DEVELOPMENT OF A

TOKEN RING LOCAL AREA NETWORK

A THESIS

Submitted to the Department of Computer Science & Engineering, Bangladesh University of Engineering and Technology, Dhaka, in partial fulfillment of the requirements for the degree

of

MASTER OF SCIENCE IN COMPUTER ENGINEERING

by

MD. ANWARUL HASAN.

August 17, 1988.

Bangladesh University of Engineering and Technology, Dhaka.

Accepted as satisfactory for partial fulfillment of the requirements for the degree of Master of Science in Engineering (Computer).

BOARD OF EXAMINERS

i. -- (signature) 29/8/88

Dr. Syed Mahbubur Rahman
Associate Professor & Head
Department of Computer Science and Engineering, BUET, Dhaka.

Chairman and Supervisor

ii. -- (signature) 29/8/88

Dr. Mohammand Abdur Rashid
Associate Professor
Department of Applied Physics and Electronics,
Dhaka University, Dhaka.

Member (External)

iii. -- (signature) 17/8/88

Mr. Mainuddin Mohammed Kamrul Murshed
Assistant Professor
Department of Computer Science and Engineering, BUET, Dhaka.

Member

CERTIFICATE

This is to certify that the work presented in this thesis is done by me under the complete supervision of Dr. Syed Mahbubur Rahman, Associate Professor & Head, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh. It is also certify that neither this thesis nor any part thereof has been submitted elsewhere for the award of any degree or diploma.

Counter signed by supervisor

Signature of the candidate

(Dr. SYED MAHBUBUR RAHMAN)

(MD. ANWARUL HASAN)

ACKNOWLEDGMENT

The author expresses his sincere gratitude to Dr. Syed Mahbubur Rahman, Associate Professor and Head, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka. This research work was done under his supervision. His guidance and suggestions at all the stages of this research have made possible to complete this thesis. The author acknowledges his profound indebtedness to him.

The author takes this opportunity to express his heart-felt gratitude and thanks to Dr. Mohammad Abdur Rashid, Associate Professor, Department of Applied Physics and Electronics, Dhaka University, Dhaka for his kind cooperation and valuable suggestions for this research.

The author is indebted to Mr. M. M. Kamrul Murshed, Assistant Professor, Department of Computer Science and Engineering, BUET, Dhaka, for his sincere cooperation and providing valuable materials for this research.

The allout support and services rendered by the faculty members and the staff of Microcomputer Laboratory of the Department of Compute Science and Engineering, BUET are also acknowledged with sincere thanks.

ABSTRACT

A computer communication protocol has been set up for data transfer among a group of microcomputers connected in a Token Ring Local Area Network. Using several defined control characters, the network provides some modifications with respect to the existing network protocols. The network provides the facilities of sharing of costly computer resources like harddisks, printers, different softwares which may not be attached to all the computers of the network. One of the main ideas behind the development of the network to use hardware elements as minimum as possible and thereby to reduce the cost of implementation. The characteristics of the control token of the network have been modified with respect to the existing protocols to limit the hardware supports and at the same time to restore the normal speed of processing of local functions of a computer. The implemented software provides communication facilities like interractive communication, file send, file receive and remote printing. Hinds are given at the end to carry out future work in this area which will ultimately expand the network facilities.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Computer Communication and Local Area Network

The exchange of information between computers for the purpose of cooperative action is generally referred to as computer communication.

A simple model of communication is represented in Figure 1.1 by a block diagram.

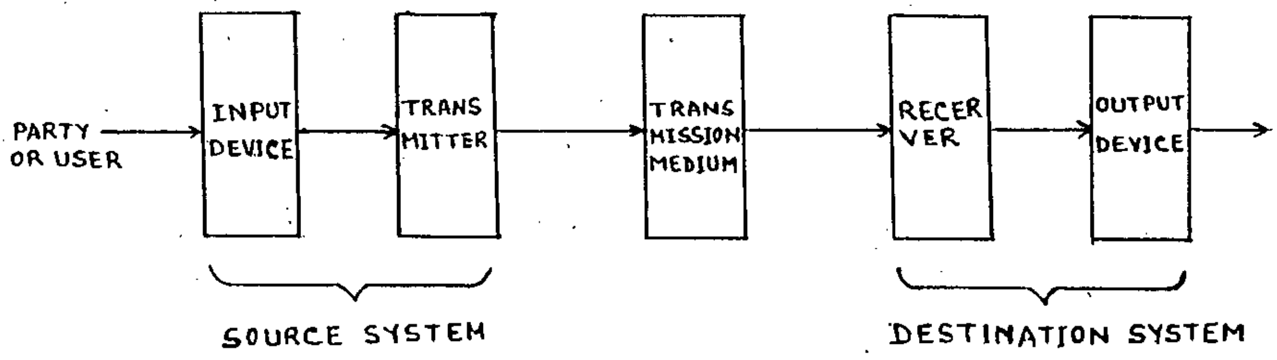


Fig. 1.1 A simple model of communication

In computer communication the source and destination systems may consist of computers, terminals and/or other data processing devices.

In its most simple form, computer communication may take place between two computers that are connected by some form of point-to-point transmission medium. But it is impractical to directly connect all the computers when the number of computers increases and when the computers are very far apart. The solution to this problem is to attach each computer to a communication network. When two or more computers are interconnected in this way via a communication network, the set of computers is referred to as a computer network.

When a group of computers situated within a limited geographical area are interconnected for interchange of information among the computers or other networks a local area network is formed. It can serve variety of devices and supports minis, mainframes, terminals and other peripherals.

A local area network can exploit the distributed computational facilities and thereby reduces the computational cost. The data transfer rates on this network are high enough to satisfy most requirements and provide wide band data transmission capability at low cost.

1.2 Importance of Computer Networking

Interlinking of computers is a very powerful process for information flow from one device to another. User's from different points on the network can easily transfer information. Besides transformation of files and other updated information does not require external processing of the computers .

It is not expected that every user will be able to include all the computer facilities (like printer, hard disk, a large number of costly softwares etc.) with his computer system. When the user is connected to a local area network, resources of different working stations can be shared through this network. This reduces the cost of purchasing different computer resouces for all user's and eliminates the need of physical transformation of devices from one location to another.

A local area network can provide the facilities of sharing database inventory files by two or more users at different geographical locations. This feature reduces the cost of buying softwares for each individual station.

1.3 Objective of the Research Work

A completely separate wide area network for exclusive use of digital data transmission is neither justifiable nor viable in the present context of Bangladesh. But sharing of costly computer resources by establishing a communication link among a group of computers is essential for our country. Thus a suitable low cost technology must be used for computer data communication. Token ring local area network is one of the most popular method to bring the computer resources located within a limited geographical area under a network. As a result, a token ring local area network will be developed by keeping the use of costly hardware elements as minimum as possible.

1.4 Expected Result of the Research Work

The proposed system is expected to be Ring Network. The system will have adaplibility towards extension of the network and the costs will be simply additive on the excess elements involved. The medium access technique will be based on possession of a control token. In the proposed network, a collections of microcomputers will be allowed to intercommunicate through a series of point-to-point coaxial cables with dedicated operations.

The communication process will be interrupt driven. As a result, normal processing at either end will not be obstructed. An user will get interractive communication, file transfer, remote printing and other network facilities from the proposed system.

CHAPTER 2

AN OVERVIEW OF LOCAL AREA NETWORK

2.1 Introduction

The user of a lone computer can access only the resources associated with his computer. The user's scope of exchange of computer information for the purpose of cooperation is restricted unless a computer communication is established by connecting all those computers. Thus the exchange of information among computers is referred to as computer communication and the system on to which all the computers are connected to establish a computer communication is referred to as Computer Network. It is obvious that the interconnection is affected by the network system that allows the computers to exchange information to support the user's requirements. The basic function performed by a network is to provide a path by which a user can access services or connect with other users. Intelligent and meaningful communication, however, requires the exchange of information in an orderly manner using a well defined and structured set of rules and conventions. What is communicated, how it is communicated, and when it is communicated must conform to those mutually acceptable rules and conventions. These sets of rules and conventions are called protocols.

2.2 Classification

Computer networks are characterized according to their use of available band width, geographical extension of the network elements, the way in which those elements are attached to the network. Among those, geographical extension

1) Supports data communication among the participating devices located within a span of 2 to 5 kilometers.

2) Supports data transmission at relatively high speed (1 to 20 Megabyte).

3) Provides the flexibility to attach new devices to the network without much difficulties.

4) Provides the flexibility to interconnect with other networks.

5) Should include features that faciliate network maintenance, diagnostics and services.

6) Uses packet switching techniques for communication.

7) Uses twisted pair as an inexpensive transmission medium or uses coaxial cable or optical fiber as an high speed transmission medium.

8) Should be precisely defined in terms of layered protocols.

9) Uses either bus, tree or ring topology.

10) Provides baseband as well as broadband transmission techniques.

of the network elements is the most widely used basis on which computer networks are usually characterized. When an interconnection is made such that the communicating devices dispersed over tens of hundreds of kilometers, it is referred to as Wide Area Network (WAN) or a Long Haul Network (LHN). A network is an MAN (Metropolitan Area Network) if it serves users who are spread over several buildings in close proximity in an urban environment covering tens of kilometers. On the other hand a Local Area Network (LAN) covers atmost a diameter of several kilometers and is typically confined to a single building or a set of closely situated buildings such as a university campus.

Local Area Networks support minis, mainframes, terminals and other peripherals. In many cases these networks can carry voice and graphics information in addition to data. The LAN is probably the best choice when a variety of devices and a mix of traffic types are involved.

2.3 Characteristics of Local Area Network

A Local Area Network is a group of interconnected computers situated within a limited geographical area for the interchange of information among the computers and between remote hosts or other networks by sharing some transmission facilities. The transmission facilities may include the transmission media, network interfaces and communication protocols. Followings are the characteristics of a typical LAN.

2.4 LAN Architecture

Topology :

The manners in which the devices participating in the network are interconnected for information interchange is referred to as topology. How easily a local area network may be implemented depends to a large extend on its topology. The basic topologies used in LANs are

1) Star Topology
2) Ring Topology
3) Bus Topology

Star Topology :

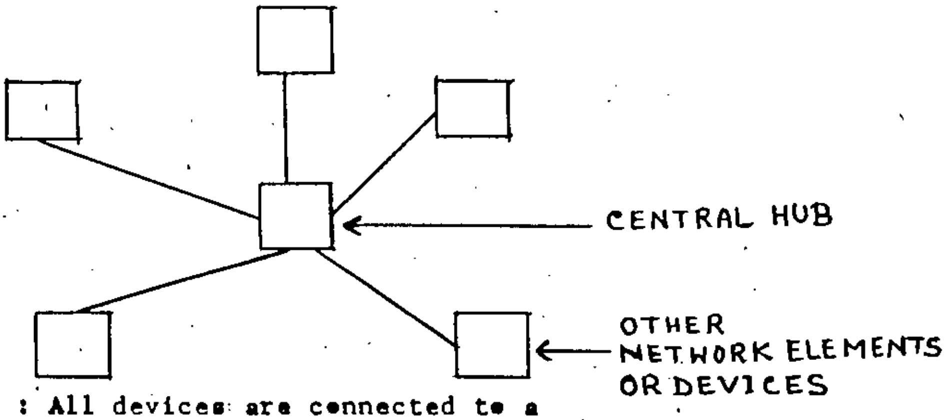


Fig. 2.1 Star Topology : All devices are connected to a central hub

In a star topology there is a central hub which is connected to all the devices by point-to-point links. The central hub contains a switch to be shared by all devices.

Twisted pair is usually used to link the stations to the central switch.

Ring Topology :

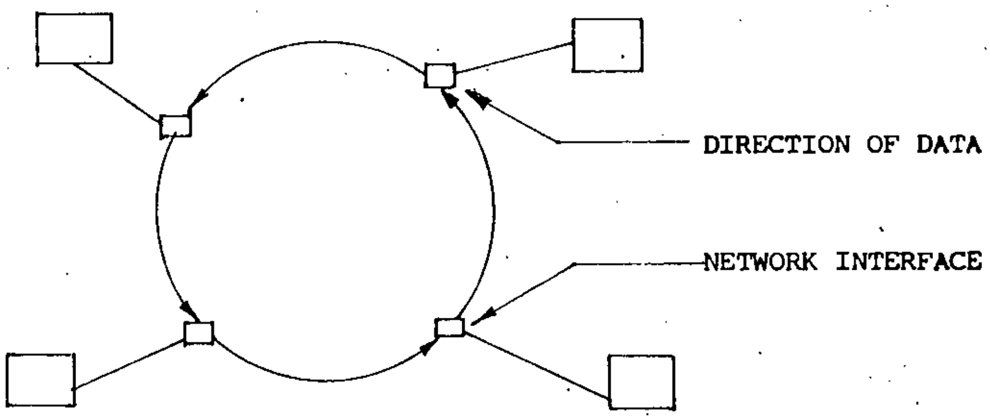


Fig.: 2.1 Ring Topology: Host computers are connected to a ring

The ring topology consists of a closed loop. Data circulates around the ring uniderectionally. On a series of point-to-point links. There is no need of routing decisions. A station wishing to transmit waits for its next turn and then sends data out on to the ring in the form of packet. The packet contain source and destination address fields as well as data. As the packet circulates, the destination station copies the data into a local buffer. A distributed control protocol is used to determine the sequence in which communication continues. Since multiple devices share the ring, control is needed to determine at what time which device can insert packets.

Bus Topology :

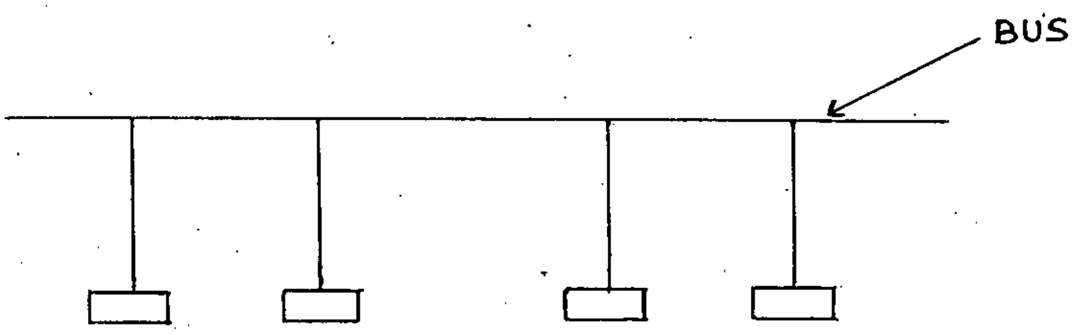


Fig. 2.3 Bus Topology : All stations are connected to a common bus

The bus topology uses multipoint transmission medium i.e. all nodes of the network are directly tied to a single transmission medium. Because all devices share a common medium, only a pair of devices on a bus can communicate at a time. But unlike the ring topology, the pair of devices can communicate directly without disturbing the remaining devices. Transmitted messages flow away from the originating node in both directions. A distributed medium access protocol is used to determine which station may transmit next. The bus topology requires no routing decision.

Tree Topology :

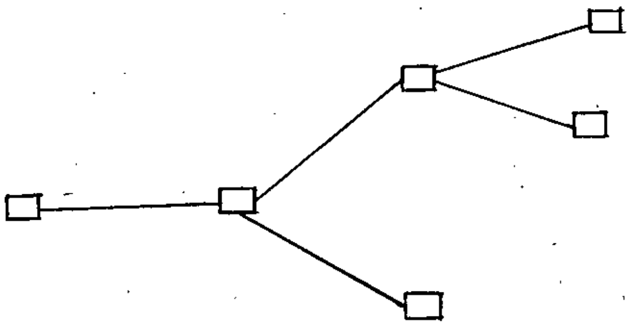


Fig. 2.4 Tree Topology : Stations form a tree

The tree topology is a generalization of the bus topology. The transmission medium is branched to accommodate the devices participating in the network. Tree and Bus topology typically use coaxial cables or twisted pair.

Multiple Bus :

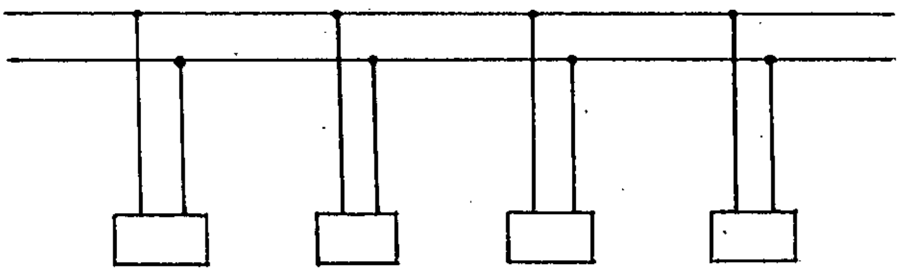


Fig. 2.5 Multiple Bus : Stations are connected to two buses

Instead of a single common bus the topology uses two separate bus as transmission medium. One bus is used to transmit data in left direction and the other is used for the reverse case.

2.5 Topology Design

Topology of the interconnection of a Local Area Network plays an important role to determine

i) The cost of implementation
ii) The performance of the network
iii) Intricated technology involved
iv) Reliability of the network
v) Degree of robustness of the network
vi) Simplicity of the software involved
vii) Extent of the flexibility of the network.

Adopting any one of the above topologies or its improved version is based on the need to fulfill the particular requirement of a certain environment. Topology design is therefore, a critical phase of network synthesis partly because of routing, flow control and similar other behaviours of the network. The locations of nodes and connection of the links in a topology design determine the transit time through the network. Again for reliable and security considerations, some networks may be required to provide more than distinct path for each node pair.

The first step of topology design is to obtain a feasible network topology. The next step is to optimize the network topology with regard to the environment where the network would be established. Topology design approach is depicted in the figure :

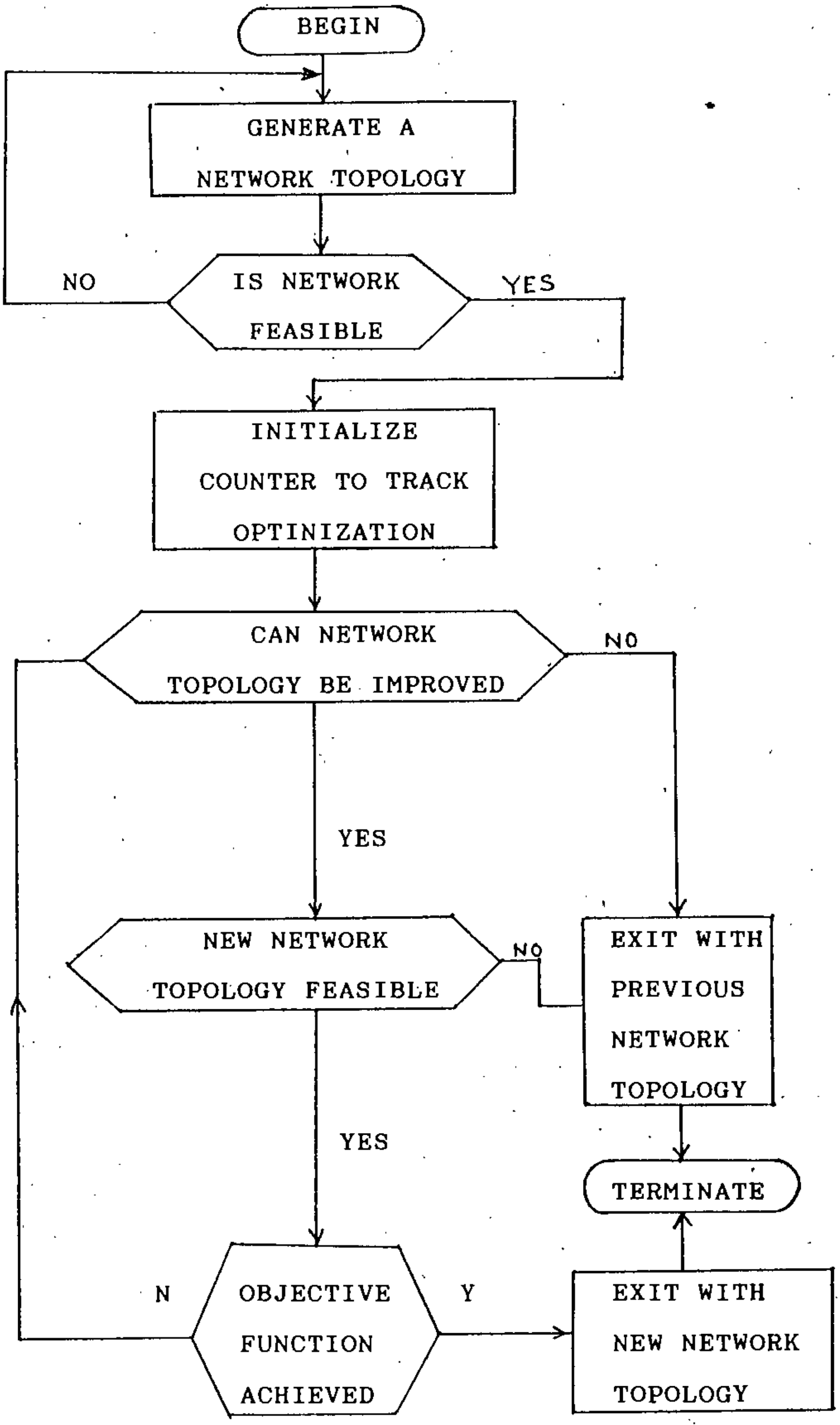


Fig. 2.6 Flow diagram of topology design

2.6 SWITCHING PRINCIPLES

There are different techniques of data transfer in communication networks. A communication network may be categorized on the basis of architecture and techniques used to transfer data. The following types are in common use:

1. Switched communication network
   a) Circuit-switched network
   b) Message-switched network
   c) Packet-switched network

2. Broadcast communication network
   a) Packet radio network
   b) Satellite network
   c) Local area network

A circuit switched network transmits a message by providing a complete path of transmission links from the message originating node to the destination node. This path is set up by a special signaling message sent by the originating node to the destination node. A response to this signaling message from the destination node informs the originating node to proceed with data transmission. This data is transmitted progressively over all the channel in the path with no intermediate store-and-forward delays. The entire fixed-delay path is allocated to this transmission, until the sender releases this path.

A message-switched network transmits a message among the nodes by moving the message through various transmission links and message buffers. A message is stored and then transmitted to the next node along the message path. A message transmission from a node does not start until a buffer at the next node on the route has been allocated for it. The path or the route for message transmission may be fixed, or it may be determined dynamically as the message progresses toward its destination node.

A packet-switched network differs from a message-switched network in that long messages are first decomposed into fixed-size segments called packets. These packets independently traverse the network until they reach the desired node, where they are reassembled into the corresponding message. Thus, many packets of the same message may in transmission simultaneously, thereby providing an important advantage of packet switching: pipelining effect. Packets belonging to a particular message are held in the memory buffers of a destination node until all the packets required to assemble the entire message have arrived. Thus, besides the store-and-forward buffers, nodes in the packet-switching networks also require reassembly buffers. A packet-switched network handles the stream of packets in two ways: datagram and virtual circuits. In datagram approach, each packet is treated independently, just as each message is treated independently in message-switched network. The packets of a particular message with same destination address do not all follow the same route. This is also possible that the packets will be delivered at different sequence from the one in which they were sent. In the virtual circuit approach, a logical connection is established before any packets are sent. Once the connection has established two stations of the network may exchange data over the logical or virtual circuit that has been established. Each packet contains a virtual circuit identifier as well as data. Each node on the preestablished route knows where to direct such packets.

Two similar types of broadcast networks are packet radio and satellite networks. In both cases, stations transmit and receive via antenna, all stations share the same channel or radio frequency. In packet radio network, all stations are within transmission range of each other, and broadcast directly from one station to other stations. In a satellite network, data are not transferred directly from transmitter to receiver but are relayed via satellite. Each station transmits to the satellite ; the satellite repeats the transmission an it is received by multiple stations.

A local network is a communication network confined to a small area. In a bus local network, all stations are connected to a common bus in a multipoint configuration. A transmission by any station propagates along the length of the medium in both directions and can be received by all other stations. A ring local area network consists of a closed loop, with each station attached to a simple repeating element. A transmission from any station circulates around the ring past all other stations, and can be received by each station as it goes by.

2.7 Transmission Media :

It is the lowest level of computer networking and provides the physical connection between the devices and the network. The widely used media in LANs are

i) Twisted Pair
ii) Coaxial Cable
iii) Optical Fiber

Twisted pair wiring is the most common communications transmission medium and is typically used for low speed data communication. The advantage of twisted pair over the other two media is its lower cost.

Coaxial cable provides higher performance requirements. It provides higher throughput and can support a larger number of devices and can span greater distances than twisted pair.

Optical fiber cable is even of greater capacity than coaxial cable and is being introduced in LAN potentially. It has been, however, little used so far due to cost and technical limitations. Optical fiber cable is immune to electrical interference and thus provides excellent security and reliability.

2.8 Access Protocols :

The access protocols defines a set of formal rules and conventions to be obeyed by the communicating devices to ensure a smooth information interchange. One of the main tasks of the access protocols is to provide a means for resolving contention when multiple devices want access simultaneously to the same transmission medium. Besides, access protocols may provides priority to a particular devices if required and may also limit transmission to a particular device at a time.

LANs of bus/tree topology usually present the following two access protocols :

i) CSMA/CD
ii) Token Bus

CSMA/CD : The most commonly used access protocol technique for bus/tree topologies is Carrier Sense Multiple Access with Collision Detection (CSMA/CD). This is also referred to as Listen While Talk (LWT). In this technique all stations have independent access to the medium. Any station wishing to transmit some thing must at first listen or sense the carrier to see if the channel is idle. If the channel is sensed busy, the station must wait till the medium becomes free.

Once transmission from a station starts, a certain amount of time elapses to reach the data to the destination station. If no other station starts transmission within this short period of time interval following the start of the transmission, the transmission succeeds. If on the other hand, any other station starts transmission being unaware of the fact that another station has already started transmitting packet that has not yet been removed from the medium, the transmissions are said to collide. The CSMA/CD protocol, however, detects any such collision. When a collision occurs, all stations which are involved in this mismatch, cease transmission immediately and wait for a certain amount of time determined from a predefined statistical calculation.

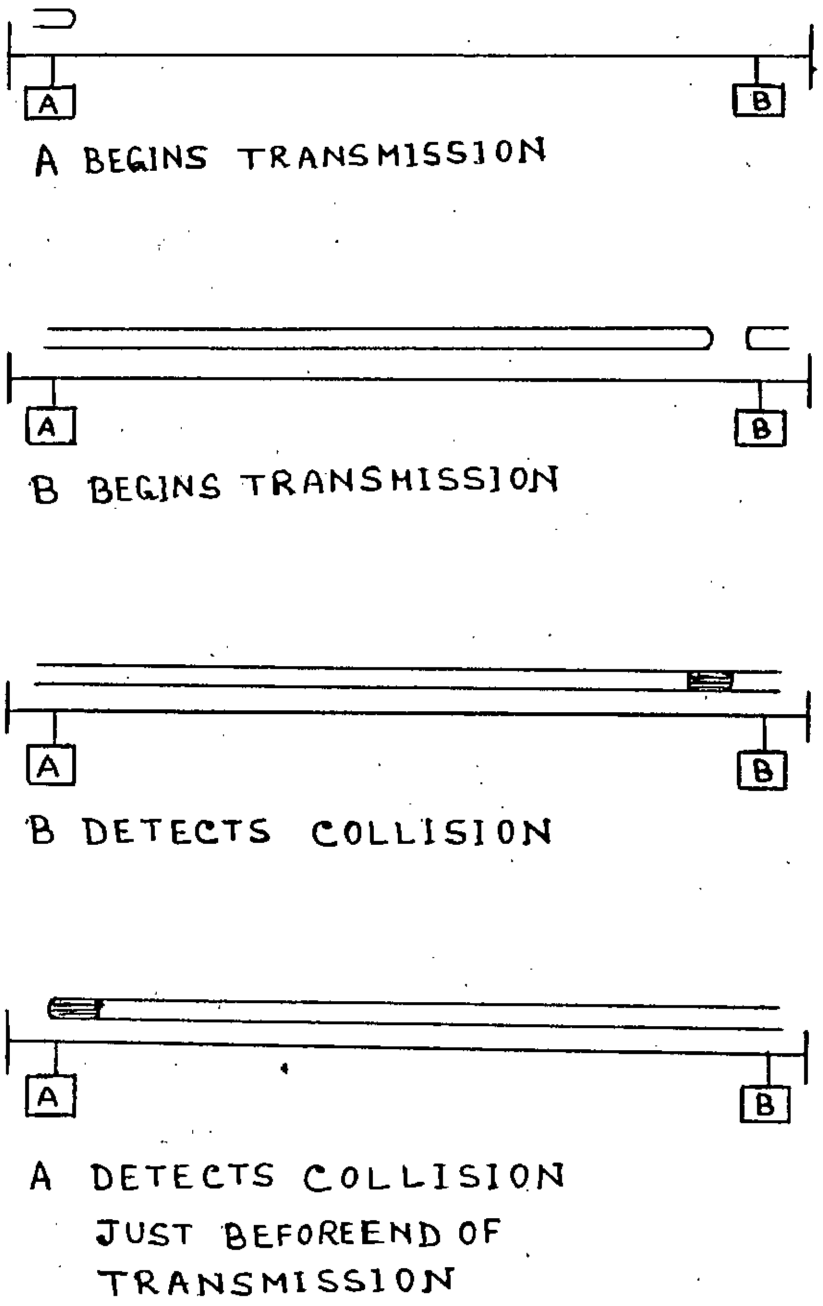


Fig. 2.7 Collision in a bus/tree topology

<u>Token Ring</u> :

To avoid any transmission collision the use of token passing techniques is quite old and still very popular especially for ring technique. This token ring technique is based on the use of a small token packet that circulates around the ring. The idle token is sequentially passed from one active Ring Interface Unit (RIU) to another in the ring. A station wishing to transmit data waits until it detects a free token passes by. The station then changes the token to a busy one and sends the entire packet with proper destination address. As the packet passes the different active RIUs one by one, each RIU checks if the destination address in the packet matches with its own address. If the destination address does not match with its own address then the RIU sends this packet to the next RIU on the ring. On the other hand if the destination address matches then the entire packet is copied by the RIU and passes this packet to the next RIU so that ultimately it reaches the transmitting RIU as an acknowledgment. The packet on the ring thus makes a round trip and is purged by the transmitting station. The transmitting station then inserts a new free token on the ring.

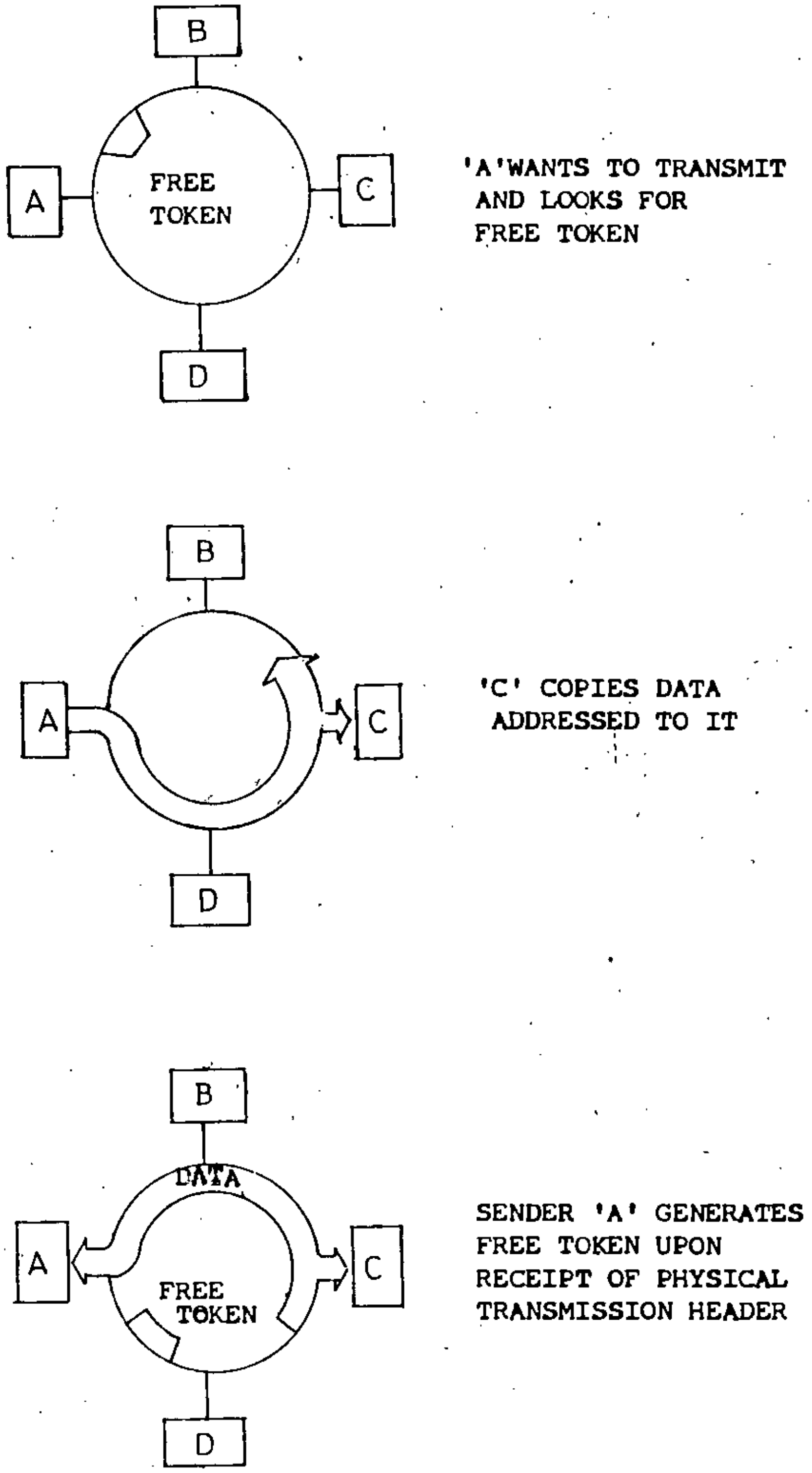


Fig.2.8 : Packet transfer in a Token Ring network

Token Bus :

In this access protocol the manner in which the stations are attached to the transmission medium is same as bus topology. But the stations form a logical ring with respect to the sequence of receiving the token on the ring. That means the stations are assigned positions in an orderly sequence with the last member of the sequence followed by the first. The physical ordering of the stations on the bus is irrelevant and independent of the logical ring.

Upon receiving a token, a station is granted of the medium for a specified time. This station may transmit one or more packets and may poll stations and receive responses. When the stations transmission is finished, or the allowed time has elapsed, it passes the token on the next station in logical sequence. The new station has now the right to begin transmission.

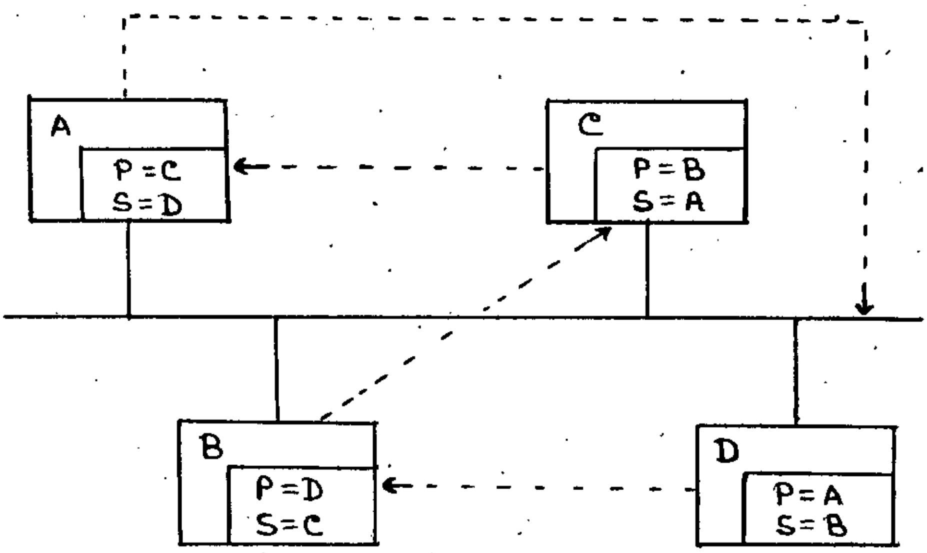


Fig. 2.9 Logical ring is formed by the stations

Open System LAN :

A network has hardware and software to support communication between or among the computers. Communications hardware is reasonably standard. But when communication is desired among heterogeneous devices, the software development effort is really difficult in its true sense. Again a one at-a-time special purpose approach to communication software development is too costly to be acceptable. The solution of this problem is to adopt and implement a common set of conventions. The International Organization for Standardization (ISO) developed the Open System Interconnection Reference Model (OSI-RM) which is a framework for defining standards for linking heterogeneous Computers.

The Reference Model of Open System Interconnection provides a common basis for the coordination of standards development for the purpose of system interconnection. It, however, allows existing standards to be placed into perspective with in the overall Reference Model. The OSI Refence Model consists of a hierarchy of seven layers.

This model provides a frame work that allows for data communication standards to be developed in an orderly and comprehensive manner. It eliminates the trouble of data communication among different devices of different data formats.

LEVEL-7 APPLICATION PROTOCOL

User level and Application Dependent
Services and Procedures

LEVEL-6 PRESENTATION PROTOCOL

Data Formats, Codes and representation
Transformation and Encryption

LEVEL-5 SESSION PROTOCOL

Control of Dialog Between Processes, Segmenting,
Buffering, Abnormal recovery

LEVEL-4 TRANSPORTATION PROTOCOL

End-To-End Control, Packet or Message
Assembly/Disassembly, Priority etc.

LEVEL-3 NETWORK PROTOCOL

Network Management, Block or Packet
Structure, Message Format

LEVEL-2 DATA LINK PROTOCOL

Data flow Initialization, Control, Termination
Error Recovery

LEVEL-1 PHYSICAL PROTOCOL

Facility- Electrical, Functional,
Mechanical Interface

Fig. 2.10 Seven Layers of OSI (RM)

1. Physical Layer : The physical layer is concerned with the transmission of sequence of bits. It does not look at the structure of sequence of new bits . It is also concerned with electrical, mechanical functional and procedural characteristics to access the physical transmission medium. Example : RS232C

2. Data Link Layer : This layer is concerned for the reliable transfer of information across the physical link. It provides the means to activate, maintain and deactivate the link. Data link protocols usually include some means of error detection, flow control and synchronization.

Example of standards at his layer is HDLC.

3. Network Layer : It provides the upper layers with independence from the data transmission and switching technologies used to connect systems, i.e. this layer provides transparency of data transfer between transport entities. The network service is responsible for establishing, maintaining and terminating connection across the intervening communications facilities.

Example : X.25 Layer 3 standard.

4. Transport Layer : The service of the layer is to provide a reliable mechanism for the exchange of data between processes in different systems. It supports end-to-end recovery, flow control and transparency of data transfer be-

tween end points. This layer may employ either virtual circuit or datagram switching tecniques to support reliable communication.

5. Session Layer : The session layer is responsible for the management of dialogue between applications. The session layer provides two application processes a means to establish and use a connection which is referred to as session.

6. Presentation Layer : The presentation layer is concerned with data representation. It eliminates the differences in format and data representation. Thus it provides independence to the application processes from differences in syntax.

7. Application layer : This layer is responsible for the access of the application processes to the OSI environment. A variety of application specific protocols are provided here which might include services to facilitate file transfer, information retrieval, text editing etc.

The advantage of OSI approach is that it solves the heterogeneous computer communication problems. Two systems can communicate effectively if they implement the same set of communication functions into the same set of layers and if peer layers share a common protocol.

CHAPTER 3

TOPOLOGY AND
NETWORKING TECHNIQUES

3.1 Topology Selection :

The ring topology is generally feasible when a LAN designer wants to keep the network implementation process simple and economic. The ring type LAN is the primer but it is very popular among network designers. The extension of the network is possible in ring system and the required costs for such extension is simply additive on the excess elements involved.

3.2 Network Interface :

A typical token ring LAN interface mainly requires

1) A packet processing unit (8088, 8086, Z80 etc.) : It processes data packets coming from the preceding station and it sends necessary information in a packet form to the next station of the loop. The packet processing unit on the interface relives the central processing unit of the host computer from processing the packets.

2) A transmitter-receiver unit : This unit receives data packets from the preceding interface and interrupts the packet processing unit of its own interface board to process the packet. The transmit-receiver unit also transmit the data packets to the next interface.

3) A memory unit : This memory unit consists of two parts. One part (ROM) holds the program for the interface and the

other part (RAM) acts as a buffer memory.

In the proposed token ring network there will be no separate interface between the network and the host computer. The above elements are available with the system unit of a computer. The RS232C will be used as a node within the network. The processor of the host computer will perform major tasks which are performed by interface processor in a conventional token ring network. Now if a small token is made to circulate constantly around the ring, then the processors of all the computers of the network will remain busy just to make a route for the token. And this type of involvement of the processors of the computers will ensure wastage of more time with the decrease of the number of the computers within the network. As a result, processing of local functions of a computer will be delayed. The problem is solved by holding the control token with a station instead of allowing the token to circulate around the loop. Which station will hold the control token is determined by the medium access rules and these are stated in section 3.4.

3.3 Networking Techniques :

A communication network is a shared resource. It provides the sharing of transmission facilities among a group of stations, which reduces the cost incurred by any pair of stations.

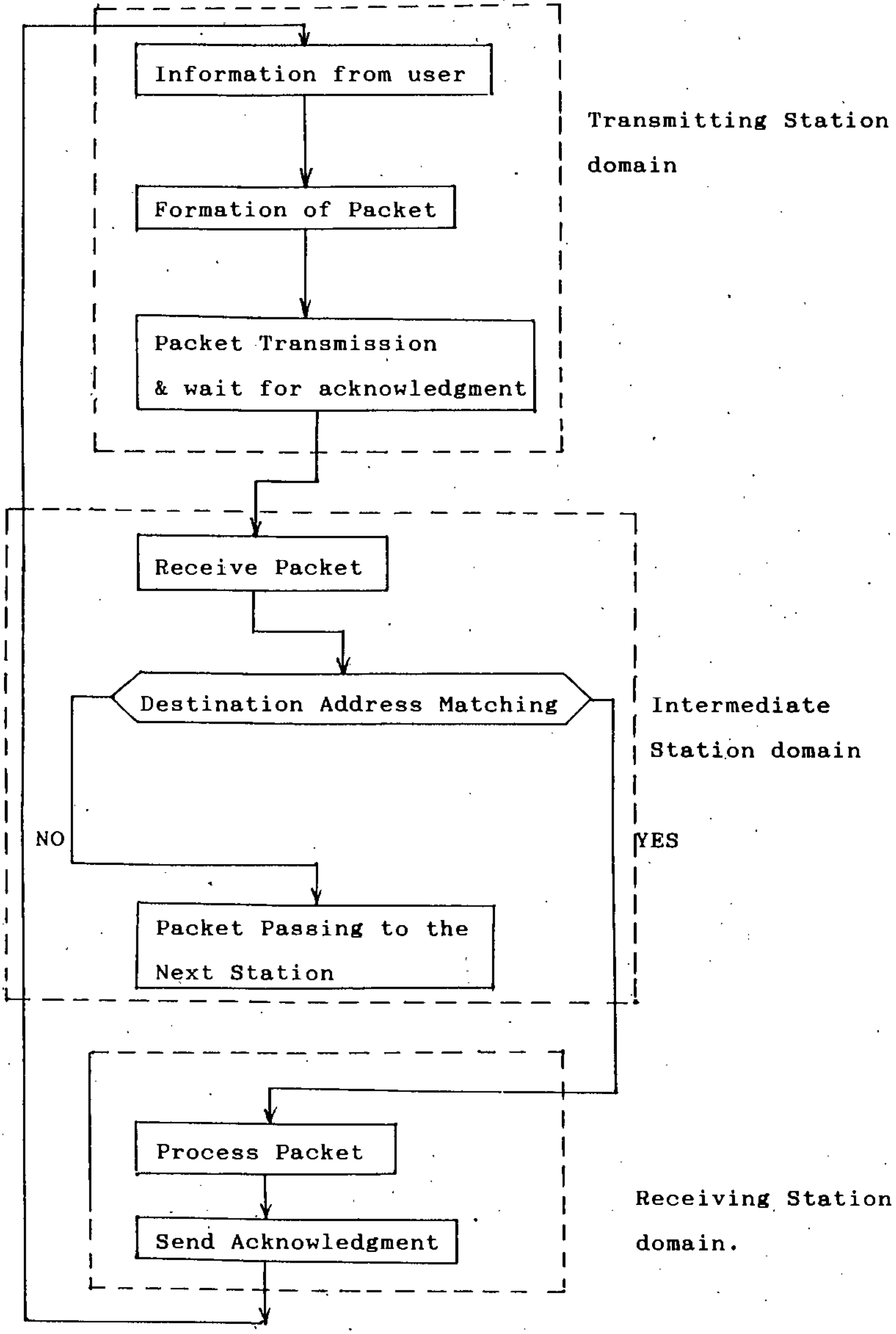


Fig. 3.1 Three domains of a station's actives for a network function

The topology of the proposed network does not allow in its true sense any dedicated communication path between two station. There is no need to establish a station-to-station circuit connection and then transmit data and consequently disconnect the connection. Instead the data to be transmitted are at first processed to form one or more packets. Each packet contains the source and destination addresses. The station with free token then sends the first packet to the next station within the loop. The adjacent station receives this packet and process the packet to determine the address mentioned in the packet. If the destination address as mentioned in the packet matches with the address of this receiving station, it will send an acknowledgment to the transmitting station down the loop. Otherwise it will just pass the packet to the next station within loop. If the addressee as mentioned in the packet is not active within the loop, the process of token passing to the next station continues until the packet reaches the original transmitting station.

Thus data transmitted by one station on the proposed network is received not only by the destination station, but also by the other stations in between the source and destination stations.

3.4 Medium Access Control Techniques :

The network will use a common transmission medium for the communication between any two stations. There must be some access control technique so that only one station can successfully transmit on the shared medium at a time. In a centralized network, a controller is designed to determine which station will access on the network. This scheme however, has the disadvantages of a single point failure and high propagation delay. In the proposed network all the stations collectively perform a medium access control function to determine who will access on the network.

In the proposed token ring the small token will not circulate constantly around the ring when all the stations are idle. Instead, a free control token will be in the possession of the station which has transmitted most recently. Any other station wishing to communicate with another station must send an interrupt to the down stream station asking the free token. If the station at the down stream holds the free token, it will send the token to the calling station through the other down stream stations. The free token like any other packet will always travel on the ring in a particular direction (clockwise or anticlockwise). This constraints of the proposed network limits the speed of the data transfer on the network. On the other hand the unidircetional traveling of the packet minimizes the number of wires in the cables and thereby reduces the cost of the cables for connecting two consequtive stations through RS232C.

Token Asking

If a station with no control token receives a request for control token from its previous station, it will then pass the request to the next station in the loop. The processes of passing the request of the original calling station will continue until the token is found. If the calling station can identify that the token is not available on the network, it will then generate a new free token and then it will start communication.

Token Generation :

There are two cases when the network will be required to generate a new token.

Case I : All the computers of a network are turned off. If an user of the network turns on his computer and then loads the network software, there will be no token on the network. This is because the software does not generate a token when it is simply loaded into the memory of the computer. Similarly when a second user turns on the computer & loads the software there will be still no token. The first user will be able to recognize this failure by identifying interrupt for token request which was originally generated by him and again comes to him after passing through the different computers of the network. In this case the network software remaining resident in the computer memory of user # 1 will generate a new token and will permit him/her to communicate with the user # 2.

Case II : Some cases may arise when the existing token will be lost. If a computer, which was processing the token, is shut down, or rebooted, then the existing token will be lost. Besides, if the token possessing computer fails to recognize the interrupt for token request from a computer up stream, a new token will still then be generated. This later case however will bring another uneasy situation of duplicate token within a network.

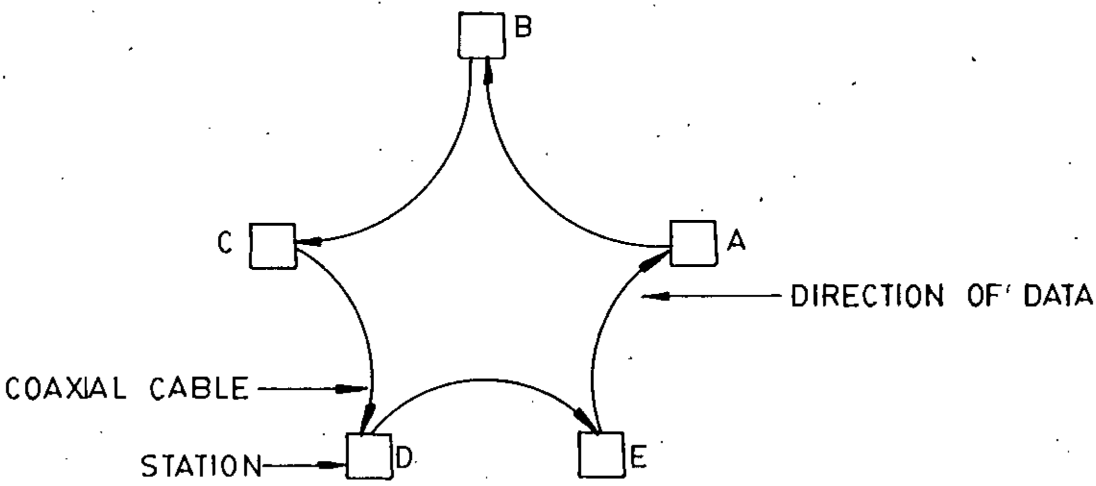


Fig.3.2 No station has yet transmitted any data after turn on. Hence no control token on the network system.

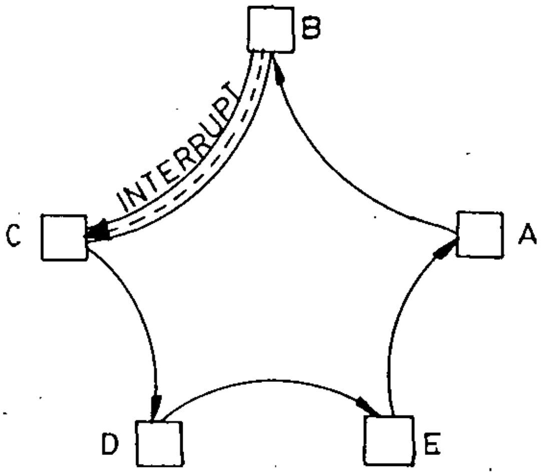


Fig.3.3 Station B wants to start transmission. It sends the interrupt "B wants control token"to the station C. Station C has no control token. As a result C passes the interrupt "B wants control token" to D. Since no station has control token hence the interrupt " B wants control token" ultimately comes back B.. B then generates a token assuming that there is no control token on the network system.

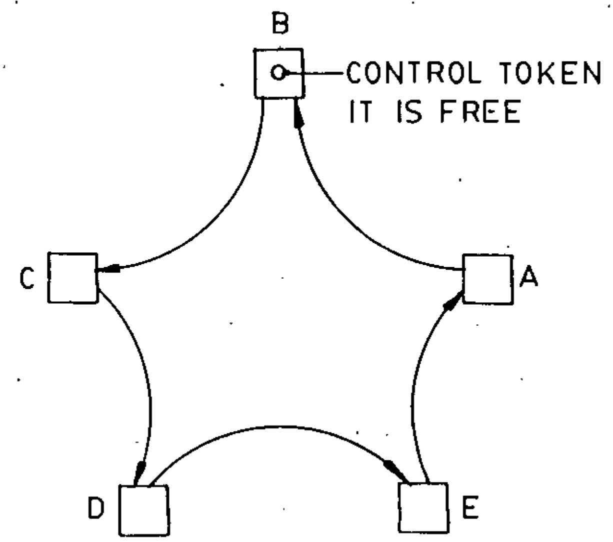


Fig.3.4 Control is in B's possession. And it is a free token.

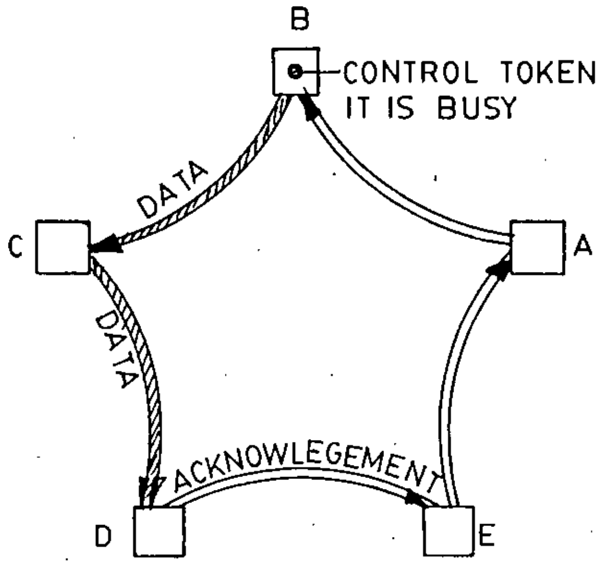


Fig.3.5 B has already stared transmission. B still pos sesses the control token, it is now busy token.

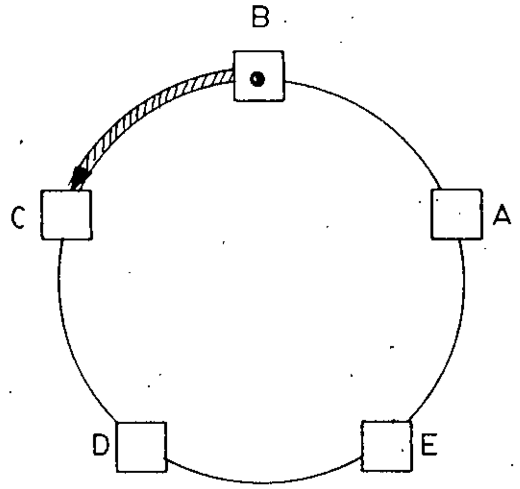


Fig.3.6 A wants to transmit. So it sends the interrupt "A wants the control token" to B. But B rejects the request as the token is now at busy state.

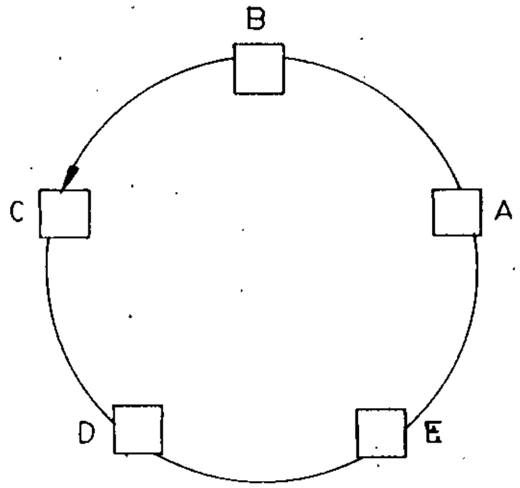
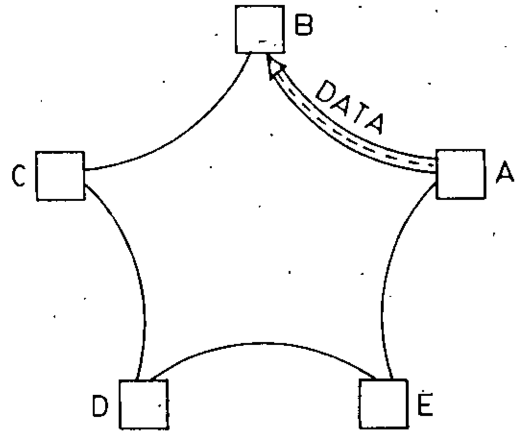


Fig.3.7 B has completed transmission. Now B possesses the free token. The token will remain with B until no other station wants the token.



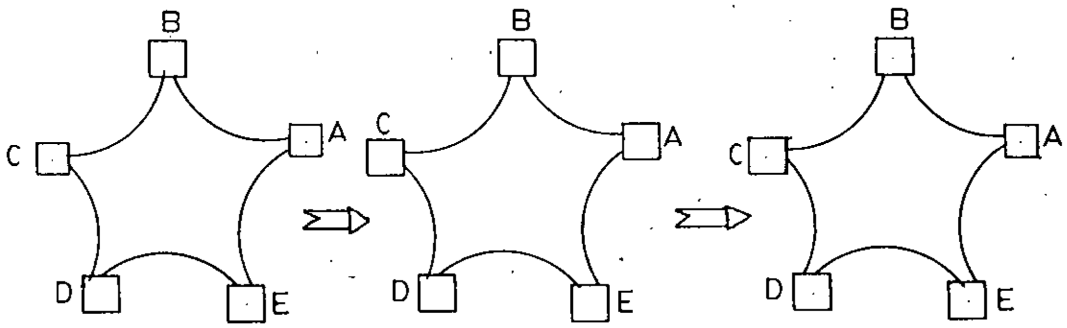


Fig.3.8 After a few seconds A again sends the interrupt. B then send the control token to A via station C, D and E.

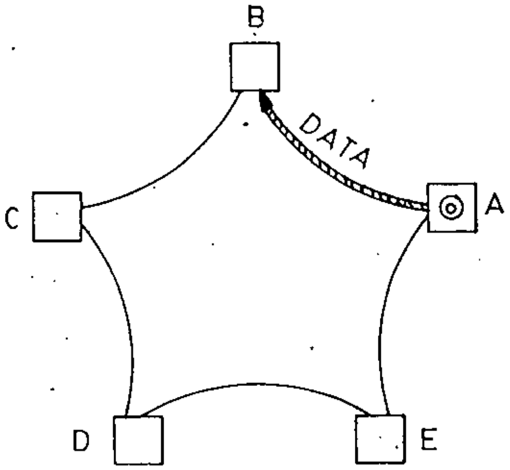


Fig.3.9 Now A is transmitting data to a station of this network.

Avoiding Duplicate Token :

If a computer possesses a token and it receives an interrupt for any thing other than token, it will assume that there are more than one token on the network and this duplicate token gives other station the capability to begin transmission. In this case the network software will remove the token remaining within itself.

Node Design Concept :

It has been stated that the network does not require any special interface between each computer and the network ring. Instead, the processing unit of the system of each computer and an asynchronous communication adapter are used to handle the hardware side of communication processes. The main elements of a probable node are a processing unit (like Intel 8088/8086 or Z80), an asynchronous receiver-transmitter(e.g. Intel 8250), an interrupt controller (e.g. Intel 8259), an EIA driver (e.g. 75150), an EIA receiver (e.g. 75154), etc. The block diagram in Fig. depicts the probable node design concept.

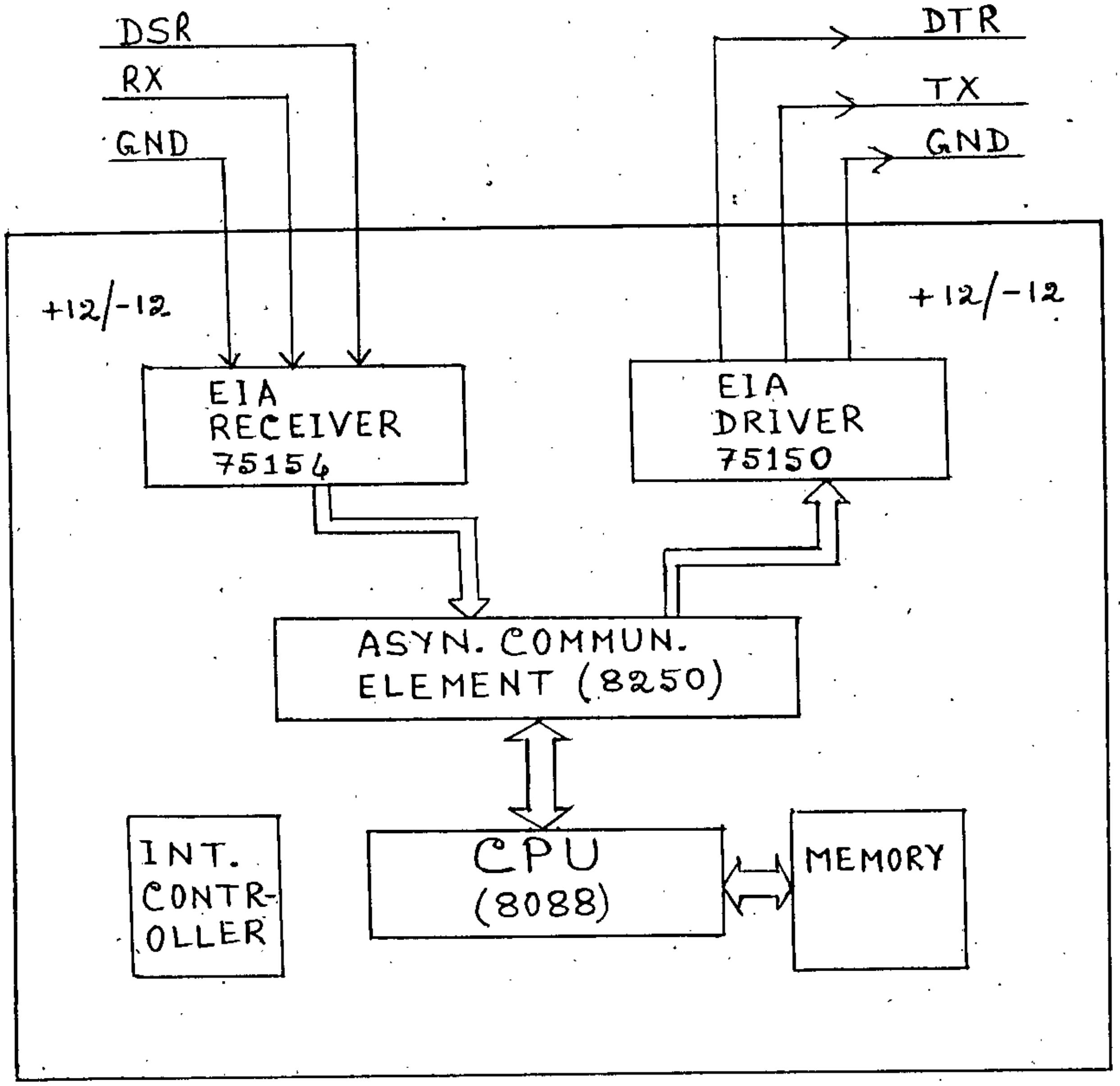


Fig. 3.10 Elements of a probable node

CHAPTER 4

NETWORK FUNCTIONS

4.1 Functions of the Network

The network software is responsible for establishing an effective and meaningfull communication link between two users. When two computers are engaged in a communication work (like file transmission), the other computers of the network involves in that process for successfull communication between the two users. For this following functions are included in the network software :

1) Frame Control
2) Error Control
3) Screen Control
4) Abnormality Recovery
5) Initialization Control
6) Sender Identification
7) Receiver Identification

1) Frame Control : Two types of frames are used in the network communications. One is referred to as Text Frame and the other is the Control Frame. Text frames are used to transmit or receive messages or contents of a file whereas control frames are used to carry requests and acknowledgments for the smooth transmission of a message or a file. The requests and acknowledgments of a control frame may be

a) Request to create a file
b) Request to open a file for read

to 1. The size of the control frame is fixed and it is of 11-byte long. The size of the text frame is, however, flexible with an upper limit of 267. First nine bytes are used as header and the last 2 bytes are used for 16-bit CRC. The remaining 256 bytes may be used for the transmission of text.

2) Error Control : It provides CRC check of each packet. The input of the Error Control routine are the contents of registers SI and BX. The content of SI points the data packet and that of BX indicates the number of bytes in the packet. The Error Control routine outputs a 16 bit CRC in the BX register. However, if the received packet contains any error, then the content of BX will be zero instead of the 16 bit CRC.

3) Screen Control : The network software changes the computer screen in two instances.

i) If an user wants to start communication with any other station, then a new screen with a network menu replaces the user's screen. After completion of the communication the user's original screen, however, comes back.

ii) If an user receives any message from any other user of the network, the user's screen at receiver is replaced by a new screen showing the received message. The user, however, can go back to his original screen by simply pressing any key of his keyboard.

c) Request to send a file

d) Request to send control token

e) Acknowledgment of a successful packet reception

f) Acknowledgment of reception of data with error etc.

The structure of the text frame is shown below :

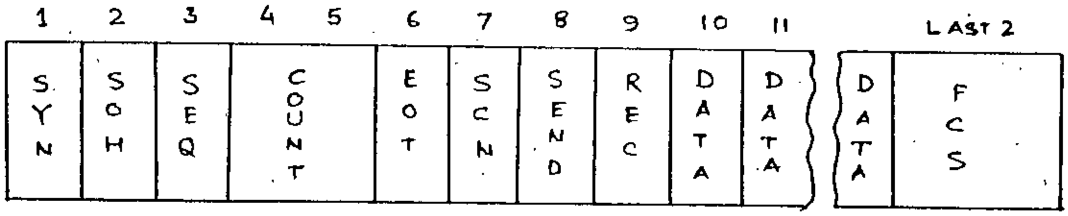


Fig. 4.1 Text frame structure

1st byte : Synchronization byte, SYN

2nd byte : Start of header for text frame, SOH

3rd byte : Sequence number of packet, SEQ

4th &

5th byte : Number of character in the packet

6th byte : End of Text Marker, EOT

7th byte : Code for Screen display / File write, SCN

8th byte : Sender's address, SEND

9th byte : Receiver's address, REC

10th byte to

3rd last byte: Text to be transmitted, Data

(Maximum 256 byte)

Last 2 bytes : 16 bit CRC.

The structure of the Control frame is given below :

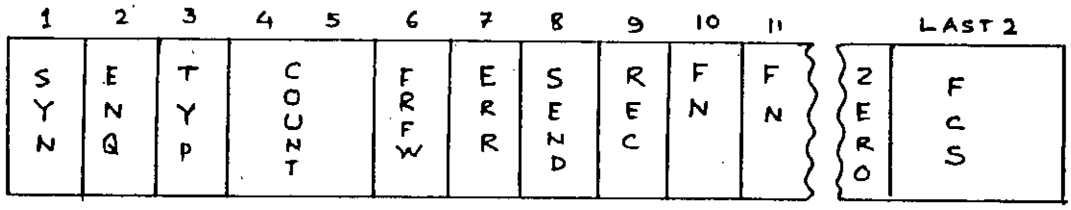


Fig. 4.2 Control frame structure

1st byte	:	Synchronization, SYN
2nd byte	:	Enquary, ENQ
3rd byte	:	Type of Enquary, TYP
4th and 5th byte	:	Count of Characters in packet, COUNT
6th byte	:	File Read or File Write marker, FR/FW
7th byte	:	Error code if any, ERR
8th byte	:	Sender's address, SEND
9th byte	:	Receiver's address, REC
10th byte to last 4th byte	:	File name
3rd last byte	:	ASCIIZ for file name, ZERO
last 2 bytes	:	16 bit CRC

After receiving a packet, the network software checks the ENQ byte to determine the type of frame. If ENQ = 1, then it is a text frame and if ENQ = 5, it is a control frame. Contents of DATA of the text frame is displayed on the screen if SCN = 1. On the other hand the contents of DATA of the text frame are written on a file if SCN is equat

4) Initialization Control : It deals with setting up the required ICs to their respective mode of operations. Communication port is initialized with the following parameters.

Baud rate	:	9600
Parity	:	No. Parity
No. of Stop bit	:	1
No. of data bit	:	8

Besides it enables the modem status interrupts for 8250 interrupt signal and also enables IRQ4 by programming 8259 Interrupt Masking Register.

4.2 Modes of Operation

There are three modes of operation in the network.

Idle Mode : When no computer is engaged in communication through the network, then the mode is called idle mode. In idle mode the control token does not circulate round the ring. Instead, the control token remains with the computer which has used the token most recently for communication.

Transmit Mode : The network software provides two types of transmit mode - enquary transmit mode and text transmit mode. In enquary transmit mode, a station can interrupt the next station to pass a request or an acknowledgment. In this mode, the station does not require the control token to be with itself. As for an example when a station successfully receive a message from another station, it then send an acknowledgment to the message transmitting station.

On the other hand, in text transmit mode either message or content of a file or a request to send a file is transmitted from a station. For this the station which initiated the communication must have the control token. Sending message by an user from station A to station B is an example of test transmitting mode.

Receive Mode : There is no clear cut distiction between an Idle mode and a Receive mode. When a computer receives an interrupt from the preceding station it goes to receive mode and tries to receive a packet from that station. After a packet is received, the 16-bit CRC is checked by the network software. If the packet is found to be error free, the content of the packet is processed for subsequent action. On the other hand a negative acknowledgment is send if any error is found in the packet by checking the CRC bits.

The received packet may be either a control frame or a text frame. Both of these frames are served by a routine called Command Processor. The content of text frame is displayed on the screen if the 7th byte of the frame, SCN = 0. Otherwise the content is written onto an already created file. If SCN is not equal to 0 and EOT = 17H then after writing the content of the frame on to the file, the network close the file and goes back to Idle mode.

4.3 Asynchronous Communication Adapter

IBM Asynchronous Communication Adapters or equivalents are used for the network. Each computer of the network must have such an adapter. Two jumper models of the adapter are used to select RS-232C operation and to select the primary address for the adapter. The primary addresses are listed below.

I/O Decode (Hex) Primary Adapter	Register Selected	DLAB State
3F8	TX Buffer	DLAB = 0(WRITE)
3F8	RX Buffer	DLAB = 0(READ)
3F9	Interrupt Enable Register	
3FA	Interrupt Identification Register	
3FB	Line Control Register	
3FC	Modem Control Register	
3FD	Line Status	
3FE	Modem Status Register	

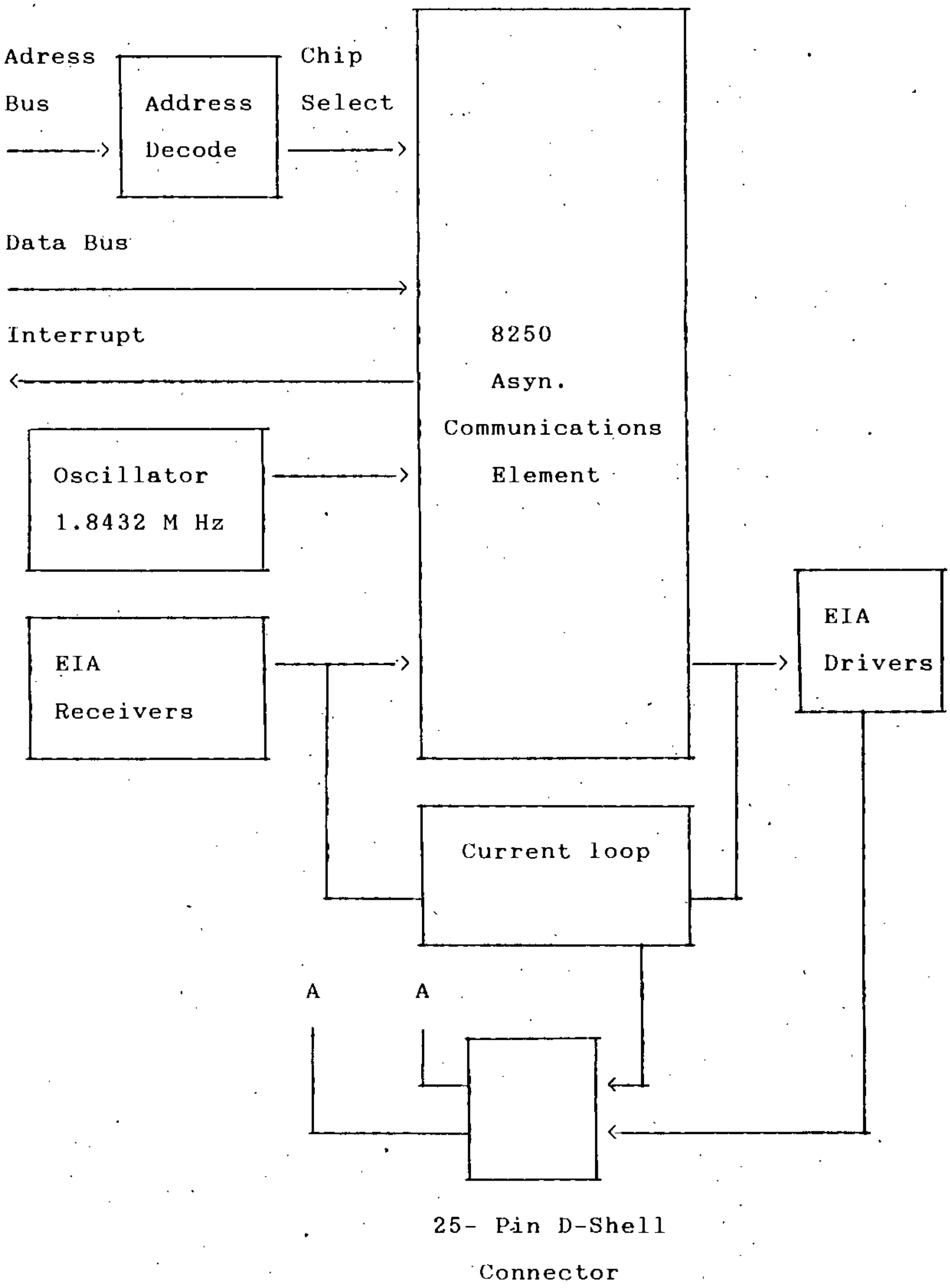


Fig. 4.3 Block diagram of asynchronous communication adapter

The heart of the adapter is a INS8250 LSI Chip or equivalent. Main features of the adapter are :

* It adds and removes start bits, stop bits and parity bits.

* Programmable baud rate generator allows operation from 50 baud to 9600 baud.

* 5, 6, 7 or 8 bit character with 1, 1-1/2 or 2 stop bits.

* A prioritized interrupt system controls transmit, receive, error, line status and data set interrupts.

* Full double buffering eliminates need for precise synchronization.

* Modem control functions : CTR, RTS, DSR, DTR, RI, car rier detect :

The figure on the next page shows that block diagram of the adapter.

Voltage Interchange Information

```
                    Invalid Levels
+ 15 Vdc  -----------------------------------

                    On Function
+  3 Vdc  -----------------------------------

   0 Vdc            Invalid Levels

-  3 Vdc  -----------------------------------

                    Off Functions
- 15 Vdc  -----------------------------------

                    Invalid Levels
```

Binary State	Signal Condition	Interface Control Function
Binary (0)	Spacing	On
Binary (1)	Marking	Off

Interrupt

One interrupt line is provided by the adapter to the system. This interrupt is IRQ4 for the primary adapter. To allow the communication card to send interrupts to the system, bit 3 of the modem control register must be set to 1 (high). At this point, any interrupts allowed by the interrupt enable register (IER) will cause an interrupt. The instructions to set Modem Control Register's 3rd bit to 1 are

```
MOV  DX,  3FCH
MOV  AL,  08H
OUT  DX,  AL
```

4.4 Data Link Control

It establishes data link between two computers, maintains the already established data link until the completion of data transfer and releases the data link when data transformation takes place successfully. This data link control does the task of programming the communication adapter and interrupt controller, Intel 8259 or equivalent. It supervises the formation of packets and checks any error within the packet.

Following are the important functions which are included in data link control to provid a means to establish, maintain and release data links between two computers.

Transmitting Station Ready

When a computer is ready to transmit a packet, the DTR pin of 8250 is made low. The DTR output signal set to an active low by programming bit 0 (DTR) of the Modem Control Register (MCR) to a high level. This is done as

```
MOV   DX,   3FCH     ;     Loading address of MCR
MOV   AL,   01
MOV   DX,   AL       ;     Writing '1' to DTR bit of MSR
```

The DTR pin of 8250 of the transmitting station is connected to the DSR pin of 8250 of the receiving station through EIA driver and receiver as shown on the next page.

The low output signal of the DTR of Transmitting Station interrupts the receiving station and informs that the transmitting station is ready to send a packet.

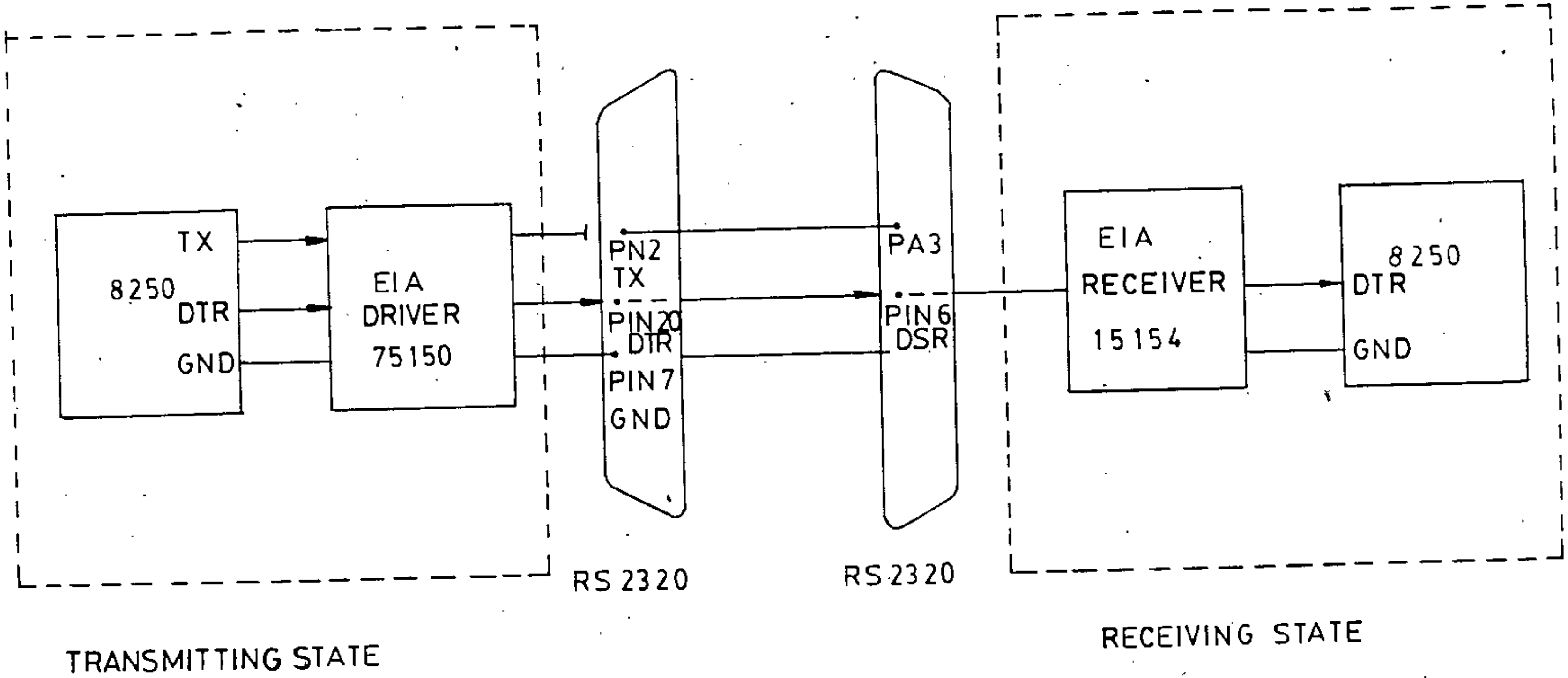


FIG. 4.4 CONNECTION OF TWO CONSENTIVE STATION

Transmitter Holding Register Empty (THRE)

Line Status Register (LSR) provides Status information to the processor concerning data transfer. Bit 5 of LSR is THRE indicator. This bit indicates that the 8250 is ready to accept a new character for transmission. The following is for checking LSR.

```
THRE :  MOV   DX,   3FDH      ;    Load address of LSR
        IN    AL,   DX
        AND   20H             ;    Check bit 5 of LSR
        J2    THRE
```

Loading Transmitter Holding Register (THR)

The Transmitter Holding Register contains the character to be transmitted serially. The address of THR for primary adapter is 3F8H. The program segment to load THR is as follows :

(AL contains the character to be transmitted)

```
        MOV   DX,   3F8H      ;    Loading address of THR
        out   DX,   AL        ;    Loading THR with the
                                   required character.
```

Testing of the Character Arrival

Whenever a complete incoming character is received and transferred into the receiver buffer register, the bit 0 of Line Status Register (LSR) is set to 1. The program segment for testing of LSR for character arrival is given bellow :

```
DR :    MOV   DX,    3FDH        ;     Loading address of LSR
        IN    AL,    DX          ;
        SHR   AL,    AL, 1       ;     Check whether character
                                 ;     arrives
        JNC   DR                 ;
```

Unloading Receiver Buffer Register (RBR)

The Receiver Buffer Register contains the received character. If bit 0 of LSR is found to be 1 in the previous test, then the following program segment used for unloading the RBR.

```
MOV   DX,    3F8H      ;     Loading the address of RBR
IN    AL,    DX        ;     Unloading the character
                       ;     from RBR
MOV   [SI], AL         ;     Store the received
                       ;     character into input buffer.
```

(SI contains the offset of input buffer.)

4.5 Function Documentation

Flowcharts of the algorithms of the important functions of the network software appear on the following pages.

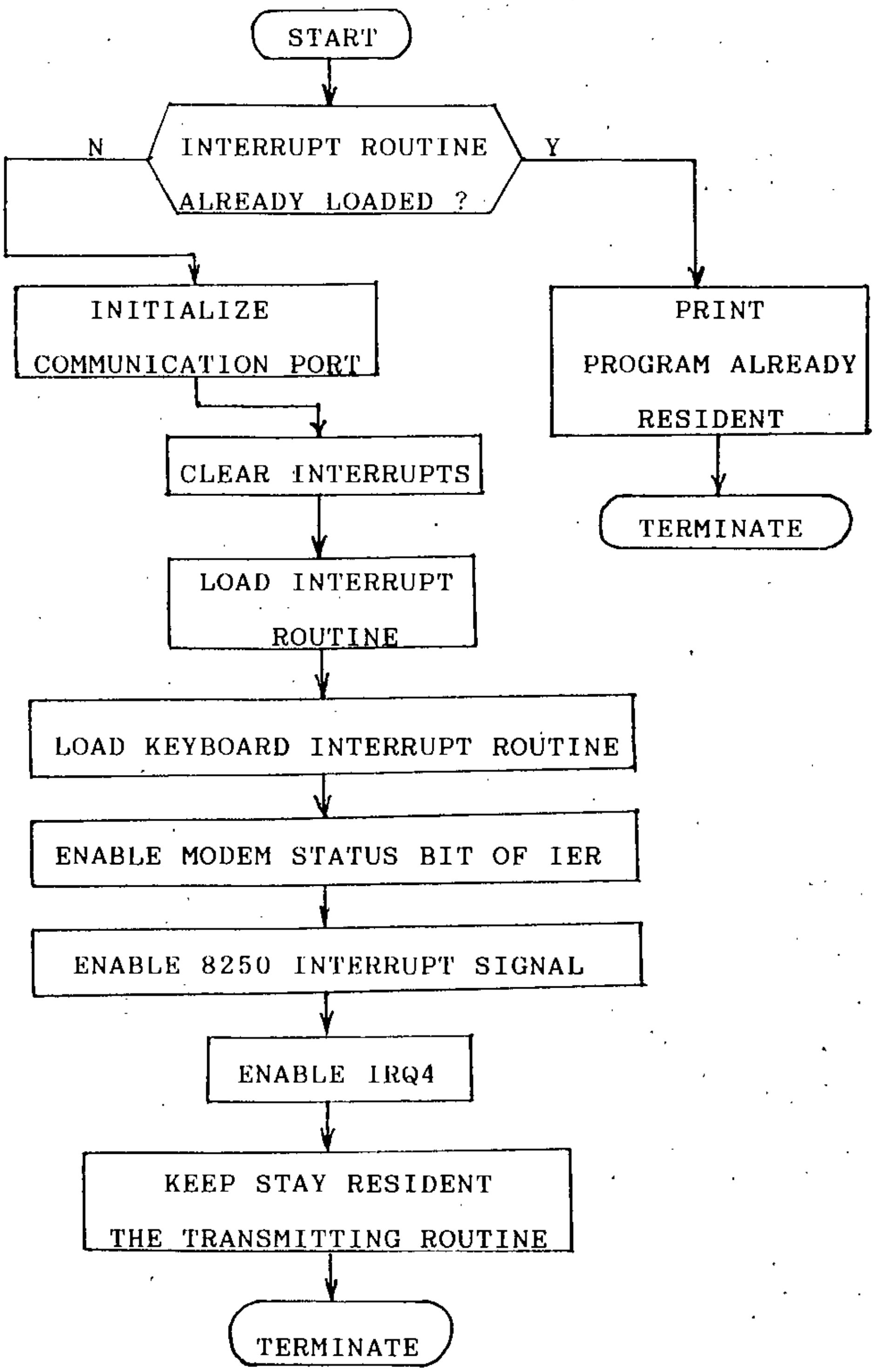


Fig. 4.5 Flow diagram of Loading routine

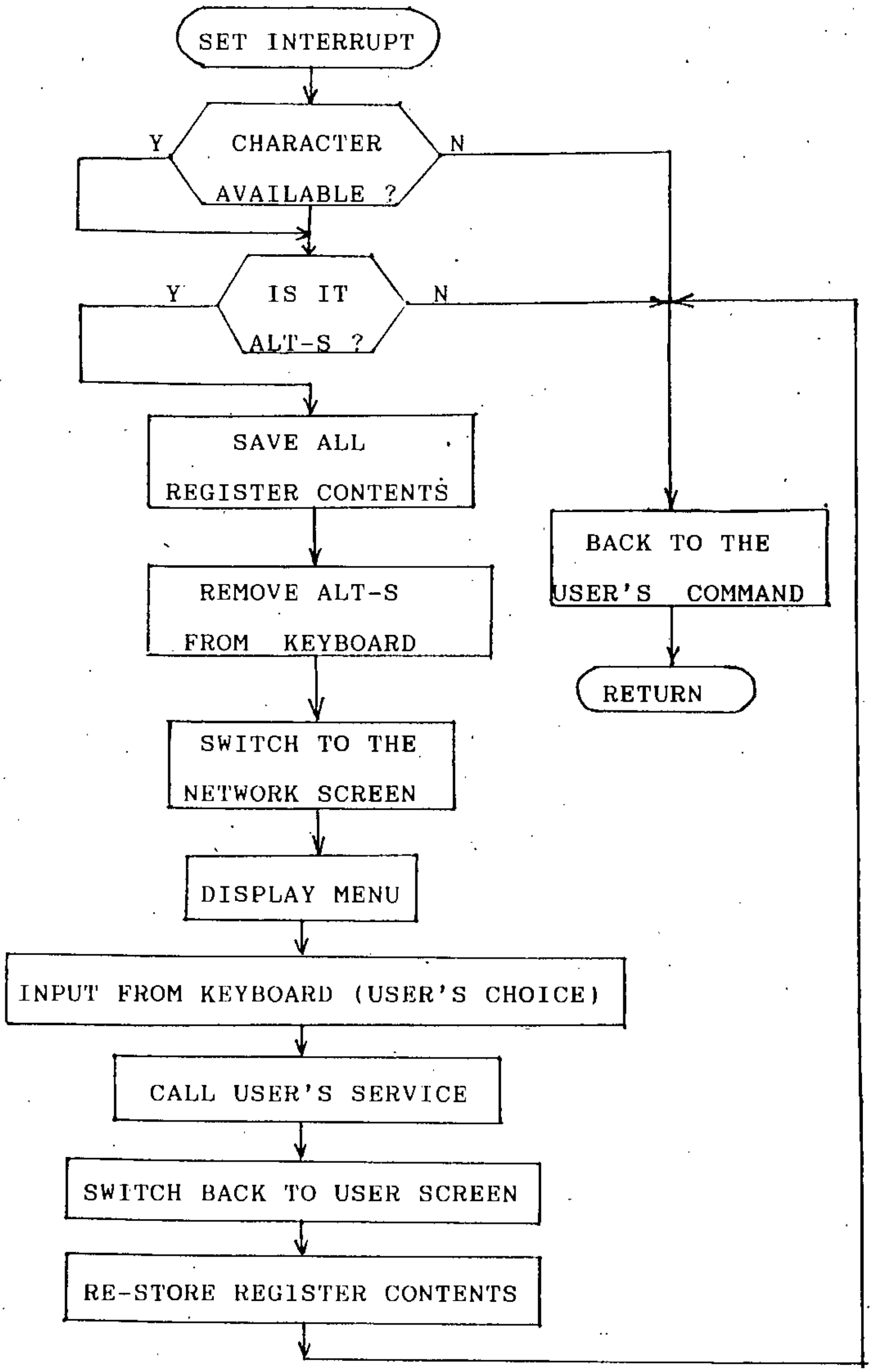


Fig. 4.6 Flow diagram of Keyboard Interrupt routine

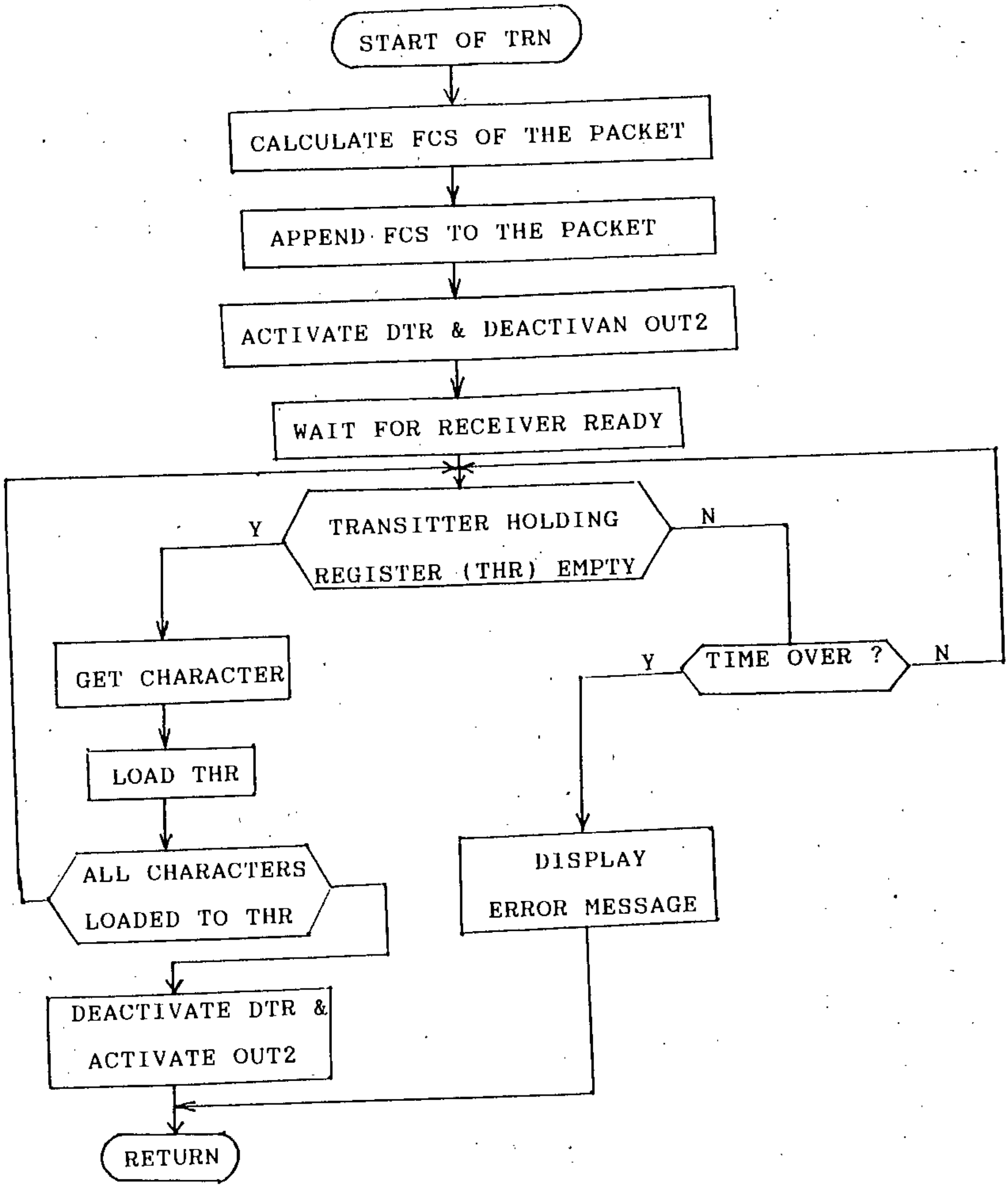


Fig. 4.7 Flow diagram of Transmitting routine

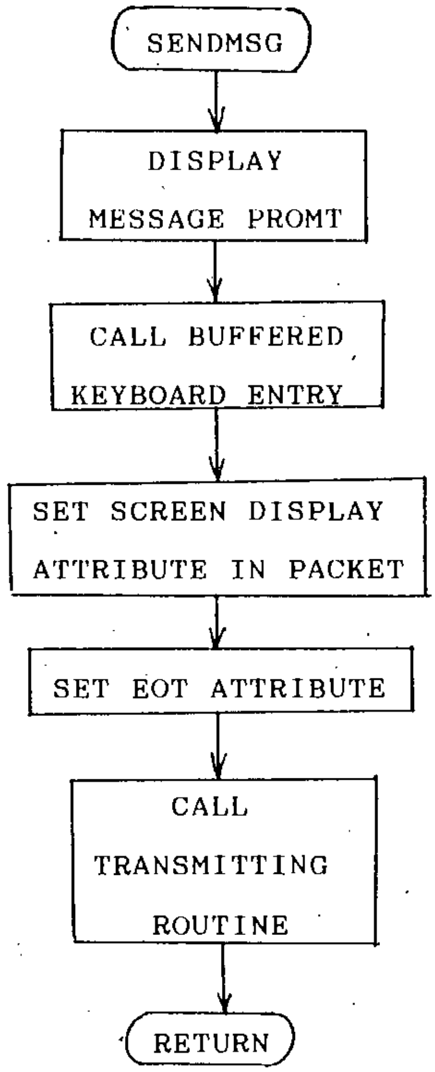


Fig. 4.8 Flow diagram of Message Send routine

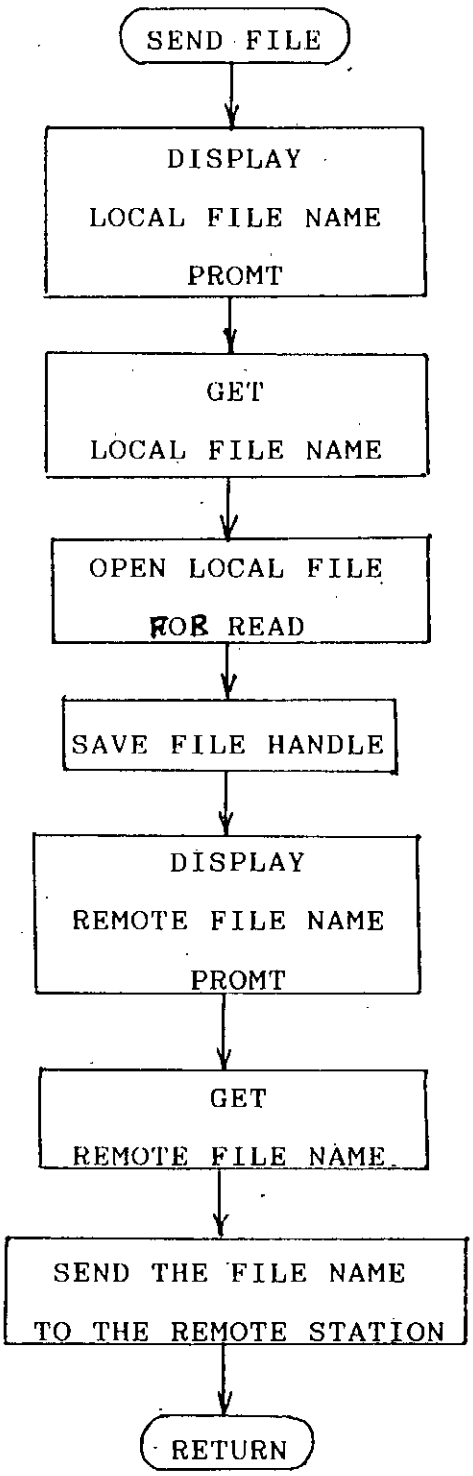


Fig. 4.9 Flow diagram of Send File Request routine

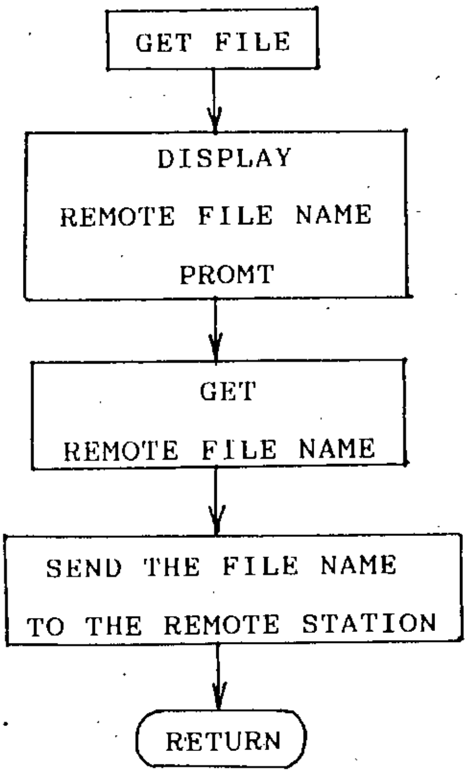


Fig. 4.10 Flow diagram of Get File Request routine

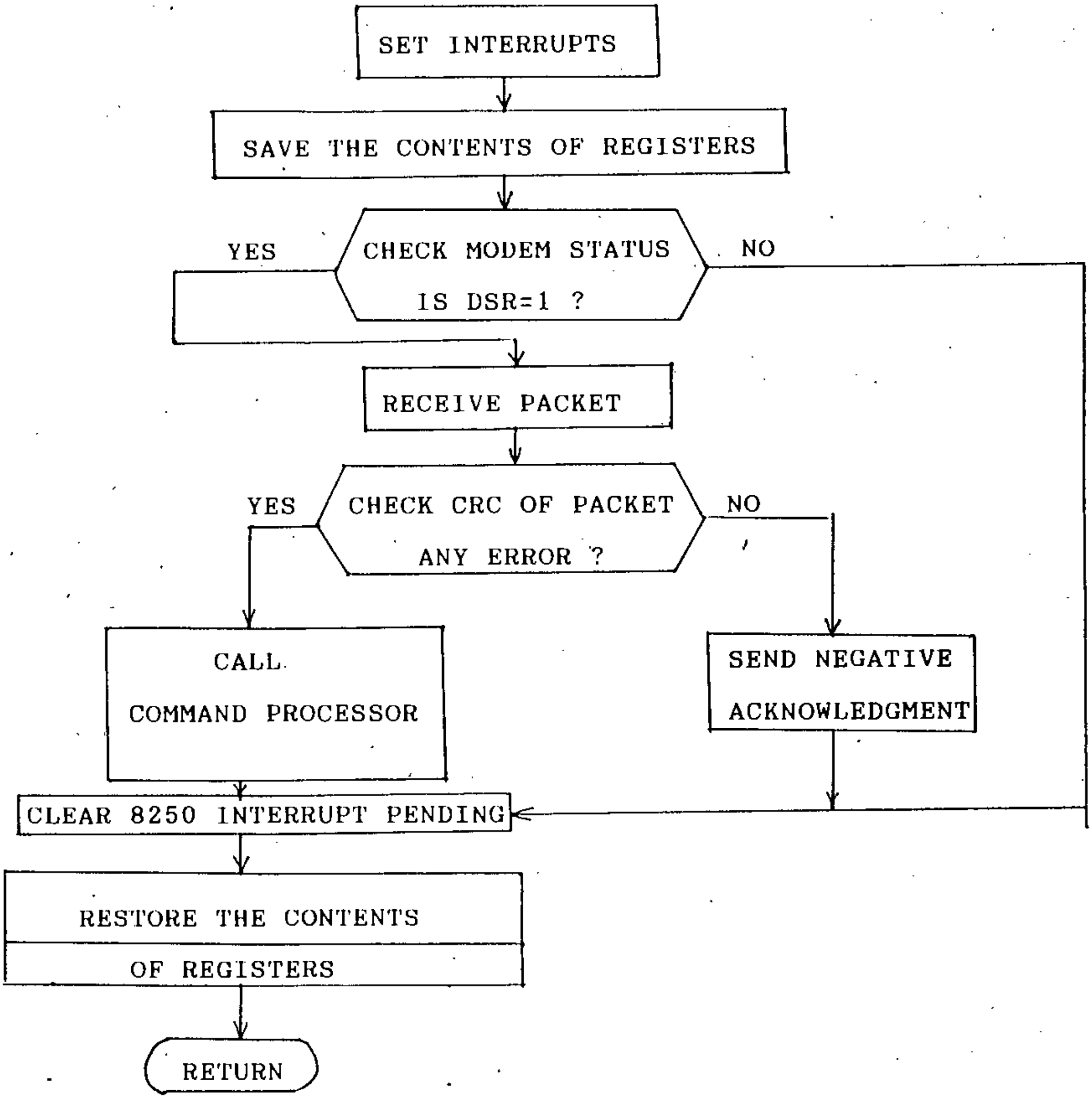


Fig. 4.11 Flow diagram of Communication Interrupt routine

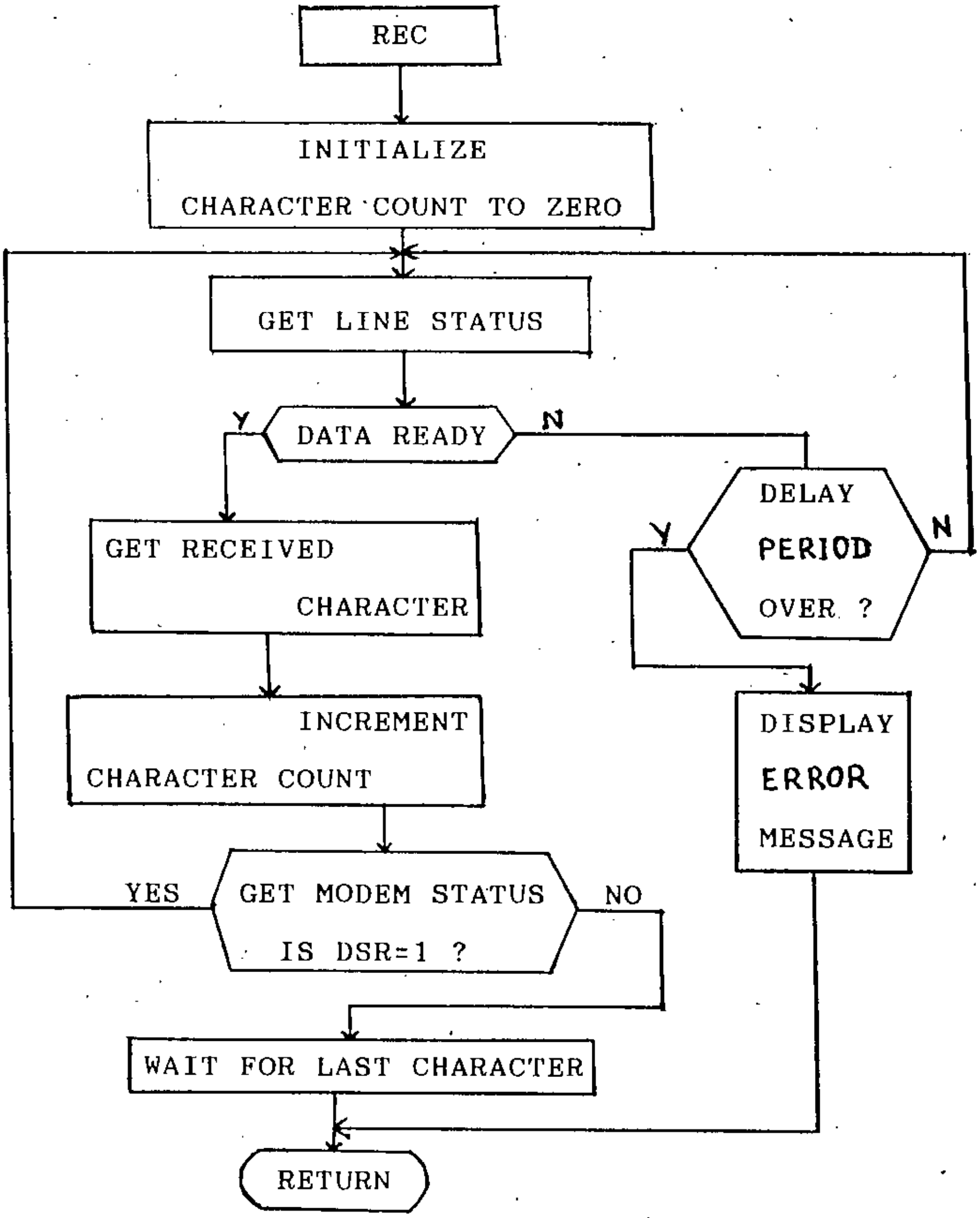


Fig. 4.12 Flow diagram of Receive routine

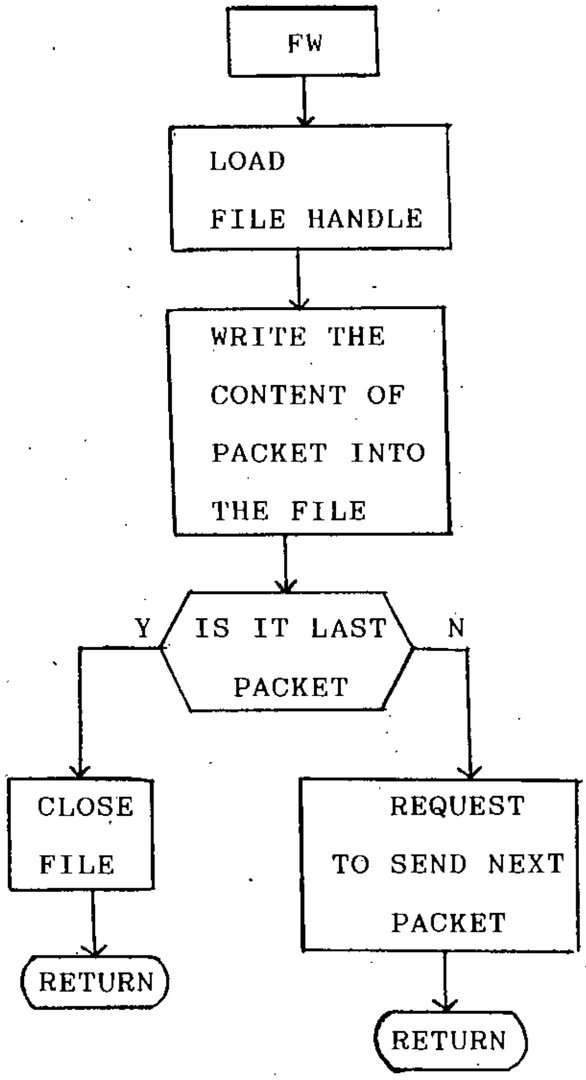


Fig. 4.13 Flow diagram of File Write routine

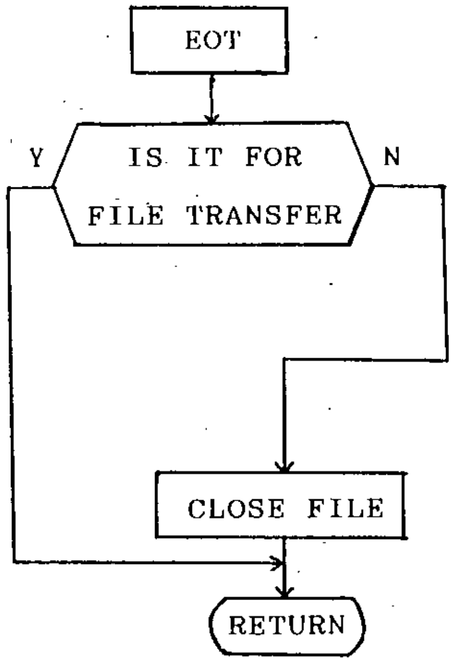


Fig. 4.14 Flow diagram of End of Text routine

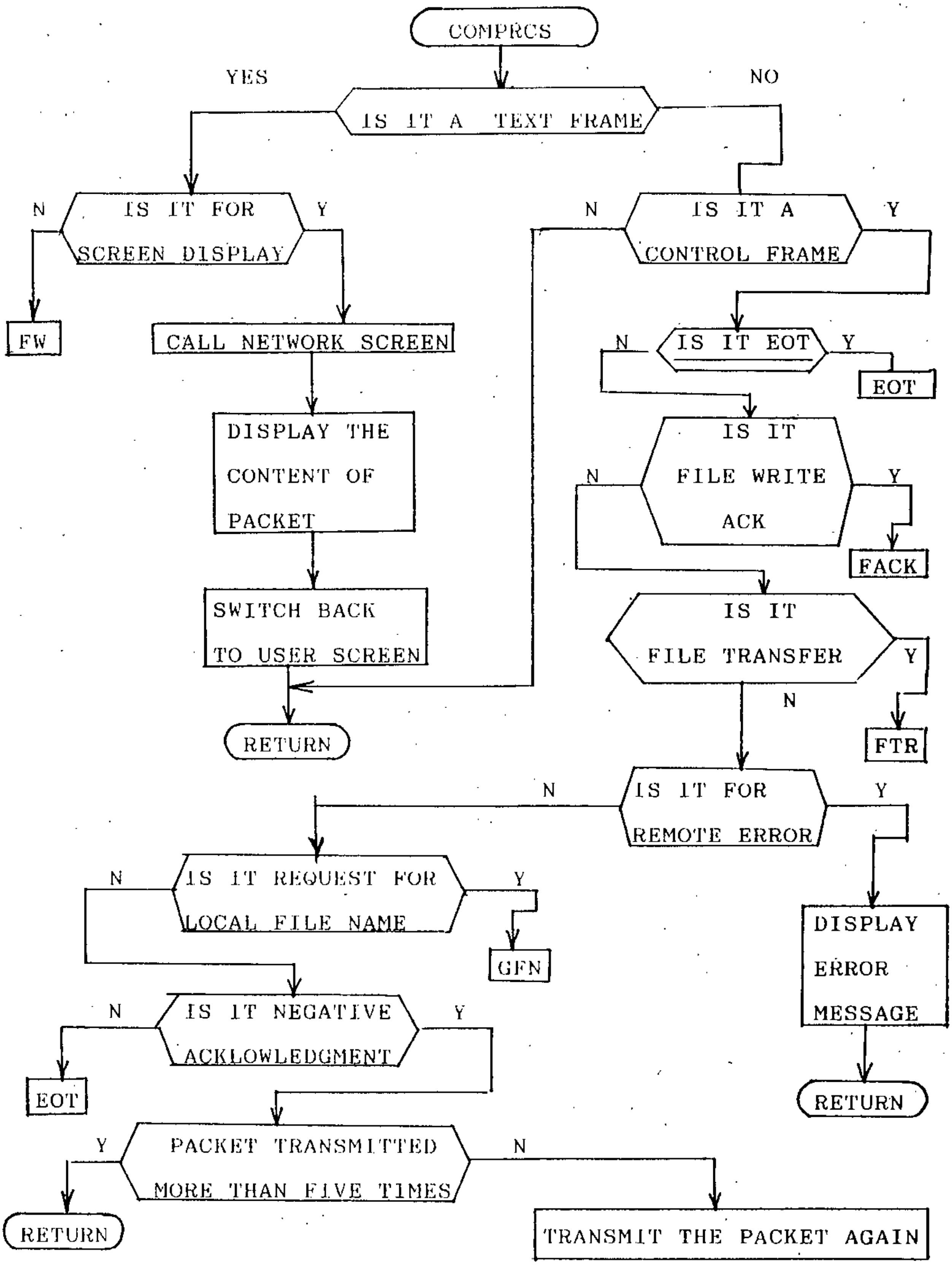


Fig. 4.15 Flow diagram of Command Processor routine

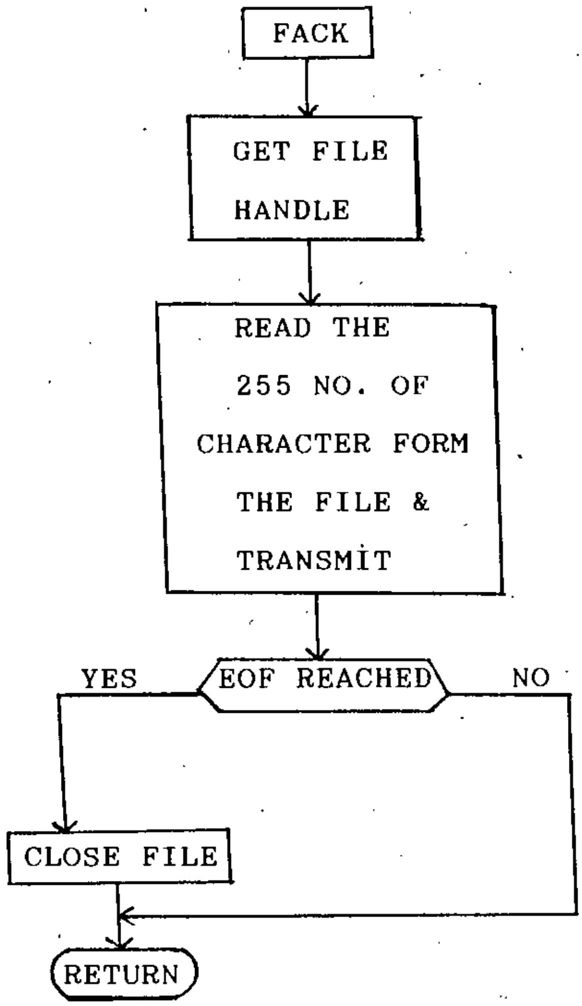


Fig. 4.16 Flow diagram of File Acknowledge routine

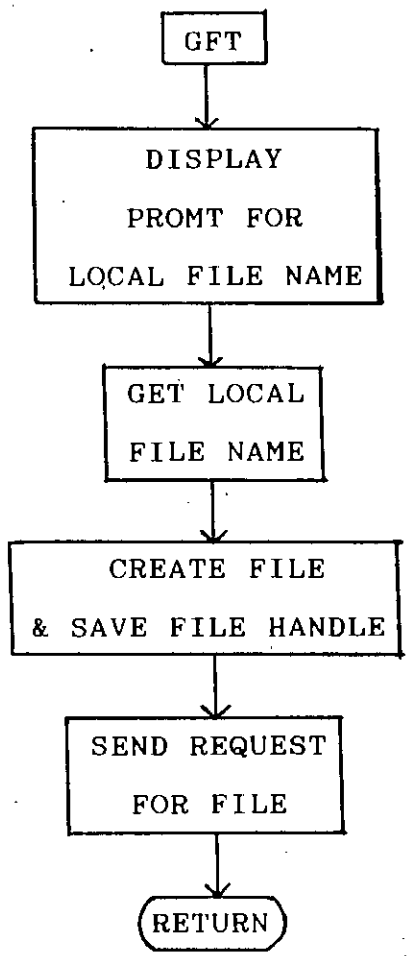


Fig. 4.17 Flow diagram of File Name getting routine

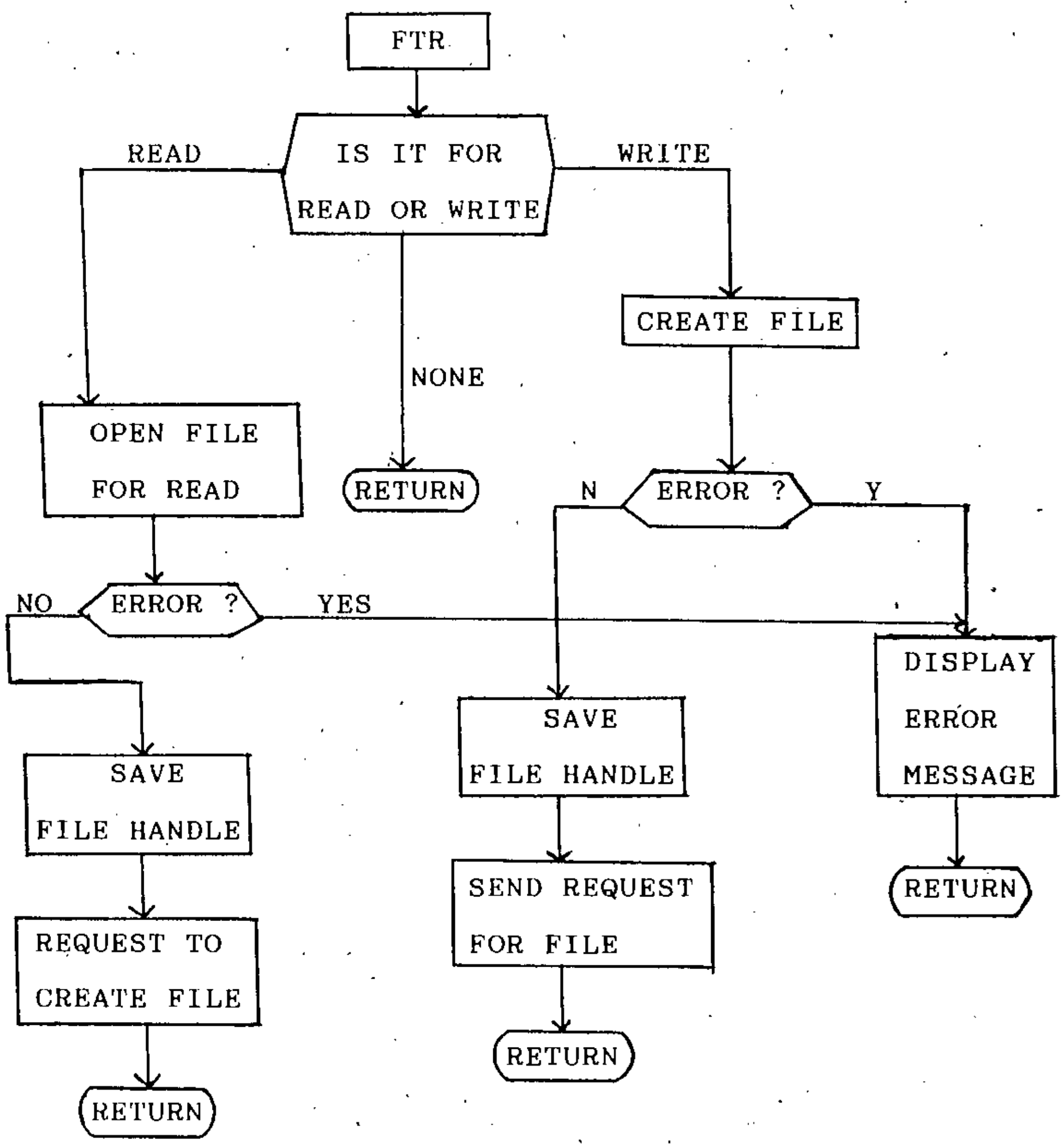


Fig. 4.18 Flow diagram of File Transfer routine

CHAPTER 5

PROTOCOL ANALYSIS

5.1 Introduction

The choice of a LAN architecture is based on many factors. But one of the most important factors is the performance of the protocol. Throughput and response time of a network under heavy load are often given special importance. Among the many factors which influences the efficiency of a protocol are whether frames are of fixed size or variable length, whether piggybacking is used or not, whether the protocol is pipelined or stop-and-wait, whether the line is half- or full-duplex, and the statistical characteristics of the transmission errors.

5.2 Analysis of the Protocol

For the analysis of the protocol, it is assumed that there are N active stations in the network and each station is always prepared to transmit a packet. This allows to develop an expression for maximum achievable throughput. The following notations will be used for the analysis of the protocol.

C = Channel Capacity in bps

A = Number of bits in Acknowledgment

D = Number of bits in Text frame

P_E = Probability of a bit in Error

H = Number of bits in the frame header

I = Interrupt service time and propagation delay

P_L = Probability that an Acknowledgment frame or a Text frame is damaged or lost

P_T = Probability that a Text frame is damaged or lost in a node

P_A = Probability that an Acknowledgment frame is damaged or lost in a node.

R = Mean number of retransmission of text frame

T = Timeout interval

U = Channel Utilization

It is assumed that the sender begins to send a frame at time 0.

Case I: Without considering the effects of errors.

Since the capacity of the channel is C bps, then the time taken to send a frame is (D+H)/C.

As the interrupt service time and propagation delay is I, the last bit arrives at the 2nd computer at (D+H)/C+I.

The last bit arrives at the 2- and 3-node apart stations at 2*((D+H)/C + I) and 3*((D+H)/C + I) respectively.

If the destination station is n-node apart from the transmitting station then the last bit arrives at the receiver at time n*((D+H)/C + I).

After receiving the text frame, the receiver will send the acknowledgment. Since there are N number of active stations on the network and since the bit-flow on the network loop is unidirectional hence the time at which the transmitter receives the last bit of acknowledgment is $n*((D+H)/C + I) + (N-n)*(A/C+I)$

The band width occupied by this frame is C multiplied by the time taken or

$$C * [n*((D+H)/C + I) + (N-n)*(A/C + I)]$$

$$= (D+H)*n + (N-n)*A + N*I*C$$

The number of bits actually transferred is D. So the channel efficiency, $U=D/[(D+H)*n+(N-n)*A+N*I*C]$

Average Channel Utilization: Since there are N number of active stations with equal priority and since the bit-flow on the network loop is unidirectional, hence the distance between the sender and receiver may be taken on average as N/2.

Hence on average the last bit arrives at the receiver at time $0.5*N*((D+H)/C + I)$ and the time at which the transmitter receives the last bit of acknowledgment is $0.5*N*[((D+H)/C + I) + (A/C + 1)]$.

So, Average Channel Utilization

$$U_{av} = D/[0.5*N*(D+H + A + 2*I*C)].$$

Since D+H+A >> 2*I+C

$$U_{av} = D/[0.5*N*(D+H+A)]$$
$$= 2/[N*(1+2*H/D)]$$

Case II: Considering the transmission errors.

If the frame is damaged or lost , the sender will time out T seconds after the last bit it has sent. Thus an unsuccessful transmission uses D+H+C*T bits worth of transmission capacity. If the mean number of transmissions per frame is R, then the total capacity used for the R bad frames and one good frame is

$$R*(D+H+C*T) + (D+H)*n + (N-n)*A + N*I$$

A frame is successfull if both the text frame and the acknowledgement frame are correctly received. The propability of success for a text frame is $(1-P_T)^n$. Similarly the probability of success for an ackowledgement frame is $(1-P_A)^{(N-n)}$.

Hence the probability of total success is $(1-P_T)^n * (1-P_A)^{(N-n)}$.
And the probability of failure. $L = 1 - (1-P_T)^n * (1-P_A)^{(N-n)}$.

The probability of that exactly k attemps are needed is $(1-L)*L^{(k-1)}$. So the expected number of transmission per frame is 1/(1-L) and an expected number of retransmissions, R = L/(1-L)

Hence Channel Utilization

$$U = D/[\ (L/(1-L))*(D+H+C*T) + (D+H)*n + (N-n)*A + N*I\ ]$$

where $T = N*((D+H)/C+I)$

Average Channel Utilization : In this case n becomes N/2. Hence the probability of failure, $L = 1-(1-P_T)^{N/2} * (1-P_A)^{N/2}$. Since A=H is used in the protocol and the probability that a bit is in error is P_E, so we obtain

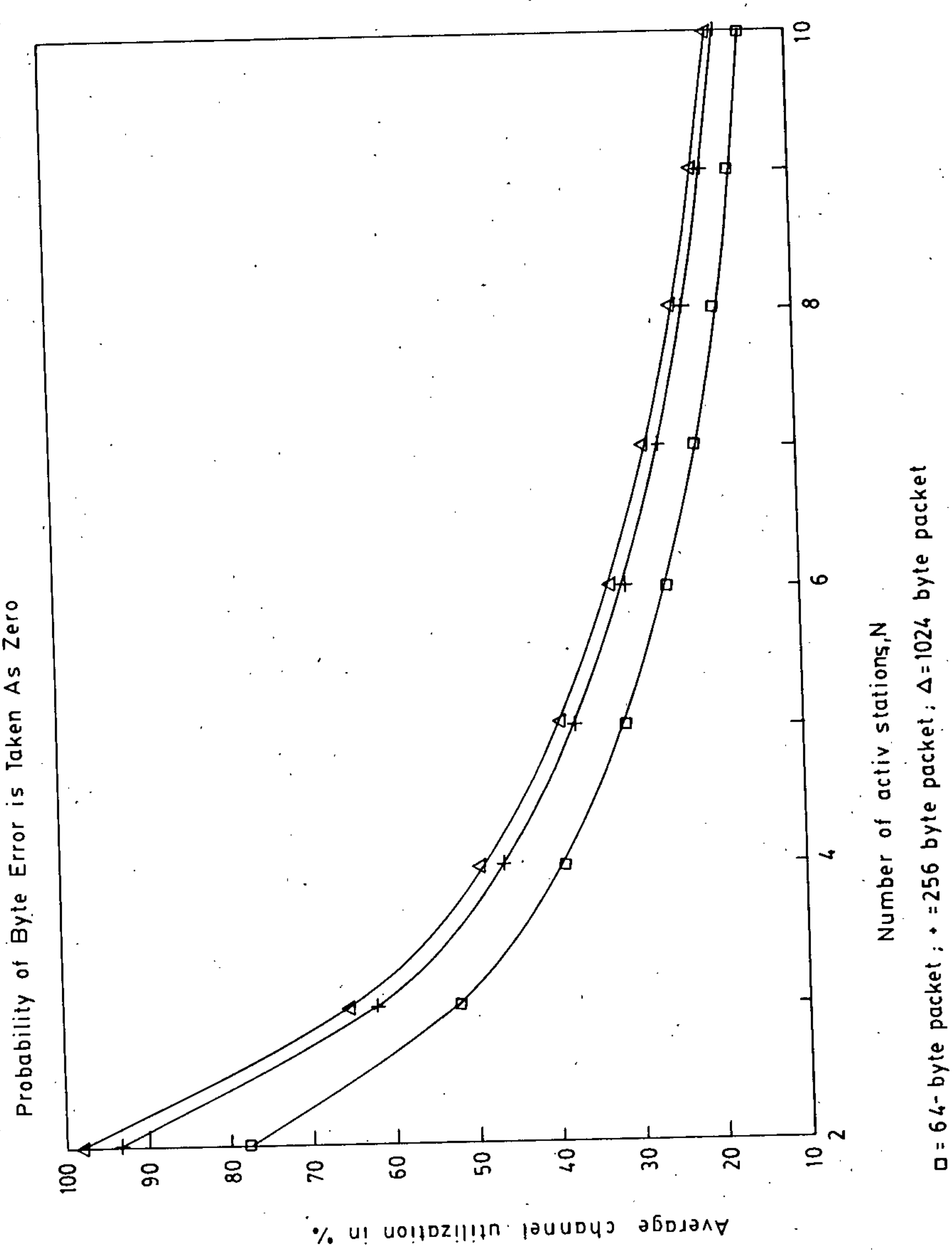
$$L = 1-(1-P_E)^{N*(H+D)/2}*(1-P_E)^{N*H/2}$$
$$= 1-(1-P_E)^{N*(2H+D)/2}$$

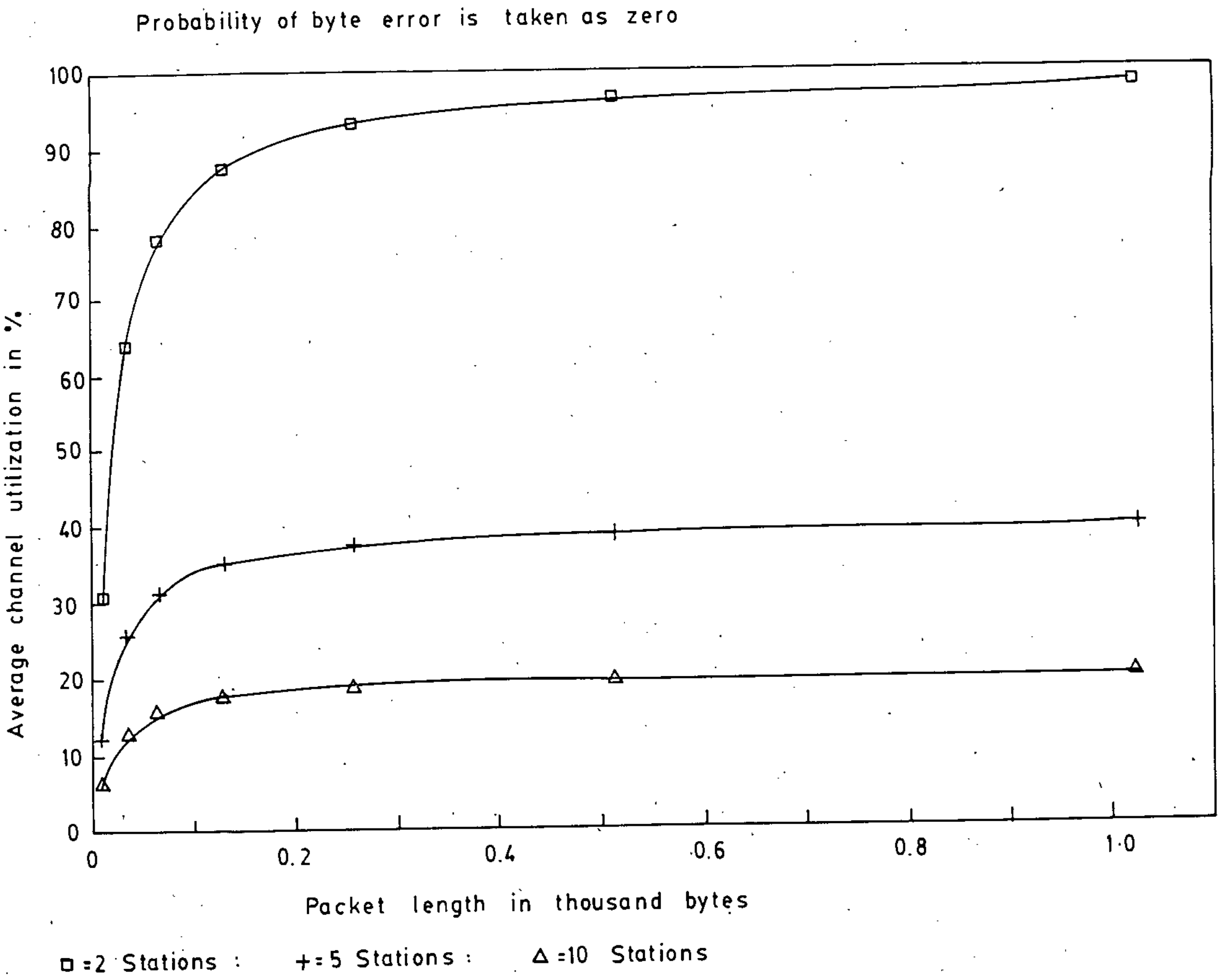
and $U_{av} = D/[(L/(1-L))*(D+H+C*T)+0.5*N*(D+H+A+2*I*C)$

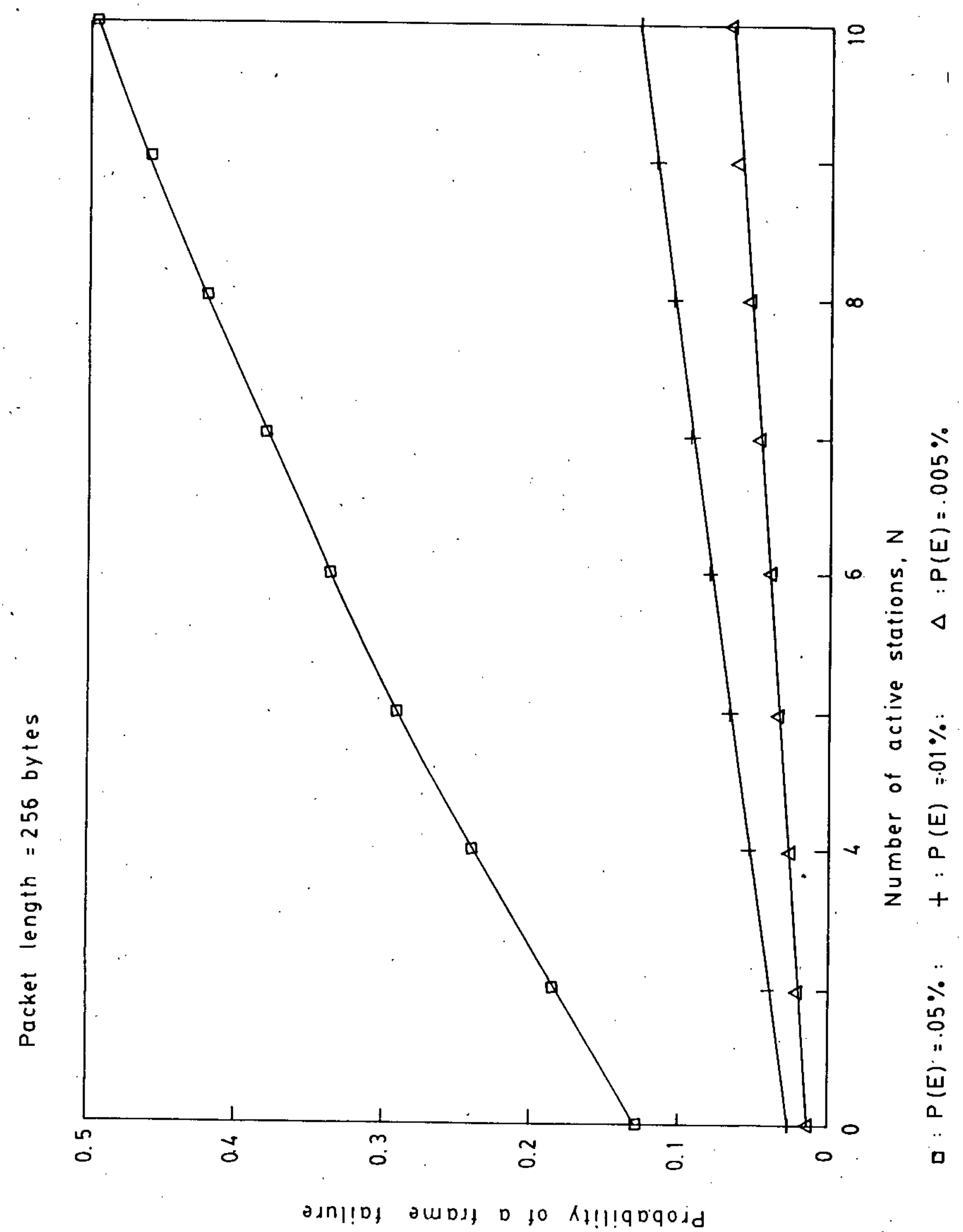
Since a serial card is used for communication between two nodes (i.e. channel capacity, C is low) and the time needed for the interrupt service is in the range of milisecond, we can ignore the terms containing the product of C and I. Now the above expression becomes

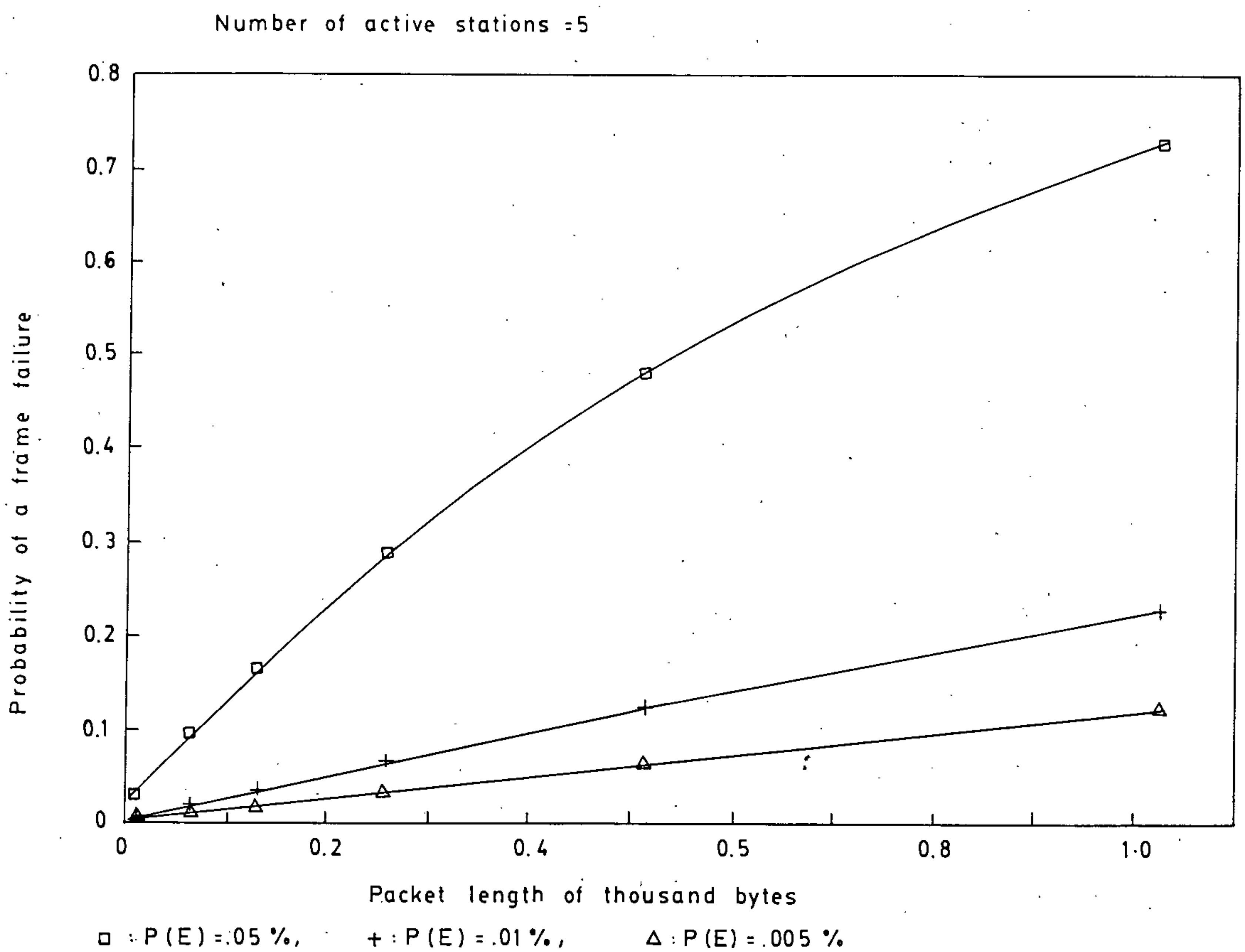
$$U_{av} = 1/[(L/(1-L))*(1+H/D) + 0.5*N*(1+2*H/D)]$$

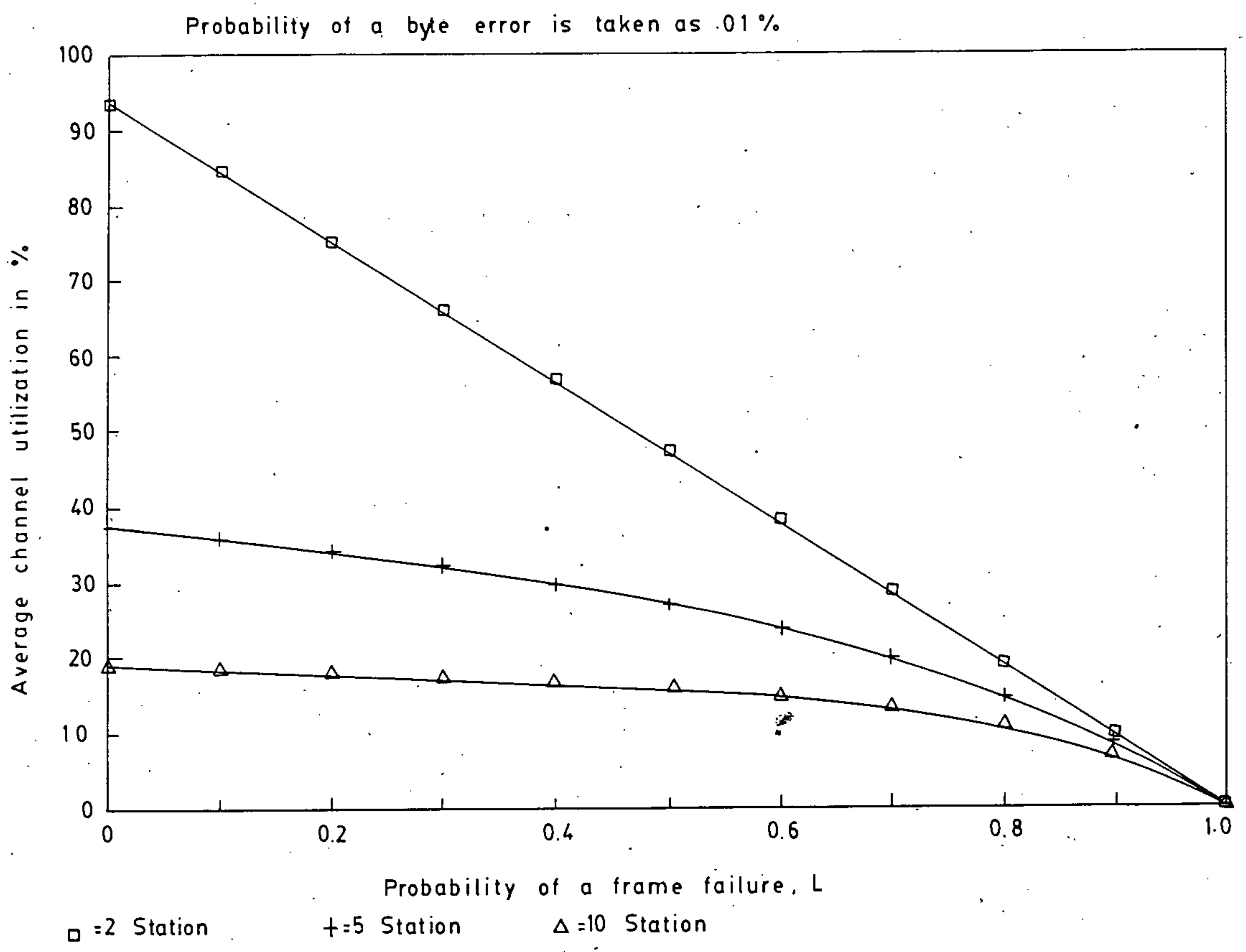
The followings are the graphical reprentation of different values.


Probability of Byte Error is Taken As Zero
Average channel utilization in %
100
90
80
70
60
50
40
30
20
10
2
4
6
8
10
Number of activ stations,N
□ = 64- byte packet ; + =256 byte packet ; Δ=1024 byte packet

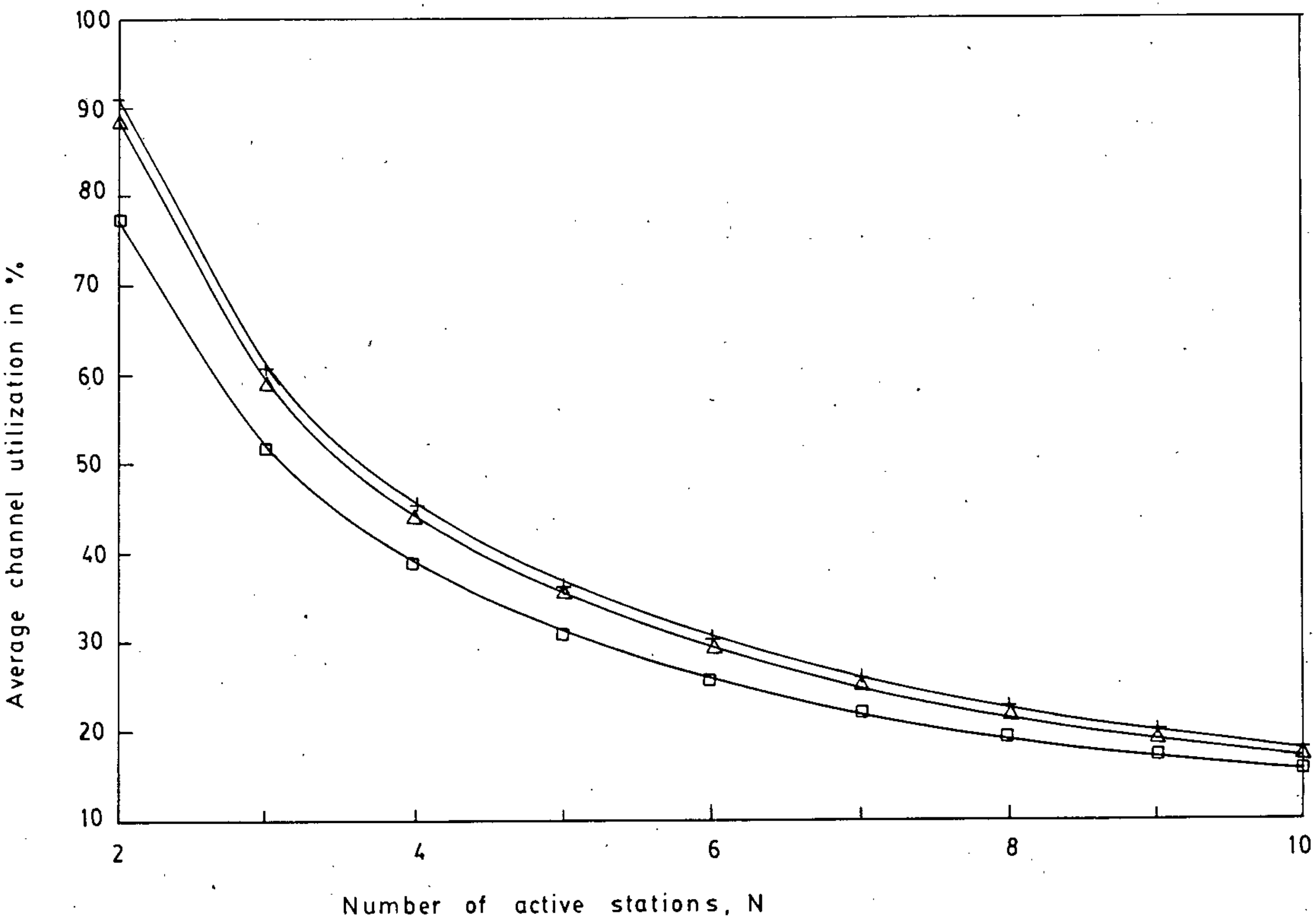

Probability of byte error is taken as zero
Average channel utilization in %
100
90
80
70
60
50
40
30
20
10
0
0.2
0.4
0.6
0.8
1.0
Packet length in thousand bytes
□ =2 Stations : + = 5 Stations : Δ =10 Stations


Packet length = 256 bytes
Probability of a frame failure
0
0.1
0.2
0.3
0.4
0.5
0
4
6
8
10
Number of active stations, N
□ : P(E) = .05% : + : P(E) = .01% : Δ : P(E) = .005%

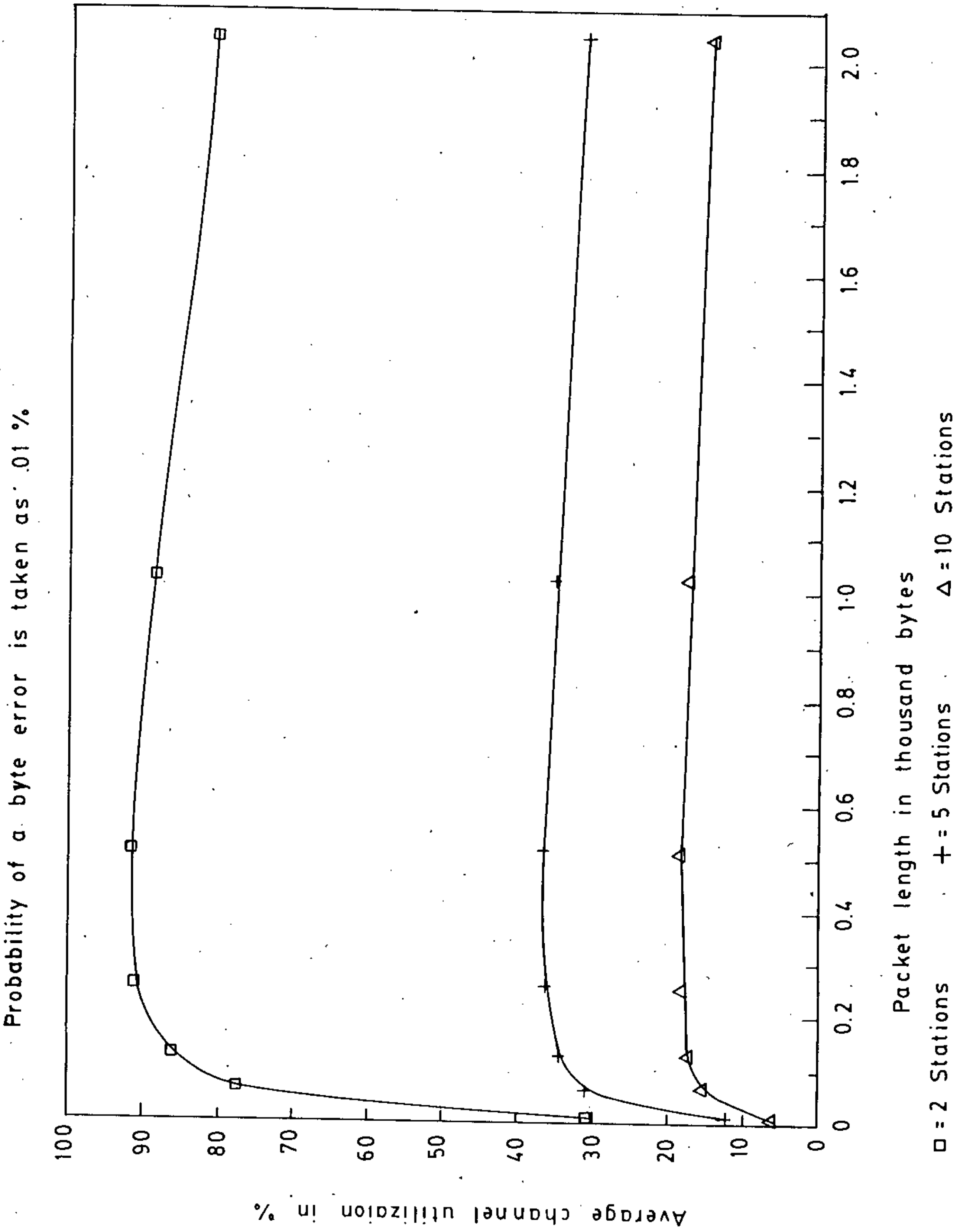

Number of active stations =5
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Probability of a frame failure
0
0.2
0.4
0.5
0.8
1.0
Packet length of thousand bytes
□ : P(E) = .05 %, + : P(E) = .01 %, Δ : P(E) = .005 %


Probability of a byte error is taken as .01 %
Average channel utilization in %
100
90
80
70
60
50
40
30
20
10
0
0
0.2
0.4
0.6
0.8
1.0
Probability of a frame failure, L
□ =2 Station
+ =5 Station
Δ =10 Station

Probability of a byte error is taken as .01 %



□ = 64-byte packet: + = 256-byte packet: Δ = 1024-byte packet



Probability of a byte error is taken as .01 %
Average channel utilizaion in %
100
90
80
70
60
50
40
30
20
10
0
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Packet length in thousand bytes
□ = 2 Stations
+ = 5 Stations
Δ = 10 Stations

CHAPTER 6.

CONCLUSIONS

6.1 CONCLUSIONS

The token ring local area network has been developed to provide information interchange among a group of microcomputers by using minimum hardware elements. The function of the interface of a computer to the network is supported by the asynchronous communication adapter or equivalent. This adapter is very much popular and may be even found to be fabricated on the mother board of the system unit of a microcomputer.

There is no need to load the communication software every time a user wishes to begin communication. The communication software is kept stay resident in the memory after the computer is booted. The loading module of the communication software does not allow multiple copies of the softwares in the memory as stay resident. It thus prevents the blocking of spaces in the memory by the stay resident programs. Communication starts by interrupting the current processing of the computer and the halted processing is restored at the end of communication. The baud rate, parity bit, number of start and stop bits can be changed by modifying the loading module slightly. The network software provides two screens- one is for user's normal local functions and the other is for the network functions.

The use of asynchronous communication adapter and the processor of the system as an interface to the token ring local area network reduces the cost of the hardware elements. The extension of the network is very simple and the required cost is simply additive on the excess elements involved. If a new computer with the asynchronous communication adapter is to be brought under the network, the hardware elements needed are a 25-pin connector and a 3-wire cable.

However, any type of communication between any two computers of the network requires the involvement of all the active computers. Thus, with the increase of the number of the microcomputers in the network, the performance of the normal processing of each computer will deteriorate. This kind of drawback is not, however, uncommon in other available contemporary network systems.

The main objective of the research was to develop a low cost network for digital data transmission among a group of computers within a limited geographical area. Now it can be stated that though the network software does not provide all the sophisticated features which are otherwise available in many contemporary systems, this network still has its own distinction for its low implementation cost, minimum extra hardware requirement, easy maintenance and simple procedure of installation and operation.

6.2 FUTURE SCOPE OF WORK

The networking technique may be the basis of implementation of other desirable network facilities like remote logging, interfacing with other network. A modem or a serial card may be used to interface the network with other networks.

Besides, the maximum length of packet size may be varied to determine its effect on the loop utilization and on the speed of data transfer on the network. Study of Mean transmission time, Mean waiting time, the maximum radius of ring loop may be other areas of future work.

```
;*****************************************************************************
;
; NETWORK SOFTWARE
; This software will load the interrupt routine
; into the computer memory and stay resident until
; a reboot.
; The software has two parts:
;  i) Transmit Routine
; ii) Receive Routine
;
;*****************************************************************************
;
;
CODE       SEGMENT    'CODE'
MAIN       PROC       FAR
ASSUME CS:CODE,DS:CODE,SS:CODE,ES:CODE
           JMP        LOAD
FLAG       DW         0AAAAH
MAIN       ENDP
;
INTR       PROC       FAR
           STI                     ;SET INTERRUPT FLAG
           PUSH       AX           ;SAVE REGISTERS
           PUSH       BX
           PUSH       CX
           PUSH       DX
           PUSH       SI
           PUSH       DI
           PUSH       DS
           PUSH       ES
           MOV        AX,CS
           MOV        DS,AX
           MOV        AL,20H
           OUT        20H,AL
           MOV        DX,3FEH      ;ADDRESS OF MSR
           IN         AL,DX        ;INPUT MODEM STATUS
           AND        AL,20H       ;TEST FOR DSR = 1
           JZ         IB0          ;OK+3/HEL ;NO THEN ITS A FALSE INTERRUPT
           MOV        SI,OFFSET INBUFR
           INT        62H          ;RECEIVE PACKET
           OR         BX,BX        ;CHECK FOR RECEIVER ERROR
           JZ         IB1          ;IF ERR THEN DISPLAY ERR MSG.
           MOV        SI,OFFSET INBUFR
           CALL       FCS          ;CHECK CRC OF RECEIVED PACKET
           OR         BX,BX
           JZ         OK
IB0:       JMP        IO           ;OK+3
OK:        CMP        DL,INBUFR+8
           JZ         OK1
           MOV        CX,WORD PTR INBUFR+3
           SUB        CX,1     ;9
```

```
        MOV     BX,WORD PTR INBUFR+3-2
        SUB     BX,2
        MOV     AX,0
        CMP     BYTE PTR INBUFR+1,1
        JZ      I6
        CMP     BYTE PTR INBUFR+1,5
        JNZ     I0
I7:     MOV     SI, OFFSET INBUFR+2
        ADD     SI, AX
        MOV     DL,[SI]
        MOV     SI, OFFSET CFRAME+2
        ADD     SI,AX
        MOV     [SI],DL
        INC     AX
        LOOP    I7
        MOV     SI, OFFSET CFRAME
        CALL    TRN
        JMP     I0
IB1:    JMP     I1
I6:     MOV     SI, OFFSET INBUFR+2
        ADD     SI,AX
        MOV     DL,[SI]
        MOV     SI, OFFSET OUTBUFR+2
        ADD     SI,AX
        MOV     [SI],DL
        INC     AX
        LOOP    I6
        MOV     SI, OFFSET OUTBUFR
        CALL    TRN
        JMP     I0
OK1:    MOV     DL, INBUFR+7
        MOV     SI, OFFSET DESTN
        MOV     [SI],DL
        CALL    COMPRCS     ;CALL COMMAND PROCESSOR
I0:     MOV     DX,3FEH     ;ADDR. OF MSR
        IN      AL,DX       ;READ IN [illegible] TO CLEAR 8250 INT PENDING
        POP     ES
        POP     DS
        POP     DI
        POP     SI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        IRET                    ;RETURN TO FOREGROUND JOB.
I1:     SUB     AL,AL
        MOV     DH,3
        INT     60H
        JMP     I0              ;OK+3
INTR    ENDP
;
```

```
KBDCHK  PROC    FAR
        STI                     ;ENABLE INTERUPT
        PUSH    AX              ;SAVE USER COMMAND
        MOV     AH,1            ;CHECK IF CHAR AVAILABLE
        INT     0BH
        JZ      NOCHR1          ;NO CHAR
        CMP     AX,1F00H        ;YES, THEN IS IT ALT-S
        JNZ     NOCHR1          ;NO, THEN DO AS USER WANTS
KC2:    PUSH    DS              ;SAVE REGISTERS
        PUSH    ES
        PUSH    SI
        PUSH    DI
        PUSH    DX
        PUSH    CX
        PUSH    BX
        SUB     AH,AH           ;REMOVE ALT-S FROM KBD BUFR
        INT     0BH
        MOV     AX,CS           ;SET DATA SEGMENT REG
        MOV     DS,AX
        MOV     AL,0AAH
        CALL    SCRN1           ;SWITCH TO SCREEN 1
;
        MOV     SI,OFFSET ADPRMT
        MOV     CX,17
        MOV     DX,102H
        INT     60H             ;display ADDRESS prompt
        MOV     SI,OFFSET DESTN
        MOV     DH,2
        INT     60H             ;INPUT DATA FROM KBD
        MOV     SI, OFFSET OUTBUFR
        MOV     DL, DESTN
        MOV     [SI+8],DL
        MOV     SI, OFFSET CFRAME
        MOV     [SI+8],DL
        MOV     DL,ADDR
        MOV     [SI+7],DL
        MOV     SI,OFFSET OUTBUFR
        MOV     [SI+7],DL
;
        INT     61H
        AND     AL,0DFH
        CMP     AL,'Q'
        JZ      EXIT
        CMP     AL,'S'
        JZ      FILESEND
        CMP     AL,'G'
        JZ      FILESEND-3
        CMP     AL,'M'
        JZ      SENDMSG
        CMP     AL,'C'
        JNZ     EXIT
```

```
          MOV      AL,12
          MOV      DH,4
          INT      60H
          JMP      EXIT
NOCHR1:   JMP      NOCHR
SENDMSG:  MOV      SI,OFFSET MESS       ;INBUFR-10
          MOV      CX,10
          MOV      DX,102H
          INT      60H          ;display message prompt
          MOV      SI,OFFSET OUTBUFR+9      ;7
          MOV      DH,2
          INT      60H          ;INPUT DATA FROM KBD
          SUB      AX,AX
          MOV      SI,OFFSET OUTBUFR
          MOV      WORD PTR [SI+3],AX
          MOV      WORD PTR [SI+5],0017H
          ADD      BX,9             ;7
          MOV      SI,OFFSET OUTBUFR
          CALL     TRN          ;AND THEN TRANSMIT IT
EXIT:     CALL     CHSCR        ;SWITCH BACK TO USER'S SCREEN
EXIT9:    POP      BX           ;RESTORE REGISTERS
          POP      CX
          POP      DX
          POP      DI
          POP      SI
          POP      ES
          POP      DS
NOCHR:    POP      AX           ;GET BACK TO USER'S COMMAND
          CMP      AH,I
          JZ       FILESEND-8
          INT      0BH
nn1:      IRET
          INT      0BH
          RET      2
          JMP      FILEGET

FILESEND:
          CMP      BYTE PTR TSTAT+1,0  ;CHECK FOR CHNL BUSY
          JNZ      BUSY
          MOV      SI,OFFSET SLOC         ;MSG+71
          MOV      CX,29                  ;1
          MOV      DX,102H
          INT      60H
          MOV      SI,OFFSET CFRAME+32   ;2;READ NAME OF FILE TO SEND
          MOV      DH,2
          INT      60H
          DEC      BX
          DEC      BX
          MOV      SI, OFFSET CFRAME+32
          MOV      BYTE PTR [SI+BX],0
          INC      BX
```

```
         MOV     DX,SI
         MOV     AX,3D00H   ;OPEN FILE FOR READING
         INT     21H
         JC      ERR
         MOV     WORD PTR HANDLE,AX     ;SAVE HANDLE
         MOV     SI,OFFSET SREM         ;MSG+92
         MOV     CX,29                  ;8
         MOV     DX,102H
         INT     60H
         MOV     SI,OFFSET CFRAME+9      ;6;READ REMOTE DEV/FILE NAME
         MOV     DH,2
         INT     60H
         DEC     BX
         DEC     BX
         MOV     SI, OFFSET CFRAME+9
         MOV     BYTE PTR [SI+BX],0
         INC     BX
         MOV     SI,OFFSET CFRAME
         MOV     BYTE PTR [SI+2],11H
         MOV     BYTE PTR [SI+5],01H
         MOV     BYTE PTR TSTAT+4,1      ;SEND FILE OPCODE
         ADD     BX,9                    ;7
         CALL    TRN
         JMP     EXIT9
BUSY:    MOV     SI,OFFSET MSG-14      ;DISPLAY CHNL  BUSY MSG.
         MOV     CX,14
         MOV     DX,102H
         INT     60H
         JMP     EXIT
ERR:     MOV     DH,3
         INT     60H
         JMP     EXIT

FILEGET: CMP     BYTE PTR TSTAT+1,0
         JNZ     BUSY
         MOV     SI,OFFSET GREM          ;INBUFR-65
         MOV     CX,29                   ;7
         MOV     DX,102H
         INT     60H
         MOV     SI,OFFSET CFRAME+9      ;6
         MOV     DH,2
         INT     60H
         DEC     BX
         DEC     BX
         MOV     SI, OFFSET CFRAME+9
         MOV     BYTE PTR [SI+BX],0
         INC     BX
         MOV     SI,OFFSET CFRAME
         MOV     BYTE PTR [SI+2],11H
         MOV     BYTE PTR [SI+5],0
         ADD     BX,9                ;7
```

```
        CALL      TRN
        JMP       EXIT9
KBDCHK  ENDP.
;
;
;////////////////////////////////////////////////////////////;
;          SUBROUTINE FOR TRANSMITTING DATA PACKET             ;
;    Input  : SI = Pointer to packet                           ;
;             BX = # of bytes to transmit                      ;
;    Output : None                                             ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
TRN     PROC      NEAR
;
;        PUSH     SI
;       MOV       SI,OFFSET PTRN
;        MOV      CX,5
;        MOV      DX,102H
;        INT      60H
;        POP      SI
;
        MOV       AX,BX       ;BYTE COUNT IN AX
        INC       AX          ;ADD 2 TO BYTE COUNT FOR FCS
        INC       AX
        PUSH      AX          ;SAVE BYTE COUN
        PUSH      SI          ;SAVE POINTER TO PACKET
        MOV       [SI+3],AX   ;BYTE COUNT IN PACKET
        CALL      FCS         ;CALCULATE FCS FOR THE PACKET
        MOV       [SI],BX     ;APPEND FCS TO THE PACKET
        POP       SI          ;RESTORE POINTER
        POP       BX          ;GET BYTE COUNT
;
        MOV       CX, 1000
TT9:    AND       AL,20H
        LOOP      TT9
        MOV       DX,3FCH     ;ADDRESS OF MCR
        MOV       AL,1        ;ACTIVATE DTR & DEACTIVATE OUT2
        OUT       DX,AL
        MOV       CX,4800
T1:     MOV       DX,3FEH     ;ADDRESS OF MSR
        IN        AL,DX       ;GET MODEM STATUS
        AND       AL,20H      ;ISOLATE DTR BIT
    ;    JNZ      T2          ;WAIT UNTIL DSR = 1
        LOOP      T1
T2:     MOV       CX,BX
        mov       bx,5000
tt4:    dec       bx
        cmp       bx,0
        jz        t5
        MOV       DX,3FDH     ;ADDRESS OF LSR
        IN        AL,DX       ;GET LINE STATUS
```

```
        AND     AL,20H      ;CHECK FOR THR EMPTY
        JnZ     tt3    ;T2+2       ;NO THEN WAIT
        jmp     tt4
tt3:    MOV     DX,3F8H     ;ADDRESS OF THR
        MOV     AL,[SI]     ;GET NEXT CHARACTER
        INC     SI          ;INCREMENT POINTER
        OUT     DX,AL       ;LOAD THR
        LOOP    T2+2
        MOV     CX,5000      ;100 GIVE TIME FOR THE LAST CHAR
T4:     MOV     DX,3FDH
        IN      AL,DX       ;GET LINE STATUS
        AND     AL,40H      ;CHECK TxSR EMPTY
        JZ      T5          ;YES THEN T4
        LOOP    T4          ;ELSE WAIT
T5:     MOV     DX,3FCH     ;ADDRESS OF MCR
        MOV     AL,8        ;DEACTIVATE DTR & ACTIVATE OUT2
        OUT     DX,AL
        RET                 ;RETURN TO CALLER
TRN     ENDP
;
SCRN1   PROC    NEAR
        PUSH    AX
        PUSH    CX
        PUSH    DI

        MOV     BX,0
        CMP     AL,0AAH
        JZ      D2
        MOV     BX,51
D2:     MOV     AX,501H
        INT     10H
        MOV     AX,0B900H
        MOV     ES,AX
        MOV     DI,3840
        MOV     SI,OFFSET MSG+20
        ADD     SI,BX
        MOV     CX,51
D1:     MOV     AH,14H      ;DISPLAY MENU
        MOV     AL,[SI]
        STOS    WORD PTR ES:[DI]
        INC     SI
        LOOP    D1
        POP     DI
        POP     CX
        POP     AX
        RET
SCRN1   ENDP

        MOV     AX,0B900H
        MOV     ES,AX
        MOV     DI,3840
```

```
         MOV       SI,OFFSET PMT2
         MOV       CX,51
C1:      MOV       AH,0ACH     ;DISPLAY 'PRESS ANY KEY TO RESUME'
         MOV       AL,[SI]
         STOS      WORD PTR ES:[DI]
         INC       SI
         LOOP      C1
         INT       61H         ;WAIT FOR ANY KEY PRESS
         MOV       AX,500H     ;SWITCH BACK TO USER SCREEN
         INT       10H
         RET                   ;Return
CHSCR    ENDP
;
;//////////////////////////////////////////////////////////////;
;      SUBROUTINE FOR EVALUATING CRC OF DATA PACKET          ;
;  Input : BX : no. of data bytes                            ;
;          SI : pointer to packet                            ;
;  Output: BX : 16 - bit CRC                                 ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
FCS      PROC   NEAR
         PUSH   CX
         PUSH   DX
         SUB    DX,DX          ;INITIALIZE CRC TO ZERO

CRC0:    MOV    AL,[SI]        ;GET NEXT DATA BYTE
         INC    SI             ;INCREMENT POINTER
         MOV    CX,8           ;INITIALIZE BIT COUNT
CRC1:    MOV    AH,AL
         XOR    AH,DL
         AND    DL,0FEH
         SHR    AL,1
         SHR    AH,1
         JNC    CRC2
         XOR    DX,4002H
         OR     DX,1
CRC2:    ROR    DX,1
         LOOP   CRC1
         DEC    BX
         JNZ    CRC0
         MOV    BX,DX
         POP    DX
         POP    CX
         RET
FCS      ENDP
;
ACKNLG   PROC      NEAR
;
         MOV       SI,OFFSET CFRAME
         MOV       [SI+2],AL
         MOV       AH,BYTE PTR INBUFR+2
         MOV       [SI+5],AH
```

```
          MOV       DL,ADDR
          MOV       [SI+7],DL
          MOV       DL, DESTN
          MOV       [SI+8],DL
          MOV       BX,9          ;6
          CALL      TRN
          RET
ACKNLG    ENDP
;
;////////////////////////////////////////////////////////////////;
;                     COMMAND PROCESSOR                           ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
COMPRCS   PROC      NEAR
;          INT       60H
;
          CMP       BYTE PTR INBUFR+1,1       ;IS IT SOH?
          JNZ       CF           ;NO THEN GO CHECK FOR ENQ
          MOV       CX,WORD PTR INBUFR+3  ;BYTE COUNT IN CX
          SUB       CX,11        ;9
          CMP       BYTE PTR INBUFR+6,0   ;PACKET FOR SCREEN DISPLAY
          JNZ       FW
          CALL      SCRN1
          MOV       SI,OFFSET INBUFR+9      ;7
          MOV       DX,10EH
          INT       60H
;          MOV       AL,6
;          CALL      ACKNLG
          CMP       BYTE PTR INBUFR+5,17H  ;IS IT LAST BLOCK
          JNZ       FW-1
          CALL      CHSCR        ;YES THEN SWITCH BACK TO USER SCRN
          RET
;fw:       mov       si,offset inbufr+7
;          mov       dx,10ch
;          int       60h
FW:       MOV       DL,INBUFR+7
          CMP       DL,DESTN
          JNZ       FW_RET
          MOV       BX,WORD PTR HANDLE+2  ;FILE WRITE
          MOV       DX,OFFSET INBUFR+9    ;7
          MOV       AH,40H
          INT       21H
          JC        DC4
          MOV       AL,6
          CALL      ACKNLG
          CMP       BYTE PTR INBUFR+5,17H
          JNZ       DC2-1
          MOV       BX,WORD PTR HANDLE+2
          MOV       AH,3EH
          INT       21H
          CALL      CHSCR
```

```
FW_RET:  RET
;
DC2:     MOV     SI,OFFSET CFRAME+62
         MOV     CX,14
         MOV     DX,102H
         INT     60H
         MOV     AL,BYTE PTR INBUFR+5
DC4:     MOV     DH,3
         INT     60H
         CALL    CHSCR
         RET
DC3:     MOV     AL,BYTE PTR INBUFR+6
         OR      AL,AL
         JNZ     FACK-6
         CALL    GFN
         JC      DC4
         MOV     WORD PTR HANDLE+2,AX
         MOV     AL,6
         CALL    ACKNLG
         RET
;
CF:      CMP     BYTE PTR INBUFR+1,5    ;IS IT ENQ
         JNZ     CF-1
         MOV     AL,BYTE PTR INBUFR+2
         CMP     AL,4        ;IS IT EOT ?
         JZ      EOT
         CMP     AL,6        ;IS IT ACK
         JZ      FACK
         CMP     AL,11H      ;IS IT DC1
         JZ      FACK-3
         CMP     AL,12H      ;IS IT DC2
         JZ      DC2
         CMP     AL,13H      ;IS IT DC3
         JZ      DC3
         CMP     AL,15H      ;IS IT NAK
         JNZ     EOT
;
         INC     BYTE PTR TSTAT+2    ;INCREMENT NAKCOUNT
         CMP     BYTE PTR TSTAT+2,5 ;MORE THAN 5 NAKS ?
         JZ      EOT
         MOV     BX,WORD PTR OUTBUFR+3
         DEC     BX
         DEC     BX
         CALL    TRN
         RET
EOT:     MOV     AL,12
         MOV     DH,4
         INT     60H
         CMP     BYTE PTR TSTAT+1,1
         JZ      CF8
         MOV     BX,WORD PTR HANDLE
         MOV     AH,3EH
```

```
        INT       21H
        MOV       BYTE PTR TSTAT+1,0
CF8:    MOV       BYTE PTR TSTAT,0
        RET
        JMP       SF
        JMP       DC1
FACK:   CMP       BYTE PTR TSTAT,1
        JNZ       CF9
        MOV       BYTE PTR TSTAT+2,0   ;RESET NAK COUNT
        MOV       BYTE PTR TSTAT,0     ;RESET NAK COUNT
        CMP       BYTE PTR TSTAT+3,1   ;MORE PACKETS TO SEND?
        JNZ       CF20     ;9          ;NO, THEN RETURN
SF:     MOV       DL,INBUFR+7
        CMP       DL,DESTN
        JNZ       CF9
        MOV       BX,WORD PTR HANDLE
        MOV       AH,3FH
        MOV       CX,255
        MOV       DX,OFFSET OUTBUFR+9    ;7
        INT       21H
        JC        FERR
        MOV       SI,OFFSET OUTBUFR
        MOV       BYTE PTR [SI+6],01
        CMP       AX,255
        JNE       CF9+1
        MOV       BYTE PTR TSTAT+3,1    ;SET MORE FLAG
        MOV       BYTE PTR [SI+5],1     ;SET WRITE CODE
CF10:   MOV       BX,AX
        MOV       DL,ADDR
        MOV       [SI+7],DL
        MOV       DL, DESTN
        MOV       [SI+8],DL
        ADD       BX,9           ;7
        CALL      TRN
        MOV       BYTE PTR TSTAT,1
CF9:    RET
        MOV       BYTE PTR TSTAT+3,0    ;RESET MORE FLAG
        MOV       BYTE PTR [SI+5],17H   ;SET EOTB FLAG
        JMP       CF10
CF20:   CALL      CHSCR
        RET
DC1:    CALL      SCRN1
        MOV       DX,OFFSET INBUFR+9        ;6
        MOV       AL,BYTE PTR INBUFR+5
        OR        AL,AL      ;IS IT OPEN FILE FOR READ
        JZ        OFR
        CMP       AL,1       ;IS IT OPEN FOR WRITE
        JNZ       CF9
        MOV       AH,3CH
        MOV       CX,20H
        INT       21H
```

```
         JC        FERR
         MOV       WORD PTR HANDLE+2,AX     ;file write handle
         MOV       SI,OFFSET CFRAME
         MOV       BYTE PTR [SI+6],1
         JMP       DREADY
FERR:    MOV       AH,AL
         MOV       AL,12H
         MOV       SI,OFFSET CFRAME
         MOV       [SI+2],AL
         MOV       [SI+5],AH
         MOV       DL,ADDR
         MOV       [SI+7],DL
         MOV       DL,DESTN
         MOV       [SI+8],DL
         MOV       BYTE PTR [SI+6],1     ;ERROR IN CH#
         MOV       BX,9                  ;7
         CALL      TRN
         RET
OFR:     MOV       AX,3D00H
         INT       21H
         JC        FERR
         MOV       WORD PTR HANDLE,AX    ;file read handle
         MOV       SI,OFFSET CFRAME
         MOV       BYTE PTR [SI+6],0
         MOV       BYTE PTR TSTAT+3,1    ;SET MORE FLAG
         MOV       BYTE PTR TSTAT,1      ;SET TRN STATUS FLAG
DREADY:  MOV       AX,0113H   ;DC3 & CH#
         MOV       [SI+2],AL
         MOV       [SI+5],AH
         MOV       DL,ADDR
         MOV       [SI+7],DL
         MOV       DL, DESTN
         MOV       [SI+8],DL
         MOV       BX,9        ;7        ;BYTE COUNT
         CALL      TRN
         RET
COMPRCS  ENDP
GFN      PROC      NEAR
         MOV       SI,OFFSET GLOC        ;INBUFR-38
         MOV       CX,29                 ;8
         MOV       DX,102H
         INT       60H
         MOV       SI,OFFSET CFRAME+32
         MOV       DH,2
         INT       60H
         DEC       BX
         DEC       BX
         MOV       BYTE PTR [BX+SI],0
         INC       BX
         MOV       DX,SI
         MOV       AH,3CH
```

```
         MOV       CX,20H
         INT       21H
         RET
GFN      ENDP
;
TSTAT    DB        0            ;TRANSMITTER STATUS
         DB        0            ;CHANNEL STATUS
         DB        0            ;NAK COUNT
         DB        0            ;MORE FLAG
         DB        0            ;OPN. CODE
HANDLE   DW        0
         DW        0
         DW        ?
         DB        'CHANNEL BUSY',13,10
MSG      DB        'RECEIVER NOT READY',13,10
         DB        'M=Message G=GetFile S=SendFile C=ClearScreen '
         DB        'Q=Quit'
         DB        'FILE TRANSFORMATION IS GOING ON.   PLEASE WAIT
SLOC     DB        'SEND FILE <local  file name>:'
SREM     DB        'TO <remote file/device name>:'
GREM     DB        'GET FILE <remote  file name>:'
GLOC     DB        'TO <local file/device  name>:'
MESS     DB        'MESSAGE : '
INBUFR   DB        267 DUP (?)          ;INPUT BUFFER
OUTBUFR  DB        22,1          ;SYN,SOH
         DB        5 DUP (?)     ;HEADER :SEQ #,COUNT(16),CH #,OPCODE
         DB        258 DUP (?)   ;TEXT
CFRAME   DB        22,5          ;SYN,ENQ
         DB        30 DUP (?)
         DB        30 DUP (?)
         DB        'REMOTE ERROR: '
PMT2     DB        'PRESS ANY KEY TO RESUME
RECEIVE  DB        'RECEIVER : '
INVAL    DB        'INVALID RECEINVER. TRY AGAIN'
PMT3     DB        'ADDRESSEE ABSENT      '
ADDR     DB        '0'
ADPRMT   DB        'REMOTE ADDRESS : '
DESTN    DB        10 DUP (?)
;
PINTR    DB        'INTR '
PTRN     DB        'TRN  '
PSCRN1   DB        'SCRN '
PCHSCR   DB        'CHSC '
PFCS     DB        'FCS  '
PACK     DB        'ACK  '
PCOMP    DB        'COMP '
;
LOAD     PROC      NEAR
         SUB       AX,AX
         MOV       DS,AX
         MOV       ES,AX
```

```
        MOV     SI,30H
        MOV     AX,OFFSET INTR
        CMP     [SI],AX
        JZ      L2
;
L1:     MOV     AX,00E3H  ;INITIALIZE
        MOV     DX,0000H
        INT     14H
        CLI
        MOV     SI,48
        MOV     AX,OFFSET INTR
        MOV     [SI],AX    ;SAVE OFFSET OF SERVICE ROUTINE
        MOV     AX,CS
        MOV     [SI+2],AX ;SAVE SEGMENT OF SERVICE ROUTINE
        MOV     SI,58H
        MOV     DI,2CH
        MOV     CX,4
        REP MOVSB
        MOV     AX,OFFSET KBDCHK
        MOV     SI,58H
        MOV     [SI],AX
        MOV     AX,CS
        MOV     [SI+2],AX
        MOV     AL,8
        MOV     DX,3F9H    ;ADDR OF IER
        OUT     DX,AL
        MOV     AL,08H
        MOV     DX,3FCH   ;ADDR. OF MCR
        OUT     DX,AL     ;ENABLE 8250 INTR. SIGNAL
        IN      AL,21H    ;READ 8259 IMR
        AND     AL,0EFH   ;ENABLE IRQ4
        OUT     21H,AL
        STI
        MOV     DX,OFFSET LOAD
        MOV     AX,3103H
        INT     21H
        DB      'PROGRAM IS ALREADY RESIDENT',7,13,10,'$'
L2:     MOV     AX,[SI+2]
        MOV     DS,AX
        MOV     SI,OFFSET FLAG
        CMP     WORD PTR [SI],0AAAAH
        JNZ     L1
        STI
        MOV     AX,CS
        MOV     DS,AX
        MOV     DX,OFFSET L2 - 31
        MOV     AH,09H
        INT     21H
        MOV     AH,4CH
        INT     21H
LOAD    ENDP
CODE    ENDS
```

```
CODE      SEGMENT   PUBLIC  'CODE'
MAIN      PROC      FAR
ASSUME CS:CODE,DS:CODE,SS:CODE,ES:CODE
          JMP       LOAD
FLAG      DW        0A25DH
MAIN      ENDP
;
;//////////////////////////////////////////////////////;
;       INT 60H FUNCTION DISPATCHER PROCEDURE          ;
;    INPUT  : DH = FUNCTION NO.                        ;
;              0 =  Get a character from kbd.          ;
;              1 =  Display string                     ;
;              2 =  Buffered keyboard entry            ;
;              3 =  Display error message              ;
;              4 =  Display character                  ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
DISTB     PROC      FAR
          STI
          OR        DH,DH
          JZ        GC         ;GET CHAR
          DEC       DH
          JZ        GC+3       ;DISPLAY STRING
          DEC       DH
          JZ        GC+6       ;BUFFERED KEYBOARD
          DEC       DH
          JZ        GC+9       ;ERROR MESSAGE
          DEC       DH
          JZ        DC         ;DISPLAY CHARACTER
          IRET
DC:       CALL      DISPCHAR
          IRET
GC:       JMP       NEAR PTR KBD
          JMP       NEAR PTR DISP
          JMP       NEAR PTR BUFKBD
          JMP       NEAR PTR DISPERR
DISTB     ENDP
;
;//////////////////////////////////////////////////////////;
;              PROCEDURE DISPLAY CHARACTER                 ;
;   INPUT  : AH = Attribute                                ;
;            AL = Character to display                     ;
;   OUTPUT : None                                          ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
DISPCHAR PROC       NEAR
          PUSH      AX
          PUSH      BX
          PUSH      CX
          PUSH      DX
          CMP       AL,13      ;IS IT CR ?
```

```
        JNZ     DC1
        MOV     AH,3
        MOV     BH,1
        INT     10H
        SUB     DL,DL
        JMP     DC6
DC1:    CMP     AL,10       ;IS IT LF ?
        JNZ     DC2
        MOV     AH,3
        MOV     BH,1
        INT     10H
        INC     DH
        CMP     DH,23
        JA      DC7
        JMP     DC6
DC2:    CMP     AL,8        ;IS IT BKSPC ?
        JNZ     DC3
        MOV     BH,1
        MOV     AH,3
        INT     10H
        OR      DL,DL
        JZ      DC6
        DEC     DL
        JMP     DC6
DC3:    CMP     AL,12       ;IS IT FF ?
        JNZ     DC4
        MOV     AX,600H
        MOV     DX,174FH
        SUB     CX,CX
        MOV     BH,7
        INT     10H
        SUB     DX,DX
        JMP     DC6
DC4:    MOV     BL,AH
        MOV     AH,9
        MOV     BH,1
        MOV     CX,1
        INT     10H
        MOV     AH,3
        MOV     BH,1
        INT     10H
        INC     DL
        CMP     DL,80       ;END OF LINE REACHED ?
        JNA     DC6
DC5:    INC     DH
        SUB     DL,DL
        CMP     DH,23       ;END OF PAGE REACHED?
        JA      DC7
DC6:    MOV     AH,2
        MOV     BH,1
        INT     10H
```

```
DC8:     POP      DX
         POP      CX
         POP      BX
         POP      AX
         RET
DC7:     MOV      AX,601H     ;SCROLL ONE LINE
         MOV      DX,174FH    ;LOWER RIGHT CORNER
         SUB      CX,CX       ;UPPER LEFT CORNER
         MOV      BH,7        ;FILL WITH BLANK
         INT      10H
         JMP      DC8
DISPCHAR ENDP
;
;////////////////////////////////////////////////////////;
;               PROCEDURE DISPLAY STRING                 ;
;   INPUT : SI = Pointer to string                       ;
;           CX = # of characters to display              ;
;           DL = Display Attribute                       ;
;   OUTPUT: None                                         ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
DISP     PROC     FAR
D0:      MOV      AL,[SI]
         MOV      AH,DL
         CALL     DISPCHAR
         INC      SI
         LOOP     D0
         IRET
DISP     ENDP
;
;///////////////////////////////////////////////////////;
;           READ A CHARACTER FROM KEYBOARD              ;
;  INPUT  : None                                        ;
;  OUTPUT : AH = Scan code                              ;
;           AL = Character code                         ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
KBD      PROC     FAR
         STI
K1:      MOV      AH,1        ;CHECK KBD STATUS
         INT      0BH
         JZ       K1          ;WAIT UNTIL KEY PRESSED
         SUB      AH,AH       ;GET KEY PRESSED
         INT      0BH
         IRET
KBD      ENDP
;
;/////////////////////////////////////////////////////;
;              BUFFERED KEYBOARD ENTRY                ;
;   Input  : SI = Pointer to buffer                   ;
;   Output : BX = No. of characters entered           ;
```

```
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
BUFKBD   PROC     FAR
         STI
         SUB      BX,BX        ;INITIALIZE CHAR COUNT
BK0:     INT      61H          ;READ A CHAR FROM KBD
         MOV      AH,3         ;VIDEO ATTRIBUTE IN AH
         CALL     DISPCHAR     ;DISPLAY THE CHAR
         CMP      AL,13        ;IS IT CR
         JZ       BK2
         CMP      AL,8         ;IS IT BKSPC ?
         JZ       BK1
         MOV      [SI+BX],AL     ;STORE THE CHAR
         INC      BX           ;INCREMENT CHAR COUNT
         CMP      BX,254
         JA       BK2
         JMP      BK0          ;GO GET NEXT CHAR
BK1:     OR       BX,BX        ;IS BX = 0 ?
         JZ       BK0          ;YES THEN IGNORE BKSPC
         DEC      BX           ;DECREMENT CHAR COUNT
         JMP      BK0
BK2:     MOV      WORD PTR [SI+BX],0D0AH
         MOV      AL,0AH
         CALL     DISPCHAR
         INC      BX
         INC      BX
         IRET
BUFKBD   ENDP
;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;                 DISPLAY ERROR MESSAGE                     ;
;    INPUT :  AL = ERROR CODE                               ;
;////////////////////////////////////////////////////////////;
;
DISPERR  PROC     NEAR
         PUSH     DS
         MOV      CX,CS
         MOV      DS,CX
         MOV      BL,19
         MUL      BL
         MOV      SI,OFFSET MSG
         ADD      SI,AX
         MOV      CX,19
         MOV      DX,104H
         INT      60H
         MOV      SI,OFFSET MSG-2
         MOV      CX,2
         MOV      DH,1
         INT      60H
         POP      DS
         IRET
```

```
DISPERR  ENDP
;
;//////////////////////////////////////////////////////;
;    PROCEDURE  TO RECEIVE PACKET FROM COMMN. CHANNEL    ;
;   Input : SI = Pointer to buffer                       ;
;   Output: BX = 0   if error                            ;
;              = no. of data bytes received otherwise    ;
;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\;
;
REC        PROC      FAR
;
;           PUSH      SI
;           MOV       SI,OFFSET PREC
;           MOV       CX,5
;           MOV       DX,102H
;           INT       60H
;           POP       SI
;
           SUB       BX,BX        ;INTIALIZE CHAR COUNT TO 0
;           MOV       DX,3FCH      ;ADDRESS OF MCR
;           MOV       AL,1         ;ACTIVATE DTR & DEACTIVATE OUT2
;           OUT       DX,AL
R1:        CALL      DATARDY      ;WAIT FOR RBR FULL
           OR        CX,CX        ;CHECK FOR DEVICE TIMEOUT
           JNZ       R2           ;NO THEN GO GET CHAR
           SUB       BX,BX        ;SET ERROR FLAG
           JMP       R3           ;RETURN
;
R2:        MOV       DX,3F8H               ;ADDRESS OF RBR
           IN        AL,DX                 ;GET RECEIVED CHAR. IN AL
           MOV       [SI],AL               ;STORE CHAR. IN BUFFER
           INC       SI                    ;INCREMENT POINTER
           INC       BX                    ;INCREMENT CHAR. COUNT
           MOV       DX,3FEH               ;ADDRESS OF MSR
           IN        AL,DX                 ;GET MODEM STATUS
           AND       AL,20H                ;TEST FOR DSR = 1
           JNZ       R1                    ;YES, THEN GET NEXT CHAR.
;
           CALL      DATARDY      ;WAIT FOR LAST CHAR
           OR        CX,CX
           JZ        R3
           MOV       DX,3F8H
           IN        AL,DX
           MOV       [SI],AL
           INC       BX
;
R3:        MOV       DX,3FCH      ;ADDRESS OF MCR
           MOV       AL,8         ;DECTIVATE DTR & ACTIVATE OUT2
           OUT       DX,AL
           IRET
REC        ENDP
```

```
;
DATARDY  PROC     NEAR
         MOV      CX,12000    ;SET TRY COUNT FOR DATA READY
DR1:     MOV      DX,3FDH     ;ADDRESS OF LSR
         IN       AL,DX       ;GET LINE STATUS
         SHR      AL,1        ;DATA READY BIT IN CF
         JC       DR2         ;GO AND READ CHAR IF DR=1
         LOOP     DR1
DR2:     RET
DATARDY  ENDP
;
;
         DB       10,13
MSG      DB       'DEVICE TIMEOUT      '
         DB       'INVALID FUNCTION    '
         DB       'FILE NOT FOUND      '
         DB       'PATH NOT FOUND      '
         DB       'TOO MANY OPEN FILES'
         DB       'ACCESS DENIED       '
         DB       'INVALID HANDLE      '
PREC     DB       'REC  '
;
LOAD     PROC     NEAR
         SUB      AX,AX
         MOV      DS,AX
         MOV      ES,AX
         MOV      SI,180H
         MOV      AX,OFFSET DISTB
         CMP      [SI],AX
         JZ       L2
;
L1:      CLI
         MOV      SI,180H
         MOV      AX,OFFSET DISTB
         MOV      [SI],AX         ;SAVE OFFSET OF DISTB ROUTINE
         MOV      AX,CS
         MOV      [SI+2],AX       ;SAVE SEGMENT OF DISP ROUTINE
         MOV      [SI+6],AX
         MOV      [SI+10],AX
         MOV      AX,OFFSET KBD
         MOV      [SI+4],AX
         MOV      AX,OFFSET REC
         MOV      [SI+8],AX
         STI
         MOV      DX,OFFSET LOAD
         MOV      AX,3103H
         INT      21H
MSG1     DB       'PROGRAM IS ALREADY RESIDENT',7,13,10,'$'
L2:      MOV      AX,[SI+2]
         MOV      DS,AX
         MOV      SI,OFFSET FLAG
```

```
        CMP     WORD PTR [SI],0A25DH
        JNZ     L1
        STI
        MOV     AX,CS
        MOV     DS,AX
        MOV     DX,OFFSET MSG1
        MOV     AH,9
        INT     21H
        MOV     AH,4CH
        INT     21H
LOAD    ENDP
CODE    ENDS
        END
```

REFERENCES

1. Tanenbaum, A.S. : 'Computer Network', Prentice Hall Inc., 1981.

2. K. C. E. Gee : Introduction to Local Area Computer Networks, John Wiley & Sons, NY, 1983.

3. Murshed, Mainuddin Md. Kamrul : 'Design of a Star Type Local Area Network (LAN) Interface Utilizing The Exiting Telecommunication System in BUET Campus', M. Sc. Engg. (Computer) Thesis, BUET.

4. Welch, Samuel : 'Signalling in Telecommunications Networks', Peter Peregrinus Ltd., 1979.

5. Chorafas, Dimitries N. : 'Data Communications for Distributed Information Systems', Petrocelli Books, Inc., 1980.

6. Rahman, Hakikur : 'Design and Implementation of a TOKEN Ring Local Area Network Interface', M.E. Thesis, American University of Beirut, 1983.

7. Postel, Jonathan B. : 'Internework Protocol Approaches'.IEE Local Area Network Interface', M.E. Thesis, April 1980.

8. Zimmetmann, Hubert : 'OSI Reference Model- The ISO Model Of Architecture for Open Systems Interconnections', IEEE Transactions on Communications, Vol. COM-28, No.4, April 1980.

9. Utter, Donald F., : 'Writting Requirements for Inter faces between Computers', IEE Transctions on Communications, Vol. COM-30, No.6, June 1982.

10. Danthine, Andre A.S. : 'Protocol Representation with Finite-State Models', IEEE Transactions on Communication, Vol. No. 4, April 1980.

11. Intel Corporation : 'Component Data Catalog'.

12. Stallings, W. : 'Computer Networks and Data Communication'.