ABSTRACT

SIMULATION ANALYSIS OF FDDI NETWORKS USING NETWORK II.5 SOFTWARE PACKAGE

by

Anastassis A. Lazarides

In recent years, one of the most exciting advances in media has been the use of fiber optics in LANs. The bandwidth provided by Fiber Optic Technology has drastically increased the number of new applications that can be supported by communication networks. In order to support a variety of services, in 1986, the American National Standard Institute (ANSI) Accredited Standards Committee (ASC) X3, and the ASC X3T9.5 Task Group developed a new standard; the Fiber Distributed Data Interface (FDDI) [1]. This is a high speed (100 Mbps) optical communication network based on a token passing mode of operation. The Medium Access Control (MAC) Protocol selected for this network attempts to provide priority services, as well as bounded delay transmission for real time applications [2].

This thesis presents results for the Voice-Data performance of the Medium Access Control (MAC) protocol, selected for the FDDI network, using the NETWORK II.5 [6] software package. This protocol can provide priority services to different types of traffic, as well as guarantee bounded delays for real-time applications. The effect of various system parameters on performance is investigated.

SIMULATION ANALYSIS OF FDDI NETWORK USING NETWORK II.5 SOFTWARE PACKAGE

by
Anastassis A. Lazarides

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

May 1994

APPROVAL PAGE

SIMULATION ANALYSIS OF THE FDDI NETWORK USING NETWORK II.5 SOFTWARE PACKAGE

Anastassis A. Lazarides

Dr. Mengchu Zhou, Thesis Advisor Date
Assistant Professor of the Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member Date
Assistant Professor of the Electrical and Computer Engineering, NJIT

Dr. David Wang, Committee Member Date
Assistant Professor of the Computer and Information Science, NJIT

BIOGRAPHICAL SKETCH

Author: Anastassis A. Lazarides

Degree: Master of Science in Electrical Engineering

Date: May 1994

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1994
- Bachelor of Engineering in Electrical Engineering
  City College of the City University of New York, New York, NY, 1992

Major: Electrical Engineering

This Work is dedicated to:
the memory of my mother Dora,
my father Andreas,
my sister Pambina and
my aunt Dia.

ACKNOWLEDGMENT

The author wishes to express his sincere graditute to his advisor, Professor Mengchu Zhou, for his guidance, friendship, and moral support throughout this paper. Special thanks to Professors Edwin Hou, and David Wang for serving as members of the committee.

The use of NETWORK II.5 Software Package developed by CACI Products Company is also acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Evolution of FDDI under integrated data and voice transmissions

The combination of office automation with voice/data integration would appear to be a very rational way to reduce administrative expenses. Unfortunately, these two types of traffic have different characteristics and require different types of service from a network. In particular, voice is stream traffic that has a long holding time and requires real time delivery, while data is bursty traffic that can tolerate long delays. Also, voice is tolerant of errors, while data requires reliable delivery. The following problems must be addressed to have efficient data/voice integration:

- Choice of the local area network.
- Good transmission protocol
- Good flow control scheme.

The flow control scheme must also permit the data traffic to use as much of the network capacity as possible when the voice traffic is low.

Telephone transmission equipment is designed to pass the spectrum from 300 Hz to 3400 Hz [26]. The telephone network is optimized for analog voice transmission. The telephone local loop from the central office to subscribers is usually composed of 24-gauge 2 wire twisted pair wires. At voice band frequencies these wires attenuate the signals less than 6 dB per Km. Data signals have much bandwidths and the attenuation at these frequencies is more than 20 dB per Km. Telephone lines are often interconnected at two or three points between the central office and the subscriber's location. This is acceptable with analog signals but digital signals, especially at high transmission rates, have a tendency to reflect at each interconnection point. While the control of signal reflections on the voice band is needed only on very long distance lines, the control of signal reflection over the greater bandwidth of data is much more complex and be required for lines of only a few hundred feet. On the other hand, computer network configurations are optimized for digital signals. The interconnection of almost all peripherals to the main processor is over short, high quality, multiwire cables. The DC voltages carried over the wires are less than 15V and AC is never utilized. This enables computers and terminals to use higher speed and less expensive components at each end of the line. In contrast, telephone lines must carry analog signals that have higher supply voltages in the range of 48V DC.

The main advantages of the integrated transmission of voice and data are:

- One set of wiring can be used for data as well as voice communication.
- Simplified communication facilities management.
- A single integrated user interface to most functions required in business today.
- Use of shared resources such as modems, printers and disks.
- The combination of voice functions such as voice mail with similar data functions such as electronic mail.

The Fiber Distributed Data Interface (FDDI) is an ANSI proposal for a 100Mb/s fiber-token ring. Originally developed as a packet switching network, FDDI has been enhanced to have the real time transmission necessary to support voice signals as in a circuit switching network. FDDI also has the potential for enhancements to satisfy the diverse needs of real time voice, video, and sensor data streams.

A token passing ring configuration is used to implement the FDDI protocol. Various studies on its performance have established its merit for a variety of data transfer applications. Token passing networks are less sensitive to changes in network loads, offer reasonably short transmission delays under light loads, and carry more traffic under heavy loads than other network configurations.

1.2 Voice transmission protocol

Voice calls last for a few minutes, and for more than 60% of the conversation time the channel remains idle. This results because only one speaker is active at any particular time. Furthermore, there are pauses between sentences, phrases, and even between syllables. Voice traffic can be modeled as having alternating talk spots and silences, Fig. 1-1, with the generation of voice packets at a constant rate during talk spots and no packet generation during silences. To preserve the integrity of a conversation voice packets must be delivered within some time bound (typically 150-200 msec). In many data applications, a delay as much as 200 msec does not present a problem. Conversation is inherently robust, and as result, speech can be reconstructed at the destination user with acceptable quality, provided that the voice packet loss is less than some specified fraction (typically 2%).

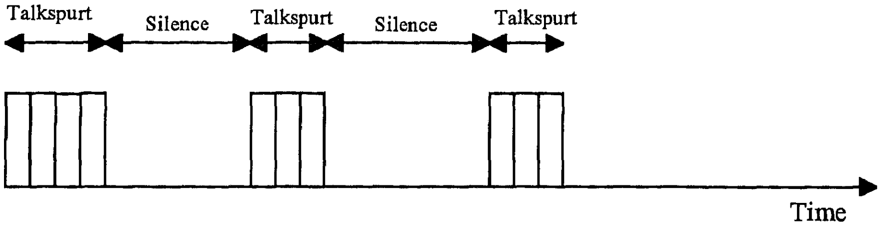


Figure 1-1. Arrival process of voice packets.

Now let us review the stages of the voice packet processing as shown in Fig. 1-2. First the analog signal, S(t), generated when a speaker is in the talk spot mode, is digitized by sampling at 8000 samples per second. The A/D converter, Codec, makes 8000 samples per second (125 μsec/sample) because the Nyquist theorem says that this is sufficient to capture all the information from a 4-KHz bandwidth. Then it is quantized, typically at 8 bits per sample, to produce a PCM (Pulse Code Modulation) bit stream. A voice encoder operating at V bits/sec combines the bits from several samples, after some possible data compression to produce a packet of length L_v bits. Thus, every $B=L_v/V$ sec the voice encoder outputs a voice packet. A header of length L_H bits is added to each packet before it is transmitted. Since an access delay is usually encountered before the packet can be transmitted, it must be temporarily buffered until it can be transmitted. Voice packets are generated at the rate of one packet per B seconds (packetization interval).

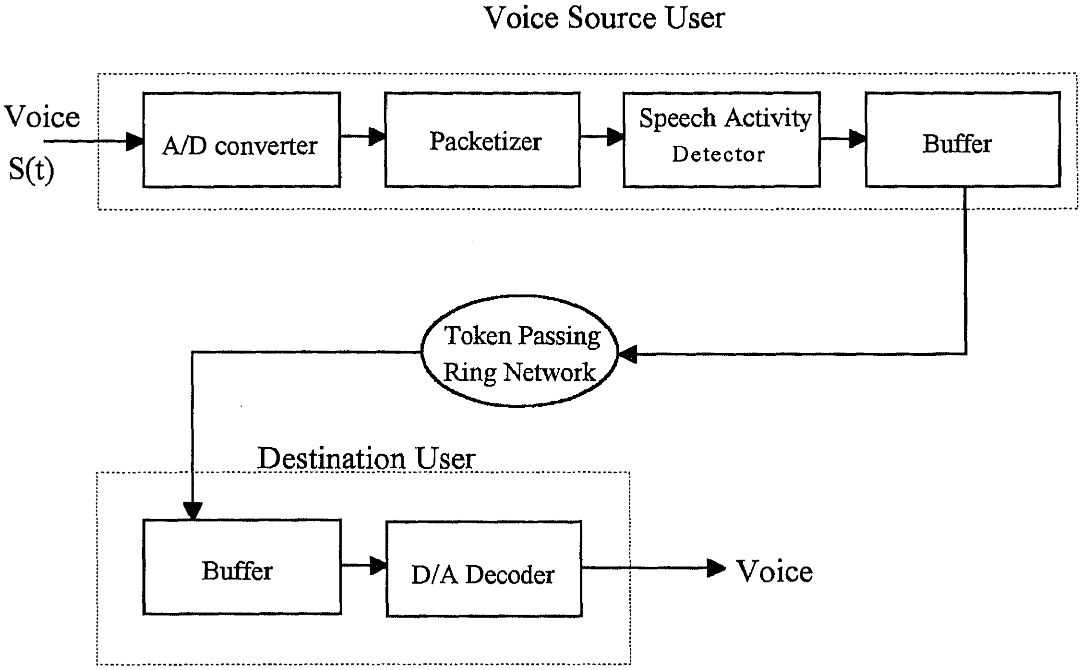


Figure 1-2. Voice packet processing.

1.3 Data transmission protocol

The traffic characteristics and performance requirements for voice and data are quite different [27]. Data traffic can be categorized into two basic types: interactive and bulk data. Interactive data traffic is bursty in nature; the actual proportion of bandwidth utilized is typically very small. It consists of short messages incurring small network delays. Bulk data, on the other hand, consists of long messages and requires high throughput, but real time data delivery is not of primary importance. Strict error control and recovery procedures are required for both types of data. Figure 1-3 shows the typical implementation of a LAN.

Now consider an integrated voice and data token ring with $N=N_V+N_D$ station, where N_V is the total number of voice stations and N_D is the total number of data stations. We assume that no data flow control is practiced; therefore, any data station that has a packet to transmit will do so upon acquisition of the token. Since we want the voice packets to be transmitted in as synchronous manner as possible, it is necessary to ensure that the token cycle durations are within the limits required for voice transmission. A voice station when active alternate randomly between talk spot and silence modes.

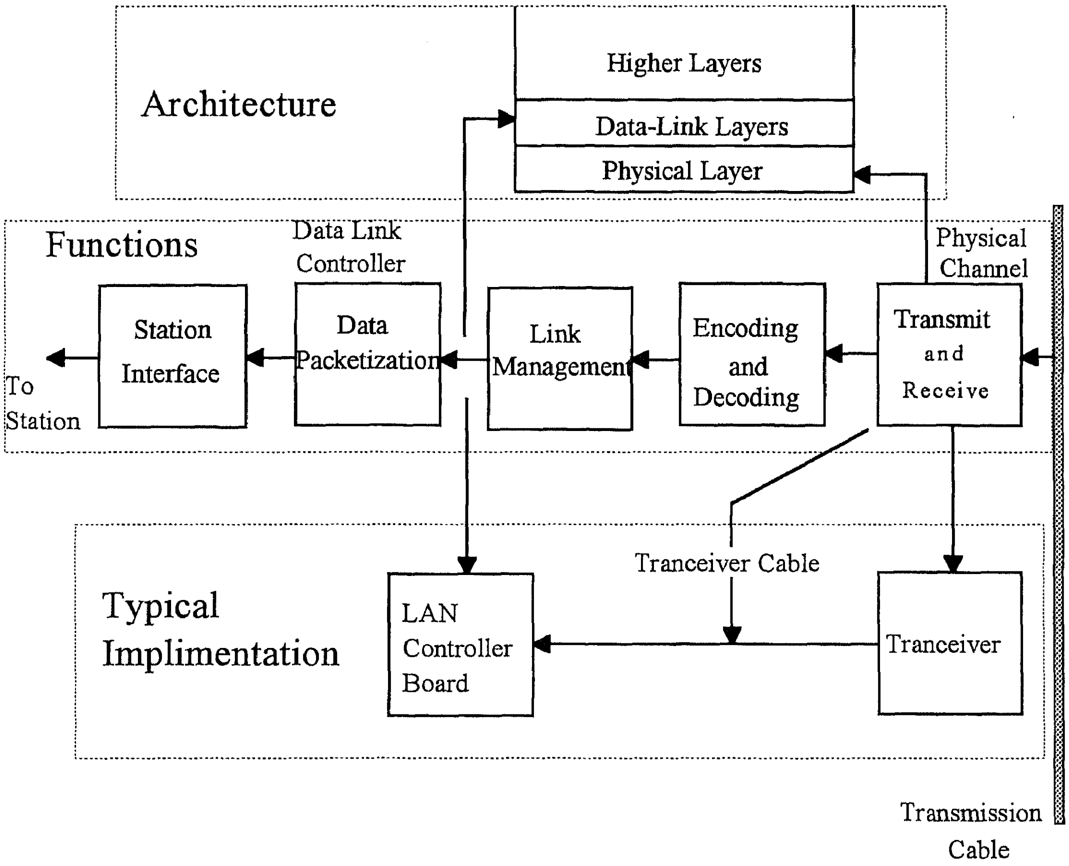


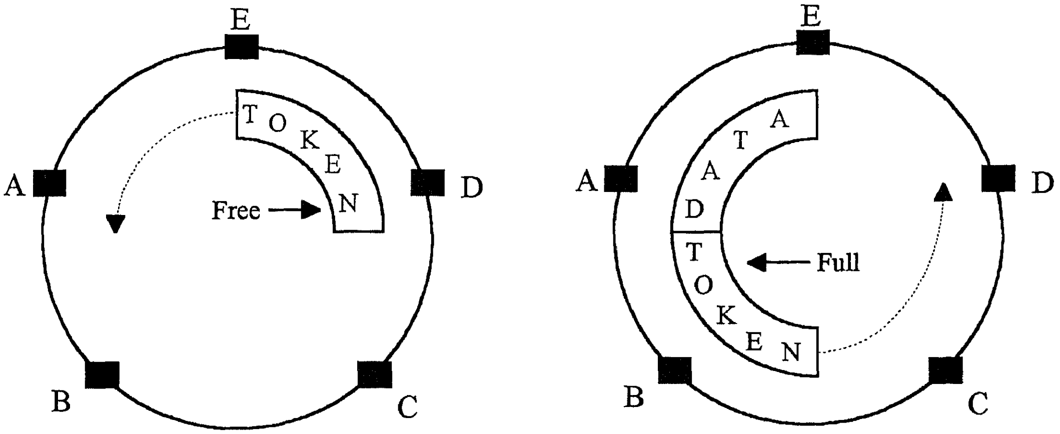
Figure 1-3. Architecture and implementation of LAN.

1.4 Token ring

The ring topology is illustrated in Fig. 1-4. The stations are connected to a concentric ring through a ring interface unit (RIU). Each RIU is responsible for monitoring the data passing through it, as well as for regenerating the transmission and passing it to the next station. If the address in the header of the transmission indicates that the data is destined for a station, the interface unit copies the data and passes the information to the user DTE (Data Terminal Equipment) or DTEs attached to it.

If the ring is idle (that is, no user data is occupying the ring), a "free" token is passed around the ring from node to node. The token is used to control the use of the ring by a free or busy indication. A busy token is an indication that a station has seized the ring and is transmitting data. A free token indicates that the ring is free and any station that has data to transmit can cease the token and transmit data. The control of the ring is passed sequentially from a node to another around the ring. This technique is called an implicit token system because any station is allowed to transmit data when it receives a free token.

During the period that the station has seized the token, it has control of the ring. Upon seizing the token (i.e., marking the token busy), the transmitting station (station A in Fig. 1-4) inserts data behind the token and passes the data through the ring. As each RIU monitors the data, it regenerates the transmission, checks the address in the header of the data, and passes the data on to the next station. Eventually, the data is received at the original transmitting station. This station is required to mark the token free and pass the token on to the next station on the ring. This requirement prevents one station from monopolizing the entire ring. If the token passes around the ring without being used, then the station can once again seize the token and transmit data.



Token (Empty) Circles the Ring

Station A Seizes the Ring

Figure 1-4. Token ring process.

1.5 Objectives and organization

The objective of this thesis is to investigate the voice/data performance of the Medium Access Control (MAC) protocol, selected for the FDDI network, through simulation.

The following issues will be addressed:

- Effect of voice on throughput and packet delay
- Effect of the target token rotation timer on voice delay
- Effect of the target token rotation timer on data throughput and
- Effect of network size on throughput and data delay

Chapter 1, discusses the evolution of FDDI under voice and data considerations. Chapter 2, discribes the FDDI architecture and network giving its characteristics, topology, applications, advantages/disadvantages, the FDDI standards and frame formats. Chapter 3, presents previous studies on FDDI networks. Chapter 4, describes the software model used for the simulation and the simulation parameters while chapter 5 presents the simulation results addressed above in the objective issues. Conclusions and future FDDI standards are presented in chapter 6.

CHAPTER 2

FDDI ARCHITECTURE AND NETWORK

2.1 Characteristics

With high-resolution graphics, Computer Aided Design / Computer Aided Manufacturing (CAD/CAM), multimedia, and other high-bandwidth applications becoming more popular, many desktop applications require higher transmission capacities over Local Area Networks (LANs). Many internetworks also require high capacities for LAN-to-LAN or LAN-to-WAN (Wide Area Networks) interconnections. FDDI (Fiber Distributed Data Interface) is a high performance fiber optic token ring LAN running at 100 Mbps over distances up to 200 km with up to 1000 physical connections (stations) connected [7, 8, 9, 10]. Attached workstations can be between 2 and 60 kilometers apart, depending on the type of fiber-optic cable used. Although FDDI is not an IEEE (Institute of Electrical and Electronics Engineers) standard, it incorporates elements of the IEEE 802 project [3]. For example, FDDI uses frame formats similar to those found in the IEEE 802.5 [4] frame format, and the FDDI data-link layer uses the IEEE 802.2 Logical Link Control (LLC) [5].

Unlike the IEEE 802.5 token ring, the FDDI protocol allows multiple frames on a ring at any time, thus improving its throughput characteristic. It is designed to provide both synchronous and asynchronous service. With synchronous services a user receives a preallocated maximum bandwidth (i.e., time to transmit specified frames) and a guarantee of a maximum delay per frame. Packet voice is one example of a service with these requirements. In asynchronous transmission all stations contend for the ring bandwidth, and response times are allocated dynamically. Asynchronous frames may have up to 8 levels of priority from 0 - 7 (000 - 111 binary) if desired, with 0 the lowest level of priority and 7 the highest, while synchronous frames are all of equal priority. If all available bandwidth is used by stations with synchronous and high-priority asynchronous bandwidth allocations, stations with no synchronous allocation and lower asynchronous priorities may not be able to transmit at all. An FDDI station receiving the token (frame with a specific format), may capture it and transmit any waiting frames up to a specified time limit. Each station on the ring retransmits the frames and copies the frames that are addressed to it. Once the waiting frames are transmitted onto the ring, or the station time allotment is exhausted, it issues a new token. A station not having frames waiting passes the token on when receiving it. After a frame returns to the sending station, that station is responsible for removing it from the ring. A copying station is able to set status bits in the frames to indicate if errors were detected. Based on these status bits, the sending station can determine successful transmission of the frame. However, error recovery and retransmission are not handled in the MAC protocol and are left for higher layers.

2.2 Topology

The FDDI cabling consists of two fiber rings, one transmitting clockwise and the other transmitting counterclockwise, as illustrated in Fig.2-1(i). If either one breaks down, the other can be used as a backup. Under normal conditions, traffic flows only on one ring; the other ring is idle. If both break at the same point, for example, due to a fire or other accident in the cable duct, the two rings can be joined into a single ring approximately twice as long, as shown in Fig. 2-1(ii). Two station classes are defined. Stations connecting to both rings are called Class A stations; stations connecting to one ring are referred to as Class B stations. Class A stations can reconfigure in the event of a failure in either the fiber or the station; Class B stations cannot. To do so, the stations (stations A) on either side of the fault automatically reconfigure internally to provide a loopback path. Each station contains relays that can be used to route traffic in the opposite direction onto the other ring. When the network is in operation, one of the rings may be designated as a standby ring only, or it may be used for concurrent transmission. Thus it is possible to have data flowing on both channels for an aggregate throughput of 200 Mbps.

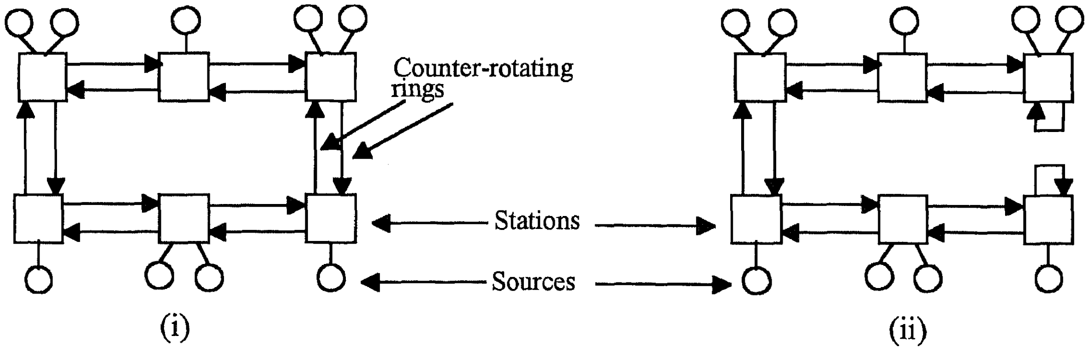


Figure 2-1. (i) FDDI consists of two counter-rotating rings. (ii) In the event of failure of both rings at one point, the two rings can be joined together to form a single long ring.

2.3 Applications

The fibers themselves are not affected by power line surges, electromagnetic interference, or corrosive chemicals in the air, so they can be used in a harsh factory environments unsuitable for coaxial cable. Fibers are also very thin, a big plus for companies with thousands of cables and bulging cable ducts. Also the FDDI satisfies the throughput requirements of the following three fundamental LAN categories:

- Computer-room networks that support communications among large mainframes computers, minicomputers, and their associated high-speed storage devices.
- Backbone networks that tie together various types of LANs such as a token ring, Ethernet, a token bus, or starLAN. An example is shown in Fig. 2-2.
- High data-rate LAN's, that connect very high speed minicomputers, engineering workstations, or high-end personal computers requiring a fast LAN to support high-bandwidth applications such as video and CAD/CAM.

With its high transmission capacity and large geographical coverage, FDDI is ideally suited for campuswide or metropolitan area network (MAN) applications.

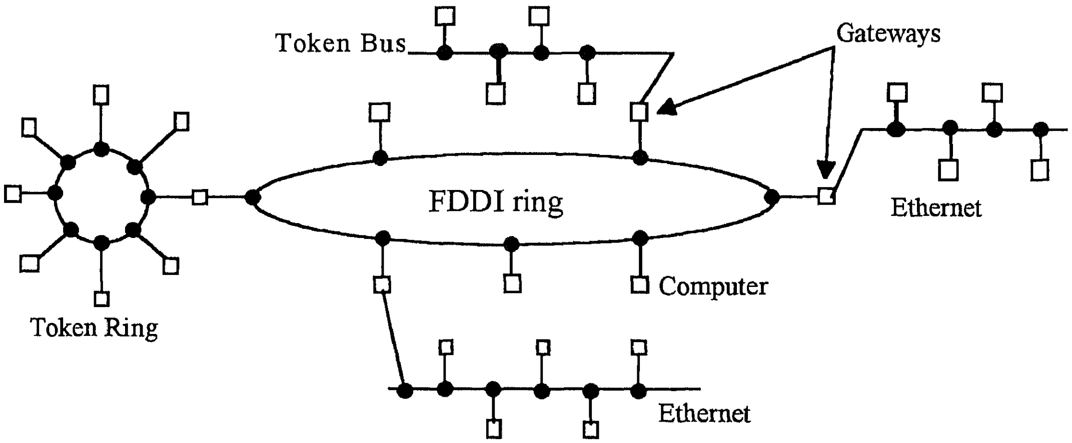


Figure 2-2. An FDDI ring being used as a backbone to connect LAN's and computers.

2.4 Advantages and disadvantages

This new type of data transmission has a number of advantages over twisted-pair and coaxial cable. Besides data transmission rates far in excess of either of these older media, fiber-optic cabling is immune to electromagnetic or radio-frequency interference, and capable of sending signals several miles without loss. This mode of transmission is also virtually immune to unauthorized reception giving thus a high level of network security. The FDDI system has been designed to provide coverage over potentially much larger distances than the IEEE 802.5 token ring thanks to its fiber-optic media which provides extremely high bandwidth with little power loss. Its maximum cable length is specified as 200 km (when wrapped or connected during a fault condition), with a maximum extent of 100 km end-to-end. As noted earlier, the FDDI token ring system allows multiple frames (both synchronous and asynchronous) to be moving simultaneously around the ring, improving the throughput capacity over that of the IEEE 802.5 token ring scheme, which allows only one frame at a time on the ring.

On the other hand, fiber optic technology is most efficient in a point to point construction and its least effective in a bus configuration. A main disadvantage against bus configuration is the large transmission time factor which would be double the one-way travel time. Over large distances and high data rates, the transmission time factor would present an enormous liability to the network. At present, fiber-optic cabling is too expensive for most installations, and its sophisticated technology makes it difficult to add new workstations after initial installation. Also, fiber optics is an unfamiliar technology requiring skills most network engineers do not have. Fibers are difficult to splice and even more difficult to tap. The latter point can also be seen as an advantage: security is excellent because fiber does not radiate and wiretappers will have as much trouble as the network owners in tapping it. Fiber networks are also inherently unidirectional, and the interfaces are considerably more expensive than electrical interfaces. However, the advantages of fiber optics are so great that much work is being directed to improving the technology and reducing the cost.

2.5 Network physical description

In an FDDI system, N stations transmitting both synchronous (voice) and asynchronous (data) traffic are connected as a Local Area Network (LAN). Each station in the system can have more than one voice or data source. Stations can also have either voice or data sources only. A general FDDI network connecting N stations with a total of Vs voice and Ds data sources is shown in Fig. 2-3. The communication rules of the network are controlled by the MAC protocol which is a Timed Token Rotation (TTR) Protocol, described later in this chapter.

Data traffic is transmitted in fixed size packets. These can be independent messages or segments of longer messages. The voice traffic, from each voice source, consists of packets which are formed by packetization of the incoming bit stream created by the digitization and encoding of an analog voice signal. This voice signal consists of talkspurt and silence periods. Only the talkspurts are packetized and transmitted.

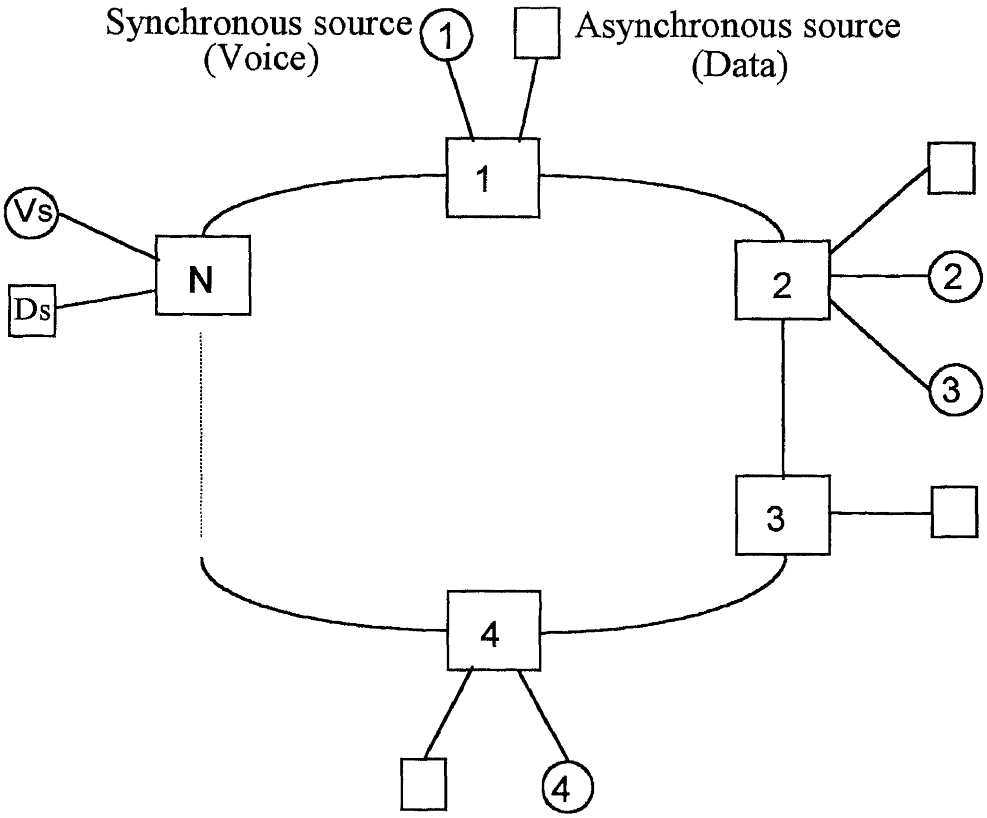


Figure 2-3. General FDDI network, N stations, vs synchronous sources.

2.6 FDDI standards

Like the IEEE standards for Ethernet and token ring networks, the ANSI standards for FDDI address specific aspects of the architecture (see Fig. 2-4 [19]). ANSI Physical Layer Medium Dependent (PMD) [11] and Physical Layer (PHY) [1] standards handle the International Organization for Standardization Open Systems Interconnection (ISO/OSI) [12, 13] model's physical layer functions; the MAC [2] standard and the IEEE 802.2 LLC [5] standard, handle the ISO/OSI model's data link layer functions. This thesis, investigates the performance of the MAC protocol under Voice/Data Integration.

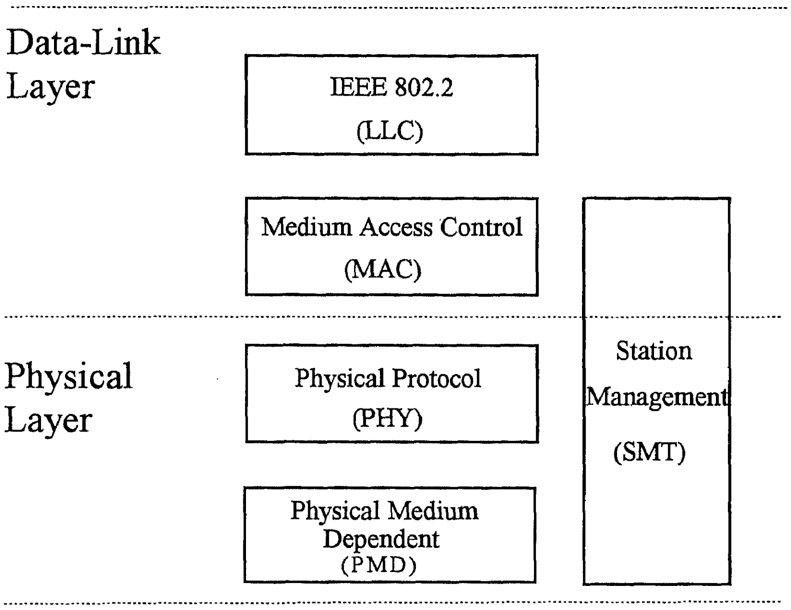


Figure 2-4. FDDI relationships to the OSI model.

The FDDI PMD layer

The PMD sublayer occupies the lowest portion of the ISO/OSI model's physical layer [11]. This layer provides the physical connection to the network, specifies cables and connectors, and defines the communication between nodes (See Fig. 2-5). It includes specifications for the optical signal, such as power levels and acceptable bit error rates. The PMD sublayer of the FDDI standard, specifies reliability enhancing techniques. These techniques include the use of wiring concentrators and automatic optical bypass switches, which makes it easier to locate faults and to bypass nonfunctioning stations. As the FDDI standard has evolved, several PMD options have emerged. These include Physical Layer Medium Dependent, using 62.5/125 micron fiber (X3.166- 1990 or ISO 9314-3, [11]) and

Single-Mode Fiber Physical Layer Medium Dependent (X3.194- 1991 or ISO 9314-4, [18]).

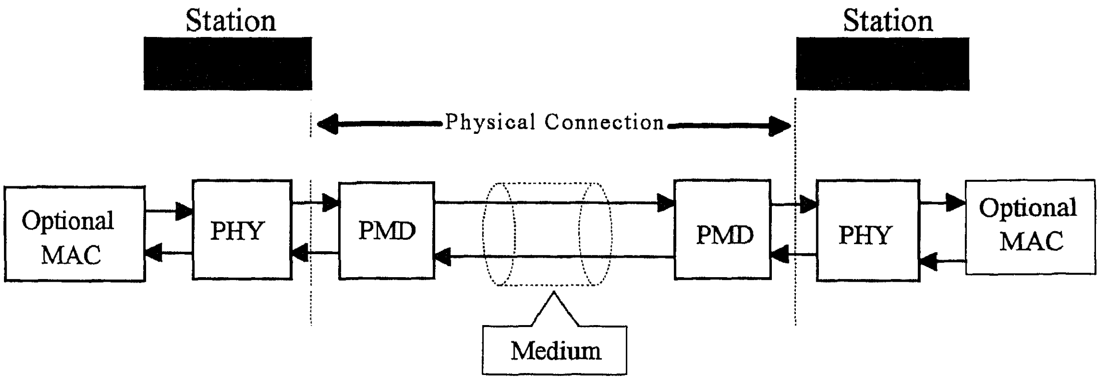


Figure 2-5. FDDI sublayer entities.
(Courtecy of Digital Equipment Corporation)

Fiber-optic PMD standards

The two currently approved PMD standards -X3.166 and X3.194- specify different types of fiber-optic cable. X3.166 sometimes referred to as MMF-PMD, uses multimode fiber-optic cable. Multimode fiber allows cable lengths between FDDI stations to extend to 2 kilometers. The cable is manufactured with a 62.5-micron core, and a 125-micron cladding (written 62.5/125). Alternatives, such as 50/125, 85/125, and 100/140 micron fiber are also allowed. The light sources are LED's with a center wavelength of between 1,270 and 1,380 nanometers. The maximum attenuation (end-to-end between FDDI stations) is 11 dB. The attenuation, known as the power budget, includes the cable attenuation and losses from other fiber-optic components, such as connectors, splices and so on. With the specifications for MMF-PMD discussed here, the FDDI attachment has a Bit Error Rate (BER) of $1x10^{-12}$. The X3.194 standard, sometimes referred to as the

SMF-PMD, uses single-mode fiber. This standard is intended for use in campus environments, which require longer transmission distances. The SMF-PMD supports links between FDDI stations of up to 60 kilometers and uses lasers as sources for the optical signal. Depending on the components used, an SMF-PMD FDDI network may have a power budget ranging from 10 to 32 dB.

The FDDI PHY layer

Reviewing Fig. 2-4, the PHY sublayer is the second layer in the FDDI architecture [1]. Together with the PMD sublayer, it provides the functions of the ISO/OSI model's physical layer. The PHY sublayer makes the logical connection between the PMD and the data-link layer, providing functions, such as defining encoding, and decoding symbols (a specific bit pattern that conveys data or control information), and clock synchronization.

FDDI symbol encoding

In the transmitting direction, the symbol originates in the higher layers of the protocol, and the MAC sublayer passes it to the PHY sublayer. One of the PHY sublayer processes then encodes the symbol for transmission on the fiber-optic link. In the receiving direction the process is reversed. The encoding conveys the data and clock information in the transmitted stream. This encoding scheme is referred to as 4B/5B and stands for 4-bit to 5-bit conversion. In this scheme, each four-bit chunk of data is coded into a symbol with five cells. Each cell contains a signal element (presence or absence of light). For example, the binary data pattern "0110" is coded into the symbol "6", which is represented by the five-bit pattern "01110". The pattern 01110 is then encoded using NRZ-I (NonReturn - to - Zero Invert - on - ones) encoding, which calls for a "1" to be represented with a transition at the beginning of the bit interval and a "0" to be represented with no transition at the beginning of the bit interval. The pattern 01110 is encoded into the signal shown in Fig. 2-6. Four-bit binary data combinations use only 16 of the 32 possible five-bit symbols. Those not used for data are used for line state symbols, start and end delimiters, control indicators, and other purposes. The sixteen binary data bit combinations were chosen to ensure that no more than three consecutive 0 bits are transmitted in each frame so a signal transition occurs every one, two, or three bit times. This provides for adequate signal synchronization and assures proper decoding of the received data at the other end of the link. The transmitter and receiver clocks operate at 125 MHz, sending 5 bits per symbol, which results in an effective data rate of 100 Mbps for the 4 bits that the higher (MAC) layer passed down for communication. Therefore we see that the FDDI is 80% efficient.

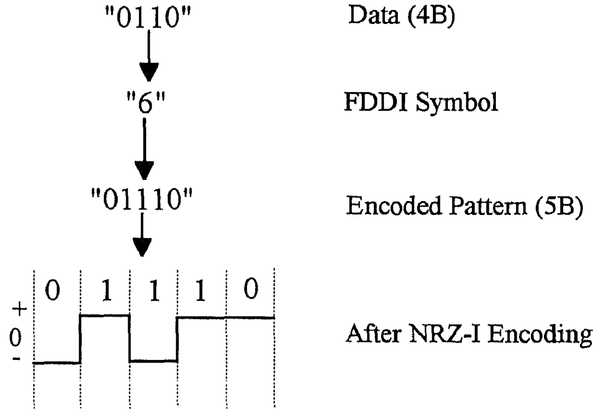


Figure 2-6. FDDI encoding scheme.

FDDI ring latency

Another PHY layer parameter is the ring latency, or the time required for the token to circulate the ring. The ring latency consists of the cumulative effect of the station and cable delays as a particular data transmission circulates the ring. The MAC and SMT layers contain timers that depend on the latency value. The PHY layer defines the algorithms used to calculate the range of latency values that these higher layers will incorporate into their functions.

The FDDI MAC layer

The FDDI MAC layer performs functions similar to those of the IEEE 802 MAC layers. It defines the addressing scheme used on the network, the frame format, and error control techniques [2]. The MAC layer determines the FDDI transmission modes (in both synchronous and asynchronous as discussed before), ring timers and counters, and algorithms for ring initialization and proper operation, as discussed in the following sections.

Ring timers and counters

The FDDI MAC layer also governs the operational relationships between various stations on the ring. There are timing requirements for the FDDI Token Ring Protocol. Station timers cooperatively attempt to maintain a specified token rotation time by using the observed network load to regulate the amount of time that a station may transmit. There is a natural bound based on timer settings. This guarantees that the token will return within a specified time period and enables FDDI to support applications that require guaranteed bandwidth, such as real-time control and voice transmission. To do so, the MAC layer defines three timers and one counter. These four entities are closely associated with token and frame transmission. The token holding timer (THT) controls the length of time a particular station may transmit asynchronous frames. The token rotation timer (TRT) schedules transmissions on the ring. A third timer , not discussed in this thesis, the valid transmission timer (TVX) is used to recover from transient ring error conditions. If this timer expires, it means that the station has seen no valid transmissions (that is, no ring activity) and that ring initialization procedures are necessary to activate the ring. Finally, the late counter (Late_CT) counts the number of TRT time periods since the MAC layer was reset or a token was received.

Ring initialization

The time allotment is controlled by a Timed Token Rotation (TTR) protocol. This supports both synchronous and asynchronous service, with synchronous service having priority. Although the asynchronous service can have up to 8 levels of priority, as noted earlier, we focus in this thesis on a single level of asynchronous service only. On ring initialization all stations on the ring go through a "claim token bidding process" (bid for the right to initialize the ring using its preferred token rotation timer (TRT) value), during which a key system parameter, the Target Token Rotation Time (TTRT), is negotiated [2]. This turns out to be the expected (average) token rotation time (TRT) on the ring. Each station requests a value T_Req for the TTRT. The minimum value of the T_Req is chosen as the operational value T_Opr of TTRT where that station having the minimum T_Req wins the bid and gains the right to issue the first nonrestricted token, initiating operation of the ring. The maximum token rotation time (TRT) around the ring is found to be 2T_Opr [14]. A station should thus request as its value of T_Req during the claim bidding process, a value that is one half of its absolute maximum token rotation time (TRT). The minimum value T_Opr is acquired by all stations and each receives a fractional Synchronous Time (ST) allotment of this time to be used in transmitting synchronous traffic.

Ring operation

Now consider the timed token rotation protocol [15]. Each station has two timers involved in this protocol. A timer called the token rotation timer (TRT) controls access to the ring. This timer is initialized to T_Opr. If TRT > 0 when a token arrives, the token is said to be on time. In this thesis we will assume that the token always arrives at the station, since we will not study the simulation analysis of the system for cases such as errors, corrections, recovers e.t.c., even though we present the whole protocol here. On the arrival of the token, the value of TRT is transferred to the second timer, the token holding timer, THT, and TRT is reset to T_Opr. Any waiting synchronous (voice) traffic is then transmitted, up to the maximum allotment ST. On expiration of ST, or completion of synchronous transmission, whichever comes first, THT is enabled and any waiting asynchronous (data) frames are transmitted until THT expires, or there are no further frames to transmit, whichever comes first. If THT expires while an asynchronous frame is being transmitted, completion of that frame is allowed. The left-hand side of the flow chart of Fig. 2-7 [16], representing the full Timed Token Rotation protocol of FDDI, portrays this early token arrival procedure. As noted earlier, this will be the case of study of this thesis.

If TRT expires before the token arrival, a counter called Late_CT is incremented by 1. TRT is the reset to T_Opr. A token arriving with Late_CT > 0 is said to be late. Synchronous transmission is allowed, up to the limit of the ST - time allotment, but no asynchronous traffic can be transmitted. The late token arrival procedure is diagrammed on the right-hand side of Fig. 2-7. On the arrival of a late token, Late_CT is reset to 0, and TRT continues to operate (it is not reset).

If Late_CT > 0 and TRT expires with no token as yet received, this is a sign that there may be a problem on the ring, and a ring recovery process is initiated. A token can thus arrive up to one additional time T_Opr late (the right hand side of Fig. 2-7), with ring recovery not required. This accounts for the statement made earlier that the maximum token rotation time around the ring is 2T_Opr [14].

All the stations on the ring are responsible for monitoring the function of the token-passing protocol and for initializing the ring if an invalid condition occurs. Error condition can be an extended period of inactivity, loss of power on certain nodes or extended periods of data transmission without a token. When a frame detects either condition, it begins a process of initializing the ring. Claim token procedure is performed to redetermine the TTRT value of the network. When a serious failure occurs, such as break in the ring, a beacon process is used. Like IEEE 802.5, FDDI uses beaconing to isolate serious failures such as breaks in the ring. Nodes detecting serious ring failures send beacon frames until they see a beacon frame from the upstream neighbor. Eventually, the only station sending beacon frames is the one immediately downstream from the problem. Beacon frames are sent until the beaconing node receives its own beacon frame, at which point it assumes the problem has been fixed and initiates the claim token process.

The FDDI LLC layer

The LLC layer resides at the upper portion of the data-link layer and convey user information [5]. FDDI networks use the IEEE 802.2 protocol for this function. The LLC sublayer is responsible for providing a transmission path that appears error free to the network layer. Furthermore, the functions involved are to be transparent to upper layers. Within the IEEE structure, these functions are provided by the IEEE 802.2 protocol specification. LLC services are compatible with various Media Access Control (MAC) standards.

The FDDI SMT function

The Station Management (SMT) function provides the station-level control process necessary to assure proper operation of the various FDDI layers. Fig. 3(b) shows how SMT operates on the PMD, PHY, and MAC layers in the FDDI architecture. The ASC X3 has defined SMT, and it is currently in draft form [17]. SMT defines three areas of FDDI management: Frame-based management gathers information about current FDDI network operation. Connection Management (CMT) handles the physical connections and network topology at the PMD and PHY layers. Ring Management (RMT) deals with the

MAC layer's logical ring operation, such as the proper circulation of the token. For example, as with IEEE 802.5, each station monitors the ring for problems that might require ring initialization. Also, beaconing is used to isolate serious failures such as breaks in the ring.

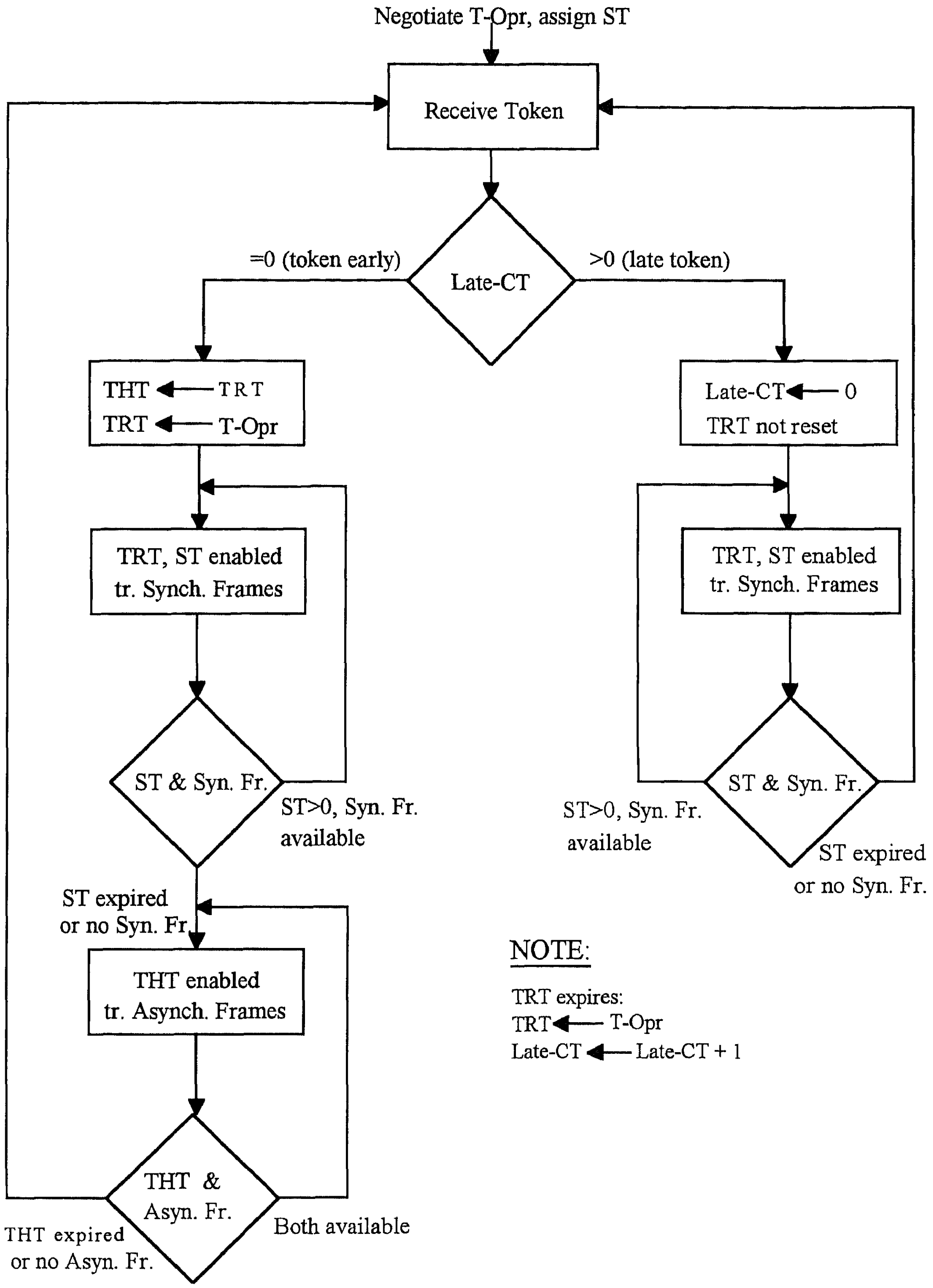
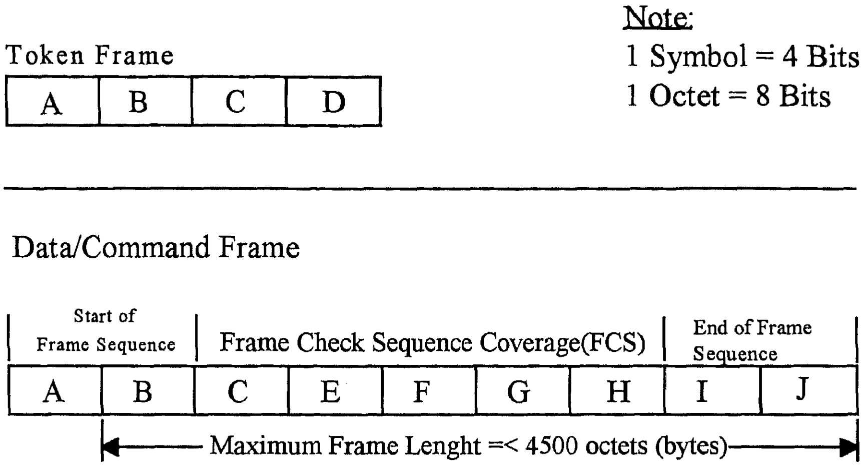


Figure 2-7. Flow diagram, FDDI timer token rotation protocol [16].

2.7 FDDI frame formats



A = Preamble (16 or more Symbols)
B = Start Delimiter (2 Symbols)
C = Frame Control (2 Symbols)
D = End Delimiter (2 Symbols)
E = Destination Address (4 or 12 Symbols)
F = Source Address (4 or 12 Symbols)
G = Information (0 or more Symbols)
H = Frame Check Sequencer (8 Symbols)
I = End Delimiter (1 Symbol)
J = Frame Status (3 or more Symbols)

Figure 2-8. FDDI frame formats.

FDDI frame formats and field functions

Like IEEE 802.5, the FDDI protocol describes two basic types of frames: tokens and data/command frames. These two frame formats are shown in Fig. 2-8 and surveyed in the following paragraphs. A token has a preamble, a start delimiter, a frame control byte, and an end delimiter. When ring interfaces are not transmitting ring maintenance commands or data, they simply circulate a token around the ring. After a ring interface transmits a ring maintenance command or data frame, it issues a new token, essentially appending it to the end of the frame. Data or command frames vary in size but cannot exceed 4500 bytes. Command frames carry ring maintenance commands for MAC sublayer software and no actual data for overlying protocol layers. Data frames carry data for overlying protocol layers. In addition to fields found in the token, both data and command frames include ring addresses of transmitter and receiver interfaces, a frame check sequence (FCS), and a final Frame Status field carrying aknowledgment information back to the frame transmitter.

Preamble and start delimiter:

When an interface is not repeating or transmitting part of a frame, it constantly transmits IDLE symbols (all 1s). At least 16 of these five-bit symbols must separate frames. This preamble pattern sets up the receiver's signal clock for any frames that may follow. The two start delimiter symbols alert the receiver to the start of the frame or token.

Frame control:

The frame control field indicates:

- Distinguishes between a frame or token
- Whether the frame is part of a synchronous or asynchronous transmission
- Whether the frame uses a 16- or 48- bit address
- The type of command (if a command frame)

Destination address:

This field contains the destination station address. When a receiver reads its own address in this field, it copies the following source address and information fields into memory as it repeats the frame onto the ring.

Source address:

This field contains the source station address. The originating station's address is coded here.

Information:

This field (present in data frames) contains a variable number of symbols containing information to be passed up the protocol stack to overlying link-control software.

Frame check sequence:

The frame check sequence (FCS) in this field is like the 802.3 and 802.5 FCS and is used to ensure that the frame is received correctly. The frame's destination computes a Cyclic Redundancy Check (CRC) on the Control, Address, and Information fields. When the CRC computed at the destination does not match the FCS included in the frame, the destination declares an error.

End delimiter:

This field contains the symbol "T" (terminate). Tokens contain two T symbols (TT), while data/command frames contain one T symbol.

Frame status:

Each FDDI destination uses a minimum of three symbols in this field to inform the frame's source of the frame's disposition. When the source receives these symbols unchanged, it assumes that the destination is not on the ring. By modifying this field, the destination can indicate that it found the frame acceptable and copied it to memory, or that it found the frame unacceptable due to some condition like inadequate buffer space or FCS error.

Calculation of the maximum frame transmission

As noted earlier, the maximum cable lenght of the FDDI system is specified as 200 km, with a maximum extent of 100 km end-to-end. The maximum number of physical connections has been set at 1000. Using a maximum latency figure of 15 symbols or 0.6 μsec per physical connection to retransmit the Starting Delimiter (SD) field , this indicates that the maximum total station latency around the ring is 600 μsec. Maximum propagation delay, end-to-end, using the figure of 5 μsec/km as the delay per km, is 1 msec. The maximum ring latency is then L_{max}= 1.6 msec. The maximum frame length (Fig. 3(f)) is specified as 9000 symbols (4500 octets), not including the 16-symbol preamble. At the 100 Mbps symbol rate (125-Mbps line rate, as already noted), the maximum frame transmission time is 0.36 msec.

CHAPTER 3

PREVIOUS WORK

3.1 Calculation of the minimum TTRT (T_Opr)

Following the discussion of section 3.2.5 for the ring initialization procedure, an estimation of the minimum operational value T_Opr is presented. Specifically, suppose there are V_s synchronous stations on the ring. The total synchronous allotment is then V_s x ST sec., where ST is the Synchronous Time allocation. T_Opr should then be chosen large enough to accommodate this synchronous allotment. It should also be large enough to allow at least one data (asynchronous) frame per token rotation (taking into account the case where a frame has just begun transmission when Token Holding Time-THT runs out; recall that once a frame begins transmission, it is allowed to complete), plus the time to transmit the token. Since T_Opr turns out to be the average time for a token to rotate around the ring, it is clear that T_Opr must satisfy the inequality:

$$T\_Opr >= V_s \cdot ST + F\_Max + Token\_Time + L \qquad (1)$$

Here F_Max is the maximum frame transmission time of 0.36 msec (calculated earlier), the token transmission time Token_Time is 0.88 µsec (including the 16-symbol preamble Fig. 3(f)), and L is the ring latency discussed earlier.

The operational value T_Opr of the Target Token Rotation Time TTRT is bounded by default minimum value of 4 msec and a default maximum value of 165 msec. This latter value is chosen to ensure stable ring recovery [2].

3.2 Calculation of the maximum token rotation time

The actual proof of the 2T_Opr bound on the maximum token rotation time is given by [14]. Note from the TRT protocol (Fig. 2-7), that the maximum time a station can have for transmitting asynchronous frames is T_Opr + F_Max - L. This time is given by the value of the timer TRT on arrival of a token. Since TRT was reset to T_Opr at the previous token arrival, its maximum value at the current arrival time can only be T_Opr - L, and is possible only if none of the other stations on the ring had asynchronous traffic to transmit. Say the station Token Holding Timer, THT, starting with this value of TRT on the arrival of the token, expires just as an asynchronous frame transmission begins. The frame will continue transmission to completion, for a maximum time F_Max. The total maximum time available for asynchronous transmission is thus T_Opr - L + F_Max. A little thought will indicate that under this condition THT for all subsequent stations on the ring will have expired as the token reaches each in turn. Hence they cannot transmit any asynchronous frames. They can, however, transmit any waiting synchronous frames to their maximum overall allotment, $V_s \cdot ST$. From the previous section, this is bounded by T_Opr - F_Max - L, neglecting the token transmission time. Adding this to the maximum asynchronous time T_Opr - L + F_Max, and adding an additional L units of time for this next token rotation, the bound on the time between initiation of the transmission of the asynchronous frames and the next receipt of the token is 2T_Opr -L, which is in turn bounded by 2T_Opr.

3.3 Calculation of the maximum data utilization

Consider the case where data traffic only is present. Karvelas and Leon-Garcia gave a bound on the maximum data utilization [25]. Most FDDI rings, particularly those used to interconnect LAN's and workstations, would be expected to operate in this mode (asynchronous traffic only). This bound also depends on T_Opr and L, increasing with increases in the former, decreasing with increases in the latter. This is to be expected, since the value of the token holding timer THT at the time of arrival of the token determines the amount of data traffic that can be transmitted at a station, and THT is reinitialized each cycle to T_Opr. As the latency L increases, THT is reduced correspondingly. To find this bound, it was assumed that data frames were always available. The maximum data utilization was found to be given by:

$$p_{d,max} = N(T - L)/(NT + L) < 1 \qquad (2)$$

$$T = T\_Opr + h_d$$

where, h_d is the maximum asynchronous frame length and N the number of station on the ring.

3.4 Effect of increasing T_Opr too much

According to [23], there are two reasons for not increasing T_Opr too much: the synchronous delay time bound 2T_Opr may become too large. Recall that if synchronous traffic is present, its periodic nature requires the token to arrive often enough to allow each synchronous packet to be transmitted before a second one arrives. In addition, it turns out that the worst case data access delay in heavy traffic is given by (N-1) T_Opr +

2L [23]. Too large a value of T_Opr can thus result in inordinate data delays, hence large queue sizes, in heavy traffic. The value of 8 msec has been proposed as a "good" choice of T_Opr, providing relatively high throughput and moderate delays over a wide range of network cases [23].

3.5 Effect of station number and network capacity on FDDI performance

Ghani and Schwartz [24], studied the effect of increasing the number of stations in a ring and the effect of network capacity on FDDI performance. Increasing the number of stations on the FDDI ring, would be expected to deteriorate the FDDI performance considerably by increasing latency. However, since each additional station only introduces 0.6 msec of latency at 100 Mbps, the effect would not be expected to be significant for large rings in which the propagation delay dominates. This is the situation for the reference case with a 50 km ring circumference. The propagation delay around the ring is 250 μsec using a figure of 5 μsec/km propagation delay. Going from 15 to 100 stations adds 51 μsec to the total latency increasing it from the reference case value of 250 μsec to 301 μsec. Using (2) to get a quick approximation to the maximum data utilization, it was found that $p_{d,max}$ has changed from 0.95 with 15 stations to approximately , 0.94 with 100 stations. However, if you add up to 1000 stations, latency becomes much larger and definitely will be an issue of consideration.

Ghani and Schwartz [24], also in their work, showed by simulation the effect of increasing the network capacity (bandwidth) from 100 Mbps to 1.2 Gbps on the FDDI performance. They found out that bandwidth have no effect on FDDI performance.

CHAPTER 4

NETWORK II.5 SOFTWARE MODEL

For this thesis, NETWORK II.5 [6] was used to measure the performance of an FDDI network for varying network characteristics. The model was designed to run for all input parameters. After the program ran, the results of each input combination will be analyzed and the output data will be plotted and reported. The software model used throughout the analysis performance of the FDDI network is shown in Fig. 4-1 below.

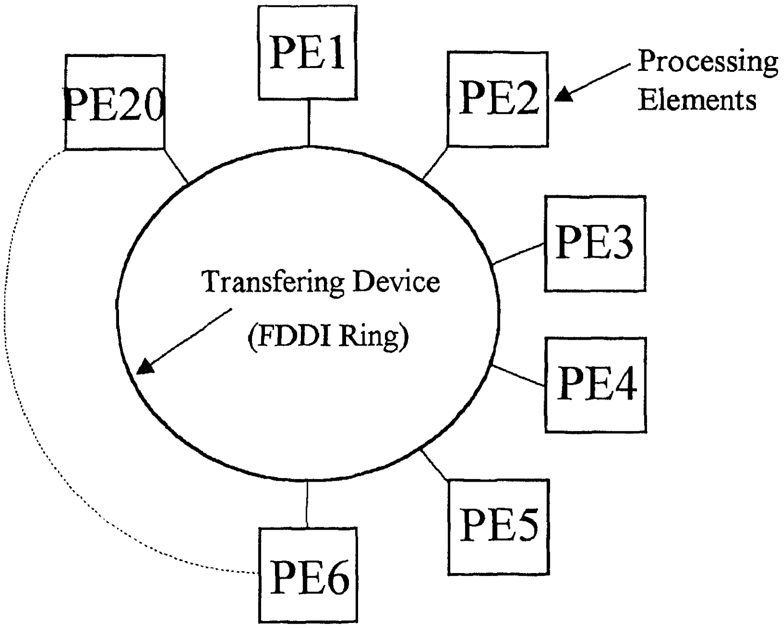


Figure 4-1. The NETWORK II.5 software model.

The objective of this thesis is to investigate the voice/data performance of FDDI networks using several input parameters. The following assumptions take place:

- Packet size is fixed
- Arrival rate at each station is the same and follows Poisson distribution
- The system is symmetric

Input Parameters:

Number of voice sources = {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}

Network Size in Km = {20, 40, 60, 80, 100}

TTRT (Target Token Rotation Time) = {5, 6, 7, 8, 9, 10, 11, 12, 13}

4.1 Processing elements (PE's)

The stations connected to the ring (Transferring Device) are modeled using the NETWORK II.5 Processing Elements. The Processing Elements have two sections: the hardware section and the software section. The hardware section is used to set attributes like the distance between two stations the length of each message, and its destination. Each PE is capable of transmitting data or packet voice sources, or both. The software section tells the Processing Elements of how often to generate messages using Poisson distribution, when to send messages provided that its preconditions are met (token received) and what is the priority associated with that message (TTRT value).

4.2 Transfer device (TD)

Transfer Devices (TD's) are the links connecting Processing Elements. They are used to move data between two Processing Elements. They can connect as many of these elements as desired. In this thesis 20 Processing Elements (stations) are connected to the transferring device. Each Transfer Device has a defined specification giving the transfer speed, transfer overhead and protocol definition. Transmissions move between Processing Elements over the Transferring Device in the form of messages. Transfer Devices automatically organize every message transmitted into words and blocks (packets). The Transferring Device specifies the amount of time required to transfer a word and to transfer a block. The protocol defines the method to resolve contention between Processing Elements for a Transfer Device. In this paper we investigate the FDDI protocol which is a priority token ring protocol. The attributes of the transferring device are presented below giving values and descriptions.

Cycle time = 0.008 μsec (Baud rate = 125 Mbps)

Bits per cycle = 1

Cycles per word = 4

Words per block (packet) = 3200

Word overhead time = 0.008 μsec

Block overhead time = 200 μsec

Number of stations linked = 20

Protocol = priority token ring

Station Latency = 0.6 μsec or 60 bits per station for a 100Mbps ring

Cable delays = 5 μsec per km

Packetization Interval = 10 msec

Synchronous Allocation Time = 6.4 μsec (100 Mbps)

Note that a cycle time of 0.008 μsec equates to 125 Mbps. Since FDDI employs 4 to 5 bit encoding the actual data rate is limited to 100 Mbps. The encoding is noted as a word overhead time of 0.008μsec (1 bit time).

4.2.1 Transfer device attribute descriptions

Cycle Time: represents the amount of time in microseconds required for the transfer device to complete one cycle.

Bits per cycle: represents the number of bits of data that will be transmitted in one transfer device cycle.

Cycles per word: Cycles per word, when multiplied by bits per cycle, indicate the number of bits per word of data.

Words per block: represent the number of words of data contained in one block. Words are the lowest level of a user-defined packet structure. Blocks are the highest level of a user-defined packet structure.

Word Overhead Time: represents the amount of time added to each word transferred by the transfer device, used for adding parity bits, start 1 stop bits, and so on.

Block Overhead Time: represents the amount of time added to each block transferred by the transfer device to account for destination addresses, checksums, and so on, used to represent station latency.

Connection List: of a transfer device contains the names of all Processing Elements that are linked to the Transfer Device.

Protocol: represents the method of passing data that the transferring device uses in a simulation and defines the method of resolving contention between Processing Elements.

Packetization interval: If TTRT is greater than the voice packetization interval, then it discards the voice packet because the new packet overwrites the old packet. The voice packetization interval is always set twice the TTRT so that it is guaranteed that there will be a voice packet ready for transmission at a station upon arrival of the token. Using a reference TTRT of 5msec, throughout this analysis the voice packetization interval is set at 10 msec.

This says that if you increase TTRT more than 10 msec there is no guarantee that the voice packet will be sent on time because the token will delay to arrive and therefore that voice packet will be discarded by the generation of a new packet.

Synchronous Allocation Time: The guaranteed time to transmit synchronous frames.

4.2.2 The priority token ring protocol of NETWORK II.5

The priority token ring protocol offers a generic capability that can be used to model the Fiber Distributed Data Interface (FDDI) Token Ring. It is defined by a token rotation sequence, allocations for synchronous traffic and target token rotation times for asynchronous traffic. An unlimited number of priority classes may be specified for asynchronous traffic.

When simulation starts, the first PE in the TD Connection List is given the token. When the token is released, it is passed to the next PE in the connection list. After the token has been passed to each PE in the connection list, it is once again passed to the first PE in the connection list and this cycle will continue for the duration of the run. Both synchronous and asynchronous timers may be specified which will determine how long an individual PE will retain the token.

When a PE receives the token, it will first transmit using its synchronous allocation. A PE's synchronous allocation is a guaranteed amount of time that the PE may use the TD every time that PE receives the token. Each connection within the TD connection list that represents a PE will have its own synchronous allocation defined. The synchronous allocation is tested before the start of every transmission and is not used to interrupt a transmission. Therefore, a PE will continue to transmit after the expiration of the synchronous allocation for any transmission begun before the allocation expires.

Asynchronous traffic is regulated by the Target Token Rotation Times of a connection and the Token Holding Time. Whenever a token is received, the time between the current time and when the token was last received is recorded as the current Token Rotation Time (TRT). The Token Holding Time is computed as the difference between the defined Target Token Rotation Time (TTRT), and the Token Rotation Time (TRT). When asynchronous traffic begins, the THT clock is started beginning with this assigned value. The time spent on synchronous traffic does not affect the THT clock. Before each asynchronous transmission, the PE compares the THT to the time of the selected TTRT. If the THT is equal to or greater than the TTRT, the token is passed to the next PE in the

Connection List. A TTRT will not interrupt a transmission, but is used to determine if the transmission should begin.

CHAPTER 5

SIMULATION AND RESULTS

A computer simulation which evaluates the effects of voice and data transmission for a FDDI network is run for several configurations. The basic algorithm for the program is given by Network II.5 software package. Several modifications to the model are used to evaluate different performance parameters of a FDDI network.

5.1 Simulation algorithm

The computer model for the FDDI performance is structured to simulate the performance of the network in the presence of both voice and data traffic.

The program defines several constants that are used during the simulation. Several of these constants remain unchanged during all of the simulations. The number of stations attached to the FDDI ring for each simulation is 20. Each of these stations is separated by the maximum distance of fiber (2 Km) allowed in the ANSI standard for multimode fiber. The voice packetization interval is at 10 msec and all packet field lengths remain constant throughout the simulation.

Other constants are converted into variables as the performance of the network is evaluated with respect to that parameter. The first simulation that is run calculated the average voice and data packet delays for a given number of voice sources. Other simulations are run to show the effect of the target token rotation timer on voice delay and the effect of network size on the maximum throughput rate.

The simulation consists of a number of complete token rotations around a FDDI ring. During the first rotation of the token only voice packets are transmitted if they are available at a station. Both voice and data packets are generated using Poisson (exponential) interarrival distributions. Also during the first token rotation, each station stores the arrival time of the token for use during the next cycle.

During each successive rotation of the token stations transmit available voice packets. After the voice transmission, data is transmitted until the station's counter for data transmission times out or all data packets are transmitted. The allowable time for data transmission is a function of the target token rotation time and the actual token rotation time. This process continues for a number of cycles determined by a constant at the beginning of the program. For the final runs of the simulation the number of token rotations was set to 1000. In addition, a test for infinite system delay was inserted into the program which monitors the average system delay. After a given number of cycles the average system delay should reach a stable point, if it continues to increase then the network is overloaded and the system delay is infinite.

5.2 Effect of voice on throughput and packet delay

The maximum data throughput rate for a FDDI network is found to decrease with an increase in the number of voice sources. One side effect of increasing the number of voice sources is that a fewer number of data packets will be transmitted because data throughput must be reduced and because less time is allotted to each station for asynchronous transmission. For this series of simulations the target token rotation time is held constant at 5 msec. The following ten graphs Figs. 5-1 to 5-10, show the packet delay time as a function of data throughput. Each graph shows both the voice and data packet delay as a function of the data throughput rate for a given number of voice sources. The delay for a voice packet increases somewhat linearly as the data traffic increases. However, the delay for data packet increases exponentially as the network reaches an overload condition. The delay time for a data packet tracks the voice delay curve until 60 - 70% of the maximum throughput, about 10 Mbits before the maximum throughput rate. At this point the delay for a data packet increases sharply approaching infinity. The use of the timed token protocol, designed to provide equal access onto the ring , as well as allowing synchronous traffic to be transmitted with a bound on delay, reduces the expected improvement somewhat. This is due to the limit on data traffic throughput, which leads in turn, at high utilizations, to correspondingly higher access delay. Note that the voice delay is calculated by the difference between the voice packet arrival time and the real time plus the time it takes to transmit the voice packet. The data delay is calculated in a similar manner.

Increasing data throughput causes increased delays in both voice and data. The delays are the result of each station holding the token for a longer period of time in order to transmit the increased number of packets.

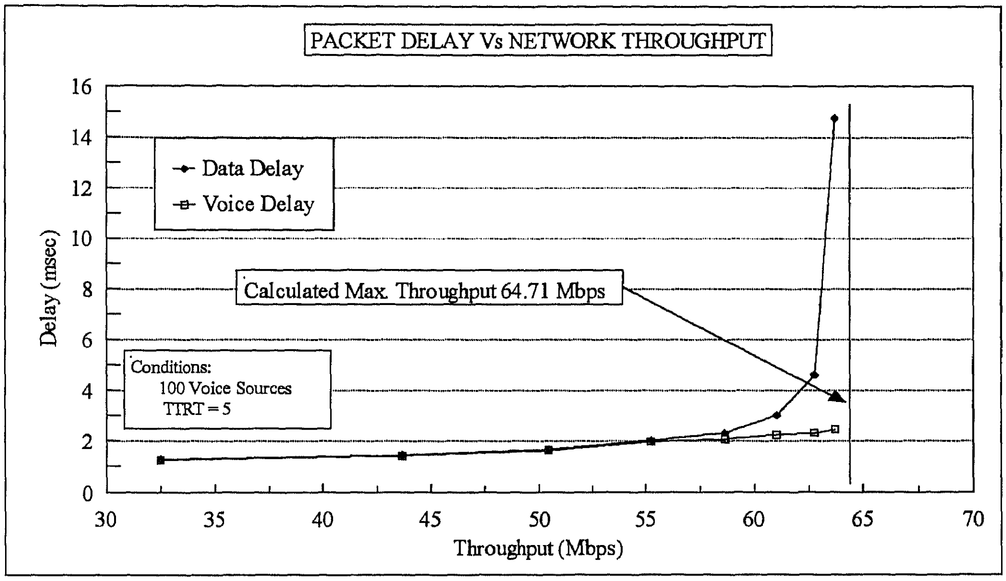


Figure 5-1. Packet delay vs network throughput (100 Voice sources).

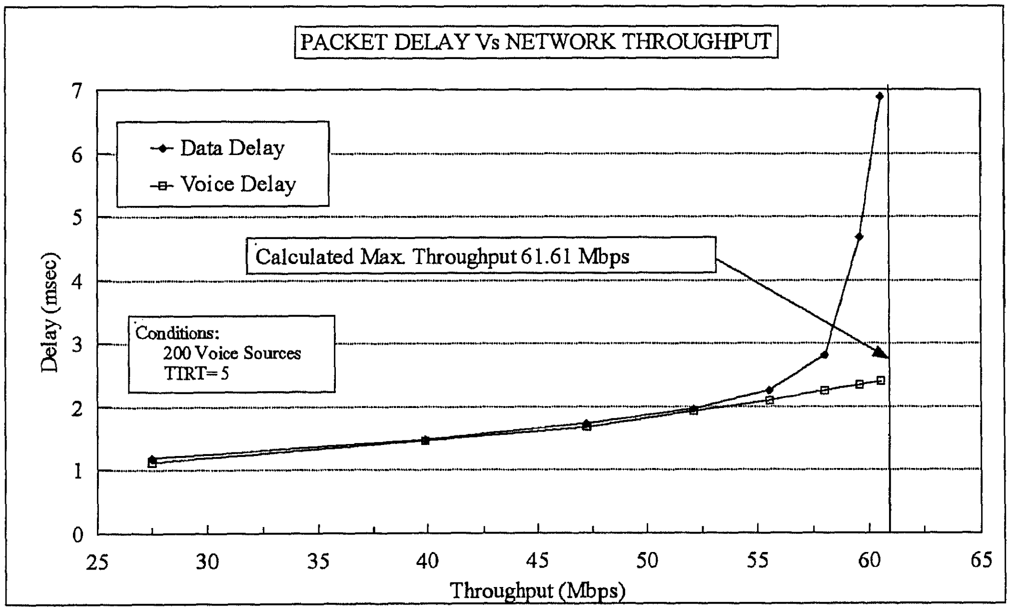


Figure 5-2. Packet delay vs network throughput (200 Voice sources).

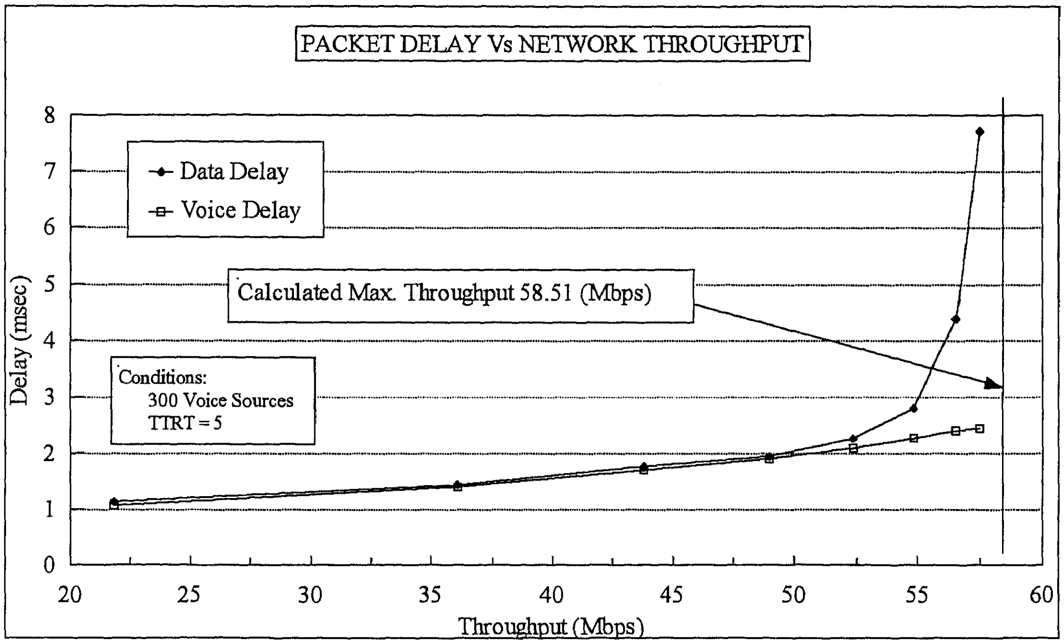


Figure 5-3. Packet delay vs network throughput (300 Voice sources).

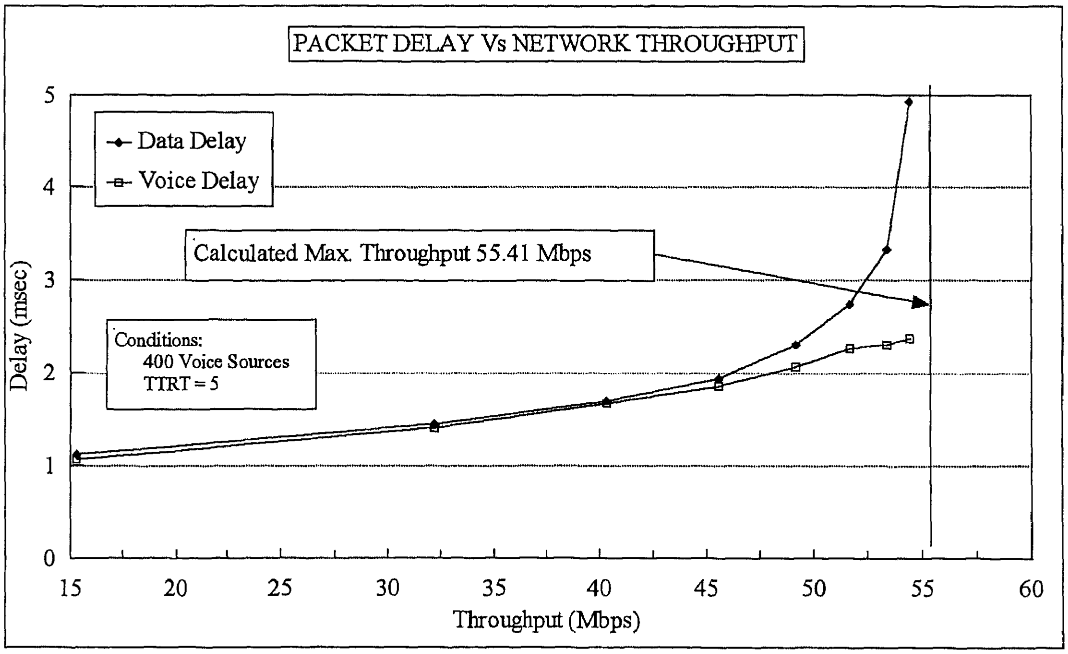


Figure 5-4. Packet delay vs network throughput (400 Voice sources).

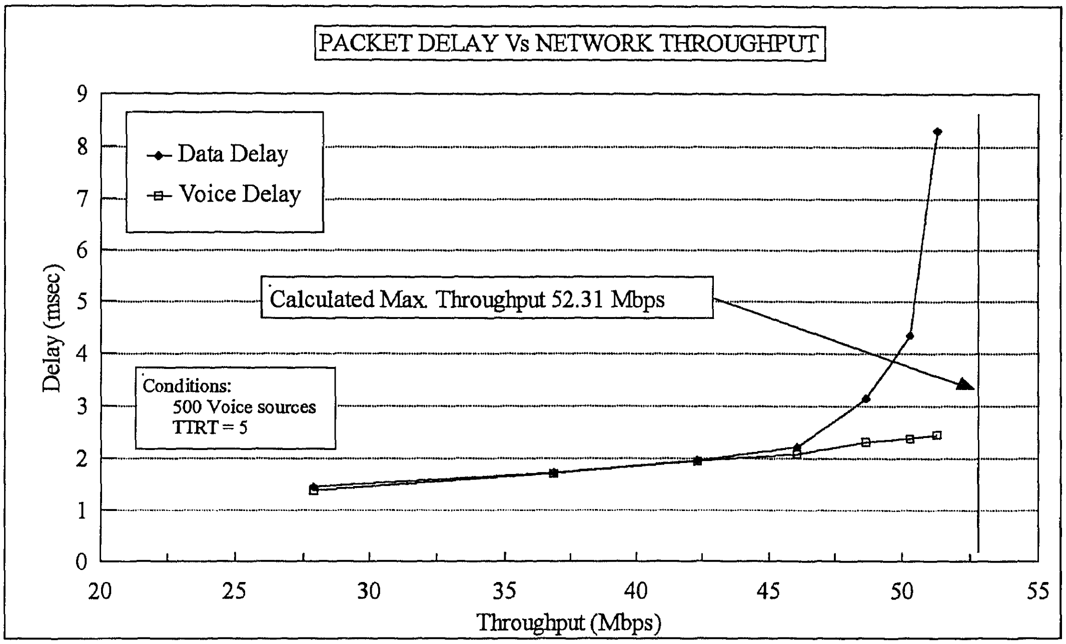


Figure 5-5. Packet delay vs network throughput (500 Voice sources).

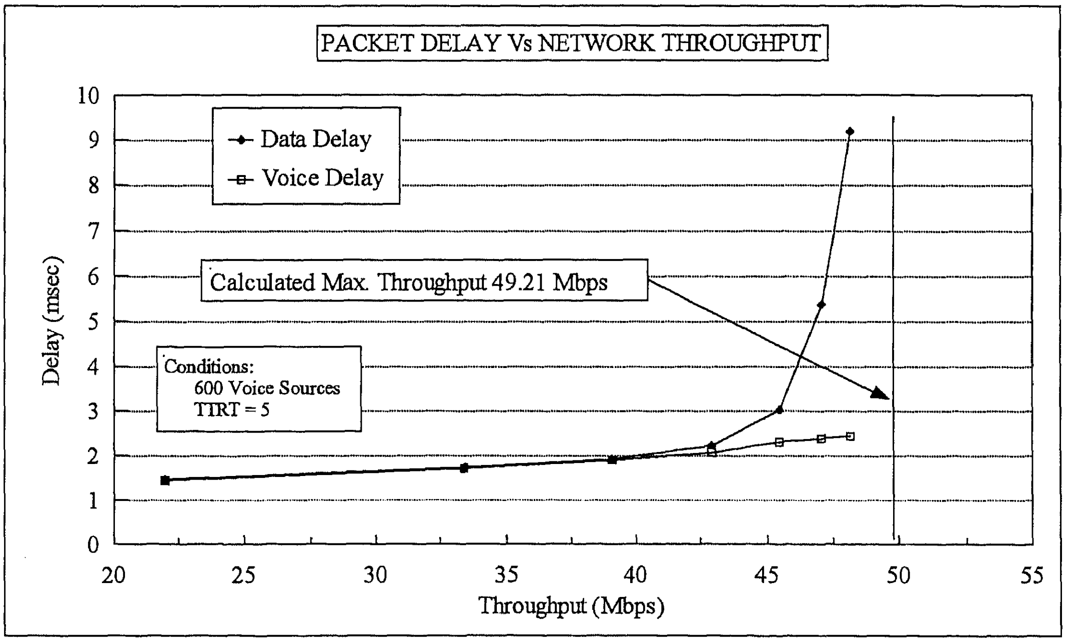


Figure 5-6. Packet delay vs network throughput (600 Voice sources).

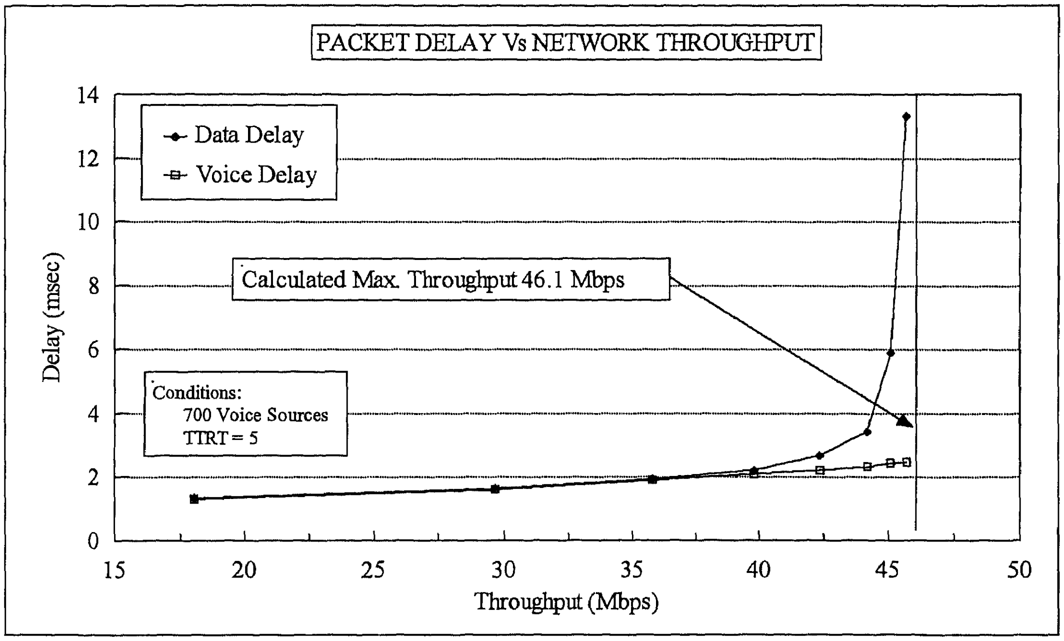


Figure 5-7. Packet delay vs network throughput (700 Voice sources).

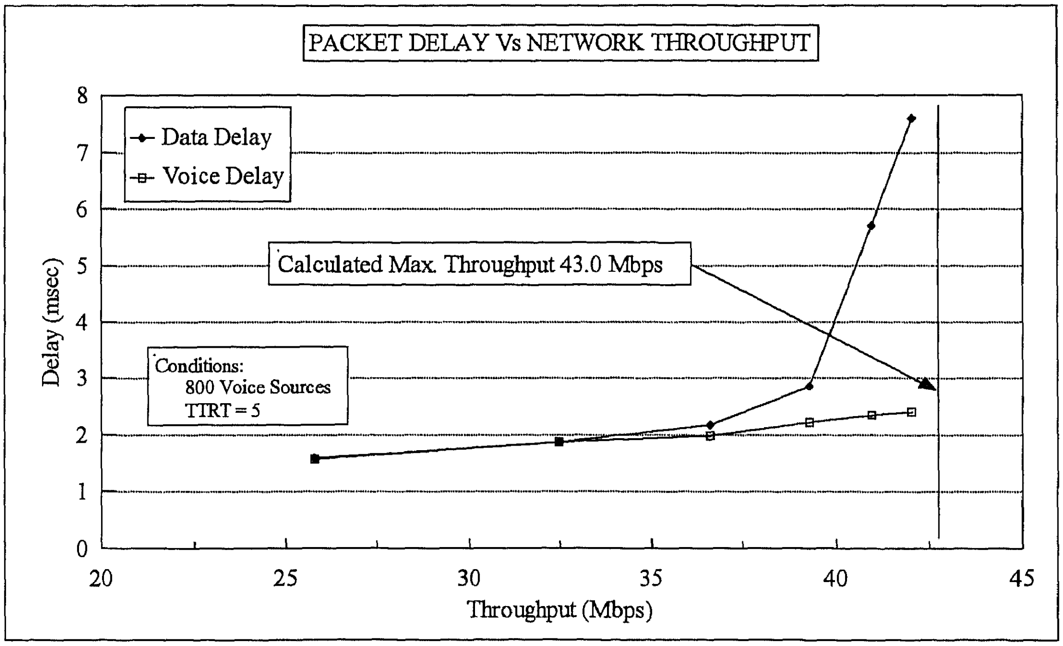


Figure 5-8. Packet delay vs network throughput (800 Voice sources).

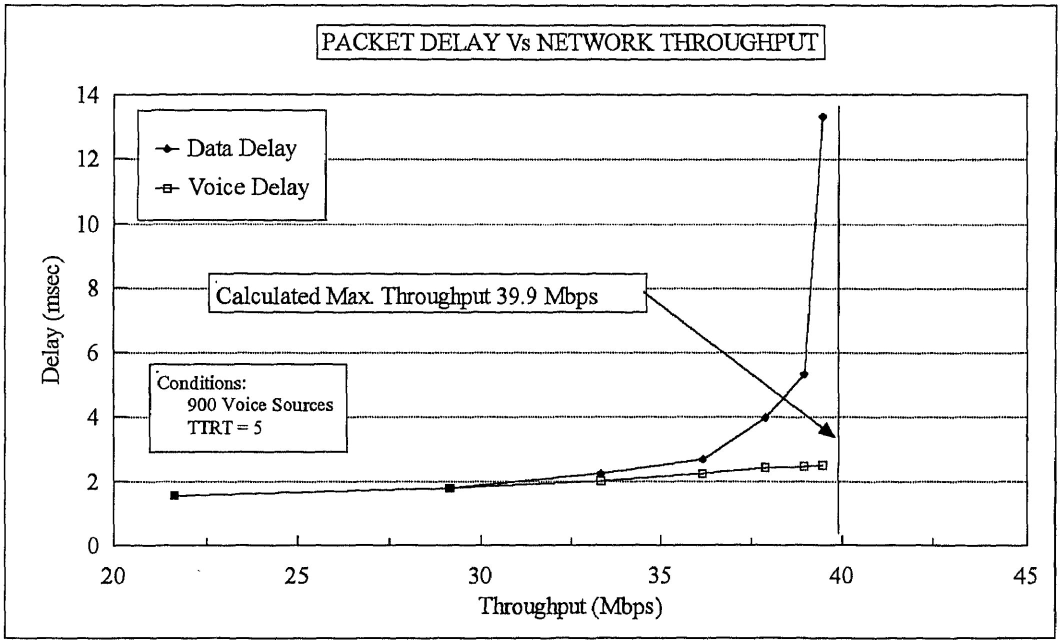


Figure 5-9. Packet delay vs network throughput (900 Voice sources).

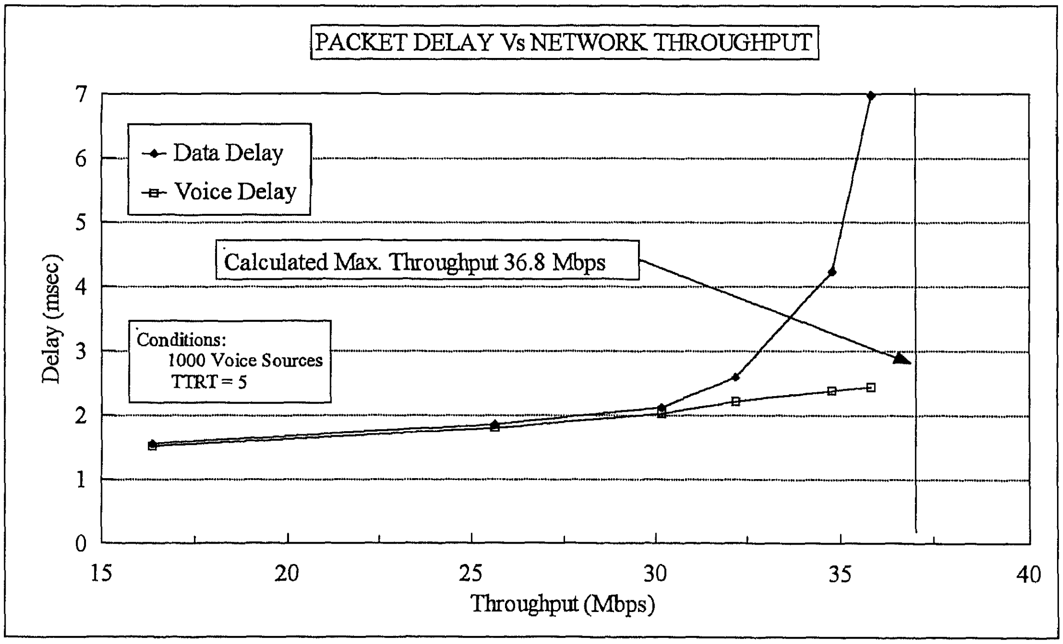


Figure 5-10. Packet delay vs network throughput (1000 Voice sources).

5.3 Effect of the target token rotation timer on voice delay

As the target token rotation timer is increased both the maximum voice packet waiting time and the average voice delay increase. This is shown in Fig. 5-11. Both the maximum voice wait time and the average voice packet delay increase as a function of target token rotation timer until the target token rotation timer reaches the voice packetization interval (10 msec). At this point the stations on the network begin to discard voice packets. The average delay for a voice packet also reaches its maximum value at this point. Also, the maximum voice packet wait is equal to the voice packetization interval. The percentage of discarded voice packets also begins to increase sharply at this point. This percentage is shown as a function of the target token rotation timer on the same graph using a second Y axis. The percentage of discarded packets remains zero until the TTRT reaches the voice packetization interval. It then increases rapidly to the point where almost 25% of the voice packet are discarded when the TTRT is 3 msec past the voice packetization interval. At this point the quality of the voice transmission is completely incoherent. It can be noted that within 0.5 msec of the voice packetization interval only 2% of the voice packets are discarded and the quality of the voice transmission will still be reasonable.

An additional observation for data throughput is noted and Fig. 5-12, is plotted to show that the data throughput increases as the target token rotation timer increases. This is an obvious conclusion because the time for allowable data transmission is extended by the increase in the target token rotation timer. The use of the timed token rotation protocol to provide guaranteed service to synchronous traffic limits the amount of asynchronous data traffic on the network. The data access delay and the maximum data throughput turn out to depend critically on the operational Target Token Rotation Time (TTRT) T_Opr.

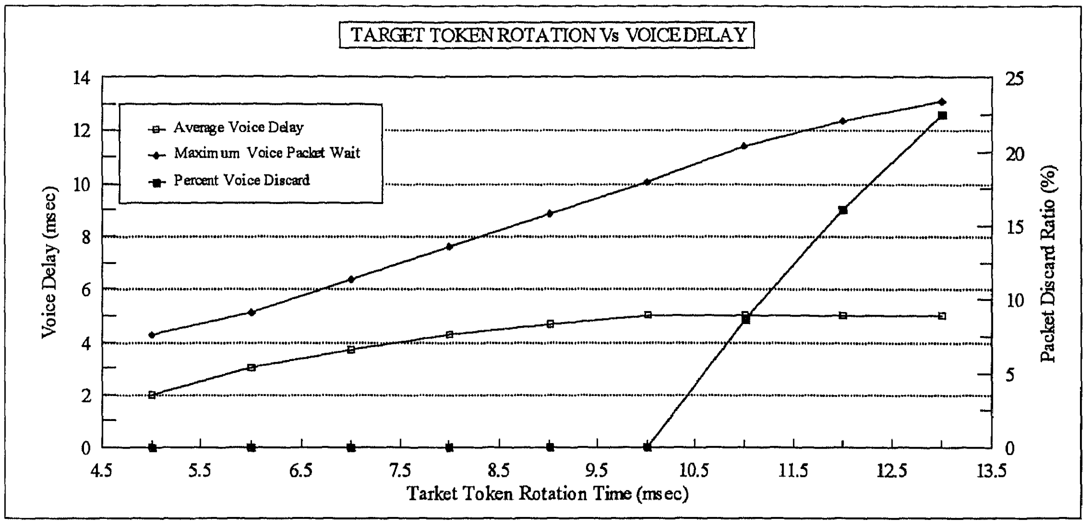


Figure 5-11. Target token rotation vs voice delay.

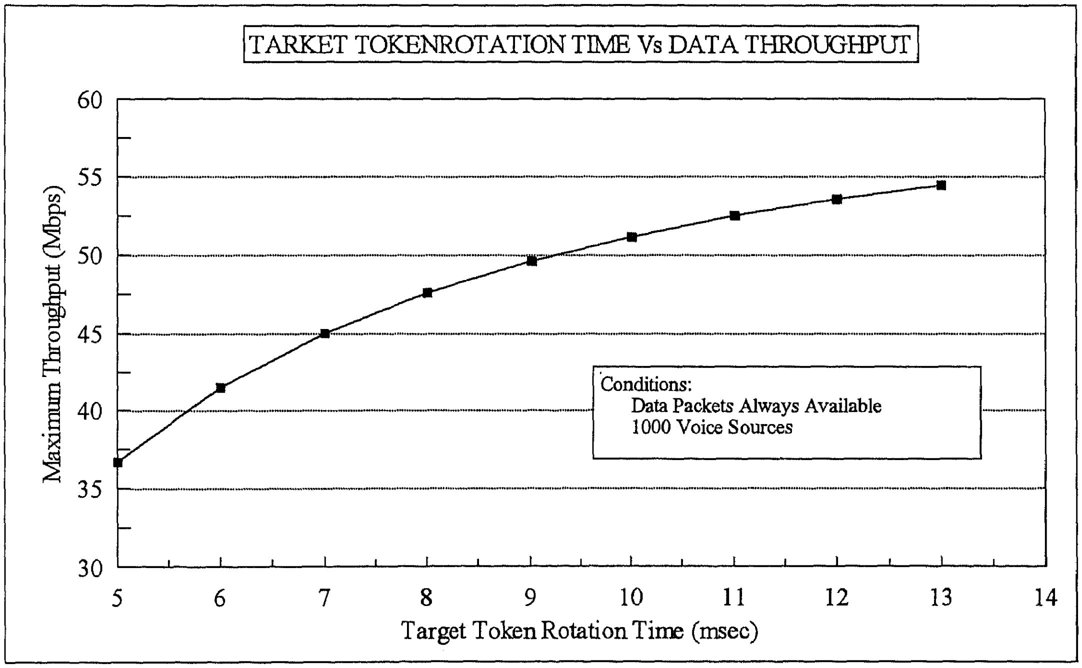
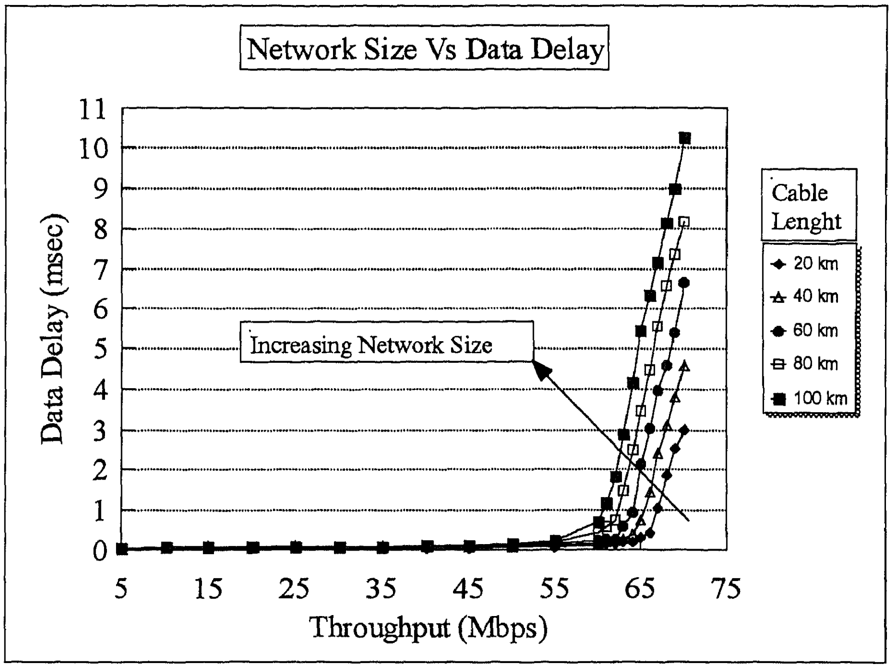


Figure 5-12. Target token rotation time vs data throughput.

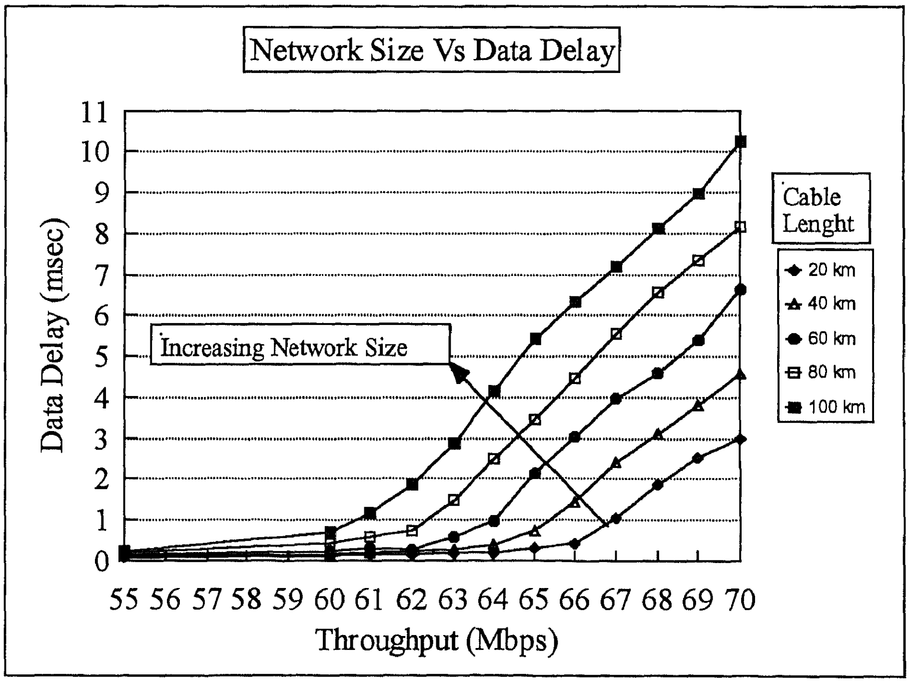
5.4 Effect of network size on throughput and data delay

Figures 5-13 and 5-14, show the effect of increasing the network's size from the reference value of 20 km to a maximum value of 100 km. The FDDI performance deteriorates considerably, however, because of increasing latency (TTRT has been kept fixed at 5 msec). It is thus clear that FDDI rings cannot be made too large.

When a station receives a packet there is a delay before that packet is retransmitted to the next station on the ring. There is overhead for the serial to parallel conversion and the 5 bit to 4 bit decoding. There is also additional overhead for the address comparison. This delay is known as the station latency. The maximum throughput increases as the station latency decreases. This is to be expected, since the value of the Token Holding Timer THT at the time of arrival of the token, determines the amount of data traffic that can be transmitted at a station. As latency increases, THT is reduced correspondingly and thus throughput is decreased.



Figures 5-13. Network size vs data delay.



Figures 5-14. Network size vs data delay (zoom).

CHAPTER 6

CONCLUSIONS AND FUTURE FDDI STANDARDS

The characteristic differences in the nature of voice and data traffic pose a problem for most network configurations. Data traffic needs to be transmitted with very high reliability and lower bit error rates (BER). For the FDDI network the BER for data traffic is specified as 1 x 10^{-12} with an optical power loss of 11 dB between stations. But the packet delay time is not necessarily a critical factor for data transmission unless it increase to an unreasonable or infinite delay. Voice traffic, on the other hand, can tolerate bit errors to the point where several percent of the voice packets are lost and the transmission will still be intelligible. Delay time for voice packets, however, is critical and high delay times will be noticeable to the user. Simulation results for this thesis indicate that the FDDI approximate analysis tends to overestimate the access delay [24]. It thus provides a conservative estimate of the delay, which makes it useful to evaluate the impact of various parameters on the design performances. These parameters include average frame length and distribution, choice of T_Opr, ring length, number of stations, latency per station etc.

It can be concluded that the FDDI standard has addressed the transmission issues for voice and data traffic. The use of the target token rotation timer and higher priority for voice traffic enables a FDDI network to transmit a reasonable amount of voice traffic and also provide data throughput greater than most copper based LANs. The transmission distance for FDDI is also considerably longer than a copper based LAN. With the addition of single mode (Laser based) FDDI products, for which the ANSI standard exists, the distance between stations can be increased up to 60 Km long. This makes the FDDI standard a viable network for long distance, high bandwidth applications which require both voice and data communications.

Since end users seem to have a great need for bandwidth, FDDI's prospects look promising. The ANSI X.3 committee has been working on FDDI enhancements for some time. A category of enhancements, known as FDDI-II [20, 21, 22] which is a successor of FDDI, adds circuit-switched capabilities to the basic packet-switched services. These capabilities are often referred to as isochronous communication and are intended for voice and video applications. Only one commercial product currently supports FDDI-II, a Network Interface Card (NIC) from MultiMedia LAN's Inc, Charlotte, N.C.. This type of products has led some analysts to speculate that other technologies, such as Asynchronous Transfer Mode (ATM), a broadband Integrated Services Digital Network (ISDN) technology, may emerge as the preferred high-bandwidth transport medium.

APPENDIX A

TABLES OF OUTPUT RESULTS

Table 1. Packet delay vs network throughput (100 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
32.5	1.28	1.23
43.63	1.44	1.41
50.45	1.69	1.64
55.26	2.01	1.98
58.68	2.31	2.05
61.05	3.03	2.25
62.76	4.64	2.33
63.68	14.74	2.44

Table 2. Packet delay vs network throughput (200 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
27.5	1.19	1.12
39.88	1.48	1.45
47.24	1.73	1.68
52.11	1.96	1.93
55.53	2.24	2.09
58.03	2.81	2.24
59.61	4.68	2.33
60.52	6.9	2.4

Table 3. Packet delay vs network throughput (300 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
21.84	1.13	1.06
36.05	1.44	1.38
43.82	1.75	1.69
48.95	1.94	1.88
52.37	2.26	2.09
54.88	2.81	2.25
56.53	4.38	2.38
57.5	7.73	2.44

Table 4. Packet delay vs network throughput (400 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
15.29	1.11	1.07
32.21	1.44	1.41
40.3	1.7	1.67
45.59	1.93	1.85
49.18	2.3	2.07
51.68	2.74	2.27
53.38	3.33	2.3
54.41	4.93	2.37

Table 5. Packet delay vs network throughput (500 Voice sources).

Throughput (Mbps)	Data delay (msec)	Voice Delay (msec)
27.85	1.43	1.36
36.87	1.71	1.69
42.33	1.96	1.93
46.04	2.21	2.07
48.63	3.14	2.29
50.26	4.36	2.36
51.3	8.29	2.43

Table 6. Packet delay vs network throughput (600 Voice sources).

Throughput (Mbps)	Data delay (msec)	Voice Delay (msec)
21.98	1.46	1.43
33.37	1.72	1.7
39.07	1.93	1.89
42.91	2.22	2.06
45.47	3.02	2.3
47.09	5.36	2.37
48.14	9.19	2.44

Table 7. Packet delay vs network throughput (700 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
18.02	1.33	1.31
29.65	1.62	1.61
35.81	1.93	1.9
39.77	2.22	2.11
42.33	2.67	2.22
44.19	3.44	2.33
45.1	5.89	2.43
45.7	13.33	2.47

Table 8. Packet delay vs network throughput (800 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
25.79	1.58	1.57
32.45	1.88	1.86
36.6	2.18	1.97
39.29	2.86	2.22
40.98	5.71	2.35
42.04	7.59	2.41

Table 9. Packet delay vs network throughput (900 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
21.65	1.56	1.55
29.13	1.78	1.76
33.31	2.22	2.01
36.16	2.67	2.22
37.89	3.97	2.41
38.97	5.33	2.45
39.46	13.33	2.51

Table 10. Packet delay vs network throughput (1000 Voice sources).

Throughput (Mbps)	**Data delay (msec)**	**Voice Delay (msec)**
16.38	1.55	1.51
25.63	1.86	1.8
30.16	2.12	2.02
32.19	2.59	2.22
34.76	4.24	2.38
35.81	6.98	2.43

Table 11. Target token rotation vs voice delay.

TTRT (msec)	Average Voice Delay (msec)	Maximum Voice Packet Wait (msec)	Packet Discard Ratio (%)
5	2.01	4.23	0
6	3.03	5.12	0
7	3.71	6.38	0
8	4.23	7.58	0
9	4.65	8.86	0
10	4.97	10.04	0
11	4.97	11.38	8.57
12	4.97	12.39	16.07
13	4.97	13.11	22.5

Table 12. Target token rotation time vs data throughput.

TTRT (msec)	Maximum Throughput (Mbps)
5	36.7
6	41.52
7	45.01
8	47.56
9	49.57
10	51.15
11	52.5
12	53.6
13	54.51

Table 13. Network size vs data delay.

Throughput	20 km	40 km	60 km	80 km	100 km
5	0.02	0.03	0.03	0.04	0.05
10	0.02	0.03	0.04	0.05	0.05
15	0.02	0.03	0.04	0.05	0.06
20	0.02	0.03	0.04	0.05	0.06
25	0.03	0.04	0.05	0.06	0.07
30	0.03	0.04	0.05	0.07	0.08
35	0.03	0.05	0.06	0.08	0.09
40	0.04	0.06	0.07	0.09	0.11
45	0.04	0.07	0.09	0.1	0.12
50	0.06	0.08	0.11	0.13	0.16
55	0.08	0.11	0.14	0.17	0.22
60	0.12	0.16	0.22	0.4	0.69
61	0.13	0.19	0.29	0.58	1.15
62	0.14	0.23	0.27	0.75	1.84
63	0.17	0.25	0.59	1.46	2.87
64	0.2	0.37	0.94	2.48	4.15
65	0.31	0.74	2.12	3.44	5.43
66	0.42	1.44	3.04	4.45	6.33
67	1.03	2.41	3.97	5.54	7.16
68	1.85	3.1	4.56	6.55	8.11
69	2.51	3.8	5.39	7.33	8.96
70	3	4.58	6.66	8.17	10.24

APPENDIX B

NETWORK II.5 PROGRAM CODE

```
* UNTITLED
***** NETGIN RELEASE  7.02  FILE SAVED 05/06/1994 19:34:13

***** GLOBAL VARIABLES
GLOBAL FLAGS =
 TEXT SCALE FACTOR = 3  24
 DIAGRAM BOUNDARIES = 0.        336.000     100.800       0.
 ANTITHETIC VARIATE = NO
 RANDOMIZER =  0
 CLOCK = YES
 CLOCK INCREMENT =   .002000000 SEC
 BATCH = YES
 INPUT LISTING = NO
 DEFAULT LISTING = NO
 LENGTH =   .100000000 SEC
 PLOT DATA FILE = YES
 PLOT START =  0.         SEC
 PLOT STOP =   .100000000 SEC
 GRAPH1 = PE 3
 GRAPH2 = PE 15
 RUNTIME WARNINGS = TERMINAL

***** STATISTICAL DISTRIBUTION FUNCTIONS
STATISTICAL DISTRIBUTIONS =
 NAME = DATA LENGTH
  TYPE = UNIFORM
   LOWER.BOUND =   1000.000
   UPPER.BOUND =  15000.000
 NAME = DISTANCE DELAY
  TYPE = UNIFORM
   LOWER.BOUND =     25.000
   UPPER.BOUND =     25.000
 NAME = T_OPR
  TYPE = UNIFORM
   LOWER.BOUND =   5000.000
   UPPER.BOUND =   5000.000
 NAME = EXPONENTIAL
  TYPE = EXPONENTIAL
   MEAN =  10000.000
   UPPER.BOUND =  15000.000
   STREAM =   13

***** PROCESSING ELEMENTS
HARDWARE TYPE = PROCESSING
 NAME = PE 1
```

```
  LOCATION = 10.000        6.000
  STYLE/COLORS = 1   15     3
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
    INSTRUCTION TYPE = MESSAGE
      NAME ; SEND VOICE
        LENGTH ;  640.0 BITS
        MESSAGE ; VOICE 1
        DESTINATION PROCESSOR ; PE 3
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
      NAME ; SEND DATA
        LENGTH ; DATA LENGTH
        MESSAGE ; DATA 1
        DESTINATION PROCESSOR ; PE 3
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
NAME = PE 2
  LOCATION = 26.000        19.000
  STYLE/COLORS = 1   13     5
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
    INSTRUCTION TYPE = MESSAGE
      NAME ; SEND VOICE
        LENGTH ;  640.0 BITS
        MESSAGE ; VOICE 2
        DESTINATION PROCESSOR ; PE 4
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
      NAME ; SEND DATA
        LENGTH ; DATA LENGTH
        MESSAGE ; DATA 2
        DESTINATION PROCESSOR ; PE 4
        QUEUE FLAG ; YES
```

```
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 3
 LOCATION = 42.000       6.000
 STYLE/COLORS = 1   11     7
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 3
    DESTINATION PROCESSOR ; PE 5
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 3
    DESTINATION PROCESSOR ; PE 5
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 4
 LOCATION = 58.000       19.000
 STYLE/COLORS = 1    7    11
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 4
    DESTINATION PROCESSOR ; PE 6
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
```

```
      NAME ; SEND DATA
        LENGTH ; DATA LENGTH
        MESSAGE ; DATA 4
        DESTINATION PROCESSOR ; PE 6
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
NAME = PE 5
  LOCATION = 74.000        6.000
  STYLE/COLORS = 1    5     13
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
    INSTRUCTION TYPE = MESSAGE
      NAME ; SEND VOICE
        LENGTH ;  640.0 BITS
        MESSAGE ; VOICE 5
        DESTINATION PROCESSOR ; PE 7
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
      NAME ; SEND DATA
        LENGTH ; DATA LENGTH
        MESSAGE ; DATA 5
        DESTINATION PROCESSOR ; PE 7
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
NAME = PE 6
  LOCATION = 90.000        19.000
  STYLE/COLORS = 1    3     15
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
    INSTRUCTION TYPE = MESSAGE
      NAME ; SEND VOICE
        LENGTH ;  640.0 BITS
        MESSAGE ; VOICE 6
        DESTINATION PROCESSOR ; PE 8
```

```
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
      NAME ; SEND DATA
        LENGTH ; DATA LENGTH
        MESSAGE ; DATA 6
        DESTINATION PROCESSOR ; PE 8
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
NAME = PE 7
  LOCATION = 106.000       6.000
  STYLE/COLORS = 1   15     3
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
    INSTRUCTION TYPE = MESSAGE
      NAME ; SEND VOICE
        LENGTH ;  640.0 BITS
        MESSAGE ; VOICE 7
        DESTINATION PROCESSOR ; PE 9
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
      NAME ; SEND DATA
        LENGTH ; DATA LENGTH
        MESSAGE ; DATA 7
        DESTINATION PROCESSOR ; PE 9
        QUEUE FLAG ; YES
        RESUME FLAG ; NO
        INHIBIT MESSAGE TO SELF ; NO
        ALLOWABLE BUSSES ;
          FDDI
NAME = PE 8
  LOCATION = 122.000       19.000
  STYLE/COLORS = 1   13     5
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
```

```
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 8
    DESTINATION PROCESSOR ; PE 10
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 8
    DESTINATION PROCESSOR ; PE 10
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 9
 LOCATION = 138.000        6.000
 STYLE/COLORS = 1   11      7
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 9
    DESTINATION PROCESSOR ; PE 11
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 9
    DESTINATION PROCESSOR ; PE 11
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 10
```

```
 LOCATION = 154.000       19.000
 STYLE/COLORS = 1   7    11
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 10
    DESTINATION PROCESSOR ; PE 12
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 10
    DESTINATION PROCESSOR ; PE 12
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 11
 LOCATION = 170.000        6.000
 STYLE/COLORS = 1   5    13
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 11
    DESTINATION PROCESSOR ; PE 13
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 11
    DESTINATION PROCESSOR ; PE 13
    QUEUE FLAG ; YES
```

```
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 12
 LOCATION = 186.000       19.000
 STYLE/COLORS = 1    3     15
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 12
    DESTINATION PROCESSOR ; PE 14
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 12
    DESTINATION PROCESSOR ; PE 14
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 13
 LOCATION = 202.000        6.000
 STYLE/COLORS = 1   15     3
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 13
    DESTINATION PROCESSOR ; PE 15
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
```

```
    NAME ; SEND DATA
      LENGTH ; DATA LENGTH
      MESSAGE ; DATA 13
      DESTINATION PROCESSOR ; PE 15
      QUEUE FLAG ; YES
      RESUME FLAG ; NO
      INHIBIT MESSAGE TO SELF ; NO
      ALLOWABLE BUSSES ;
       FDDI
NAME = PE 14
  LOCATION = 218.000      19.000
  STYLE/COLORS = 1   13     5
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
    INSTRUCTION TYPE = MESSAGE
    NAME ; SEND VOICE
      LENGTH ;  640.0 BITS
      MESSAGE ; VOICE 14
      DESTINATION PROCESSOR ; PE 16
      QUEUE FLAG ; YES
      RESUME FLAG ; NO
      INHIBIT MESSAGE TO SELF ; NO
      ALLOWABLE BUSSES ;
       FDDI
    NAME ; SEND DATA
      LENGTH ; DATA LENGTH
      MESSAGE ; DATA 14
      DESTINATION PROCESSOR ; PE 16
      QUEUE FLAG ; YES
      RESUME FLAG ; NO
      INHIBIT MESSAGE TO SELF ; NO
      ALLOWABLE BUSSES ;
       FDDI
NAME = PE 15
  LOCATION = 234.000       6.000
  STYLE/COLORS = 1   11     7
  BASIC CYCLE TIME = 1.0 MIC
  INPUT CONTROLLER = YES
  INSTRUCTION REPERTOIRE =
    INSTRUCTION TYPE = MESSAGE
    NAME ; SEND VOICE
      LENGTH ;  640.0 BITS
      MESSAGE ; VOICE 15
      DESTINATION PROCESSOR ; PE 17
```

```
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 15
    DESTINATION PROCESSOR ; PE 17
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 16
 LOCATION = 250.000       19.000
 STYLE/COLORS = 1    7     11
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 16
    DESTINATION PROCESSOR ; PE 18
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 16
    DESTINATION PROCESSOR ; PE 18
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 17
 LOCATION = 266.000        6.000
 STYLE/COLORS = 1    5     13
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
```

```
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 17
    DESTINATION PROCESSOR ; PE 19
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 17
    DESTINATION PROCESSOR ; PE 19
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 18
 LOCATION = 282.000       19.000
 STYLE/COLORS = 1    3     15
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 18
    DESTINATION PROCESSOR ; PE 20
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 18
    DESTINATION PROCESSOR ; PE 20
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 19
```

```
 LOCATION = 298.000        6.000
 STYLE/COLORS = 1   15     3
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 19
    DESTINATION PROCESSOR ; PE 1
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 19
    DESTINATION PROCESSOR ; PE 1
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
NAME = PE 20
 LOCATION = 314.000       19.000
 STYLE/COLORS = 1   13     5
 BASIC CYCLE TIME = 1.0 MIC
 INPUT CONTROLLER = YES
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME ; SEND VOICE
    LENGTH ;  640.0 BITS
    MESSAGE ; VOICE 20
    DESTINATION PROCESSOR ; PE 2
    QUEUE FLAG ; YES
    RESUME FLAG ; NO
    INHIBIT MESSAGE TO SELF ; NO
    ALLOWABLE BUSSES ;
     FDDI
   NAME ; SEND DATA
    LENGTH ; DATA LENGTH
    MESSAGE ; DATA 20
    DESTINATION PROCESSOR ; PE 2
    QUEUE FLAG ; YES
```

```
     RESUME FLAG ; NO
     INHIBIT MESSAGE TO SELF ; NO
     ALLOWABLE BUSSES ;
      FDDI

***** TRANSFER DEVICES
HARDWARE TYPE = DATA TRANSFER
 NAME = FDDI
  DRAW TYPE = BUS
  NAME/MSG LOCATION = 333.000      11.000     333.000      17.000
  SEGMENTS =  2
   10.000       15.000
   330.000       15.000
  PROTOCOL = PRIORITY TOKEN RING
  CYCLE TIME = .008 MIC
  BITS PER CYCLE =          1
  CYCLES PER WORD =          4
  WORDS PER BLOCK =       3200
  WORD OVERHEAD TIME = .008 MIC
  BLOCK OVERHEAD TIME = 200.0 MIC
  BUS CONNECTIONS =
   PE 1
    SEGMENTS = 18.000       15.000       18.000       11.000
    STYLE/WIDTH = 1   60
    SYNCHRONOUS = 6.4 MIC
    MINIMUM SYNCHRONOUS PRIORITY =        10
    TARGET TOKEN ROTATION TIMES =
     PRIORITY     9 = 13000.0 MIC
     PRIORITY     8 = 12000.0 MIC
     PRIORITY     7 = 11000.0 MIC
     PRIORITY     6 = 10000.0 MIC
     PRIORITY     5 = 9000.0 MIC
     PRIORITY     4 = 8000.0 MIC
     PRIORITY     3 = 7000.0 MIC
     PRIORITY     2 = 6000.0 MIC
     PRIORITY     1 = 5000.0 MIC
   PE 2
    SEGMENTS = 34.000       15.000       34.000       19.000
    STYLE/WIDTH = 1   60
    SYNCHRONOUS = 6.4 MIC
    MINIMUM SYNCHRONOUS PRIORITY =        10
    TARGET TOKEN ROTATION TIMES =
     PRIORITY     9 = 13000.0 MIC
     PRIORITY     8 = 12000.0 MIC
     PRIORITY     7 = 11000.0 MIC
```

```
   PRIORITY    6 = 10000.0 MIC
   PRIORITY    5 = 9000.0 MIC
   PRIORITY    4 = 8000.0 MIC
   PRIORITY    3 = 7000.0 MIC
   PRIORITY    2 = 6000.0 MIC
   PRIORITY    1 = 5000.0 MIC
PE 3
  SEGMENTS = 50.000     15.000     50.000     11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =       10
  TARGET TOKEN ROTATION TIMES =
   PRIORITY    9 = 13000.0 MIC
   PRIORITY    8 = 12000.0 MIC
   PRIORITY    7 = 11000.0 MIC
   PRIORITY    6 = 10000.0 MIC
   PRIORITY    5 = 9000.0 MIC
   PRIORITY    4 = 8000.0 MIC
   PRIORITY    3 = 7000.0 MIC
   PRIORITY    2 = 6000.0 MIC
   PRIORITY    1 = 5000.0 MIC
PE 4
  SEGMENTS = 66.000     15.000     66.000     19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =       10
  TARGET TOKEN ROTATION TIMES =
   PRIORITY    9 = 13000.0 MIC
   PRIORITY    8 = 12000.0 MIC
   PRIORITY    7 = 11000.0 MIC
   PRIORITY    6 = 10000.0 MIC
   PRIORITY    5 = 9000.0 MIC
   PRIORITY    4 = 8000.0 MIC
   PRIORITY    3 = 7000.0 MIC
   PRIORITY    2 = 6000.0 MIC
   PRIORITY    1 = 5000.0 MIC
PE 5
  SEGMENTS = 82.000     15.000     82.000     11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =       10
  TARGET TOKEN ROTATION TIMES =
   PRIORITY    9 = 13000.0 MIC
   PRIORITY    8 = 12000.0 MIC
   PRIORITY    7 = 11000.0 MIC
```

```
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 6
  SEGMENTS = 98.000       15.000       98.000       19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 7
  SEGMENTS = 114.000       15.000       114.000       11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 8
  SEGMENTS = 130.000       15.000       130.000       19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
```

```
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 9
  SEGMENTS = 146.000       15.000       146.000       11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 10
  SEGMENTS = 162.000       15.000       162.000       19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 11
  SEGMENTS = 178.000       15.000       178.000       11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
```

```
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 12
  SEGMENTS = 194.000       15.000       194.000       19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 13
  SEGMENTS = 210.000       15.000       210.000       11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 14
  SEGMENTS = 226.000       15.000       226.000       19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =        10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
```

```
   PRIORITY    6 = 10000.0 MIC
   PRIORITY    5 = 9000.0 MIC
   PRIORITY    4 = 8000.0 MIC
   PRIORITY    3 = 7000.0 MIC
   PRIORITY    2 = 6000.0 MIC
   PRIORITY    1 = 5000.0 MIC
PE 15
  SEGMENTS = 242.000      15.000      242.000      11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =       10
  TARGET TOKEN ROTATION TIMES =
   PRIORITY    9 = 13000.0 MIC
   PRIORITY    8 = 12000.0 MIC
   PRIORITY    7 = 11000.0 MIC
   PRIORITY    6 = 10000.0 MIC
   PRIORITY    5 = 9000.0 MIC
   PRIORITY    4 = 8000.0 MIC
   PRIORITY    3 = 7000.0 MIC
   PRIORITY    2 = 6000.0 MIC
   PRIORITY    1 = 5000.0 MIC
PE 16
  SEGMENTS = 258.000      15.000      258.000      19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =       10
  TARGET TOKEN ROTATION TIMES =
   PRIORITY    9 = 13000.0 MIC
   PRIORITY    8 = 12000.0 MIC
   PRIORITY    7 = 11000.0 MIC
   PRIORITY    6 = 10000.0 MIC
   PRIORITY    5 = 9000.0 MIC
   PRIORITY    4 = 8000.0 MIC
   PRIORITY    3 = 7000.0 MIC
   PRIORITY    2 = 6000.0 MIC
   PRIORITY    1 = 5000.0 MIC
PE 17
  SEGMENTS = 274.000      15.000      274.000      11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =       10
  TARGET TOKEN ROTATION TIMES =
   PRIORITY    9 = 13000.0 MIC
   PRIORITY    8 = 12000.0 MIC
   PRIORITY    7 = 11000.0 MIC
```

```
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 18
  SEGMENTS = 290.000       15.000       290.000       19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =         10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 19
  SEGMENTS = 306.000       15.000       306.000       11.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =         10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
    PRIORITY     6 = 10000.0 MIC
    PRIORITY     5 = 9000.0 MIC
    PRIORITY     4 = 8000.0 MIC
    PRIORITY     3 = 7000.0 MIC
    PRIORITY     2 = 6000.0 MIC
    PRIORITY     1 = 5000.0 MIC
PE 20
  SEGMENTS = 322.000       15.000       322.000       19.000
  STYLE/WIDTH = 1   60
  SYNCHRONOUS = 6.4 MIC
  MINIMUM SYNCHRONOUS PRIORITY =         10
  TARGET TOKEN ROTATION TIMES =
    PRIORITY     9 = 13000.0 MIC
    PRIORITY     8 = 12000.0 MIC
    PRIORITY     7 = 11000.0 MIC
```

```
     PRIORITY     6 = 10000.0 MIC
     PRIORITY     5 = 9000.0 MIC
     PRIORITY     4 = 8000.0 MIC
     PRIORITY     3 = 7000.0 MIC
     PRIORITY     2 = 6000.0 MIC
     PRIORITY     1 = 5000.0 MIC

***** MODULES
SOFTWARE TYPE = MODULE
 NAME = TVOICE 1
  PRIORITY =          1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  ITERATION PERIOD = EXPONENTIAL
  DELAY = DISTANCE DELAY
  RESIDENT PROCESSORS =
   PE 1
  INSTRUCTION LIST =
   EXECUTE A TOTAL OF ;  1.0 SEND VOICE
  ANDED SUCCESSORS =
   CHAIN TO ; TDATA 1
 NAME = TDATA 1
  PRIORITY =          1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  INSTRUCTION LIST =
   EXECUTE A TOTAL OF ;  1.0 SEND DATA
  ANDED PREDECESSOR LIST =
   TVOICE 1
 NAME = TVOICE 2
  PRIORITY =          1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  ITERATION PERIOD = EXPONENTIAL
  DELAY = DISTANCE DELAY
  RESIDENT PROCESSORS =
   PE 2
  INSTRUCTION LIST =
   EXECUTE A TOTAL OF ;  1.0 SEND VOICE
  ANDED SUCCESSORS =
   CHAIN TO ; TDATA 2
 NAME = TDATA 2
  PRIORITY =          1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
```

```
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 2
NAME = TVOICE 3
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 3
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 3
NAME = TDATA 3
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 3
NAME = TVOICE 4
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 4
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 4
NAME = TDATA 4
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 4
NAME = TVOICE 5
```

```
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 5
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 5
NAME = TDATA 5
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 5
NAME = TVOICE 6
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 6
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 6
NAME = TDATA 6
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 6
NAME = TVOICE 7
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
```

```
 RESIDENT PROCESSORS =
  PE 7
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 7
NAME = TDATA 7
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 7
NAME = TVOICE 8
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 8
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 8
NAME = TDATA 8
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 8
NAME = TVOICE 9
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 9
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
```

```
  CHAIN TO ; TDATA 9
NAME = TDATA 9
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 9
NAME = TVOICE 10
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 10
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 10
NAME = TDATA 10
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 10
NAME = TVOICE 11
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 11
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 11
NAME = TDATA 11
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
```

```
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 11
NAME = TVOICE 12
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 12
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 12
NAME = TDATA 12
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 12
NAME = TVOICE 13
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 13
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 13
NAME = TDATA 13
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 13
NAME = TVOICE 14
```

```
 PRIORITY =        1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 14
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 14
NAME = TDATA 14
 PRIORITY =        1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 14
NAME = TVOICE 15
 PRIORITY =        1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 15
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 15
NAME = TDATA 15
 PRIORITY =        1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 15
NAME = TVOICE 16
 PRIORITY =        1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
```

```
 RESIDENT PROCESSORS =
  PE 16
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 16
NAME = TDATA 16
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 16
NAME = TVOICE 17
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 17
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
  CHAIN TO ; TDATA 17
NAME = TDATA 17
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND DATA
 ANDED PREDECESSOR LIST =
  TVOICE 17
NAME = TVOICE 18
 PRIORITY =          1
 INTERRUPTIBILITY FLAG = NO
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = EXPONENTIAL
 DELAY = DISTANCE DELAY
 RESIDENT PROCESSORS =
  PE 18
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF ;  1.0 SEND VOICE
 ANDED SUCCESSORS =
```

```
    CHAIN TO ; TDATA 18
NAME = TDATA 18
  PRIORITY =         1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  INSTRUCTION LIST =
    EXECUTE A TOTAL OF ;  1.0 SEND DATA
  ANDED PREDECESSOR LIST =
    TVOICE 18
NAME = TVOICE 19
  PRIORITY =         1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  ITERATION PERIOD = EXPONENTIAL
  DELAY = DISTANCE DELAY
  RESIDENT PROCESSORS =
    PE 19
  INSTRUCTION LIST =
    EXECUTE A TOTAL OF ;  1.0 SEND VOICE
  ANDED SUCCESSORS =
    CHAIN TO ; TDATA 19
NAME = TDATA 19
  PRIORITY =         1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  INSTRUCTION LIST =
    EXECUTE A TOTAL OF ;  1.0 SEND DATA
  ANDED PREDECESSOR LIST =
    TVOICE 19
NAME = TVOICE 20
  PRIORITY =         1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  ITERATION PERIOD = EXPONENTIAL
  DELAY = DISTANCE DELAY
  RESIDENT PROCESSORS =
    PE 20
  INSTRUCTION LIST =
    EXECUTE A TOTAL OF ;  1.0 SEND VOICE
  ANDED SUCCESSORS =
    CHAIN TO ; TDATA 20
NAME = TDATA 20
  PRIORITY =         1
  INTERRUPTIBILITY FLAG = NO
  CONCURRENT EXECUTION = YES
```

```
INSTRUCTION LIST =
 EXECUTE A TOTAL OF ;  1.0 SEND DATA
ANDED PREDECESSOR LIST =
 TVOICE 20
```

REFERENCES

[1] American National Standards for Information Systems, Fiber Distributed Data Interface (FDDI) - Token Ring Physical Layer Protocol (PHY), ANSI X3.144 - 1988, American National Standards Inst. Inc., New York, 1988.

[2] American National Standards for Information Systems, Fiber Distributed Data Interface (FDDI) - Token Ring Medium Access Control (MAC), ANSI X3.139 - 1987, American National Standards Inst. Inc., New York, 1987.

[3] Institute of Electrical and Electronics Engineers, Local and Metropolitan Area Networks, Overview and Architecture IEEE Std 802, 1990.

[4] Institute of Electrical and Electronics Engineers,Token - Ring Access Method and Physical Layer Specifications, IEEE Std 802.5, 1989.

[5] Institute of Electrical and Electronics Engineers, Logical Link Control (LLC), ISO 8802 - 2, IEEE Std 802.2, 1989.

[6] NETWORK II.5 is a registered trademark and service mark of CACI Products Company, 3333 North Torrey Pines Court, La Jolla, California 92037.

[7] W. E. Burr,: "The FDDI Optical Data Link," IEEE Communications Magazine, vol. 25, pp.18 - 23, May 1986.

[8] S. P. Joshi,: "High - Performance Networks - A Focus on the Fiber Distributed Data Interface (FDDI) Standard," IEEE Micro Magazine, vol. 6, pp. 8 - 14, June 1986.

[9] F. E. Ross,: "Rings are Round for Good!," IEEE Network Magazine, vol. 1, pp. 31 - 38, Jan. 1987.

[10] F. E. Ross,: " FDDI - A Tutorial," IEEE Communications Magazine, vol. 24, pp. 10 - 15, May 1986.

[11] American National Standards Institute, Fiber Distributed Data Interface (FDDI) - Token Ring Physical Layer Medium Dependent (PMD), ANSI X3.166, 1990.

[12] International Organization for Standarization, Information Processing Systems - Open Systems Interconnection, Basic Reference Model, ISO 7498, 1984.

[13] The International Telegraph and Telephone Consultative Committee, Recomendation X.200, Blue Book, vol. VIII.4, 1988.

[14] M. J. Johnson,: "Proof That Timing Requirements of the FDDI Token Ring Protocol are Satisfied," IEEE Communications Magazine, vol. 35, no. 6, pp. 620 - 625, June 1987.

[15] R. M. Grow,: "A Timed Token Protocol for Local Area Networks," Proceed. Electro/82, Token Access Protocols, May 1982.

[16] R. Sanker and Y.Y. Yang,: "Performance Analysis of FDDI," IEEE 14th Annual Conference on Local Computer Networks, pp. 328 - 332, 1989.

[17] Accredited Standards Committee X3, FDDI Station Management (SMT), Draft Proposed American National Standard X3T9.5/84-89, June 25, 1992.

[18] American National Standards Institute, Fiber Distributed Data Interface (FDDI) - Single-Mode Fiber Physical Layer Medium Dependent (SMF-PMD), ANSI X3-184, 1991.

[19] F. E. Ross,: "An Overview of FDDI: The Fiber Distributed Data Interface," IEEE Journal on Selected Areas in Communications, vol. 7, no. 7, pp. 1043-1051, Sept. 1989.

[20] C. C. Kessler,: "Inside FDDI-II." LAN Magazine, pp. 117 - 125, Mar. 1991.

[21] M. S. Wolter,: "ANSI X3T9.5 Update: Future FDDI Standards," ConneXions, the Interpretability Report , pp. 21 - 23, Oct. 1991.

[22] R. Halgren,: "FDDI-II to Provide Multimedia Pipe to the Desktop," Lightwave, pp. 40 - 43, Sept. 1992.

[23] R. Jain,: "Performance Analysis of FDDI Token Ring Networks: Effect of parameters and guidelines for setting TTRT," SIGCOMM '90, Philadelphia, pp. 264 - 275, Sept. 1990. Also appears in IEEE LTS Magazine, vol. 2, no. 2, pp. 16 - 22, May 1991.

[24] S. Ghani and M. Schwartz,: "Comparison of DQDB and FDDI MAC Access Protocols," Proc. 16th Conference on Local Computer Networks, Minneapolis, pp. 84 - 95, Oct. 1991.

[25] D. Karvelas and A. Leon-Garcia,: "Performance Analysis of the Medium Access Control Protocol of the FDDI Token Ring Network," IEEE Globecon '88, Hollywood, Fla., pp. 1119 - 1123, Nov. 1988.

[26] S. J. Bigelow, Understanding Telephone Electronics, Third Edition, 1992.

[27] S. Schatt, Understanding Local Area Networks, Third Edition, 1992.