Laboratory Manual of Computer Networks (CN)

For completion of term work of VIIIth Semester
curriculum program

Bachelor of Technology

in

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Department of Electronics and Telecommunication Engineering
Dr. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY
LONERE-402 103

2014-2015

Index

Experiment No.1

Aim : Study of networking commands and the addressing formats.

Theory :

Part 1: Addressing Formats

A networks recognize two kinds of addresses: logical (or Network layer) and physical (or MAC or hardware) addresses. MAC addresses are assigned to a device's network interface card at the factory by its manufacturer. Logical addresses can be manually or automatically assigned and must follow rules set by the protocol standards. In the TCP/IP protocol suite, IP is the core protocol responsible for logical addressing. For this reason, addresses on TCP/IP-based networks are often called IP addresses. IP addresses are assigned and used according to very specific parameters. Each IP address is a unique 32-bit number, divided into four octets, or sets of 8-bits, that are separated by periods. (Because 8 bits equals a byte, each octet is a byte and an IP address is thus composed of 4 bytes.) An example of a valid IP address is 144.92.43.178. An IP address contains two types of information: network and host. From the first octet you can determine the network class. Three types of network classes are used on modern LANs: Class A, Class B, and Class C. Table 1 summarizes characteristics of the three commonly used classes of TCP/IP-based networks.

Table 1: Commonly used TCP/IP classes

Network Class	**Beginning Octet**	**Number of Networks**	**Maximum addressable Host per Network**
A	1-126	126	16,777
B	128-191	>16,000	65,534
C	192-223	>2,000,000	254

In addition, Class D and Class E addresses do exist, but are rarely used. Class D addresses, which begin with an octet whose value is between 224 and 239, are reserved for a special type of transmission called Multicasting. IETF (Internet Engineering Task Force) reserves Class E addresses, which begin with an octet whose value is between 240 and 254, for experimental use. One should never assign Class D or Class E addresses to devices on the network. Although 8 bits have 256 possible combinations, only the numbers 1 through 254 can be used to identify networks and hosts in an IP address. The number 0 is reserved to act as a placeholder when referring to an entire group of computers on a network—for example, "10.0.0.0" represents all of the devices whose first octet is "10." The number 255 is reserved for broadcast transmissions. For example, sending a message to the address 255.255.255.255 will send a message to all devices connected to your network segment.

A portion of each IP address contains clues about the network class. An IP address whose first octet is in the range of 1-126 belongs to a Class A network. All IP addresses for device. on a Class A segment share the same first octet, or bits 0 through 7, as shown in figure for example, nodes with the following IP addresses may belong to the same

Class A network:

23.78.110.109, 23.164.32.97, 23.48.112.43, and 23.108.37.22.

In this example, “23” is the *network ID*. The second through fourth octets (bits 8 through 31) in a Class A address identify the host.

An IP whose first octet is in the range of 128-191 belongs to a Class B network. All IP addresses for devices on a Class B segment share the first two octets, or bits 0 through 15. For example, nodes with the following IP addresses may belong to the same Class B network:
168.34.88.29, 168.34.55.41, 168.34.73.49, and 168.34.205.113.

In this example, “168.34” is the network ID. The third and fourth octets (bits 16 through 31) on a Class B network identify the host, as shown in figure 7.

An IP address whose first octet is in the range of 192-223 belongs to a Class C network. All IP addresses for devices on a Class C segment share the first three octets, or bits 0 through 23. For example, nodes with the following addresses may belong to the same Class C network: 204.139.118.7, 204.139.118.54, 204.139.118.14, and 204.139.118.31. In this example, “204.139.118” is the network ID. The fourth octet (bits 24 through 31) on a Class C network identifies the host, as shown in Fig.1.

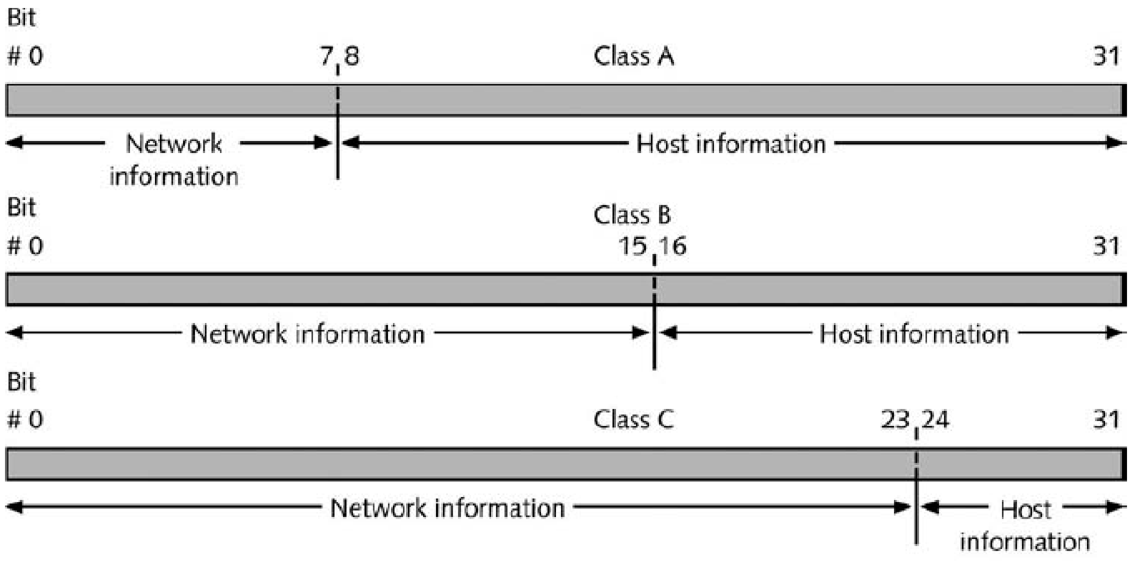
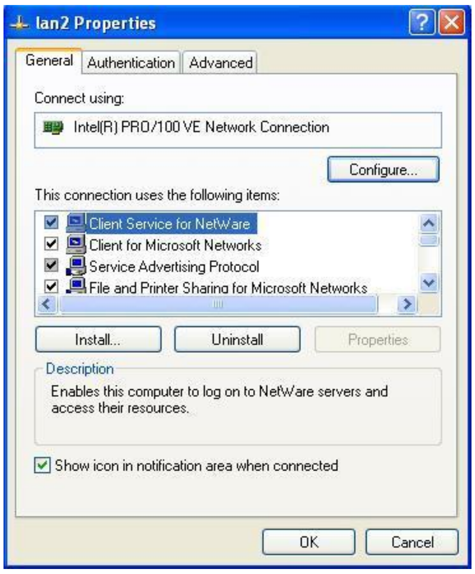


Fig.1. IP addresses and their classes

Checking & Changing IP Address of the system

- Right click ‘My Network Places’ icon from right bottom corner on desktop
- Choose ‘Ethernet’
- Choose ‘Properties’


lan2 Properties
General | Authentication | Advanced
Configure...
Install... | Uninstall | Properties
OK | Cancel

Connect using:

Intel(R) PRO/100 VE Network Connection

This connection uses the following items:

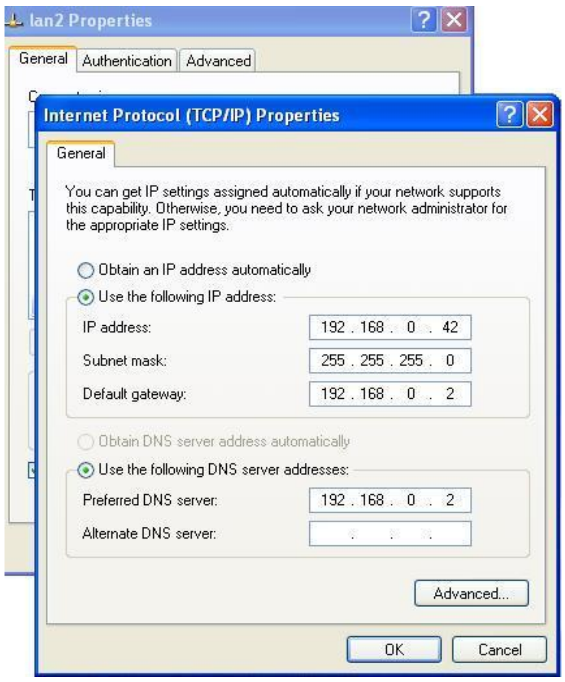
- ✓ Client Service for NetWare
- ✓ Client for Microsoft Networks
- ✓ Service Advertising Protocol
- ✓ File and Printer Sharing for Microsoft Networks

Description

Enables this computer to log on to NetWare servers and access their resources.

✓ Show icon in notification area when connected

Here choose ‘Internet Protocol (TCP/IP)’ checkbox. The ‘Properties’ button will be enabled automatically. Double Click ‘Properties’ button.


lan2 Properties
General | Authentication | Advanced
Internet Protocol (TCP/IP) Properties
General
Advanced...
OK | Cancel

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address:	192 . 168 . 0 . 42
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	192 . 168 . 0 . 2

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

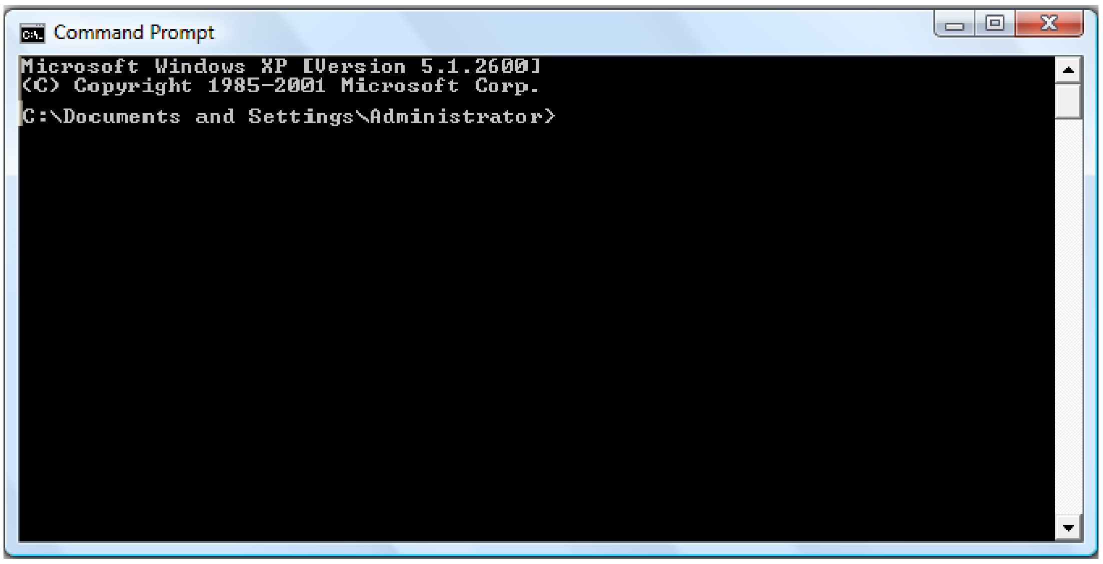
Preferred DNS server:	192 . 168 . 0 . 2
Alternate DNS server:	. . .

Here one can set the IP Address & Default gateway.

Part 2: Networking Commands

Operating System:

Operating system is a set of software that controls and manages hardware and basic system operations for a computer. The operating system loads programs into the computer's memory, runs these programs, and manages peripherals like disks and printers. In the PC operating systems MS-DOS and PC DOS, a number of standard system commands were provided for common tasks such as listing files on a disk or moving files. Some commands were built into the command interpreter; others existed as external commands on disk. Over the several generations of DOS, commands were added for the additional functions of the operating system. In the current Microsoft Windows operating system, a text-mode command prompt window can still be used. Following are the some DOS commands.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>
```

Fig 2. Command Prompt

1. **systeminfo**

Displays complete system information for Microsoft Windows XP Professional computers. This command is only available to Microsoft Windows XP Professional computers and is not available in Microsoft Windows XP Home. The above command would display information about the computer and the operating system, including networking information, and installed hot fixes.

2. dir

The dir command allows you to see the available files in the current and/or parent directories. **Dir** is often used as an abbreviation for directory. The dir command is used to display a list of files and folders contained inside the folder that you are currently working in. The dir command also displays other important information like the hard drive's serial number, the total number of files listed, their combined size, the total amount of free space left on the drive, and more

3. ipconfig

'Ipconfig' is a DOS utility that can be used from MS-DOS and an MS-DOS shell to display the network settings currently assigned and given by a network. This command can be utilized to verify a network connection as well as to verify your network settings. The default is to display only the IP address, subnet mask and default gateway for each adapter bound to TCP/IP.

```
Connection-specific DNS Suffix  : hsd1.ut.comcast.net.
IP Address. . . . . . . . . . . . : 192.168.201.245
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 192.168.201.1
```

4. ipconfig /all

ipconfig /all is a DOS utility that can be used from MS-DOS. It provides a total information about the system. It also provides NIC cards details i.e. name of card, physical address of card, model of the card. It also provides window IP configuration i.e. host name of the machine, node type DHCP details.

5. ping

Short for **Packet InterNet Groper**, **ping** is a utility used to verify whether or not a network data packet is capable of being distributed to an address without errors. The ping utility is commonly used to check for network errors. In computer gaming, a **ping**, also known as a **high ping** or **low ping**, is a measurement of how fast the connection to the game server the player has. For example, a player with a low 50 ms ping is going to have a better game experience than a player with a 250 ms ping.

Conclusion:

Experiment No: 2

Aim : Study of different Networking Devices and configuration.

Theory:

A network, is a system of interconnected computers (and devices) that operate interactively. Any number of computers may be connected into a network, from two to dozens, hundreds, thousands or even millions. Networks typically include other devices such as printers, external hard drives, modems and routers, etc. Nearly everyone involved with computers at any level now requires knowledge and skills with networks computer systems. Today networks are widely used in not only the workplace; but increasingly in homes.

HUB :

Hubs/repeaters are used to connect together two or more Ethernet segments of any media type. In larger designs, signal quality begins to deteriorate as segments exceed their maximum length. Hubs provide the signal amplification required to allow a segment to be extended a greater distance. A hub takes any incoming signal and repeats it out all ports. Ethernet hubs are necessary in star topologies such as 10BASE-T. A multi-port twisted pair hub allows several point-to-point segments to be joined into one network. One end of the point-to-point link is attached to the hub and the other is attached to the computer. If the hub is attached to a backbone, then all computers at the end of the twisted pair segments can communicate with all the hosts on the backbone. The number and type of hubs in any one-collision domain is limited by the Ethernet rules.

The hub is considered the least common denominator in device concentrators. Hubs offer an inexpensive option for transporting data between devices, but hubs don't offer any form of intelligence. Hubs can be active or passive. An **active hub** strengthens and regenerates the incoming signals before sending the data on to its destination. **Passive hubs** do nothing with the signal.

As shown in Fig.1 the hub is a hardware device that contains multiple, independent ports that match the cable type of the network. Most common hubs interconnect Category 3 or 5 twisted-pair cable with RJ-45 ends, although Coax BNC and Fiber Optic BNC hubs also exist.

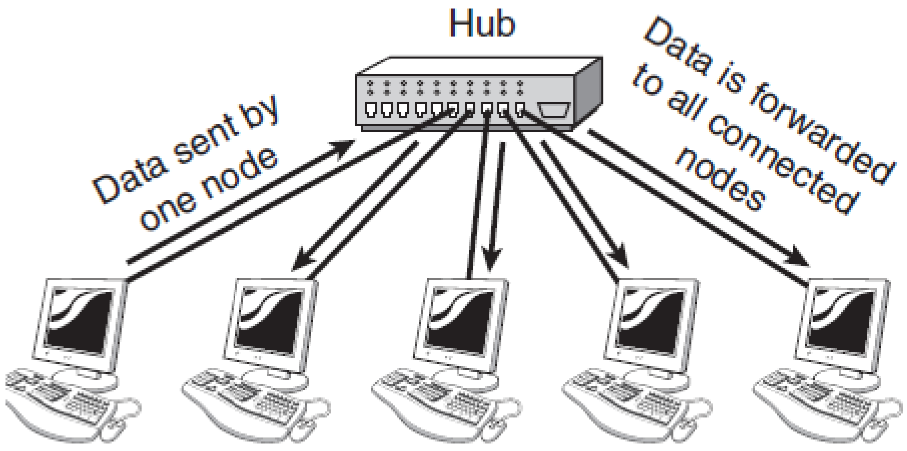


Fig.1. A HUB

SWITCHES

Like hubs, switches are the connectivity points of an Ethernet network. Devices connect to switches via twisted-pair cabling, one cable for each device. The difference between hubs and switches is in how the devices deal with the data that they receive. Whereas a hub forwards the data it receives to all of the ports on the device, a switch forwards it only to the port that connects to the destination device. It does this by learning the MAC address of the devices attached to it, and then by matching the destination MAC address in the data it receives. Figure 3.1 shows how a switch works. Data sent by one node Data is forwarded only to the destination node.

Working of Switch:

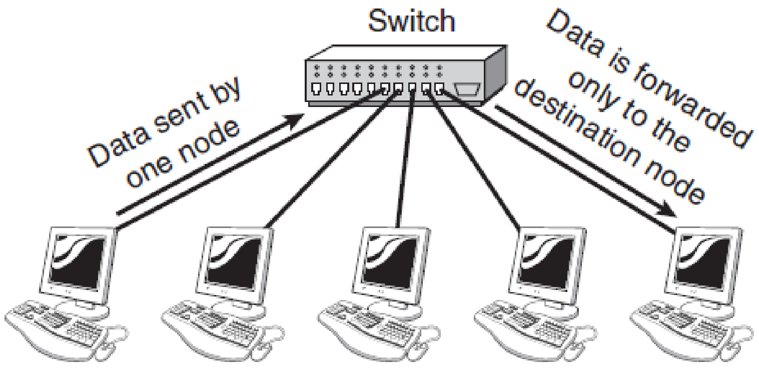


Fig. 2. A Switch

By forwarding data only to the connection that should receive it, the switch can improve network performance in two ways. First, by creating a direct path between two devices and controlling their communication, it can greatly reduce the number of collisions on the network. It may be recalled, collisions occur on Ethernet networks when two devices attempt to transmit at exactly the same time.

BRIDGES:

Bridges are networking devices that connect networks. Sometimes it is necessary to divide networks into subnets to reduce the amount of traffic on each larger subnet or for security reasons. Once divided, the bridge connects the two subnets and manages the traffic flow between them. Today, network switches have largely replaced bridges. A bridge functions by blocking or forwarding data, based on the destination MAC address written into each frame of data. If the bridge believes the destination address is on a network other than that from which the data was received, it can forward the data to the other networks to which it is connected. If the address is not on the other side of the bridge, the data is blocked from passing. Bridges "learn" the MAC addresses of devices on connected networks by "listening" to network traffic and recording the network from which the traffic originates. Fig. 3 shows a representation of a bridge.

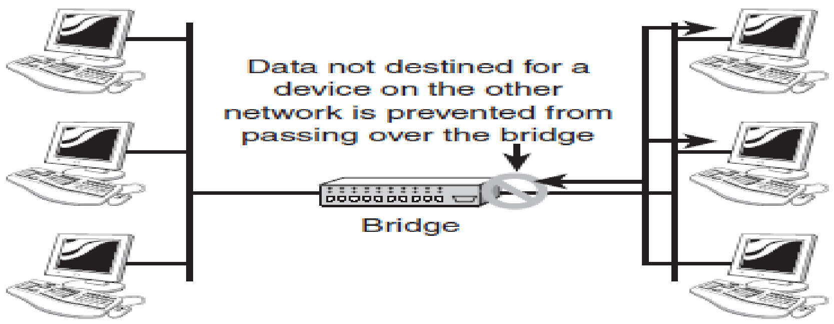

Data not destined for a device on the other network is prevented from passing over the bridge
Bridge

Fig.3. Bridge

ROUTERS:

Routers are an increasingly common sight in any network environment, from a small home office that uses one to connect to an Internet service provider (ISP) to a corporate IT environment where racks of routers manage data communication with disparate remote sites.

Routers make internetworking possible, and in view of this, they warrant detailed attention. Routers are network devices that literally route data around the network.

By examining data as it arrives, the router can determine the destination address for the data; then, by using tables of defined routes, the router determines the best way for the data to continue its journey. Unlike bridges and switches, which use the hardware-configured MAC address to determine the destination of the data, routers use the software-configured network address to make decisions. This approach makes routers more functional than bridges or switches, and it also makes them more complex because they have to work harder to determine the information.

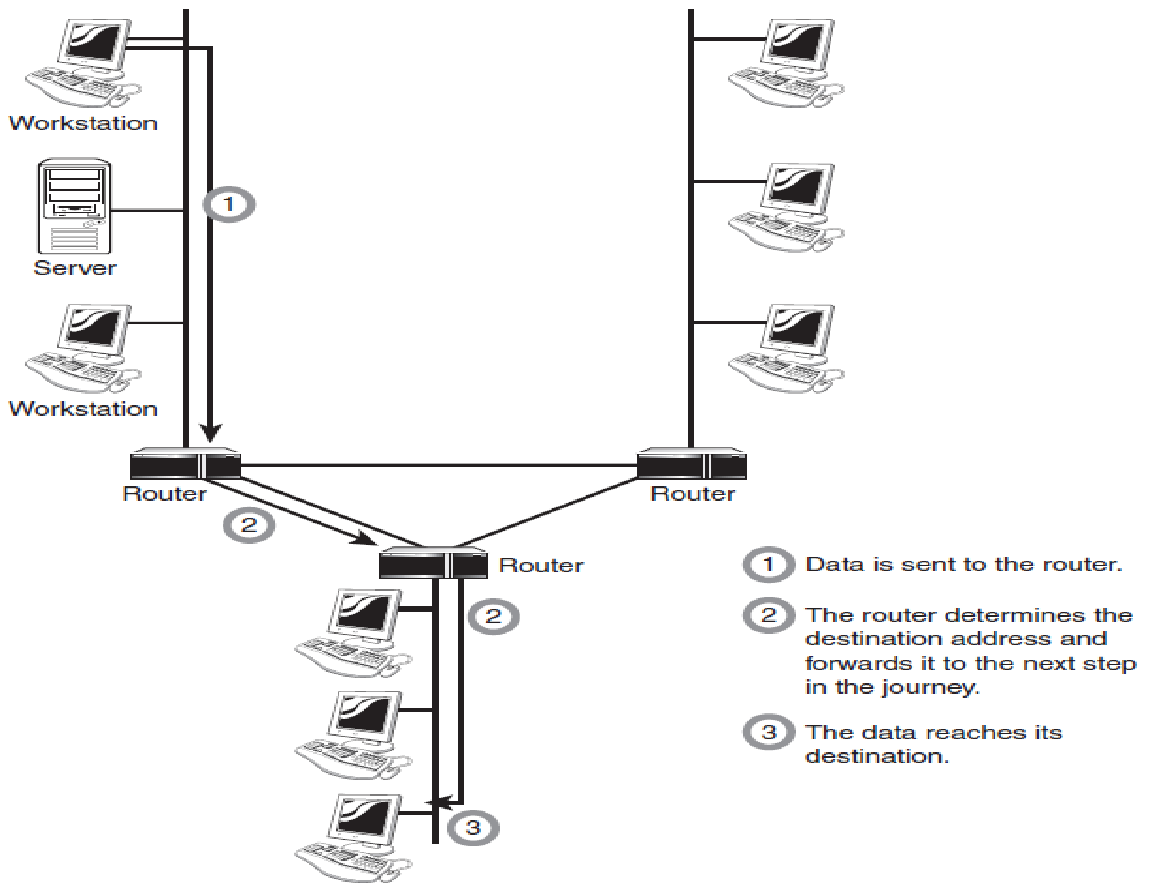

Workstation
Server
Workstation
1
Router
Router
2
Router
2
3
1 Data is sent to the router.
2 The router determines the destination address and forwards it to the next step in the journey.
3 The data reaches its destination.

Fig.4 Routers

GATEWAYS:

The term gateway is applied to any device, system, or software application that can perform the function of translating data from one format to another. The key feature of a gateway is that it converts the format of the data, not the data itself. You can use gateway functionality in many ways. For example, a router that can route data from an IPX network to an IP network is, technically, a gateway. The same can be said of a translational bridge that, as described earlier in this chapter, converts from an Ethernet network to a Token Ring network and back again.

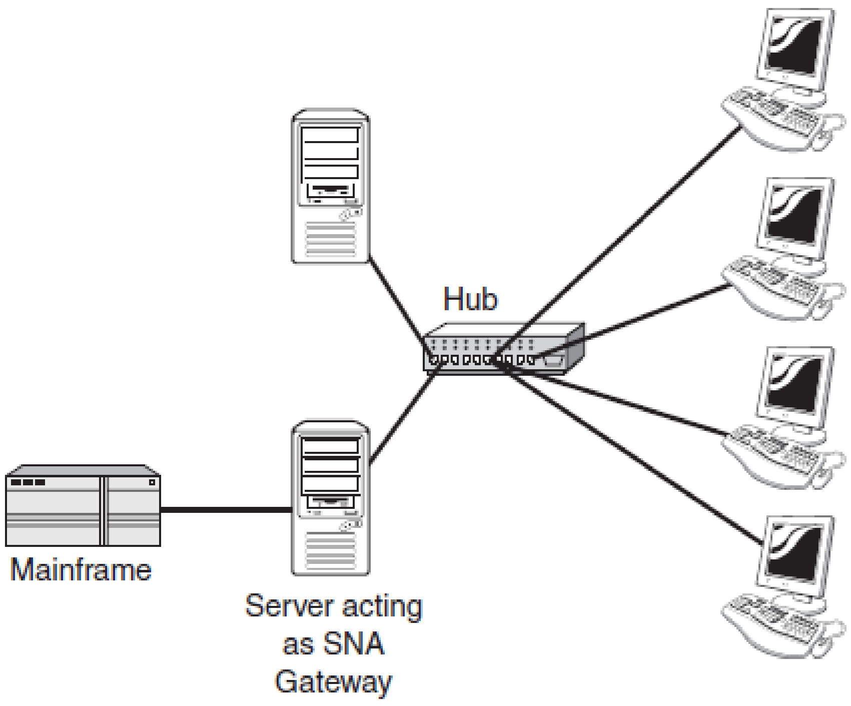


Fig.5 Gateways

Conclusion:

.

Experiment No : 3

Aim : Study & Implementation of cable designs in Networking.

Theory:

All networks need cables. The three main types are twisted-pair cable (TP), coaxial cable, and fiber-optic cable (FDDI—Fiber Distributed Data Interface).

Twisted-Pair Cable:

Twisted-pair cable, shown in Fig. 1, consists of two insulated strands of copper wire twisted around each other to form a pair. One or more twisted pairs are used in a twisted-pair cable. The purpose of twisting the wires is to eliminate electrical interference from other wires and outside sources such as motors. By twisting the wires, any electrical noise from the adjacent pair will be canceled. The more twists per linear foot, the greater the effect.

Twisted-pair wiring comes in two types: Shielded (STP) and Unshielded (UTP). STP has a foil or wire braid wrapped around the individual wires of the pairs; UTP does not. The STP cable uses a woven-copper braided jacket, which is a higher-quality, more protective jacket than UTP.

Fig .1:Twisted pair cable

Of the two types, UTP is the most common. UTP cables which can be further divided into five categories:

- **Category 1:** Traditional telephone cable. Carries voice but not the data.
- **Category 2:** Certified UTP for data transmission of up to 4 Mbps (Megabits per second). It has four twisted pairs.
- **Category 3:** Certified UTP for data transmission of up to 10 Mbps. It has four twisted pairs.
- **Category 4:** Certified UTP for data transmissions up to 16 Mbps. It has four twisted pairs.
- **Category 5:** Certified for data transmissions up to 100 Mbps. It has four twisted pairs of copper wire.

The following colors are standard for CAT5.

1. Blue
2. White /Blue
3. Orange

4. White/Orange
5. Green
6. White/Green
7. Brown
8. White/Brown

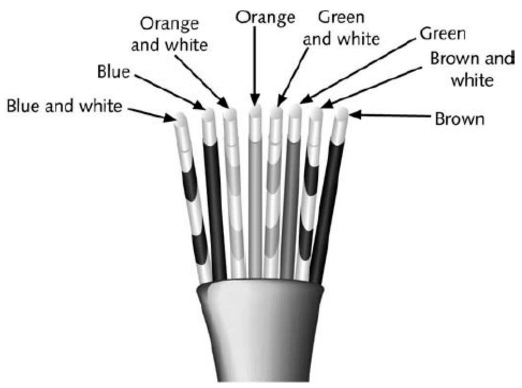


Fig. 2: A CAT 5 UTP cable with pairs untwisted

RJ-45 -- 10Base-T

RJ-45's are also used for network connections. The most common 10Base-T scheme is known as TIA/EIA T568B, which is:

Pin	Function	Pin	Function
1	Receive	5	unused
2	Receive	6	Transmit
3	Transmit	7	unused
4	unused	8	unused

Fig. 3: RJ-45 connectors

10Base-T straight through cable:

The "straight through" cable is used to connect a network device (LAN card, etc.) to a hub or switch. This cable has TIA/EIA T568B connectors on each end (RJ-45). Each connector is wired like this:

Pin	**Wire**
1	Orange
2	White / orange
3	Blue
4	White / green
5	Green
6	White / blue
7	Brown
8	White / brown

10 Base-T crossover cable:

This cable is used to connect two identical devices. You can connect two LAN cards together to create a miniature network. Or you can connect two hubs together to expand your network. Technically, one connector is TIA/EIA T568B, and the other is TIA/EIA T568A.

Connector 1		Connector 2	
Pin	**Wire**	**Pin**	**Wire**
1.	White / orange	1	White / green
2.	Orange	2	Green
3.	White / green	3	White / orange
4.	Blue	4	Blue
5.	White / blue	5	White / blue
6.	Green	6	Orange
7.	White / brown	7	White / brown
8.	Brown	8	Brown

Twisted-pair cable has several advantages over other types of cable (coaxial and fiber-optic)—it is readily available, easy to install, and inexpensive. Among its disadvantages are its sensitivity to EMI (Electromagnetic Interference) and susceptibility to eavesdropping; it does not support communication at distances of greater than 100 feet; and it requires the addition of a hub (a multiple network connection point) if it is to be used with more than two computers.

Coaxial Cable:

Coaxial cable (see Fig.4) is made of two conductors that share the same axis; the center is a copper wire that is insulated by a plastic coating and then wrapped with an outer conductor (usually a wire braid). This outer conductor around the insulation serves as electrical shielding for the signal being carried by the inner conductor. Outside the outer conductor is a tough insulating plastic tube that provides physical and electrical protection. At one time, coaxial cable was the most widely used network cabling. However, with improvements and the lower cost of twisted-pair cables, it has lost its popularity.

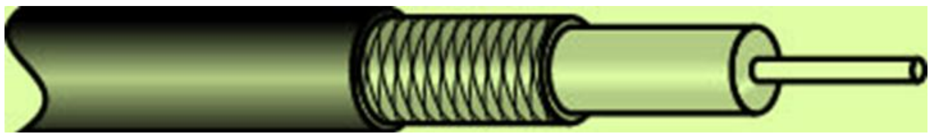


Fig 4: Coaxial cable

Coaxial cable is found in two types: thin (*ThinNet*) and thick (*ThickNet*). Of the two, ThinNet is the easiest to use. It is about one-quarter of an inch in diameter, making it flexible and easy to work with (it is similar to the material commonly used for cable TV). ThinNet can carry a signal about 605 feet (185 meters) before the signal strength begins to suffer. ThickNet, onthe other hand, is about three-eighths of an inch in diameter. This makes it a better conductor—it can carry a signal about 1,640 feet (500 meters) before signal strength begins to suffer. The disadvantage of ThickNet over ThinNet is that it is more difficult to work with. The ThickNet version is also known as standard Ethernet cable.

When compared to twisted-pair, coaxial cable is the better choice even though it costs more. It is a standard technology that resists rough treatment and EMI. Although more resistant, it is still susceptible to EMI and eavesdropping.

Use coaxial cable if one needs:

- A medium that can transmit voice, video, and data.
- To transmit data longer distances than less expensive cabling.
- A familiar technology that offers reasonable data security.

A Mixed-Cable System:

Many networks use both Twisted-Pair and Coaxial Cable. Twisted-pair cable is used on a per-floor basis to run wires to individual workstations. Coaxial cable is used to wire multiple floors together. Coaxial cable should also be considered for a small network because you can purchase prefabricated cables (with end connectors installed) in various lengths.

Fiber-Optic Cable:

Fiber-optic cable (see Fig.5) is made of light-conducting glass or plastic fibers. It can carry data signals in the form of modulated pulses of light. The plastic-core cables are easier to install, but do not carry signals as far as glass-core cables. Multiple fiber cores can be bundled in the center of the protective tubing.

Fig.5: Fiber-optic cable

When both material and installation costs are taken into account, fiber-optic cable can prove to be no more expensive than twisted-pair or coaxial cable. Fiber has some advantages over copper wire; it is immune to EMI and detection outside the cable and provides a reliable and secure transmission media. It also supports very high bandwidths (the amount of information the cable can carry), so it can handle thousands of times more data than twisted-pair or coaxial cable.

Cable lengths can run from .25 to 2.0 kilometers depending on the fiber-optic cable and network. If you need to network multiple buildings, this should be the cable of choice. Fiber- optic cable systems require the use of fiber-compatible NIC.

Specifying the Right Cable:

In order to ensure trouble-free operation, network cabling must match the system requirements. Cable specifications are based on three factors: speed, bandwidth, and length. Cables are designated with names like 10Base5. Speed is the first number in the identification¾representing the maximum transmission speed (bandwidth) in Mbps. This will be 1, 5, 10, or 100. Band is the second part of the identification. It is either base or broad depending upon whether the cable is baseband or broadband.
The last part of the identification refers to the cable length or cable type. If the unit is a number, it is the maximum length of the cable segments in hundreds of meters (1 meter is approximately 3.3 feet). In some cases, it can refer to 50-meter increments (1Base5 is five 50-meter increments-250 meters). In other cases, it represents cable type: T (twisted-pair) or F (fiber-optic). Table 1 provides the common types of cables and their specifications.

Table 1: The common types of cables and their specifications

Name	Description	Type	Segment	Speed
10BaseT	Common	UTP twisted-pair	5 to 100 meters	10 Mbps
10Base2	Ethernet Thin Net	Coaxial	185 meters	10 Mbps
10Base5	Thick Ethernet	Coaxial	500 meters	10 Mbps
100BaseT	Becoming common	Twisted-pair	.5 to 100 meters	100 Mbps

Conclusions:

Experiment No:4

Aim: Implementation of PC to PC with IEEE 802.3.

Theory:

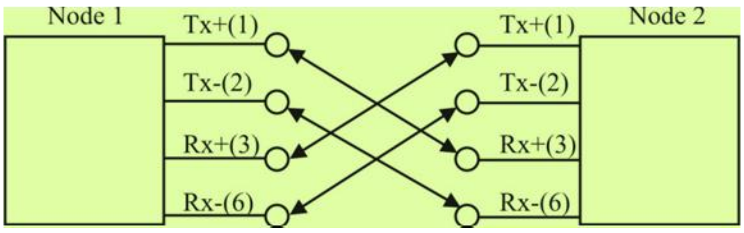
Equipment Needed:

- **ST5002A** kit
- **ST5002A** Software
- 2-CAT5 Cables
- 42 mm Patch Cords

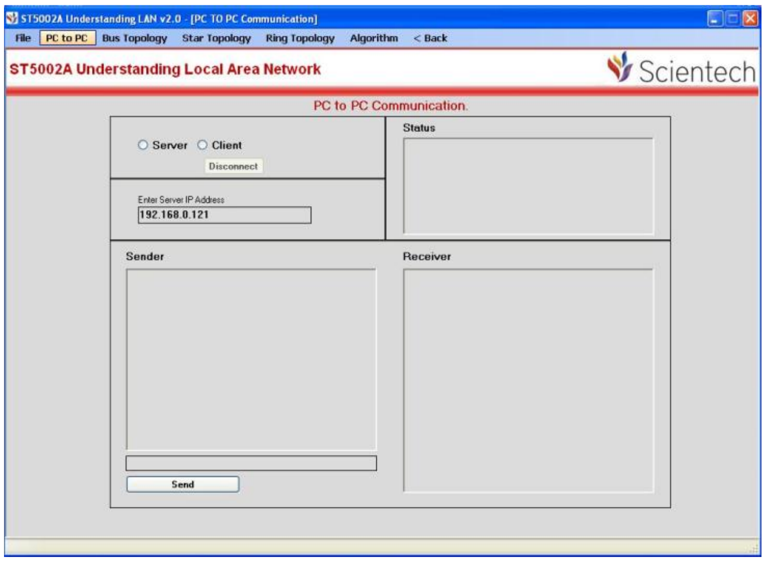
Procedure:

1. Connect CAT5 cables between computer (RJ-45 is on back panel of the computer) and RJ-45 connectors on ST5002A.

PC to PC Communication

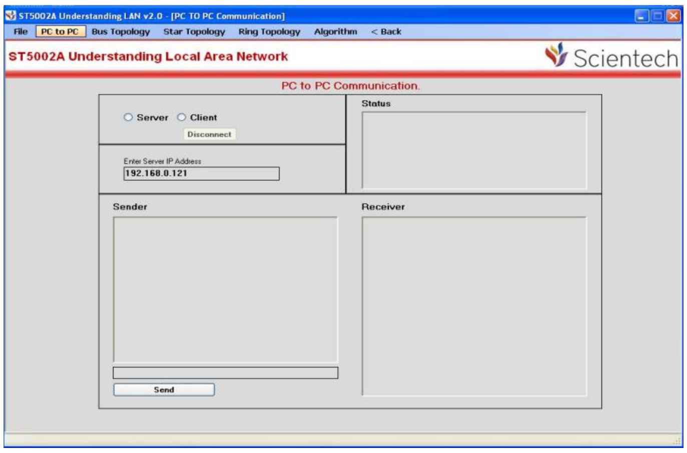


2. Connect patch cords between Tx & Rx (Tx+ to Rx+; Tx- to Rx-) for both ends.
3. Run **ST5002A** installed software on both nodes and follow the steps
4. Click on ‘PC to PC Communication’



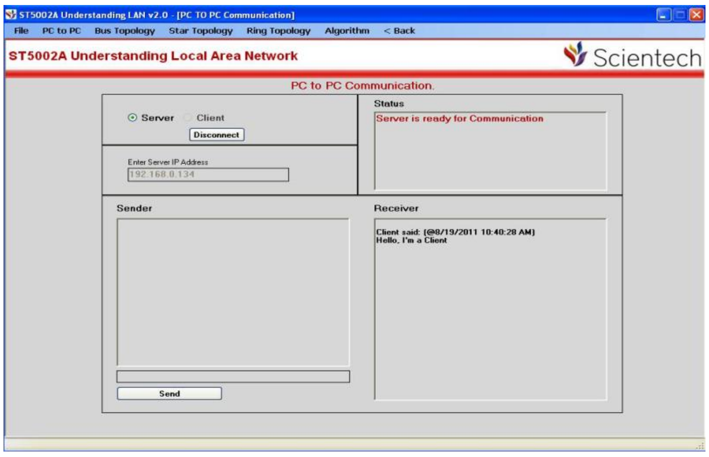
Enter Server IP Address: 192.168.0.121

5. A window will appear on screen as shown in figure.



Enter Server IP Address: 192.168.0.121

6. Select 'Server' on PC1



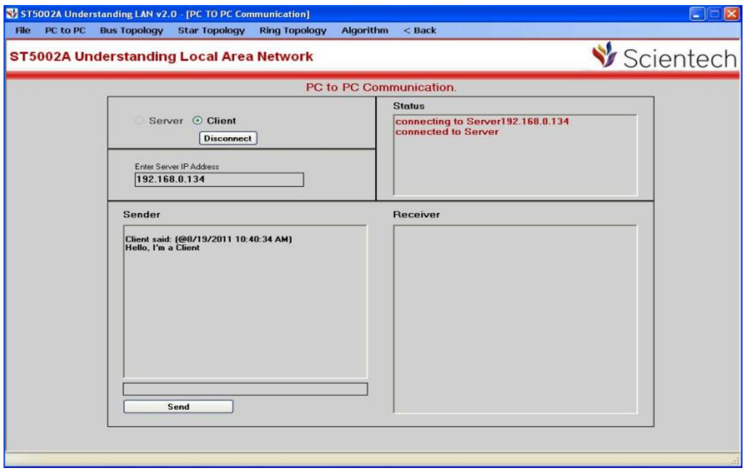
[✓ Server]

Enter Server IP Address: 192.168.0.134

Status: Server is ready for Communication

```
Client said: (@8/19/2011 10:40:28 AM)
Hello, I'm a Client
```

7. Enter the IP Address of PC1 into Client Textbox of PC2

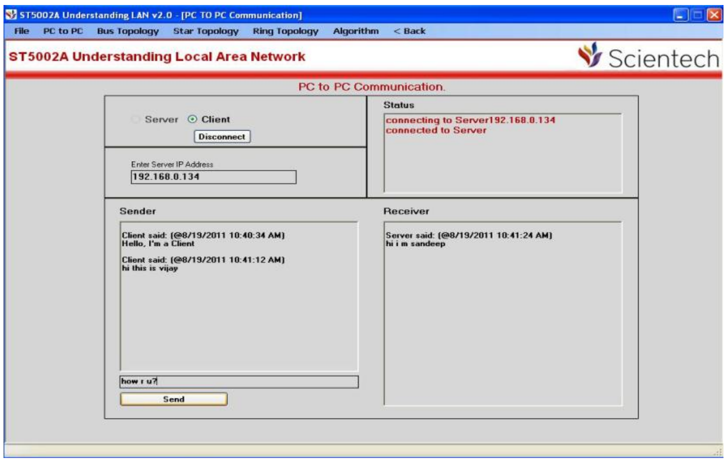


Enter Server IP Address: 192.168.0.134

```
connecting to Server192.168.0.134
connected to Server
```

```
Client said: (@8/19/2011 10:40:34 AM)
Hello, I'm a Client
```

8. Now you can have a chat application between server and client.
9. Enter data here and press 'Enter ' or click on 'Send Data'



Enter Server IP Address: 192.168.0.134

```
connecting to Server192.168.0.134
connected to Server
```

```
Client said: (@8/19/2011 10:40:34 AM)
Hello, I'm a Client

Client said: (@8/19/2011 10:41:12 AM)
hi this is vijay
```

```
Server said: (@8/19/2011 10:41:24 AM)
hi i m sandeep
```

how r u?

Note: If you want to interchange server and client, then please restart application.

Conclusion:

Experiment No:5

Aim: Implementation of Peer to Peer Network.

Theory:

The simplest form of a network is a *peer-to-peer network*. In a peer-to-peer network, every computer can communicate directly with every other computer. By default, no computer on a peer-to-peer network has more authority than another. However, each computer can be configured to share only some of its resources and keep other resources inaccessible to the network. Traditional peer-to-peer networks typically consist of two or more general-purpose personal computers, with modest processing capabilities. Every computer is capable of sending and receiving information to and from every other computer, as shown in Fig.1.

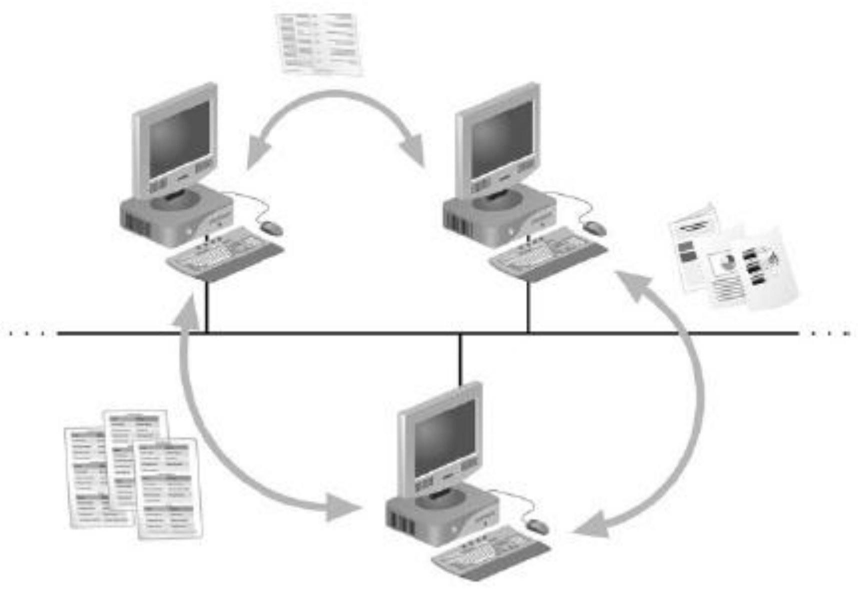


Fig.1: Resource sharing on a simple peer-to-peer network

Experiment 6

Aim: Implementation of Client- Server network

Theory:

Another way of designing a network is to use a central computer, known as a ***server***, to facilitate communication and resource sharing between other computers on the network, which are known as *clients.* Clients usually take the form of personal computers, also known as *workstations.*

A network that uses a server to enable clients to share data, data storage space, and devices is known as a *client/server network*. (The *term client/server architecture* is sometimes used to refer to the design of a network in which clients rely on servers for resource sharing and processing.) In terms of resource sharing and control, you can compare the client/server network to a public library. Just as a librarian manages the use of books and other media by patrons, a server manages the use of shared resources by clients. For example, if a patron does not have the credentials to check out books, the librarian prevents him from doing so. Similarly, a server allows only authorized clients to access its resources.

Every computer on a client/server network acts as a client or a server. (It's possible, but uncommon, for some computers to act as both.) Clients on a network can still run applications from and save data to their local hard disk. But by connecting to a server, they also have the option of using shared applications, data, and devices. Clients on a client/server network do not share their resources directly with each other, but rather use the server as an intermediary. Fig.1 illustrates how resources are shared on a client/server network.

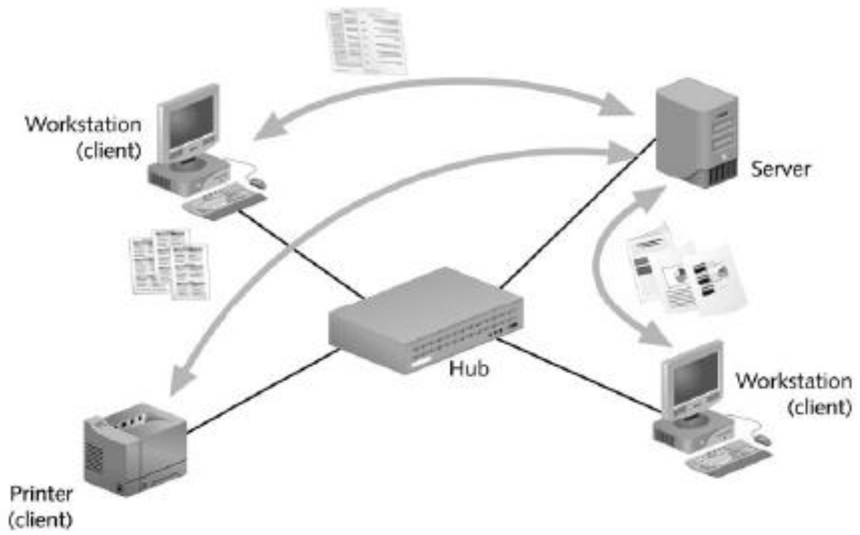


Fig.1. Resources sharing on a client/server network.

To function as a server, a computer must be running a *network operating system (NOS)***,** a special type of software designed to:

- Manage data and other resources for a number of clients Ensure that only authorized users access the network Control which type of files a user can open and read
- Restrict when and from where users can access the network Dictate which rules computers will use to communicate Supply applications to clients

Examples of popular network operating systems include Microsoft Windows Server 2003, Novell NetWare, Unix, and Linux. (By contrast, a standalone computer, or a client computer, uses a less-powerful operating system, such as Windows XP.)

Usually, servers have more memory, processing, and storage capacity than clients. They may even be equipped with special hardware designed to provide network management functions beyond that provided by the network operating system. For example, a server may contain an extra hard disk and specialized software so that if the primary hard disk fails, the secondary hard disk automatically takes its place.

Although client/server networks are typically more complex in their design and maintenance than peer-to-peer networks, they offer many advantages over peer-to-peer networks, such as:

- User log on accounts and passwords for anyone on a server-based network can be assigned in one place.
- Access to multiple shared resources (such as data files or printers) can be centrally granted to a single user or groups of users.
- Problems on the network can be tracked, diagnosed, and often fixed from one location.
- Servers are optimized to handle heavy processing loads and dedicated to handling requests from clients, enabling faster response time.
- Because of their efficient processing and larger disk storage, servers can connect more than a handful of computers on a network.

Together, these advantages make client/server networks more easily manageable, more secure, and more powerful than peer-to-peer networks. They are also more *scalable* that is, they can be more easily added onto and extended than peer-to-peer networks. Because client/server networks are the most popular type of network for medium and large-scale organizations.

Experiment No:7

Aim: Implementation of star topology.

Equipment Needed:

- ST5002A Kit,
- 4 CAT5 Cables

Theory:

1. Star Topology

In a Star Topology each computer is directly connected to the centralized Hub or a Switch. In this way, when computer A sends a data packet for computer B, the data flows through the Hub or Switch to which both computer A and B are connected. Different types of cables can be used in this scenario like coaxial cable, fiber optic cable and twisted pair cable.

The star topology reduces the chance of network failure by connecting all of the systems to a central node. When applied to a bus-based network, this central hub rebroadcasts all transmissions received from any peripheral node to all peripheral nodes on the network, sometimes including the originating node. All peripheral nodes may thus communicate with all others by transmitting to, and receiving from, the central node only. The failure of a transmission line linking any peripheral node to the central node will result in the isolation of that peripheral node from all others, but the rest of the systems will be unaffected.

If the star central node is passive, the originating node must be able to tolerate the reception of an echo of its own transmission, delayed by the two-way transmission time (i.e. to and from the central node) plus any delay generated in the central node. An active star network has an active central node that usually has the means to prevent echo-related problems.

A tree topology can be viewed as a collection of star networks arranged in a hierarchy. This tree has individual peripheral nodes (i.e. leaves) which are required to transmit to and receive from one other node only and are not required to act as repeaters or regenerators. Unlike the star network, the function of the central node may be distributed. As in the conventional star network, individual nodes may thus still be isolated from the network by a single-point failure of a transmission path to the node. If a link connecting a leaf fails, that leaf is isolated; if a connection to a non-leaf node fails, an entire section of the network becomes isolated from the rest.

In order to alleviate the amount of network traffic that comes from broadcasting everything everywhere, more advanced central nodes were developed that would keep track of the identities of different systems connected to the network. These network switches will "learn" the layout of the network by first broadcasting data packets everywhere, then observing where response packets come from.

Comparing Star networks to other types of network: Advantages

- Easy to implement and extend, even in large networks
- Well suited for temporary networks (quick setup)
- The failure of a non-central node will not have major effects on the functionality of the network

Disadvantages

- Limited cable length and number of stations
- Maintenance costs may be higher in the long run
- Failure of the central node can disable the entire network

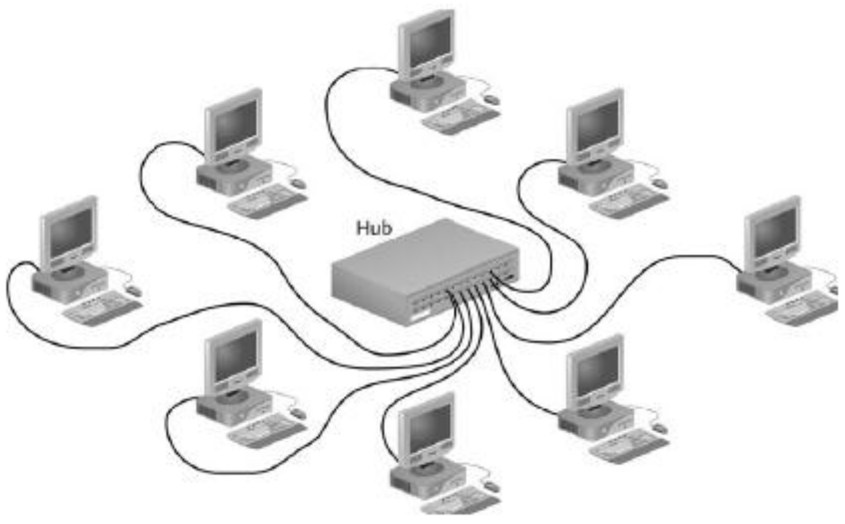
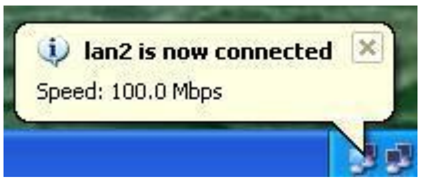


Fig.1: Star topology

Procedure:

1. Switch on the Systems.
2. Set the IP Addresses, Default gateway & Computer name of all the systems which you want to connect to the ST5002A kit.(If they are already set then no need to change them, just note down them on a paper for the reference).
3. Connect CAT5 cables on ST5002A kit in 'Star Topology' section.
4. Connect other ends of cables to the individual system's RJ-45 Connector on the back panel of CPU
5. Check the Status of LEDs on the ST5002A kit. The upper LED on each node shows the LINK Connection or LINK Establishment. The lower LED shows the Speed indication, if it is glowing then the speed is 100 Mbps otherwise the speed is 10 Mbps.

6. On the bottom right corner of the desktop, a popup message box can be seen as shown in Fig.2.



lan2 is now connected

Speed: 100.0 Mbps

7. Change the sharing properties of any folder on all systems
8. Now one can share the files.

Conclusions:

Experiment 8

Objective: Ethernet LAN protocol to create scenario and Study the performance of CSMA/CD (Carrier Sense Multiple Access with Collision Detection) Protocol through simulation.

Equipments Needed:

ST5002A

ST5002A Software

4- CAT5 cables.

Theory:

A network's *access method* is its method of controlling how network nodes access the communications channel. In comparing a network to a highway, the on-ramps would be one part of the highway's access method. A busy highway might use stoplights at each on-ramp to allow only one person to merge into traffic every five seconds. After merging, cars are restricted to lanes and each lane is limited as to how many cars it can hold at one time. All of these highway controls are designed to avoid collisions and help drivers get to their destinations. On networks, similar restrictions apply to the way in which multiple computers share a finite amount of bandwidth on a network. These controls make up the network's access method.

The access method used in Ethernet is called *CSMA/CD (Carrier Sense Multiple Access with Collision Detection)*. All Ethernet networks, independent of their speed or frame type, rely on CSMA/CD. To understand Ethernet, you must first understand CSMA/CD. Take a minute to think about the full name "Carrier Sense Multiple Access with Collision Detection." The term "Carrier Sense" refers to the fact that Ethernet NICs listen on the network and wait until they detect (or sense) that no other nodes are transmitting data over the signal (or carrier) on the communications channel before they begin to transmit. The term "Multiple Access" refers to the fact that several Ethernet nodes can be connected to a network and can monitor traffic, or access the media, simultaneously.

In CSMA/CD, when a node wants to transmit data it must first access the transmission media and determine whether the channel is free. If the channel is not free, it waits and checks again after a very brief amount of time. If the channel is free, the node transmits its data. Any node can transmit data after it determines that the channel is free. But what if two nodes simultaneously check the channel, determine that it's free, and begin to transmit? When this happens, their two transmissions interfere with each other; this is known as a *collision*.

The last part of the term CSMA/CD, "collision detection," refers to the way nodes respond to a collision. In the event of a collision, the network performs a series of steps known as the collision detection routine. If a node's NIC determines that its data has been involved in a collision, it immediately stops transmitting. Next, in a process called *jamming*, the NIC issues a special 32-bit sequence that indicates to the rest of the network nodes that its previous transmission was faulty and that those data frames are invalid. After waiting, the NIC determines if the line is again available; if it is available, the NIC retransmits its data.

On heavily trafficked networks, collisions are fairly common. It is not surprising that the more nodes there are transmitting data on a network, the more collisions that will take place. (Although a collision rate greater than 5% of all traffic is unusual and may point to a problematic NIC or poor cabling on the network.) When an Ethernet network grows to include a particularly large number of nodes, you may see performance suffer as a result of collisions. This “critical mass” number depends on the type and volume of data that the network regularly transmits. Collisions can corrupt data or truncate data frames, so it is important that the network detect and compensate for them. Fig. 1 depicts the way CSMA/CD regulates data flow to avoid and, if necessary, detect collisions.

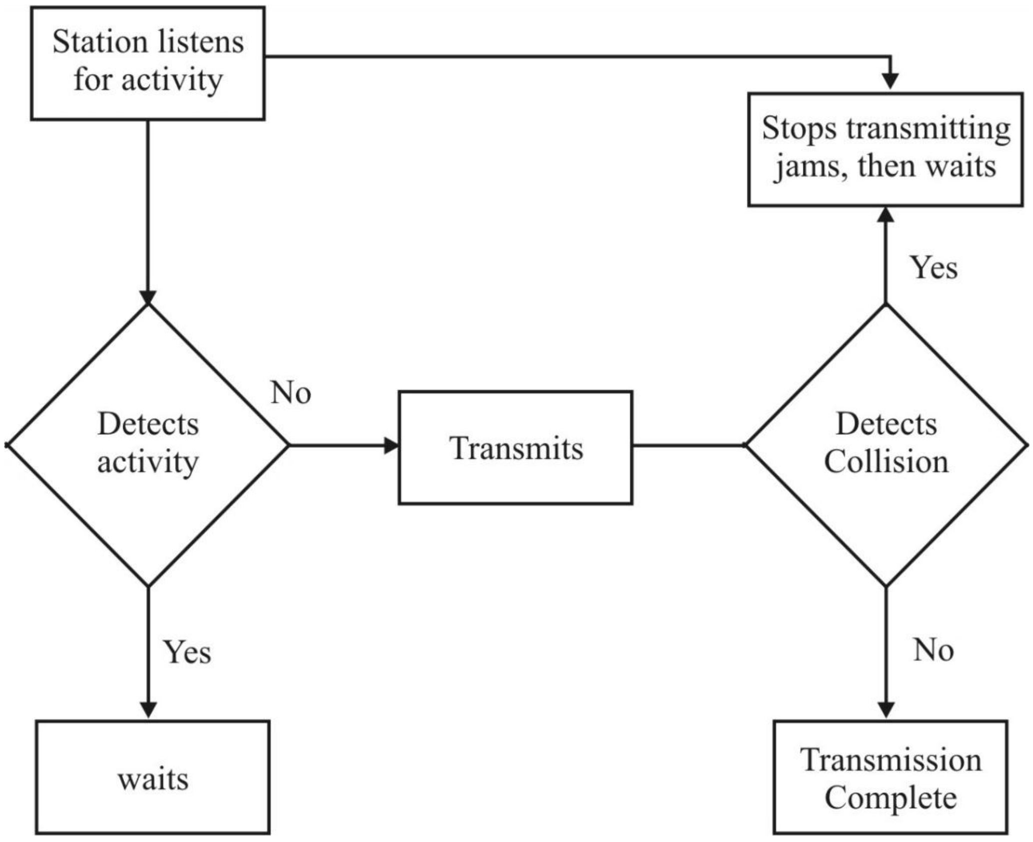


Fig 1. CSMA/CD Process

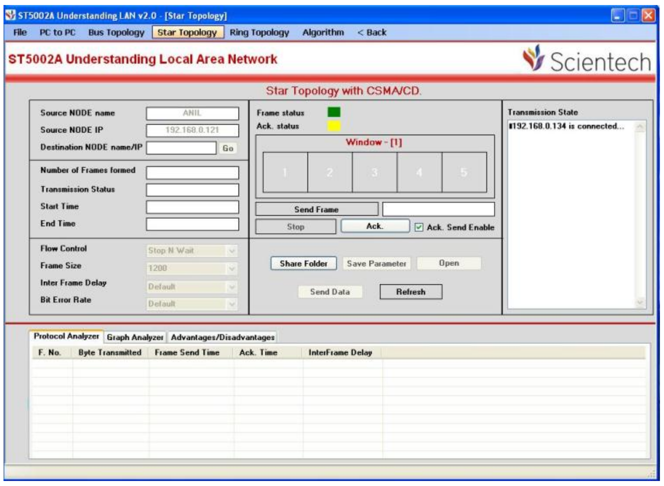
On an Ethernet network, a *collision domain* is the portion of a network in which collisions occur if two nodes transmit data at the same time. When designing an Ethernet network, it’s important to note that because repeaters simply regenerate any signal they receive, they repeat collisions just as they repeat data. Thus, connecting multiple parts of a network with repeaters results in a larger collision domain. Higher-layer connectivity devices, such as switches and routers, however, can separate collision domains.

Collision domains play a role in the Ethernet cabling distance limitations. For example, if there is more than 100 meters distance between two nodes on a segment connected to the same 100BASE-TX network bus, data propagation delays will be too long for CSMA/CD to be effective. A *data propagation delay* is the length of time data takes to travel from one point on the segment to another point.

When data takes a long time, CSMA/CD's collision detection routine cannot identify collisions accurately. In other words, one node on the segment might begin its CSMA/CD routine and determine that the channel is free even though a second node has begun transmitting, because the second node's data is taking so long to reach the first node. At rates of 100 or 1000 Mbps, data travels so quickly that NICs can't always keep up with the collision detection and retransmission routines. For example, because of the speed employed on a 100BASE-TX network, the window of time for the NIC to both detect and compensate for the error is much less than that of a 10BASE-T network. To minimize undetected collisions, 100BASE-TX networks can support only a maximum of three network segments connected with two hubs, whereas 10BaseT buses can support a maximum of five network segments connected with four hubs. This shorter path reduces the highest potential propagation delay between nodes.

Procedure:

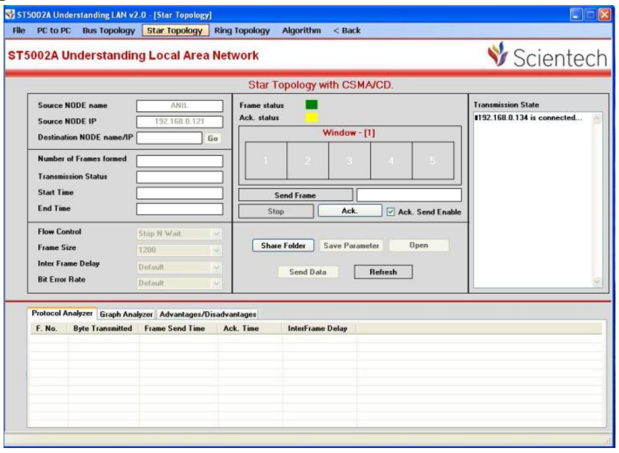
1. Connect the computers to the ST5002A using RJ45 cables
2. Switch on the power supply of ST5002A.
3. Open **ST5002A** software
4. Click on 'Star Topology'



Source NODE name: ANIL
Source NODE IP: 192.168.0.121
Destination NODE name/IP:
Number of Frames formed:
Transmission Status:
Start Time:
End Time:
Flow Control: Stop N Wait
Frame Size: 1200
Inter Frame Delay: Default
Bit Error Rate: Default
Frame status: 🟩
Ack. status: 🟨
✓ Ack. Send Enable
Transmission State
192.168.0.134 is connected...

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

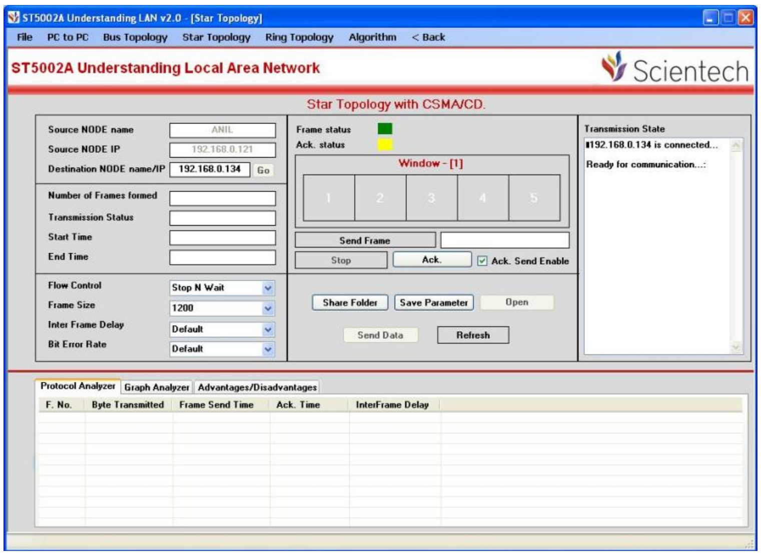
5. A window will appear on screen as shown.



Source NODE name: ANIL
Source NODE IP: 192.168.0.121
Destination NODE name/IP:
Number of Frames formed:
Transmission Status:
Start Time:
End Time:
Flow Control: Stop N Wait
Frame Size: 1200
Inter Frame Delay: Default
Bit Error Rate: Default
Frame status: 🟩
Ack. status: 🟨
✓ Ack. Send Enable
Transmission State
192.168.0.134 is connected...

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

6. You will find IP address and Computer name of each connected node.



Source NODE name	ANIL
Source NODE IP	192.168.0.121
Destination NODE name/IP	192.168.0.134
Number of Frames formed	
Transmission Status	
Start Time	
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default

Frame status 🟩
Ack. status 🟨
✓ Ack. Send Enable

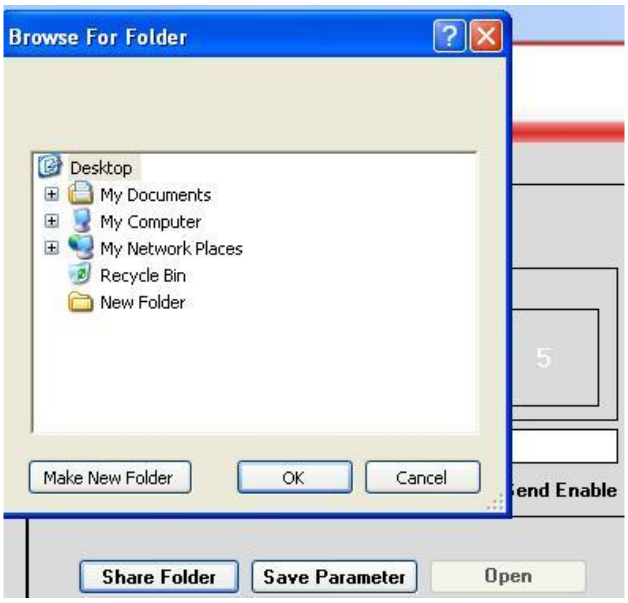
Transmission State

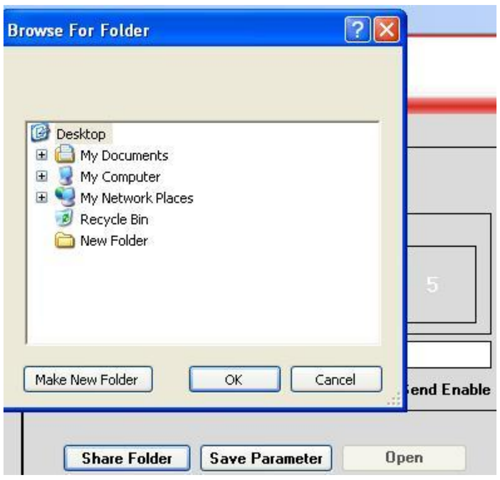
192.168.0.134 is connected...

Ready for communication...:

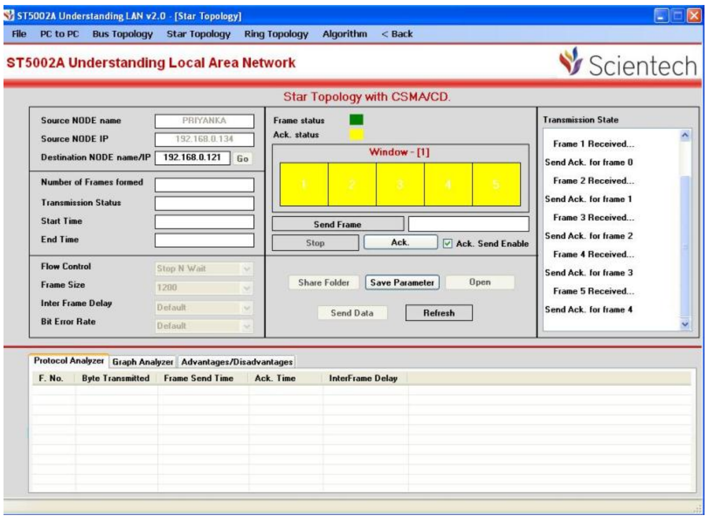
F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

7. Click on 'Sharing Folder' and select a folder except 'Desktop' and Drives. You can also make a new folder.





8. Repeat Step-6 on each node connected to ST5002A
9. Now enter the destination node Name/IP Address.


ST5002A Understanding LAN v2.0 - [Star Topology]
File PC to PC Bus Topology Star Topology Ring Topology Algorithm < Back
ST5002A Understanding Local Area Network
Scientech
Star Topology with CSMA/CD.
Window - [1]
Send Frame
Stop
Ack.
Share Folder
Save Parameter
Open
Send Data
Refresh
Protocol Analyzer
Graph Analyzer
Advantages/Disadvantages

Source NODE name	PRIYANKA
Source NODE IP	192.168.0.134
Destination NODE name/IP	192.168.0.121
Number of Frames formed	
Transmission Status	
Start Time	
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default

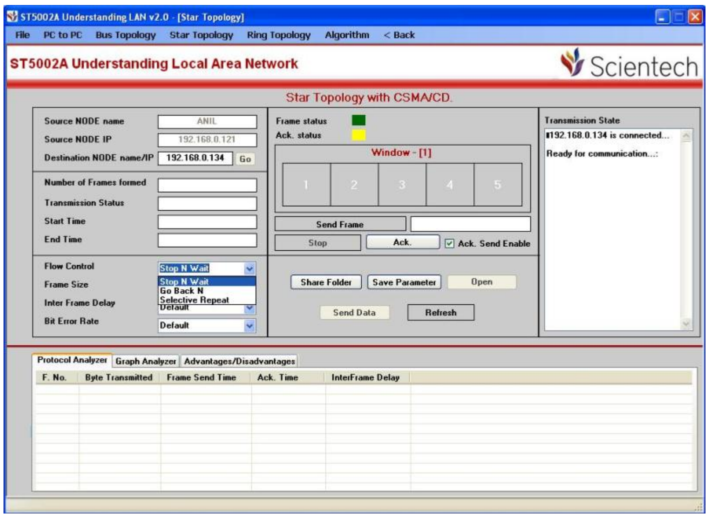
Frame status 🟩
Ack. status 🟨
✓ Ack. Send Enable

Transmission State

```
  Frame 1 Received...
Send Ack. for frame 0
  Frame 2 Received...
Send Ack. for frame 1
  Frame 3 Received...
Send Ack. for frame 2
  Frame 4 Received...
Send Ack. for frame 3
  Frame 5 Received...
Send Ack. for frame 4
```

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

10. Now choose the parameters (The parameters should be same on each node connected to the ST5002A).


ST5002A Understanding LAN v2.0 - [Star Topology]
File PC to PC Bus Topology Star Topology Ring Topology Algorithm < Back
ST5002A Understanding Local Area Network
Scientech
Star Topology with CSMA/CD.
Window - [1]
Send Frame
Stop
Ack.
Share Folder
Save Parameter
Open
Send Data
Refresh
Protocol Analyzer
Graph Analyzer
Advantages/Disadvantages

Source NODE name	ANIL
Source NODE IP	192.168.0.121
Destination NODE name/IP	192.168.0.134
Number of Frames formed	
Transmission Status	
Start Time	
End Time	
Flow Control	Stop N Wait (Stop N Wait / Go Back N / Selective Repeat)
Frame Size	
Inter Frame Delay	Default
Bit Error Rate	Default

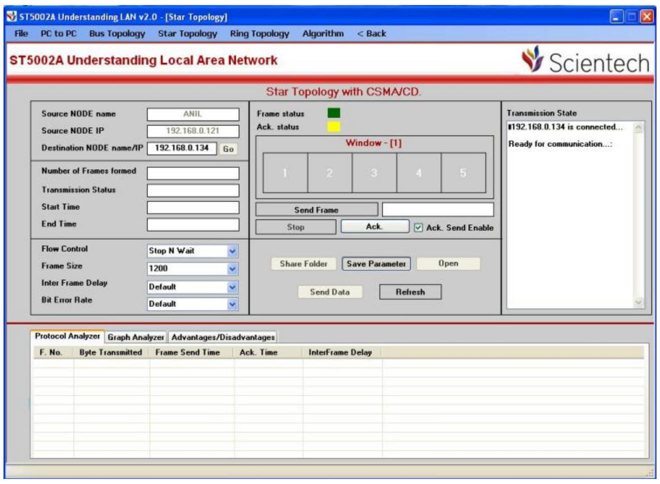
Frame status 🟩
Ack. status 🟨
✓ Ack. Send Enable

Transmission State

```
192.168.0.134 is connected...
Ready for communication...:
```

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

11. Click on ‘Save Parameters’



ST5002A Understanding Local Area Network

Star Topology with CSMA/CD.

Source NODE name: ANIL
Source NODE IP: 192.168.0.121
Destination NODE name/IP: 192.168.0.134
Number of Frames formed:
Transmission Status:
Start Time:
End Time:
Flow Control: Stop N Wait
Frame Size: 1200
Inter Frame Delay: Default
Bit Error Rate: Default

Frame status 🟩
Ack. status 🟨
Window - [1]

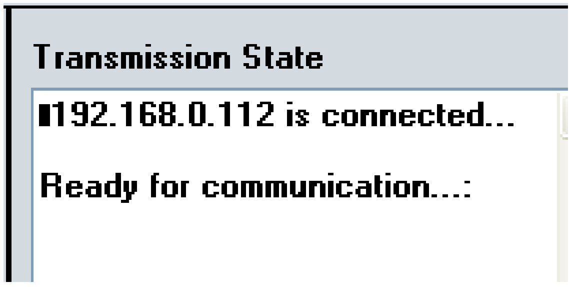
1	2	3	4	5

☑ Ack. Send Enable

Transmission State
▮192.168.0.134 is connected...
Ready for communication...:

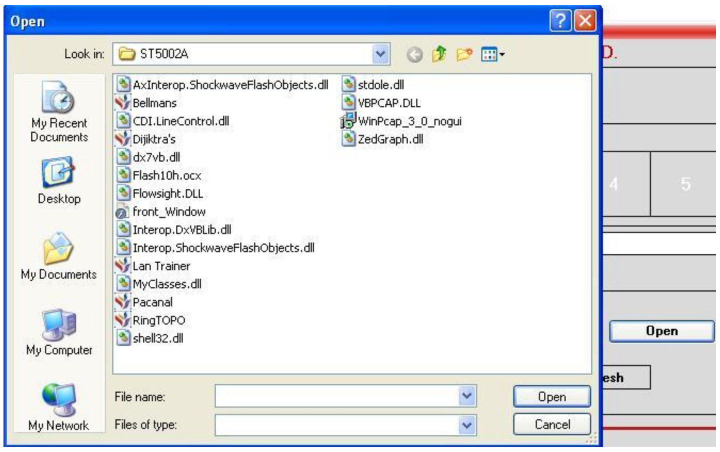
F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

12. ‘Ready for communication’ message will be displayed in status window of both source node & destination node.



Transmission State
▮192.168.0.112 is connected...
Ready for communication...:

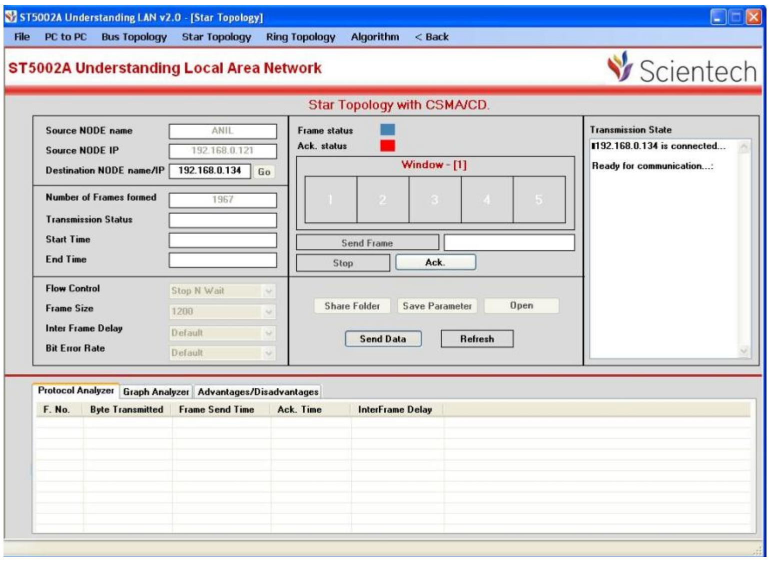
13. Click on ‘Open’ to open a .txt file to transmit.



AxInterop.ShockwaveFlashObjects.dll
Bellmans
CDI.LineControl.dll
Dijiktra's
dx7vb.dll
Flash10h.ocx
Flowsight.DLL
front_Window
Interop.DxVBLib.dll
Interop.ShockwaveFlashObjects.dll
Lan Trainer
MyClasses.dll
Pacanal
RingTOPO
shell32.dll
stdole.dll
VBPCAP.DLL
WinPcap_3_0_nogui
ZedGraph.dll

14. Now click on open

15. Now click on send data


ST5002A Understanding LAN v2.0 - [Star Topology]
File PC to PC Bus Topology Star Topology Ring Topology Algorithm < Back
ST5002A Understanding Local Area Network
Scientech
Star Topology with CSMA/CD.
Window - [1]
1 2 3 4 5
Send Frame Stop Ack.
Share Folder Save Parameter Open
Send Data Refresh
Protocol Analyzer Graph Analyzer Advantages/Disadvantages

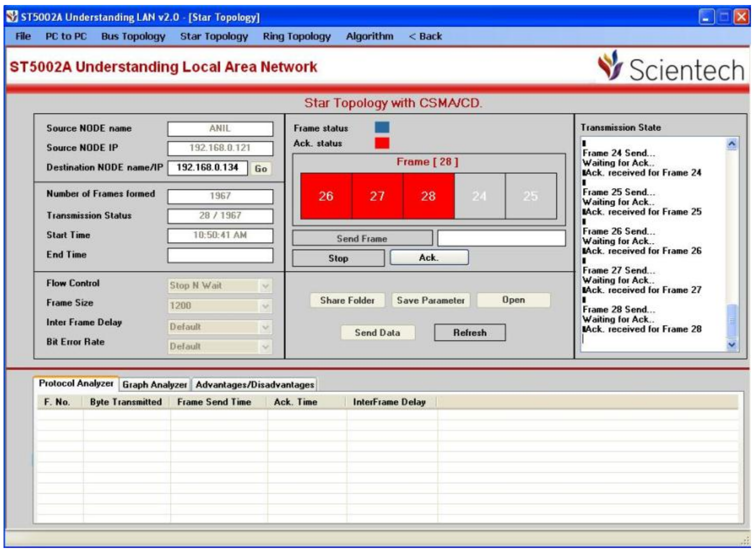
Source NODE name	ANIL
Source NODE IP	192.168.0.121
Destination NODE name/IP	192.168.0.134
Number of Frames formed	1967
Transmission Status	
Start Time	
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default
Frame status	🟦
Ack. status	🟥

Transmission State

```
192.168.0.134 is connected...
Ready for communication...:
```

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

16. One can see a frame status with blue color and Acknowledge status with red color.


ST5002A Understanding LAN v2.0 - [Star Topology]
File PC to PC Bus Topology Star Topology Ring Topology Algorithm < Back
ST5002A Understanding Local Area Network
Scientech
Star Topology with CSMA/CD.
Frame [28]
26 27 28 24 25
Send Frame Stop Ack.
Share Folder Save Parameter Open
Send Data Refresh
Protocol Analyzer Graph Analyzer Advantages/Disadvantages

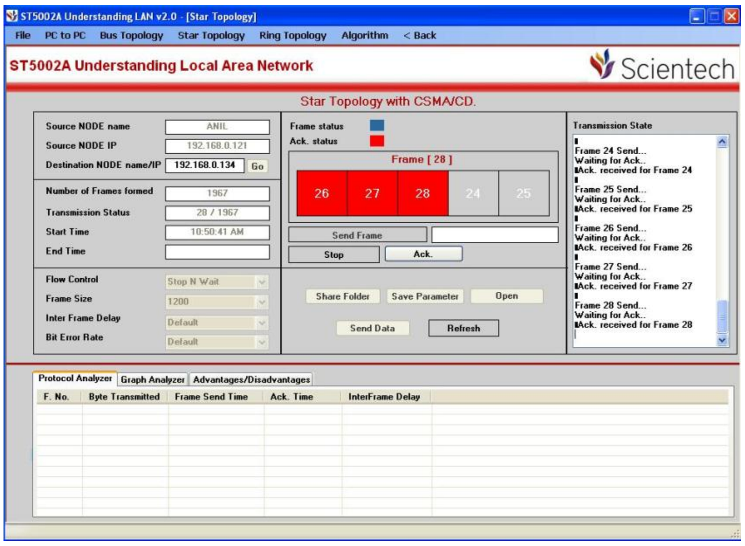
Source NODE name	ANIL
Source NODE IP	192.168.0.121
Destination NODE name/IP	192.168.0.134
Number of Frames formed	1967
Transmission Status	28 / 1967
Start Time	10:50:41 AM
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default
Frame status	🟦
Ack. status	🟥

Transmission State

```
Frame 24 Send...
Waiting for Ack..
Ack. received for Frame 24
Frame 25 Send...
Waiting for Ack..
Ack. received for Frame 25
Frame 26 Send...
Waiting for Ack..
Ack. received for Frame 26
Frame 27 Send...
Waiting for Ack..
Ack. received for Frame 27
Frame 28 Send...
Waiting for Ack..
Ack. received for Frame 28
```

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

17. One can see number of frames formed.


ST5002A Understanding LAN v2.0 - [Star Topology]
File PC to PC Bus Topology Star Topology Ring Topology Algorithm < Back
ST5002A Understanding Local Area Network
Scientech
Star Topology with CSMA/CD.
Frame [28]
26 27 28 24 25
Send Frame Stop Ack.
Share Folder Save Parameter Open
Send Data Refresh
Protocol Analyzer Graph Analyzer Advantages/Disadvantages

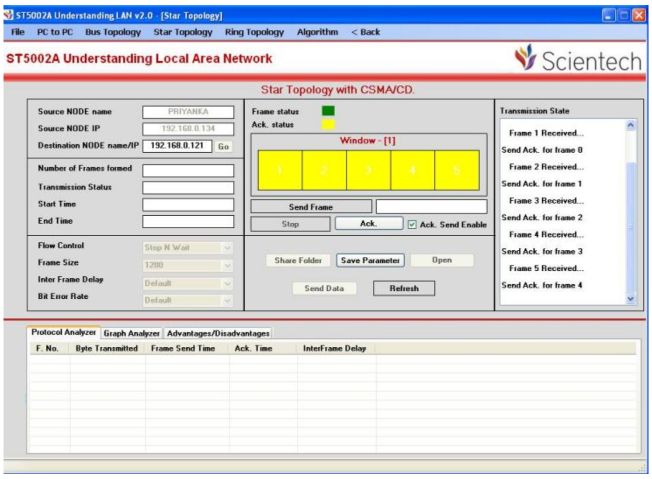
Source NODE name	ANIL
Source NODE IP	192.168.0.121
Destination NODE name/IP	192.168.0.134
Number of Frames formed	1967
Transmission Status	28 / 1967
Start Time	10:50:41 AM
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default
Frame status	🟦
Ack. status	🟥

Transmission State

```
Frame 24 Send...
Waiting for Ack..
Ack. received for Frame 24
Frame 25 Send...
Waiting for Ack..
Ack. received for Frame 25
Frame 26 Send...
Waiting for Ack..
Ack. received for Frame 26
Frame 27 Send...
Waiting for Ack..
Ack. received for Frame 27
Frame 28 Send...
Waiting for Ack..
Ack. received for Frame 28
```

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

18. One can stop transfer of file by removing check from Ack. Send enable button on receiver side here green color represents frame received and yellow color represents Acknowledgement sent.


ST5002A Understanding LAN v2.0 - [Star Topology]

File | PC to PC | Bus Topology | Star Topology | Ring Topology | Algorithm | < Back

ST5002A Understanding Local Area Network

Star Topology with CSMA/CD.

Source NODE name	PRIYANKA
Source NODE IP	192.168.0.134
Destination NODE name/IP	192.168.0.121
Number of Frames formed	
Transmission Status	
Start Time	
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default

Frame status 🟩
Ack. status 🟨

Window - [1]

1 2 3 4 5

[✓] Ack. Send Enable

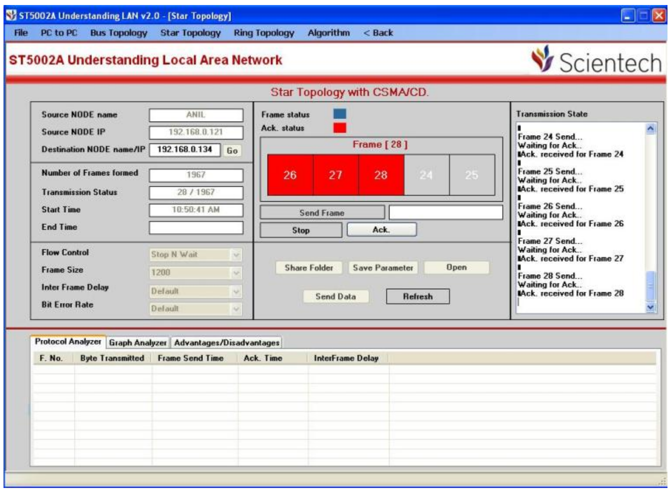
Transmission State

Frame 1 Received...
Send Ack. for frame 0
Frame 2 Received...
Send Ack. for frame 1
Frame 3 Received...
Send Ack. for frame 2
Frame 4 Received...
Send Ack. for frame 3
Frame 5 Received...
Send Ack. for frame 4

Protocol Analyzer | Graph Analyzer | Advantages/Disadvantages

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

19. Observe on sender side same status waiting for Acknowledgement.


ST5002A Understanding LAN v2.0 - [Star Topology]

File | PC to PC | Bus Topology | Star Topology | Ring Topology | Algorithm | < Back

ST5002A Understanding Local Area Network

Star Topology with CSMA/CD.

Source NODE name	ANIL
Source NODE IP	192.168.0.121
Destination NODE name/IP	192.168.0.134
Number of Frames formed	1967
Transmission Status	28 / 1967
Start Time	10:50:41 AM
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default

Frame status 🟦
Ack. status 🟥

Frame [28]

26 27 28 24 25

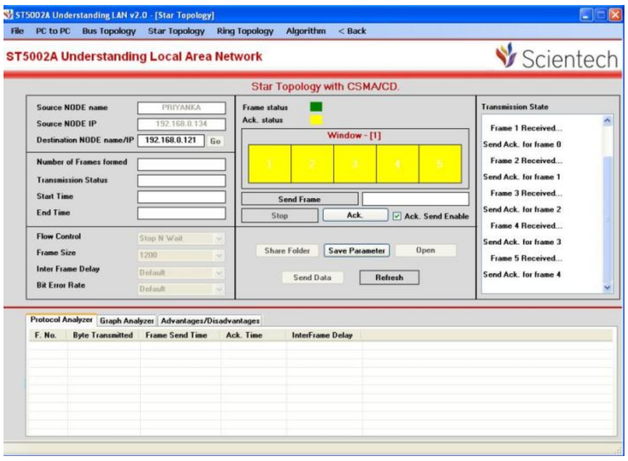
Transmission State

Frame 24 Send...
Waiting for Ack..
Ack. received for Frame 24
Frame 25 Send...
Waiting for Ack..
Ack. received for Frame 25
Frame 26 Send...
Waiting for Ack..
Ack. received for Frame 26
Frame 27 Send...
Waiting for Ack..
Ack. received for Frame 27
Frame 28 Send...
Waiting for Ack..
Ack. received for Frame 28

Protocol Analyzer | Graph Analyzer | Advantages/Disadvantages

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

20. To start transfer of file continuously check on Ack. Send enable button on receiver side


ST5002A Understanding LAN v2.0 - [Star Topology]

File | PC to PC | Bus Topology | Star Topology | Ring Topology | Algorithm | < Back

ST5002A Understanding Local Area Network

Star Topology with CSMA/CD.

Source NODE name	PRIYANKA
Source NODE IP	192.168.0.134
Destination NODE name/IP	192.168.0.121
Number of Frames formed	
Transmission Status	
Start Time	
End Time	
Flow Control	Stop N Wait
Frame Size	1200
Inter Frame Delay	Default
Bit Error Rate	Default

Frame status 🟩
Ack. status 🟨

Window - [1]

1 2 3 4 5

[✓] Ack. Send Enable

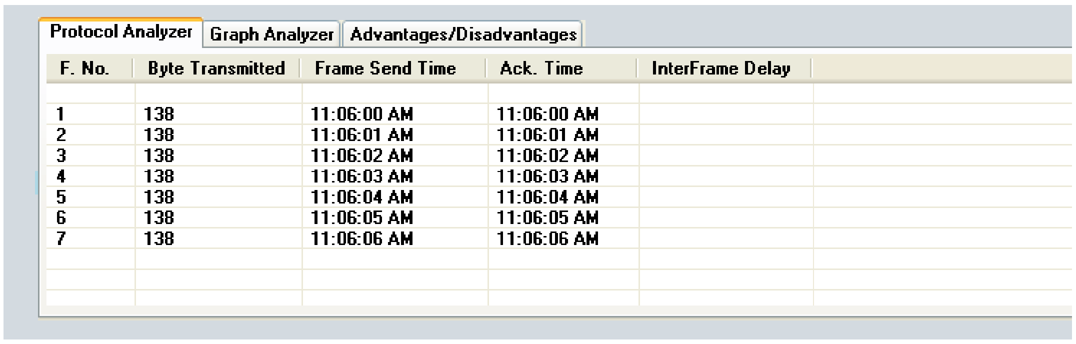
Transmission State

Frame 1 Received...
Send Ack. for frame 0
Frame 2 Received...
Send Ack. for frame 1
Frame 3 Received...
Send Ack. for frame 2
Frame 4 Received...
Send Ack. for frame 3
Frame 5 Received...
Send Ack. for frame 4

Protocol Analyzer | Graph Analyzer | Advantages/Disadvantages

F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay

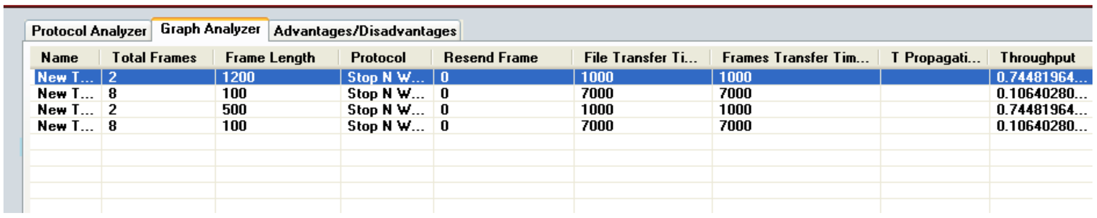
21. One can see the detail timings of each packet.



F. No.	Byte Transmitted	Frame Send Time	Ack. Time	InterFrame Delay
1	138	11:06:00 AM	11:06:00 AM	
2	138	11:06:01 AM	11:06:01 AM	
3	138	11:06:02 AM	11:06:02 AM	
4	138	11:06:03 AM	11:06:03 AM	
5	138	11:06:04 AM	11:06:04 AM	
6	138	11:06:05 AM	11:06:05 AM	
7	138	11:06:06 AM	11:06:06 AM	

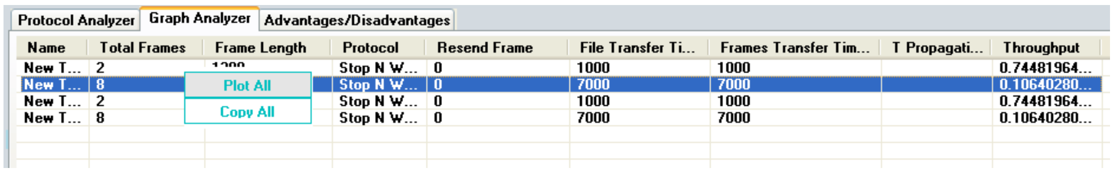
22. Status window will show, 'File Transmitted…..'

23. One can also observe throughput of same file when using different frame size



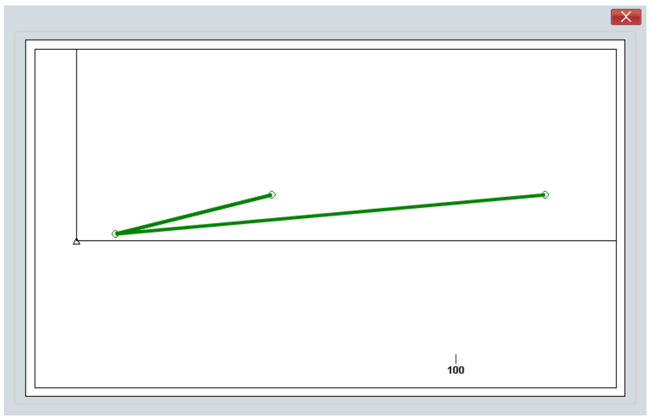
Name	Total Frames	Frame Length	Protocol	Resend Frame	File Transfer Ti...	Frames Transfer Tim...	T Propagati...	Throughput
New T...	2	1200	Stop N W...	0	1000	1000		0.74481964...
New T...	8	100	Stop N W...	0	7000	7000		0.10640280...
New T...	2	500	Stop N W...	0	1000	1000		0.74481964...
New T...	8	100	Stop N W...	0	7000	7000		0.10640280...

24. Right click with mouse and user will get listing plot all



Name	Total Frames	Frame Length	Protocol	Resend Frame	File Transfer Ti...	Frames Transfer Tim...	T Propagati...	Throughput
New T...	2	1200	Stop N W...	0	1000	1000		0.74481964...
New T...	8		Stop N W...	0	7000	7000		0.10640280...
New T...	2		Stop N W...	0	1000	1000		0.74481964...
New T...	8		Stop N W...	0	7000	7000		0.10640280...

25. And select plot all a graph will be displayed user can also zoom particular area by selecting that area



Conclusion:

Experiment No.9

Objective: Implementation of Bus topology using 10 Base 2

Equipments Needed:

ST5002A Kit

4 RJ45 Cables

Theory:

A *bus topology* consists of a single cable connecting all nodes on a network without intervening connectivity devices. The single cable is called the *bus* and can support only one channel for communication; as a result, every node shares the bus's total capacity. Most bus Networks-for example, Thinnest and Thickest-use coaxial cable as their physical medium.

On a bus topology network, devices share the responsibility for getting data from one point to another. Each node on a bus network passively listens for data directed to it. When one node wants to transmit data to another node, it broadcasts an alert to the entire network, informing all nodes that a transmission is being sent; the destination node then picks up the transmission. Nodes other than the sending and receiving nodes ignore the message.

For example, suppose that you want to send an instant message to your friend XYZ, who works across the hall, asking whether he wants to have lunch with you. You click the Send button after typing your message, and the data stream that contains your message is sent to your NIC. Your NIC then sends a message across the shared wire that essentially says, "I have a message for XYZ's computer." The message passes by every NIC between your computer and XYZ's computer until XYZ's computer recognizes that the message is meant for it and responds by accepting the data.

At the ends of each bus network 50-ohm resistors known as terminators. Terminators stop Signals after they have reached the end of the wire. Without these devices, signals on a bus network would travel endlessly between the two ends of the network-a phenomenon known as signal bounce-and new signals could not get through. To understand this concept, imagine that you and a partner, standing at opposite sides of a canyon, are yelling to each other. When you call out, your words echo; when your partner replies, his words also echo. Now imagine that the echoes never fade. After a short while, you could not continue conversing because all of the previously generated sound waves would still be bouncing around, creating too much noise for you to hear anything else. On a network, terminators prevent this problem by halting the transmission of old signals. In some cases, a hub provides termination for one end of a segment. A bus network must also be grounded at one end to help remove static electricity that could adversely affect the signal. Fig.1 depicts a terminated bus network.

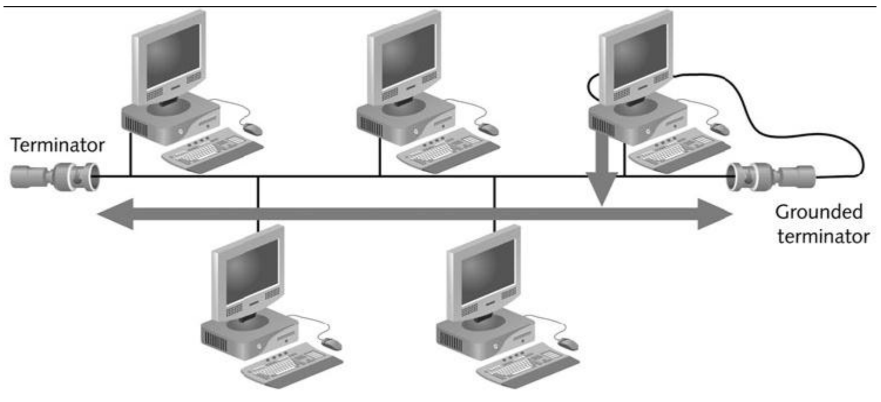


Fig 1 A terminated bus topology network

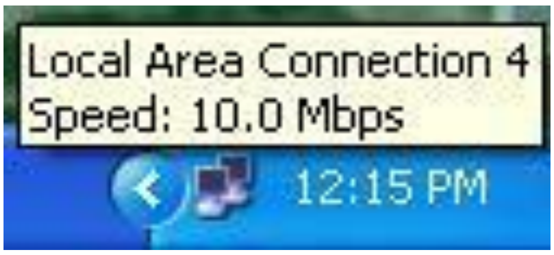
Although networks based on a bus topology are relatively inexpensive to set up, they do not scale well. As you add more nodes, the network's performance degrades. Because of the single-channel limitation, the more nodes on a bus network, the more slowly the network will transmit and deliver data. For example, suppose a bus network in your small office supports two workstations and a server, and saving a file to the server takes two seconds. During that time, your NIC first checks the communication channel to ensure it is free, then issues data directed to the server. When the data reaches the server, the server accepts it. Suppose, however, that your business experiences tremendous growth, and you add five workstations during one weekend. The following Monday, when you attempt to save a file to the server, the save process might take five seconds, because the new workstations may also be using the communications channel, and your workstation may have to wait for a chance to transmit. As this example illustrates, a bus topology is rarely practical for networks with more than a dozen workstations.

Bus networks are also difficult to troubleshoot, because it is a challenge to identify fault locations. To understand why, think of the game called "telephone," in which one person whispers a phrase into the ear of the next person, who whispers the phrase into the ear of another person, and so on, until the final person in line repeats the phrase aloud. The vast majority of the time, the phrase recited by the last person bears little resemblance to the original phrase. When the game ends, it's hard to determine precisely where in the chain the individual errors cropped up. Similarly, errors may occur at any intermediate point on a bus network, but at the receiving end it's possible to tell only that an error occurred. Finding the source of the error can prove very difficult.

A final disadvantage to bus networks is that they are not very fault-tolerant, because a break or a defect in the bus affects the entire network. As a result, and because of the other disadvantages associated with this topology, you will rarely see a network run on a pure bus topology. You may, however, encounter hybrid topologies that include a bus component.

Procedure:

1. You have to install SPX / IPX protocol; this is given in detail.
2. Switch on the systems
3. Connect RJ45 to RJ45 cables between ST5002A kit & systems
4. Connect End Terminators on the both end of Back bone
5. Here you have to disable the TCP/IP Properties of NICs.


Local Area Connection 4
Speed: 10.0 Mbps
12:15 PM

6. At the bottom right corner of desktop you can see the popup window as shown in figure 132.
7. Now you can share the files of systems connected in Bus topology.
8. If you remove the End Terminators, then you find that you are unable to share the files in the network.

The detail study of TOKEN BUS protocol; which we normally used in bus topology is given in experiment number 17.

Conclusion:

Experiment No.10

Objective: To create the scenario and study the performance of token bus protocols through simulation.

Equipments needed:

ST5002A.

ST5002A software.

4-RJ45 to RJ45 cables.

Two End Terminators.

Theory:

IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange) is a protocol originally developed by Xerox, then modified and adopted by Novell in the 1980s for its NetWare network operating system. IPX/SPX is required to ensure the interoperability of LANs running NetWare versions 3.2 and lower and can be used with LANs running higher versions of the NetWare operating system. On versions 5.0 and higher of NetWare, IPX/SPX has been replaced by TCP/IP as the default protocol. You will probably only use IPX/SPX if your clients must connect with older NetWare systems. To ensure interoperability, other operating systems can use IPX/SPX. Microsoft's implementation of IPX/SPX is called NWLink.

IPX/SPX, like TCP/IP, is a combination of protocols that reside at different layers of the OSI Model. Also like TCP/IP, IPX/SPX carries network addressing information, so it is routable. The IPX and SPX Protocols. The core protocols of IPX/SPX provide services at the Transport and Network layers of the OSI Model. As you might guess, the most significant core protocols are IPX and SPX.

IPX (Internetwork Packet Exchange) operates at the Network layer of the OSI Model and provides logical addressing and internetworking services, similar to IP in the TCP/IP suite. Like IP, IPX also uses datagrams to transport data and its datagrams also contain source and destination addresses. Furthermore, IPX is a connectionless service because it does not require a session to be established before it transmits, and it does not guarantee that data will be delivered in sequence or without errors. In summary, it is an efficient sub protocol with limited capabilities. All IPX/SPX communication relies upon IPX, however, and upper-layer protocols handle the functions that IPX cannot perform.

SPX (Sequenced Packet Exchange) belongs to the Transport layer of the OSI Model. It works in tandem with IPX to ensure that data are received whole, in sequence, and error free. SPX, like TCP in the TCP/IP suite, is a connection-oriented protocol and therefore must verify that a session has been established with the destination node before it will transmit data. It can detect whether a packet was not received in its entire form. If it discovers a packet has been lost or corrupted, SPX will resend the packet.

The SPX information is encapsulated by IPX. That is, its fields sit inside the data field of the IPX datagram. The SPX packet, like the TCP segment, contains a number of fields to ensure data reliability. An SPX packet consists of a 42-byte header followed by 0 to 534 bytes of data. An SPX packet can be as small as 42 bytes (the size of its header) or as large as 576 bytes.

Addressing in IPX/SPX:

Just as with TCP/IP-based networks, IPX/SPX-based networks require that each node on a network be assigned a unique address to avoid communication conflicts. Because IPX is the component of the protocol that handles addressing, addresses on an IPX/SPX network are called *IPX addresses*. IPX addresses contain two parts: the network address (also known as the external network number) and the node address.

Maintaining network addresses for clients running IPX/SPX is somewhat easier than maintaining addresses for TCP/IP-based networks, because IPX/SPX-based networks primarily rely on the MAC address for each workstation. To begin, the network administrator chooses a network address when installing the (older) NetWare operating system on a server. The network address must be an 8-bit hexadecimal address, which means that each of its bits can have a value of either 0–9 or A–F. An example of a valid network address is 000008A2. The network address then becomes the first part of the IPX address on all nodes that use the particular server as their primary server.

The second part of an IPX address, the node address, is by default equal to the network device's MAC address. Because every network interface card should have a unique MAC address, no possibility of duplicating IPX addresses exists under this system (unless MAC addresses have been manually altered). In addition, the use of MAC addresses means that you need not configure addresses for the IPX/SPX protocol on each client workstation. Instead, they are already defined by the NIC. Adding a MAC address to the network address example used previously, a complete IPX address for a workstation on the network might be 000008A2:0060973E97F3.

Binding Protocols on a Windows XP Workstation:

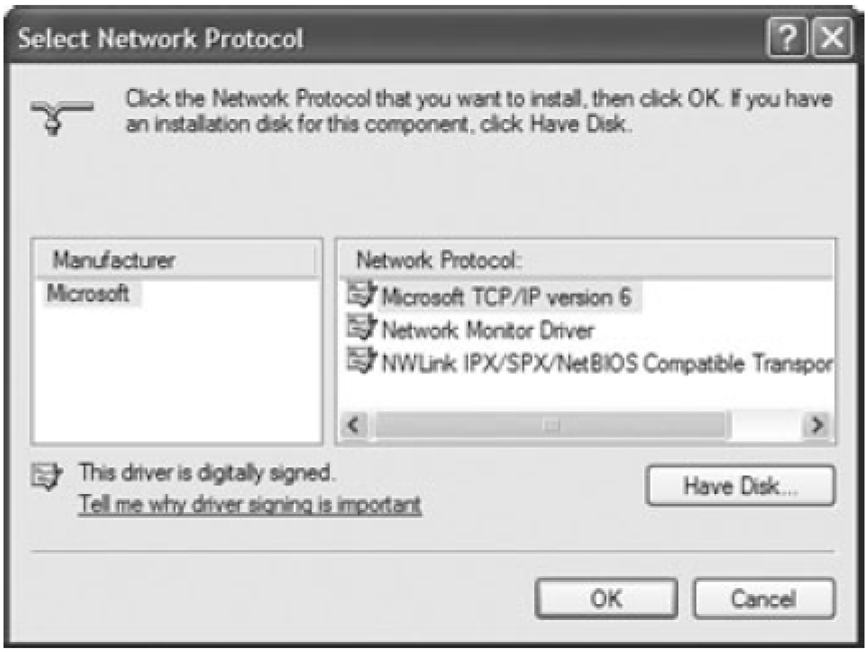
The protocols you install will depend on which operating system you are running. This section describes how to bind a protocol suite on a Windows XP client workstation. No equivalent procedure exists on a UNIX- or Linux-based computer, because UNIX and Linux only support the TCP/IP protocol suite, and the TCP/IP protocols are automatically bound to the network interface (or interfaces).

Core Network and Transport layer protocols are normally included with your computer's operating system. When enabled, these protocols attempt to bind with the network interfaces on your computer. *Binding* is the process of assigning one network component to work with another. You can manually bind protocols that are not already associated with a network interface. For optimal network performance, you should bind only those protocols that you absolutely need. For example, a Windows Server 2003 server will attempt to use bound protocols in the order in which they appear in the protocol listing until it finds the correct one for the response at hand. If not all bound protocols are necessary, this approach wastes processing time.

Normally, a workstation running the Windows XP operating system would, by default, have the TCP/IP protocol bound to its network interfaces. The following exercise shows you how to install the NWLink IPX/SPX/NetBIOS Compatible Transport protocol (which is not, by default, bound to interfaces) on a Windows XP workstation:

Log on to the workstation as an Administrator.

1. Click *Start*, then click *My Network Places*. The My Network Places window appears.
2. From the Network Tasks list, click *View network connections*. The Network Connections window appears.
3. Right-click the icon that represents your network adapter, and click *Properties* in the shortcut menu. The network adapter's Properties dialog box appears.
4. Click *Install…*. The Select Network Component Type dialog box appears.
5. From the list of network components, select *Protocol*, and then click *Add…*. The Select Network Protocol dialog box appears, as shown in Fig.1.


Select Network Protocol

Click the Network Protocol that you want to install, then click OK. If you have an installation disk for this component, click Have Disk.

Manufacturer: Microsoft

Network Protocol:
- Microsoft TCP/IP version 6
- Network Monitor Driver
- NWLink IPX/SPX/NetBIOS Compatible Transpor

This driver is digitally signed.
Tell me why driver signing is important

Fig. 1. The Windows XP Select Network Protocol dialog box

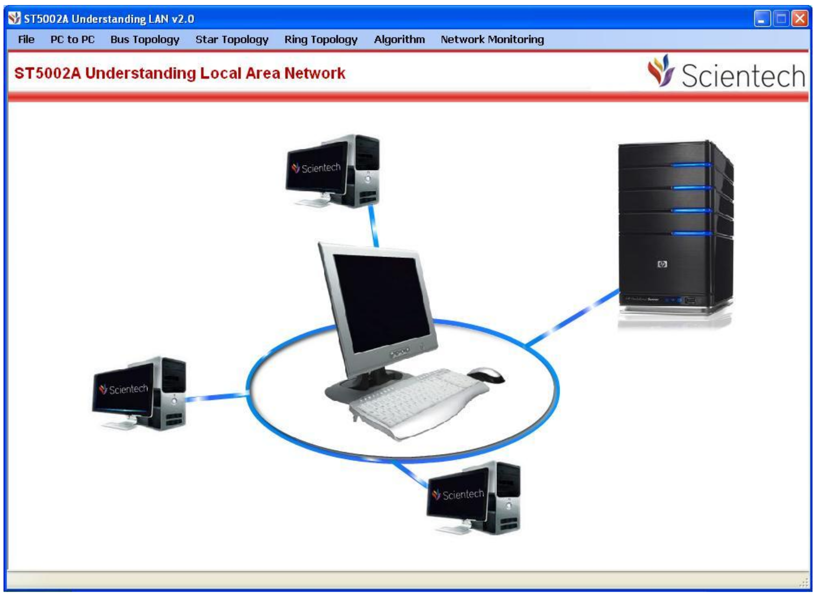
6. Select *NWLink IPX/SPX/NetBIOS Compatible Transport Protocol*, and then click *OK*.
7. Wait a moment while Windows XP adds the protocol to the network components already bound to your NIC. Your network adapter Properties dialog box appears, now with the NWLink NetBIOS and the NWLink IPX/SPX/ NetBIOS Compatible Transport protocols listed under the "This connection uses the following items:" heading.

8. Click *Close* to save your changes, and then close the Network Connections window. On a Windows XP workstation, you can install any other protocol in the same manner as you installed the NWLink protocol.

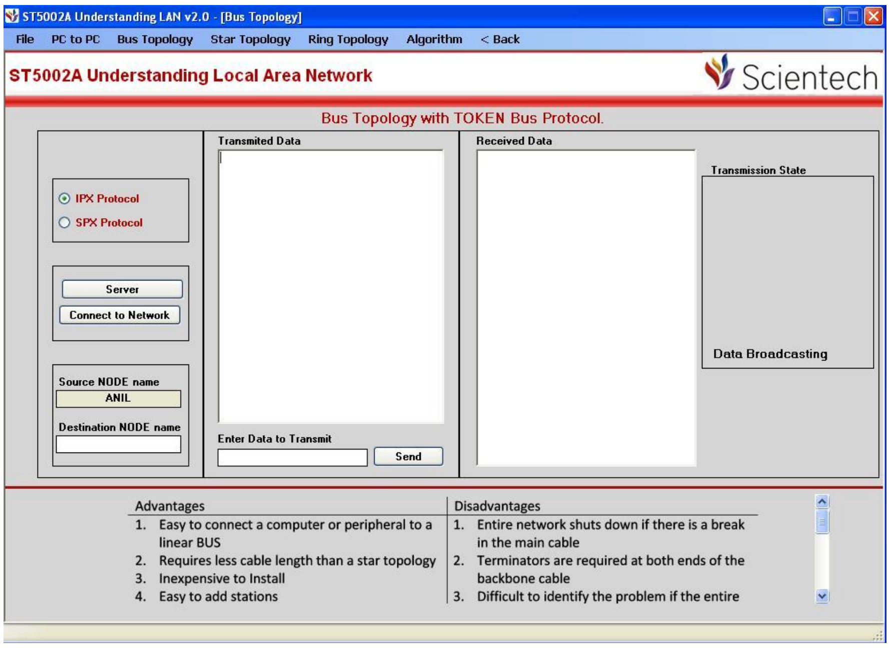
It is possible to bind multiple protocols to the same network adapter. In fact, this is necessary on networks that use more than one type of protocol. In addition, a workstation may have multiple NICs, in which case several different protocols might be bound to each NIC. What's more, the same protocol may be configured differently on different NICs. For example, let's say you managed a NetWare server that contained two NICs and provided both TCP/IP and IPX/SPX communications to many clients. Using the network operating system's protocol configuration utility, you would need to configure TCP/IP separately for each NIC. Similarly, you would need to configure IPX/SPX separately for each NIC. If you did not configure the protocols for each NIC separately, clients would not know which NIC to address when sending and receiving information to and from the server.

Procedure:

1. Connect nodes to bus topology section on **ST5002A**.
2. Connect End Terminators on **ST5002A** in Bus topology section.
3. Please check whether the SPX/IPX Protocol is installed on all the systems before performing experiment. (Refer 'How to Install SPX/IPX protocol)
4. Open **ST5002A** Software.
5. Click on 'Bus Topology'



6. A window will appear on your screen as shown in figure 134 below



ST5002A Understanding Local Area Network

Bus Topology with TOKEN Bus Protocol.

Source NODE name: ANIL

Destination NODE name:

Transmited Data

Enter Data to Transmit

Received Data

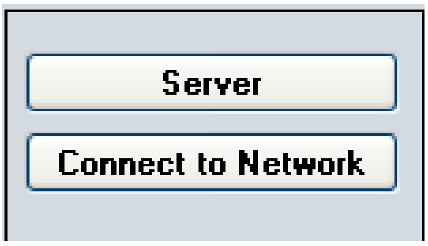
Transmission State

Data Broadcasting

Advantages
1. Easy to connect a computer or peripheral to a linear BUS
2. Requires less cable length than a star topology
3. Inexpensive to Install
4. Easy to add stations

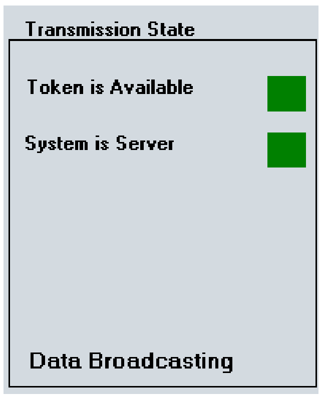
Disadvantages
1. Entire network shuts down if there is a break in the main cable
2. Terminators are required at both ends of the backbone cable
3. Difficult to identify the problem if the entire

7. Make on node as server by pressing ‘Server’ button.



8. On the other nodes you have to connect to server by pressing ‘Connect to Network’

9. You will see transmission status as shown below



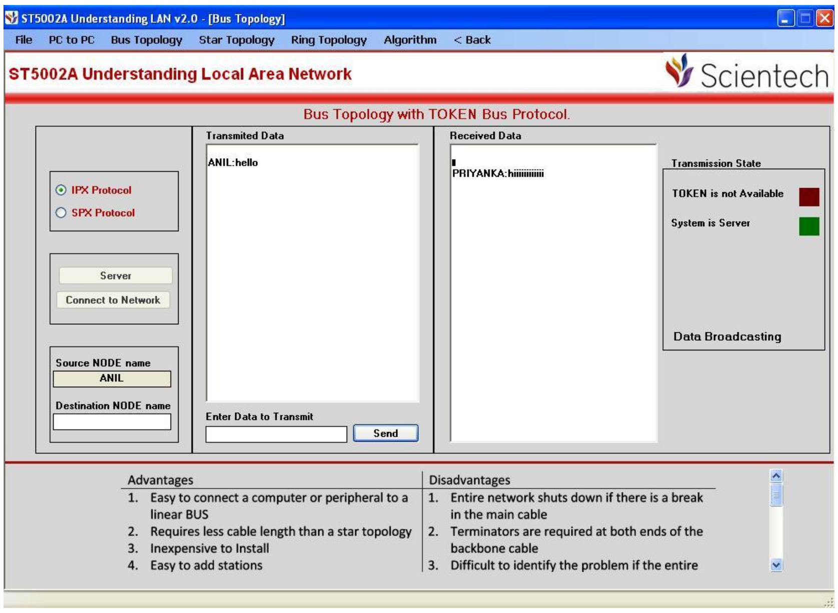
Transmission State

Token is Available 🟩

System is Server 🟩

Data Broadcasting

11. Now you can use both the protocols IPX (The data will be broadcasted to all Nodes) and SPX (The data will be sent to destination node) for transmitting the Data



ST5002A Understanding Local Area Network

Bus Topology with TOKEN Bus Protocol.

[✓ IPX Protocol] SPX Protocol

Source NODE name: ANIL

Destination NODE name:

Transmited Data

ANIL:hello

Enter Data to Transmit

Received Data

PRIYANKA:hiiiiiiiiiiii

Transmission State

TOKEN is not Available 🟥

System is Server 🟩

Data Broadcasting

Advantages	Disadvantages
1. Easy to connect a computer or peripheral to a linear BUS	1. Entire network shuts down if there is a break in the main cable
2. Requires less cable length than a star topology	2. Terminators are required at both ends of the backbone cable
3. Inexpensive to Install	3. Difficult to identify the problem if the entire
4. Easy to add stations	

Conclusion:

Experiment No.11

Objective: Implementation of Ring topology using DB9

Ring network layout ring network is a topology of computer networks where each node is connected to two other nodes, so as to create a ring.

Ring networks tend to be inefficient when compared to Star networks because data must travel through less number of points before reaching its destination. For example, if a given ring network has eight computers on it, to get from computer one to computer four, data must travel from computer one, through computers two and three, and to its destination at computer four. It could also go from computer one through eight, seven, six, and five until reaching four, but this method is slower because it travels through more computers. Ring networks also carry the disadvantage that if one of the nodes in the network breaks down then the entire network will break down with it as it requires a full circle in order to function.

Advantages:

Data is quickly transferred without a 'bottle neck'. (Very fast, all data traffic is in the same direction)

The transmission of data is relatively simple as packets travel in one direction only.

Adding additional nodes has very little impact on bandwidth

It prevents network collisions because of the media access method or architecture required.

Disadvantages:

Data packets must pass through every computer between the sender and recipient therefore this makes it slower.

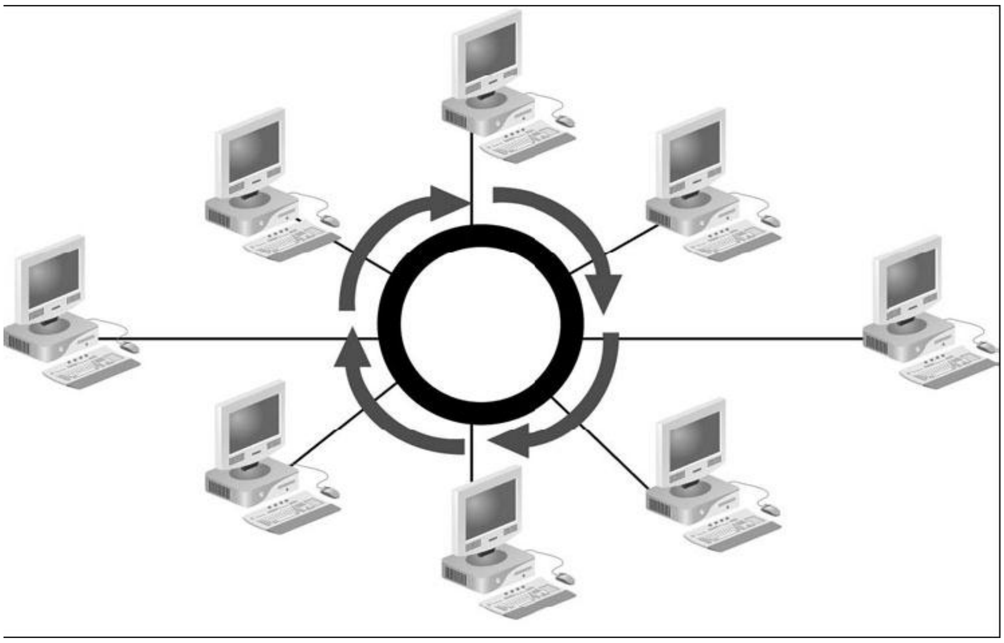
If any of the nodes fail then the ring is broken and data cannot be transmitted successfully.

It is difficult to troubleshoot the ring.

Because all stations are wired together, to add a station you must shut down the network temporarily.

In order for all computers to communicate with each other, all computers must be turned on.

Total dependence upon the one cable



Conclusion:

Experiment No.12

Objective: To create the scenario and study the performance of token ring protocols through simulation.

Equipments Needed:

ST5002A

ST5002A software

4- DB9 Cables

4- 2mm patch cords

Theory:

Now that you have learnt about the many forms of Ethernet, you are ready to learn about Token Ring, a less common, but still important network access method. Token Ring is a network technology first developed by IBM in the 1980s. In the early 1990s, the Token Ring architecture competed strongly with Ethernet to be the most popular access method. Since that time, the economics, speed, and reliability of Ethernet have improved, leaving Token Ring behind. Because IBM developed Token Ring, a few IBM-centric IT Departments continue to use it. Other network managers have changed their former Token Ring networks into Ethernet networks.

Token Ring networks have traditionally been more expensive to implement than Ethernet networks. Proponents of the Token Ring technology argue that, although some of its connectivity hardware is more expensive, its reliability results in less downtime and lower network management costs than Ethernet. On a practical level, Token Ring has probably lost the battle for superiority because its developers were slower to develop high-speed standards. Token Ring networks can run at either 4, 16, or 100 Mbps. The 100-Mbps Token Ring standard, finalized in 1999, is known as *HSTR (High-Speed Token Ring).* HSTR can use either twisted-pair or fiber-optic cable as its transmission medium. Although it is as reliable and efficient, it is still less common than Ethernet because of its higher cost and lagging speed.

Token Ring networks use the token-passing routine and a star-ring hybrid physical topology. In *token passing*, a 3-byte packet, called a token, is transmitted from one node to another in a circular fashion around the ring. When a station has something to send, it picks up the token, changes it to a frame, and then adds the header, information, and trailer fields. The header includes the address of the destination node. All nodes read the frame as it traverses the ring to determine whether they are the intended recipient of the message. If they are, they pick up the data, and then retransmit the frame to the next station on the ring. When the frame finally reaches the originating station, the originating workstation reissues a free token that can then be used by another station. The token-passing control scheme avoids the possibility for collisions. This fact makes Token Ring more reliable and efficient than Ethernet. It also does not impose distance limitations on the length of a LAN segment, unlike CSMA/CD.

On a Token Ring network, one workstation, called the active monitor, acts as the controller for token passing. Specifically, the *active monitor* maintains the timing for ring passing, monitors token and frame transmission, detects lost tokens, and corrects errors when a timing error or other disruption occurs. Only one workstation on the ring can act as the active monitor at any given time.

Token Ring Operation:

Token Ring and IEEE 802.5 are two principal examples of token-passing networks (FDDI is the other). *Token-passing networks* move a small frame, called a token, around the network. Possession of the token grants the right to transmit. If a node receiving the token has no information to send, it passes the token to the next end station. Each station can hold the token for a maximum period of time.

If a station possessing the token does have information to transmit, it seizes the token, alters 1 bit of the token (which turns the token into a start-of-frame sequence), appends the information that it wants to transmit, and sends this information to the next station on the ring. While the information frame is circling the ring, no token is on the network (unless the ring supports early token release), which means that other stations wanting to transmit must wait. Therefore, collisions cannot occur in Token Ring networks. If early token release is supported, a new token can be released when frame transmission is complete.

The information frame circulates the ring until it reaches the intended destination station, which copies the information for further processing. The information frame continues to circle the ring and is finally removed when it reaches the sending station. The sending station can check the returning frame to see whether the frame was seen and subsequently copied by the destination.

Unlike CSMA/CD networks (such as Ethernet), token-passing networks are *deterministic*, which means that it is possible to calculate the maximum time that will pass before any end station will be capable of transmitting. This feature and several reliability features, which are discussed in the section "Fault-Management Mechanisms," later in this chapter, make Token Ring networks ideal for applications in which delay must be predictable and robust network operation is important. Factory automation environments are examples of such applications.

Priority System:

Token Ring networks use a sophisticated priority system that permits certain user-designated, high-priority stations to use the network more frequently. Token Ring frames have two fields that control priority: the priority field and the reservation field.

Only stations with a priority equal to or higher than the priority value contained in a token can seize that token. After the token is seized and changed to an information frame, only stations with a priority value higher than that of the transmitting station can reserve the token for the next pass around the network. When the next token is generated, it includes the higher priority of the reserving station. Stations that raise a token's priority level must reinstate the previous priority after their transmission is complete.

Fault-Management Mechanisms:

Token Ring networks employ several mechanisms for detecting and compensating for network faults. For example, one station in the Token Ring network is selected to be the *active monitor*. This station, which potentially can be any station on the network, acts as a centralized source of timing information for other ring stations and performs a variety of ring-maintenance functions. One of these functions is the removal of continuously circulating frames from the ring. When a sending device fails, its frame may continue to circle the ring. This can prevent other stations from transmitting their own frames and essentially can lock up the network. The active monitor can detect such frames, remove them from the ring, and generate a new token.

The IBM Token Ring network's star topology also contributes to overall network reliability. Because all information in a Token Ring network is seen by active MSAUs, these devices can be programmed to check for problems and selectively remove stations from the ring, if necessary.

A Token Ring algorithm called *beaconing* detects and tries to repair certain network faults. Whenever a station detects a serious problem with the network (such as a cable break), it sends a beacon frame, which defines a failure domain. This domain includes the station reporting the failure, its nearest active upstream neighbour (NAUN), and everything in between. Beaconing initiates a process called *auto reconfiguration*, in which nodes within the failure domain automatically perform diagnostics in an attempt to reconfigure the network around the failed areas. Physically, the MSAU can accomplish this through electrical reconfiguration.

Frame Format:

Token Ring and IEEE 802.5 support two basic frame types: tokens and data/command frames. Tokens are 3 bytes in length and consist of a start delimiter, an access control byte, and an end delimiter. Data/command frames vary in size, depending on the size of the Information field. Data frames carry information for upper-layer protocols, while command frames contain control information and have no data for upper-layer protocols. Both formats are shown in Fig.1.

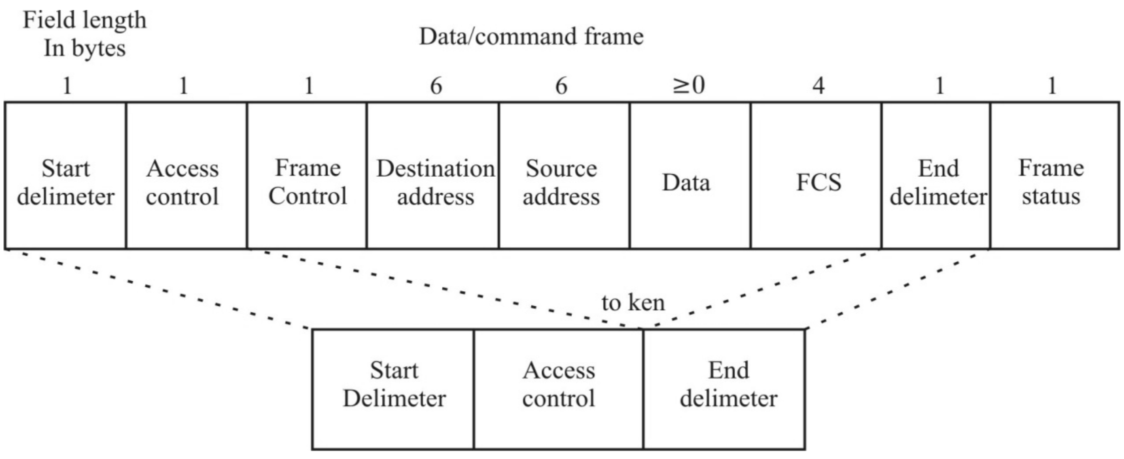


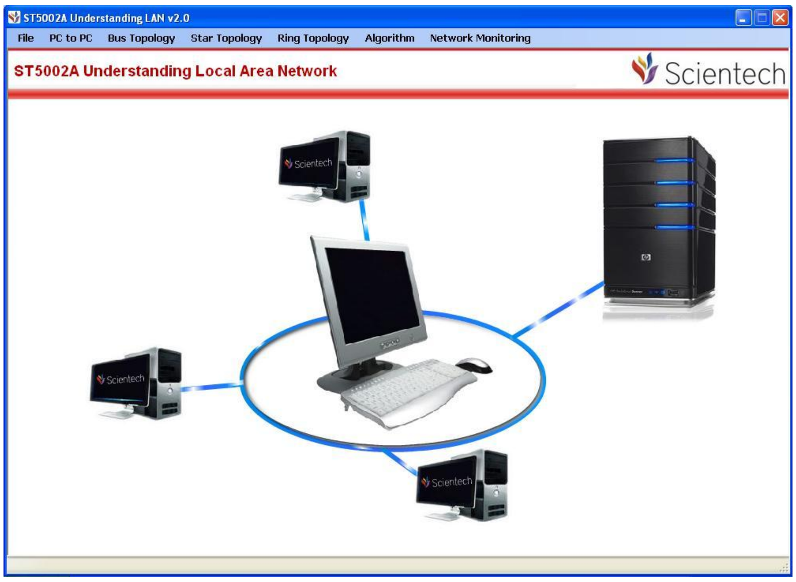
Fig.1. Frame Format for Token Ring

Token Ring technology was developed in the 1970s by IBM. Token-passing networks move a small frame, called a token, around the network. Possession of the token grants the right to transmit. If a node receiving the token has no information to send, it passes the token to the next end station. Each station can hold the token for a maximum period of time.

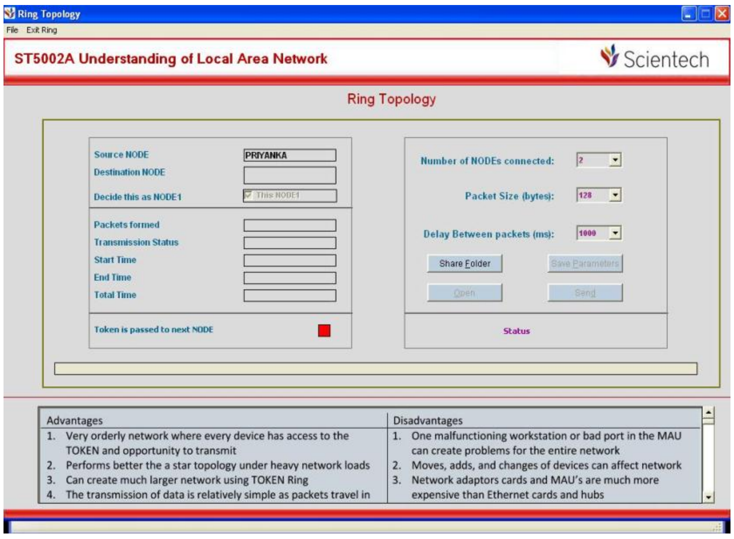
If a station possessing the token does have information to transmit, it seizes the token, alters 1 bit of the token (which turns the token into a start-of-frame sequence), appends the information that it wants to transmit, and sends this information to the next station on the ring.

Procedure:

1. Connect DB9 Cables to ST5002A as well as each computer node connected to the ST5002A
2. Connect patch cords to 'Ring In' & 'Ring Out' terminals on the ST5002A to form a RING
3. Run the **ST5002A** software on each computer connected to the **ST5002A** & follow the steps
4. Click on "Ring Topology"



5. Select a node to treat him as Node1 by checking the 'Decide this as NODE1' checkbox, automatically other nodes as decided as NODE2, 3, & 4 according to connection



Ring Topology

File Exit Ring

ST5002A Understanding of Local Area Network

Scientech

Ring Topology

Source NODE: PRIYANKA
Destination NODE:
Decide this as NODE1: ✓ This NODE1

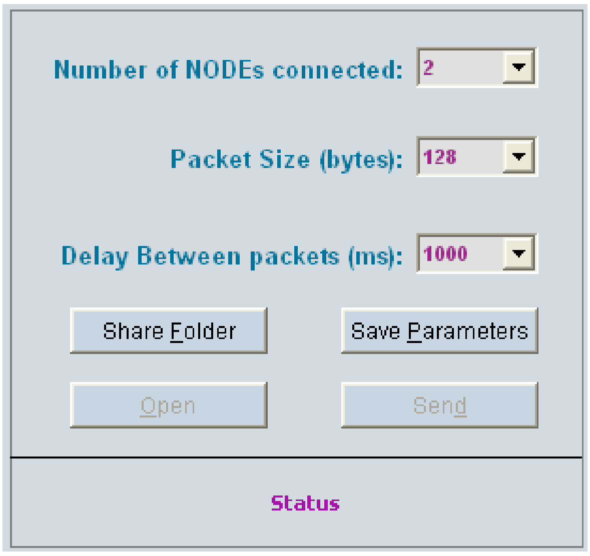
Packets formed:
Transmission Status:
Start Time:
End Time:
Total Time:

Token is passed to next NODE 🟥

Number of NODEs connected: 2
Packet Size (bytes): 128
Delay Between packets (ms): 1000

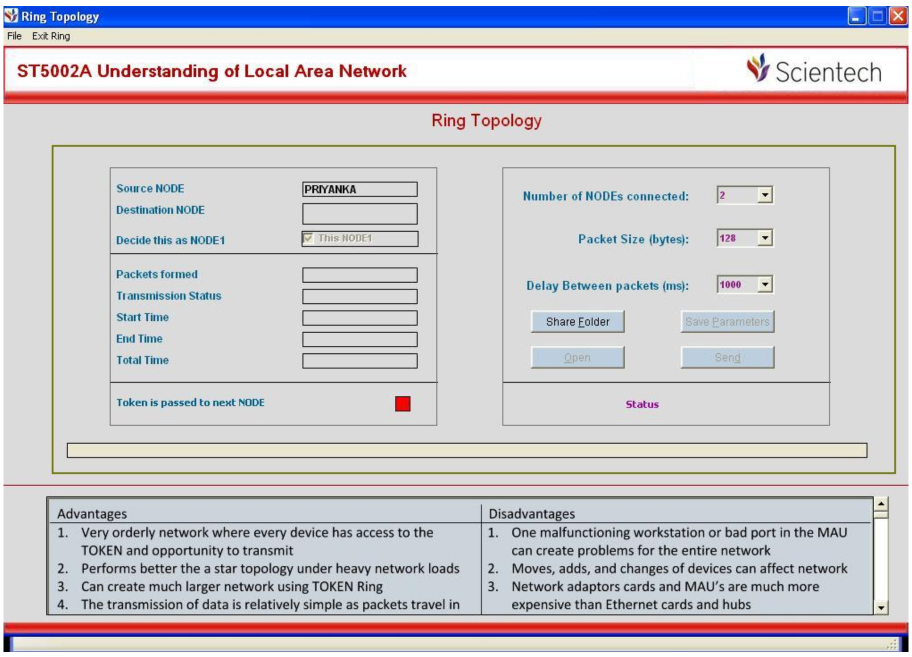
Advantages	Disadvantages
1. Very orderly network where every device has access to the TOKEN and opportunity to transmit	1. One malfunctioning workstation or bad port in the MAU can create problems for the entire network
2. Performs better the a star topology under heavy network loads	2. Moves, adds, and changes of devices can affect network
3. Can create much larger network using TOKEN Ring	3. Network adaptors cards and MAU's are much more expensive than Ethernet cards and hubs
4. The transmission of data is relatively simple as packets travel in	

6. The Node which have Token is having authority to transmit the file
7. Before Transmission click "Sharing Folder" to share the folder for RING topology



Number of NODEs connected: 2
Packet Size (bytes): 128
Delay Between packets (ms): 1000

8. For transmission select all the parameters & then browse the file for transmission
9. Click "Send" to transmit the file
10. Frame format for Ring topology according to IEEE standards is shown at the bottom side of the window
11. Screen shot is as shown as shown below.



Source NODE: PRIYANKA
Destination NODE:
Decide this as NODE1: ✓ This NODE1
Packets formed:
Transmission Status:
Start Time:
End Time:
Total Time:
Token is passed to next NODE 🟥

Number of NODEs connected: 2
Packet Size (bytes): 128
Delay Between packets (ms): 1000

Advantages	Disadvantages
1. Very orderly network where every device has access to the TOKEN and opportunity to transmit	1. One malfunctioning workstation or bad port in the MAU can create problems for the entire network
2. Performs better the a star topology under heavy network loads	2. Moves, adds, and changes of devices can affect network
3. Can create much larger network using TOKEN Ring	3. Network adaptors cards and MAU's are much more expensive than Ethernet cards and hubs
4. The transmission of data is relatively simple as packets travel in	

Conclusion:

Experiment No:13

Aim: Implementation of Distance Vector Routing and Link state routing/Dijkistra's algorithm.

1. Distance Vector Routing

Equipments Needed:

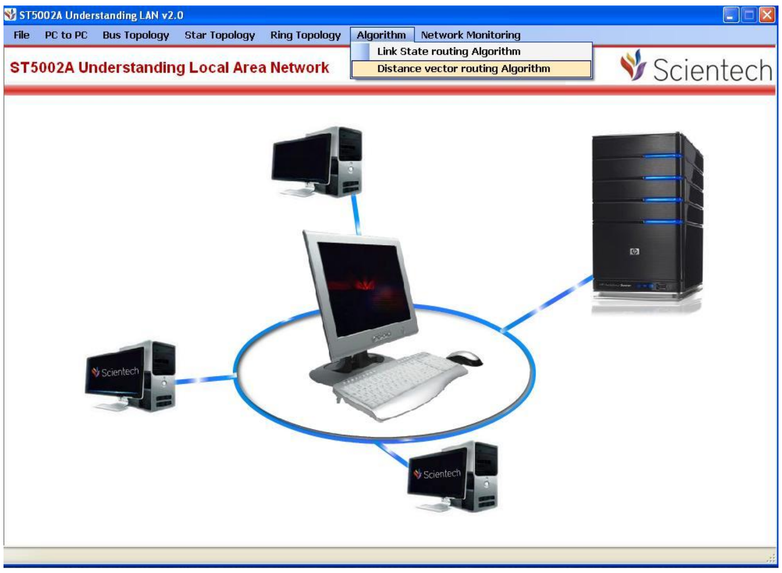
ST5002A

ST5002A Software

4- CAT5 cables

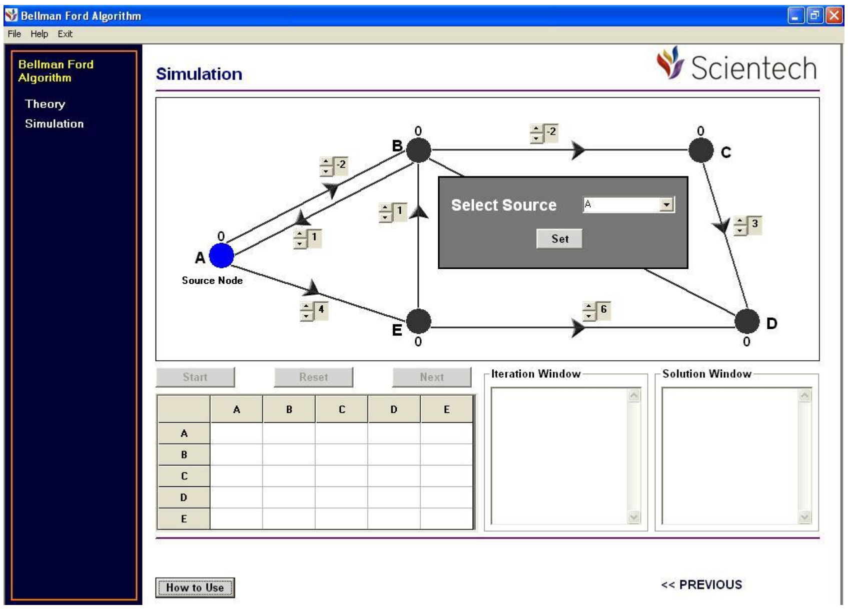
Procedure:

1. Connect the computers to the ST5002A using RJ45 cables
2. Switch on the power supply of ST5002A.
3. Open **ST5002A** software
4. Go to Algorithm menu from top and select Distance vector Routing algorithm


ST5002A Understanding LAN v2.0
File
PC to PC
Bus Topology
Star Topology
Ring Topology
Algorithm
Network Monitoring
Link State routing Algorithm
Distance vector routing Algorithm
ST5002A Understanding Local Area Network
Scientech

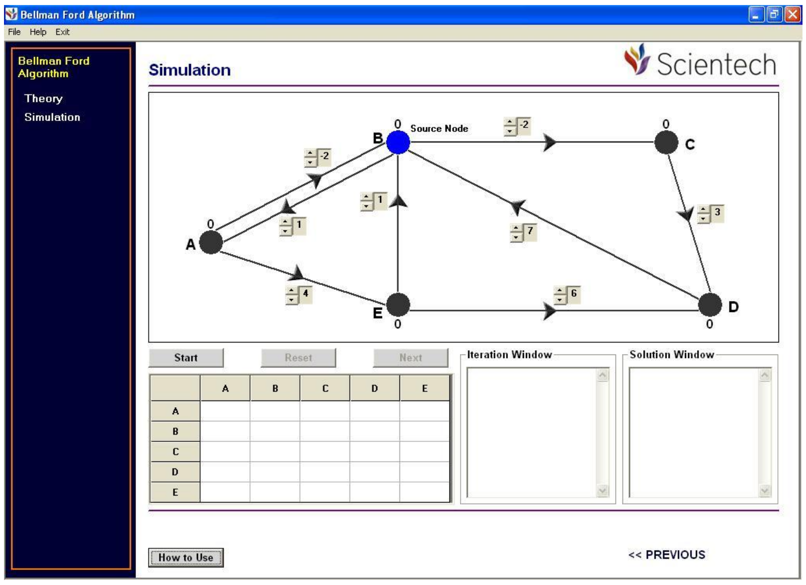
5. A window will appear

6. Now select any node from drop down list and press set button



	A	B	C	D	E
A					
B					
C					
D					
E					

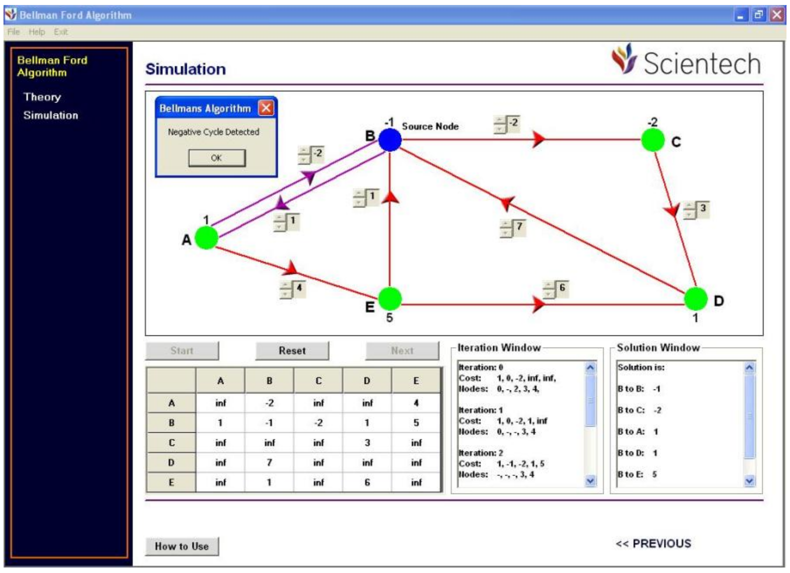
7. Now observe how it detects shortest path from A to selected node.



	A	B	C	D	E
A					
B					
C					
D					
E					

9. Now click on start button it shows it detect path

8. Note press Next button it will display iteration and solution

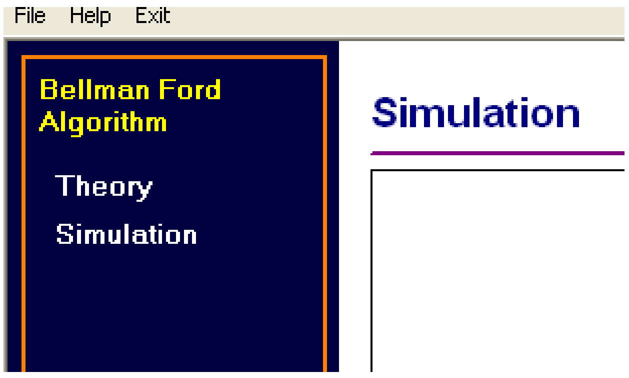


	A	B	C	D	E
A	inf	-2	inf	inf	4
B	1	-1	-2	1	5
C	inf	inf	inf	3	inf
D	inf	7	inf	inf	inf
E	inf	1	inf	6	inf

```
Iteration Window
Iteration: 0
Cost:     1, 0, -2, inf, inf,
Nodes:    0, -, 2, 3, 4,

Iteration: 1
Cost:     1, 0, -2, 1, inf
Nodes:    0, -, -, 3, 4

Iteration: 2
Cost:     1, -1, -2, 1, 5
Nodes:    -, -, -, 3, 4
```

```
Solution Window
Solution is:

B to B:   -1

B to C:   -2

B to A:   1

B to D:   1

B to E:   5
```

9. And press continuously and observe the changes it will display all possibility to reach on selected Node
10. And finally it give message Negative cycle detected conform that all possibilities are checked
11. User can also take help from side panel like how it works, how to use



2. Link state routing/ Dijkstra's algorithm

Equipments Needed:

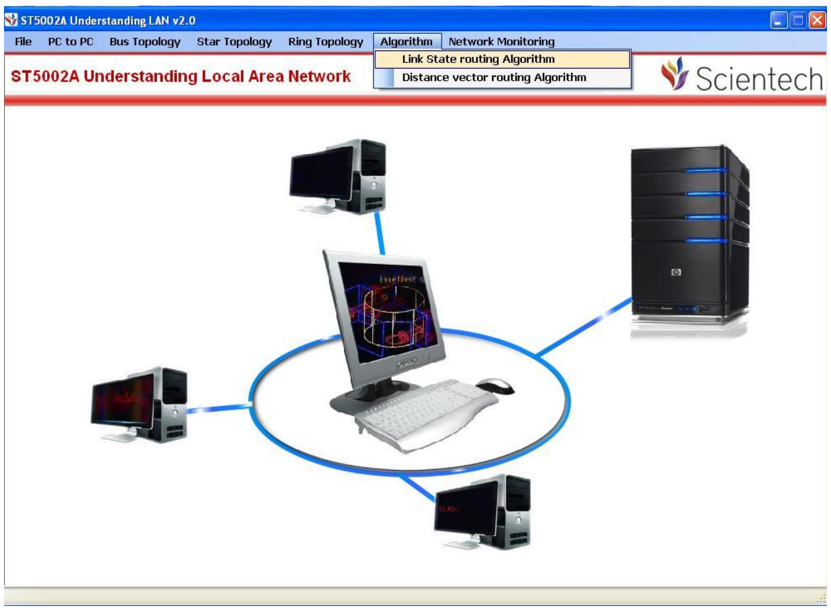
ST5002A

ST5002A Software

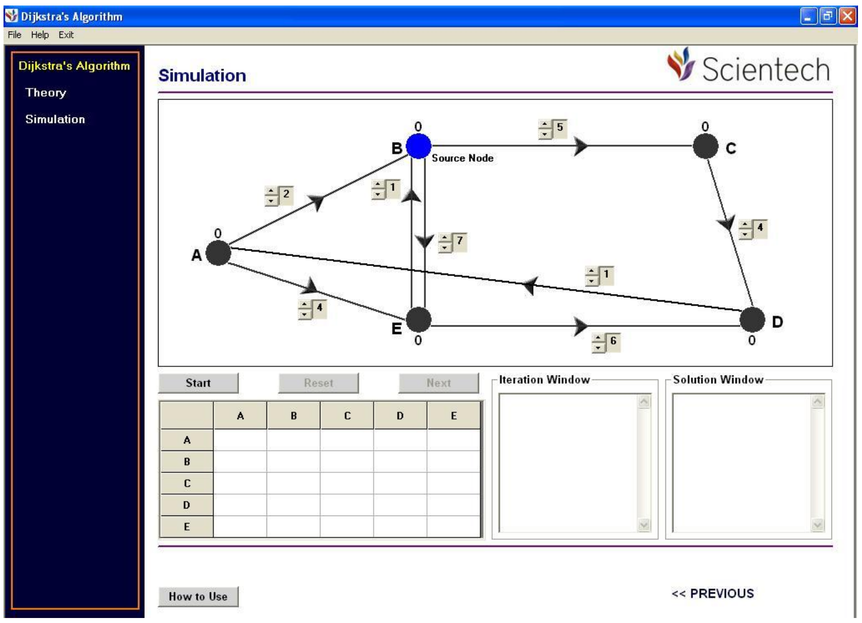
4- CAT5 cables

Procedure:

1. Connect the computers to the ST5002A using RJ45 cables
2. Switch on the power supply of ST5002A.
3. Open **ST5002A** software
5. Go to Algorithm menu from top and select Dijkstra’s algorithm.

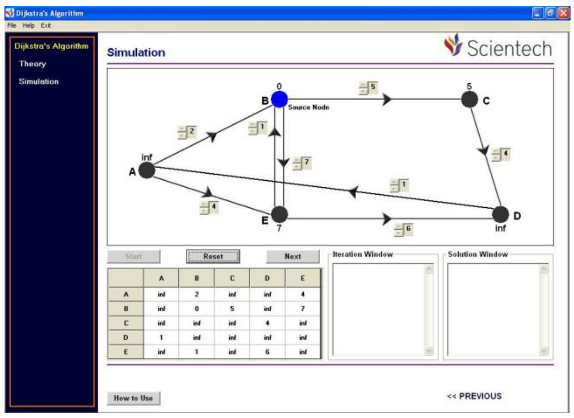

ST5002A Understanding LAN v2.0
File
PC to PC
Bus Topology
Star Topology
Ring Topology
Algorithm
Network Monitoring
Link State routing Algorithm
Distance vector routing Algorithm
ST5002A Understanding Local Area Network
Scientech

6. Now window will appear as shown



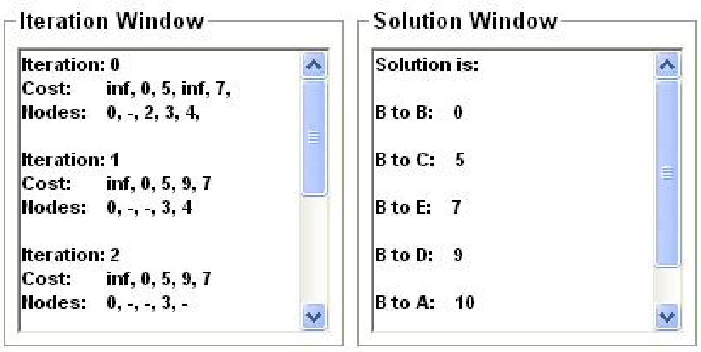
	A	B	C	D	E
A					
B					
C					
D					
E					

7. Now select source node we select A and press set window appear as shown



	A	B	C	D	E
A	inf	2	inf	inf	4
B	inf	0	5	inf	7
C	inf	inf	inf	4	inf
D	1	inf	inf	inf	inf
E	inf	1	inf	6	inf

8. Now press start it will show all possibilities


Iteration Window

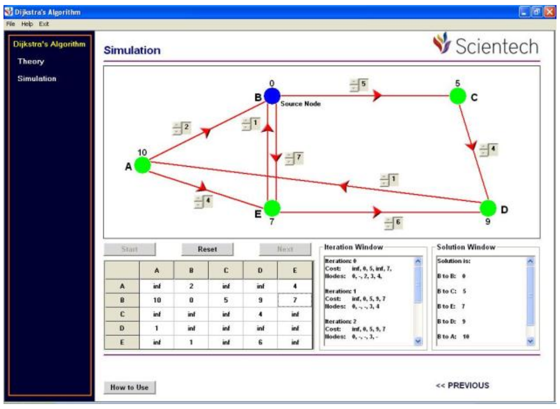
```
Iteration: 0
Cost:     inf, 0, 5, inf, 7,
Nodes:    0, -, 2, 3, 4,

Iteration: 1
Cost:     inf, 0, 5, 9, 7
Nodes:    0, -, -, 3, 4

Iteration: 2
Cost:     inf, 0, 5, 9, 7
Nodes:    0, -, -, 3, -
```

Solution Window

```
Solution is:

B to B:   0

B to C:   5

B to E:   7

B to D:   9

B to A:   10
```

9. Now press next it will show how to reach on nodes and press Next continuously it will

show all iterations



	A	B	C	D	E
A	inf	2	inf	inf	4
B	10	0	5	9	7
C	inf	inf	inf	4	inf
D	1	inf	inf	inf	inf
E	inf	1	inf	6	inf

Iteration Window

```
Iteration: 0
Cost:     inf, 0, 5, inf, 7,
Nodes:    0, -, 2, 3, 4,

Iteration: 1
Cost:     inf, 0, 5, 9, 7
Nodes:    0, -, -, 3, 4

Iteration: 2
Cost:     inf, 0, 5, 9, 7
Nodes:    0, -, -, 3, -
```

Solution Window

```
Solution is:

B to B:  0

B to C:  5

B to E:  7

B to D:  9

B to A:  10
```

10. User can also take help from side panel like how it works, how to use

Conclusion:

Experiment No.14

Objective: Implementation of Data Encryption and Decryption

Equipments needed:

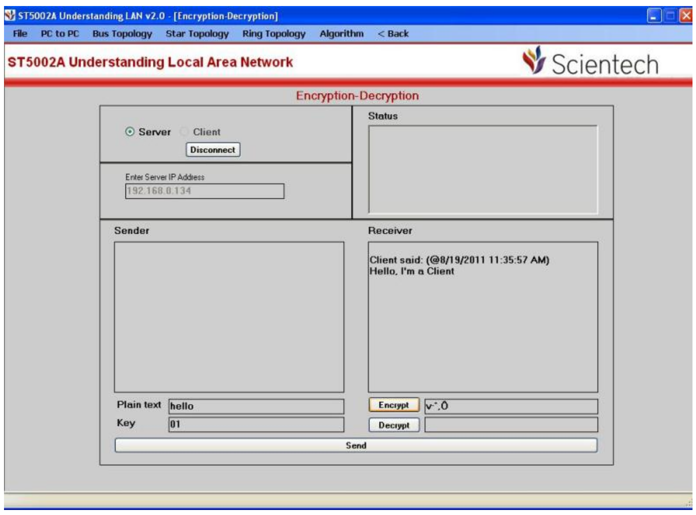
ST5002A

ST5002A Software

4- CAT5 cables

Procedure:

1. Connect the computers to the ST5002A using RJ45 cables
2. Switch on the power supply of ST5002A.
3. Open **ST5002A** software
4. Go to Network Monitoring on top menu of software



Enter Server IP Address: 192.168.0.134

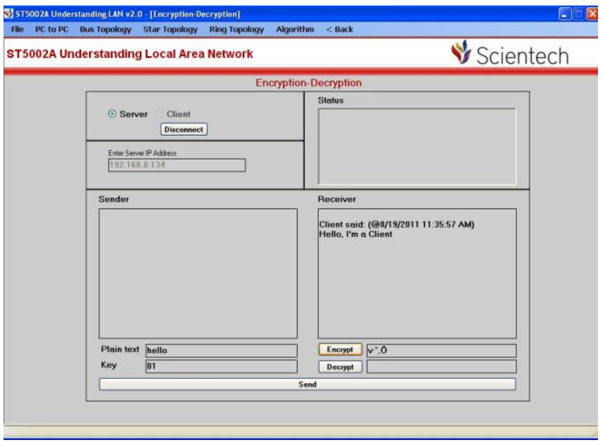
Client said: (@8/19/2011 11:35:57 AM)
Hello, I'm a Client

Plain text: hello

Key: 01

Encrypt: v˜,Ô

5. A window will appear as shown



Enter Server IP Address: 192.168.0.134

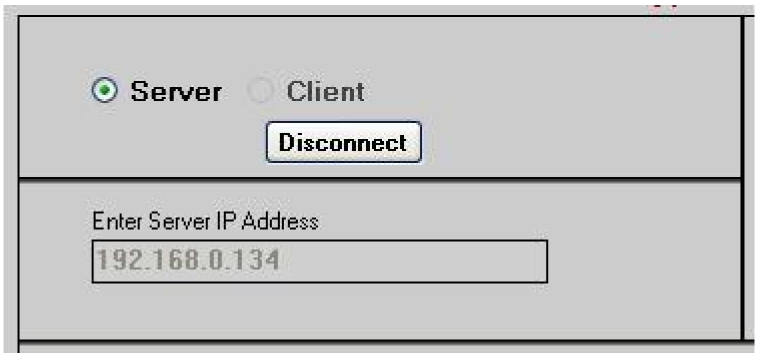
Client said: (@8/19/2011 11:35:57 AM)
Hello, I'm a Client

Plain text: hello

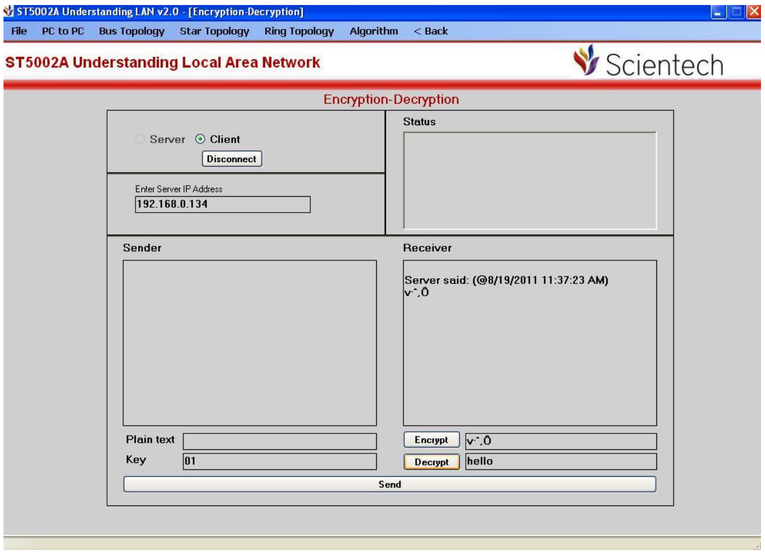
Key: 01

Encrypt: v˜,Ô

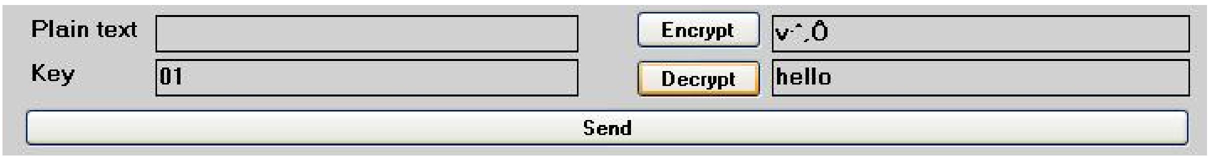
6. Select server at one PC


Server [✓] Client
Disconnect
Enter Server IP Address
192.168.0.134

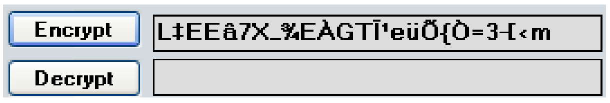
7. And on the client side enter the IP address of Server and click on client radio button as shown


ST5002A Understanding LAN v2.0 - [Encryption-Decryption]
File PC to PC Bus Topology Star Topology Ring Topology Algorithm < Back
ST5002A Understanding Local Area Network
Scientech
Encryption-Decryption
Server [✓ Client]
Disconnect
Enter Server IP Address
192.168.0.134
Status
Sender
Receiver
Server said: (@8/19/2011 11:37:23 AM)
v˙˙,Ô
Plain text
Key 01
Encrypt v˙˙,Ô
Decrypt hello
Send

8. Now on Server side type any plain text enter any string in Key tab which will we used for decrypting data at the receiver side as shown

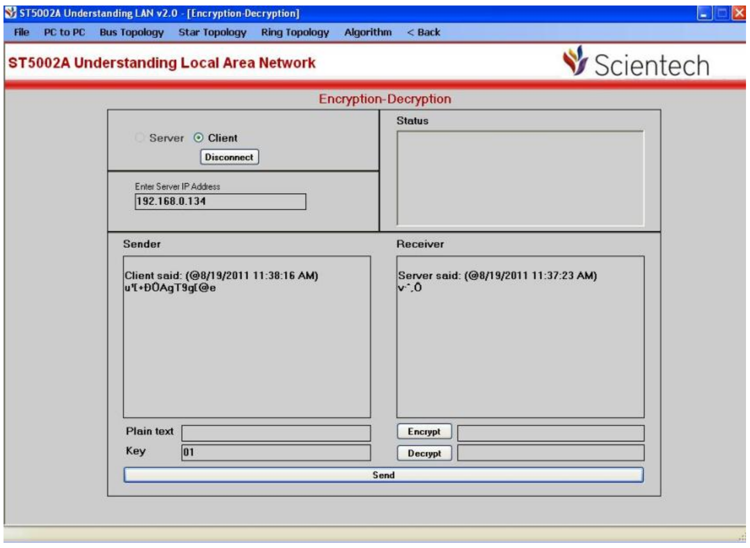

Plain text
Key 01
Encrypt v˙˙,Ô
Decrypt hello
Send

9. Now click on Encrypt button on Server side window will appear as shown



L‡EEâ7X_%EÀGTĪ'eüÕ{Ò=3-[‹m

10. It will show some encrypted data now press send button and observe on receiver side



Server [✓ Client]

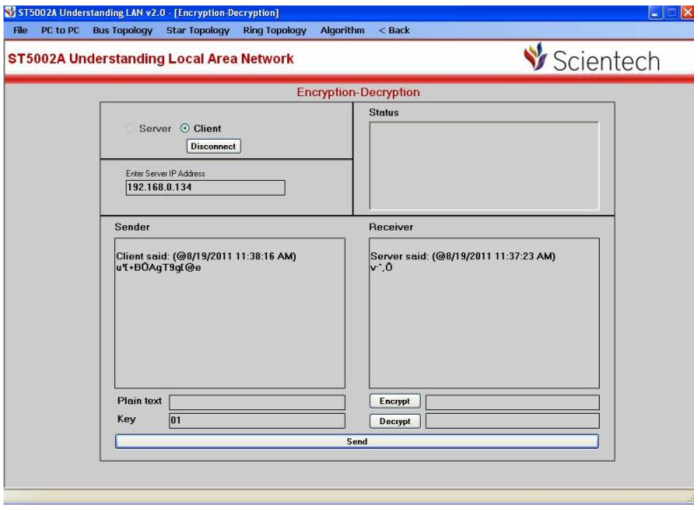
Enter Server IP Address: 192.168.0.134

Client said: (@8/19/2011 11:38:16 AM)
u'[+ĐÖAgT9g[@e

Server said: (@8/19/2011 11:37:23 AM)
v·ˆ,Ö

Plain text:
Key: 01

11. Now enter key inside key tab and press decrypt



Server [✓ Client]

Enter Server IP Address: 192.168.0.134

Client said: (@8/19/2011 11:38:16 AM)
u'[+ĐÖAgT9g[@e

Server said: (@8/19/2011 11:37:23 AM)
v·ˆ,Ö

Plain text:
Key: 01

12. And receiver can see the data which is send by sender by inputing correct key.

Conclusion:

Experiment No.15

Objective: Study of Socket programming

Theory:

What is Windows Sockets:

The Windows Sockets specification defines a network programming interface for Microsoft Windows which is based on the "socket". paradigm popularized in the Berkeley Software Distribution (BSD) from the University of California at Berkeley.

It encompasses both familiar Berkeley socket style routines and a set of Windows specific extensions designed to allow the programmer to take advantage of the message-driven nature of Windows.

The Windows Sockets Specification is intended to provide a single API to which application developers can program and multiple network software vendors can conform. Furthermore, in the context of a particular version of Microsoft Windows, it defines a binary interface (ABI) such that an application written to the Windows

Sockets API can work with a conformant protocol implementation from any network software vendor. This specification thus defines the library calls and associated semantics to which an application developer can program and which a network software vendor can implement.

Network software which conforms to this Windows Sockets specification will be considered "Windows Sockets Compliant". Suppliers of interfaces which are "Windows Sockets Compliant" shall be referred to as "Windows Sockets Suppliers".

To be Windows Sockets Compliant, a vendor must implement 100% of this Windows Sockets specification.

Applications which are capable of operating with any "Windows Sockets Compliant" protocol implementation will be considered as having a "Windows Sockets Interface" and will be referred to as "Windows Sockets Applications".

This version of the Windows Sockets specification defines and documents the use of the API in conjunction with the Internet Protocol Suite (IPS, generally referred to as TCP/IP). Specifically, all Windows Sockets implementations support both stream (TCP) and datagram (UDP) sockets.

While the use of this API with alternative protocol stacks is not precluded (and is expected to be the subject of future revisions of the specification), such usage is beyond the scope of this version of the specification.

Basic concepts:

The basic building block for communication is the socket. A socket is an endpoint of communication to which a name may be bound. Each socket in use has a type and an associated process. Sockets exist within communication domains. A communication domain is an abstraction introduced to bundle common properties of threads communicating through sockets. Sockets normally exchange data only with sockets in the same domain (it may be possible to cross domain boundaries, but only if some translation process is performed). The Windows Sockets facilities support a single communication domain: the Internet domain, which is used by processes which communicate using the Internet Protocol Suite. (Future versions of this specification may include additional domains.)

Sockets are typed according to the communication properties visible to a user.

Applications are presumed to communicate only between sockets of the same type, although there is nothing that prevents communication between sockets of different types should the underlying communication protocols support this.

Two types of sockets currently are available to a user. A stream socket provides for the bi-directional, reliable, sequenced, and unduplicated flow of data without record boundaries.

A datagram socket supports bi-directional flow of data which is not promised to be sequenced, reliable, or unduplicated. That is, a process receiving messages on a datagram socket may find messages duplicated, and, possibly, in an order different from the order in which it was sent. An important characteristic of a datagram socket is that record boundaries in data are preserved. Datagram sockets closely model the facilities found in many contemporary packet switched networks such as Ethernet.

Client-server model:

The most commonly used paradigm in constructing distributed applications is the client/server model. In this scheme client applications request services from a server application. This implies an asymmetry in establishing communication between the client and server.

The client and server require a well-known set of conventions before service may be rendered (and accepted). This set of conventions comprises a protocol which must be implemented at both ends of a connection. Depending on the situation, the protocol may be symmetric or asymmetric. In a symmetric protocol, either side may play the master or slave roles. In an asymmetric protocol, one side is immutably recognized as the master, with the other as the slave. An example of a symmetric protocol is the TELNET protocol used in the Internet for remote terminal emulation. An example of an asymmetric protocol is the Internet file transfer protocol, FTP. No matter whether the specific protocol used in obtaining a service is symmetric or asymmetric, when accessing a service there is a ``client process" and a ``server process".

A server application normally listens at a well-known address for service requests.

That is, the server process remains dormant until a connection is requested by a client's connection to the server's address. At such a time the server process ``wakes up" and services the client, performing whatever appropriate actions the client requests of it. While connection-based services are the norm, some services are based on the use of datagram sockets.

Out-of-band data:

Note: The following discussion of out-of-band data, also referred to as TCP Urgent data, follows the model used in the Berkeley software distribution. Users and implementers should be aware of the fact that there are at present two conflicting interpretations of RFC 793 (in which the concept is introduced), and that the implementation of out-of-band data in the Berkeley Software Distribution does not conform to the Host Requirements laid down in RFC 1122. To minimize interoperability problems, applications writers are advised not to use out-of-band data unless this is required in order to interoperate with an existing service. Windows

Sockets suppliers are urged to document the out-of-band semantics (BSD or RFC 1122) which their product implements. It is beyond the scope of this specification to mandate a particular set of semantics for out-of-band data handling.

The stream socket abstraction includes the notion of ``out of band" data. Out-of-band data is a logically independent transmission channel associated with each pair of connected stream sockets. Out-of-band data is delivered to the user independently of normal data. The abstraction defines that the out-of-band data facilities must support the reliable delivery of at least one out-of-band message at a time. This message may contain at least one byte of data, and at least one message may be pending delivery to the user at any one time. For communications protocols which support only in-band signaling (i.e. the urgent data is delivered in sequence with the normal data), the system normally extracts the data from the normal data stream and stores it separately.

This allows users to choose between receiving the urgent data in order and receiving it out of sequence without having to buffer all the intervening data. It is possible to ``peek" at out-of-band data.

An application may prefer to process out-of-band data "in-line", as part of the normal data stream. This is achieved by setting the socket option SO_OOBINLINE (see setsockopt ()). In this case, the application may wish to determine whether any of the unread data is "urgent" (the term usually applied to in-line out-of-band data). To facilitate this, the Windows Sockets implementation will maintain a logical "mark" in the data stream to indicate the point at which the out-of-band data was sent. An application can use the SIOCATMARK ioctlsocket () command to determine whether there is any unread data preceding the mark. For example, it might use this to resynchronize with its peer by ensuring that all data up to the mark in the data stream is discarded when appropriate.

The WSAAsyncSelect () routine is particularly well suited to handling notification of the presence of out-of-band-data.

Broadcasting:

By using a datagram socket, it is possible to send broadcast packets on many networks supported by the system. The network itself must support broadcast: the system provides no simulation of broadcast in software. Broadcast messages can place a high load on a network, since they force every host on the network to service them.

Consequently, the ability to send broadcast packets has been limited to sockets which are explicitly marked as allowing broadcasting. Broadcast is typically used for one of two reasons: it is desired to find a resource on a local network without prior knowledge of its address, or important functions such as routing require that information be sent to all accessible neighbours.

The destination address of the message to be broadcast depends on the network(s) on which the message is to be broadcast. The Internet domain supports a shorthand notation for broadcast on the local network, the address INADDR_BROADCAST.

Received broadcast messages contain the senders address and port, as datagram sockets must be bound before use.

Some types of network support the notion of different types of broadcast. For example, the IEEE 802.5 token ring architecture supports the use of link-level broadcast indicators, which control whether broadcasts are forwarded by bridges. The Windows Sockets specification does not provide any mechanism whereby an application can determine the type of underlying network, nor any way to control the semantics of broadcasting.

Byte Ordering:

The Intel byte ordering is like that of the DEC VAX, and therefore differs from the Internet and 68000-type processor byte ordering. Thus care must be taken to ensure correct orientation.

Any reference to IP addresses or port numbers passed to or from a Windows Sockets routine must be in network order. This includes the IP address and port fields of a *struct sockaddr_in* (but not the *sin_family* field).

Consider an application which normally contacts a server on the TCP port corresponding to the "time" service, but which provides a mechanism for the user to specify that an alternative port is to be used. The port number returned by getservbyname () is already in network order, which is the format required constructing an address, so no translation is required. However if the user elects to use a different port, entered as an integer, the application must convert this from host to network order (using the htons () function) before using it to construct an address.

Conversely, if the application wishes to display the number of the port within an address (returned via, e.g., getpeername ()), the port number must be converted from network to host order (using ntohs ()) before it can be displayed.

Since the Intel and Internet byte orders are different, the conversions described above are unavoidable. Application writers are cautioned that they should use the standard conversion functions provided as part of the Windows Sockets API rather than writing their own conversion code, since future implementations of Windows Sockets are likely to run on systems for which the host order is identical to the network byte order. Only applications which use the standard conversion functions are likely to be portable.

Socket Functions:

The Windows Sockets specification includes the following Berkeley-style socket routines:

accept () An incoming connection is acknowledged and associated with an immediately created socket. The original socket is returned to the listening state.

Bind () Assign a local name to an unnamed socket.

Close socket () Remove a socket descriptor from the per-process object reference table. Only blocks if SO_LINGER is set.

connect () Initiate a connection on the specified socket.

getpeername() Retrieve the name of the peer connected to the specified socket descriptor.

getsockname() Retrieve the current name for the specified socket

getsockopt() Retrieve options associated with the specified socket descriptor.

htonl() Convert a 32-bit quantity from host byte order to network byte order.

htons() Convert a 16-bit quantity from host byte order to network byte order.

inet_addr() Converts a character string representing a number in the Internet standard ``." notation to an Internet address value.

inet_ntoa() Converts an Internet address value to an ASCII string in ``." notation i.e.

``a.b.c.d".

ioctlsocket() Provide control for descriptors.

listen() Listen for incoming connections on a specified socket.

ntohl() Convert a 32-bit quantity from network byte order to host byte order.

ntohs() Convert a 16-bit quantity from network byte order to host byte order.

recv()* Receive data from a connected socket.

recvfrom()* Receive data from either a connected or unconnected socket.

select()* Perform synchronous I/O multiplexing.

send()* Send data to a connected socket.

sendto()* Send data to either a connected or unconnected socket.

setsockopt() Store options associated with the specified socket descriptor.

shutdown() Shut down part of a full-duplex connection.

socket() Create an endpoint for communication and return a socket descriptor.

* The routine can block if acting on a blocking socket.

Conclusion: