . . . . : fe80::494c:b6fa:cc57:d035%44
IPv4 Address. . . . . : 172.20.112.1
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . :

Unknown adapter Local Area Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 10:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 11:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::6152:e1cb:609:f795%19
IPv4 Address. . . . . : 192.168.0.102
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1

2. 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 basic TCP/IP configuration for all adapters

C:\Users\Greha>**ipconfig -all**

Windows IP Configuration

Host Name . . . . . : DESKTOP-DC7H32B
Primary Dns Suffix . . . . . :

Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Unknown adapter ProtonVPN TUN:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : ProtonVPN Tunnel
Physical Address. . . . . :
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes

Ethernet adapter vEthernet (WSL):

Connection-specific DNS Suffix . :
Description . . . . . : Hyper-V Virtual Ethernet Adapter
Physical Address. . . . . : 00-15-5D-2D-16-84
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::494c:b6fa:cc57:d035%44(Preferred)
IPv4 Address. . . . . : 172.20.112.1(Preferred)
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 738202973
DHCPv6 Client DUID. . . . . : 00-01-00-01-28-F7-08-4D-64-6C-80-53-98-47
NetBIOS over Tcpip. . . . . : Enabled

Unknown adapter Local Area Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : TAP-ProtonVPN Windows Adapter V9
Physical Address. . . . . : 00-FF-67-A2-02-8A
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection* 10:

Media State . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter

Physical Address. . . . . : 66-6C-80-53-98-47

DHCP Enabled. . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection* 11:

Media State . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2

Physical Address. . . . . : 76-6C-80-53-98-47

DHCP Enabled. . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . :

Description . . . . . : Qualcomm QCA61x4A 802.11ac Wireless Adapter

Physical Address. . . . . : 64-6C-80-53-98-47

DHCP Enabled. . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Link-local IPv6 Address . . . . . : fe80::6152:e1cb:609:f795%19(Preferred)

IPv4 Address. . . . . : 192.168.0.102(Preferred)

Subnet Mask . . . . . : 255.255.255.0

Lease Obtained. . . . . : Sunday, April 3, 2022 9:16:38 PM

Lease Expires . . . . . : Monday, April 4, 2022 4:17:42 AM

Default Gateway . . . . . : 192.168.0.1

DHCP Server . . . . . : 192.168.0.1

DHCPv6 IAID . . . . . : 291794048

DHCPv6 Client DUID. . . . . : 00-01-00-01-28-F7-08-4D-64-6C-80-53-98-47

DNS Servers . . . . . : 192.168.0.1

NetBIOS over Tcpip. . . . . : Enabled

3. 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\Greha>**ping 192.168.0.102**
Pinging 192.168.0.102 with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.102:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

4. 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\Greha>**ping -t 192.168.0.102**
Pinging 192.168.0.102 with 32 bytes of data:

Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128
Reply from 192.168.0.102: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.102:

Packets: Sent = 12, Received = 12, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

Control-C

^C

5. 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, and 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\Greha>**netstat**

Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:49670	DESKTOP-DC7H32B:49671	ESTABLISHED

```
TCP    127.0.0.1:49671        DESKTOP-DC7H32B:49670  ESTABLISHED
TCP    127.0.0.1:49672        DESKTOP-DC7H32B:49673  ESTABLISHED
TCP    127.0.0.1:49673        DESKTOP-DC7H32B:49672  ESTABLISHED
TCP    127.0.0.1:54444        DESKTOP-DC7H32B:54445  ESTABLISHED
TCP    127.0.0.1:54445        DESKTOP-DC7H32B:54444  ESTABLISHED
TCP    127.0.0.1:58398        DESKTOP-DC7H32B:58399  ESTABLISHED
TCP    127.0.0.1:58399        DESKTOP-DC7H32B:58398  ESTABLISHED
TCP    127.0.0.1:58400        DESKTOP-DC7H32B:58401  ESTABLISHED
TCP    127.0.0.1:58401        DESKTOP-DC7H32B:58400  ESTABLISHED
TCP    127.0.0.1:58402        DESKTOP-DC7H32B:58403  ESTABLISHED
TCP    127.0.0.1:58403        DESKTOP-DC7H32B:58402  ESTABLISHED
TCP    127.0.0.1:58404        DESKTOP-DC7H32B:58405  ESTABLISHED
TCP    127.0.0.1:58405        DESKTOP-DC7H32B:58404  ESTABLISHED
TCP    127.0.0.1:61391        DESKTOP-DC7H32B:65001  ESTABLISHED
TCP    127.0.0.1:63143        DESKTOP-DC7H32B:63144  ESTABLISHED
TCP    127.0.0.1:63144        DESKTOP-DC7H32B:63143  ESTABLISHED
TCP    127.0.0.1:65001        DESKTOP-DC7H32B:61391  ESTABLISHED
TCP    192.168.0.102:49461    20.197.71.89:443       ESTABLISHED
TCP    192.168.0.102:49927    25:443                 TIME_WAIT
TCP    192.168.0.102:49930    ec2-52-10-149-213:443  TIME_WAIT
TCP    192.168.0.102:49931    ec2-52-10-149-213:443  TIME_WAIT
TCP    192.168.0.102:49933    200:443                TIME_WAIT
TCP    192.168.0.102:49935    ec2-52-10-149-213:443  TIME_WAIT
TCP    192.168.0.102:49936    ec2-52-10-149-213:443  TIME_WAIT
TCP    192.168.0.102:49937    162.125.69.19:443      ESTABLISHED
TCP    192.168.0.102:49939    200:443                TIME_WAIT
TCP    192.168.0.102:49940    ec2-52-10-149-213:443  TIME_WAIT
TCP    192.168.0.102:49941    ec2-52-10-149-213:443  TIME_WAIT
TCP    192.168.0.102:49942    151.101.154.248:443    ESTABLISHED
TCP    192.168.0.102:49943    ec2-35-81-100-74:443   ESTABLISHED
TCP    192.168.0.102:49944    ec2-35-81-100-74:443   ESTABLISHED
TCP    192.168.0.102:50464    162.125.19.9:443       ESTABLISHED
TCP    192.168.0.102:50546    219:443                ESTABLISHED
```

6. netstat -an

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

C:\Users\Greha>**netstat -an**

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5700	0.0.0.0:0	LISTENING
TCP	0.0.0.0:6646	0.0.0.0:0	LISTENING
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING
TCP	0.0.0.0:17500	0.0.0.0:0	LISTENING
TCP	0.0.0.0:27121	0.0.0.0:0	LISTENING
TCP	0.0.0.0:33060	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49674	0.0.0.0:0	LISTENING
TCP	127.0.0.1:843	0.0.0.0:0	LISTENING
TCP	127.0.0.1:6463	0.0.0.0:0	LISTENING
TCP	127.0.0.1:8884	0.0.0.0:0	LISTENING
TCP	127.0.0.1:9012	0.0.0.0:0	LISTENING
TCP	127.0.0.1:17600	0.0.0.0:0	LISTENING
TCP	127.0.0.1:27017	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49670	127.0.0.1:49671	ESTABLISHED
TCP	127.0.0.1:49671	127.0.0.1:49670	ESTABLISHED

```
TCP    127.0.0.1:49672        127.0.0.1:49673        ESTABLISHED
TCP    127.0.0.1:49673        127.0.0.1:49672        ESTABLISHED
TCP    127.0.0.1:49702        0.0.0.0:0              LISTENING
TCP    127.0.0.1:54444        127.0.0.1:54445        ESTABLISHED
TCP    127.0.0.1:54445        127.0.0.1:54444        ESTABLISHED
TCP    127.0.0.1:58398        127.0.0.1:58399        ESTABLISHED
TCP    127.0.0.1:58399        127.0.0.1:58398        ESTABLISHED
TCP    127.0.0.1:58400        127.0.0.1:58401        ESTABLISHED
TCP    127.0.0.1:58401        127.0.0.1:58400        ESTABLISHED
TCP    127.0.0.1:58402        127.0.0.1:58403        ESTABLISHED
TCP    127.0.0.1:58403        127.0.0.1:58402        ESTABLISHED
TCP    127.0.0.1:58404        127.0.0.1:58405        ESTABLISHED
TCP    127.0.0.1:58405        127.0.0.1:58404        ESTABLISHED
TCP    127.0.0.1:61391        127.0.0.1:65001        ESTABLISHED
TCP    127.0.0.1:61589        0.0.0.0:0              LISTENING
TCP    127.0.0.1:63143        127.0.0.1:63144        ESTABLISHED
TCP    127.0.0.1:63144        127.0.0.1:63143        ESTABLISHED
TCP    127.0.0.1:65001        0.0.0.0:0              LISTENING
TCP    127.0.0.1:65001        127.0.0.1:61391        ESTABLISHED
TCP    172.20.112.1:139       0.0.0.0:0              LISTENING
TCP    192.168.0.102:139      0.0.0.0:0              LISTENING
TCP    192.168.0.102:49461    20.197.71.89:443       ESTABLISHED
TCP    192.168.0.102:49948    35.81.100.74:443       TIME_WAIT
TCP    192.168.0.102:49949    35.81.100.74:443       TIME_WAIT
TCP    192.168.0.102:49950    13.67.9.5:443          TIME_WAIT
TCP    192.168.0.102:49954    35.81.100.74:443       TIME_WAIT
TCP    192.168.0.102:49955    35.81.100.74:443       TIME_WAIT
TCP    192.168.0.102:49958    20.54.24.246:443       TIME_WAIT
TCP    192.168.0.102:49959    20.54.24.246:443       ESTABLISHED
TCP    192.168.0.102:49961    35.81.100.74:443       TIME_WAIT
TCP    192.168.0.102:49962    35.81.100.74:443       TIME_WAIT
TCP    192.168.0.102:49964    35.81.100.74:443       TIME_WAIT
TCP    192.168.0.102:49965    35.81.100.74:443       TIME_WAIT
```

```
TCP    192.168.0.102:49967    13.107.213.68:443      ESTABLISHED
TCP    192.168.0.102:49969    35.197.154.200:443     TIME_WAIT
TCP    192.168.0.102:49971    52.114.16.141:443      TIME_WAIT
TCP    192.168.0.102:49972    54.149.54.1:443        TIME_WAIT
TCP    192.168.0.102:49973    54.149.54.1:443        TIME_WAIT
TCP    192.168.0.102:49974    35.197.154.200:443     TIME_WAIT
TCP    192.168.0.102:49975    35.82.117.62:443       TIME_WAIT
TCP    192.168.0.102:49976    35.186.224.13:443      TIME_WAIT
TCP    192.168.0.102:49978    35.82.117.62:443       TIME_WAIT
TCP    192.168.0.102:49980    13.69.116.104:443      TIME_WAIT
TCP    192.168.0.102:49981    13.69.116.104:443      TIME_WAIT
TCP    192.168.0.102:49982    54.149.54.1:443        ESTABLISHED
TCP    192.168.0.102:49983    54.149.54.1:443        ESTABLISHED
TCP    192.168.0.102:49984    35.186.224.13:443      ESTABLISHED
TCP    192.168.0.102:49985    35.186.224.25:443      ESTABLISHED
TCP    192.168.0.102:49986    151.101.154.248:443    ESTABLISHED
TCP    192.168.0.102:49987    151.101.154.248:443    ESTABLISHED
TCP    192.168.0.102:49988    151.101.154.248:443    ESTABLISHED
TCP    192.168.0.102:50464    162.125.19.9:443       ESTABLISHED
TCP    192.168.0.102:50546    35.247.144.219:443     ESTABLISHED
TCP    192.168.0.102:50892    35.186.224.39:443      ESTABLISHED
TCP    192.168.0.102:54328    162.159.130.234:443    ESTABLISHED
TCP    192.168.0.102:56329    35.186.224.47:443      ESTABLISHED
TCP    192.168.0.102:56334    54.159.116.102:443     ESTABLISHED
TCP    192.168.0.102:56336    23.98.104.194:443      ESTABLISHED
TCP    192.168.0.102:56338    20.197.71.89:443       ESTABLISHED
TCP    192.168.0.102:57035    52.114.32.217:443      ESTABLISHED
TCP    192.168.0.102:57036    13.76.153.29:443       ESTABLISHED
TCP    192.168.0.102:60293    162.125.19.131:443     ESTABLISHED
TCP    192.168.0.102:60854    31.13.79.53:443        ESTABLISHED
TCP    192.168.0.102:63391    35.186.224.25:443      TIME_WAIT
TCP    192.168.0.102:64232    52.114.14.201:443      ESTABLISHED
TCP    192.168.0.102:65206    104.40.53.219:443      CLOSE_WAIT
```

```
TCP    192.168.0.102:65211    52.177.138.113:443     CLOSE_WAIT
TCP    [::]:135               [::]:0                 LISTENING
TCP    [::]:445               [::]:0                 LISTENING
TCP    [::]:3306              [::]:0                 LISTENING
TCP    [::]:5700              [::]:0                 LISTENING
TCP    [::]:7680              [::]:0                 LISTENING
TCP    [::]:17500             [::]:0                 LISTENING
TCP    [::]:27121             [::]:0                 LISTENING
TCP    [::]:33060             [::]:0                 LISTENING
TCP    [::]:49664             [::]:0                 LISTENING
TCP    [::]:49665             [::]:0                 LISTENING
TCP    [::]:49666             [::]:0                 LISTENING
TCP    [::]:49667             [::]:0                 LISTENING
TCP    [::]:49668             [::]:0                 LISTENING
TCP    [::]:49674             [::]:0                 LISTENING
TCP    [::1]:49669            [::]:0                 LISTENING
UDP    0.0.0.0:53             *:*
UDP    0.0.0.0:500            *:*
UDP    0.0.0.0:4500           *:*
UDP    0.0.0.0:5050           *:*
UDP    0.0.0.0:5353           *:*
UDP    0.0.0.0:5355           *:*
UDP    0.0.0.0:6646           *:*
UDP    0.0.0.0:17500          *:*
UDP    0.0.0.0:51205          *:*
UDP    0.0.0.0:51206          *:*
UDP    0.0.0.0:51894          *:*
UDP    0.0.0.0:54227          *:*
UDP    0.0.0.0:54868          *:*
UDP    0.0.0.0:54939          142.251.42.42:443
UDP    0.0.0.0:57229          *:*
UDP    0.0.0.0:58611          162.159.135.232:443
UDP    0.0.0.0:59157          *:*
```

```
UDP    0.0.0.0:59867          *:*
UDP    0.0.0.0:59911          *:*
UDP    0.0.0.0:61404          *:*
UDP    0.0.0.0:63701          142.250.183.78:443
UDP    127.0.0.1:1900         *:*
UDP    127.0.0.1:10040        *:*
UDP    127.0.0.1:49664        127.0.0.1:49664
UDP    127.0.0.1:51785        *:*
UDP    127.0.0.1:53104        *:*
UDP    172.20.112.1:137       *:*
UDP    172.20.112.1:138       *:*
UDP    172.20.112.1:1900      *:*
UDP    172.20.112.1:2177      *:*
UDP    172.20.112.1:5353      *:*
UDP    172.20.112.1:51783     *:*
UDP    192.168.0.102:137      *:*
UDP    192.168.0.102:138      *:*
UDP    192.168.0.102:1900     *:*
UDP    192.168.0.102:2177     *:*
UDP    192.168.0.102:5353     *:*
UDP    192.168.0.102:51784    *:*
UDP    [::]:500               *:*
UDP    [::]:4500              *:*
UDP    [::]:5353              *:*
UDP    [::]:5355              *:*
UDP    [::]:51207             *:*
UDP    [::]:51894             *:*
UDP    [::]:54227             *:*
UDP    [::]:54868             *:*
UDP    [::]:57229             *:*
UDP    [::]:59158             *:*
UDP    [::]:59867             *:*
UDP    [::]:59911             *:*
```

```
UDP    [::]:61404             *:*
UDP    [::1]:1900             *:*
UDP    [::1]:5353             *:*
UDP    [::1]:51782            *:*
UDP    [fe80::494c:b6fa:cc57:d035%44]:1900  *:*
UDP    [fe80::494c:b6fa:cc57:d035%44]:2177  *:*
UDP    [fe80::494c:b6fa:cc57:d035%44]:51780  *:*
UDP    [fe80::6152:e1cb:609:f795%19]:1900  *:*
UDP    [fe80::6152:e1cb:609:f795%19]:2177  *:*
UDP    [fe80::6152:e1cb:609:f795%19]:51781  *:*
```

7. 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

Path pinging www.mu.ac.in

C:\Users\Greha>**pathping www.mu.ac.in**

Tracing route to www.mu.ac.in [14.139.125.195]

over a maximum of 30 hops:

```
 0  DESKTOP-DC7H32B [192.168.0.102]
 1  192.168.0.1
 2  100.93.152.1
 3  114.79.129.57.dvois.com [114.79.129.57]
 4    *        *        *
```

Computing statistics for 75 seconds...

^C

8. 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\Greha>**arp -a**

Interface: 192.168.0.102 --- 0x13

Internet Address	Physical Address	Type
192.168.0.1	b0-be-76-41-f4-f2	dynamic
192.168.0.122	62-a4-b7-09-91-62	dynamic
192.168.0.200	62-a4-b7-09-91-62	dynamic
192.168.0.255	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
255.255.255.255	ff-ff-ff-ff-ff-ff	static

Interface: 172.20.112.1 --- 0x2c

Internet Address	Physical Address	Type
172.20.127.255	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
255.255.255.255	ff-ff-ff-ff-ff-ff	static

9. 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\Greha>**nslookup**
Default Server: UnKnown
Address: 192.168.0.1

>

Computer Networks - Exp 3

Kartik Jolapara

60004200107 - 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.

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;
}
```

Output

```
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
```

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 the 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

```
#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

```
Kartik:CN Exp 3 -  CRC & Hamming/ (master) $ ./Hamming.exe
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 successfullyKartik:CN Exp 3 -  CRC & Hamming/ (master) $
```

Conclusion

Thus we successfully implemented Error detection and correction techniques.

Computer Networks - Exp 4

Kartik Jolapara

60004200107 - B1

Aim

To implement Djikstra’s shortest path algorithm.

Theory

The 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. The 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. The Dijkstra algorithm works for directed as well as undirected graphs. Working of Dijkstra's 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>
```

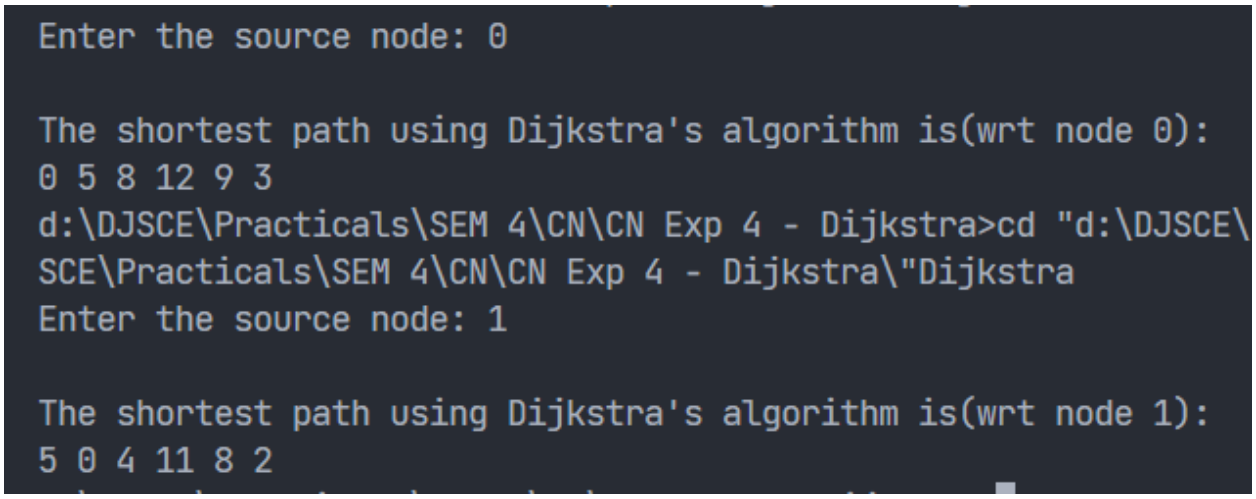
```
#include <stdbool.h>

#define MIN(x, y) (((x < y) ? x : y))

int main()
{
    int n = 6;
    int cost[6][6] = {
        {0, 7, 42069, 42069, 42069, 3},
        {7, 0, 4, 42069, 42069, 2},
        {42069, 4, 0, 8, 5, 5},
        {42069, 42069, 8, 0, 3, 42069},
        {42069, 42069, 5, 3, 0, 6},
        {3, 2, 5, 42069, 6, 0}};
    int vis[6] = {0, 0, 0, 0, 0, 0};
    int d[6];
    int source;
    printf("Enter the source node: ");
    scanf("%d", &source);
    vis[source] = 1;
```

```
for (int i = 0; i < n; i++)
{
    d[i] = cost[source][i];
}

// for (int i = 0; i < n; i++)
// {
//     printf("%d HEREEE %d\n", d[i], vis[i]);
// }
for (int i = 0; i < 6; i++)
{
    int min, check = 1;
    for (int j = 0; j < 6; j++)
    {
        if (vis[j] == 0 && check == 1)
        {
            min = j;
            check = 0;
        }
        if (vis[j] == 0 && d[j] < d[min])
        {
```

```
            min = j;
        }
        // printf("%d ", d[j]);
    }
    // printf("\n");
    // printf("ANS: %d\n", min);
    vis[min] = 1;
    for (int j = 0; j < 6; j++)
    {
        // d[j] = (d[j] < d[min] + cost[min][j]) ? d[j] : (d[min] + cost[min][j]);
        d[j] = MIN(d[j], (d[min] + cost[min][j]));
    }
    // for (int j = 0; j < n; j++)
    // {
    //     printf("D%d ", d[j]);
    // }
    // printf("\n");
}
printf("\n");
printf("The shortest path using Dijkstra's algorithm is(wrt node %d): \n", source);
for (int i = 0; i < n; i++)
```

```
    {
        printf("%d ", d[i]);
    }

    return 0;
}
```

Output



Conclusion

Dijkstra’s shortest path algorithm has been successfully executed.

Computer Networks - Exp 5

Kartik Jolapara

60004200107 - 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 in the 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 the receiver removes this ESC and keeps the data part that causes some problems or issues. In simple words, we can say that character stuffing is an 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 a transmission unit or message to be transmitted as a simple way to provide and give signaling information and data to the receiver and to avoid or ignore appearance of unintended or unnecessary control sequences. It is a type of protocol management simply performed to break up a bit pattern that results in transmission to go out of synchronization. Bit stuffing is a very essential part of the 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. Starting and ending with character stuffing\n3. Character stuffing\n4. Starting and ending Flag, with bit stuffng.\n5. 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";
    finalData = stx + dle + data + dle + stx;
    cout << "\nFinal data: " << finalData << "\n\n";
    break;
}
case 3:
{
    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 4:
{
    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 5:
        {
            cout << "Exiting..!\n";
            break;
        }
        default:
        {
            cout << "Enter an valid OPTION!!";
        }
        }
    } while (opt != 5);
    return 0;
}
```

Output

```
1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 1
Enter the number of frames: 4
Enter frame data: 12
Enter frame data: 3451
Enter frame data: 25
Enter frame data: 123563

Final data: 212434512256123563

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 2
Enter the data: fjiem#T53jkfmf88

Final data: STXDLEfjiem#T53jkfmf88DLESTX

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit
```

```
Enter an option: 3
Enter the data: fsjSml4joSTXfjldemDLEfes

Final data: STXDLEfsjSml4joSTXfjldemDLEDLEfesDLESTX

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 4
Enter the data: 1101100011101110101100111111000111111111

Final data: 0111111011011011101110101101111101111011101111110

1. Character count
2. Starting and ending with character stuffing
3. Character stuffing
4. Starting and ending Flag, with bit stuffng.
5. Exit

Enter an option: 5
Exiting..!
```

Conclusion

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

Computer Networks - Exp 6

Kartik Jolapara

60004200107 - 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 another socket reaches out to the other to form a connection. Server forms the listener socket while the 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
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 a live audience as possible without worrying about the loss of one or two frames. TCP being a reliable protocol adds 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 use a careful blend of both UDP and TCP; transmitting the critical data over TCP and the 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) {
```

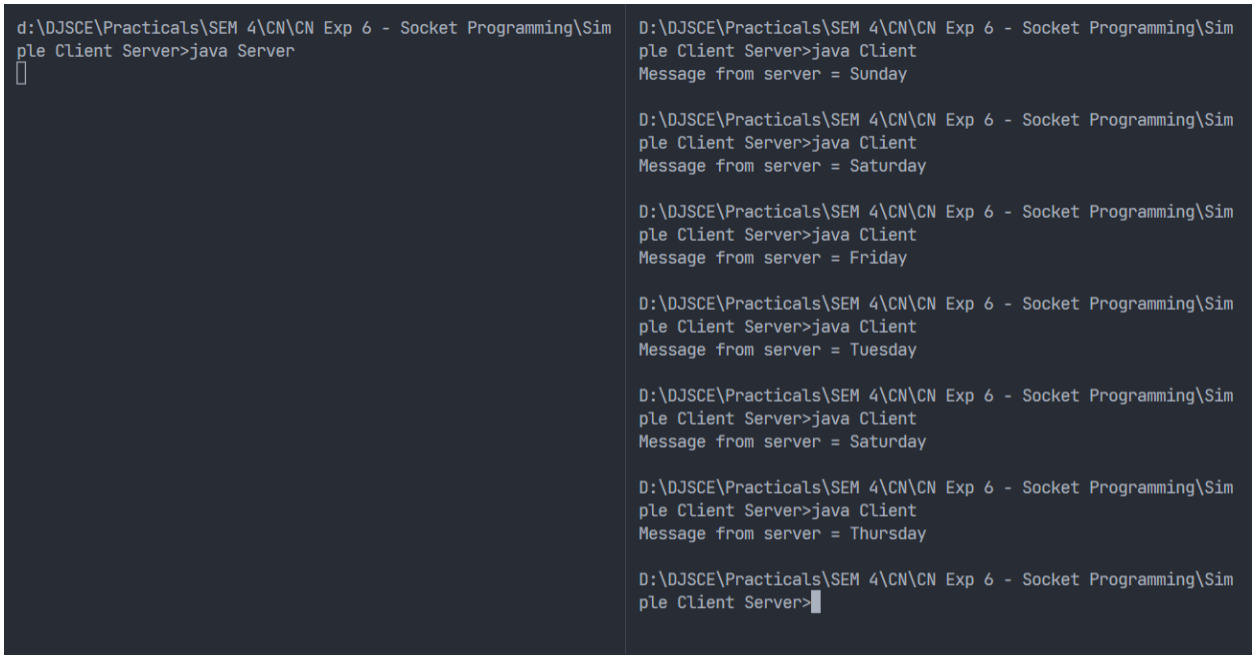
```
        Socket s1 = ss.accept();
        String week[] = { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday" };
        int i = (int) (Math.random() * week.length);
        msg = week[i];
        PrintStream ps = new PrintStream(s1.getOutputStream());
        ps.println(msg);
      }
   }
}
```

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("Message from server = " +m);
```

```
        cs.close();
    }
}
```

Output



Code(UDP)

Server

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

```
public class Server {
    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();
```

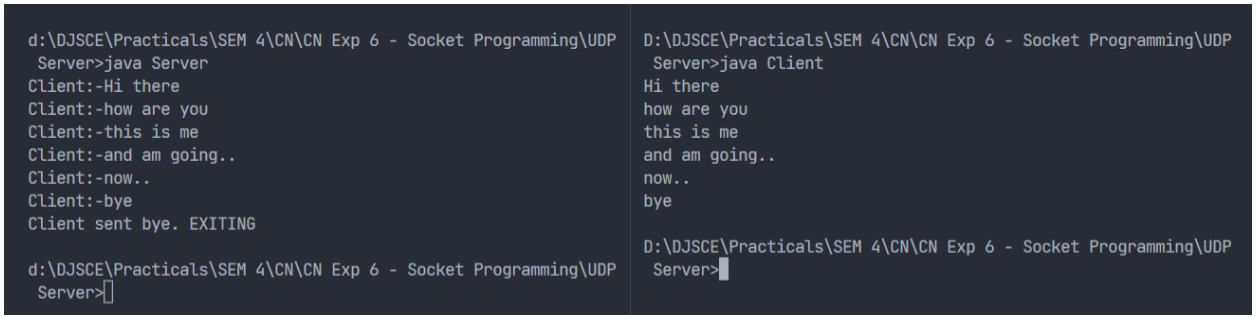
```
        int i = 0;
        while (a[i] != 0) {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}
```

Client

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

public class Client {
    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.

Computer Networks - Exp 7

Kartik Jolapara

60004200107 - 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 #==================================

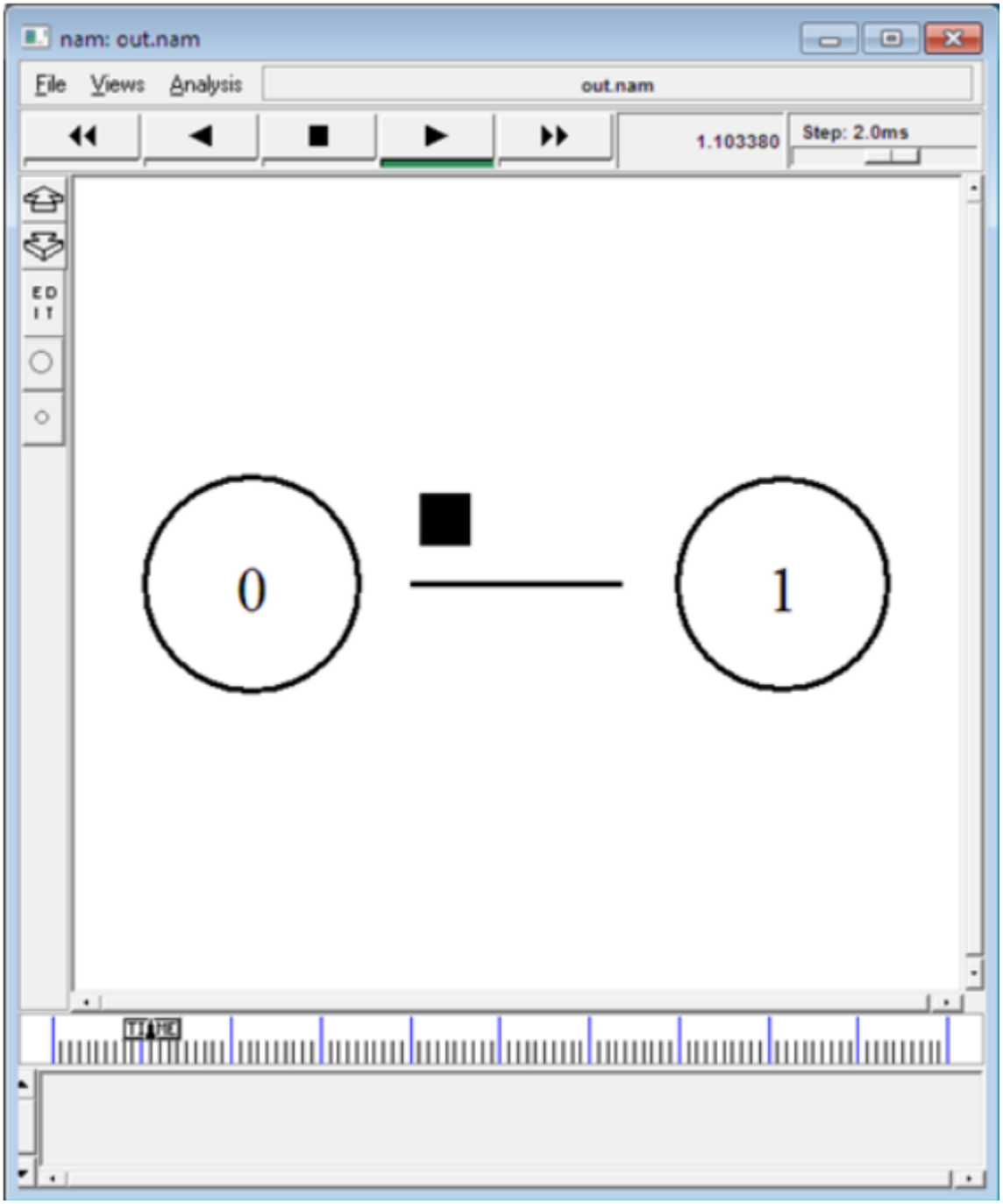
#Createlinks between nodes

$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.

Computer Networks - Exp 8

Kartik Jolapara

60004200107 - B1

Aim

To implement and understand the 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

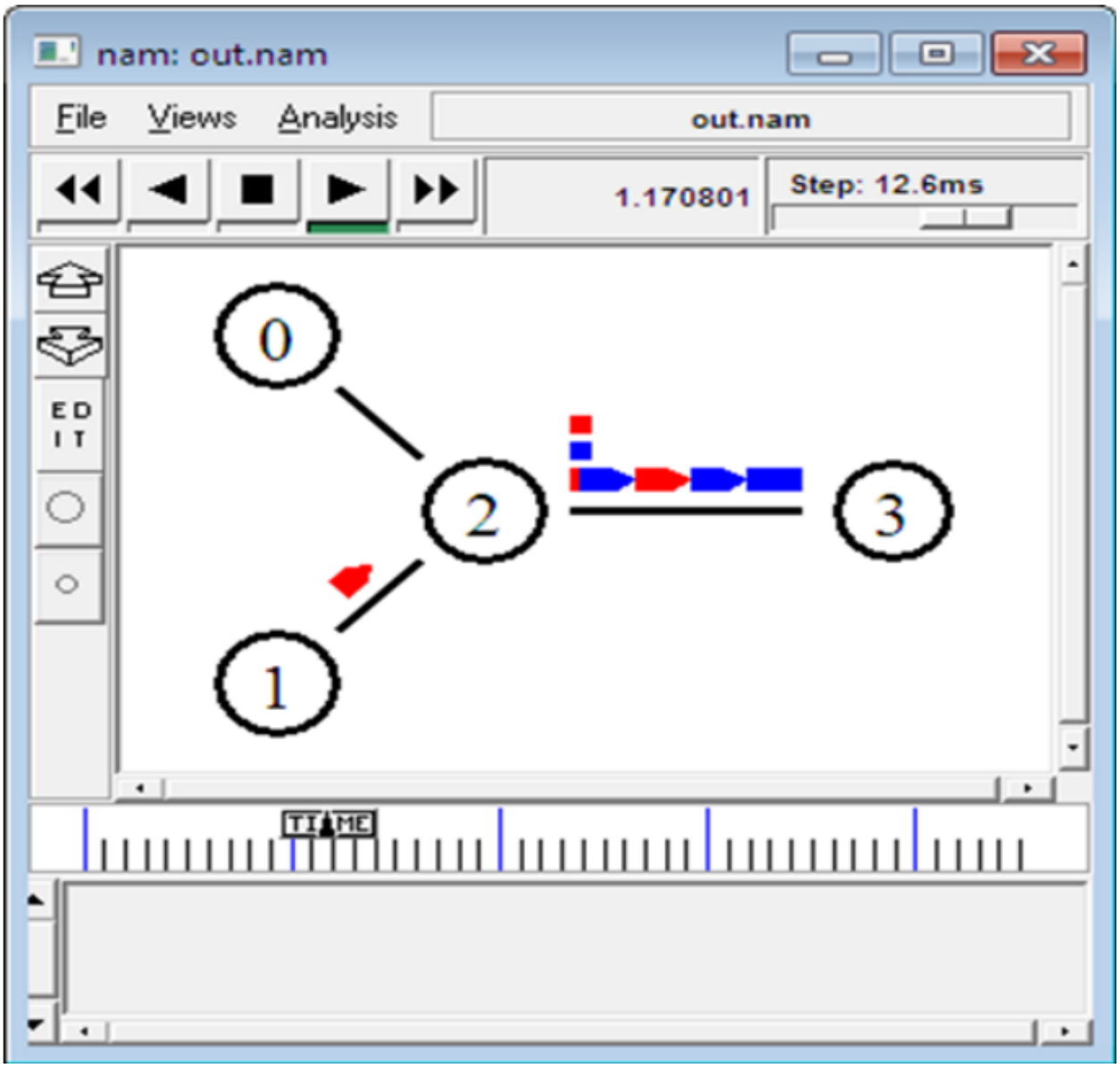
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#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 the TCP-UDP scenario in NS2.

Computer Networks - Exp 9

Kartik Jolapara

60004200107 - 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]

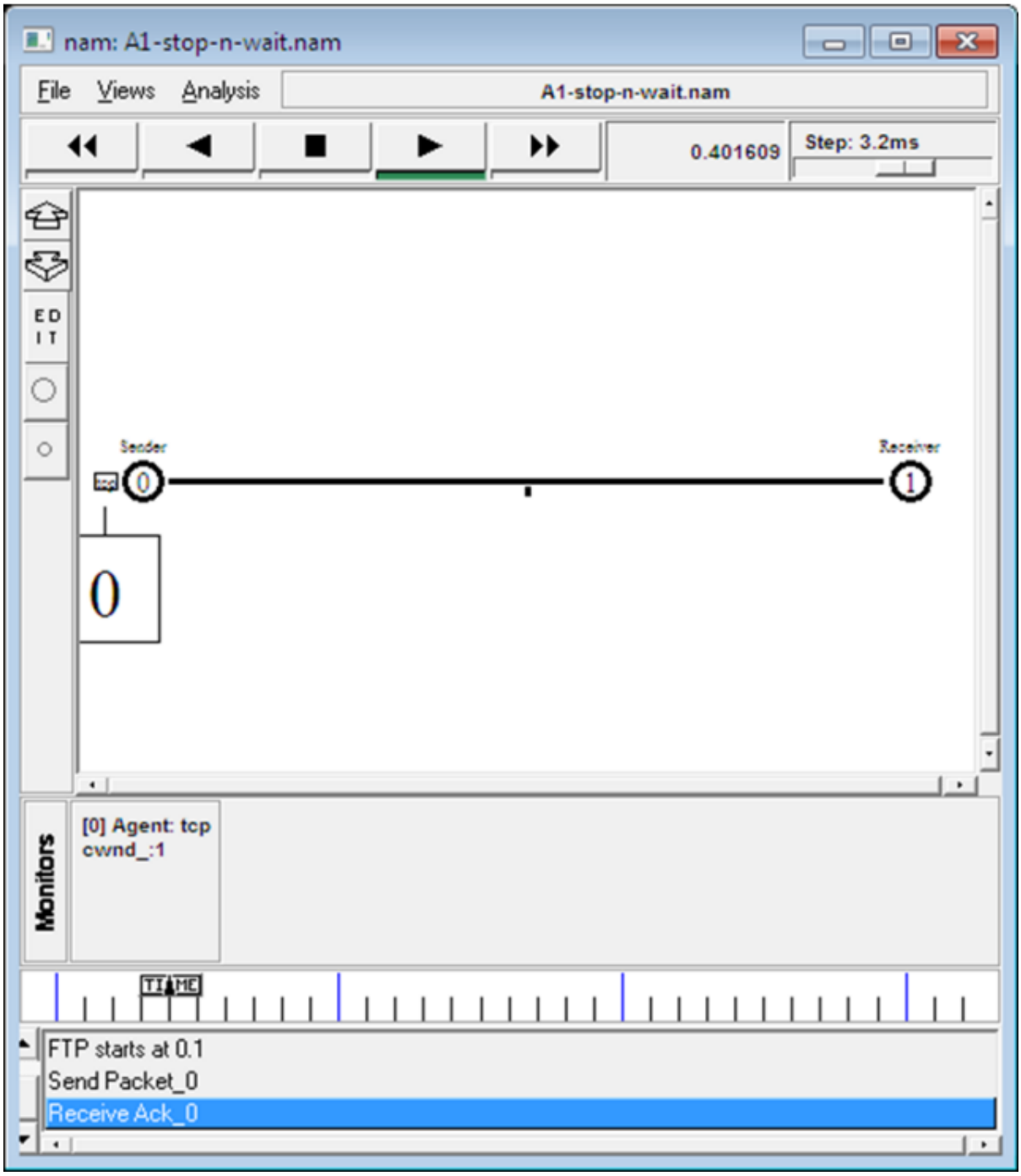
set n1 [$ns node]

```
$ns at 0.0 "$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.

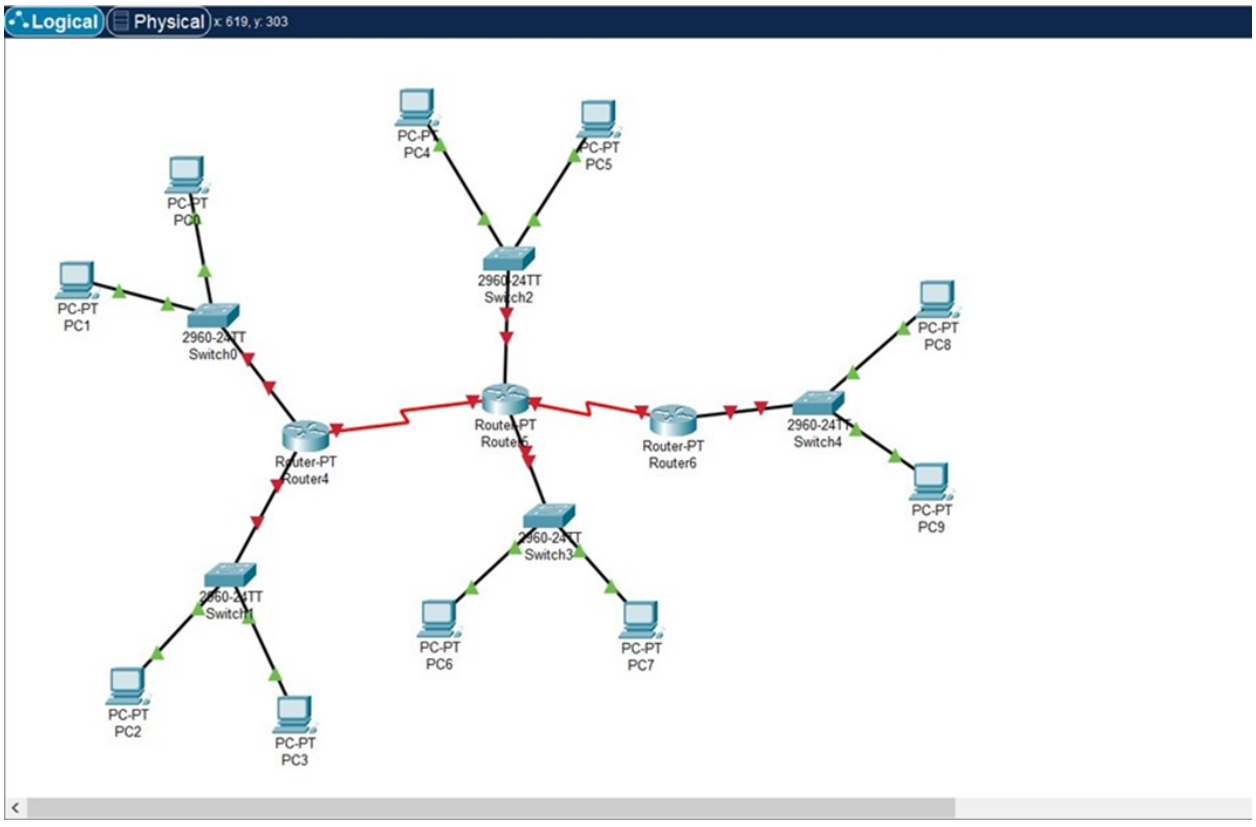
Computer Networks - Exp 10

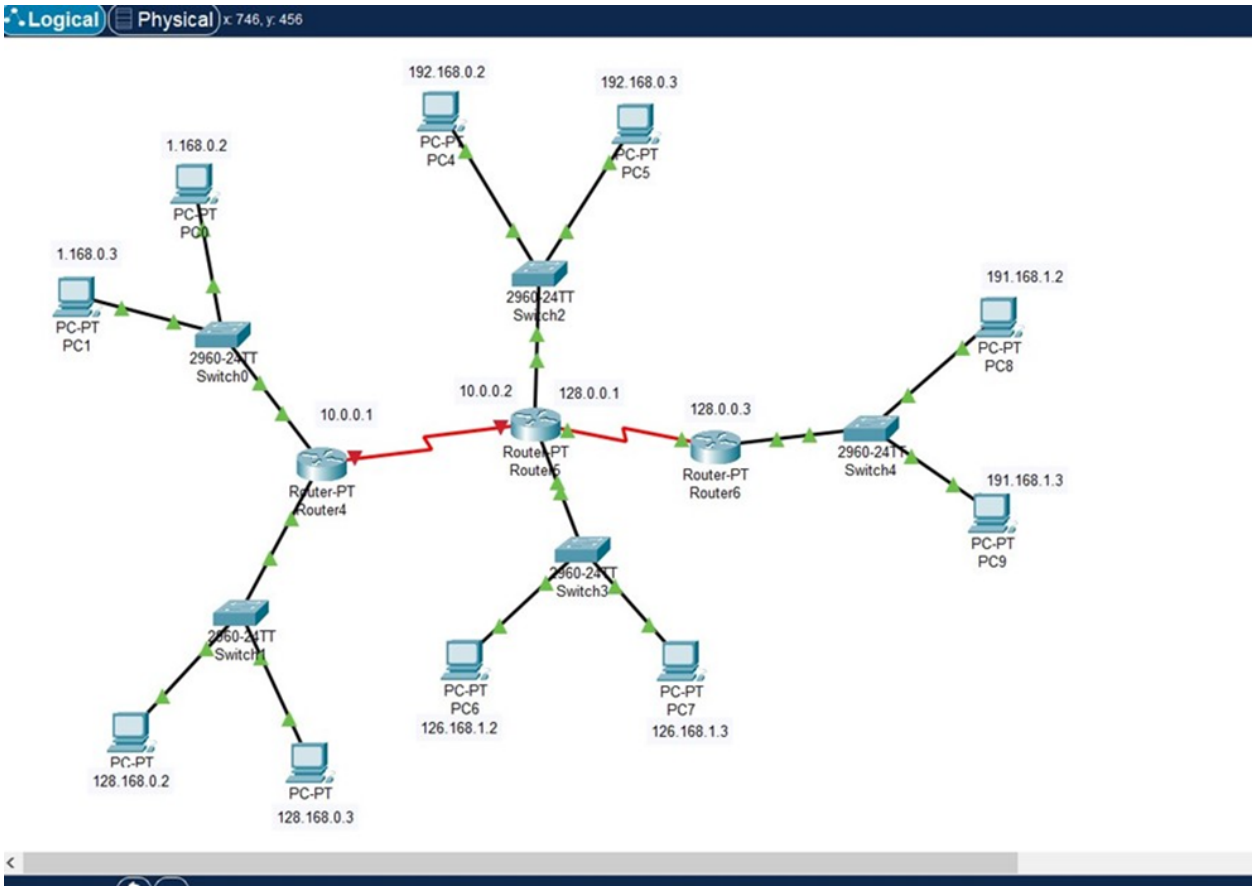
Kartik Jolapara

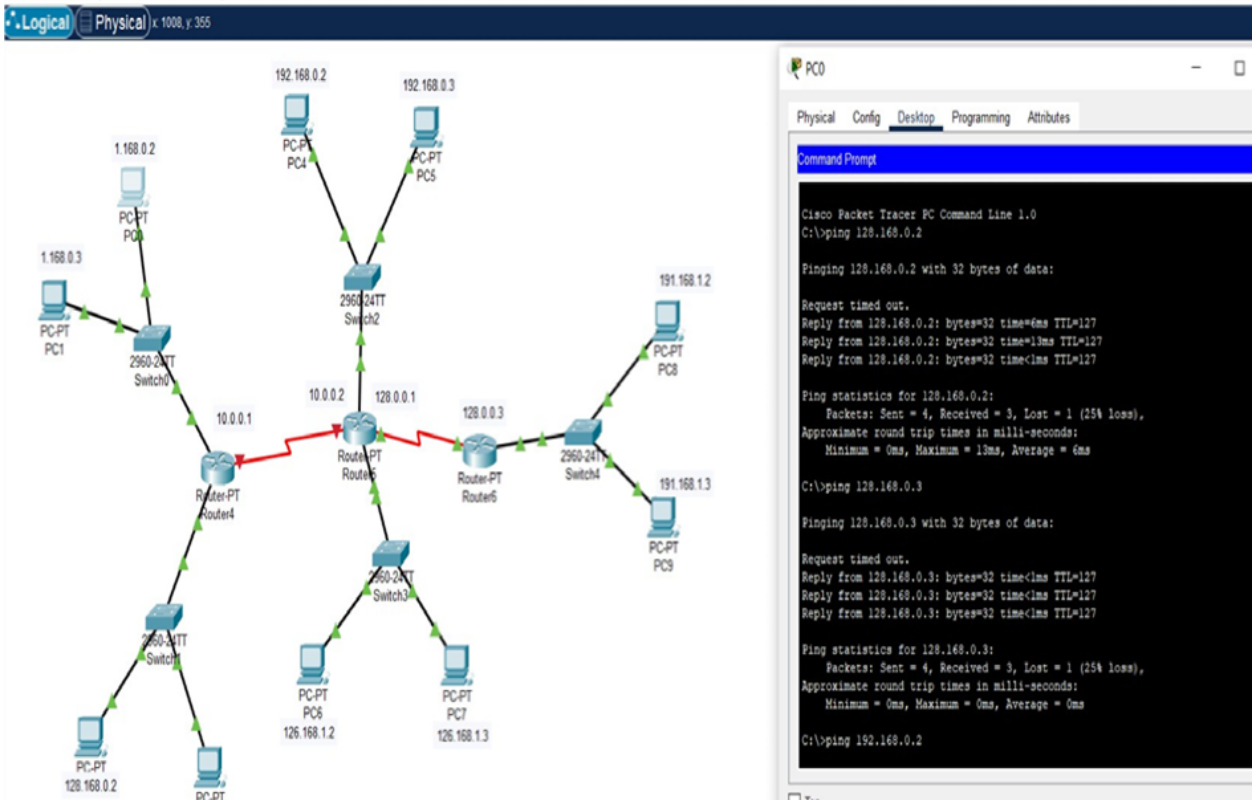
60004200107 - B1

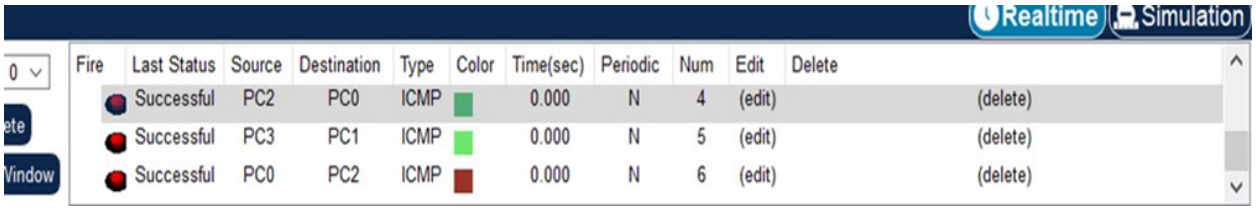
Aim

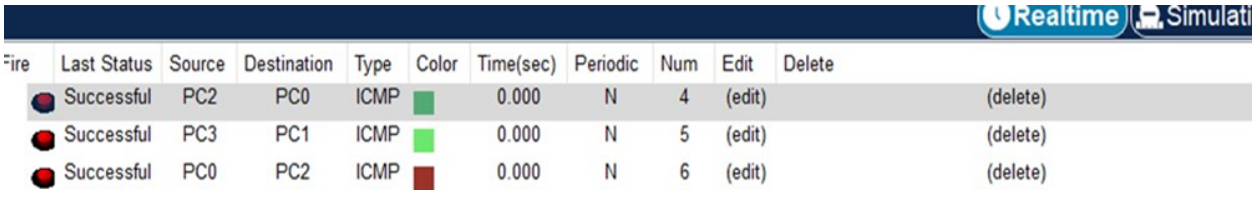
Create different networking topologies in NS2.




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
Switch0
2960-24TT
Switch2
191.168.1.2
PC-PT
PC8
10.0.0.1
10.0.0.2
128.0.0.1
128.0.0.3
Router-PT
Router4
Router-PT
Router5
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
192.168.0.2
192.168.0.3
PC-PT PC4
PC-PT PC5
1.168.0.2
PC-PT PC0
1.168.0.3
PC-PT PC1
2960-24TT Switch2
2960-24TT Switch0
10.0.0.1
10.0.0.2
128.0.0.1
128.0.0.3
Router-PT Router4
Router-PT Router5
Router-PT Router6
2960-24TT Switch4
191.168.1.2
PC-PT PC8
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
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
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
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)

Computer Networks - Exp 11

Kartik Jolapara

60004200107 - 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:

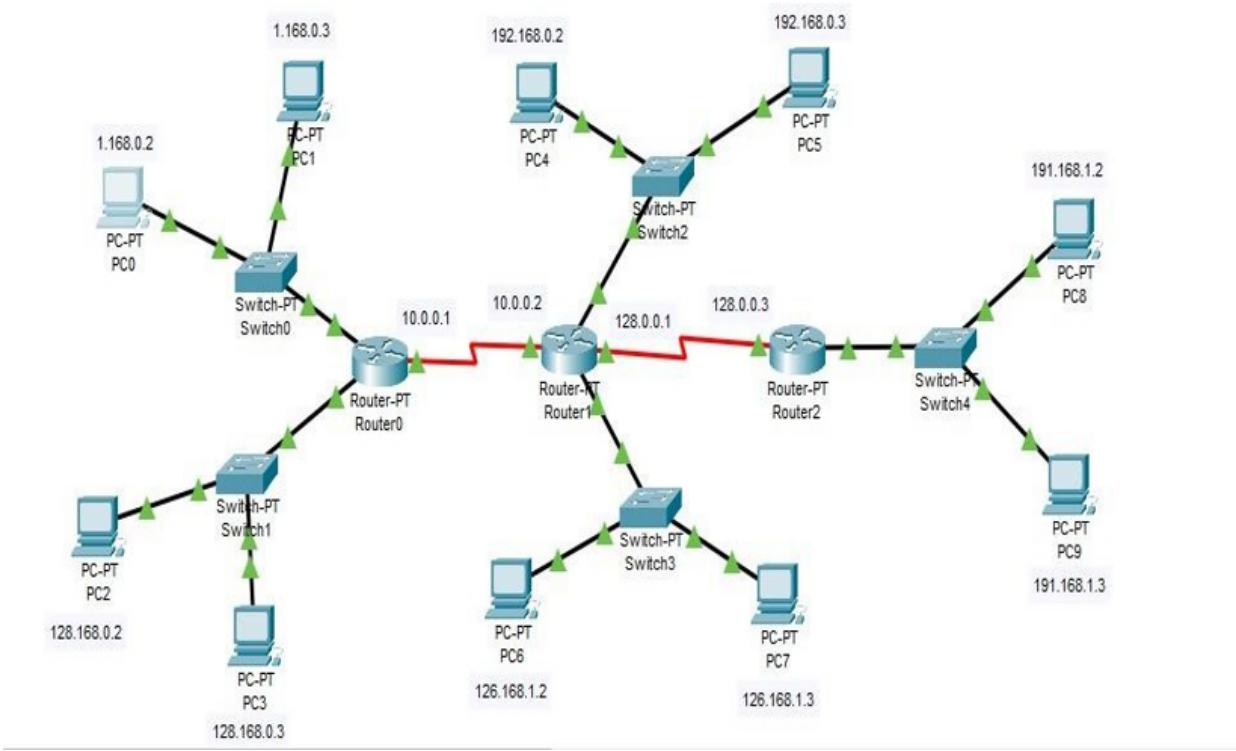
Version 1 - supports only classful routing and doesn't send subnet masks in routing updates. Uses broadcasts for updates.

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.

Diagram



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),

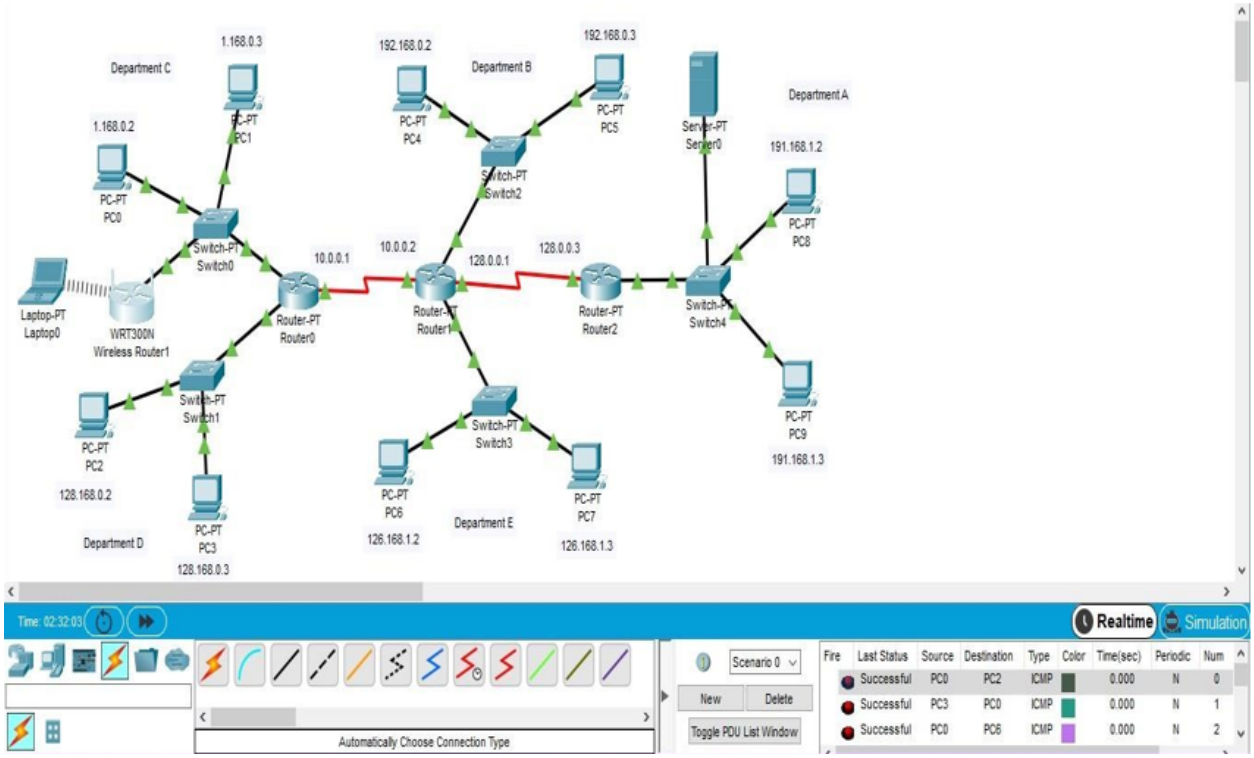
Approximate round trip times in

milli- seconds:

Minimum = 2ms, Maximum =

5ms, Average = 3ms

C:\>



Conclusion

Learned about RIP Packet Tracing and implemented using Cisco Packet Tracer.